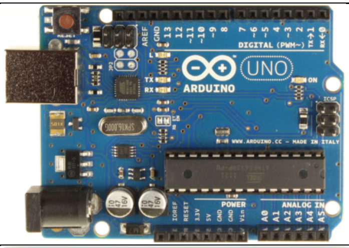
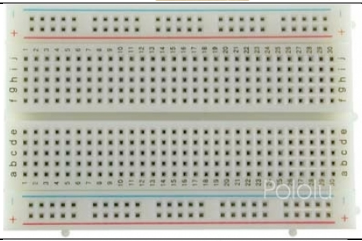
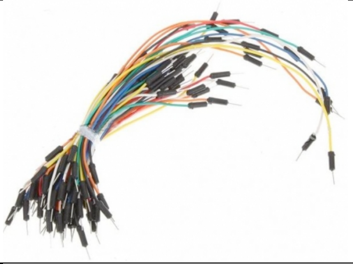
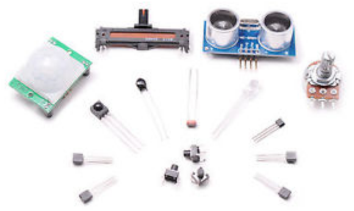
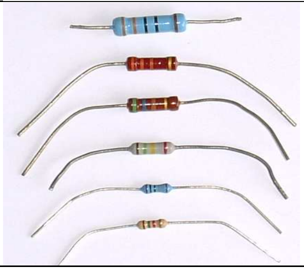
Quick Guide To Using Arduino, Sensors and Max/Max4Live

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers and anyone interested in creating interactive objects or environments.

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators.

You will need:

An Arduino This connects to the computer via USB and includes analogue and digital I/O	
A breadboard Used for prototyping new ideas quickly. Connect from +5V on Arduino to + and from GND on Arduino to -	
Some jumper leads Used to make connections on the breadboard when prototyping. When you're happy with your setup it's best to used soldered connections rather than jumper leads.	
Some sensors	
Some resistors We're using two resistors 10k (for some sensors) colours: *brown black orange gold* 1Meg (for piezo) colours: *brown black green gold*	

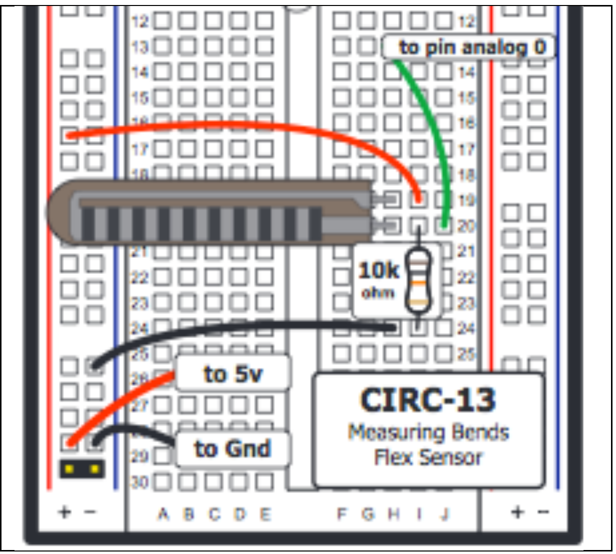
1. If you’re working from home download the Arduino software from http://www.arduino.cc/

2. There are a number of ways to interface Max and Arduino, *sensorbox* is the easiest to work with. Download the sensorbox patches here http://cycling74.com/toolbox/sensorbox/

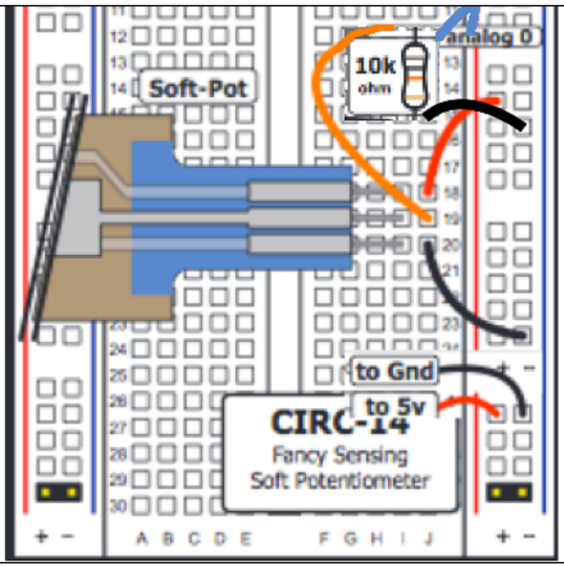
3. Don’t plug in the USB lead just yet, decide on a couple of sensors you want to use and plug them into the breadboard as outlined below

Plugging in Sensors

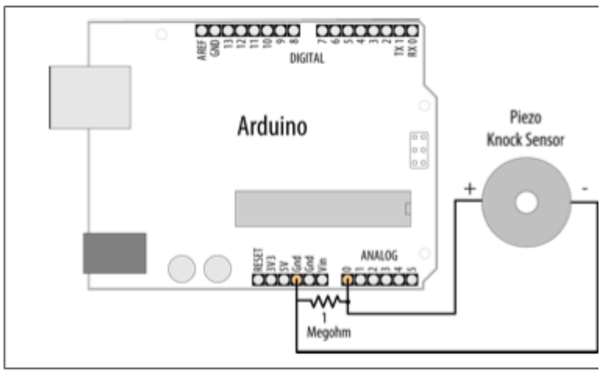
4. Force sensitive resistor (Bb = breadboard). The Arduino cannot directly sense resistance (it senses voltage) so we have to use a resistor.

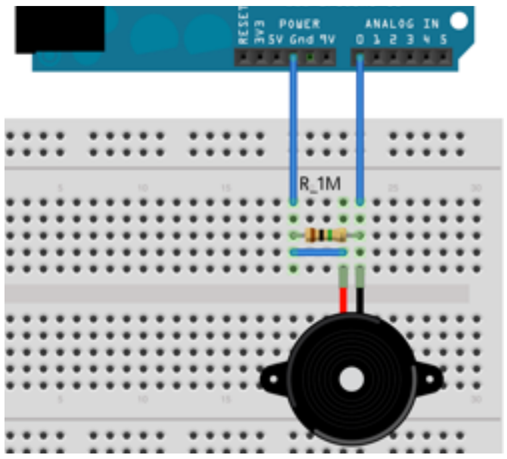
1. + on Bb to 5v Arduino 2. - on Bb to GND Arduino 3. Plug in force sensor 4. From + on Bb to force sensor pin 5. 10k resistor to force pin and another row 6. Resistor to Bb -	

5. Ribbon sensor also known as softpot, these work the same way as a normal pot (potentiometer)

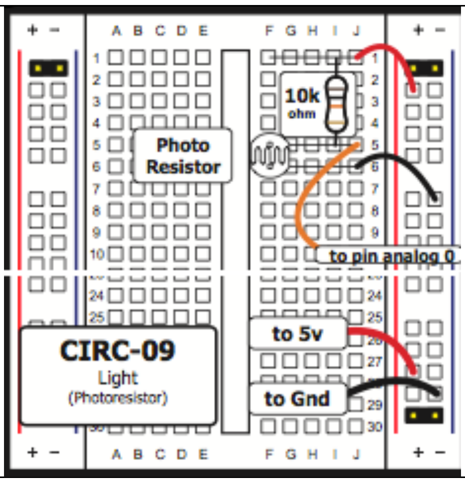
1. + on Bb to 5v Arduino - on Bb to GND Arduino 2. Plug in ribbon sensor 3. + on breadboard to outer ribbon sensor pin 4. GND on breadboard to outer ribbon sensor pin 5. Centre sensor pin to new row 6. 10k resistor in same row 7. Same row to analog in 8. Other end of resistor to GND on breadboard	

6. Piezo. A Piezo sensor responds to vibration. It works best when connected to a larger surface that vibrates. A Piezo sensor, also known as a knock sensor, produces a voltage in response to physical stress. The more it is stressed, the higher the voltage

1. + on Bb to 5v Arduino 2. - on Bb to GND Arduino 3. Plug in piezo, note 1 red, 1 black wire 4. From - on Bb to 1M resistor 5. 1M resistor to black wire on piezo 6. Red wire to 1M resistor then to analog in	 DIGITAL Arduino Piezo Knock Sensor + - ANALOG 1 Megohm

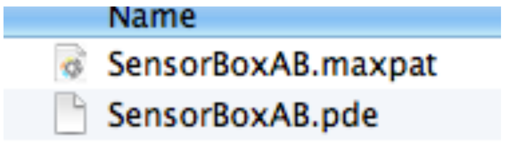


9. Light Dependent Resistor This sensor responds to changes in light levels. A low value will occur when the sensor is well lit while a high value will occur when it is in darkness.

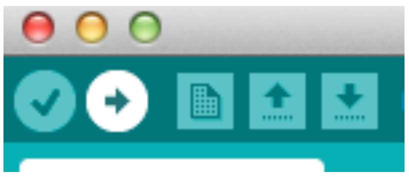
1. + on Bb to 5v Arduino 2. - on Bb to GND Arduino 3. Plug in light sensor 4. From + on Bb to light sensor pin 5. 10k resistor to light pin and another row 6. Resistor to Bb -	 10k ohm Photo Resistor to pin analog 0 to 5v to Gnd CIRC-09 Light (Photoresistor)

Connecting The Arduino

7. Plug in the USB cable. Now the USB cable is plugged in don't add or remove any cables. If you need to change any wiring first disconnect the USB port.
8. Open the Arduino software (use spotlight and search for Arduino) and open the Sensorbox example you downloaded earlier: .pde extension


Name
SensorBoxAB.maxpat
SensorBoxAB.pde

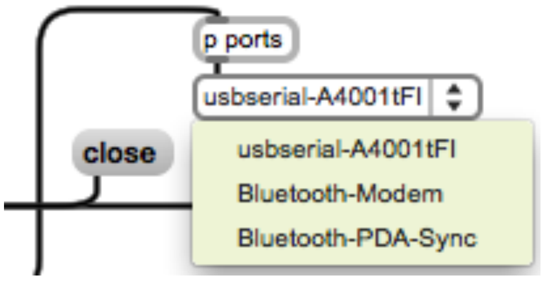
9. Press the upload button, to upload your sketch to the board (right arrow highlighted below). You should see the LED on your Arduino flash.



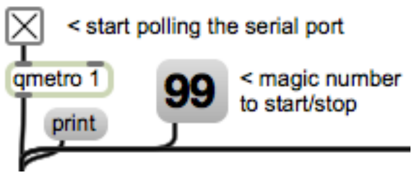
Working With Max/Max4Live

10. Open the example Sensorbox Max patch you downloaded earlier: .maxpat extension

11. Press the print message, this will list the ports available on your computer. Then click the dropdown menu in the Max patch to select the USB port.

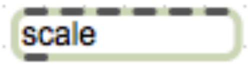


12. Turn the Max metronome on, and click the large **99** message all being well you should now have input from your sensor. Don't worry if the sliders for other analogue inputs move, unconnected inputs are *floating inputs* and will disappear when connected to ground.



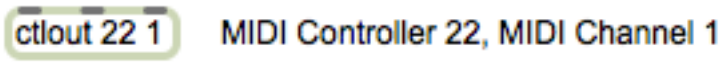
13. If you have crosstalk between some sensors you need to add a pull-down resistor: a 10k resistor attached between the analog input and the Arduino's Ground that sets the value to zero if there is no input.

14. You may need to rescale the output of the sensor so it is in appropriate range. You can either multiply to get to your target range e.g. (0 to 1 input, if you want MIDI output 0 to 127 then * 127), or use the *scale* object.

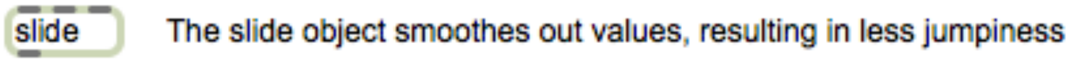

scale

Working With Live

15. From Max send the sensor output out as a MIDI controller as we have done previously.


ctlout 22 1 MIDI Controller 22, MIDI Channel 1

16. This will then be sent out using a ctlout object. You can then use Ableton Live's MIDI mapping [CMD M], to select the effect parameter you want to control.
17. You may find that the output of the sensor is a little noisy. If so you can use the *slide* object to help minimise noise and smooth the output


slide The slide object smoothes out values, resulting in less jumpiness

Designing Your Augmented Instrument

18. Connecting the sensors and ensuring they work is just one part of the design process for your augmented instrument or New Musical Instrument. There are three very important questions you need to ask and answer:

    a. Which musical parameters could I control to make my instrument increase the *musically expressive capabilities* of your instrument?

    b. What sensors and controls could I add allow the player to use the instrument in a *musically expressive* way

    c. What spare 'control bandwidth' does the performer have? Where should sensors be added to allow the performer to add extra *musical expression* without interfering with their natural playing technique? A musician who is already using both hands to play may not easily be able to use extra buttons and sliders. However you may be able to place sensors in easily accessible positions or use other movements e.g. tilt, rotation.

19. Consider how you can avoid false trigging e.g. You may not want pitchshift to be continuously controlled by tilting the instrument as it may cause uncontrollable and unpredictable shifts - though you may decide you like this!

20. Research other implementations of augmented instruments to see how other people have attempted to solve these problems (lots of examples on Moodle).

21. Remember to think about audio tracking, aside from sensors the patch could respond to changes in e..g:

    a. Pitch
    b. Amplitude
    c. Pitch ranges
    d. Scale notes/non-scale notes
    e. Playing fast or slow and any combination of these

    Look through the handout from workshop 4 for more details and examples (available on Moodle).

22. Here is a list of available Sensors
    a. Distance sensors
    b. Light sensors
    c. Ribbon sensors
    d. Acceleromotor gyroscopic
    e. Gyroscopic sensor
    f. Temperature sensor
    g. Force sensor
    h. Flex sensor

23. Final reminder:
    a. The aim is to add further *musical expression*, what other things may it be useful to control? E.g. timbre, harmonies, distortion, feedback, delay, a granular processes, synth?
    b. Consider one to many mappings that may allow you to think of a sensor in a musical way e.g. a sensor controlling *distance,* that alters reverb, volume and brightness simultaneously to change apparent distance from the listener
    c. Consider the *ergonomics* of your augmented instrument carefully – which sensors you choose and where you place them will be crucial to the playability of your instrument.
    d. *Test and refine*. Get others in your group to make suggestions and try the instrument out.

Further Reading

- The PERXE Unit Handbook *make sure you have read this carefully*
- Margolis, M (2012) *Arduino Cookbook* California USA: O'Reilly
- *moodle.port.ac.uk* Many examples of Augmented Instruments on the PERXE site. Also revisit the previous PERXE workshop handouts
- *nime.org* New Instruments for Musical Expression Conference
- Resistor Colour Codes *http://www.1728.org/resisclr.htm*

- Hugill, A. (2012) *The Digital Musician* 2nd Edition Oxford Routledge 978-0415806602
- Holmes, T. (2012) *Electronic and Experimental Music: Technology, Music and Culture* [4th Edition[Oxford Routledge 978-0415896368

You may wish to plan some of you ideas out on paper: