

Arduino Signal Filtering Library

Generated by Doxygen 1.8.1.2

Fri Nov 30 2012 01:07:02

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	SignalFilter Class Reference	5
3.1.1	Constructor & Destructor Documentation	6
3.1.1.1	SignalFilter	6
3.1.2	Member Function Documentation	6
3.1.2.1	begin	6
3.1.2.2	printSamples	7
3.1.2.3	run	7
3.1.2.4	runBessel	7
3.1.2.5	runChebyshev	7
3.1.2.6	runGrowing	7
3.1.2.7	runGrowing2	7
3.1.2.8	runMedian	7
3.1.2.9	setFilter	7
3.1.2.10	setOrder	8
3.1.3	Member Data Documentation	8
3.1.3.1	_average	8
3.1.3.2	_counter	8
3.1.3.3	_filter	8
3.1.3.4	_helper	8
3.1.3.5	_median	8
3.1.3.6	_order	8
3.1.3.7	_v	8
4	File Documentation	9
4.1	SignalFilter.cpp File Reference	9

4.2	SignalFilter.h File Reference	10
-----	---	----

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

SignalFilter	5
--	---

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

SignalFilter.cpp	9
SignalFilter.h	10

Chapter 3

Class Documentation

3.1 SignalFilter Class Reference

```
#include <SignalFilter.h>
```

Collaboration diagram for SignalFilter:

SignalFilter
<div>- _filter</div> <div>- _order</div> <div>- _average</div> <div>- _median</div> <div>- _helper</div> <div>- _counter</div> <div>- _v</div>
<div>+ SignalFilter()</div> <div>+ begin()</div> <div>+ setFilter()</div> <div>+ setOrder()</div> <div>+ printSamples()</div> <div>+ run()</div> <div>- runChebyshev()</div> <div>- runBessel()</div> <div>- runMedian()</div> <div>- runGrowing()</div> <div>- runGrowing2()</div>

Public Member Functions

- [SignalFilter \(\)](#)
Constructor.

- void `begin` ()
Begin function: set default filter options.
- void `setFilter` (char filter)
setFilter(char filter): Select filter: 'c' -> Chebyshev, 'b' -> Bessel
- void `setOrder` (int order)
selectOrder(int order): Select filter order (1 or 2)
- void `printSamples` ()
printSamples: Print out some samples (for debugging)
- int `run` (int data)
run: calls the actual filter: input=rawdata, output=filtered data

Private Member Functions

- int `runChebyshev` (int data)
runChebyshev: Runs the actual filter: input=rawdata, output=filtered data
- int `runBessel` (int data)
runBessel: Runs the actual filter: input=rawdata, output=filtered data
- int `runMedian` (int data)
runBessel: Runs the actual filter: input=rawdata, output=filtered data
- int `runGrowing` (int data)
runGrowing: Runs the actual filter: input=rawdata, output=filtered data: Growing-shrinking filter (fast)
- int `runGrowing2` (int data)
runGrowing2: Runs the actual filter: input=rawdata, output=filtered data: Growing-shrinking filter (smoother)

Private Attributes

- char `_filter`
- int `_order`
- int `_average`
- int `_median`
- int `_helper`
- int `_counter`
- int `_v` [3]

3.1.1 Constructor & Destructor Documentation

3.1.1.1 `SignalFilter::SignalFilter` ()

Constructor.

`SignalFilter` - Library to Filter Sensor Data using digital filters Available filters: Chebyshev & Bessel low pass filter (1st & 2nd order)

References `_average`, `_counter`, `_filter`, `_helper`, `_median`, `_order`, and `_v`.

3.1.2 Member Function Documentation

3.1.2.1 `void SignalFilter::begin` ()

Begin function: set default filter options.

References `setFilter()`, and `setOrder()`.

3.1.2.2 void SignalFilter::printSamples ()

printSamples: Print out some samples (for debugging)

References [_v](#).

3.1.2.3 int SignalFilter::run (int *data*)

run: calls the actual filter: input=rawdata, output=filtered data

References [_filter](#), [runBessel\(\)](#), [runChebyshev\(\)](#), [runGrowing\(\)](#), [runGrowing2\(\)](#), and [runMedian\(\)](#).

3.1.2.4 int SignalFilter::runBessel (int *data*) [private]

runBessel: Runs the actual filter: input=rawdata, output=filtered data

References [_order](#), and [_v](#).

Referenced by [run\(\)](#).

3.1.2.5 int SignalFilter::runChebyshev (int *data*) [private]

runChebyshev: Runs the actual filter: input=rawdata, output=filtered data

References [_order](#), and [_v](#).

Referenced by [run\(\)](#).

3.1.2.6 int SignalFilter::runGrowing (int *data*) [private]

runGrowing: Runs the actual filter: input=rawdata, output=filtered data: Growing-shrinking filter (fast)

References [_helper](#).

Referenced by [run\(\)](#).

3.1.2.7 int SignalFilter::runGrowing2 (int *data*) [private]

runGrowing2: Runs the actual filter: input=rawdata, output=filtered data: Growing-shrinking filter (smoother)

References [_counter](#), and [_helper](#).

Referenced by [run\(\)](#).

3.1.2.8 int SignalFilter::runMedian (int *data*) [private]

runBessel: Runs the actual filter: input=rawdata, output=filtered data

References [_median](#), and [_v](#).

Referenced by [run\(\)](#).

3.1.2.9 void SignalFilter::setFilter (char *filter*)

[setFilter\(char filter\)](#): Select filter: 'c' -> Chebyshev, 'b' -> Bessel

References [_filter](#).

Referenced by [begin\(\)](#).

3.1.2.10 void `SignalFilter::setOrder` (int *order*)

`selectOrder(int order)`: Select filter order (1 or 2)

References [_order](#).

Referenced by [begin\(\)](#).

3.1.3 Member Data Documentation

3.1.3.1 int `SignalFilter::_average` [private]

Referenced by [SignalFilter\(\)](#).

3.1.3.2 int `SignalFilter::_counter` [private]

Referenced by [runGrowing2\(\)](#), and [SignalFilter\(\)](#).

3.1.3.3 char `SignalFilter::_filter` [private]

Referenced by [run\(\)](#), [setFilter\(\)](#), and [SignalFilter\(\)](#).

3.1.3.4 int `SignalFilter::_helper` [private]

Referenced by [runGrowing\(\)](#), [runGrowing2\(\)](#), and [SignalFilter\(\)](#).

3.1.3.5 int `SignalFilter::_median` [private]

Referenced by [runMedian\(\)](#), and [SignalFilter\(\)](#).

3.1.3.6 int `SignalFilter::_order` [private]

Referenced by [runBessel\(\)](#), [runChebyshev\(\)](#), [setOrder\(\)](#), and [SignalFilter\(\)](#).

3.1.3.7 int `SignalFilter::_v[3]` [private]

Referenced by [printSamples\(\)](#), [runBessel\(\)](#), [runChebyshev\(\)](#), [runMedian\(\)](#), and [SignalFilter\(\)](#).

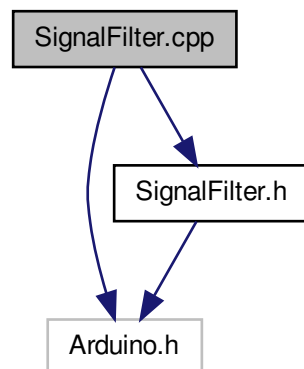
Chapter 4

File Documentation

4.1 SignalFilter.cpp File Reference

```
#include <Arduino.h>
#include <SignalFilter.h>
```

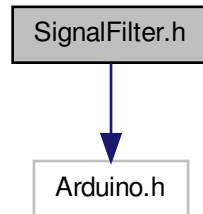
Include dependency graph for SignalFilter.cpp:



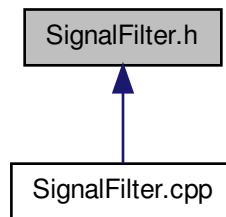
4.2 SignalFilter.h File Reference

```
#include <Arduino.h>
```

Include dependency graph for SignalFilter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [SignalFilter](#)

Index

- [_average](#)
SignalFilter, [8](#)
 - [_counter](#)
SignalFilter, [8](#)
 - [_filter](#)
SignalFilter, [8](#)
 - [_helper](#)
SignalFilter, [8](#)
 - [_median](#)
SignalFilter, [8](#)
 - [_order](#)
SignalFilter, [8](#)
 - [_v](#)
SignalFilter, [8](#)
- [begin](#)
SignalFilter, [6](#)
- [printSamples](#)
SignalFilter, [6](#)
- [run](#)
SignalFilter, [7](#)
- [runBessel](#)
SignalFilter, [7](#)
- [runChebyshev](#)
SignalFilter, [7](#)
- [runGrowing](#)
SignalFilter, [7](#)
- [runGrowing2](#)
SignalFilter, [7](#)
- [runMedian](#)
SignalFilter, [7](#)
- [setFilter](#)
SignalFilter, [7](#)
- [setOrder](#)
SignalFilter, [7](#)
- [SignalFilter](#), [5](#)
 - [_average](#), [8](#)
 - [_counter](#), [8](#)
 - [_filter](#), [8](#)
 - [_helper](#), [8](#)
 - [_median](#), [8](#)
 - [_order](#), [8](#)
 - [_v](#), [8](#)
 - [begin](#), [6](#)
 - [printSamples](#), [6](#)
 - [run](#), [7](#)
 - [runBessel](#), [7](#)
 - [runChebyshev](#), [7](#)
 - [runGrowing](#), [7](#)
 - [runGrowing2](#), [7](#)
 - [runMedian](#), [7](#)
 - [setFilter](#), [7](#)
 - [setOrder](#), [7](#)
 - [SignalFilter](#), [6](#)
 - [SignalFilter](#), [6](#)
 - [SignalFilter.cpp](#), [9](#)
 - [SignalFilter.h](#), [10](#)