My Project

Generated by Doxygen 1.8.8

Sat Mar 5 2016 22:07:40

Contents

Chapter 1

Bug List

File alsa_test.cpp

clicking noise from sinusoidal discontinuity

File ZDomain.hpp

Everything. Just everything.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 radio Namespace Reference

Contains the classes for the various types of modulation supported by the program.

Classes

- class Filter
- class Modulator
- class Sinusoid
- class ZDomain

Enumerations

- enum Age { OLD, NEW }
- enum Fractional { NUM, DEN }
- enum ModulationType {
  ModulationType::DSB_LC, ModulationType::DSB_SC, ModulationType::USB_FILTERED, ModulationType←
  ::USB_HILBERT,
  ModulationType::LSB_FILTERED, ModulationType::LSB_HILBERT, ModulationType::FM_NARROW,
  ModulationType::FM_WIDE }

Functions

- void aconj (cfloat32 ∗data, uint32 size)
- void fft (cfloat32 ∗data, uint32 size)
- void hilbert (float32 ∗data, float32 ∗dest, uint32 size)
- void ifft (cfloat32 ∗data, uint32 size)
- void makeIQ (float32 ∗data, float32 ∗dest, uint32 size)

Variables

- fparams F_BASEBAND
- fparams F_LOWERSIDEBAND
- fparams F_UPPERSIDEBAND
- const uint32 FREQ_INTERMEDIATE = 20000
- const uint32 SAMPLING_RATE = 48000

5.1.1 Detailed Description

Contains the classes for the various types of modulation supported by the program.

contains the Sinusoid class

This namespace contains all the classes, functions, and enumerations used in the application.

Author

Samuel Andrew Wisner, `awisner94@gmail.com`

5.1.2 Enumeration Type Documentation

5.1.2.1 enum radio::Age

Describes the age of a filter (from last Pass() or in this Pass())

Enumerator

OLD

NEW

Definition at line 50 of file definitions.hpp.

5.1.2.2 enum radio::Fractional

Describes the numerator and denominator of a z-domain transfer function

Enumerator

NUM

DEN

Definition at line 55 of file definitions.hpp.

5.1.2.3 enum radio::ModulationType `[strong]`

Describes a form of modulation.

Enumerator

DSB_LC

DSB_SC

USB_FILTERED

USB_HILBERT

LSB_FILTERED

LSB_HILBERT

FM_NARROW

FM_WIDE

Definition at line 60 of file definitions.hpp.

5.1.3 Function Documentation

5.1.3.1 void radio::aconj (**cfloat32** ∗ *data,* **uint32** *size*)

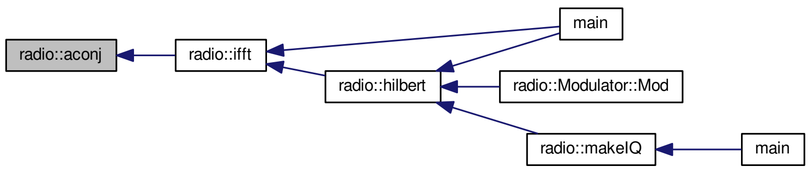
Replaces the values in an array of complex float32's with their respective conjugates.

Parameters

data	the array whose values should be replaced with their respective conjugates
size	the number of elements in the data array

Definition at line 84 of file zdomain.hpp.

Here is the caller graph for this function:



5.1.3.2 void radio::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.

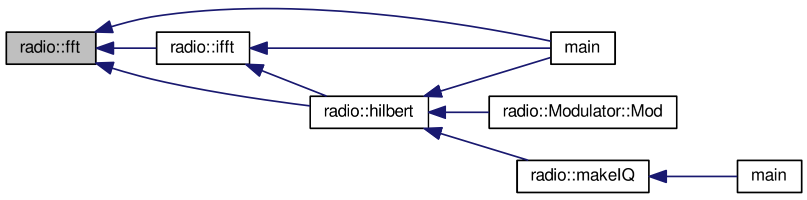
This code is based on code from http://rosettacode.org/wiki/Fast_Fourier_transform#C.↩
2B.2B.

Parameters

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

Definition at line 90 of file zdomain.hpp.

Here is the caller graph for this function:



5.1.3.3 void radio::hilbert (float32 ∗ *data*, float32 ∗ *dest*, uint32 *size*)

Performs the hilbert transfor of an array of float32's.

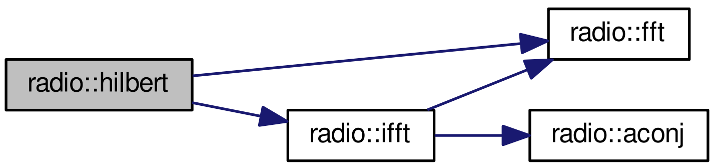
Parameters

data	the source array of the REAL numbers of which to take the Hilbert transform

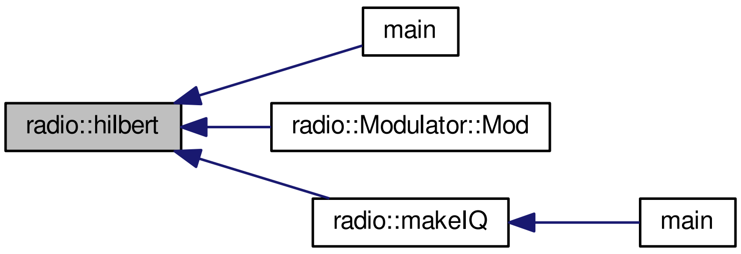
dest	the destination array of REAL numbers for the results of the Hilbert transform
size	the number of elements in the data and dest arrays

Definition at line 138 of file zdomain.hpp.

Here is the call graph for this function:



Here is the caller graph for this function:



5.1.3.4 void radio::ifft (cfloat32 ∗ *data*, uint32 *size*)

Replaces the values of an array of cfloat32's with the array's inverse DFT.

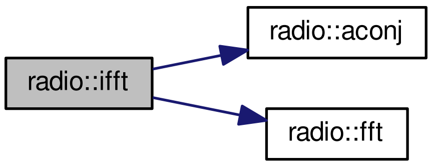
This code is based on code from `http://rosettacode.org/wiki/Fast_Fourier_transform#C.↩ 2B.2B`.

Parameters

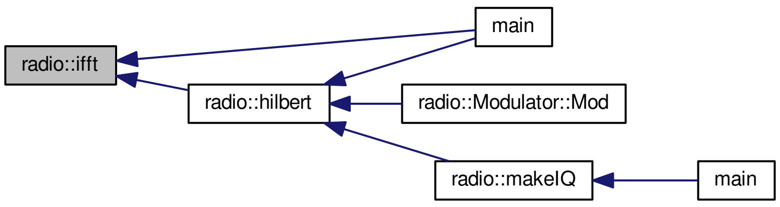
data	the array whose values should be replaced with its inverse DFT
size	the number of elements in the data array

Definition at line 158 of file zdomain.hpp.

Here is the call graph for this function:



Here is the caller graph for this function:



5.1.3.5 void radio::makeIQ (**float32** ∗ *data,* **float32** ∗ *dest,* **uint32** *size*)

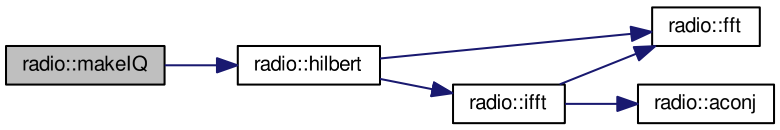
Produces an interleaved array of first an element from an original array of data and then an element from the original data's Hilbert transform. This function is intended to generate a two-channel output (I/Q output) for mixing applications.

Parameters

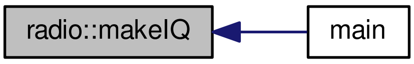
data	the original data (left channel)
dest	the interleaved data (left channel original data, right channel transformed data) twice the size of the original data array
size	the number of elements in the data array (NOT in the destination array)

Definition at line 168 of file zdomain.hpp.

Here is the call graph for this function:



Here is the caller graph for this function:



5.1.4 Variable Documentation

5.1.4.1 fparams radio::F_BASEBAND

Initial value:

```
= { std::vector<float32> {
        0.0008977019461,
            -0.002215694636,
            0.001372192986,
            0.001372192986,
            -0.002215694636,
            0.0008977019461
    }, std::vector<float32> {
        1,
            -4.678616047,
            8.822912216,
            -8.379911423,
            4.007629871,
            -0.7719064355
    } }
```

Baseband filter coefficients. Generated with MATLAB 2015A.

Definition at line 19 of file fvectors.hpp.

5.1.4.2 fparams radio::F_LOWERSIDEBAND

Initial value:

```
= { std::vector<float32> {
        0.2758038938,
            2.763578892,
            12.83915043,
            36.47584915,
            70.37084961,
            96.76893616,
            96.76893616,
            70.37084961,
            36.47584915,
            12.83915043,
            2.763578892,
            0.2758038938
    }, std::vector<float32> {
        1,
            7.605497837,
            27.34180641,
            60.83375549,
            92.60908508,
            100.8363876,
            79.74796295,
            45.49822617,
            18.1356678,
            4.690036297,
            0.6617552638,
            0.0281427335
    } }
```

Lower-sideband filter coefficients. Generated with MATLAB 2015A.

Definition at line 38 of file fvectors.hpp.

5.1.4.3 fparams radio::F_UPPERSIDEBAND

Initial value:

```
= { std::vector<float32> {
        0.001690387726,
            0.01145271584,
            0.03591799363,
            0.06576926261,
            0.0711934343,
            0.03156377375,
            -0.03156377375,
            -0.0711934343,
            -0.06576926261,
            -0.03591799363,
            -0.01145271584,
            -0.001690387726
    }, std::vector<float32> {
        1,
            9.465174675,
            41.62402725,
            112.0970993,
            205.2097626,
            267.9378662,
            254.4868011,
            175.7772827,
            86.5161972,
            28.89988136,
            5.897814751,
            0.5572910309
    } }
```

Upper-sideband filter coefficients. Generated with MATLAB 2015A.

Definition at line 69 of file fvectors.hpp.

5.1.4.4 const uint32 radio::FREQ_INTERMEDIATE = 20000

The default intermediate carrier frequency

Definition at line 24 of file modulation.hpp.

5.1.4.5 const uint32 radio::SAMPLING_RATE = 48000

The default sampling rate (frequency)

Definition at line 29 of file modulation.hpp.

Chapter 6

Class Documentation

6.1 radio::Filter Class Reference

```
#include <Filter.hpp>
```

Public Member Functions

- Filter (float32 ∗data, uint32 size, fparams &diffEq)
- void Pass ()

Protected Attributes

- uint8 eqLength
- uint32 size
- float32 ∗ data
- fparams diffEq
- fparams prev

6.1.1 Detailed Description

This class implements a z-domain filter on a specified array of float32"'s (a.k.a. singles, floats). It requires the transfer function coefficients already be calculated (i.e., it does not generate the coefficients based on desired filter characteristics). MATLAB and its Signal Processing Toolbox can be used to generate the coefficients.

While this class is designed to implement a single-section filter, several instances of the class can be created and run over the data array sequentially to effectively implement a multi-section filter.

The class is designed (but not tested!) to allow for a z-domain transfer function with different orders of the zeros (numerator) and poles (denominator).

Definition at line 31 of file Filter.hpp.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 radio::Filter::Filter (**float32** ∗ *data*, **uint32** *size*, **fparams** & *diffEq*)

Initializes Filter based on a difference equation.

Parameters

data	array to be filtered. The filtered data will be placed here.
size	number of elements in the data array
diffEq	a vector containing two vectors of float32"'s (a.k.a. singles, floats), containing the numerator and denominator coefficients, respectively, of the z-domain tranfer function of the filter in decending order (z^0, z^-1, z^-2, etc.).

Definition at line 91 of file Filter.hpp.

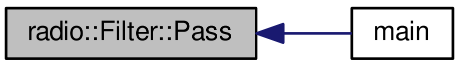
6.1.3 Member Function Documentation

6.1.3.1 void radio::Filter::Pass ( )

Passes the data array through the digital filter and accounts for x[n] and y[n] values from the previous call to Pass().

Definition at line 111 of file Filter.hpp.

Here is the caller graph for this function:



6.1.4 Member Data Documentation

6.1.4.1 float32∗ radio::Filter::data [protected]

A pointer to the data array that should be filtered when Pass() is called.

Definition at line 71 of file Filter.hpp.

6.1.4.2 fparams radio::Filter::diffEq [protected]

A vector containing two vectors of float32"'s (a.k.a. singles, floats), containing the numerator and denominator coefficients, respectively, of the z-domain tranfer function of the filter in decending order (z^0, z^-1, z^-2, etc.).

Definition at line 79 of file Filter.hpp.

6.1.4.3 uint8 radio::Filter::eqLength [protected]

The order of the filter transfer function (i.e., the maximum of the orders of the numerator and denominator).

Definition at line 60 of file Filter.hpp.

6.1.4.4 fparams radio::Filter::prev [protected]

Vectors of the original (x[n]) and filtered (y[n]) values of the data array used to calculate the first filtered values of the data array. In spite of the type name, this variable does NOT contains filter parameters but rather the same data type that fparams represents.

Definition at line 88 of file Filter.hpp.

6.1.4.5 uint32 radio::Filter::size `[protected]`

The number of elements in the data array.

Definition at line 65 of file Filter.hpp.

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

- src/Filter.hpp

6.2 radio::Modulator Class Reference

```
#include <modulation.hpp>
```

Public Member Functions

- void Mod ()

6.2.1 Detailed Description

This class, while not intended to be called directly, is a superclass for the classes of the modulation forms used in this project.

Definition at line 35 of file modulation.hpp.

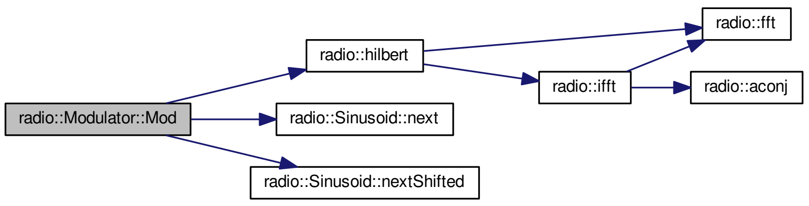
6.2.2 Member Function Documentation

6.2.2.1 void radio::Modulator::Mod ( )

Modulates the audio currently in the data array.

Definition at line 96 of file modulation.hpp.

Here is the call graph for this function:



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

- src/modulation.hpp

6.3 radio::Sinusoid Class Reference

```
#include <Sinusoid.hpp>
```

Public Member Functions

- Sinusoid (float32 frequency, uint32 samplingRate)
- ∼Sinusoid ()
- float32 next ()
- float32 nextShifted ()

6.3.1 Detailed Description

This class creates an easy-to-call sinusoid that will preserve its phase throughout its lifespan. Essentially, it is a ring buffer.

Definition at line 19 of file Sinusoid.hpp.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 radio::Sinusoid::Sinusoid (float32 *frequency,* uint32 *samplingRate*)

Creates a ring-buffer sinusoid.

Definition at line 71 of file Sinusoid.hpp.

6.3.2.2 radio::Sinusoid::∼Sinusoid ( )

Free arrays malloc'd in the constructor.

Definition at line 86 of file Sinusoid.hpp.

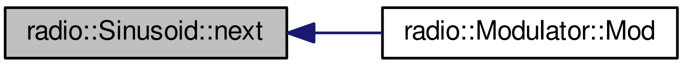
6.3.3 Member Function Documentation

6.3.3.1 float32 radio::Sinusoid::next ( )

Provides the next value of the sinusoid in a manner consistant with a ring buffer.

Definition at line 91 of file Sinusoid.hpp.

Here is the caller graph for this function:

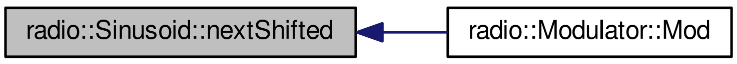


6.3.3.2 float32 radio::Sinusoid::nextShifted ( )

Provides the next value of the sinusoid shifted 90 degrees in a manner consistant with a ring buffer.

Definition at line 96 of file Sinusoid.hpp.

Here is the caller graph for this function:



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

- src/Sinusoid.hpp

6.4 radio::ZDomain Class Reference

```
#include <ZDomain.hpp>
```

Public Member Functions

- ZDomain (float32 ∗orig, float32 ∗dest, uint16 size)
- ∼ZDomain ()
- void fft ()
- void hilbert ()
- void ifft ()

Public Attributes

- uint16 size
- float32 ∗ orig
- std::complex< float32 > ∗ temp

Protected Member Functions

- void _fft ()
- void _ifft ()

6.4.1 Detailed Description

Definition at line 23 of file ZDomain.hpp.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 radio::ZDomain::ZDomain (**float32** ∗ *orig,* **float32** ∗ *dest,* **uint16** *size*)

Definition at line 78 of file ZDomain.hpp.

6.4.2.2 radio::ZDomain::∼ZDomain ( )

Definition at line 89 of file ZDomain.hpp.

6.4.3 Member Function Documentation

6.4.3.1 void radio::ZDomain::_fft ( ) [protected]

Definition at line 93 of file ZDomain.hpp.

6.4.3.2 void radio::ZDomain::_ifft ( ) [protected]

Definition at line 141 of file ZDomain.hpp.

6.4.3.3 void radio::ZDomain::fft ( )

6.4.3.4 void radio::ZDomain::hilbert ( )

Definition at line 149 of file ZDomain.hpp.

6.4.3.5 void radio::ZDomain::ifft ( )

6.4.4 Member Data Documentation

6.4.4.1 float32∗ radio::ZDomain::orig

Definition at line 34 of file ZDomain.hpp.

6.4.4.2 uint16 radio::ZDomain::size

Definition at line 29 of file ZDomain.hpp.

6.4.4.3 std::complex<float32>∗ radio::ZDomain::temp

Definition at line 39 of file ZDomain.hpp.

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

- src/ZDomain.hpp

Chapter 7

File Documentation

7.1 etc/doxygen.config File Reference

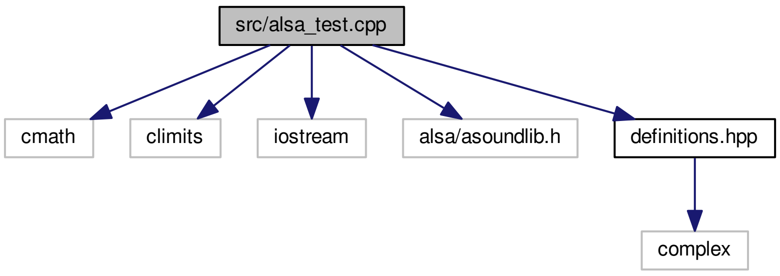
7.2 makefile File Reference

7.3 src/alsa_test.cpp File Reference

Tests sinusoidal tone generation.

```
#include <cmath>
#include <climits>
#include <iostream>
#include <alsa/asoundlib.h>
#include "definitions.hpp"
```

Include dependency graph for alsa_test.cpp:



Functions

- int main ()

7.3.1 Detailed Description

Tests sinusoidal tone generation.

Author

Samuel Andrew Wisner, awisner94@gmail.com

Bug clicking noise from sinusoidal discontinuity

Definition in file alsa_test.cpp.

7.3.2 Function Documentation

7.3.2.1 int main ( )

This program tests sinusoidal speaker output through the ALSA API. Not sure if it works. When it did at least compile and run, it produced a sinusoid with an approximately twice-per-second clicking noise.

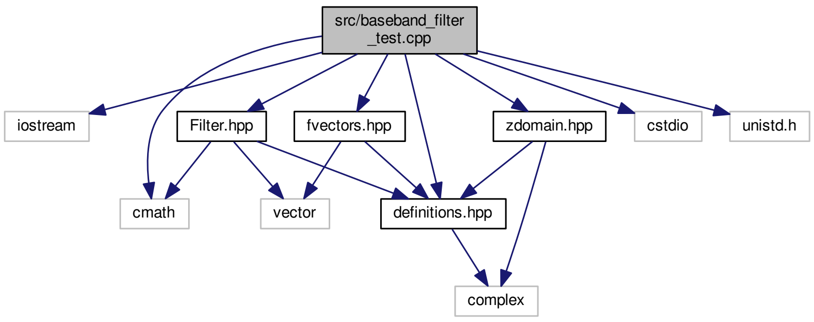
Definition at line 22 of file alsa_test.cpp.

7.4 src/baseband_filter_test.cpp File Reference

Tests sinusoidal tone generation.

```
#include <iostream>
#include <cmath>
#include <cstdio>
#include <unistd.h>
#include "definitions.hpp"
#include "Filter.hpp"
#include "fvectors.hpp"
#include "zdomain.hpp"
```

Include dependency graph for baseband_filter_test.cpp:



Functions

- int main ()

7.4.1 Detailed Description

Tests sinusoidal tone generation.

Author

Samuel Andrew Wisner, awisner94@gmail.com

Definition in file baseband_filter_test.cpp.

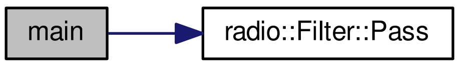
7.4.2 Function Documentation

7.4.2.1 int main ( )

This prgram tests and demonstrates the Filter class and the baseband low-pass filter (fp = 1.7 kHz, fs = 3 kHz, Ap = 0.5 dB, As = 60 dB).

Definition at line 24 of file baseband_filter_test.cpp.

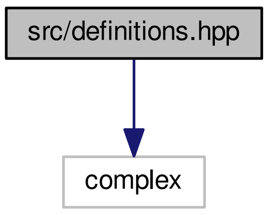
Here is the call graph for this function:



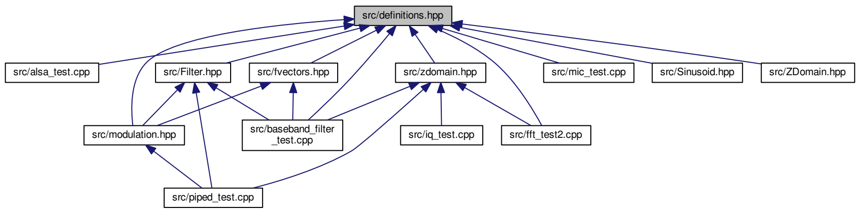
7.5 src/definitions.hpp File Reference

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

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



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



Namespaces

- radio

  Contains the classes for the various types of modulation supported by the program.

Macros

- #define ENUM signed char

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
- typedef std::vector
  < std::vector< float32 > > fparams

Enumerations

- enum radio::Age { radio::OLD, radio::NEW }
- enum radio::Fractional { radio::NUM, radio::DEN }
- enum radio::ModulationType {
  radio::ModulationType::DSB_LC, radio::ModulationType::DSB_SC, radio::ModulationType::USB_FILTERED, radio::ModulationType::USB_HILBERT,
  radio::ModulationType::LSB_FILTERED, radio::ModulationType::LSB_HILBERT, radio::ModulationType::F←
  M_NARROW, radio::ModulationType::FM_WIDE }

7.5.1 Detailed Description

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 in file definitions.hpp.

7.5.2 Macro Definition Documentation

7.5.2.1 #define ENUM signed char

Definition at line 14 of file definitions.hpp.

7.5.3 Typedef Documentation

7.5.3.1 typedef unsigned char **byte**

Definition at line 16 of file definitions.hpp.

7.5.3.2 typedef std::complex<**float32**> **cfloat32**

Defines a type for complex float32's.

Definition at line 35 of file definitions.hpp.

7.5.3.3 typedef float **float32**

Definition at line 29 of file definitions.hpp.

7.5.3.4 typedef double **float64**

Definition at line 30 of file definitions.hpp.

7.5.3.5 typedef std::vector<std::vector<**float32**> > **fparams**

Defines a type for the filter coefficients.

Definition at line 40 of file definitions.hpp.

7.5.3.6 typedef signed short **sint16**

Definition at line 21 of file definitions.hpp.

7.5.3.7 typedef signed int **sint32**

Definition at line 24 of file definitions.hpp.

7.5.3.8 typedef signed long long **sint64**

Definition at line 27 of file definitions.hpp.

7.5.3.9 typedef signed char **sint8**

Definition at line 18 of file definitions.hpp.

7.5.3.10 typedef unsigned short **uint16**

Definition at line 20 of file definitions.hpp.

7.5.3.11 typedef unsigned int **uint32**

Definition at line 23 of file definitions.hpp.

7.5.3.12 typedef unsigned long long **uint64**

Definition at line 26 of file definitions.hpp.

7.5.3.13 typedef unsigned char **uint8**

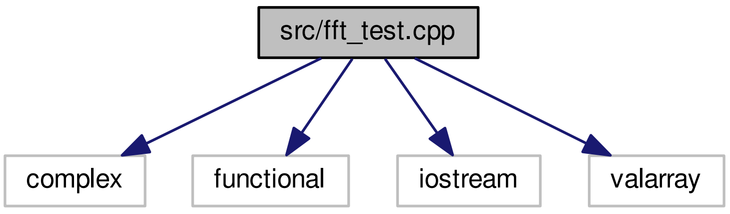
Definition at line 17 of file definitions.hpp.

7.6 src/fft_test.cpp File Reference

Tests FFT, IFFT, and Hilbert implementations.

```
#include <complex>
#include <functional>
#include <iostream>
#include <valarray>
```

Include dependency graph for fft_test.cpp:



Typedefs

- typedef std::valarray
  $<$ std::complex$<$ double $>$ $>$ CArray

Functions

- void fft (CArray &x)
- void ifft (CArray &x)
- std::complex< double > hilbert (std::complex< double > n)
- int main ()

Variables

- const double PI = 3.141592653589793238460

7.6.1 Detailed Description

Tests FFT, IFFT, and Hilbert implementations.

Author

Samuel Andrew Wisner, `awisner94@gmail.com`

Definition in file fft_test.cpp.

7.6.2 Typedef Documentation

7.6.2.1 typedef std::valarray<std::complex<double> > CArray

Definition at line 14 of file fft_test.cpp.

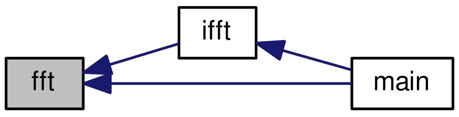
7.6.3 Function Documentation

7.6.3.1 void fft (CArray & *x*)

This code was taken from `http://rosettacode.org/wiki/Fast_Fourier_transform#C.2B.2B`.

Definition at line 23 of file fft_test.cpp.

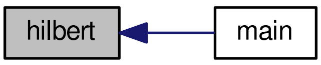
Here is the caller graph for this function:



7.6.3.2 std::complex<double> hilbert (std::complex< double > *n*)

Definition at line 87 of file fft_test.cpp.

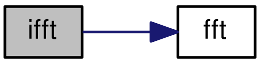
Here is the caller graph for this function:



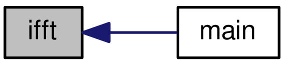
7.6.3.3 void ifft (CArray & *x*)

Definition at line 72 of file fft_test.cpp.

Here is the call graph for this function:



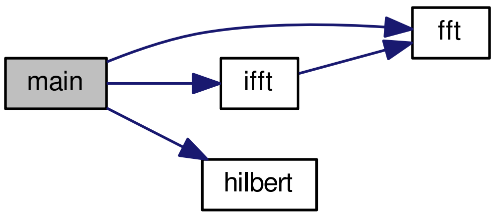
Here is the caller graph for this function:



7.6.3.4 int main ( )

Definition at line 91 of file fft_test.cpp.

Here is the call graph for this function:



7.6.4 Variable Documentation

7.6.4.1 const double PI = 3.141592653589793238460

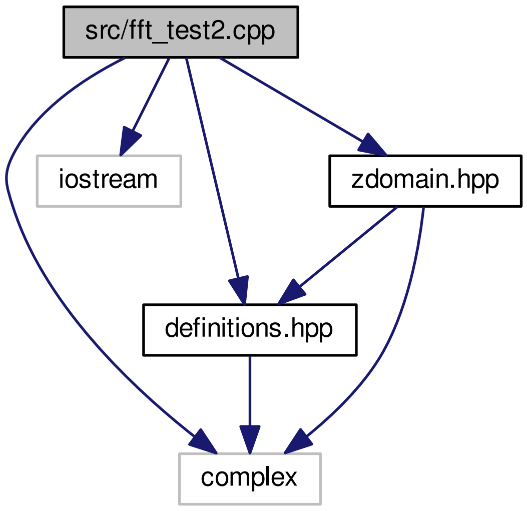
Definition at line 12 of file fft_test.cpp.

7.7 src/fft_test2.cpp File Reference

Tests FFT, IFFT, and Hilbert implementations in zdomain.hpp.

```
#include <complex>
#include <iostream>
#include "definitions.hpp"
#include "zdomain.hpp"
```

Include dependency graph for fft_test2.cpp:



Functions

- int main ()

7.7.1 Detailed Description

Tests FFT, IFFT, and Hilbert implementations in zdomain.hpp.

Author

Samuel Andrew Wisner, awisner94@gmail.com

Definition in file fft_test2.cpp.

7.7.2 Function Documentation

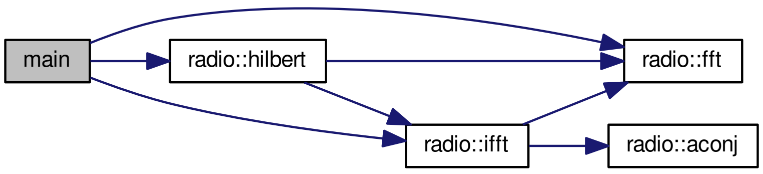
7.7.2.1 int main ( )

This program tests the fft(), ifft(), and hilbert() functions in the zdomain.hpp file.

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

Definition at line 22 of file fft_test2.cpp.

Here is the call graph for this function:

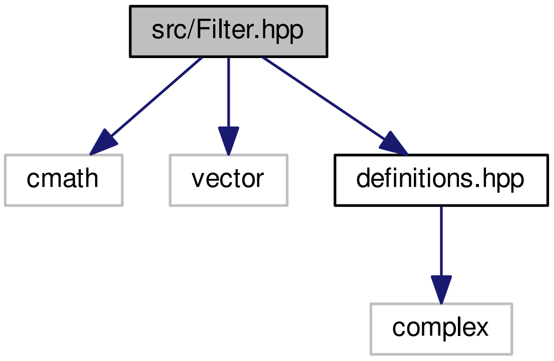


7.8 src/Filter.hpp File Reference

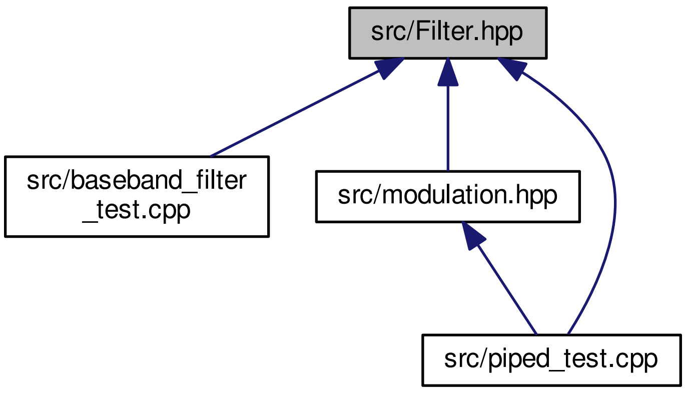
Defines the Filter class.

```
#include <cmath>
#include <vector>
#include "definitions.hpp"
```

Include dependency graph for Filter.hpp:



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



Classes

- class radio::Filter

Namespaces

- radio

  Contains the classes for the various types of modulation supported by the program.

7.8.1 Detailed Description

Defines the Filter class.

Author

Samuel Andrew Wisner, awisner94@gmail.com

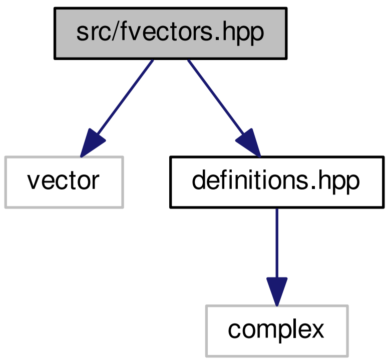
Definition in file Filter.hpp.

7.9 src/fvectors.hpp File Reference

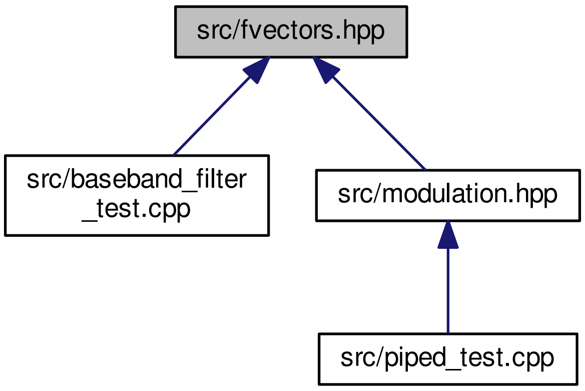
Defines the transfer function coefficients used in the instances of the Filter class in this program.

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

Include dependency graph for fvectors.hpp:



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



Namespaces

- radio

  Contains the classes for the various types of modulation supported by the program.

Variables

- fparams radio::F_BASEBAND
- fparams radio::F_LOWERSIDEBAND
- fparams radio::F_UPPERSIDEBAND

7.9.1 Detailed Description

Defines the transfer function coefficients used in the instances of the Filter class in this program.

Author

Samuel Andrew Wisner, `awisner94@gmail.com`

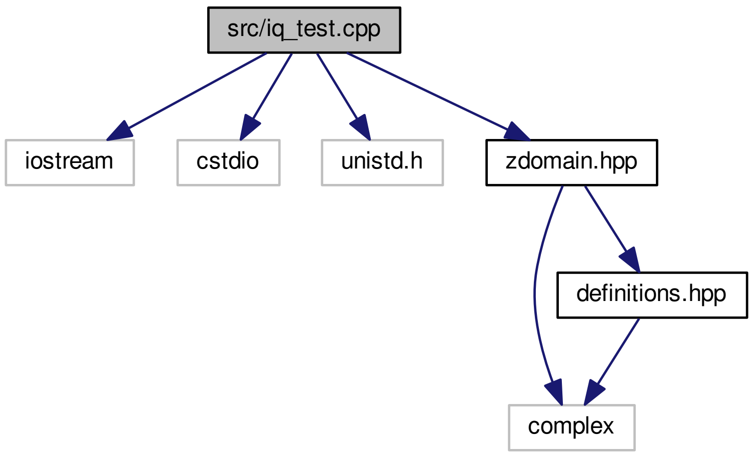
Definition in file fvectors.hpp.

7.10 src/iq_test.cpp File Reference

Generates test IQ signal.

```
#include <iostream>
#include <cstdio>
#include <unistd.h>
#include "zdomain.hpp"
```

Include dependency graph for iq_test.cpp:



Functions

- int main ()

7.10.1 Detailed Description

Generates test IQ signal.

Author

Samuel Andrew Wisner, `awisner94@gmail.com`

Definition in file iq_test.cpp.

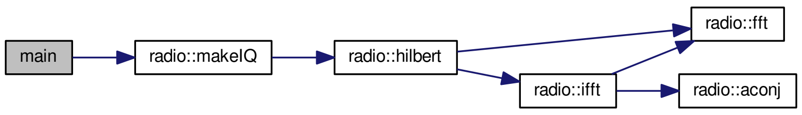
7.10.2 Function Documentation

7.10.2.1 int main ( )

This small program demonstrates the IQ generation abilities of the makeIQ() function.

Definition at line 20 of file iq_test.cpp.

Here is the call graph for this function:

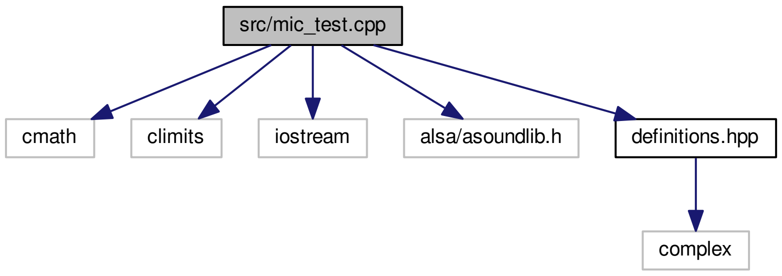


7.11 src/mic_test.cpp File Reference

Tests getting mic input via ALSA May not even compile at the moment.

```
#include <cmath>
#include <climits>
#include <iostream>
#include <alsa/asoundlib.h>
#include "definitions.hpp"
```

Include dependency graph for mic_test.cpp:



Functions

- int main ()

7.11.1 Detailed Description

Tests getting mic input via ALSA May not even compile at the moment.

Author

Samuel Andrew Wisner, awisner94@gmail.com

Definition in file mic_test.cpp.

7.11.2 Function Documentation

7.11.2.1 int main ( )

This program tests taking information from the microphone via the ALSA API. Not sure if it works.

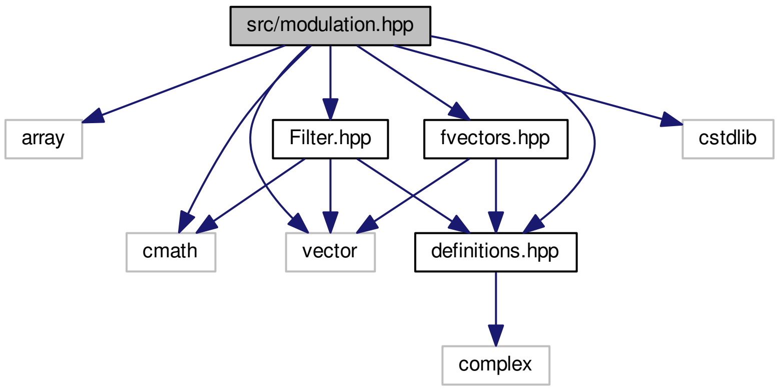
Definition at line 21 of file mic_test.cpp.

7.12 src/modulation.hpp File Reference

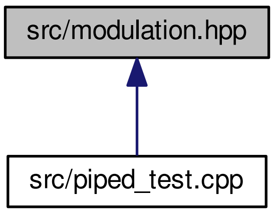
```
#include <array>
```

```
#include <cmath>
#include <cstdlib>
#include <vector>
#include "definitions.hpp"
#include "Filter.hpp"
#include "fvectors.hpp"
```

Include dependency graph for modulation.hpp:



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



Classes

- class radio::Modulator

Namespaces

- radio

  Contains the classes for the various types of modulation supported by the program.

Variables

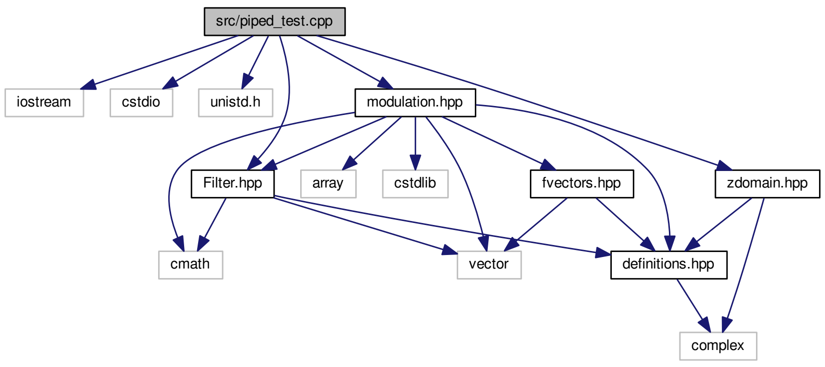
- const uint32 radio::FREQ_INTERMEDIATE = 20000

- const uint32 radio::SAMPLING_RATE = 48000

7.13 src/piped_test.cpp File Reference

```
#include <iostream>
#include <cstdio>
#include <unistd.h>
#include "Filter.hpp"
#include "modulation.hpp"
#include "zdomain.hpp"
```

Include dependency graph for piped_test.cpp:



Functions

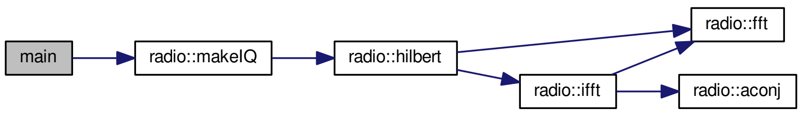
- int main ()

7.13.1 Function Documentation

7.13.1.1 int main ( )

Definition at line 12 of file piped_test.cpp.

Here is the call graph for this function:

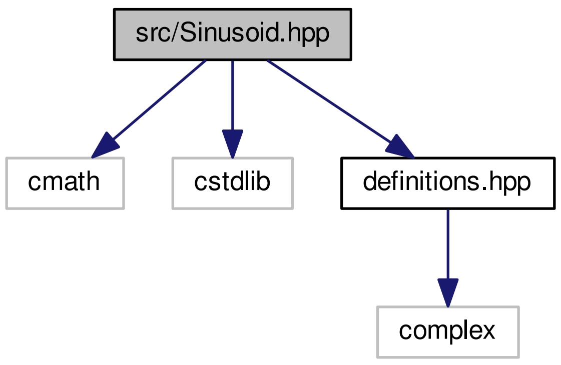


7.14 src/Sinusoid.hpp File Reference

```
#include <cmath>
```

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

Include dependency graph for Sinusoid.hpp:



Classes

- class radio::Sinusoid

Namespaces

- radio

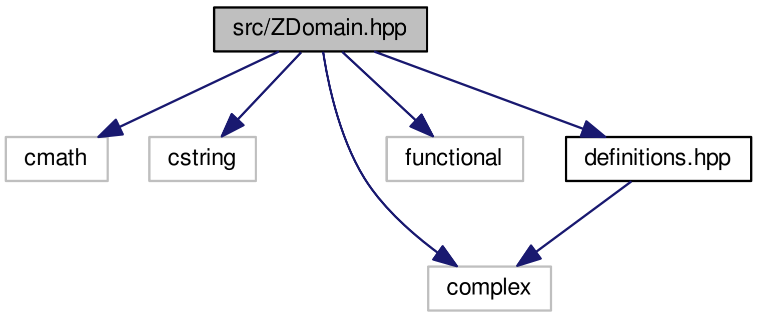
  Contains the classes for the various types of modulation supported by the program.

7.15 src/ZDomain.hpp File Reference

This is an abandoned, overly complicated attempt to have a class housing z-domain functions.

```
#include <cmath>
#include <cstring>
#include <complex>
#include <functional>
#include "definitions.hpp"
```

Include dependency graph for ZDomain.hpp:



Classes

- class radio::ZDomain

Namespaces

- radio

  Contains the classes for the various types of modulation supported by the program.

7.15.1 Detailed Description

This is an abandoned, overly complicated attempt to have a class housing z-domain functions.

Author

Samuel Andrew Wisner, `awisner94@gmail.com`

Bug Everything. Just everything.

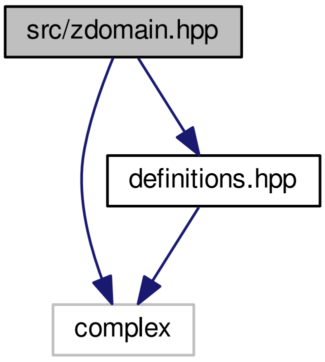
Definition in file ZDomain.hpp.

7.16 src/zdomain.hpp File Reference

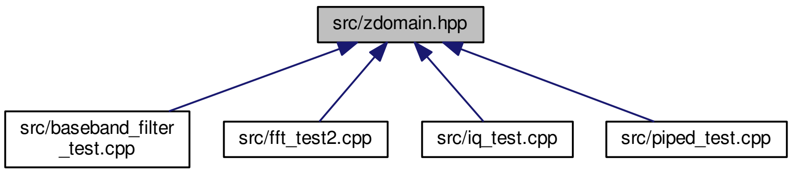
Contains the functions to manipulate sequential data in the frequency (z) domain.

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

Include dependency graph for zdomain.hpp:



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



Namespaces

- radio

    Contains the classes for the various types of modulation supported by the program.

Functions

- void radio::aconj (cfloat32 ∗data, uint32 size)
- void radio::fft (cfloat32 ∗data, uint32 size)
- void radio::hilbert (float32 ∗data, float32 ∗dest, uint32 size)
- void radio::ifft (cfloat32 ∗data, uint32 size)
- void radio::makeIQ (float32 ∗data, float32 ∗dest, uint32 size)

7.16.1 Detailed Description

Contains the functions to manipulate sequential data in the frequency (z) domain.

Author

Samuel Andrew Wisner, awisner94@gmail.com

Definition in file zdomain.hpp.

Index