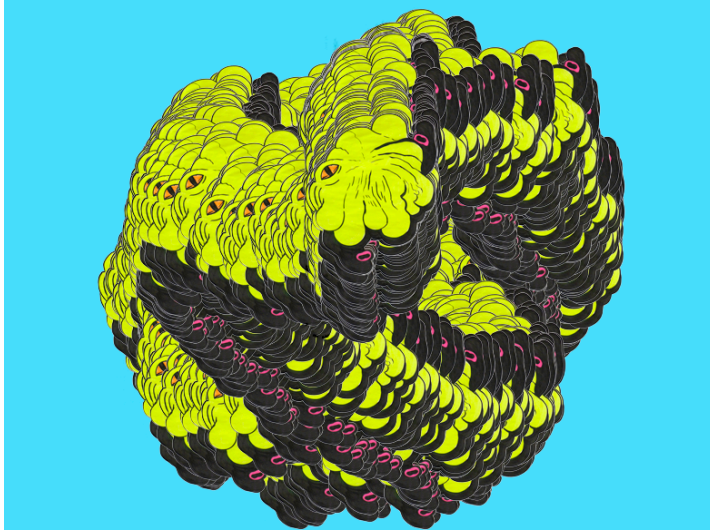


# Read analog data directly in Processing

## Arduino+Processing workshop 2

by Esteban García  
GPL, 2012



This workshop presents a fast and easy way to use data received from an analog sensor in Processing. Participants will learn to utilize the Arduino and prototype electronic boards to read meaningful data from the environment. The sensors can be affected by the light, the orientation, or a user's physical input. For this matter we will use photocell, accelerometer, and potentiometer.

### Preparation steps

1. Upload the "standardFirmata" example from the Arduino IDE in your Arduino board. If you are using Arduino UNO, there is a standard firmata for UNO.
2. Install Arduino library for processing in your libraries folder in the processing sketchbook
3. Test the following code for the Led blink to test that Arduino and processing are communicating correctly: (this code is from the arduino playground)

```
import processing.serial.*;  
import cc.arduino.*;
```

```
Arduino arduino;  
int ledPin = 13;
```

```
void setup()  
{  
  //println(Arduino.list());  
  arduino = new Arduino(this, Arduino.list()[0], 57600);
```

```

    arduino.pinMode(ledPin, Arduino.OUTPUT);
}

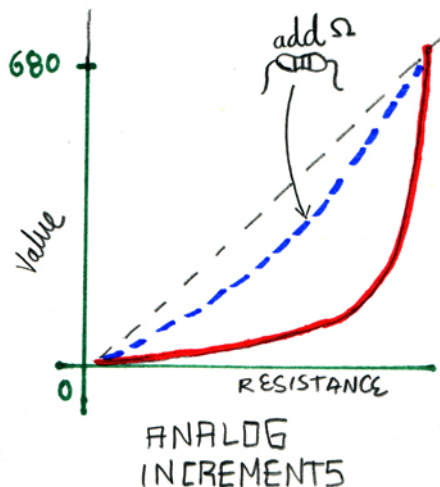
void draw()
{
    arduino.digitalWrite(ledPin, Arduino.HIGH);
    delay(1000);
    arduino.digitalWrite(ledPin, Arduino.LOW);
    delay(1000); }

```

4. If the LED in the Arduino blinks it means that your “hello world” between the Arduino firmata and processing is working.

## Analog Inputs

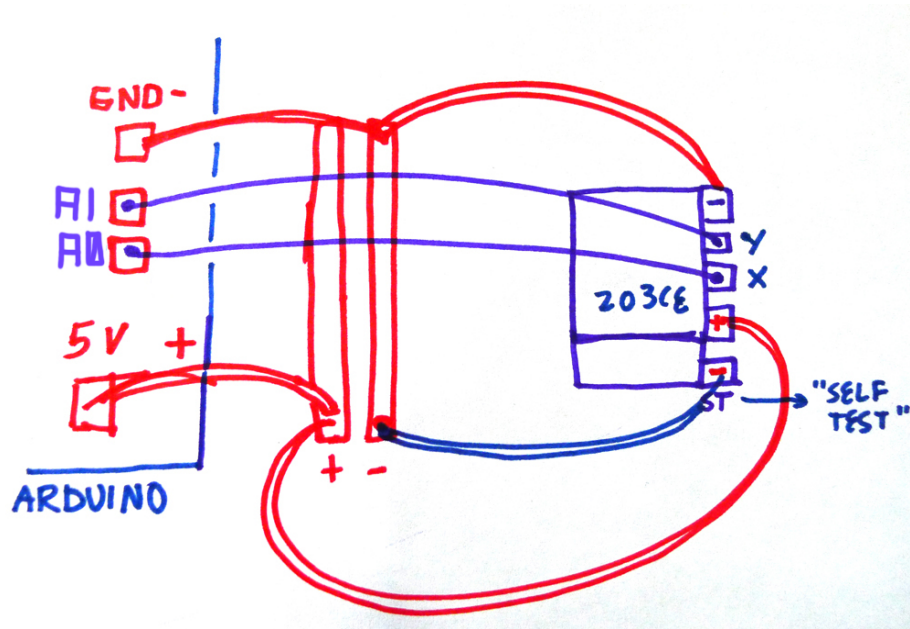
Depending on the artist's or designer's choice to interface with the environment, we will use different sensors. The photocell is sensitive to light, the potentiometer is a knob, and the accelerometer responds to its inclination. In all these cases, the user is manipulating data, and the data can be meaningful for creating user experiences. We assume that environment variables can be also computer variables or parameters, and therefore used to manipulate values in the screen. Every sensor has a different behavior, but when we read physical data in the computer we will have numbers. These numbers can be used as parameters of a computer program.



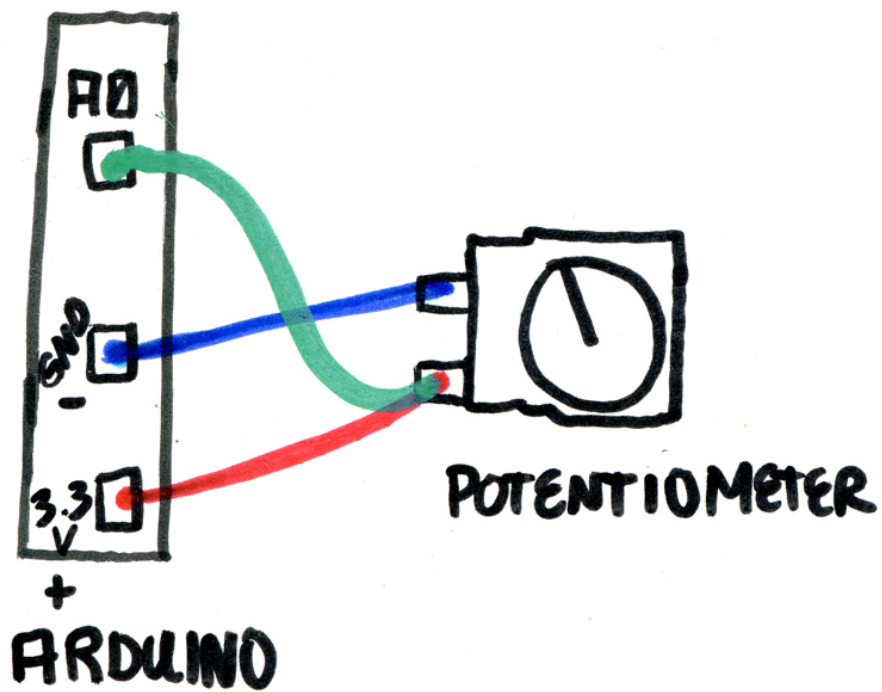
The behavior of the sensor will vary and we can have almost uniform or exponential increments depending on the sensor. Sometimes the values read will range between 0 – 1023, or as in this workshop, from 0 - 680. The difference will depend if you connect to the Arduino's 5V or 3.3V respectively.

## Test circuits

### 1. Accelerometer



### 2. Potentiometer



### 3. Photocell

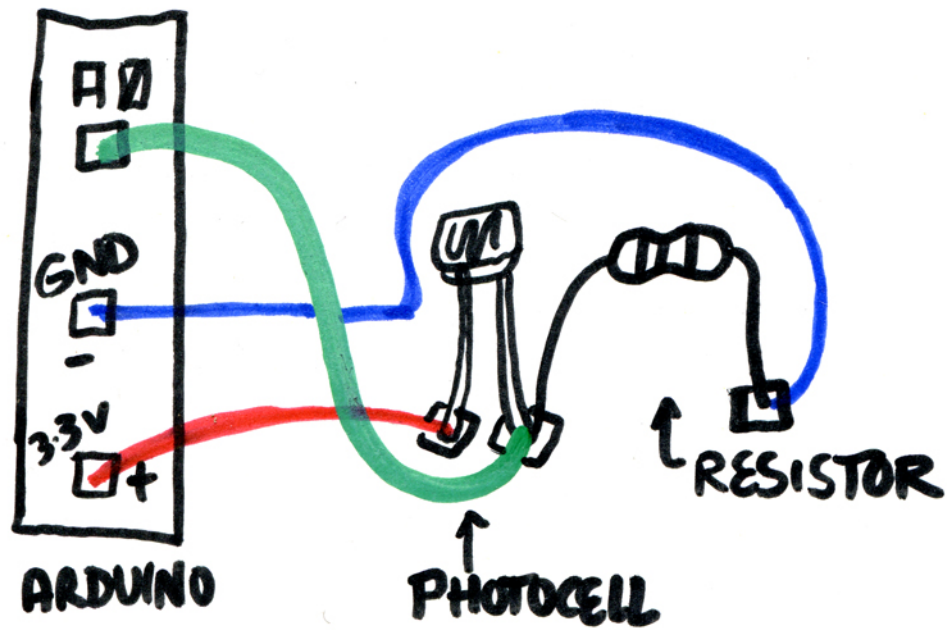
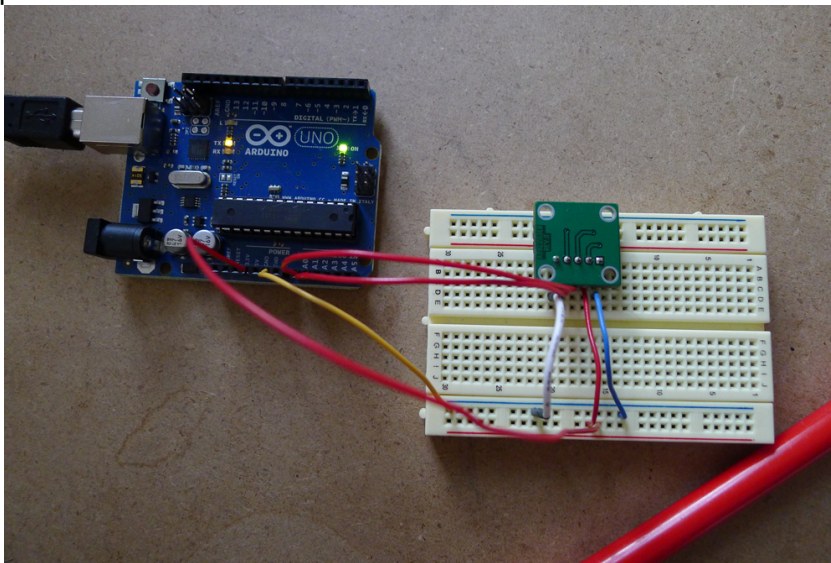


photo here:



\*Be careful to not change the polarities since this might affect the accelerometer.

## Reading 1 variable

Use the following code in processing or “arduino\_variable\_analog.pde”

```
import processing.serial.*;
import cc.arduino.*;

Arduino arduino; //creates arduino object

color back = color(64, 218, 255); //variables for the 2 colors

int sensor= 0;
int read;

float value;
PImage a;

void setup() {
  size(800, 600);
  arduino = new Arduino(this, Arduino.list()[0], 57600); //sets up
  arduino
  arduino.pinMode(sensor, Arduino.INPUT); //setup pins to be input (A0
  =0?)

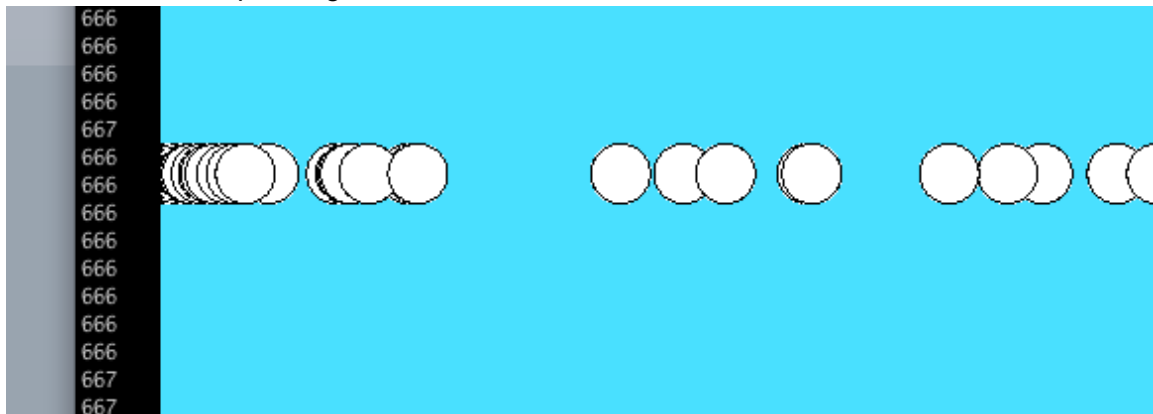
  background(back);
}

void draw() {

  read=arduino.analogRead(sensor);
  background(back);
  println (read);
  value=map(read, 0, 680, 0, width); //use to callibrate
  ellipse(value, 300,30, 30);

}
```

Use the “println” command to output your sensor’s minimum and maximum values. They will be different depending on the context.



## Reading 2 variables and adding image (accelerometer)

Use the following code in processing or “arduino\_variable\_analog.pde”

```
import processing.serial.*;
import cc.arduino.*;

Arduino arduino; //creates arduino object

color back = color(64, 218, 255); //variables for the 2 colors

int xpin= 0;
int ypin= 1;
float value=0;
PImage a;

void setup() {
  size(800, 600);
  arduino = new Arduino(this, Arduino.list()[0], 57600); //sets up
  arduino
  a = loadImage("mar.png");

  arduino.pinMode(xpin, Arduino.INPUT); //setup pins to be input (A0
=0?)
  arduino.pinMode(ypin, Arduino.INPUT);
  background(back);
}

void draw() {

  noStroke();

  //ellipse(arduino.analogRead(xpin)-130, arduino.analogRead(ypin)-
130,30, 30); //needs to be calibrated for X and Y separately
  image(a, arduino.analogRead(xpin)-180, arduino.analogRead(ypin)-300);

}
```

This program tracks down the X and Y values from the accelerometer and displays them altering the position of a graphic in the screen. It needs to be calibrated for both X and Y parameters.

Explore the possibilities of using this analog data in your future creative projects!