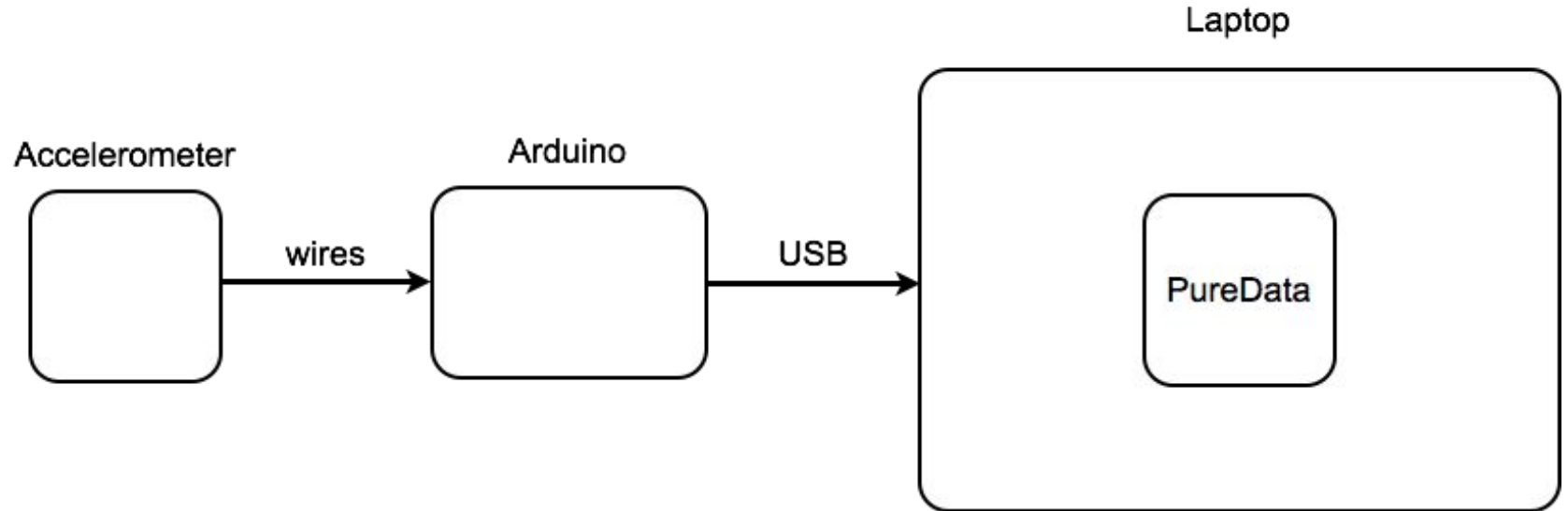


Get the Files (thumbdrive or download)

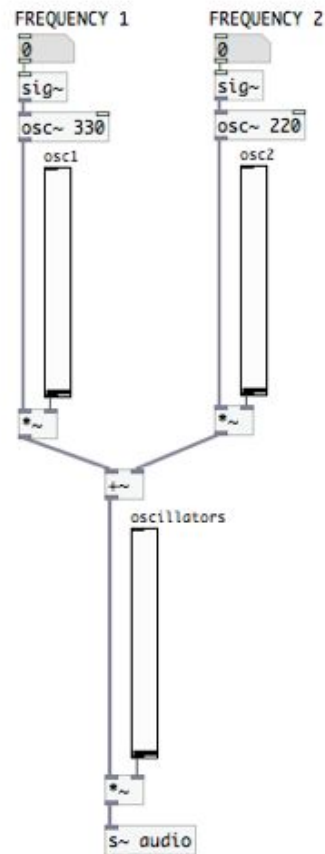
- Workshop2015.zip
 - Get from thumbdrive
 - Or download (277mb): <http://bit.ly/1PATECD>
- Unzip the file

Basic Setup



Introduction to PureData (Pd)

- <http://puredata.info/>
- real-time visual programming environment
- can be used for audio and video processing
- dataflow: functions or “objects” are linked or “patched” together



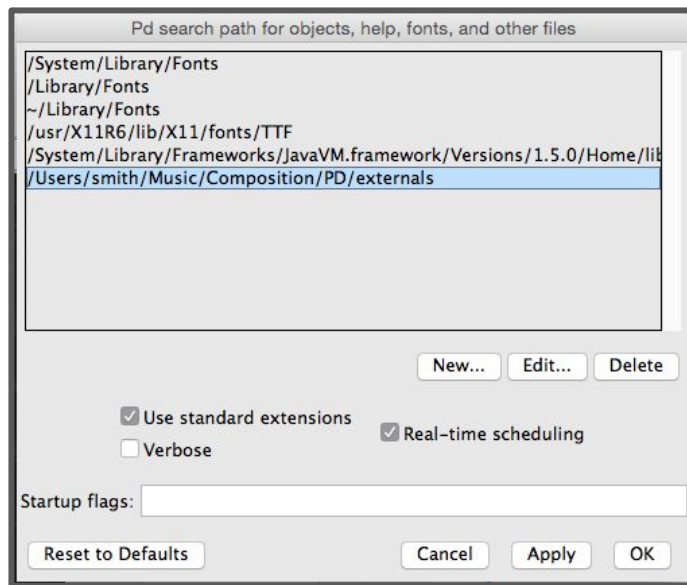
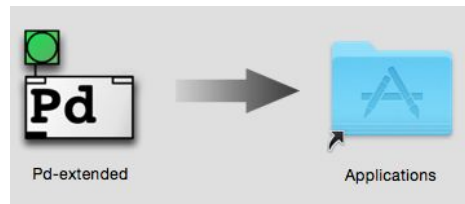
Setting up PureData

1. Install PureData (dmg file is in the “Software” folder I gave you)

Pd-0.43.4-extended-macosx105-i386.dmg

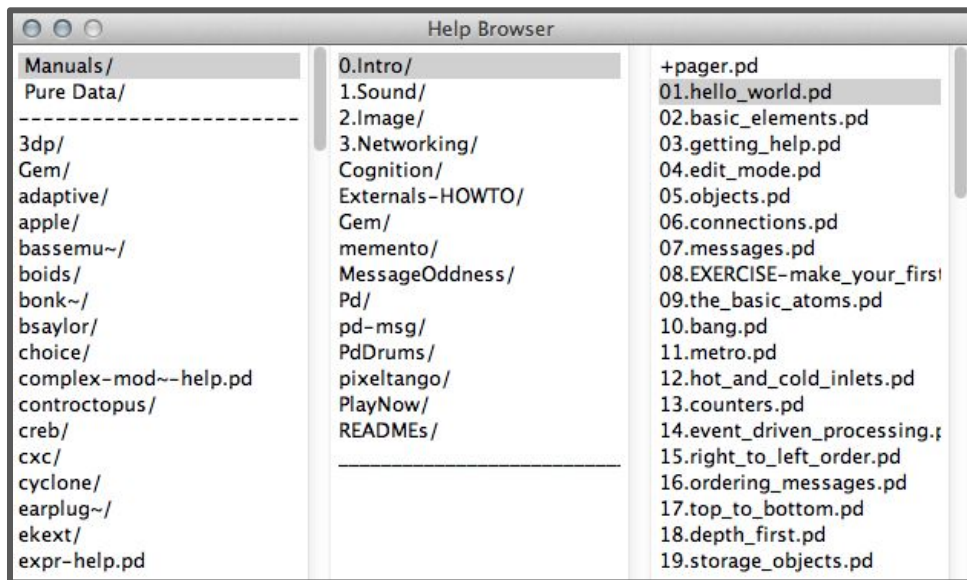
On Mountain Lion or later? You also need to install X11: XQuartz-2.7.8.dmg

2. Move the “Patches” folder to your “Documents” folder.
3. Add the “externals” folder within Patches to the search paths in Pd:
Pd-extended > Preferences > New...



PureData Basics

- Open the ‘Help Browser’ (Help > Browser)
- Select Intro Lesson ‘01.hello_world.pd’
- We’ll walk-through lessons 1-7, 10-12, 15-16, 24



Keyboard Shortcuts
you can find these in the “put” menu

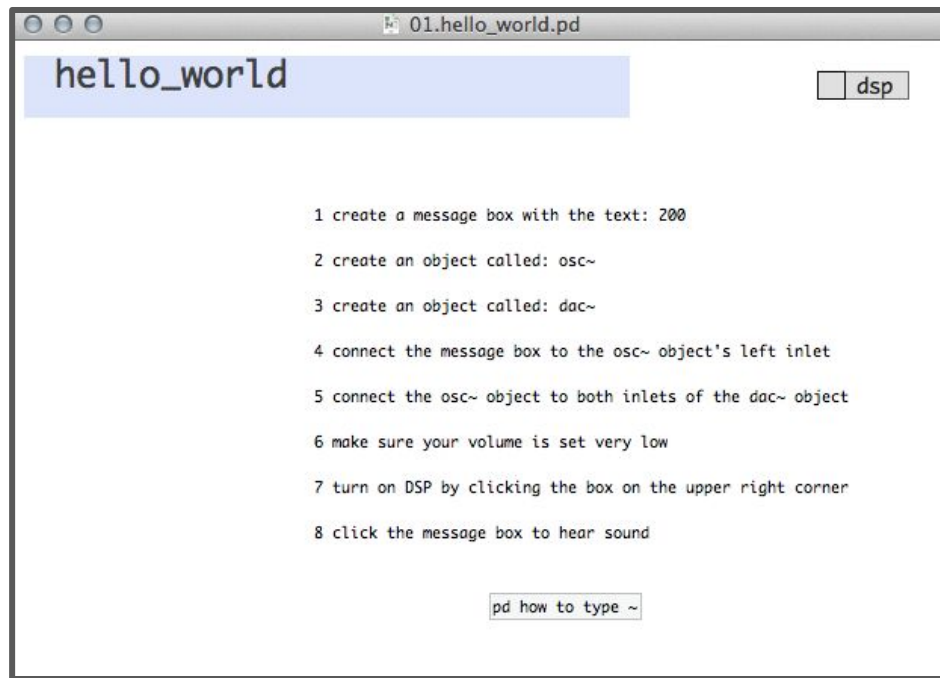
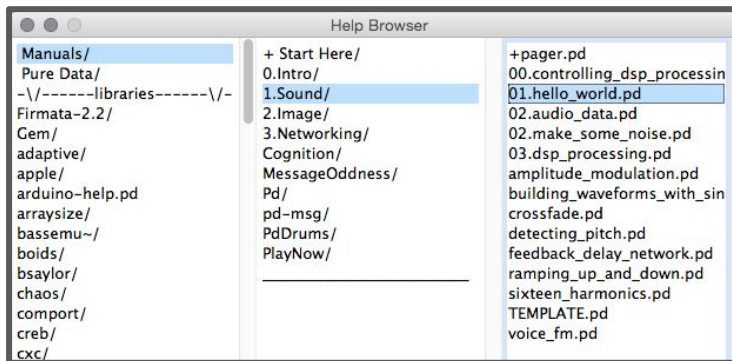
Object	⌘1
Message	⌘2
Number	⌘3
Symbol	⌘4
Comment	⌘5
Bang	⇧⌘B
Toggle	⇧⌘T
Number2	⇧⌘N
Vslider	⇧⌘V
Hslider	⇧⌘H
Vradio	⇧⌘D
Hradio	⇧⌘I
VU	⇧⌘U
Canvas	⇧⌘C

Intro to Sound in PureData

In Pd, message processing is always running, but audio processing (DSP, digital signal processing) needs to be turned on and off.

Let's start by opening Sound lesson

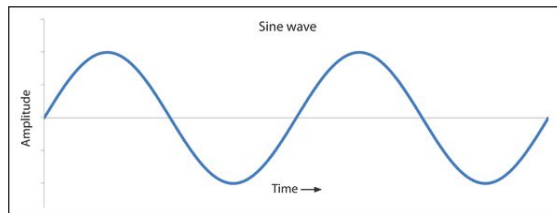
01.hello_world.pd



Some Audio Basics

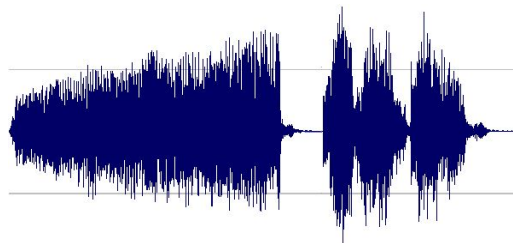
Oscillator: “Pure tone”

- Can change frequency / pitch
- Can change amplitude (volume)



Samples (audio files)

- Can play the file
- Can change amplitude (volume)



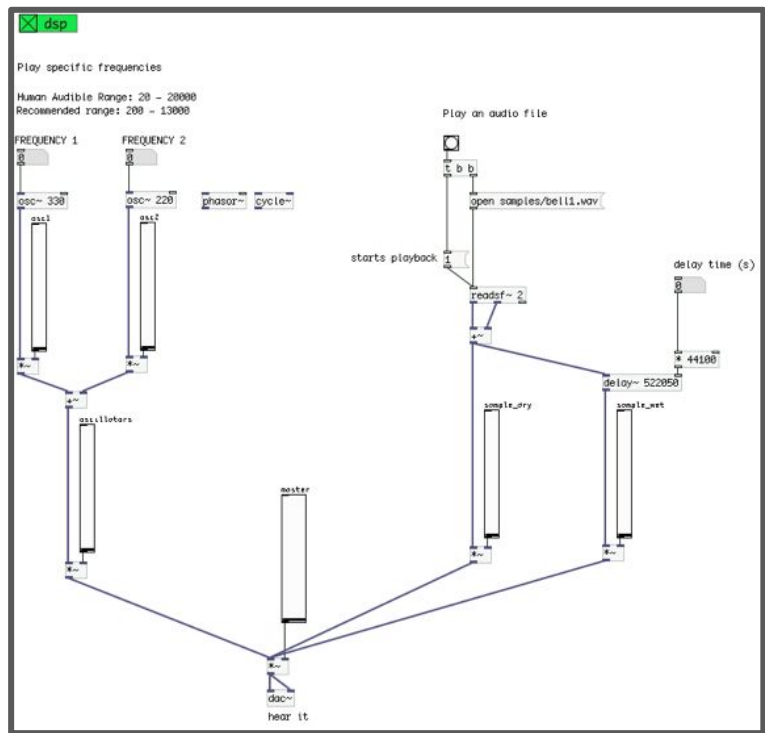
Amplitude = Level = Gain = Volume

Increase amplitude by multiplying the signal

Exploring Sound in PureData

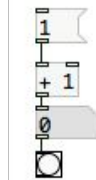
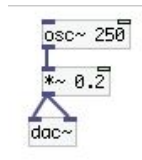
Open the first patch I gave you:

1.audio.pd



General Tips in Pd

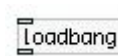
- Signal vs. data-flow



- trigger object helps to order actions from a single bang

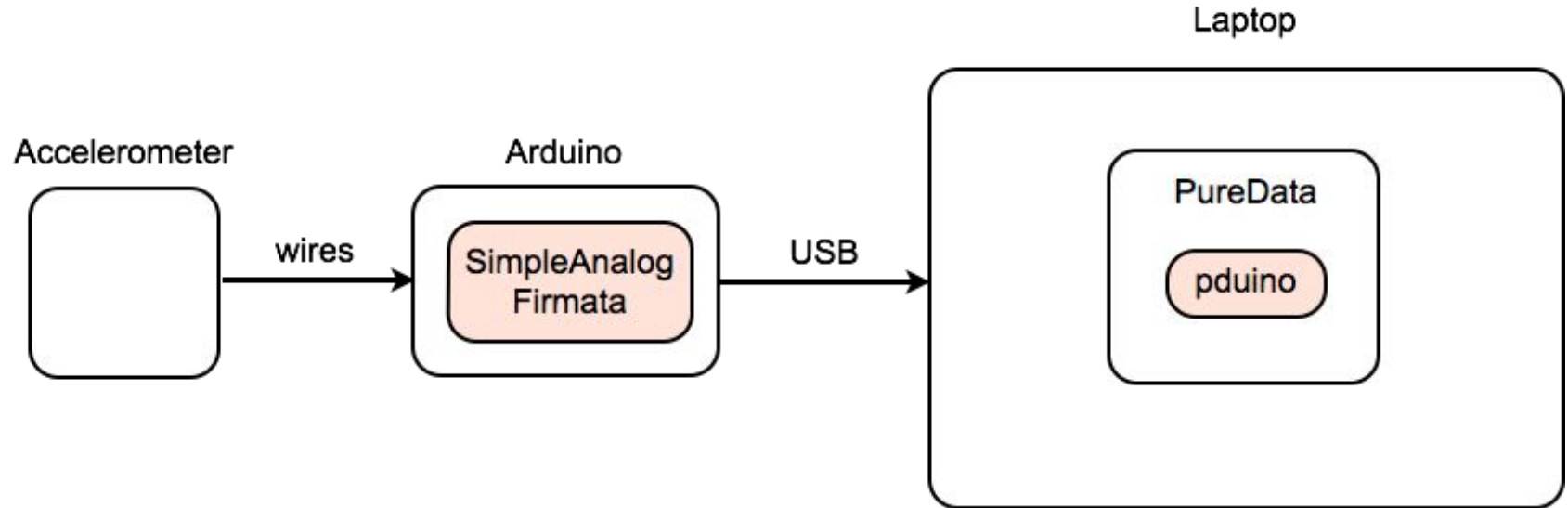


- loadbang sends a bang when the patch is first opened.



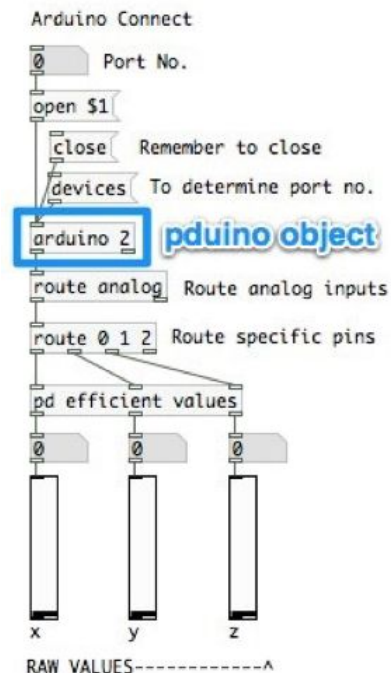
- Keyboard shortcuts are your friend

A Closer Look



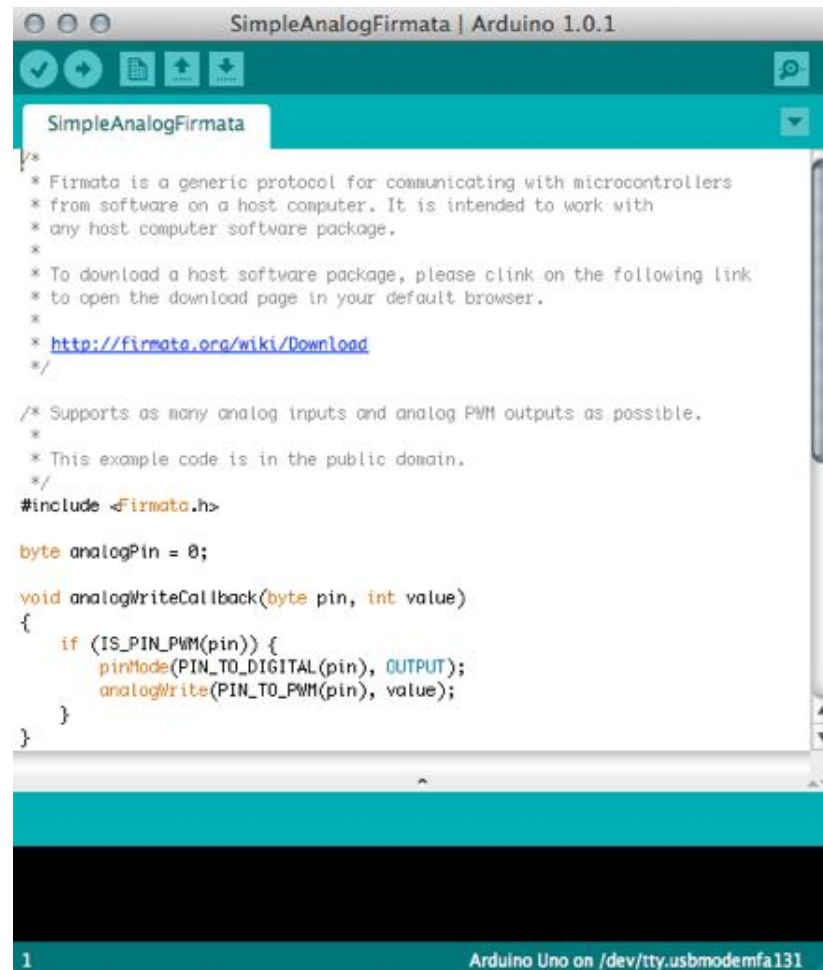
Installing pduino

- External object developed by a PureData enthusiast.
- Compatible with the StandardFirmata library on the Arduino
- Unzip the **Pduino-0.5.zip** file (in “Software”) and move its contents to the “externals” folder in your Patches folder.
- Try making an [arduino] object to test if pduino was installed correctly.



Intro to Arduino

- The Arduino board is a microcontroller that can receive data input from sensors and relay it to software on our computers
- The corresponding Arduino software is a C++ programming environment that can upload directly to the board.
- Using the Arduino software, we'll upload firmware onto the board that allows other software to talk to the board (i.e. the pduino object in PureData)
- Unzip and Install Arduino:
`arduino-1.6.5-r5-macosx.zip`



The screenshot shows the Arduino IDE interface with the title bar 'SimpleAnalogFirmata | Arduino 1.0.1'. The menu bar includes 'File', 'Edit', 'Tools', and 'Help'. The toolbar contains icons for opening files, saving, uploading, and downloading. The main text area displays the code for the 'SimpleAnalogFirmata' sketch. The code includes comments about the Firmata protocol and a link to the Firmata website. It also includes the `<Firmata.h>` header and defines an `analogPin` variable. The `analogWriteCallback` function is defined to handle digital pin writes by converting them to analog writes. The status bar at the bottom indicates '1' and 'Arduino Uno on /dev/tty.usbmodemfa131'.

```
SimpleAnalogFirmata

/*
 * Firmata is a generic protocol for communicating with microcontrollers
 * from software on a host computer. It is intended to work with
 * any host computer software package.
 *
 * To download a host software package, please click on the following link
 * to open the download page in your default browser.
 *
 * http://firmata.org/wiki/Download
 */

/* Supports as many analog inputs and analog PWM outputs as possible.
 *
 * This example code is in the public domain.
 */
#include <Firmata.h>

byte analogPin = 0;

void analogWriteCallback(byte pin, int value)
{
  if (IS_PIN_PWM(pin)) {
    pinMode(PIN_TO_DIGITAL(pin), OUTPUT);
    analogWrite(PIN_TO_PWM(pin), value);
  }
}
```

1 Arduino Uno on /dev/tty.usbmodemfa131

Configure the Arduino Board

1. Connect board to laptop with USB cable and launch the Arduino application
2. Select your board:
Tools > Board > Arduino Uno
3. Select your serial port:
Tools > Serial Port > dev/tty.usbmodem (or similar)
4. Open “SimpleAnalogFirmata”
File > Examples > Firmata > SimpleAnalogFirmata
5. Press the upload button
6. Wait for ‘Done uploading’ message, then quit the application and unplug the board



Connect Sensor to Board

Caution: make sure your board is unplugged when connecting the sensor. This is to prevent damage to the sensor if the wrong pins are connected

- GND → GND (ground)
- 3.3 V → 3.3 V (power)
- X → A0 (analog pin 0)
- Y → A1 (analog pin 1)
- Z → A2 (analog pin 2)

Connect Board to PureData

- Plug in your board once the sensor is connected
- Open PureData
- open the second patch I gave you: `2.audio_arduino.pd`

Turning Data into Sound

- Scaling / Mapping a range of values
 - basic calibration
 - volume
 - pitch
 - any slider or object that accepts numerical inputs
- Momentary Events / Triggers
 - basic calibration
- Averaging
 - advantage: smoothing
 - disadvantage: lowers responsiveness

Calibration: Scaling / Mapping

Our primary calibration tool and most useful when fine-tuning “rotating,” continuous motion.

More Sensitive

- Mapping from a small range to a large range of numbers makes it more sensitive.
- Works best with detailed movement -- requires careful movement, easier to trigger things

Less Sensitive

- Mapping from a large range to a small range of numbers makes it less sensitive.
- Allows you to be less careful about movement, though it becomes harder to trigger things, requires more force -- fast movement, “hitting”

Triggers (“hitting,” momentary)

Allows for momentary control. Use with “hitting” motions.

Method:

- Set a threshold or limit.
- When the acceleration crosses that threshold, a “bang” is triggered.
- The higher the threshold, the harder it will be to trigger.

Calibration

When calibrating, remember:

- If calibrated to be sensitive, it will be easier to trigger things and works best with careful movements.
- If calibrated to be less sensitive, it will be harder to trigger things and will require more force, but you won't need to be as careful with your smaller movements.
- Good calibration is a tricky balance between sensitivity and physical movement and orientation. It will require a lot of trial and error, and it's helpful to know beforehand the kinds of movements you will be working with.

Smoothing the Data

- Smoothing: averaging a few numbers to decrease jumpiness
- Keep in mind: The more numbers you average, the smoother it will be, but it will also be less responsive.
- To start, I do not recommend using this unless the “jumpiness” of the sensor causes problems in your patch.

3.template.pd

- This is a PureData patch with a lot of audio processing components built in.
- Could be used either as a starter patch or to copy/paste components from.

Reference Materials

Accelerometer

- ADXL 337: <https://www.sparkfun.com/products/12786>

Arduino

- <http://arduino.cc/>

PureData

- We're using pd-extended: <http://puredata.info/docs/StartHere/>
- PureData on Yosemite requires XQuartz: <http://xquartz.macosforge.org/landing/>

pduino object

- <http://at.or.at/hans/pd/objects.html#pduino>

More Resources

Online:

- Arduino website
<http://arduino.cc>
- PureData website
<http://puredata.info>
- Pd objects
<http://flexatone.net/docs/pdg/>
- Floss Manuals
<http://en.flossmanuals.net/pure-data/>
- SparkFun (sensors, components)
<http://sparkfun.com>
- Jameco (cheaper sensors, components)
<http://jameco.com>
- Google!

In Los Angeles:

- CRASHspace Pd patching circle
<http://puredata.info/community/groups/lapc>
- Radioshack
- All Electronics (Van Nuys)
<http://www.allelectronics.com>
- Apex Electronics (Sun Valley)
<http://apexelectronic.com>