

Filtro FIR en C

1.0

Generado por Doxygen 1.7.4

Miércoles, 2 de Noviembre de 2011 11:59:29

Índice general

1	Índice de archivos	1
1.1	Lista de archivos	1
2	Documentación de archivos	3
2.1	Referencia del Archivo fir.c	3
2.1.1	Descripción detallada	4
2.1.2	Documentación de las funciones	4
2.1.2.1	fir	4
2.1.2.2	ini_fir	4
2.2	Referencia del Archivo fir.h	4
2.2.1	Descripción detallada	5
2.2.2	Documentación de las funciones	6
2.2.2.1	fir	6
2.2.2.2	ini_fir	6
2.3	Referencia del Archivo genera_senales.c	6
2.3.1	Descripción detallada	7
2.3.2	Documentación de las funciones	7
2.3.2.1	impulso	7
2.3.2.2	step	8
2.4	Referencia del Archivo genera_senales.h	8
2.4.1	Descripción detallada	9
2.4.2	Documentación de las funciones	9
2.4.2.1	impulso	9
2.4.2.2	step	10
2.5	Referencia del Archivo testfir.c	10
2.5.1	Descripción detallada	11

2.5.2	Documentación de las funciones	11
2.5.2.1	main	11

Capítulo 1

Indice de archivos

1.1. Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

fir.c (Implementa las funciones de inicialización del filtro y la función de filtrado)	3
fir.h (Implementa las funciones de inicialización del filtro y la función de filtrado)	4
genera_senales.c (Genera señales sintéticas para la prueba de la implementación del filtro FIR. Genera señales tipo escalón e impulso)	6
genera_senales.h (Genera señales sintéticas para la prueba de la implementación del filtro FIR. Genera señales tipo escalón e impulso)	8
testfir.c (Programa para probar el filtro FIR implementado)	10

Capítulo 2

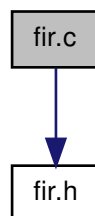
Documentación de archivos

2.1. Referencia del Archivo fir.c

Implementa las funciones de inicialización del filtro y la función de filtrado.

```
#include "fir.h"
```

Dependencia gráfica adjunta para fir.c:



Funciones

- void `ini_fir` (`sample_t` coefs[])
Inicializa el filtro.
- `sample_t` `fir` (`sample_t` *muestra)
Aplica filtrado FIR a la muestra de entrada.

Variables

- `sample_t c [TAP_LENGTH]`
Coeficientes del filtro.
- `int taps = 0`
Contador de taps disponibles para la inicialización.

2.1.1. Descripción detallada

Implementa las funciones de inicialización del filtro y la función de filtrado.

2.1.2. Documentación de las funciones

2.1.2.1. `fir (sample_t * muestra)`

Aplica filtrado FIR a la muestra de entrada.

Parámetros

<code>muestra</code>	Puntero a la muestra de la señal de entrada (tipo <code>sample_t</code>)
----------------------	---

Devuelve

Devuelve la muestra de la señal de salida del filtro.

2.1.2.2. `ini_fir (sample_t coefs[])`

Inicializa el filtro.

Parámetros

<code>coefs[]</code>	Coeficientes del filtro (array de elementos tipos <code>sample_t</code>)
----------------------	---

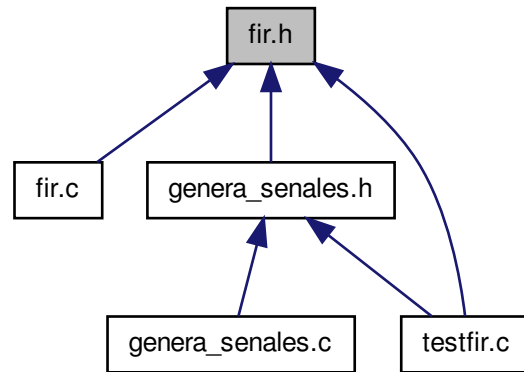
Devuelve

No tiene salida.

2.2. Referencia del Archivo `fir.h`

Implementa las funciones de inicialización del filtro y la función de filtrado.

Gráfico de los archivos que directa o indirectamente incluyen a este archivo:



'defines'

- `#define TAP_LENGTH 5`
Orden del filtro.
- `#define SIGNAL_LENGTH 15`
Cantidad de muestras.

'typedefs'

- `typedef float sample_t`
Tipo definido para manejar muestras de la señal.

Funciones

- `void ini_fir (sample_t coefs[])`
Inicializa el filtro.
- `sample_t fir (sample_t *muestra)`
Aplica filtrado FIR a la muestra de entrada.

2.2.1. Descripción detallada

Implementa las funciones de inicialización del filtro y la función de filtrado.

2.2.2. Documentación de las funciones

2.2.2.1. `sample_t fir (sample_t * muestra)`

Aplica filtrado FIR a la muestra de entrada.

Parámetros

<code>muestra</code>	Puntero a la muestra de la señal de entrada (tipo <code>sample_t</code>)
----------------------	---

Devuelve

Devuelve la muestra de la señal de salida del filtro.

2.2.2.2. `void ini_fir (sample_t coefs[])`

Inicializa el filtro.

Parámetros

<code>coefs[]</code>	Coeficientes del filtro (array de elementos tipos <code>sample_t</code>)
----------------------	---

Devuelve

No tiene salida.

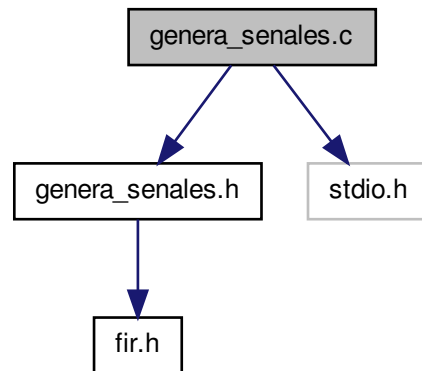
2.3. Referencia del Archivo `genera_senales.c`

Genera señales sintéticas para la prueba de la implementación del filtro FIR. Genera señales tipo escalón e impulso.

```
#include "genera_senales.h"
```

```
#include <stdio.h>
```

Dependencia gráfica adjunta para `genera_senales.c`:



Funciones

- void `step` (`sample_t` *pu, int tsubida, `sample_t` altura)
Genera señales tipo escalón.
- void `impulso` (`sample_t` *pu, int tsubida, `sample_t` altura)
Genera señales tipo impulso.

2.3.1. Descripción detallada

Genera señales sintéticas para la prueba de la implementación del filtro FIR. Genera señales tipo escalón e impulso.

2.3.2. Documentación de las funciones

2.3.2.1. `impulso` (`sample_t` * pu, int tsubida, `sample_t` altura)

Genera señales tipo impulso.

Parámetros

<i>pu</i>	Puntero al array de la señal de entrada.
<i>tsubida</i>	Retardo (en muestras) del impulso.
<i>altura</i>	Altura del impulso a generar.

Devuelve

No tiene salida.

2.3.2.2. step (sample_t * pu, int tsubida, sample_t altura)

Genera señales tipo escalón.

Parámetros

<i>pu</i>	Puntero al array de la señal de entrada.
<i>tsubida</i>	Retardo (en muestras) del escalón.
<i>altura</i>	Altura del escalón a generar.

Devuelve

No tiene salida.

2.4. Referencia del Archivo genera_senales.h

Genera señales sintéticas para la prueba de la implementación del filtro FIR. Genera señales tipo escalón e impulso.

```
#include "fir.h"
```

Dependencia gráfica adjunta para genera_senales.h:

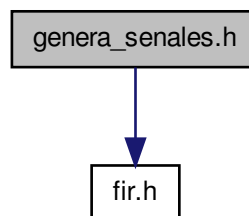
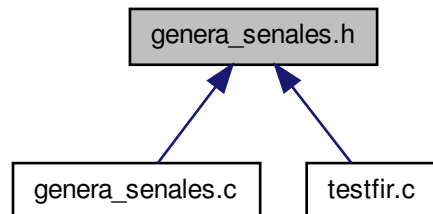


Gráfico de los archivos que directa o indirectamente incluyen a este archivo:



Funciones

- void [step](#) ([sample_t](#) *pu, int tsubida, [sample_t](#) altura)
Genera señales tipo escalón.
- void [impulso](#) ([sample_t](#) *pu, int tsubida, [sample_t](#) altura)
Genera señales tipo impulso.

2.4.1. Descripción detallada

Genera señales sintéticas para la prueba de la implementación del filtro FIR. Genera señales tipo escalón e impulso.

2.4.2. Documentación de las funciones

2.4.2.1. void impulso ([sample_t](#) * pu, int tsubida, [sample_t](#) altura)

Genera señales tipo impulso.

Parámetros

<i>pu</i>	Puntero al array de la señal de entrada.
<i>tsubida</i>	Retardo (en muestras) del impulso.
<i>altura</i>	Altura del impulso a generar.

Devuelve

No tiene salida.

2.4.2.2. void step (sample_t * pu, int tsubida, sample_t altura)

Genera señales tipo escalón.

Parámetros

<i>pu</i>	Puntero al array de la señal de entrada.
<i>tsubida</i>	Retardo (en muestras) del escalón.
<i>altura</i>	Altura del escalón a generar.

Devuelve

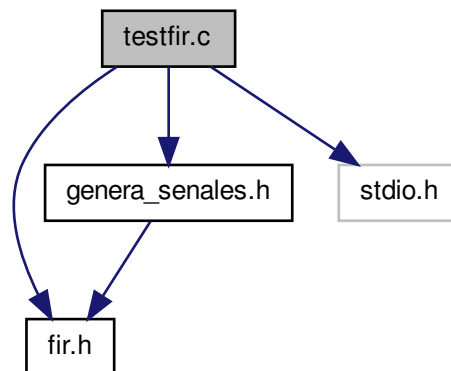
No tiene salida.

2.5. Referencia del Archivo testfir.c

Programa para probar el filtro FIR implementado.

```
#include "fir.h"
#include "genera_senales.h"
#include <stdio.h>
```

Dependencia gráfica adjunta para testfir.c:



Funciones

- int main ()

Prueba de los módulos implementados.

2.5.1. Descripción detallada

Programa para probar el filtro FIR implementado.

2.5.2. Documentación de las funciones

2.5.2.1. main ()

Prueba de los módulos implementados.

Devuelve

No tiene salida.

Señal de entrada.

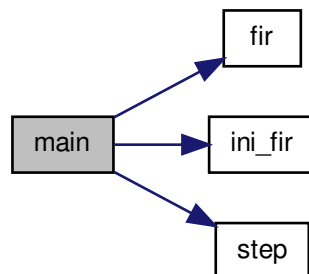
Señal de salida.

Coefficientes del FIR.

Retardo de la señal de entrada (en muestras).

Altura de la señal de entrada

Gráfico de llamadas para esta función:



Índice alfabético

- fir
 - fir.c, [4](#)
 - fir.h, [6](#)
- fir.c, [3](#)
 - fir, [4](#)
 - ini_fir, [4](#)
- fir.h, [4](#)
 - fir, [6](#)
 - ini_fir, [6](#)
- genera_senales.c, [6](#)
 - impulso, [7](#)
 - step, [8](#)
- genera_senales.h, [8](#)
 - impulso, [9](#)
 - step, [9](#)
- impulso
 - genera_senales.c, [7](#)
 - genera_senales.h, [9](#)
- ini_fir
 - fir.c, [4](#)
 - fir.h, [6](#)
- main
 - testfir.c, [11](#)
- step
 - genera_senales.c, [8](#)
 - genera_senales.h, [9](#)
- testfir.c, [10](#)
 - main, [11](#)