AD8232 Heart Rate Monitor Hookup Guide a

[learn.sparkfun.com tutorial](learn.sparkfun.com)

**Available online at: http://sfe.io/t250**

Contents

Introduction

The [AD8232](AD8232) is a neat little chip used to measure the electrical activity of the heart. This electrical activity can be charted as an ECG or Electrocardiogram. Electrocardiography is used to help diagnose various heart conditions. Now for the disclaimer:

NOTE: This device is not intended to diagnose or treat any conditions.
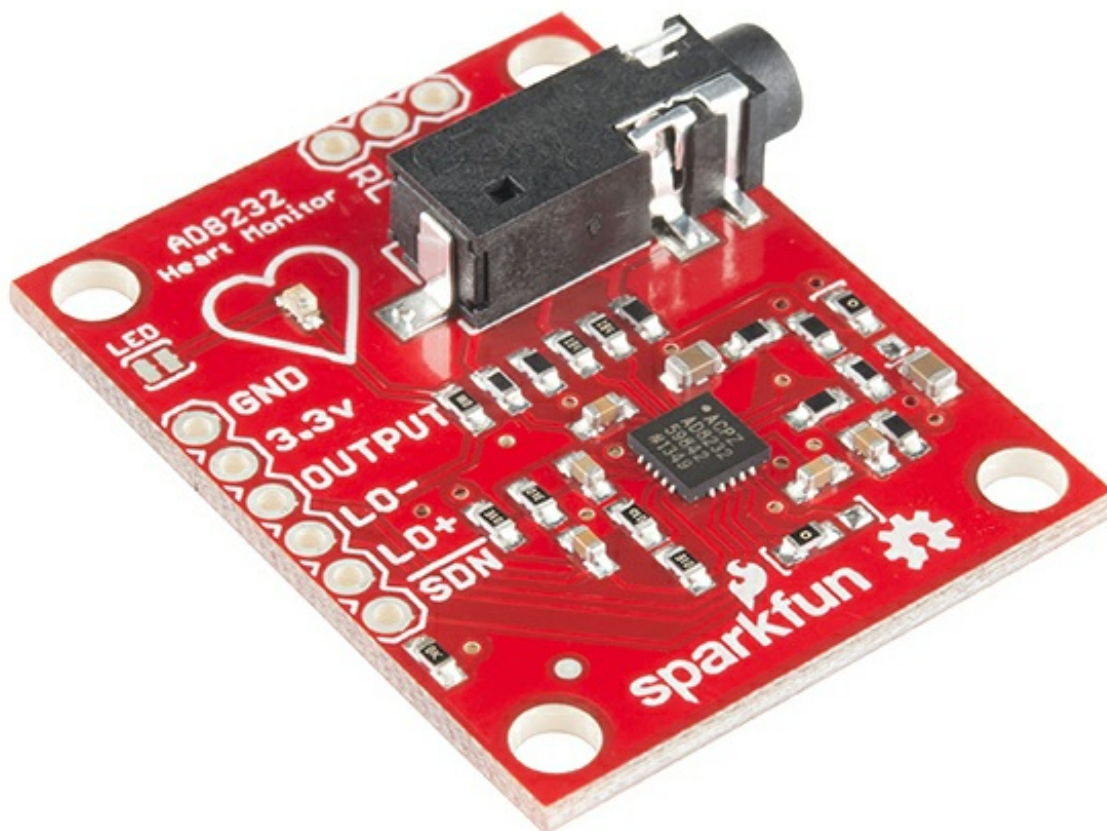
Covered in this Tutorial

In this tutorial, we will go over the basics for getting your AD8232 Heart Rate Monitor up and running. First, an overview of the board and all its features will be presented. Then, we'll show you how hook it up to your favorite microcontroller and how to create visual data using Processing.

Suggested Reading

If you are not familiar with any of the following concepts, you may want to review them before moving ahead.

- [What is an Arduino?](#)
- [Connecting Arduino to Processing](#)
- [How to Power Your Project](#)
- [Serial Communication](#)
- [Serial Terminal Basics](#)
- [Working with Wire](#)

Understanding the ECG

In general terms, lets look at what an ECG is representing and how we're able to sense it. The ECG is separated into two basic Intervals, the PR Interval and the QT Interval, described below.

Note: All information comes from the [Waves and Intervals](#) section of the [ECG Wikipedia page](#) .
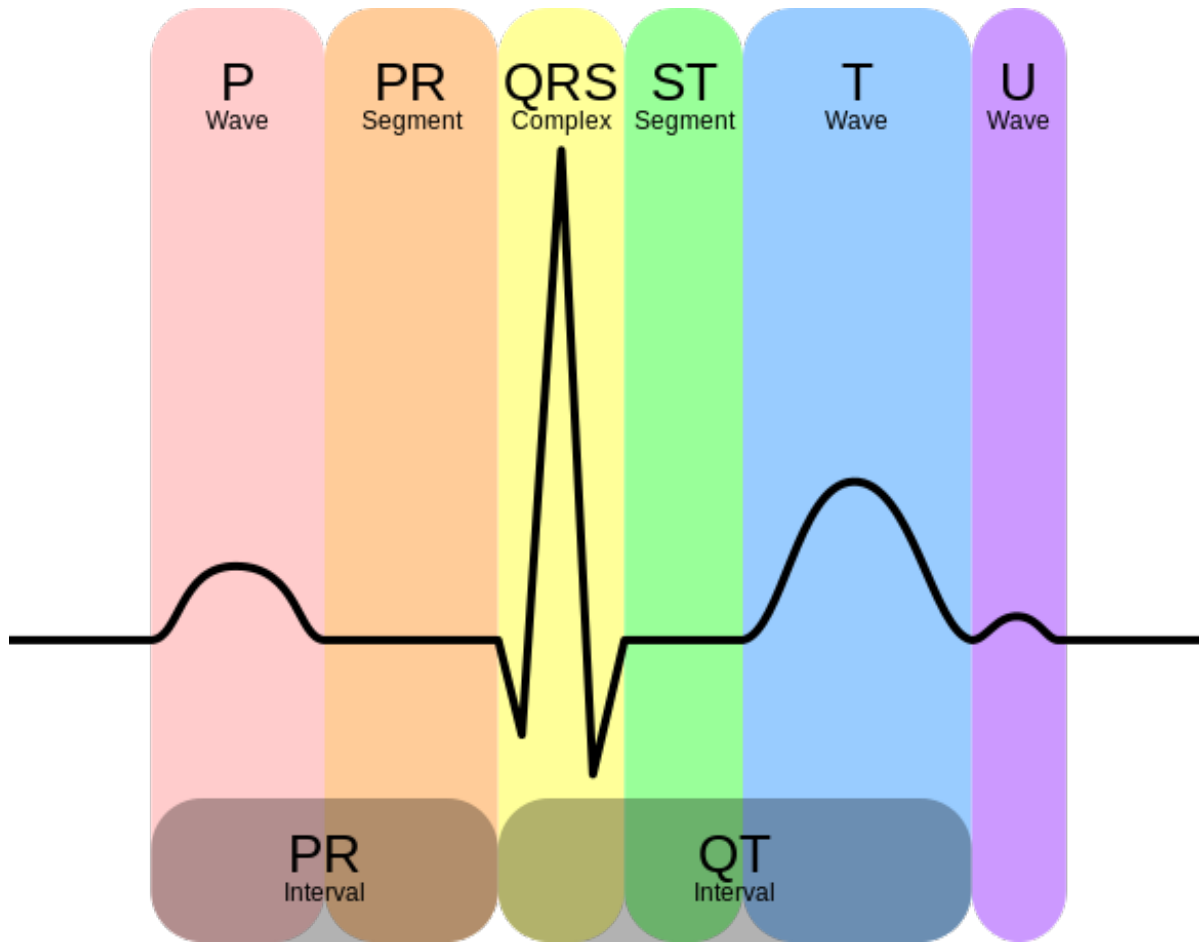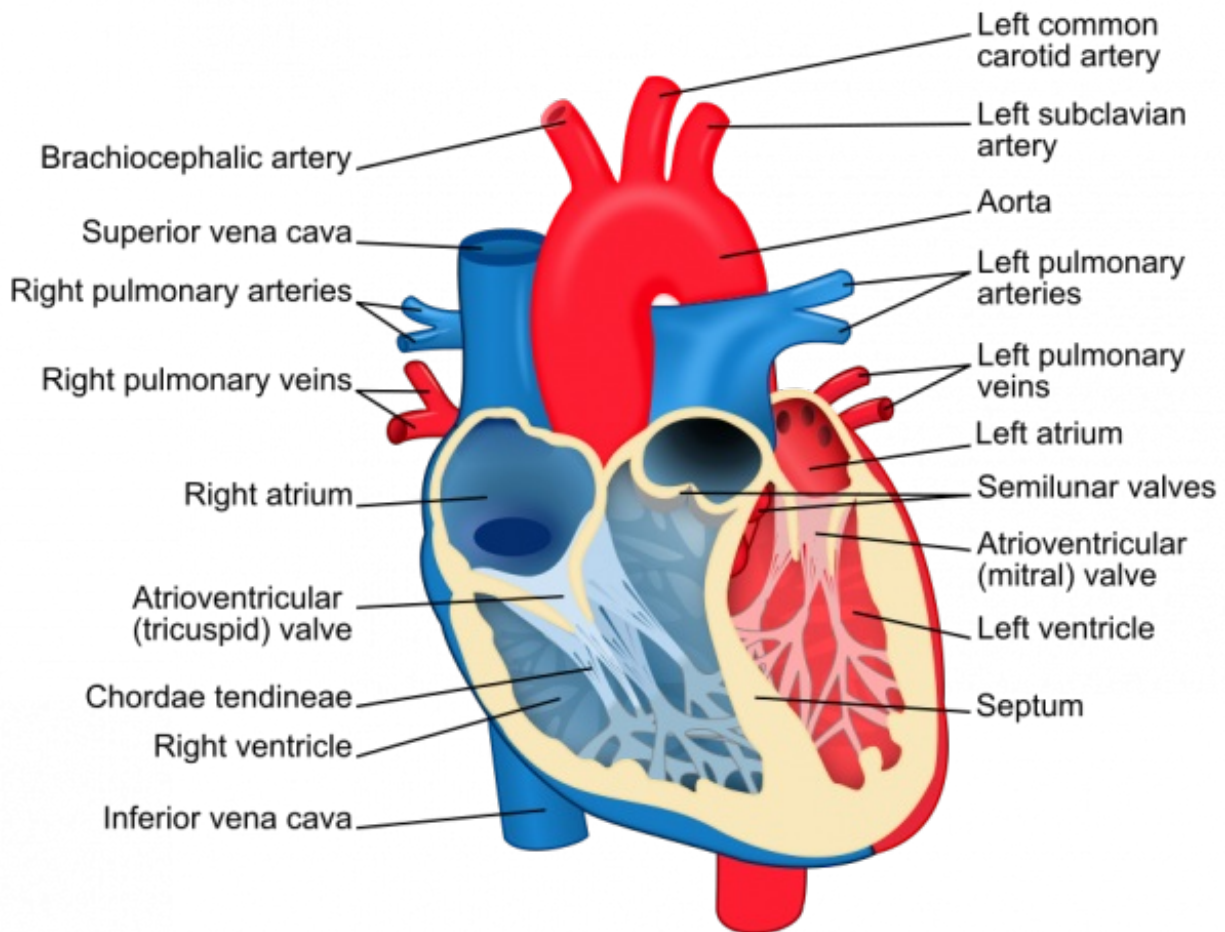
Credit Wikipedia.org

Diagram of the heart. *Credit Wikipedia.org*

PR Interval

The PR interval is the initial wave generated by an electrical impulse traveling from the right atrium to the left. The right atrium is the first chamber to see an electrical impulse. This electrical impulse causes the chambers to "depolarize". This forces it to contract and drain deoxygenated blood from both the Superior and Inferior vena cava into the right ventricle. As the electrical impulse travels across the top of the heart it then triggers the left atrium to contract. The left atrium is responsible for receiving newly oxygenated blood from the lungs into the left ventricle via the left and right pulmonary veins. *The pulmonary veins are red in the diagram because they are carrying oxygenated blood. They are still called veins because veins carry blood towards the heart. Science!*

QT Interval

The QT Interval is where things get really interesting. The QRS is a complex process that generates the signature "beep" in cardiac monitors. During QRS both ventricles begin to pump. The right ventricle begins to pump deoxygenated blood into the lungs through the left and right pulmonary arteries. *The pulmonary arteries are blue in the diagram because they are carrying deoxygenated blood. They are still called arteries because arteries carry blood away the heart. Science, Again!* The left ventricle is also begining to pump freshly oxygenated blood through the aorta and into the rest of the body. After the initial contraction comes the ST segment. The ST

segment is fairly quiet electrically as it is the time where the ventricals waiting to be "re-polarized". Finally the T wave becomes present to actively "re-ploarize", or relax the ventricles. This relaxation phase resets the ventricles to be filled again by the atriums.
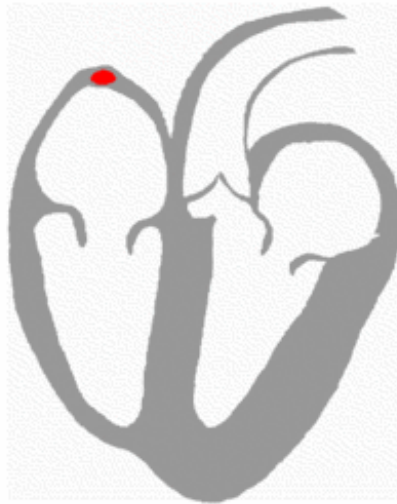
Heartbeat with corresponding ECG *Credit Wikipedia.org*

Connecting the Hardware

In this guide, we'll connect the AD8232 Breakout to an Arduino microcontroller. We will build a simple cardiac monitor that will allow you to measure the electrical activity of the heart in real time!

Materials required for this example

AD8232 Heart Rate Monitor Example SparkFun Wish List

SparkFun USB Mini-B Cable - 6 Foot
◉ CAB-11301

This is a USB 2.0 type A to Mini-B 5-pin cable. You know, the mini-B connector that usually comes with USB H…

SparkFun FTDI Basic Breakout - 3.3V
◉ DEV-09873

This is the newest revision of our [FTDI Basic](http://www.sparkfun.com/commerce/product_info.php?products_…

Jumper Wire Kit - 140pcs
◉ PRT-00124

This is a time saving kit of jumper wires - cut, stripped, and pre-bent for your prototyping pleasure. Included wit…

Break Away Headers - Straight
◉ PRT-00116

A row of headers - break to fit. 40 pins that can be cut to any size. Used with custom PCBs or general custom h…

Breadboard - Self-Adhesive (White)
◉ PRT-12002

This is your tried and true white solderless breadboard. It has 2 power buses, 10 columns, and 30 rows - a tota…

Arduino Pro Mini 328 - 3.3V/8MHz
◉ DEV-11114

It's blue! It's thin! It's the Arduino Pro Mini! SparkFun's minimal design approach to Arduino. This is a 3.3V Ardu…

Pin Connections

The AD8232 Heart Rate Monitor breaks out nine connections from the IC. We traditionally call these connections "pins" because they come from the pins on the IC, but they are actually holes that you can solder wires or header pins to.
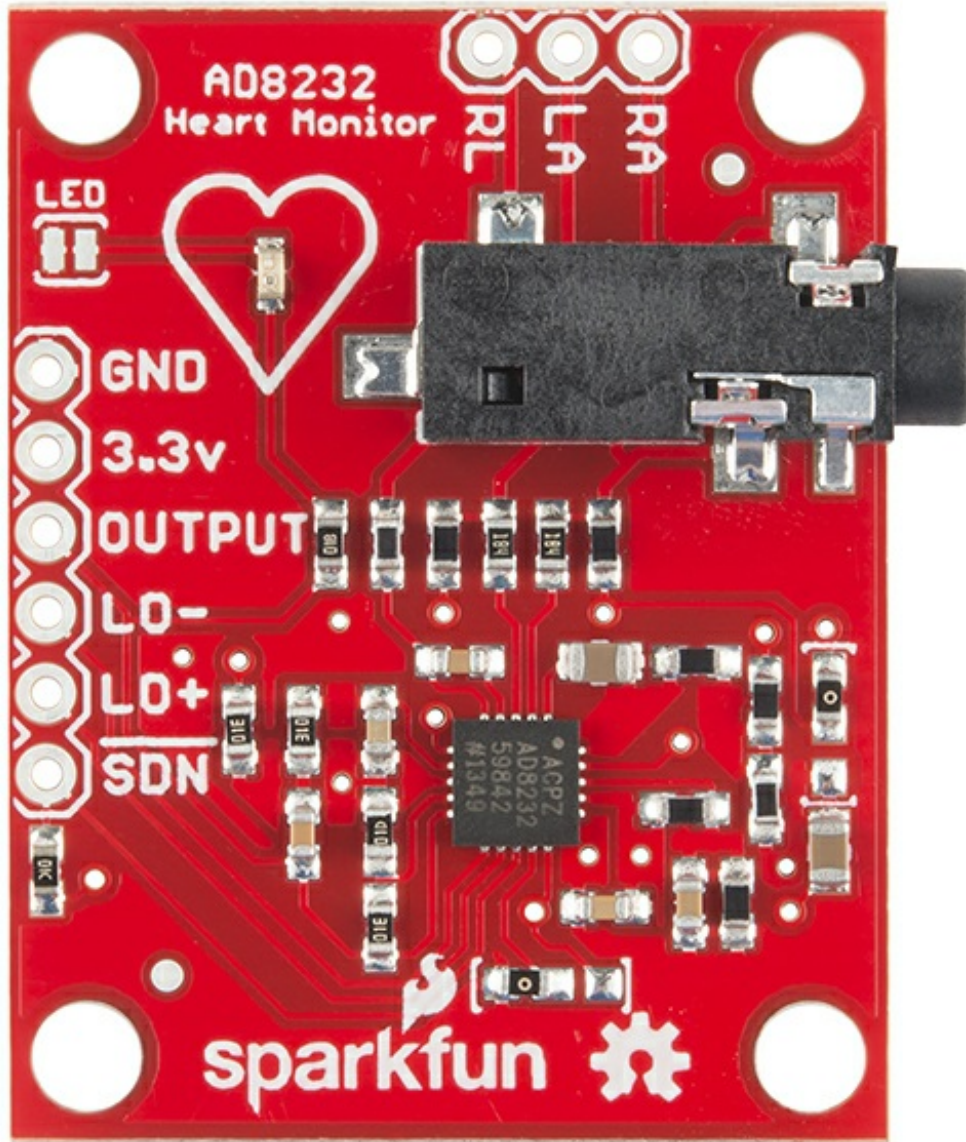
We'll connect five of the nine pins on the board to your Arduino. The five pins you need are labeled **GND**, **3.3v**, **OUTPUT**, **LO-**, and **LO+**.

| Board Label	Pin Function	Arduino Connection
GND	Ground	**GND**
3.3v	3.3v Power Supply	**3.3v**
OUTPUT	Output Signal	**A0**
LO-	Leads-off Detect -	**11**
LO+	Leads-off Detect +	**10**
SDN	Shutdown	**Not used**

Connecting Headers to the Board

You can use any method you'd like to make your connections to the board. For this example, we'll solder on a five-pin length of male-male header strip and use a breadboard and jumpers to make our connections.

Headers installed

Follow the diagram below, to make necessary connections. The SDN pin is not used in this demo. Connecting this pin to ground or "LOW" on a digital pin will power down the chip. This is useful for low power applications.
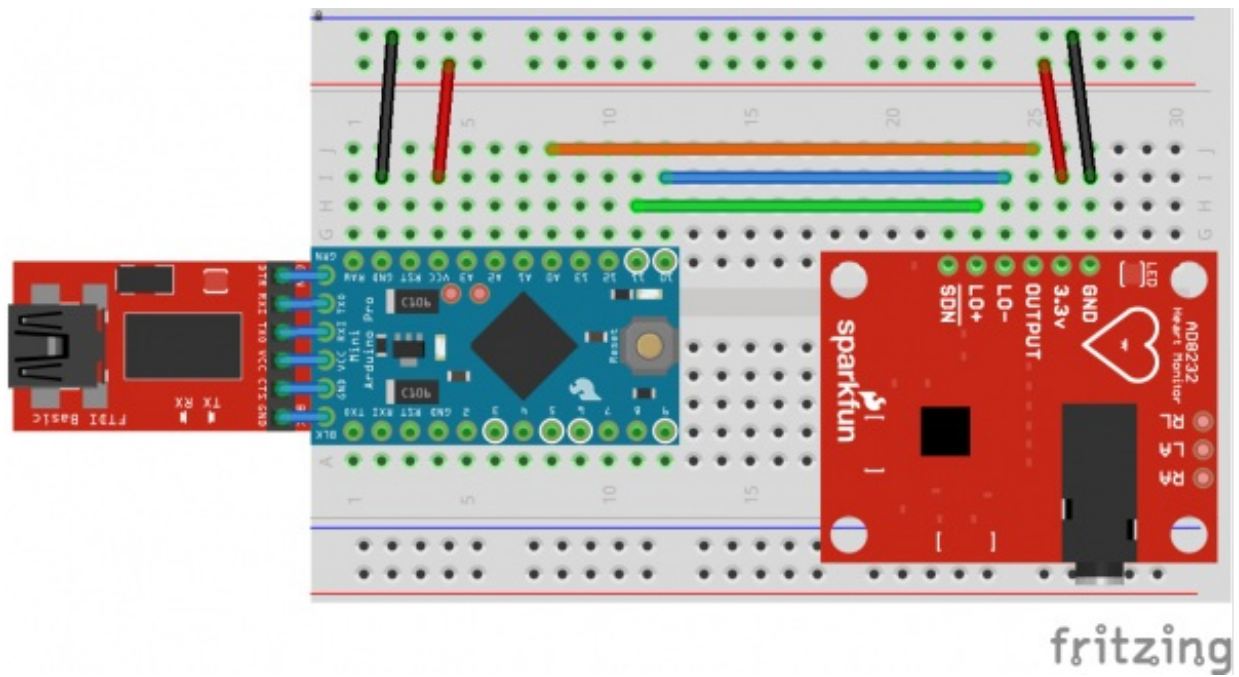
Connection Diagram

Now that the electronics are complete, let's look at sensor pad placement. It is recommended to snap the sensor pads on the leads before application to the body. The closer to the heart the pads are, the better the measurement. The cables are color coded to help identify proper placement.

Cable Color	Signal
Black	RA (Right Arm)
Blue	LA (Left Arm)
Red	RL (Right Leg)
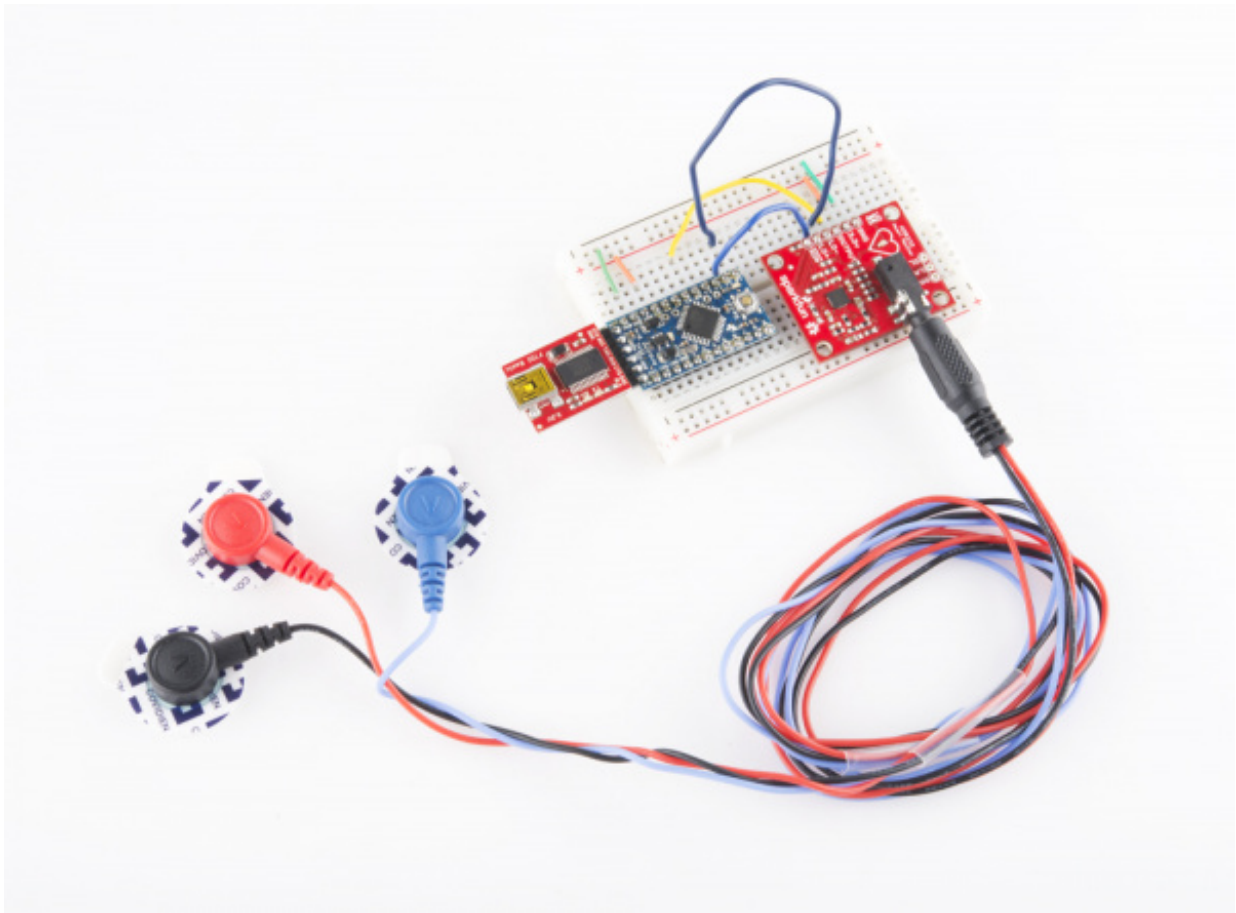
Sensors connected to Heart Monitor
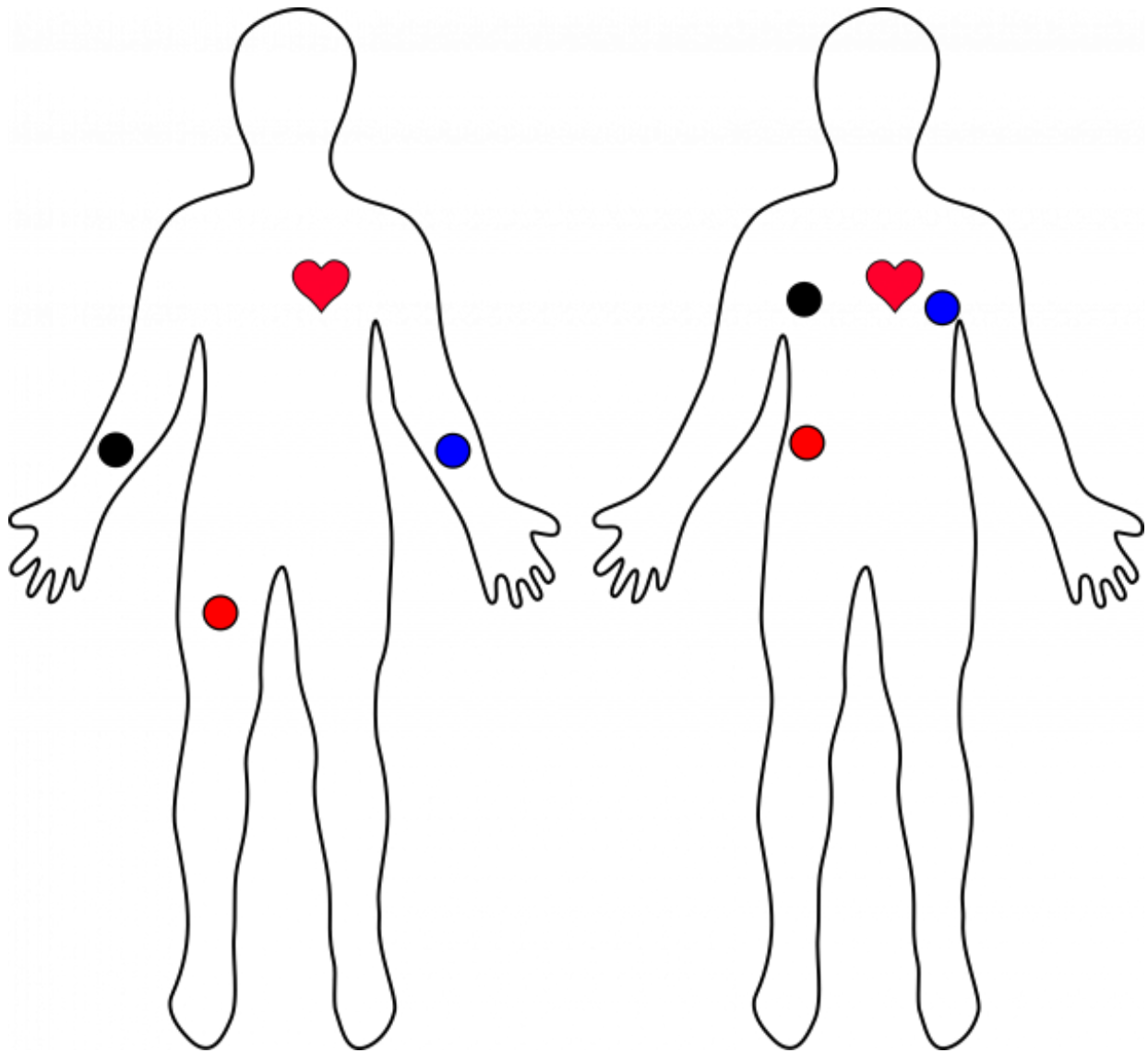
Typical Sensor Placements

Uploading the Sketch and Connecting with Processing

By this point, you should have the hardware connected and ready.

The example sketch can be found on github here. You can cut and paste the code straight from github or clone the repository and open the file. For more on using the SparkFun Github repositories, read this handy tutorial. If you've never worked with the Arduino IDE before, please see this tutorial.

Upload sketch to Arduino Pro-Mini

Now that you have a sketch running, let's get the processing sketch ready. The processing sketch will give you a visual output of what's going on. The example processing sketch can be found here. The example Processing sketch works with the **Processing IDE v2.2.1**. Later versions may not work with the example code so makes sure to download the stable release for v2.2.1 from the Processing IDE Download page.

Run the Processing sketch

If the processing sketch does not work, you may need to modify the following line:

myPort = new Serial(this, Serial.list()[2], 9600);

You may need to change the parameter inside Serial.list()[N]. A List of available COM ports will appear in the lower portion of the sketch window. Remember that COM port selection begins at 0. Typically your Arduino will appear as the highest COM number if it is the only device connected to your computer.
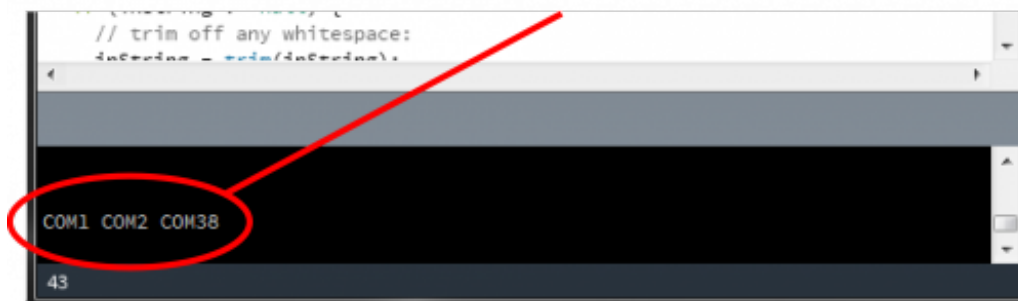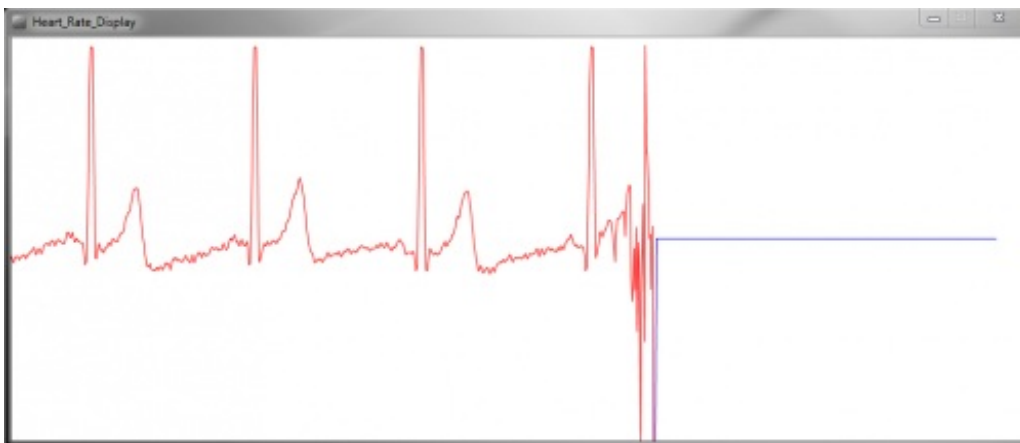

The Arduino shows up as COM38. This is found in the Serial List at array element "2".

If everything is working correctly, you should see a nice box pop up and start displaying the output signal.
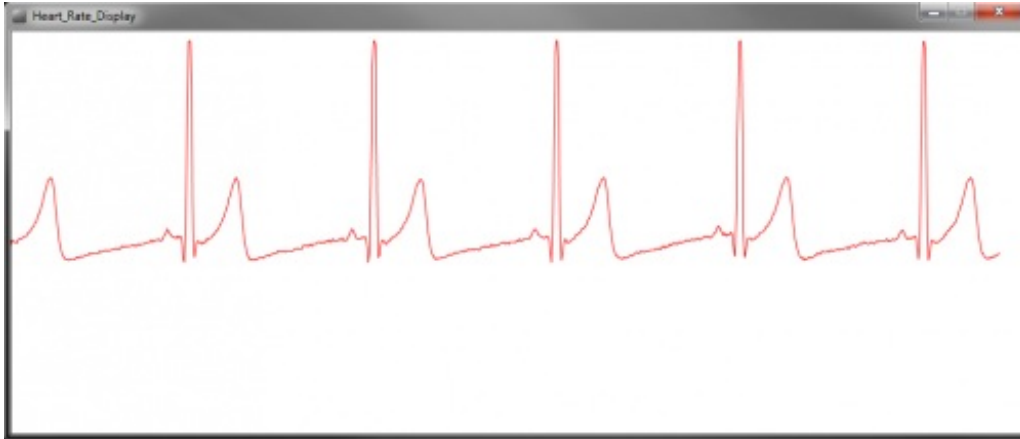
"Normal" heart rate of an SFE engineer

If your subject decides to remove the sensors, the leads off detection will kick in and display a flat blue line.

"Flight, we just lost Lovell!"

Tips and Tricks

ECG's are notoriously noisy. This is because you are measuring muscle activation. The further sensor pads are from the heart, the more muscle noise you will see. These are commonly referred to as "Motion Artifacts". So here are some simple tips to improve the signal quality.

- Keep sensor pads as close to the heart as you can.
- Make sure the RA and LA sensor pads are on correct sides of the heart.
- Try not to move too much while taking a measurement.
- Try to use fresh pads for each measurement. The pads loose the ability to pass signals with multiple applications.
- Prep and clean the area you plan to stick pads. This will help make a good connection (hair is not a good conductor).

- You may have to adjust sensor placement for different individuals.

Resources and Going Further

Thanks for reading. For more resources on the AD8232 Heart Rate Monitor, check out the following links:

- [AD8232 Datasheet](#)
- [AD8232 GitHub Repository](#)

For more Electronics Fun, check out these other SparkFun tutorials:

- [Hacking the MindWave Mobile](#)
- [Misfit Shine Teardown](#)

More Biometric products offered by SparkFun:

- [BITalino](#)
- [Pulse Sensor](#)
- [Muscle Sensor v3 Kit](#)

[learn.sparkfun.com](#) | [CC BY-SA 3.0](#) | SparkFun Electronics | Niwot, Colorado