

My Project

Generated by Doxygen 1.8.8

Wed Mar 30 2016 16:47:14

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	vaso Namespace Reference	7
4.1.1	Detailed Description	7
4.1.2	Enumeration Type Documentation	8
4.1.2.1	Side	8
4.1.3	Function Documentation	8
4.1.3.1	average	8
4.1.3.2	average	8
4.1.3.3	CurrentDataName	8
4.1.3.4	diff	9
4.1.3.5	fft	9
4.1.3.6	InitialDataName	9
4.1.3.7	mag	9
4.1.3.8	max	9
4.1.3.9	PatientName	10
4.1.3.10	play	10
4.1.3.11	process	10
4.1.3.12	ReadParams	10
4.1.3.13	smooth	11
4.1.3.14	StartProcessing	12
4.1.3.15	WriteParams	12
4.1.4	Variable Documentation	12
4.1.4.1	PATIENT_PATH	12

5	Class Documentation	13
5.1	DataParams Struct Reference	13
5.1.1	Detailed Description	13
5.2	ProcData Struct Reference	13
5.2.1	Detailed Description	13
6	File Documentation	15
6.1	bin/start File Reference	15
6.2	etc/doxygen.config File Reference	15
6.3	makefile File Reference	15
6.4	src/definitions.hpp File Reference	15
6.4.1	Macro Definition Documentation	17
6.4.1.1	ENUM	17
6.4.2	Typedef Documentation	17
6.4.2.1	byte	17
6.4.2.2	cfloat32	17
6.4.2.3	float32	17
6.4.2.4	float64	17
6.4.2.5	sint16	17
6.4.2.6	sint32	17
6.4.2.7	sint64	17
6.4.2.8	sint8	17
6.4.2.9	uint16	17
6.4.2.10	uint32	18
6.4.2.11	uint64	18
6.4.2.12	uint8	18
6.5	src/fileio.hpp File Reference	18
6.6	src/main.cpp File Reference	19
6.6.1	Function Documentation	19
6.6.1.1	main	19
6.7	src/sigmath.hpp File Reference	20
6.8	src/sound.hpp File Reference	21
6.9	src/threadproc.hpp File Reference	22
	Index	24

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

vaso	Functions related to the file I/O use in this program	7
----------------------	---	-------------------

Chapter 2

Class Index

2.1 Class List

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

DataParams	13
ProcData	13

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

makefile	15
bin/start	15
etc/doxygen.config	15
src/definitions.hpp	15
src/fileio.hpp	18
src/main.cpp	19
src/sigmath.hpp	20
src/sound.hpp	21
src/threadproc.hpp	22

Chapter 4

Namespace Documentation

4.1 vaso Namespace Reference

contains functions related to the file I/O use in this program

Enumerations

- enum [Side](#) { [Side::Left](#), [Side::Right](#) }

Functions

- std::string [CurrentDataName](#) ()
- std::string [InitialDataName](#) (auto dir)
- std::string [PatientName](#) ()
- [DataParams](#) [ReadParams](#) (auto filename)
- std::string [WriteParams](#) ([DataParams](#) params, auto filename)
- [float32](#) [average](#) ([float32](#) *data, [uint32](#) size)
- void [average](#) ([float32](#) *data, [float32](#) *avg, [uint8](#) count, [uint32](#) size)
- void [diff](#) ([float32](#) *data, [uint32](#) size)
- void [fft](#) ([cfloat32](#) *data, [uint32](#) size)
- void [mag](#) ([cfloat32](#) *orig, [float32](#) *newmags, [uint32](#) size)
- void [max](#) ([float32](#) *data, [uint32](#) size)
- void [smooth](#) ([float32](#) *data, [uint32](#) size, [uint16](#) order)
- void [play](#) (auto filename)
- void * [process](#) (void *procddata)
- void [StartProcessing](#) ([ProcData](#) procddata)

Variables

- const std::string [PATIENT_PATH](#) = "/home/pi/patients/"

4.1.1 Detailed Description

contains functions related to the file I/O use in this program

contains functions related to the program's threaded processing of audio data

contains the function(s) relating to sound

contains the functions necessary to perform the mathematical operations required by this program

This namespace contains all code related to this project.

Author

Samuel Andrew Wisner, awisner94@gmail.com
 Samuel Andrew Wisner, awisner94@gmail.com
 Nicholas K. Nolan

4.1.2 Enumeration Type Documentation

4.1.2.1 enum vaso::Side [strong]

The side of the head to which a recording pertains.

Enumerator

Left

Right

Definition at line 58 of file definitions.hpp.

4.1.3 Function Documentation

4.1.3.1 float32 vaso::average (float32 * data, uint32 size)

Takes the average of all elements in an array

Parameters

<i>data</i>	the array from which to compute the average
<i>size</i>	the number of elements in the data array

Returns

the computed average

Definition at line 105 of file sigmath.hpp.

4.1.3.2 void vaso::average (float32 * data, float32 * avg, uint8 count, uint32 size)

Element-wise averaging along the first dimension of a two-dimensional array.

Parameters

<i>data</i>	the two-dimensional array containing [count] number of arrays in the first dimension and [size] number of each elements in the second dimension
<i>avg</i>	the array of size [size] containing the averaged values of each element
<i>count</i>	the number of arrays in the first dimension of data and will likely be a constant value of 3 in this program
<i>size</i>	the number of elements in the second dimension of data

Definition at line 109 of file sigmath.hpp.

4.1.3.3 std::string vaso::CurrentDataName ()

Gets a data-based name to which the file(s) created in a session to be saved.

Returns

a partial (?) filename for the current session

Definition at line 26 of file fileio.hpp.

4.1.3.4 void vaso::diff (float32 * data, uint32 size)

Computes the left-handed first derivative of a discrete signal. The first element will be 0.

Parameters

<i>data</i>	an array containing the discrete signal data
<i>size</i>	the number of elements in data

Definition at line 113 of file sigmath.hpp.

4.1.3.5 void vaso::fft (cfloat32 * data, uint32 size)

Replaces the values of an array of cfloat32's with the array's DFT using a decimation-in-frequency algorithm.

This code is based on code from http://rosettacode.org/wiki/Fast_Fourier_transform#C.↵2B.2B.

Parameters

<i>data</i>	the array whose values should be replaced with its DFT
<i>size</i>	the number of elements in the data array

Definition at line 117 of file sigmath.hpp.

4.1.3.6 std::string vaso::InitialDataName (auto dir)

Finds the filename of the oldest (i.e., baseline) data is saved.

Parameters

<i>dir</i>	the directory which contains all patient data
------------	---

Returns

the base (?) filename to which all baseline data was saved

Definition at line 37 of file fileio.hpp.

4.1.3.7 void vaso::mag (cfloat32 * orig, float32 * newmags, uint32 size)

Computes the magnitude of an array of complex numbers.

Parameters

<i>orig</i>	the array of complex numbers
<i>newmags</i>	an array to which the magnitudes are to be written
<i>size</i>	the number of elements in orig and newmags

Definition at line 165 of file sigmath.hpp.

4.1.3.8 void vaso::max (float32 * data, uint32 size)

Finds the maximum value in an array.

Parameters

<i>data</i>	the array whose maximum value is to be found
<i>uint32</i>	size the number of elements in the data array

Definition at line 169 of file sigmath.hpp.

4.1.3.9 std::string vaso::PatientName ()

Prompts a user to enter a first, middle, and last name for a patients and creates a directory (if necessary) in which all of a patient's data can be saved.

Must warn a user if the patient folder does not already exist in order to prevent missaving data.

Returns

the directory under which all patient data is saved

Definition at line 51 of file fileio.hpp.

4.1.3.10 void vaso::play (auto filename)

Plays a WAVE file in a loop in a non-blocking manner.

Parameters

<i>filename</i>	the absolute or relative path to the WAVE file
-----------------	--

Definition at line 19 of file sound.hpp.

4.1.3.11 void * vaso::process (void * procddata)

Computes recording parameters in a separate thread in a thread-safe manner. Automatically waits for each recording to finish before processing it. This file is meant to be called ONLY from the [StartProcessing\(\)](#) function.

Parameters

<i>procddata</i>	a struct containing the values necessary to processing the audio
------------------	--

Returns

a (void) pointer to a [DataParams](#) struct containing the computed parameters for a patient

Definition at line 42 of file threadproc.hpp.

4.1.3.12 DataParams vaso::ReadParams (auto filename)

Reads the previously computed parameters found in the specified file.

Parameters

<i>filename</i>	the absolute or relative path to the file containing the patient data to read
-----------------	---

Returns

the patient parameters read

Definition at line 64 of file fileio.hpp.

4.1.3.13 void vaso::smooth (float32 * *data*, uint32 *size*, uint16 *order*)

Applies an nth-order moving-average filter to a discrete signal.

Parameters

<i>data</i>	the array containing the signal to which the filter should be applied
<i>size</i>	the number of elements in the data array
<i>order</i>	the order of the filter

Definition at line 173 of file sigmath.hpp.

4.1.3.14 void vaso::StartProcessing (ProcData *procddata*)

Begins processing the recorded data. Should be called before or immediately after the first recording.

Parameters

<i>proxdata</i>	a struct containing the values necessary to processing the audio
-----------------	--

Definition at line 46 of file threadproc.hpp.

4.1.3.15 std::string vaso::WriteParams (DataParams *params*, auto *filename*)

Writes the parameters to the specified file.

Parameters

<i>params</i>	
---------------	--

Definition at line 73 of file fileio.hpp.

4.1.4 Variable Documentation**4.1.4.1 const std::string vaso::PATIENT_PATH = "/home/pi/patients/"**

Absolute path to the folder containing the patients' data

Definition at line 18 of file fileio.hpp.

Chapter 5

Class Documentation

5.1 DataParams Struct Reference

```
#include <definitions.hpp>
```

5.1.1 Detailed Description

A structure containing the calculated results from processing the audio recordings.

Definition at line 40 of file definitions.hpp.

The documentation for this struct was generated from the following file:

- [src/definitions.hpp](#)

5.2 ProcData Struct Reference

```
#include <definitions.hpp>
```

5.2.1 Detailed Description

A structure containing information needed in the [process\(\)](#) thread.

Definition at line 47 of file definitions.hpp.

The documentation for this struct was generated from the following file:

- [src/definitions.hpp](#)

Chapter 6

File Documentation

6.1 bin/start File Reference

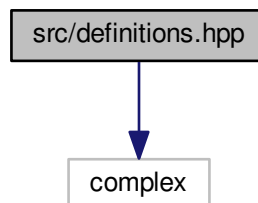
6.2 etc/doxygen.config File Reference

6.3 makefile File Reference

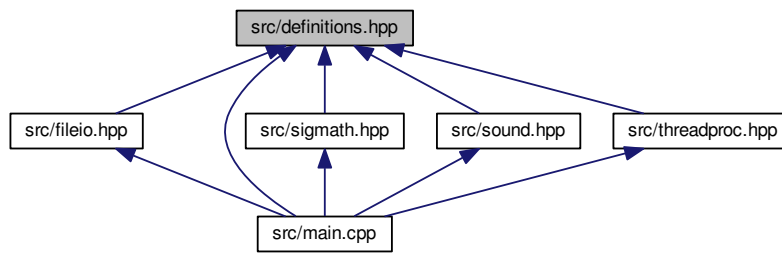
6.4 src/definitions.hpp File Reference

```
#include <complex>
```

Include dependency graph for definitions.hpp:



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



Classes

- struct [DataParams](#)
- struct [ProcData](#)

Namespaces

- [vaso](#)
contains functions related to the file I/O use in this program

Macros

- #define [ENUM](#) signed char
Contains declarations of system-independant (universal size) integers and float types, shortened type names for some commonly used types, and enumerations.

Typedefs

- typedef unsigned char [byte](#)
- typedef unsigned char [uint8](#)
- typedef signed char [sint8](#)
- typedef unsigned short [uint16](#)
- typedef signed short [sint16](#)
- typedef unsigned int [uint32](#)
- typedef signed int [sint32](#)
- typedef unsigned long long [uint64](#)
- typedef signed long long [sint64](#)
- typedef float [float32](#)
- typedef double [float64](#)
- typedef std::complex< [float32](#) > [cfloat32](#)

Enumerations

- enum [vaso::Side](#) { [vaso::Side::Left](#), [vaso::Side::Right](#) }

6.4.1 Macro Definition Documentation

6.4.1.1 `#define ENUM signed char`

Contains declarations of system-independant (universal size) integers and float types, shortened type names for some commonly used types, and enumerations.

Author

Samuel Andrew Wisner, awisner94@gmail.com

Definition at line 13 of file definitions.hpp.

6.4.2 Typedef Documentation

6.4.2.1 `typedef unsigned char byte`

Definition at line 15 of file definitions.hpp.

6.4.2.2 `typedef std::complex<float32> cfloat32`

Defines a type for complex float32's.

Definition at line 34 of file definitions.hpp.

6.4.2.3 `typedef float float32`

Definition at line 28 of file definitions.hpp.

6.4.2.4 `typedef double float64`

Definition at line 29 of file definitions.hpp.

6.4.2.5 `typedef signed short sint16`

Definition at line 20 of file definitions.hpp.

6.4.2.6 `typedef signed int sint32`

Definition at line 23 of file definitions.hpp.

6.4.2.7 `typedef signed long long sint64`

Definition at line 26 of file definitions.hpp.

6.4.2.8 `typedef signed char sint8`

Definition at line 17 of file definitions.hpp.

6.4.2.9 `typedef unsigned short uint16`

Definition at line 19 of file definitions.hpp.

6.4.2.10 typedef unsigned int uint32

Definition at line 22 of file definitions.hpp.

6.4.2.11 typedef unsigned long long uint64

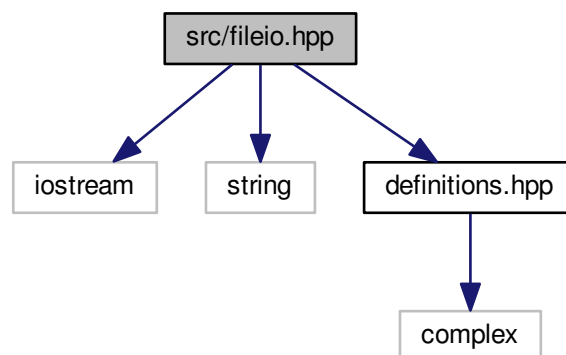
Definition at line 25 of file definitions.hpp.

6.4.2.12 typedef unsigned char uint8

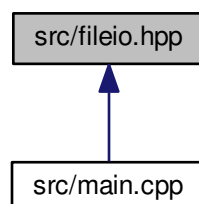
Definition at line 16 of file definitions.hpp.

6.5 src/fileio.hpp File Reference

```
#include <iostream>
#include <string>
#include "definitions.hpp"
Include dependency graph for fileio.hpp:
```



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



Namespaces

- [vaso](#)
contains functions related to the file I/O use in this program

Functions

- `std::string vaso::CurrentDataName ()`
- `std::string vaso::InitialDataName (auto dir)`
- `std::string vaso::PatientName ()`
- `DataParams vaso::ReadParams (auto filename)`
- `std::string vaso::WriteParams (DataParams params, auto filename)`

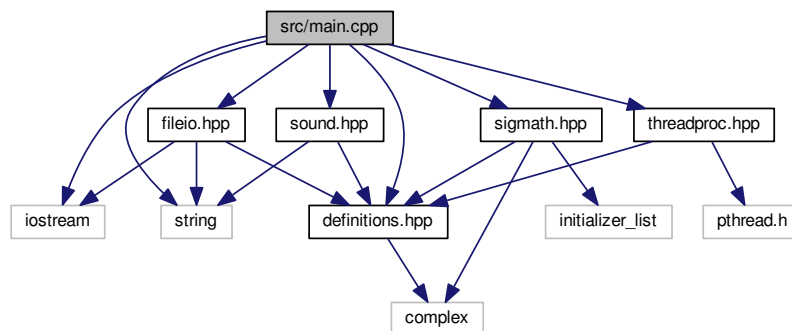
Variables

- `const std::string vaso::PATIENT_PATH = "/home/pi/patients/"`

6.6 src/main.cpp File Reference

```
#include <iostream>
#include <string>
#include "definitions.hpp"
#include "fileio.hpp"
#include "sigmath.hpp"
#include "sound.hpp"
#include "threadproc.hpp"
```

Include dependency graph for main.cpp:



Functions

- `int main (int argc, char **argv)`

6.6.1 Function Documentation

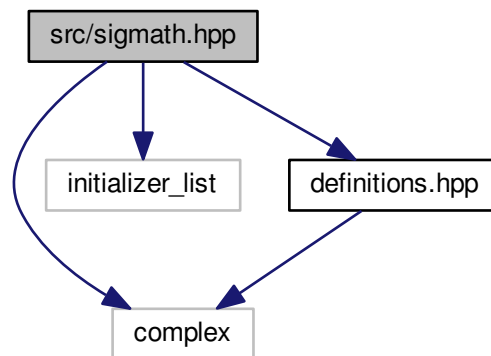
6.6.1.1 `int main (int argc, char ** argv)`

The main program for this project. It will detect vasospasms over a period of days.

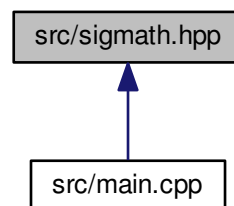
Definition at line 23 of file main.cpp.

6.7 src/sigmath.hpp File Reference

```
#include <complex>
#include <initializer_list>
#include "definitions.hpp"
Include dependency graph for sigmath.hpp:
```



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



Namespaces

- [vaso](#)
contains functions related to the file I/O use in this program

Functions

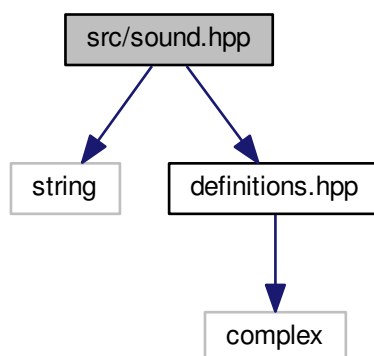
- [float32 vaso::average](#) ([float32](#) *data, [uint32](#) size)
- void [vaso::average](#) ([float32](#) *data, [float32](#) *avg, [uint8](#) count, [uint32](#) size)

- void `vaso::diff` (`float32` *data, `uint32` size)
- void `vaso::fft` (`cfloat32` *data, `uint32` size)
- void `vaso::mag` (`cfloat32` *orig, `float32` *newmags, `uint32` size)
- void `vaso::max` (`float32` *data, `uint32` size)
- void `vaso::smooth` (`float32` *data, `uint32` size, `uint16` order)

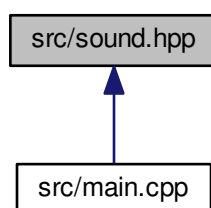
6.8 src/sound.hpp File Reference

```
#include <string>
#include "definitions.hpp"
```

Include dependency graph for sound.hpp:



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



Namespaces

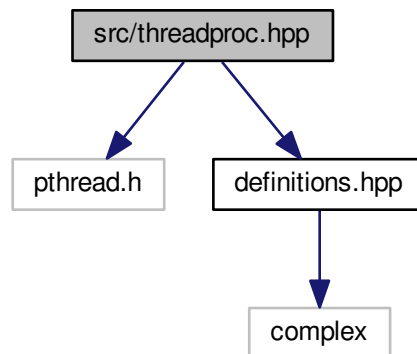
- `vaso`
contains functions related to the file I/O use in this program

Functions

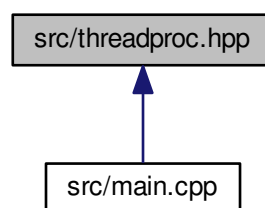
- void `vaso::play` (auto filename)

6.9 src/threadproc.hpp File Reference

```
#include <pthread.h>
#include "definitions.hpp"
Include dependency graph for threadproc.hpp:
```



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



Namespaces

- `vaso`
contains functions related to the file I/O use in this program

Functions

- void * `vaso::process` (void *procddata)

- void [vaso::StartProcessing](#) ([ProcData](#) procddata)

Index

- average
 - vaso, [8](#)
- bin/start, [15](#)
- diff
 - vaso, [9](#)
- fft
 - vaso, [9](#)
- Left
 - vaso, [8](#)
- mag
 - vaso, [9](#)
- makefile, [15](#)
- max
 - vaso, [9](#)
- play
 - vaso, [10](#)
- process
 - vaso, [10](#)
- Right
 - vaso, [8](#)
- Side
 - vaso, [8](#)
- smooth
 - vaso, [10](#)
- vaso, [7](#)
 - average, [8](#)
 - diff, [9](#)
 - fft, [9](#)
 - Left, [8](#)
 - mag, [9](#)
 - max, [9](#)
 - play, [10](#)
 - process, [10](#)
 - Right, [8](#)
 - Side, [8](#)
 - smooth, [10](#)