

Assembling

the

body-sonic

Edwin Lohmeyer
Jay Kirby

0: Materials and overview

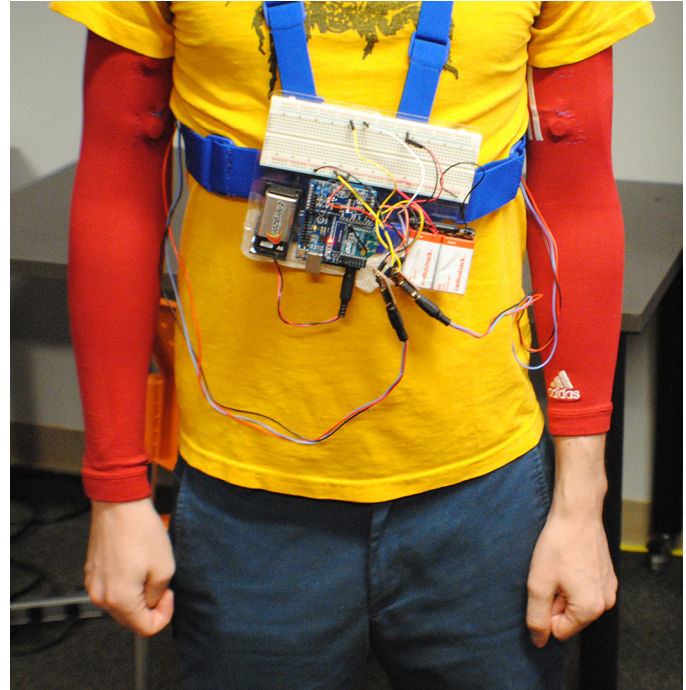
The body-sonic is a project that allows you to create synthesized sound based on the electrical output of your muscles. The project relies on the synthesized sound in conjunction with beats to create a system of sound between a person wearing the sensors and a person performing beats. These instructions will focus on assembling the sensors. You will need to look elsewhere for creating the beats. To assemble this portion of the project, you will need to connect sensors to a microcontroller, connect sensors to your body, connect the microcontroller to a computer, and run a Processing script on the computer.

OVERVIEW:

1. Load sensor-reading program onto your Arduino
2. Connect electromyography sensors to your Arduino
3. Connect electromyography sensors to your muscles
 - 3a. Use compression sleeves to connect to muscles
4. Connect Arduino to your computer
 - 4a. Use wireless to connect Arduino to computer
5. Run processing script on your computer

MATERIALS (SOFTWARE):

Arduino (<https://www.arduino.cc/en/Main/Software>)
Processing (<https://processing.org/download/>)
Beads library for Processing (<http://www.beadsproject.net/>)
XCTU (<http://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu>)



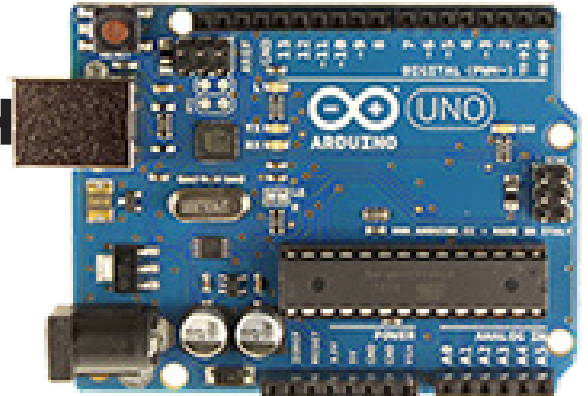
MATERIALS (HARDWARE):

- 1 Arduino Uno
- 1 SainSmart wireless shield for Arduino (optional)
- 2 XBee 1mW Wire antenna, series 1 (optional)
- 1 XBee Exploere USB Dongle (optional)
- 2 Advancer muscle sensor v3 (optional)
- 1 Breadboard
- 9v batteries
- Wires
- Soldering iron
- Compression sleeves (optional)
- Conductive fabric (optional)
- 9v battery holders (optional)
- Sewing equipment (optional)
- Materials for harness (optional, not covered in instructions)

1 Load the program

sensor_readings_for_Body_Sonic into your Arduino

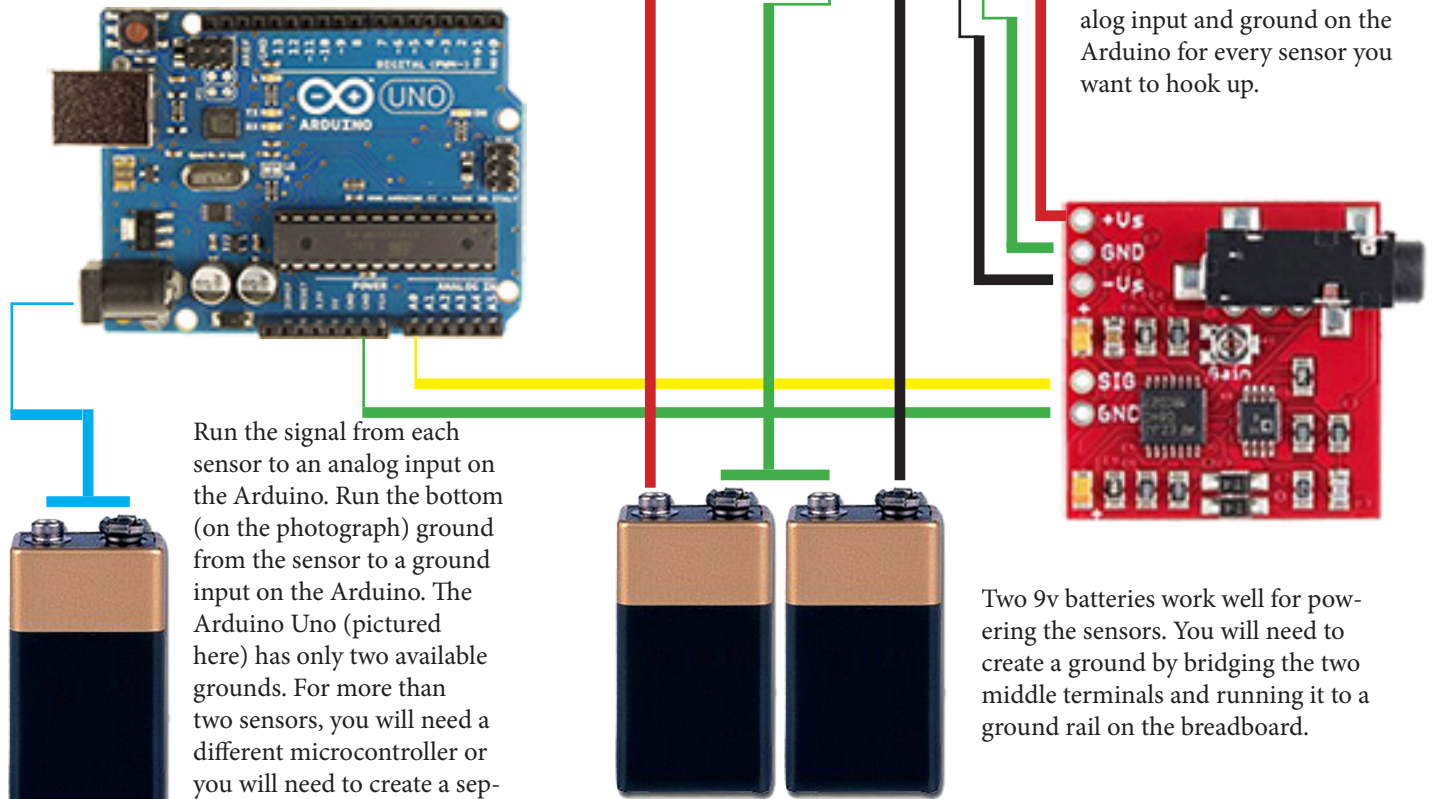
Connect your Arduino to your computer via the USB cable.



Click the upload button in the upper left of the screen to upload the program. Ensure you are connected to the correct port by verifying in the bottom right.

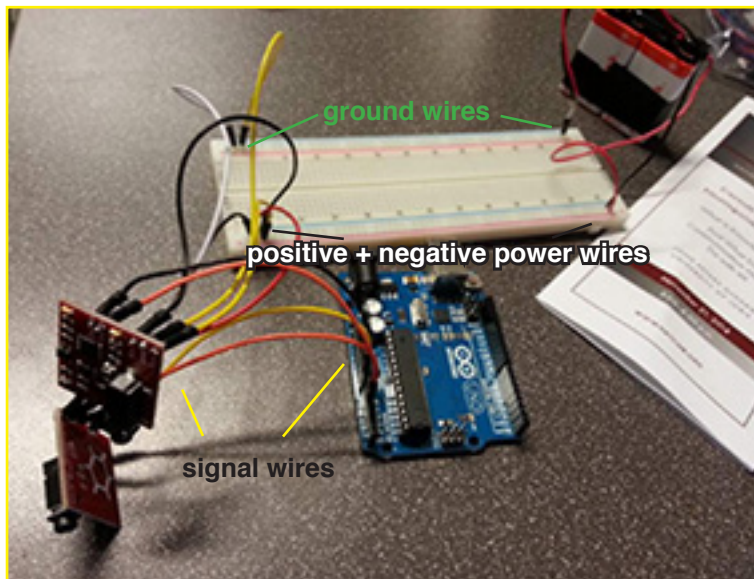
2. Connect electromyography sensors to the Arduino

Hooking up the electromyography sensors requires at least one sensor, a power source of two 9v batteries, an Arduino, and a separate power source for the Arduino if you are using a wireless connection instead of a USB connection. (See step 4 for hooking up the Arduino wirelessly.)



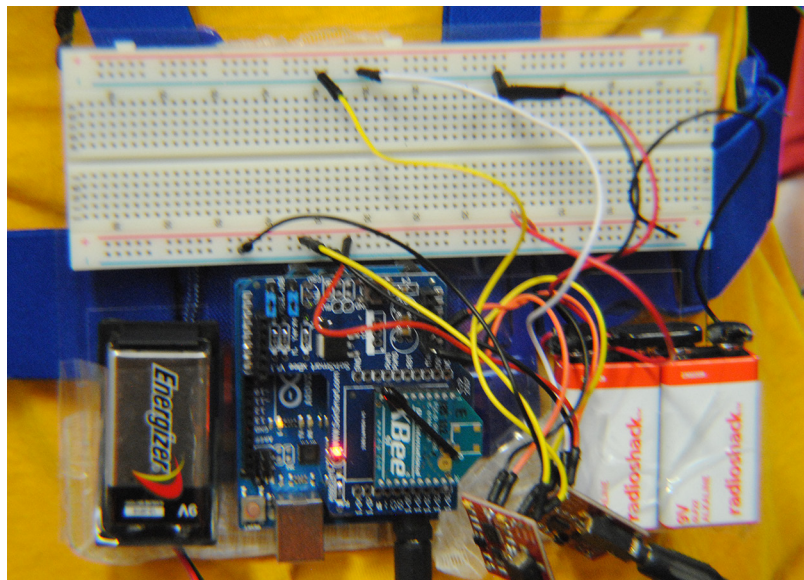
WARNING: When connecting the batteries, connect the positive wire, then the ground, then the negative wire. Connecting otherwise might fry the sensor or Arduino.

2. Connect electromyography sensors to the Arduino

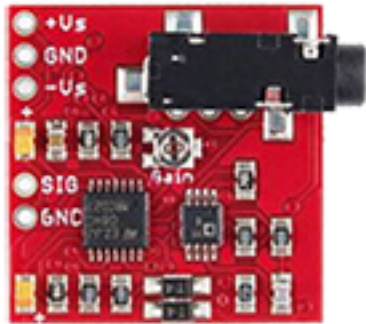


Two sensors attached to an Arduino and a breadboard. The sensors will get power from the two 9v batteries via the rail on the breadboard. The Arduino can be powered via a USB cable (not pictured).

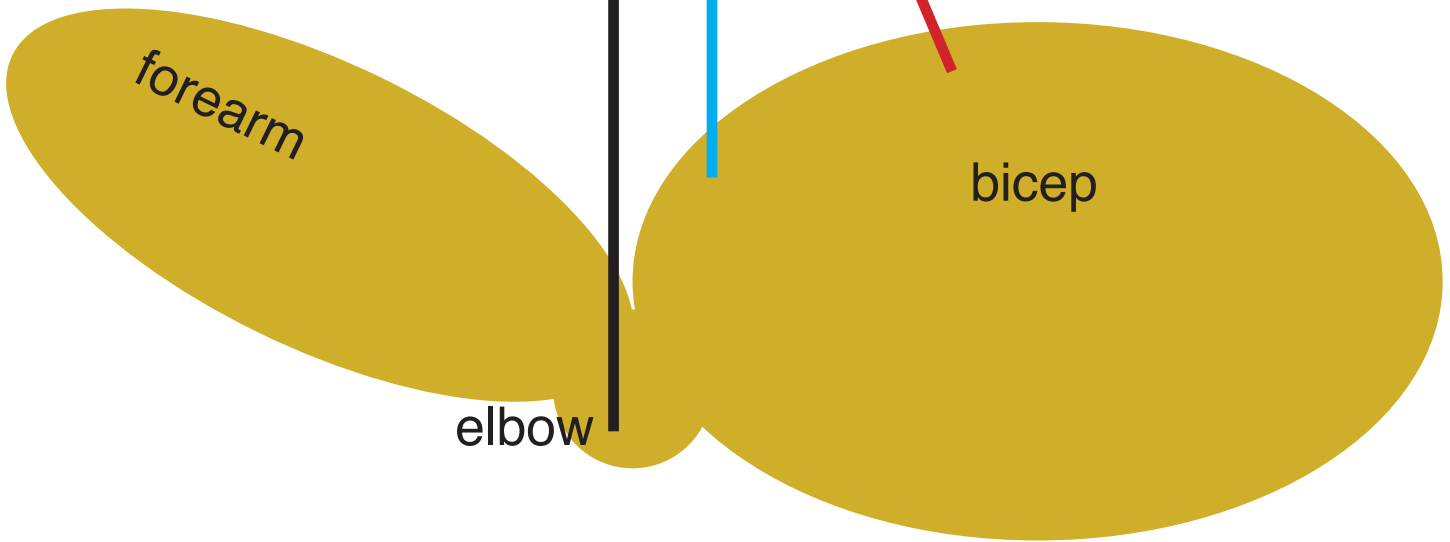
Two sensors attached to a harness. Here, a wireless shield covers some of the Arduino. Also, so the Arduino does not need to be connected via USB, it needs power from a separate source. The single 9v battery on the left powers the Arduino.



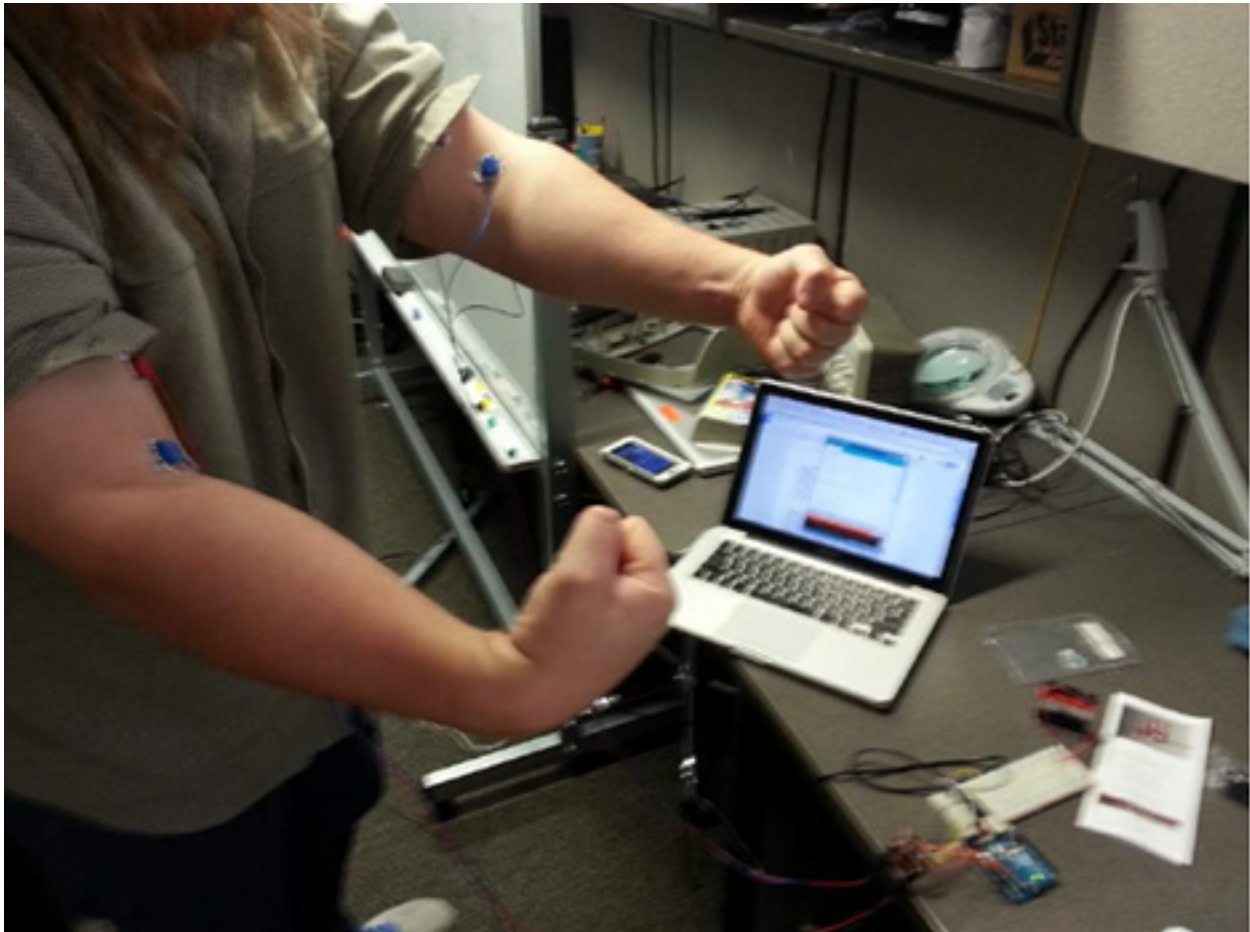
3. Connect electromyography sensors to muscles



The electromyography sensor will have three leads: one red, one blue, one black. Attach the red lead to the main body of the muscle. Attach the blue lead to the “front” of the muscle. Attach the black lead to a bony area without muscle. Below is a rough diagram of how to attach the sensors to measure the output of a bicep muscle. The sensors can be finicky, so it is often easier to use larger muscles.



3. Connect electromyography sensors to muscles

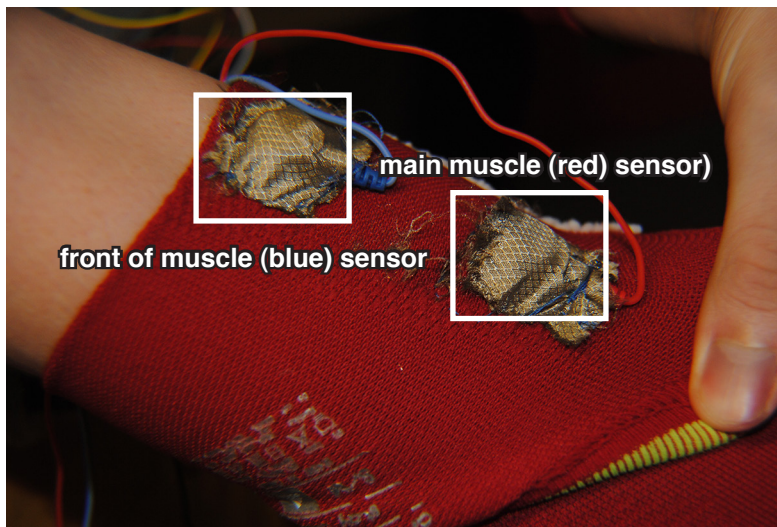


Sensors attached to biceps. The red (main muscle) and blue (front of muscle) sensor pads are visible. The black (bone/non-muscle) pads are not visible in this photograph. However, you can see the wire for the black, non-muscle lead on the left arm.

3a:

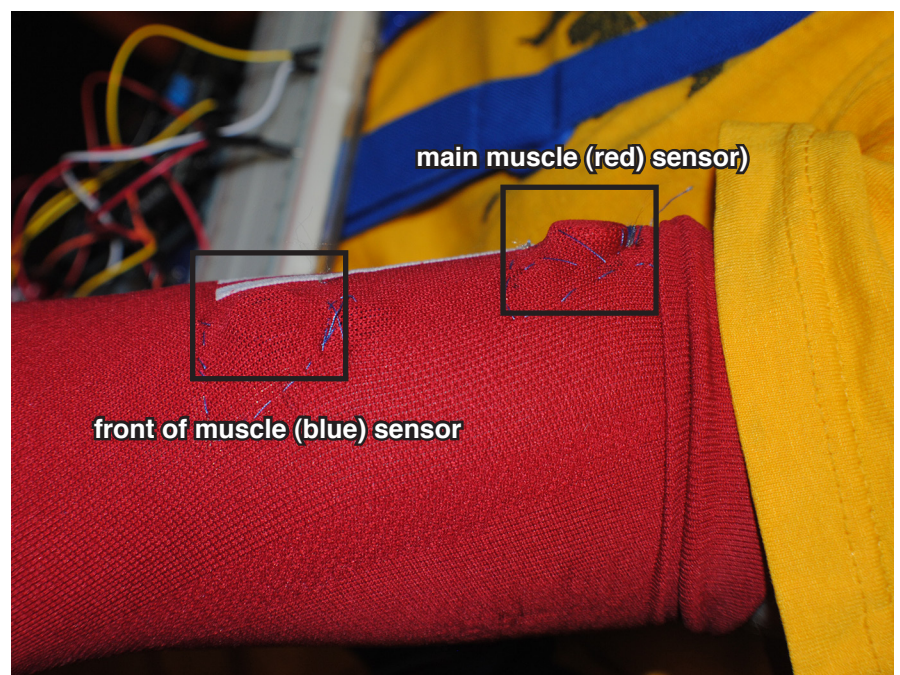
(optional)

Use compression sleeves to connect to muscles



To help hold the sensors in place, you might use a compression sleeve with conductive fabric to hold the sensors. Here, we used silver conductive fabric. Ensure the metal portion of the sensor lead is against the conductive fabric. Also, make sure to put on the compression sleeve and mark the main body, front, and non-muscle areas so that the pockets for the sensors line up with the appropriate areas of the muscles you will connect to.

This is the sleeve above, rolled up to cover the muscles. The red sensor moved up to the main body of the muscle. The blue sensor is on the front of the muscle. The black, non-muscle sensor is on the other side, against the elbow. The sensor pads included with the Advancer sensor produced good signals for one use. However, if you remove them and put them back on, they are much less effective. Further, the pads included tended to fall off or lose signal with vigorous movement. The compression sleeves with the sensors sewn in produced a strong signal every time. Additionally, the compression sleeve was not as affected by movement.



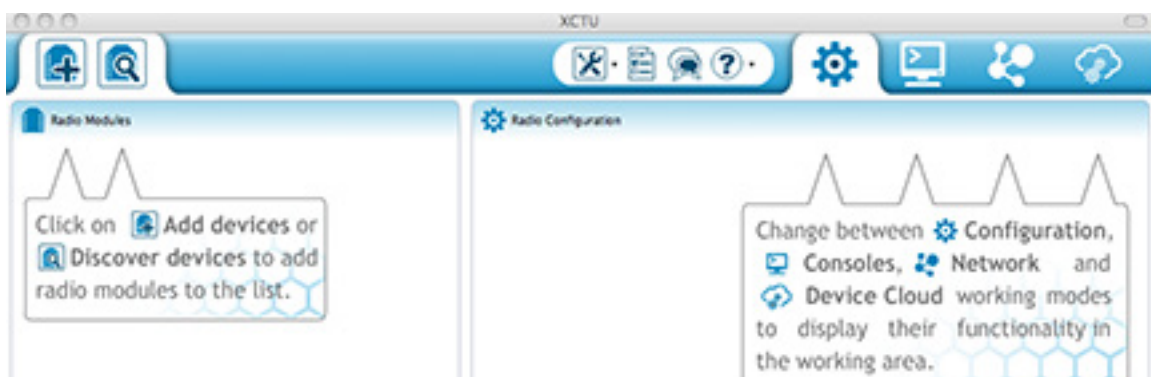
4(a): Connect Arduino to your computer (wirelessly)

To create a wired connection, simply use the Arduino's USB port. Connect this to your computer and proceed to the final step

To create a wireless connection, you will need to set up wireless receivers on both the Arduino and the computer that will run the processing script. Then, you will need to configure your wireless network.

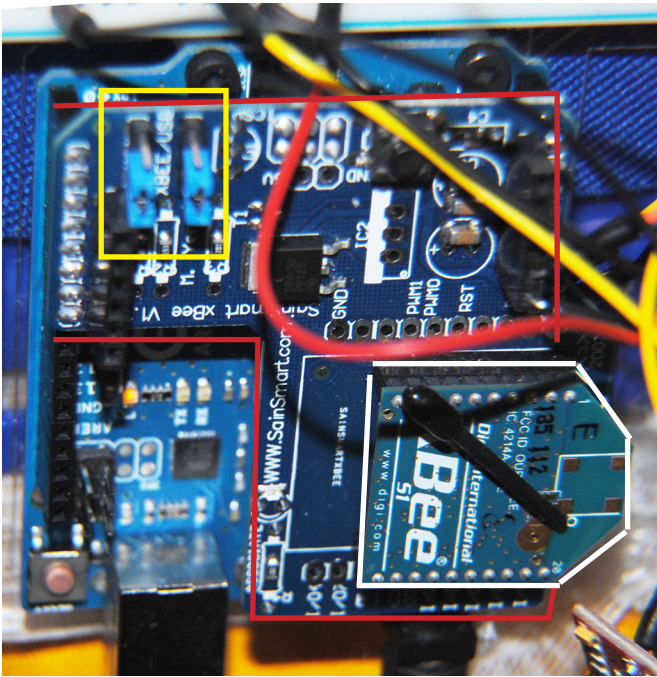


Attach one of the wireless antennas to the USB dongle as shown in the picture at left. Plug the dongle into the computer that will run the processing script and use XCTU to ensure the antenna is properly set up.



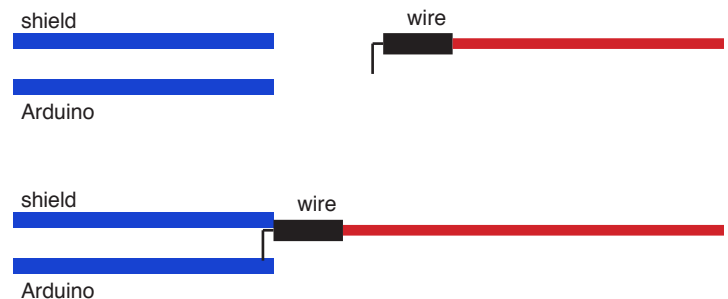
When you plug in your USB dongle, use the left side of the XCTU menu to detect the antenna. Then, use the right side of the XCTU panel to check the firmware by selecting “read device.” In addition to the firmware, this will give you numerous options for configuring the devices. Leave these alone except for the “My Address” and “Destination Address.” Set those to the same for each. After looking at each XBee in turn, be sure to “Write” your new settings to the XBee. Sometimes, the networks are plug and play. However, numerous problems might arise. Check forums on XBee, XCTU, or others for specific troubleshooting.

4a. Connect Arduino to your computer wirelessly



Note: In order for the transmitter to transmit data wirelessly to the other antenna, both jumpers in the yellow box must be set to “XBee.” To use the USB port to transfer data to the Arduino, both jumpers in the yellow box must be set to “USB.” You cannot transfer programs to the Arduino (like the `sensor_reading_for_body_sonic` program) over wireless.

Place the SainSmart XBee Shield on top of the Arduino Uno. The pins will line up. In the photograph at left, the shield is outlined in red. On an Arduino Uno, the shield will cover the analog pins and grounds that you will use to connect the sensors to the Arduino. Either purchase wires with 90 degree attachments or bend the tips of the wires in the sensors at a 90 degree angle in order to get them to fit.



Place one of the XBee series 1 antennas on the shield. In the photograph at left, the antenna is outlined in white. The shield will have the outline of the antenna on it so that you will position it correctly.

Once everything is set up, you should be able to read information from your arduino on your computer. Open the serial monitor in Arduino to check. If working properly, you should see numbers changing as you flex your muscles. We will now turn to the processing script that will turn these numbers into sound.

5. Run the processing script on your computer

Once you have your wireless (or wired) connection between your Arduino and your computer, you can run the processing script and play with the sounds on the body-sonic. Make sure to close out Arduino and XCTU if you were using either to test your connection. Failure to do so may cause interference that prevents your code from running. Then, open Processing and open either `Modulated_synth_Body_Sonic` or `Tuned_synth_Body_Sonic`. Either of these will produce synthesized sound based on the readings from the muscle sensors. The “Modulated Synth” file will produce abrasive, difficult-to-predict sounds. The “Tuned Synth” will produce sounds in a specific key. The documentation on both should help you modify those scripts to run with more sensors, in different keys, or with different modulation.

Once you have played with the body-sonic for a while, you may want to edit the code. Beads has excellent documentation for beginners in *Sonifying Processing: The Beads Tutorial* (<http://computermusicblog.com/blog/sonifyingprocessing/>). At this point, you may use the body-sonic however you desire to reveal the connections between code, people, and machines. Or you can use it to make crazy sounds!

