David Lee

Spring 2017

Running average.

Introduction:

Often when a sensor reading is taken, there is a measure of noise in that reading. An example of this is with a thermocouple where the voltage read may have spikes or dips in it that can adversely affect a process. One way to eliminate these disturbances is to use a moving average filter.

Background:

The average of a series of numbers is given by this equation:

$$\frac{\sum x_n}{N} = \bar{x}$$

Where N is the number of terms in the series and x_n represents the terms in the series. The effect of changing a term is inversely proportional to the number of terms being averaged.

$$\Delta\bar{x} = \frac{\Delta\, x_n}{N}$$

Or to but it simply: the more terms you average together, the less sensitive the average is to changing.

Library:

While there is no running average filter directly built into Arduino, there is one available here:

https://playground.arduino.cc/Main/RunningAverage

The way the library works is you create a new running average with the number of terms you want in it

$$myRunningAverage = RunningAverage(N);$$

Where N is the number of terms in the running average (must be an integer.)

To add a value use the addValue function.

$$myRunningAverage.addValue(value);$$

Where value is the number that you are giving the running average.

To get the current average use the getAverage function.

$$myRunningAverage.getAverage();$$

This returns the current average as a double.

Other notes:

- When it is first initialized, the running average will be filled with null data. This means that as values are added, the average acts as a filter with less terms in it. For example if I have a running average for 25 values but only add 15, the function acts as though it was a filter for only 15 values. When I add another value, the filter has 16 values to average. This repeats until 25 values have been passed after which point old values are thrown away.
- Adding this type of filtering does cause lag in a system's response. Because of this it may be desirable to create for loops around data acquisition lines (such as querying a sensor) to fill up the average and update the value properly.