

Chapter 15. Meeting 15, Practices: Electronics and Sensors

15.1. Announcements

- Workshop on Monday: bring controller and amps
- Controller/Interface/Instrument Design 1: comments and grades this weekend
- Due Wednesday 6 April: Controller/Interface/Instrument Design 2 Proposal
- Due Wednesday 13 April: Performance Frameworks Draft

Must email me now with special requests for groups

15.2. Reading: Tanaka: Sensor-Based Musical Instruments and Interactive Music

- Tanaka, A. 2009. “Sensor-Based Musical Instruments and Interactive Music.” In R. T. Dean, ed. *The Oxford Handbook of Computer Music*. Oxford University Press, pp. 233-257. (243-257)
- What trends are can be seen in the availability and cost of sensors?
- What are examples of biosignal instruments?
- Tanaka writes that: “musicians have the uncanny ability to appreciate and repurpose machinery and technology for expressive musical ends”: what are some examples?

15.3. Listening and Viewing: Sensorband and Atau Tanaka

- Viewing: Sensorband Performance (DEAF96): YouTube (<http://www.youtube.com/watch?v=XLS0PmY6jGM>)
- Viewing: Atau Tanaka performs using bio-metric sensor: YouTube (http://www.youtube.com/watch?v=FB_yE_Y3_8k)
- Listening: Sensorband / Atau Tanaka, Sola Produxies

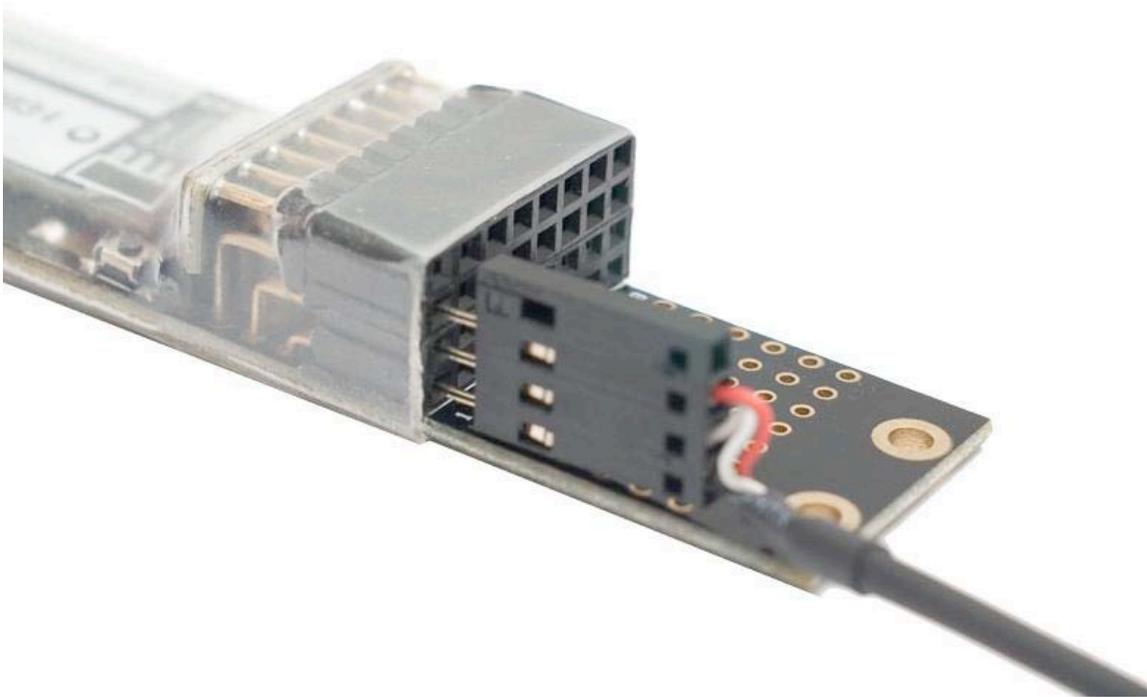
15.4. Sensor Interfaces: Tactile and Table

- James Patten's AudioPad: <http://www.jamespatten.com/audiopad>
- Sergi Jorda's ReacTable: <http://www.reactable.com/> (<http://www.reactable.com>)

15.5. Specialized Sensor Hubs

- iCube: <http://infusionsystems.com/>

Example setup: USB-microSystem with 8 inputs and 8 outputs with USB computer input (\$156)



Courtesy of Infusion Systems Ltd. Used with permission.

- Teabox <http://shop.electrotap.com/products/teabox>

Example setup: Teabox Sensor Interface with 8 inputs with neutrik combo jacks and SPDIF computer input (\$395)



© Electrotap, LLC. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <http://ocw.mit.edu/fairuse>.

- BioMuse by BioControl Systems: <http://www.biocontrol.com>

BioWave (EEG, EOG, and EMG \$505), BioFlex (EMG from arm and leg muscles \$403), BioBeat (ECG signals from chest and torso \$546)



Courtesy of BioControl Systems. Used with permission.

15.6. General Purpose Microcontrollers

- Arduino: <http://www.arduino.cc> (<http://www.arduino.cc/>)

Example setup: Arduino UNO with 6 analog inputs and 14 digital i/o (\$30)

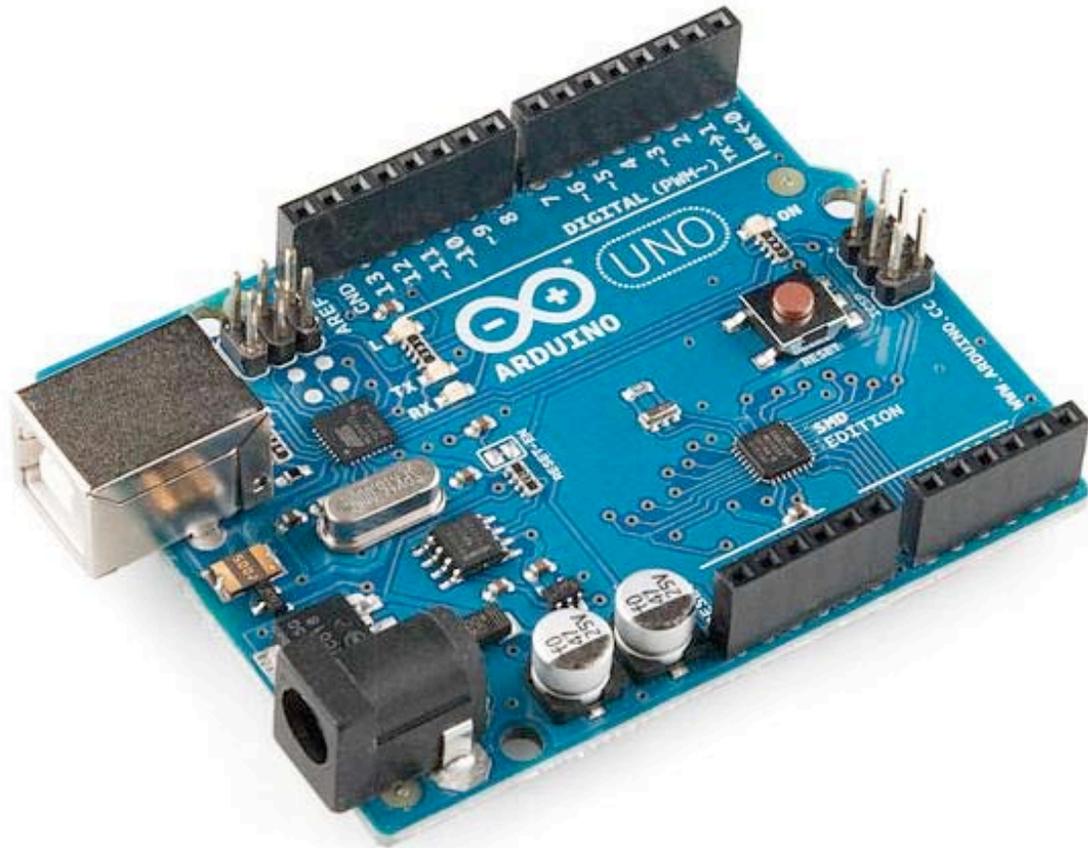
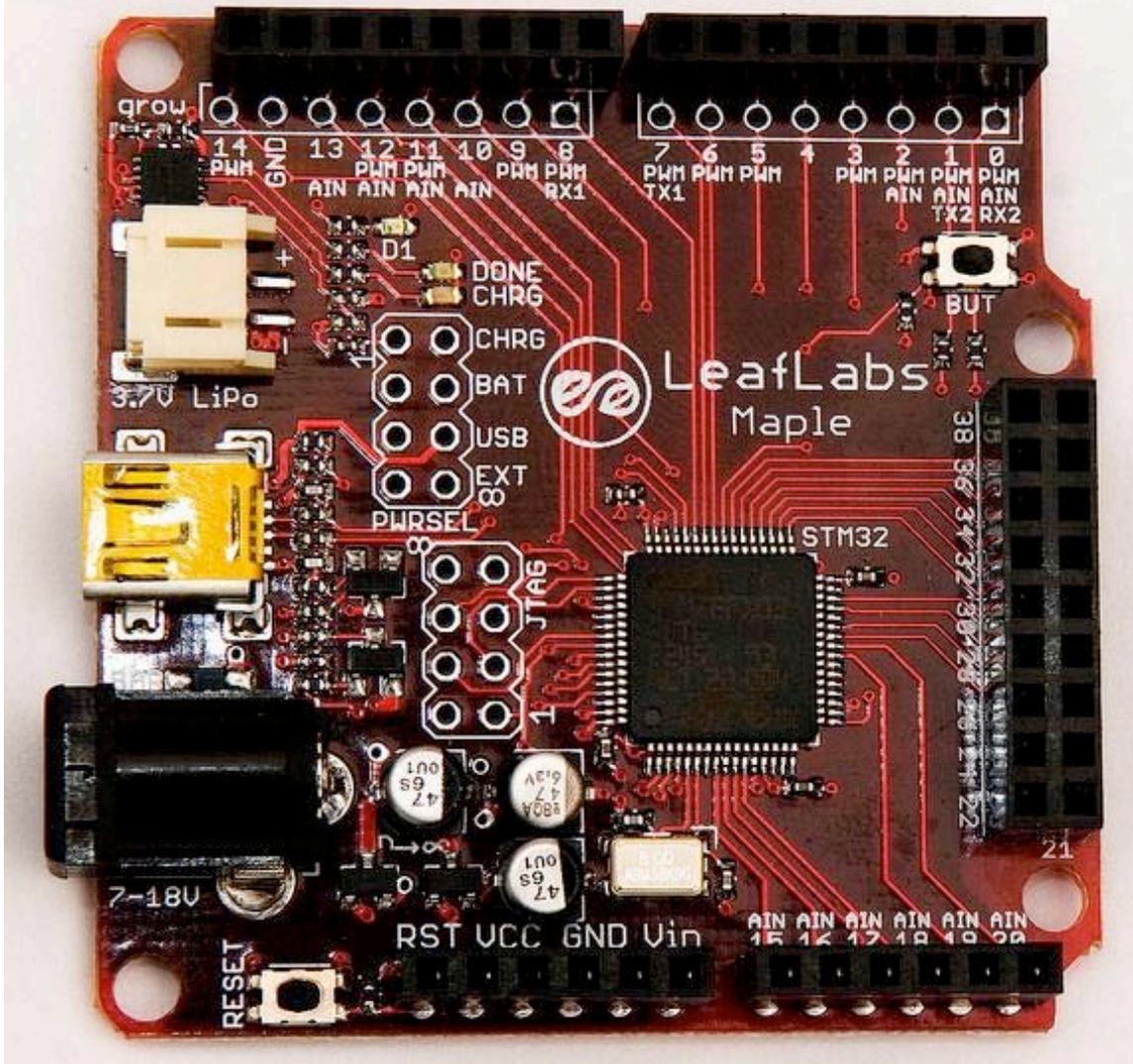


Photo courtesy of SparkFun Electronics.

- The Maple from LeafLabs <http://leaflabs.com/devices/maple/>

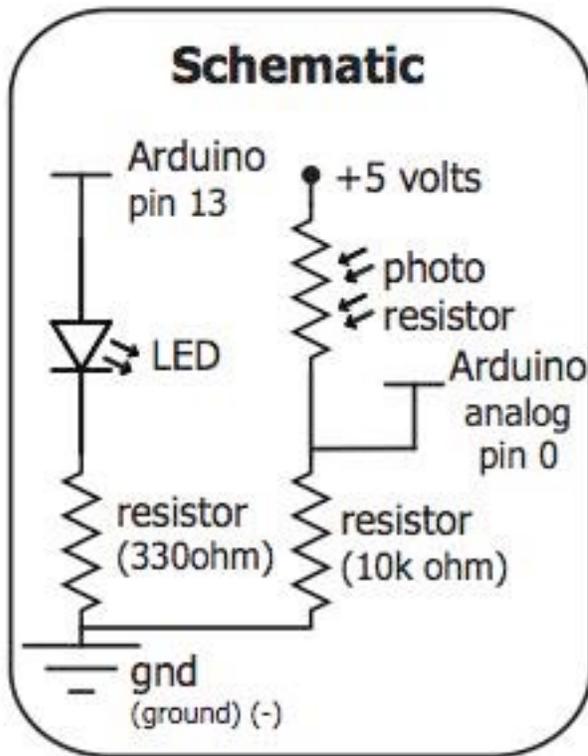
Example setup: Maple with 16 analog inputs and 39 digital i/o (\$50)



Courtesy of LeafLabs LLC. [License CC BY 3.0](https://creativecommons.org/licenses/by/3.0/)

15.7. Getting Analog Signals into the Arduino

- Can use a variety of sensors that produce variable voltage resistance
- Often need to only supply power and ground, as well as a resistor for voltage dividing
- Example of components for reading values from a photoresistor (ignore LED)



Source: Manual for SparkFun Inventor's Kit for Arduino. © :oomlout:. License CC BY-SA.

This content is excluded from our Creative Commons license. For more information, see <http://ocw.mit.edu/fairuse>.

15.8. Arduino Code for Processing and Transmitting Analog Input

- The Arduino IDE provides code-editing and uploading to Arduino over USB
- Code is a light-weight version of C++ that is compiled for the ATmega328
- Code divided into two functions: `setup()` and `loop()`
- Can read from an analog pin with `analogRead()`
- Can print to the serial interface (transmit bytes) via `Serial.print()`
- Delay time in ms with `delay()`
- Complete code for reading, mapping, and printing values from two analog pins

```
int photoPin = 0;
int flexPin = 1;
int x;
int y;

void setup() {
  Serial.begin(9600);
}

void loop () {
  Serial.print('A', BYTE); // char 65
```

```

// practical range from 200 to 900
x = analogRead(photoPin);
Serial.print(map(x, 200, 900, 1000, 2000)); // sends 3 bytes
Serial.print('\n', BYTE); // char 10

Serial.print('B', BYTE); // char 66
// raw values are between 47 and 282
y = analogRead(flexPin);
Serial.print(map(y, 47, 282, 1000, 2000));
Serial.print('\n', BYTE); // char 10

delay(30);
}

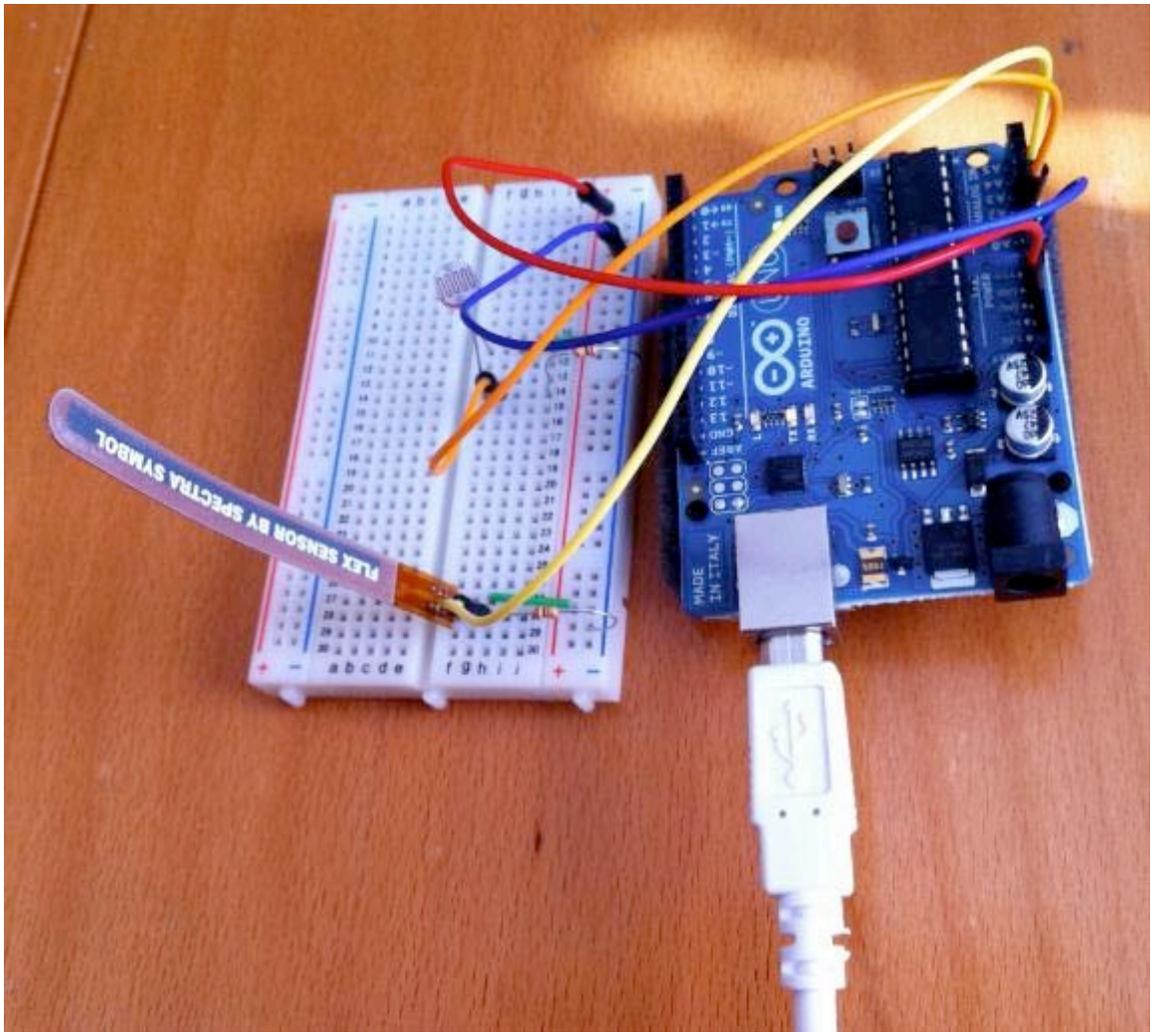
```

15.9. Reading Serial Data from Arduino in PD

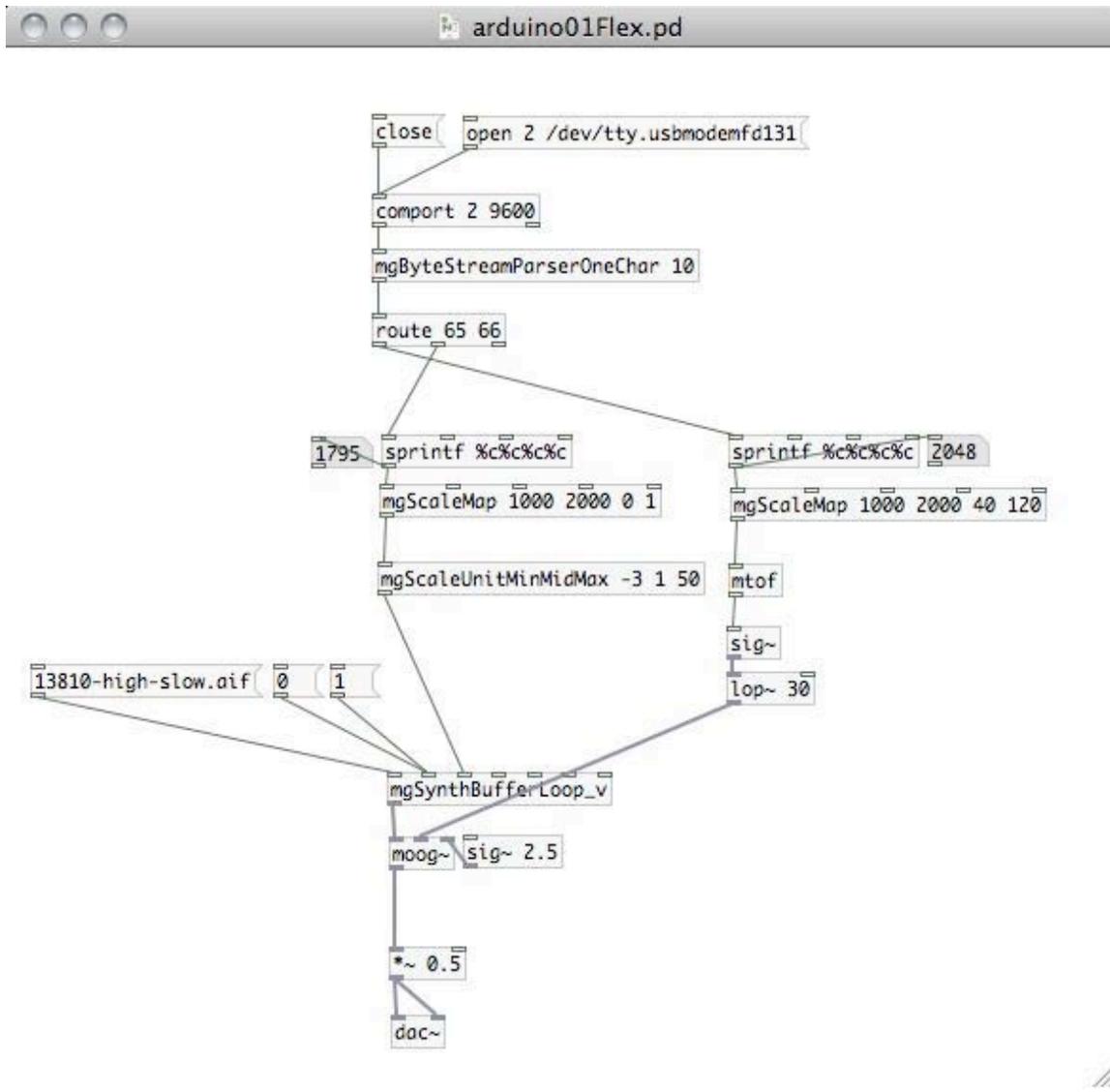
- [comport] object reads bytes from named serial port
- Byte stream needs to be broken into chunks: [mgByteStreamPaserOneChar] divides byte stream into lists based on a sentinel byte (\n, char 10, is a good choice).
- A header byte (e.g., ASCII A and B) can be used to tag individual message from different inputs
- After breaking into lists, can use [route] to get bytes for each message, and reform three-byte integers into numbers with [sprintf]

15.10. Arduino/Pd Instrument: Flex sensor and photo resistor

- Upload code to Arduino first, then open serial connection in Pd
- Flex sensor controls playback rate of looping sample; photo-resistor controls a low-pass filter
- Arduino and Breadboard



- Pd patch:



15.11. Alternative Sensors and Inputs for Arduino and other Microcontrollers

- Force Sensitive Resistors (FSR) (pressure sensor)



Photo courtesy of SparkFun Electronics.

- Flex sensors (\$13 for 4.5 inch)

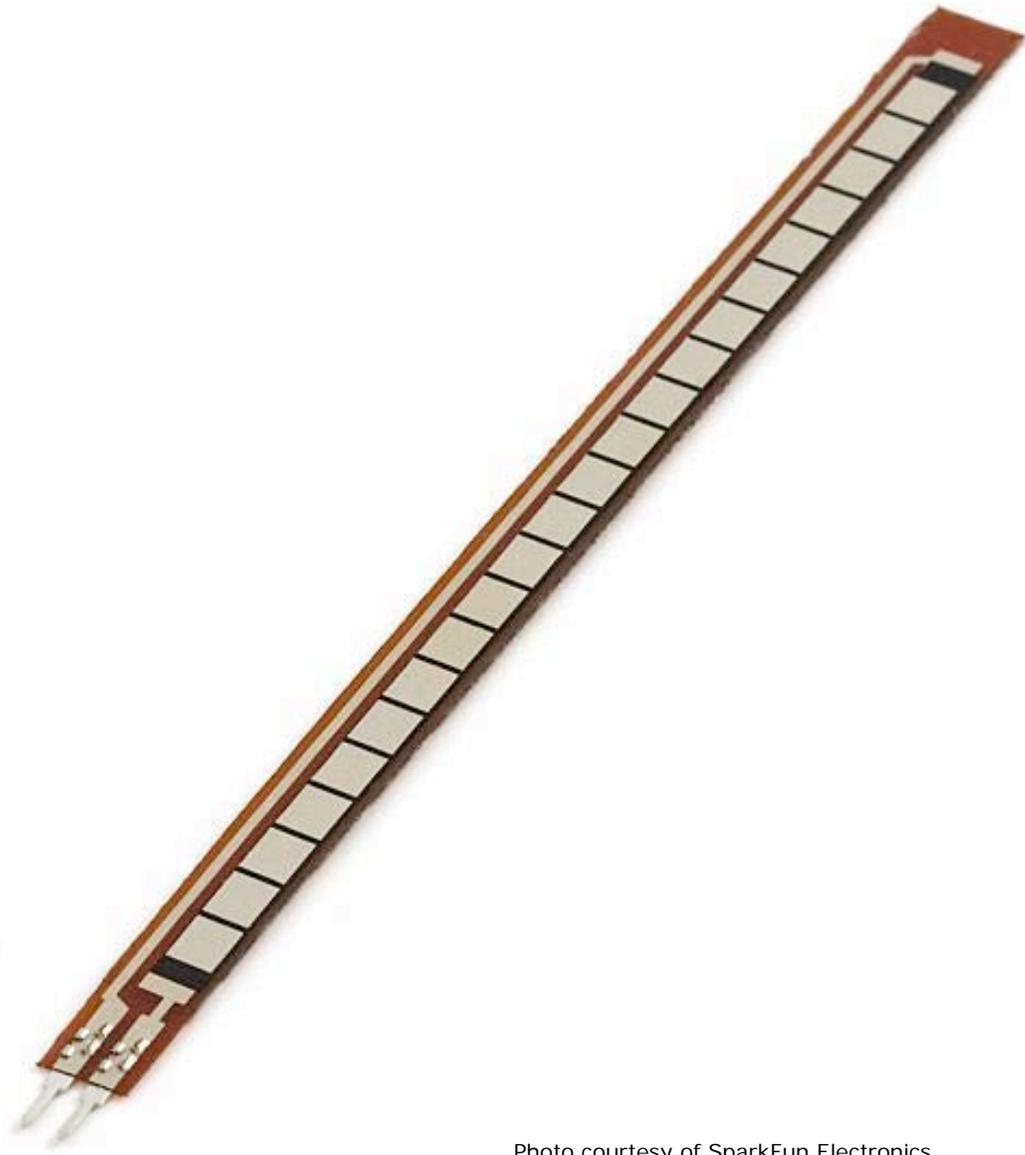


Photo courtesy of SparkFun Electronics.

- SoftPot Membrane Potentiometers (\$15)



Photo courtesy of SparkFun Electronics.

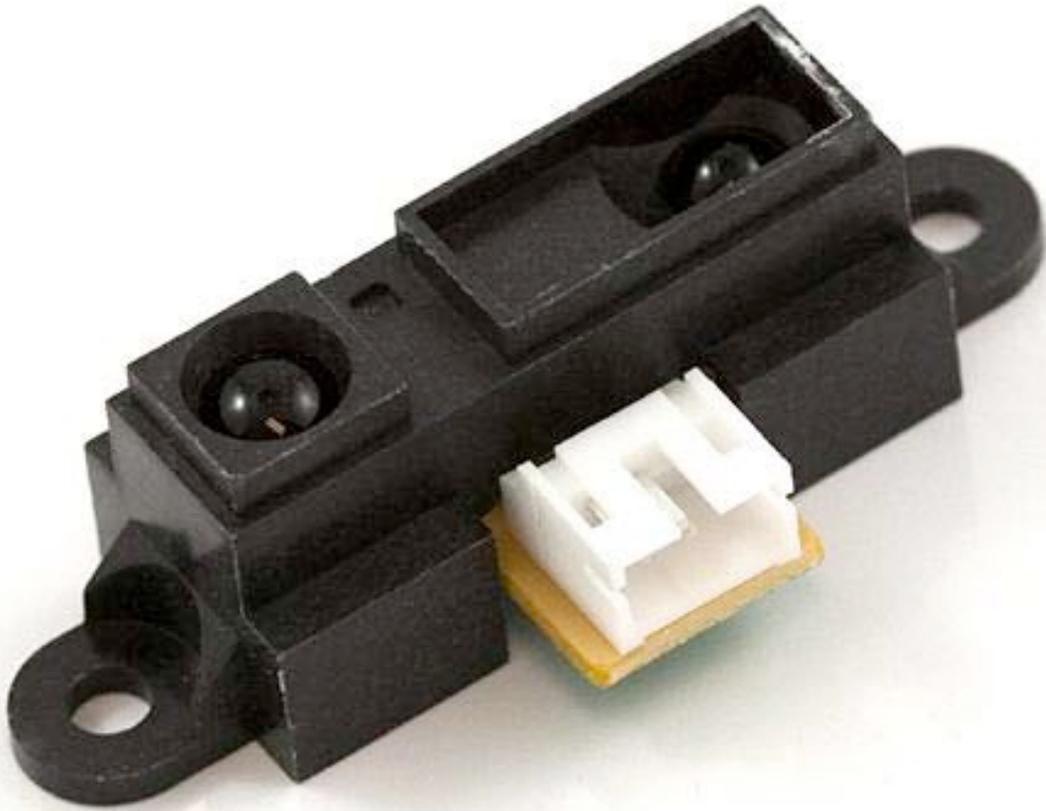


Photo courtesy of SparkFun Electronics.

- Compass Module: HMC6352 (\$35)



Photo courtesy of SparkFun Electronics.

- Triple Axis Accelerometer: ADXL330 (\$25)

Or, buy a Wiimote

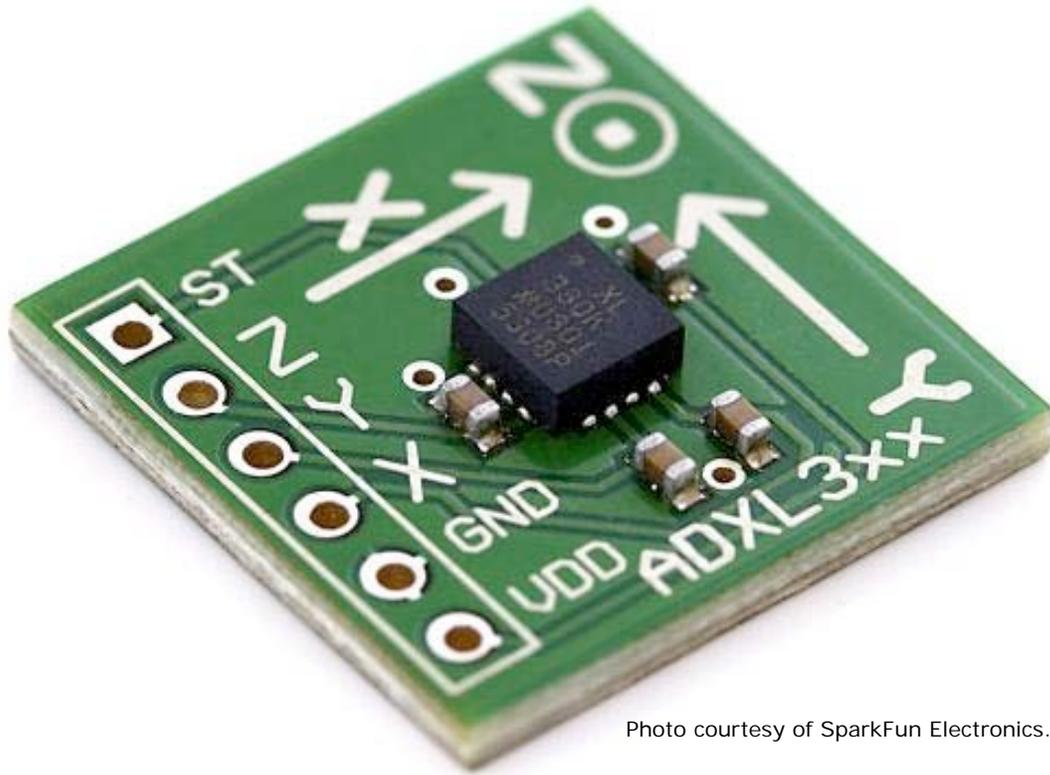


Photo courtesy of SparkFun Electronics.

15.12. Arduino/Pd Instrument: DangerBeat

- Sparkfun Danger Shield (\$30)

Bundles three sliders, three buttons, temperature, light, and knock sensor, plus LEDs, piezo buzzer, and 7-segment LED

- Arduino and DangerShield

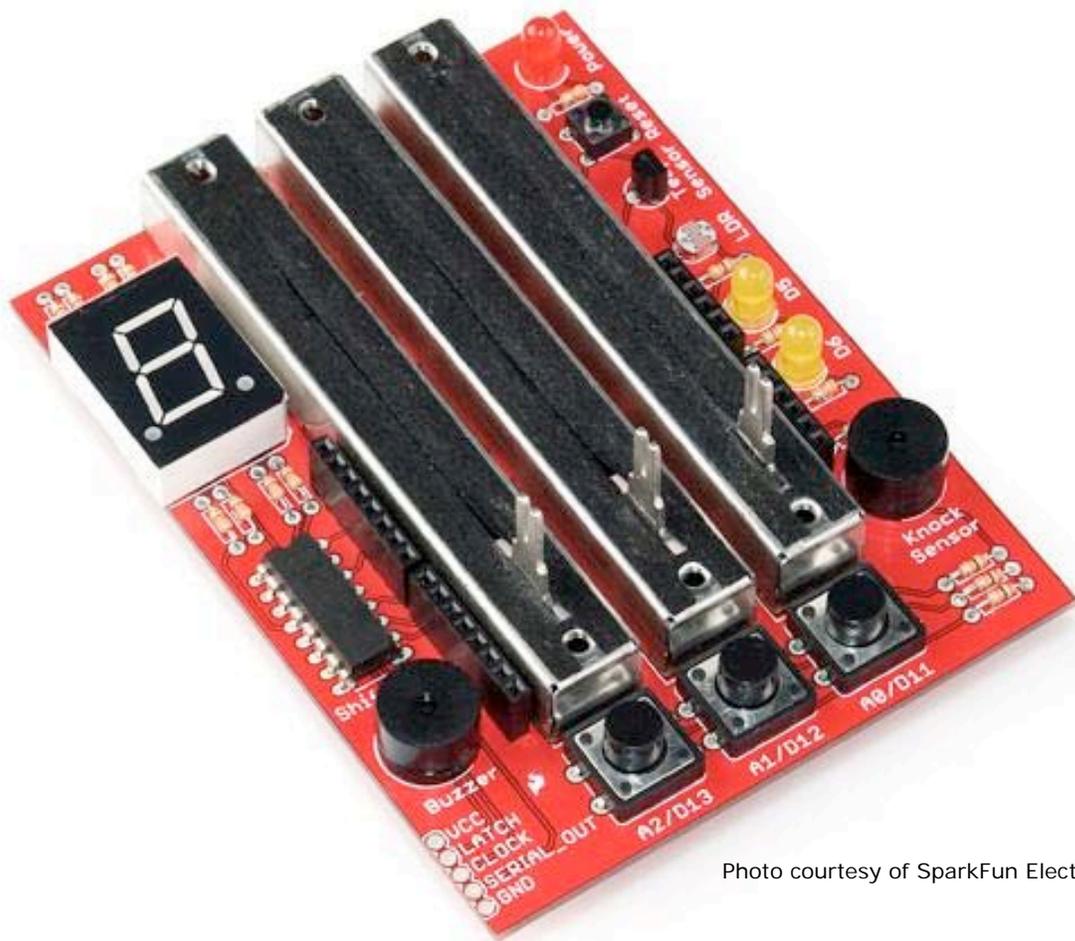


Photo courtesy of SparkFun Electronics.

- Instrument controls beat of two drum synthesis
- `martingale/interfaces/dangerShield/dangerShield.pde`: outputs data from all sensors mapped between 1000 and 2000
- `[mgHwDangerShield]`: package and manage data streams

MIT OpenCourseWare
<http://ocw.mit.edu>

21M.380 Music and Technology: Live Electronics Performance Practices
Spring 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.