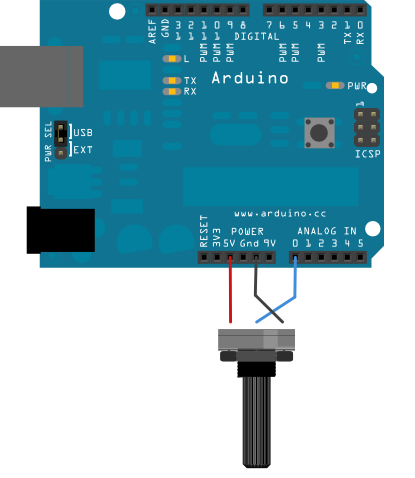
Smoothing

This sketch reads repeatedly from an analog input, calculating a running average and printing it to the computer. This example is useful for smoothing out the values from jumpy or erratic sensors, and also demonstrates the use of [arrays](https://www.arduino.cc/en/Reference/Array)to store data.

Hardware

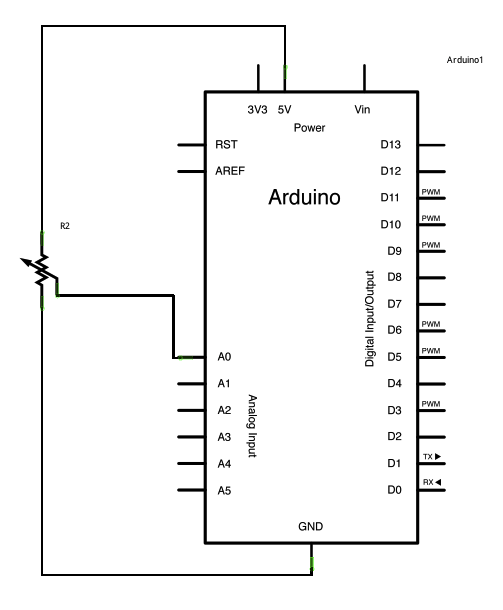
* Arduino or Genuino Board
* 10k ohm potentiometer

Circuit



Connect one pin of a potentiometer to 5V, the center pin to analog pin 0, and the the last pin to ground.

Schematic



Code

The code below sequentially stores 10 readings from your analog sensor into an arrays, one by one. With each new value, the sum of all the numbers is generated and divided, producing an average value which then be used to smooth outlying data. Because this averaging takes place each time a new value is added to the array (rather then waiting for 10 new values, for instance) there is no lag time in calculating this running average.

Altering the size of the array used, by changing numReadings to a larger value will smooth the data collected even further.

*// Define the number of samples to keep track of. The higher the number, the*  
*// more the readings will be smoothed, but the slower the output will respond to*  
*// the input. Using a constant rather than a normal variable lets us use this*  
*// value to determine the size of the readings array.*  
const int numReadings = 10;  
  
int readings[numReadings];      *// the readings from the analog input*  
int readIndex = 0;              *// the index of the current reading*  
int total = 0;                  *// the running total*  
int average = 0;                *// the average*  
  
int inputPin = A0;  
  
void **setup**() {  
  *// initialize serial communication with computer:*  
  Serial.begin(9600);  
  *// initialize all the readings to 0:*  
  for (int thisReading = 0; thisReading < numReadings; thisReading++) {  
    readings[thisReading] = 0;  
  }  
}  
  
void **loop**() {  
  *// subtract the last reading:*  
  total = total - readings[readIndex];  
  *// read from the sensor:*  
  readings[readIndex] = analogRead(inputPin);  
  *// add the reading to the total:*  
  total = total + readings[readIndex];  
  *// advance to the next position in the array:*  
  readIndex = readIndex + 1;  
  
  *// if we're at the end of the array...*  
  if (readIndex >= numReadings) {  
    *// ...wrap around to the beginning:*  
    readIndex = 0;  
  }  
  
  *// calculate the average:*  
  average = total / numReadings;  
  *// send it to the computer as ASCII digits*  
  Serial.println(average);  
  delay(1);        *// delay in between reads for stability*  
}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Your full name | Tel. | Email | Group | Su23 |
| Nguyễn Xuân Trung | 0944353601 | [Trungnxqe170172@fpt.edu.vn](mailto:Trungnxqe170172@fpt.edu.vn) | 2 – SE17B02 |  |
| Đoàn Nguyễn Huyền Trang | 0344468198 | trangdnhqe170154@fpt.edu.vn |  |  |
| Phan Phương Sinh | 0522991730 | [Sinhppqe170242@fpt.edu.vn](mailto:Sinhppqe170242@fpt.edu.vn) |  |  |
| Nguyễn Đồng Lợi | 0702772499 | loindqe170249@fpt.edu.vn |  |  |
| Đường Mỹ Hà | 0901130650 | HaDMQE170046@fpt.edu.vn |  |  |
| Screen shot:    https://www.tinkercad.com/things/kipVY1gXyhw?sharecode=uDyVz\_n3jdRWlKPbhJNCPQfu\_wyPwri9K4IqDs2P4yw | | | | |
| How it works?  The code below sequentially stores 10 readings from your analog sensor into an arrays, one by one. With each new value, the sum of all the numbers is generated and divided, producing an average value which then be used to smooth outlying data. Because this averaging takes place each time a new value is added to the array (rather then waiting for 10 new values, for instance) there is no lag time in calculating this running average. | | | | |