This example demonstrates one techinque for calibrating sensor input. The board takes sensor readings for five seconds during the startup, and tracks the highest and lowest values it gets. These sensor readings during the first five seconds of the sketch execution define the minimum and maximum of expected values for the readings taken during the loop.

Hardware Required

* Arduino or Genuino board
* LED
* analog sensor (a photoresistor will do)
* 10k ohm resistor
* 220 ohm resistor
* hook-up wires
* breadboard

Circuit

Analog sensor (e.g. potentiometer, light sensor) on Analog input 2. LED on Digital pin 9.

click the image to enlarge

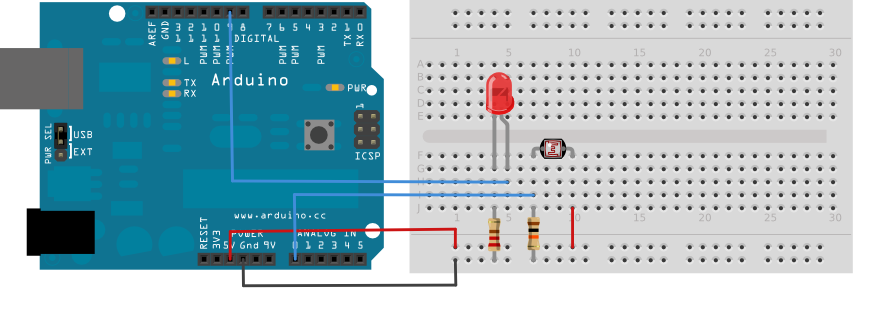
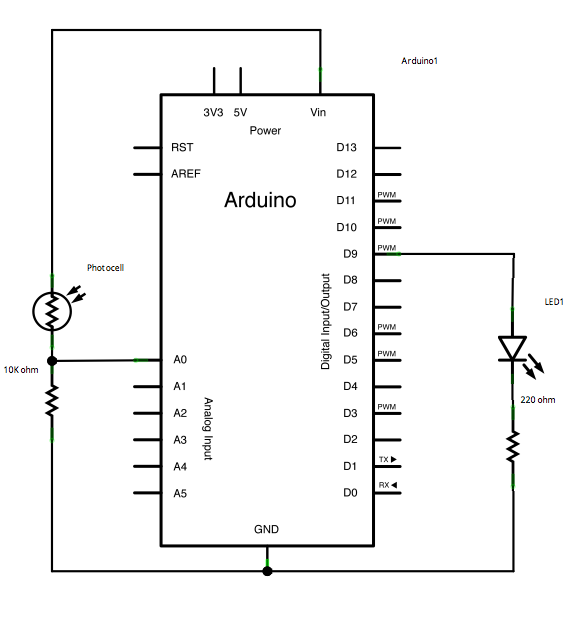
[](https://www.arduino.cc/en/uploads/Tutorial/calibration.png)

image developed using [Fritzing](http://www.fritzing.org/). For more circuit examples, see the [Fritzing project page](http://fritzing.org/projects/)

Connect an LED to digital pin 9 with a 220 ohm current limiting resistor in series. Connect a photoresistor to 5V and then to analog pin 0 with a 10K ohm resistor to ground.

Schematic

click the image to enlarge

[](https://www.arduino.cc/en/uploads/Tutorial/calibration_sch.png)

Code

Before the setup, you set initial values for the minimum and maximum like so:

int sensorMin = 1023; // minimum sensor value

int sensorMax = 0; // maximum sensor value

These may seem backwards. Initially, you set the minimum high and read for anything lower than that, saving it as the new minimum. Likewise, you set the maximum low and read for anything higher as the new maximum, like so:

// calibrate during the first five seconds

while (millis() < 5000) {

sensorValue = analogRead(sensorPin);

// record the maximum sensor value

if (sensorValue > sensorMax) {

sensorMax = sensorValue;

}

// record the minimum sensor value

if (sensorValue < sensorMin) {

sensorMin = sensorValue;

}

}

This way, any further readings you take can be mapped to the range between this minimum and maximum like so:

// apply the calibration to the sensor reading

sensorValue = map(sensorValue, sensorMin, sensorMax, 0, 255);

Here's the whole program:

*/\*  
  Calibration  
  
  Demonstrates one technique for calibrating sensor input. The sensor readings  
  during the first five seconds of the sketch execution define the minimum and  
  maximum of expected values attached to the sensor pin.  
  
  The sensor minimum and maximum initial values may seem backwards. Initially,  
  you set the minimum high and listen for anything lower, saving it as the new  
  minimum. Likewise, you set the maximum low and listen for anything higher as  
  the new maximum.  
  
  The circuit:  
  - analog sensor (potentiometer will do) attached to analog input 0  
  - LED attached from digital pin 9 to ground  
  
  created 29 Oct 2008  
  by David A Mellis  
  modified 30 Aug 2011  
  by Tom Igoe  
  
  This example code is in the public domain.  
  
  http://www.arduino.cc/en/Tutorial/Calibration  
\*/*  
  
*// These constants won't change:*  
const int sensorPin = A0;    *// pin that the sensor is attached to*  
const int ledPin = 9;        *// pin that the LED is attached to*  
  
*// variables:*  
int sensorValue = 0;         *// the sensor value*  
int sensorMin = 1023;        *// minimum sensor value*  
int sensorMax = 0;           *// maximum sensor value*  
  
  
void **setup**() {  
  *// turn on LED to signal the start of the calibration period:*  
  pinMode(13, OUTPUT);  
  digitalWrite(13, HIGH);  
  
  *// calibrate during the first five seconds*  
  while (millis() < 5000) {  
    sensorValue = analogRead(sensorPin);  
  
    *// record the maximum sensor value*  
    if (sensorValue > sensorMax) {  
      sensorMax = sensorValue;  
    }  
  
    *// record the minimum sensor value*  
    if (sensorValue < sensorMin) {  
      sensorMin = sensorValue;  
    }  
  }  
  
  *// signal the end of the calibration period*  
  digitalWrite(13, LOW);  
}  
  
void **loop**() {  
  *// read the sensor:*  
  sensorValue = analogRead(sensorPin);  
  
  *// apply the calibration to the sensor reading*  
  sensorValue = map(sensorValue, sensorMin, sensorMax, 0, 255);  
  
  *// in case the sensor value is outside the range seen during calibration*  
  sensorValue = constrain(sensorValue, 0, 255);  
  
  *// fade the LED using the calibrated value:*  
  analogWrite(ledPin, sensorValue);  
}

[[Get Code]](https://www.arduino.cc/en/Tutorial/Calibration?action=sourceblock&num=1)

See Also:

* [while](https://www.arduino.cc/en/Reference/While)()
* [millis](https://www.arduino.cc/en/Reference/Millis)()
* [constrain](https://www.arduino.cc/en/Reference/Constrain)()
* [map](https://www.arduino.cc/en/Reference/Map)()
* [If](https://www.arduino.cc/en/Reference/If)
* [AnalogInOutSerial](https://www.arduino.cc/en/Tutorial/AnalogInOutSerial) - Read an analog input pin, map the result, and then use that data to dim or brighten an LED.
* [AnalogInput](https://www.arduino.cc/en/Tutorial/AnalogInput) - Use a potentiometer to control the blinking of an LED.
* [AnalogWriteMega](https://www.arduino.cc/en/Tutorial/AnalogWriteMega) - Fade 12 LEDs on and o¬ff, one by one, using an Arduino or Genuino Mega board.
* [Fading](https://www.arduino.cc/en/Tutorial/Fading) - Use an analog output (PWM pin) to fade an LED.
* [Smoothing](https://www.arduino.cc/en/Tutorial/Smoothing) - Smooth multiple readings of an analog input.