

PROCESAMIENTO DIGITAL DE SEÑALES

Laboratorio N°2: Filtros FIR con FRDM-K64F

DOCENTES:

Parlanti, Gustavo.

Rossi, Roberto.

Molina, German.

ALUMNOS:

Gomez Augusto Rodrigo, matricula: 39807998

Ferraris, Domingo Jesús, matricula: 36656566

Alaniz Franco, matricula: 37234066

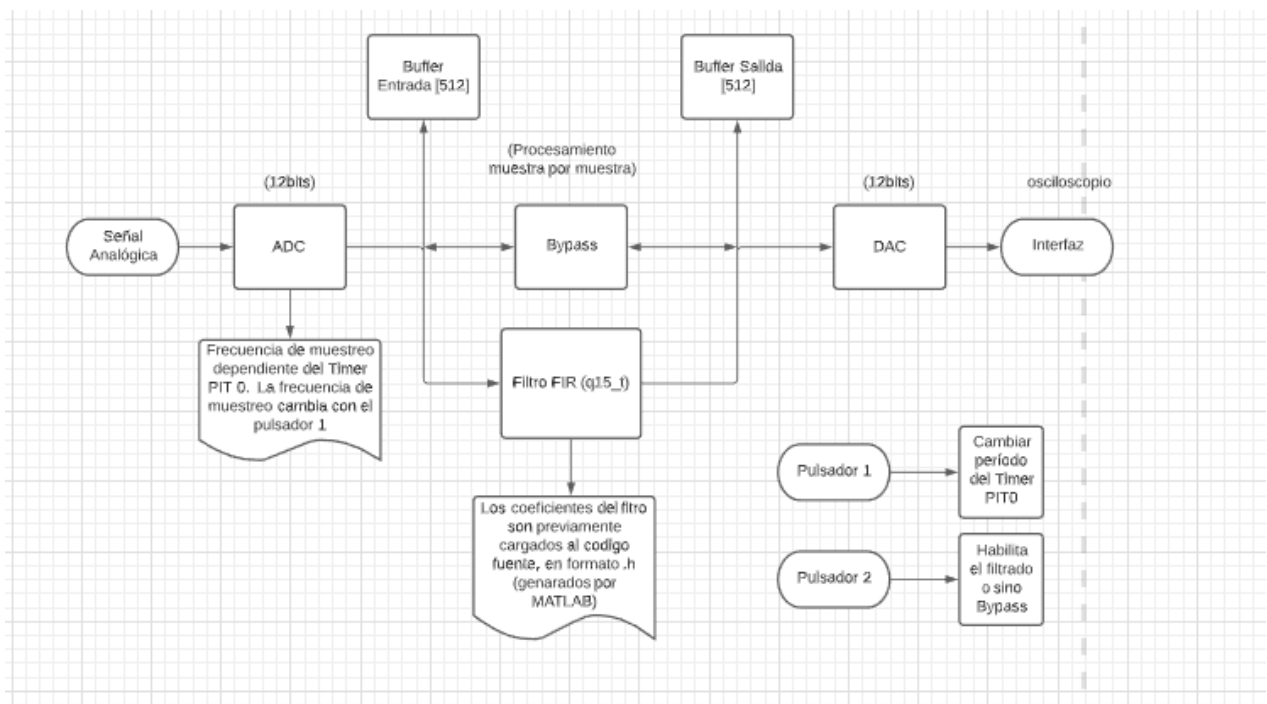
Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

Diagrama de bloques simplificado de funcionamiento:

En el siguiente diagrama se muestra el funcionamiento básico del programa, mostrando la funcionalidad de los pulsadores, y que el procesamiento de la señal se realiza muestra por muestra.

La configuración del ADC, DAC, timer PIT0, GPIO, es la misma que el trabajo N°1, por lo que no es necesario mostrar cómo se configuraron en el la vista de Peripherals.



Diseño de los filtros

_____ Para obtener los coeficientes de cada filtro, se usó la aplicación de MATLAB “Filter Designer”, luego completado los requerimientos se exportó un archivo .h con los coeficientes en formato punto fijo q15. Se adjuntan ahora imágenes más representativas del diseño de cada filtro. Las frecuencias de muestreo son 8KHz, 16 KHz, 22 KHz, 44 KHz y 48 KHz.

Cabe destacar que en algunos casos, los requerimientos del filtro no se pudieron completar en su totalidad, se tuvo que relajar las frecuencias de corte, ya que el orden del mismo resultaba ser muy grande, lo que ocasionaba problemas de retardo y por ende la performance no era la esperada.

Para todos los casos la frecuencia de corte de los filtros pasa altos se estableció en 100 Hz aprox.

Los filtros elimina banda no se logró hacer $F_r=50$ Hz (se tuvo que correr a un valor más alto) y el ancho de banda es más grande que el especificado.

Facultad de Ciencia Exactas, Físicas y Naturales

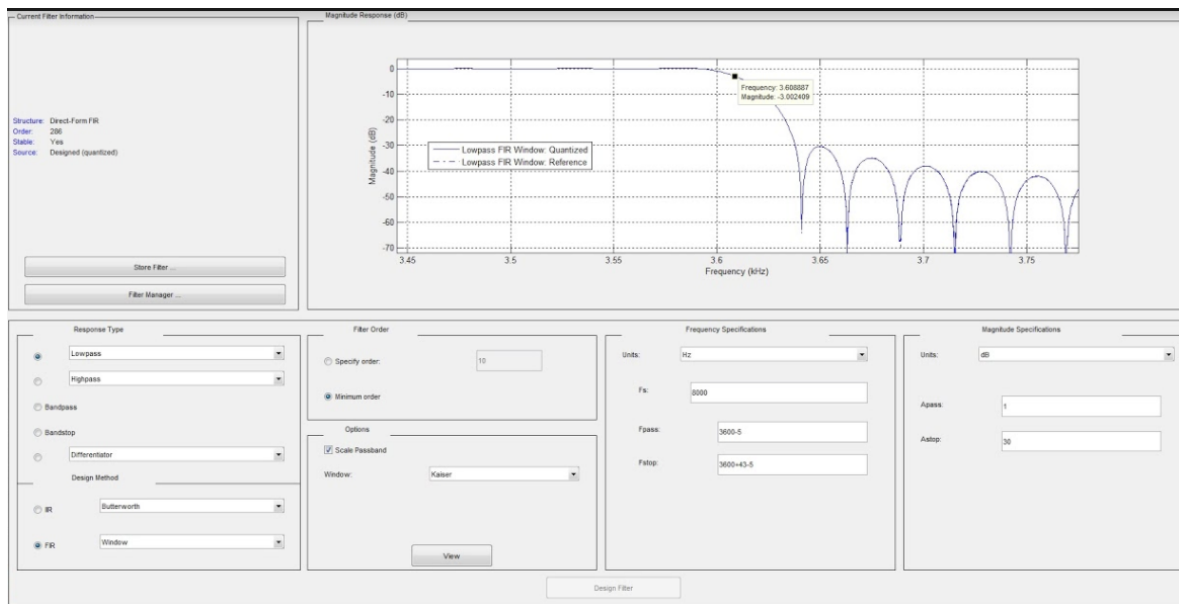
Procesamiento digital de señales

Para $F_s = 8\text{kHz}$

Para estos filtros no hubo mayores inconvenientes, por lo que en la mayoría de los casos se creó un primer filtro tentativo y de ahí se fue mejorando la selectividad, aumentando los coeficientes del filtro hasta un límite de entorno a los 200.

Filtro Paso Bajo:

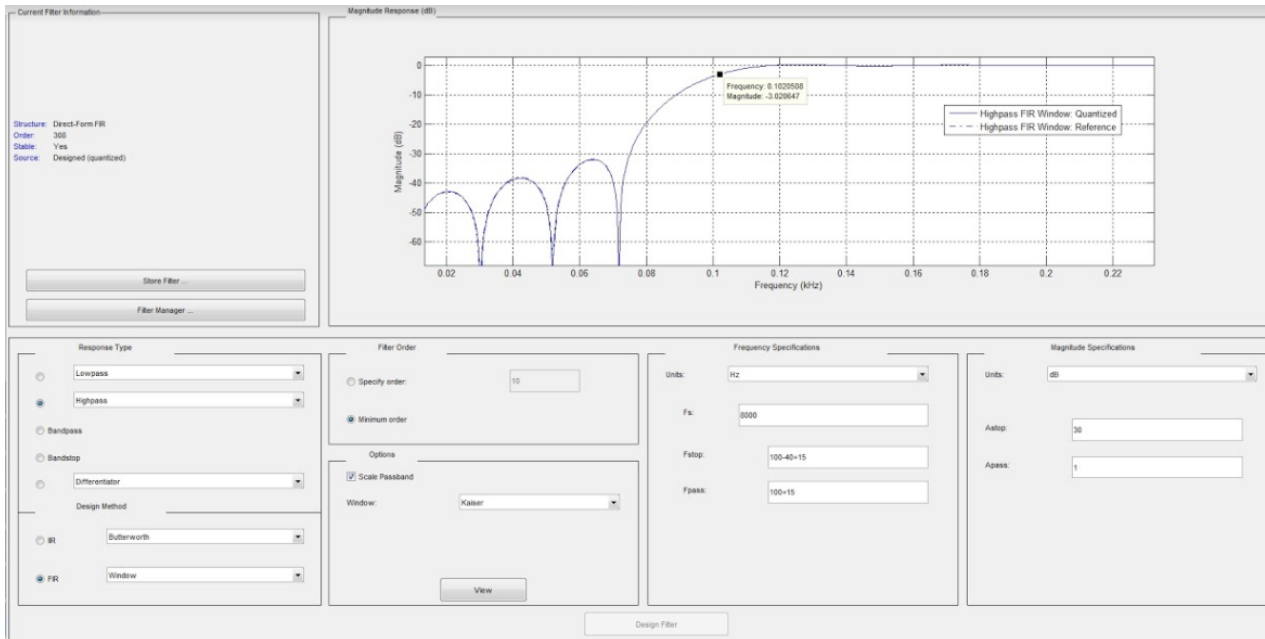
Se consideran cumplidos los requerimientos de atenuación y frecuencia de corte. Además se logró hacerlo selectivo con una banda de transición de menos de 50Hz.



Filtro Paso Alto:

En este caso tampoco tuvimos mayores inconvenientes en el diseño y podemos considerar cumplidos los requerimientos de atenuación y frecuencia de corte. Además se aumentaron los coeficientes para aumentar su selectividad.

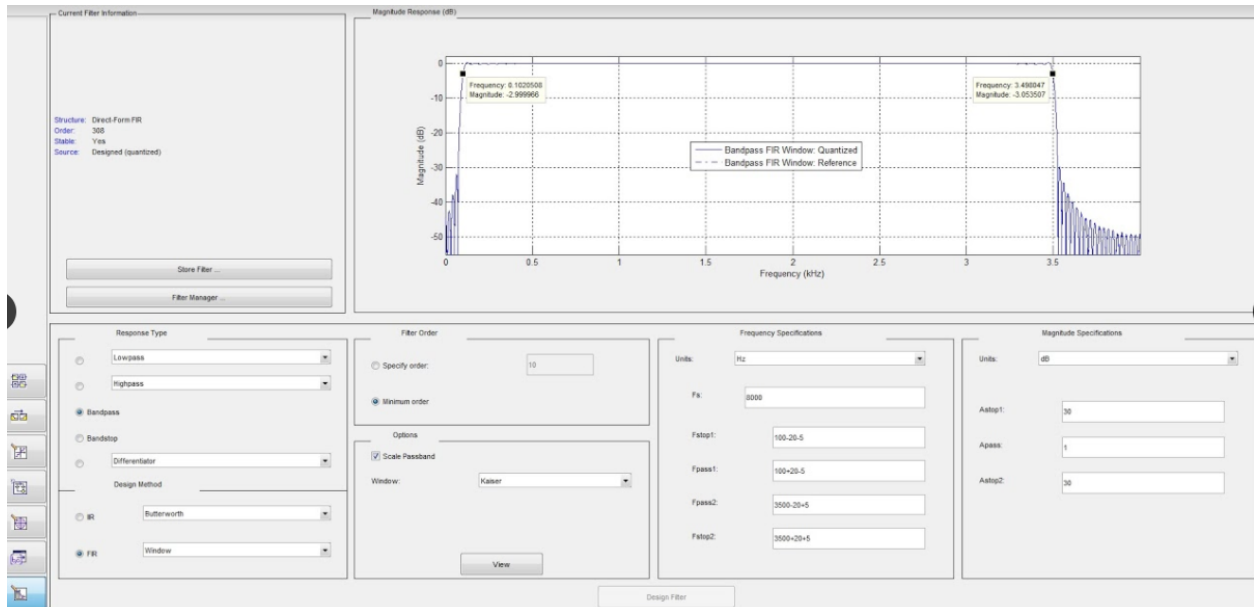
Facultad de Ciencia Exactas, Físicas y Naturales Procesamiento digital de señales



Filtro Paso Banda:

En este caso se tuvieron pequeños inconvenientes por ser baja la frecuencia de corte inferior, lo que obliga a tener una pendiente mayor para poder cumplir con la atenuación indicada, sin embargo se logró cumplir con los requerimientos y nuevamente se ajustó el orden para obtener mejor selectividad.

Facultad de Ciencia Exactas, Físicas y Naturales Procesamiento digital de señales



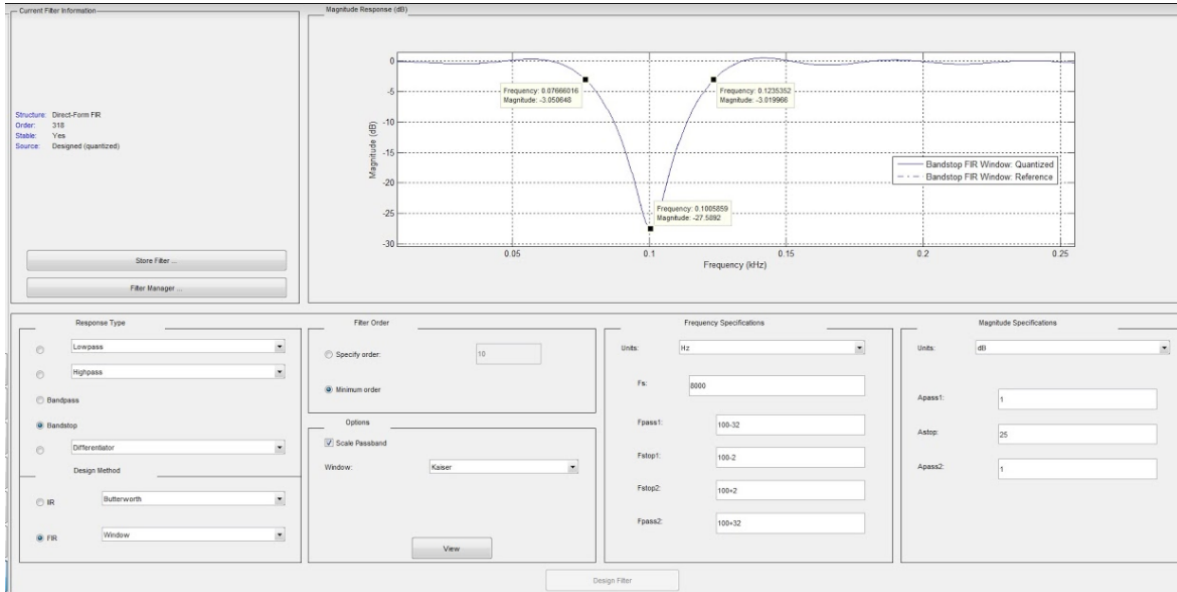
Filtro Elimina Banda:

En este caso no se logró una cantidad razonable de coeficientes al diseñarlo para un rechazo de 50Hz, ni aumentando al máximo el ancho de banda ni aumentando el ripple en la banda de paso. Por tanto se cambió la frecuencia de rechazo a 100Hz y se logró el requisito de atenuación y un

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

ancho de banda de 40Hz aprox.



Para $F_s=16\text{kHz}$

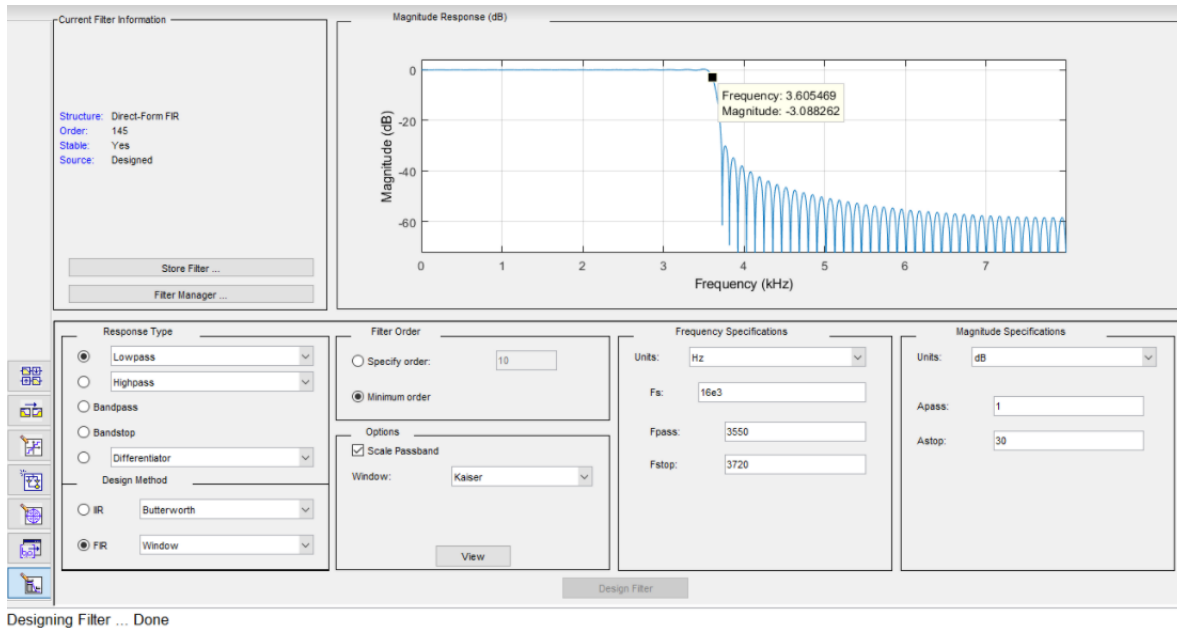
Aquí empezaron a aparecer algunos inconvenientes en los filtros pasa alto y rechaza banda por las bajas frecuencias de corte y rechazo.

Filtro Paso Bajo:

Para este filtro se logró cumplir con los requerimientos indicados, sin mayores inconvenientes y con pocos coeficientes aunque una menor selectividad.

Facultad de Ciencia Exactas, Físicas y Naturales

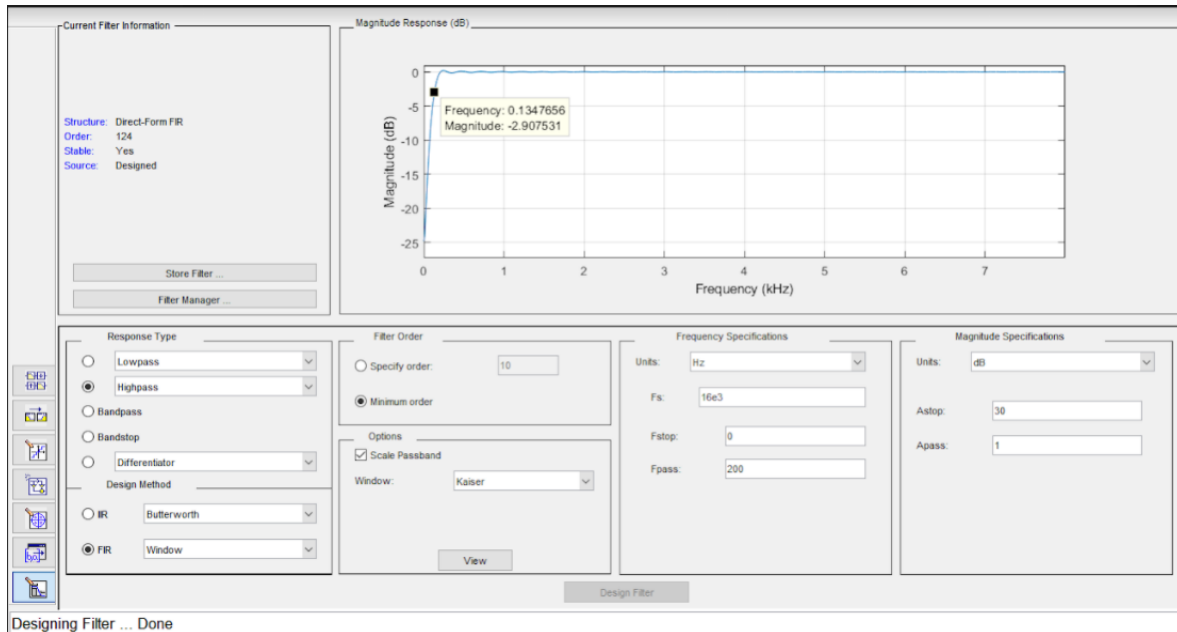
Procesamiento digital de señales



Filtro Paso Alto:

Aquí se tuvieron pequeños inconvenientes con la baja frecuencia de corte y la atenuación requerida, en este caso se aumentó la frecuencia de corte mínimamente para lograr implementar el filtro.

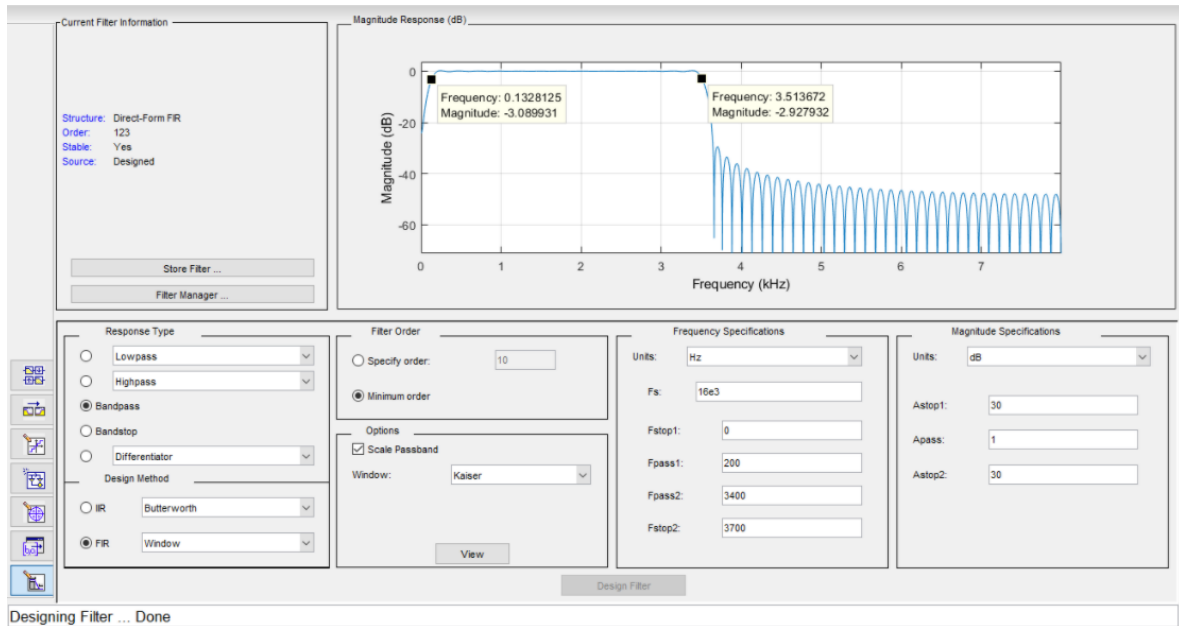
Facultad de Ciencia Exactas, Físicas y Naturales Procesamiento digital de señales



Filtro Pasa Banda:

En este caso por la baja frecuencia de corte inferior no se logró cumplir con la atenuación requerida en la banda eliminada inferior, por lo que se optó por un filtro de pocos coeficientes y con la frecuencia de corte inferior levemente aumentada para obtener una atenuación de al menos 20dB.

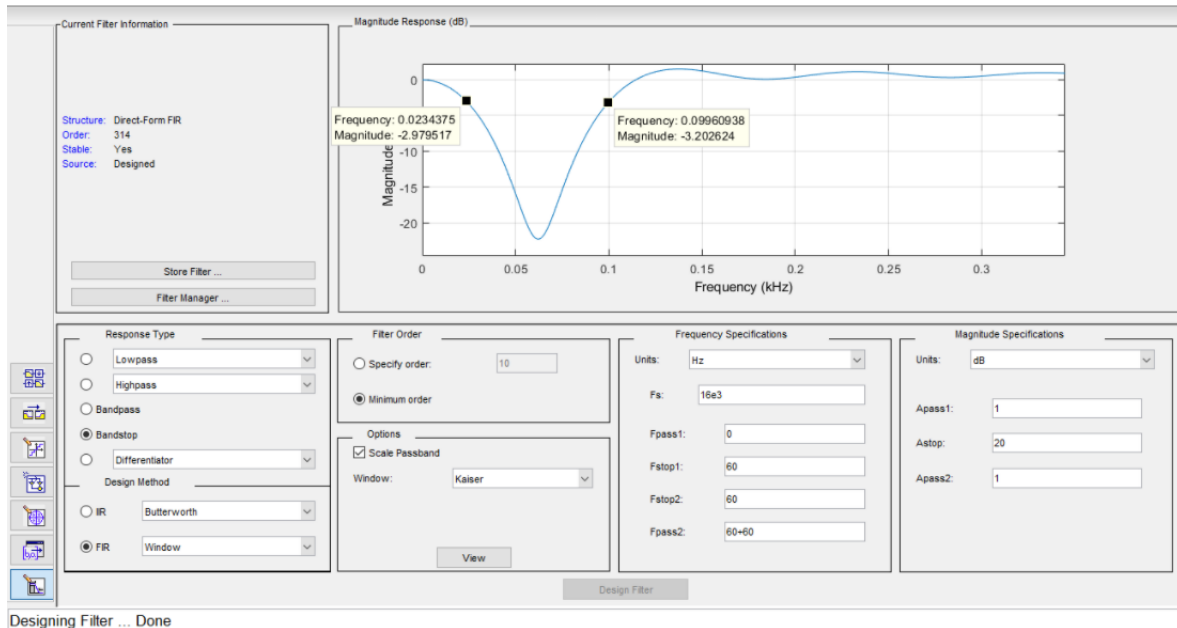
Facultad de Ciencia Exactas, Físicas y Naturales **Procesamiento digital de señales**



Filtro Elimina Banda:

_____Aquí se aumentó levemente la frecuencia de rechazo, se relajo el requerimiento de atenuación en 5dB y se aumentó el ancho de banda de forma que la banda de paso inferior llegue a los 0dB. Con todos esto se logró un orden superior a 300 coeficientes.

Facultad de Ciencia Exactas, Físicas y Naturales Procesamiento digital de señales



Para $F_s = 22\text{kHz}$

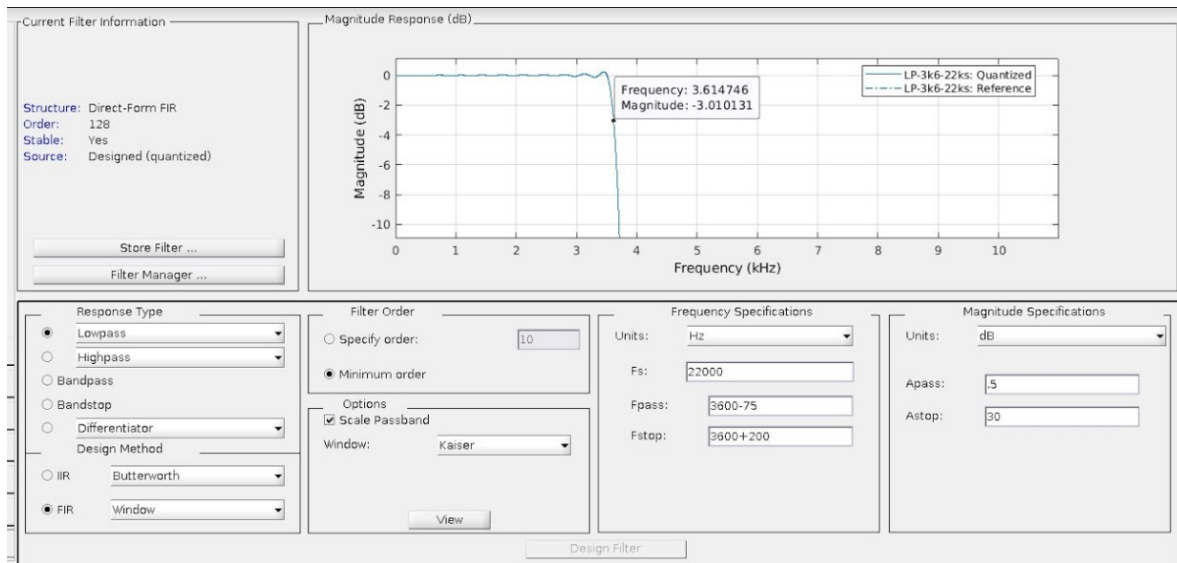
____ Para estos filtros se tomó en su mayoría más de 200 coeficientes y en algunos casos se intentó bajar el ripple en las bandas de paso para visualizar el impacto en la señal de salida.

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

Filtro Paso Bajo:

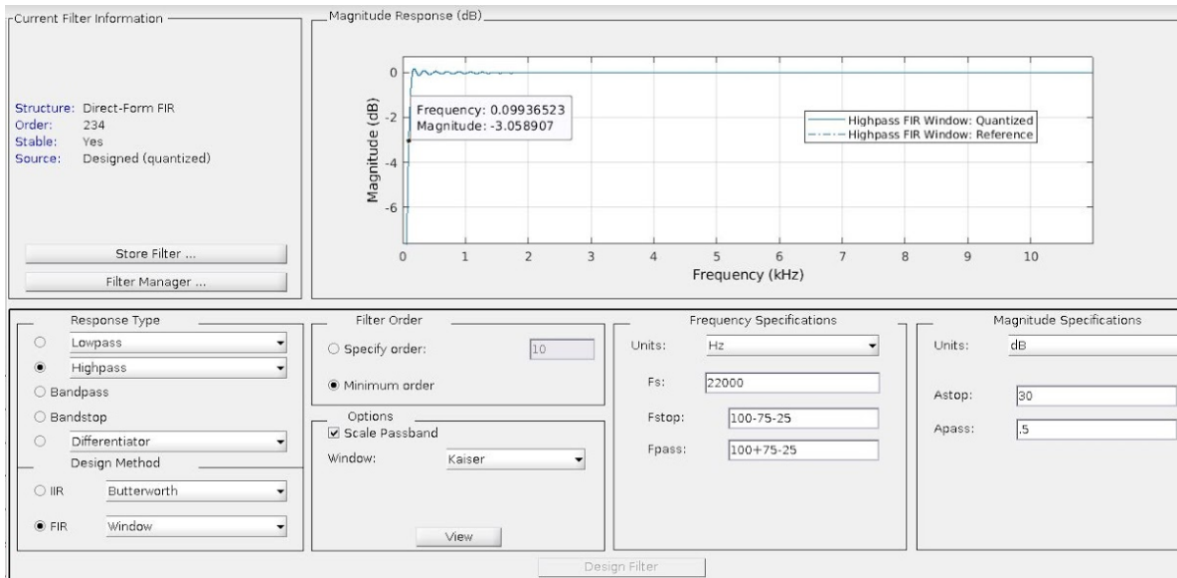
Para el paso bajo no se presentaron dificultades para cumplir con la atenuación con una pendiente ligeramente relajada y se optó por atenuar a la mitad el ripple en la banda de paso.



Filtro Paso Alto:

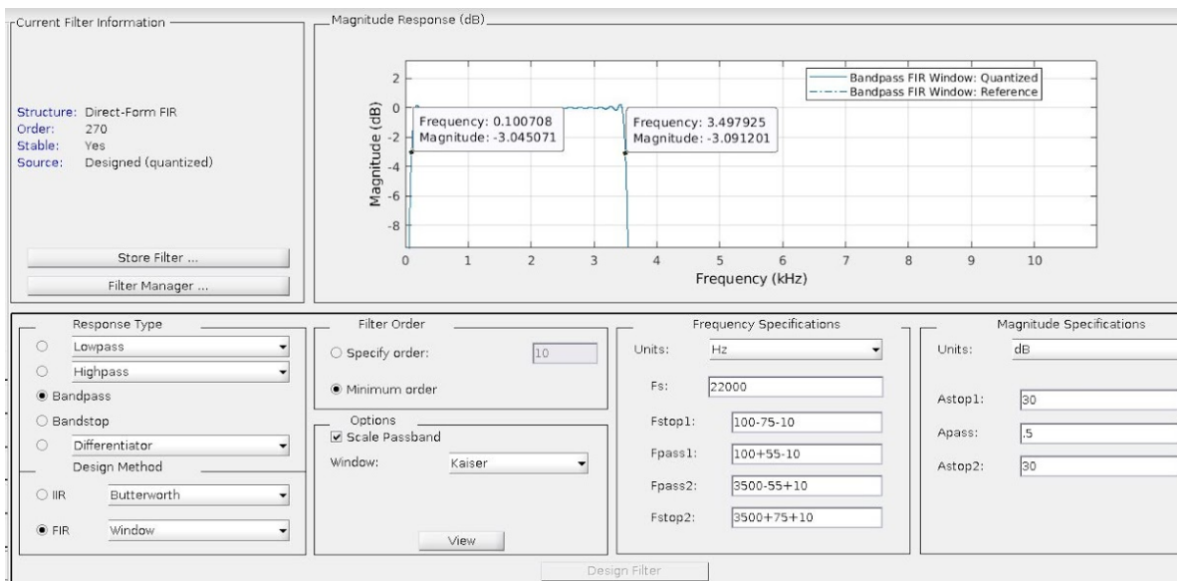
Para este caso se logró cumplir con las especificaciones y además se disminuyó el ripple en la banda de paso a 0.5dB.

Facultad de Ciencia Exactas, Físicas y Naturales Procesamiento digital de señales



Filtro Pasa Banda:

Sin sin mayores inconvenientes se logró un filtro de orden 270 de menor ripple en la banda de paso y que cumple con las especificaciones, aunque con una atenuación en la banda inferior y una selectividad un tanto relajadas.

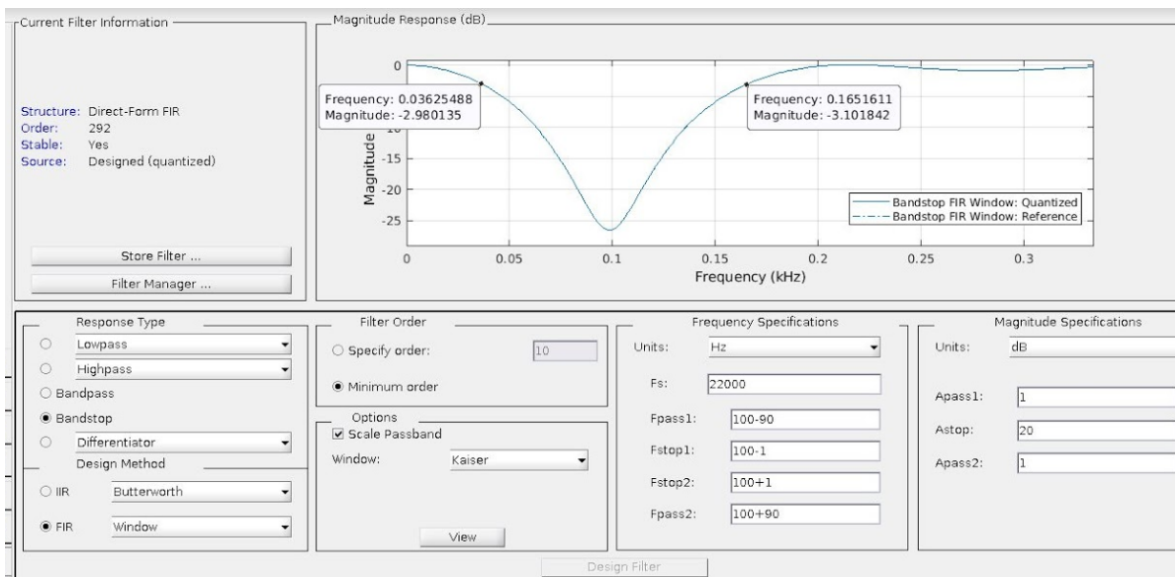


Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

Filtro Elimina Banda:

En este caso se tuvieron inconvenientes para cerrar el diseño con una cantidad de coeficientes aceptable. Se optó por correr a 100Hz la frecuencia de rechazo, aumentar el ancho de banda al máximo (160Hz aprox) y relajar la atenuación de 20dB.



Para $F_s = 44\text{kHz}$

A partir de esta frecuencia de muestreo (y también para la siguiente), empezaron a surgir problemas con respecto al tiempo de procesamiento de parte de la función intrínseca que hace el filtrado. Este tiempo empezaba a ser más grande que el período de muestreo $1/F_s$, y además teniendo en cuenta el tiempo para adquirir una muestra, hacían que el programa empezará a descartar nuevas muestras porque el procesamiento de las anteriores era demasiado lento, y la salida del DAC evidenciaba distorsiones en la señal.

Esto ocurría cuando el orden de los filtros era 100 o más, por lo que se optó por reducir el orden hasta como mucho ese valor. Lógicamente esto se

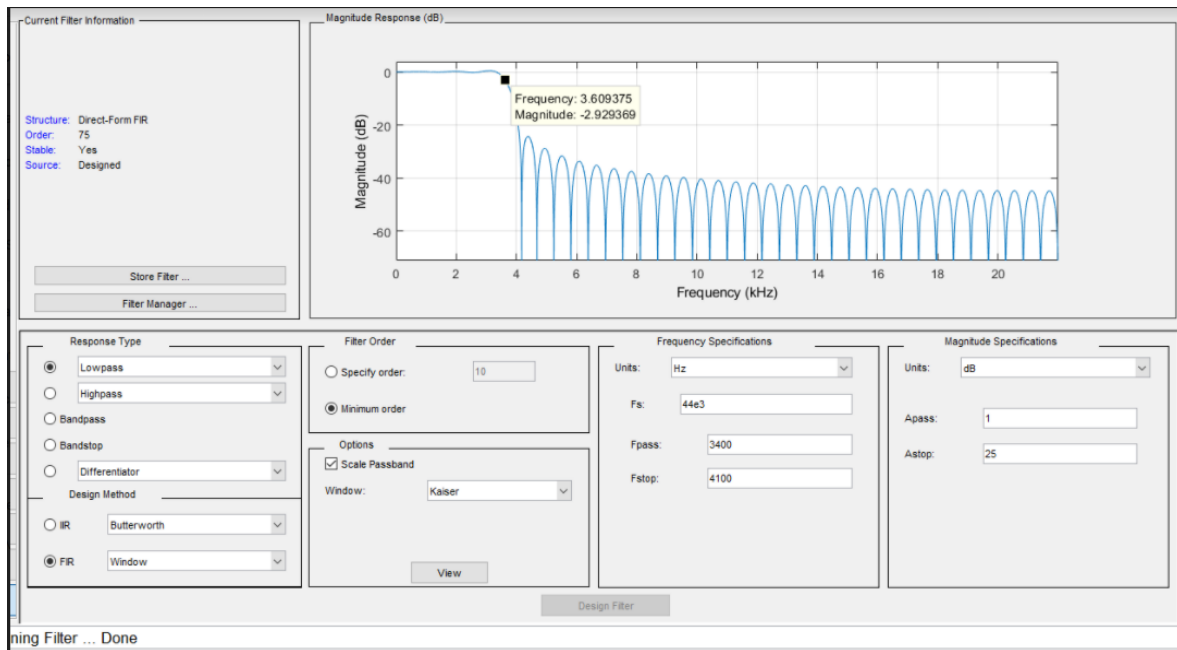
Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

logró cambiando la frecuencia de corte, relajando la pendiente en la banda de transición, y disminuyendo la atenuación en algunos casos.

Filtro Paso Bajo:

En este caso se optó por relajar el requerimiento de atenuación para cerrar el diseño con menos de 100 coeficientes.

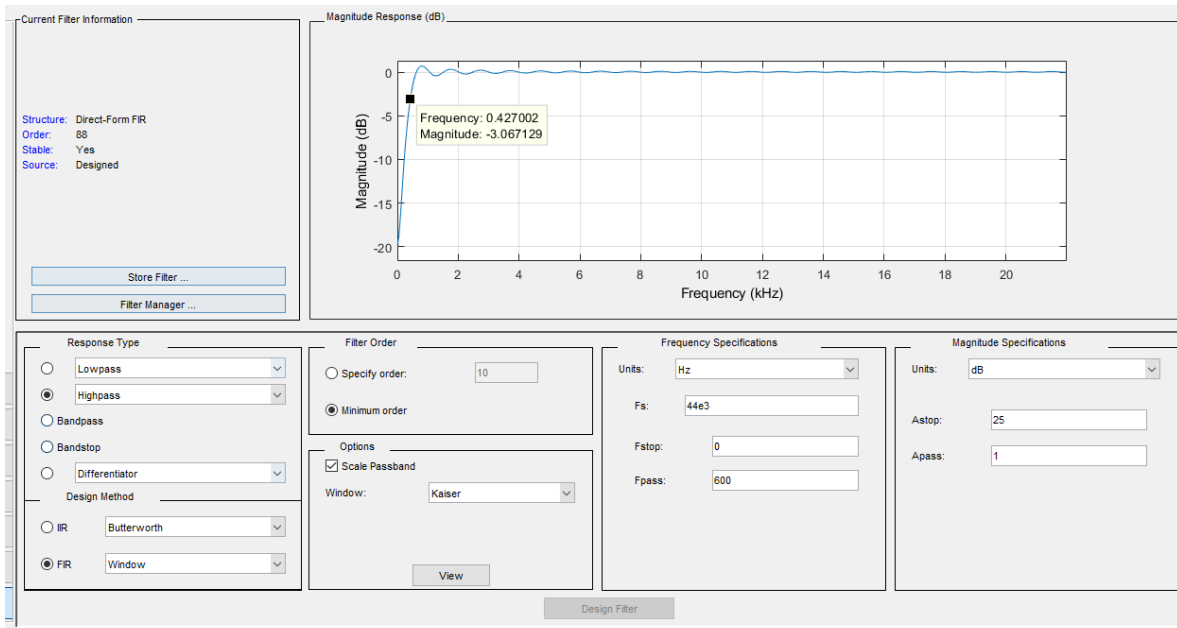


Filtro Paso Alto:

Para este filtro además de relajar la atenuación en 5dB fue inevitable correr la frecuencia de corte a 427Hz para cerrar el diseño con un orden coherente.

Facultad de Ciencia Exactas, Físicas y Naturales

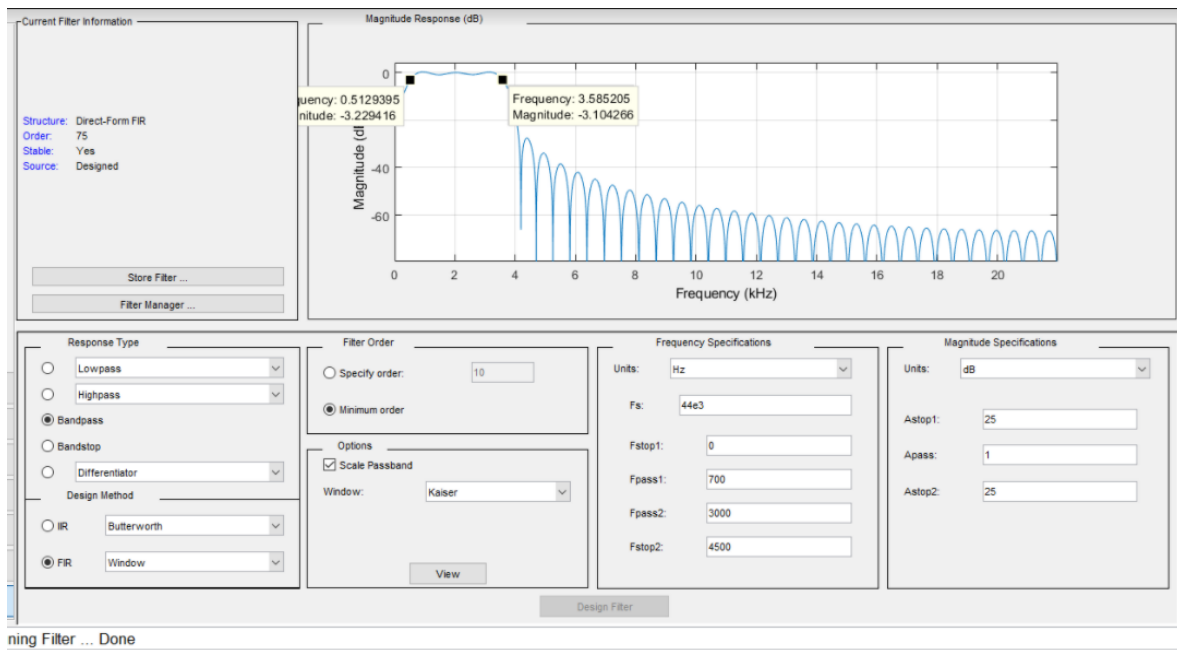
Procesamiento digital de señales



Filtro Pasa Banda:

Para el basa banda comenzaron a aparecer inconvenientes para cerrar el diseño así que se optó por correr la frecuencia de corte inferior a 510Hz aprox y relajar la atenuacion a 25dB.

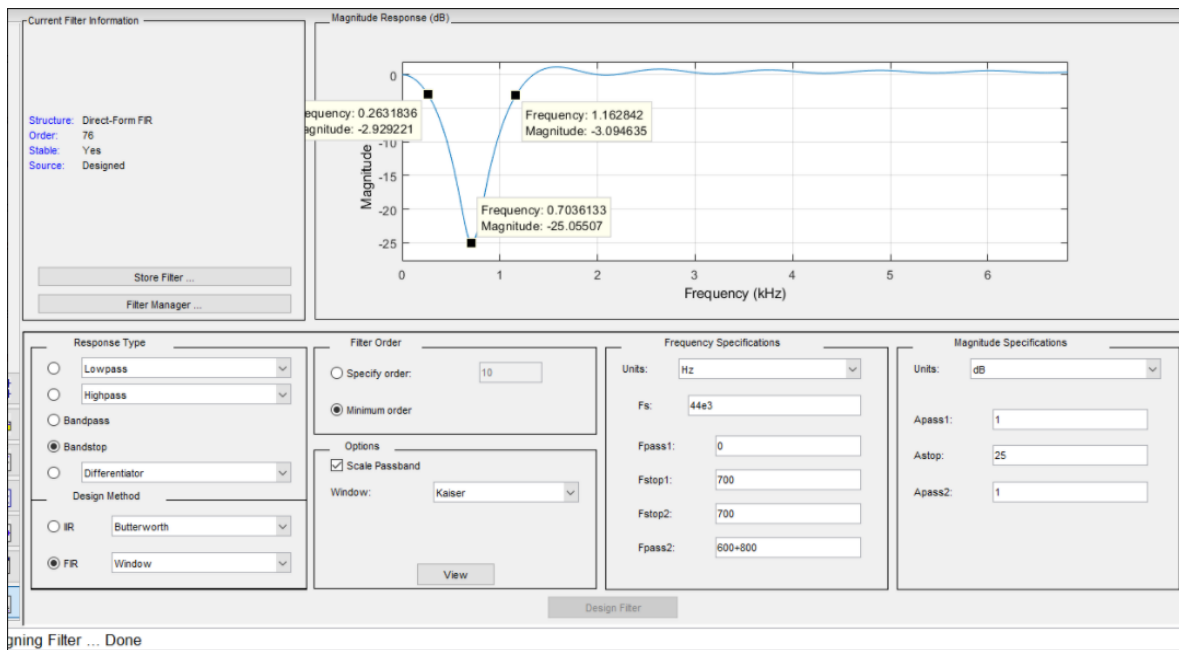
Facultad de Ciencia Exactas, Físicas y Naturales **Procesamiento digital de señales**



Filtro Elimina Banda:

Para este filtro se tuvieron los mayores inconvenientes por ser la frecuencia de rechazo demasiado baja, en este caso se optó por correr dicha frecuencia a 700Hz y aumentar el ancho de banda a casi 900Hz, pero mantener la atenuación requerida.

Facultad de Ciencia Exactas, Físicas y Naturales Procesamiento digital de señales

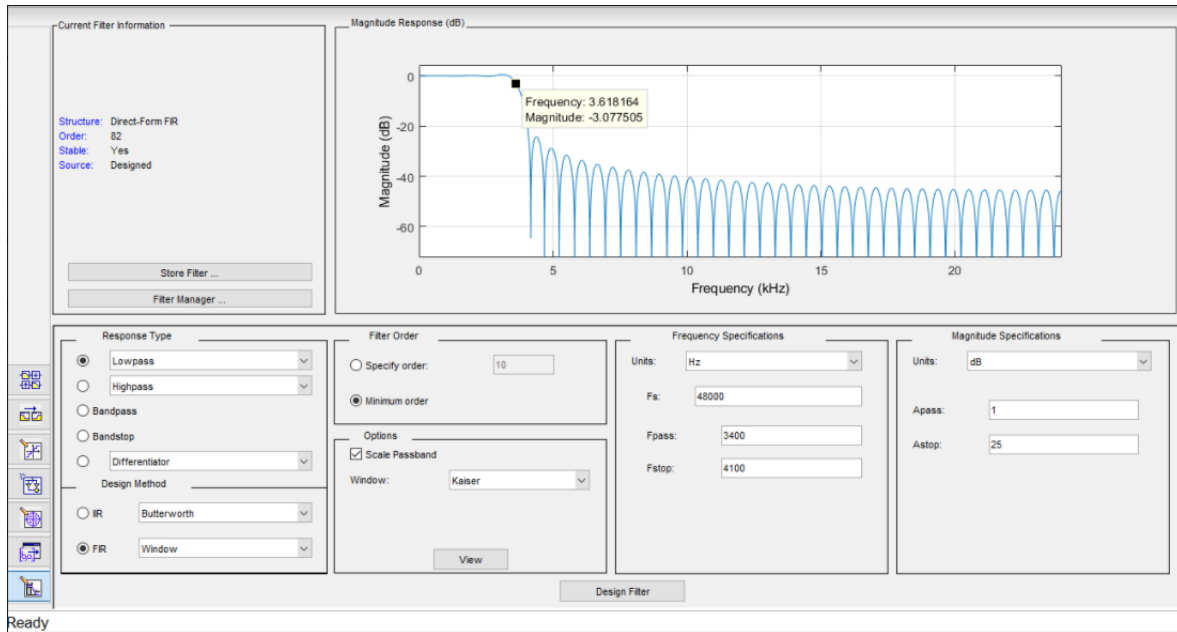


Para $F_s = 48\text{kHz}$

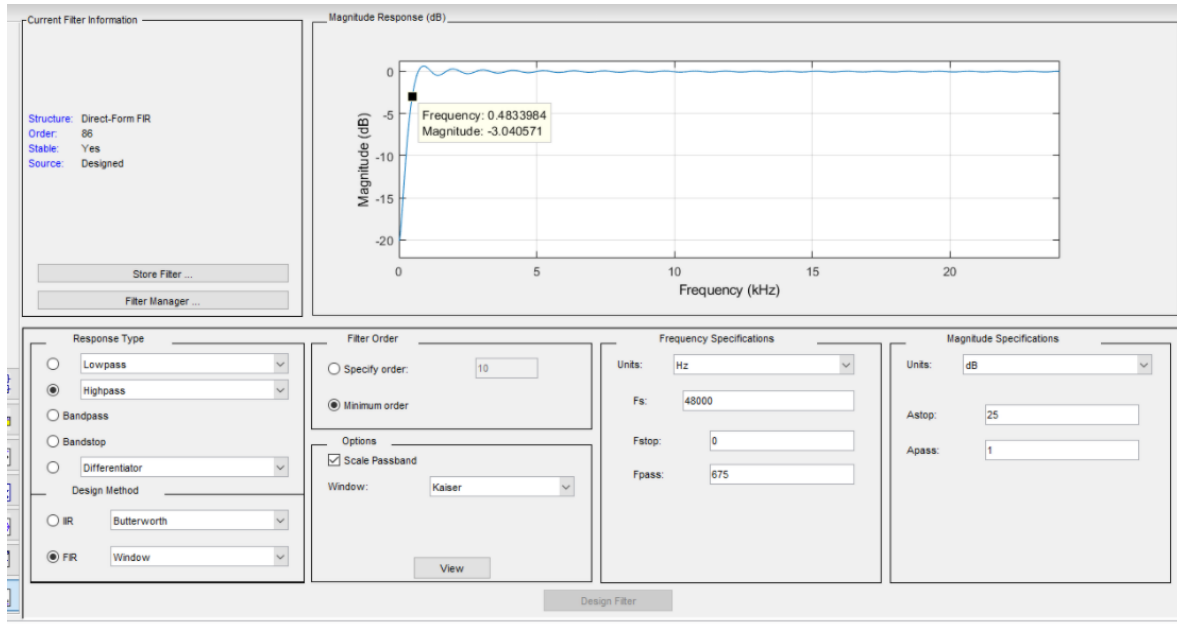
_____ Teniendo en cuenta las mismas consideraciones que se tuvieron para la frecuencia de muestreo anterior. No hubo mayores dificultades en la realización de estos filtros.

Filtro Paso Bajo:

Facultad de Ciencia Exactas, Físicas y Naturales Procesamiento digital de señales

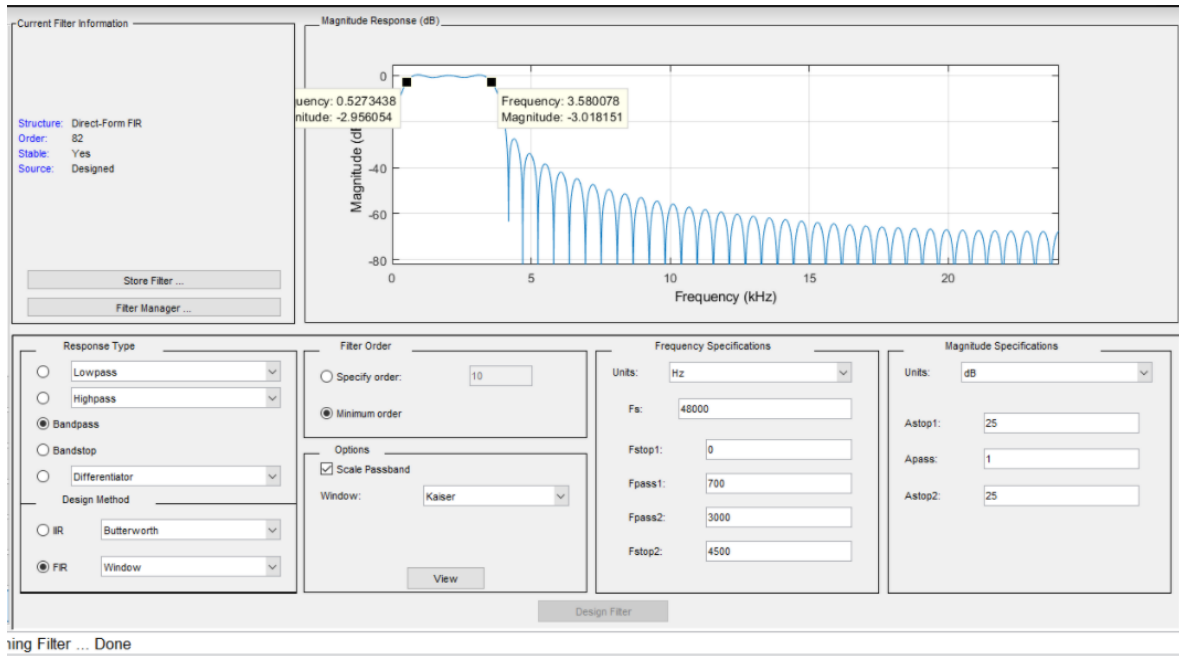


Filtro Paso Alto:

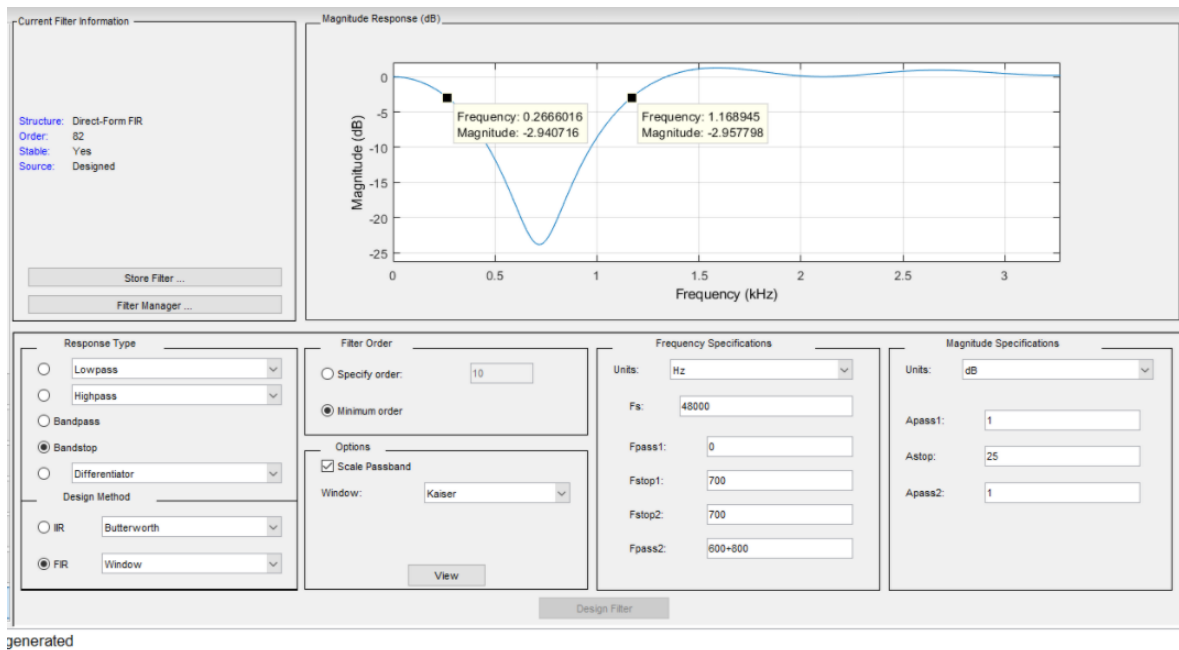


Filtro Pasa Banda:

Facultad de Ciencia Exactas, Físicas y Naturales Procesamiento digital de señales

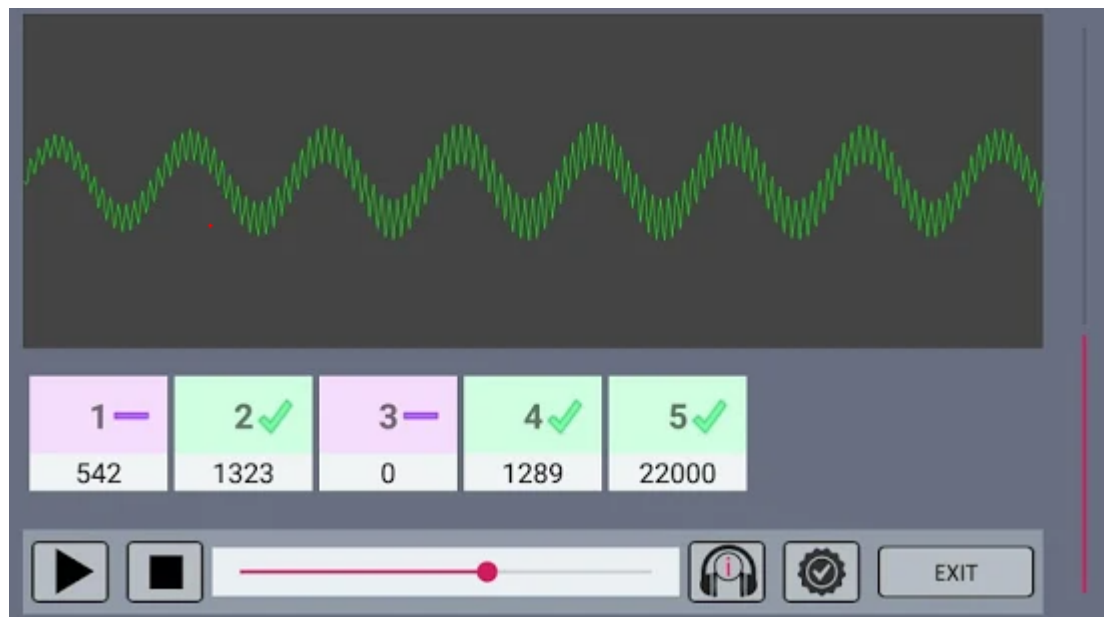


Filtro Elimina Banda:



Video demostración del funcionamiento del funcionamiento del programa:

Para poder complementar la experiencia y aprovechando que un integrante del grupo posee osciloscopio, se procedió a probar físicamente la performance del programa, para ello la señal analógica de entrada es la salida de un celular. Mediante el uso de una aplicación llamada **Multi-wave sound generator**.

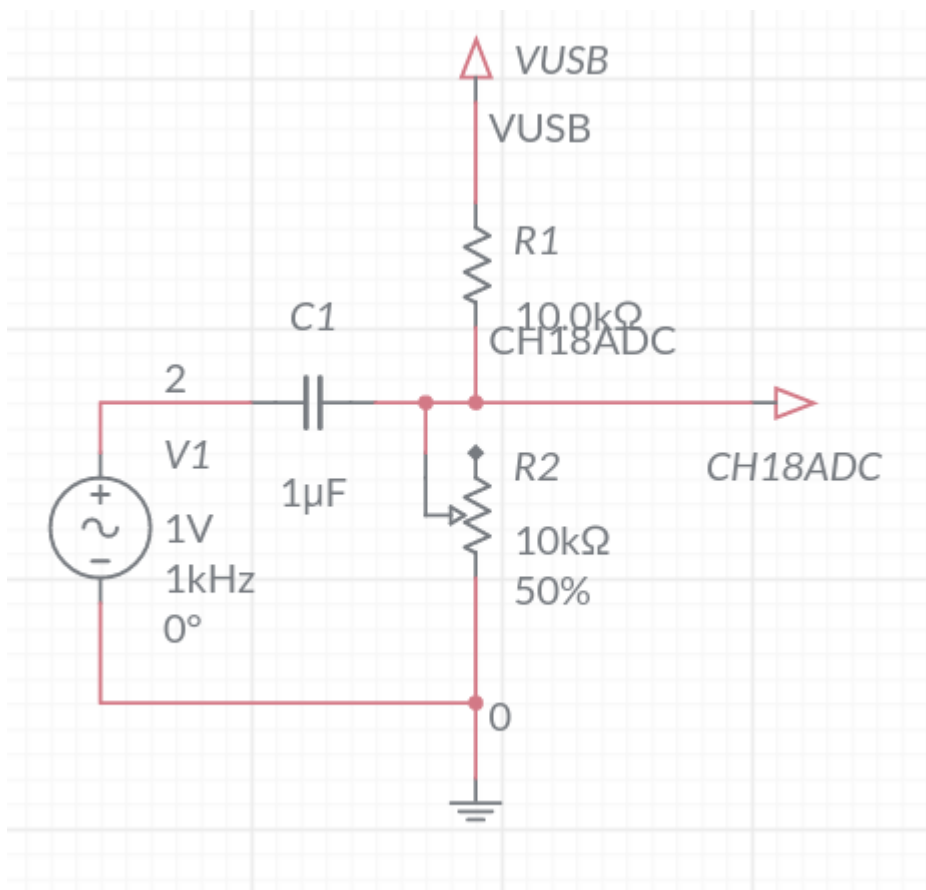


En la misma se puede programar que la salida sea la suma de varios tonos de distintas frecuencias, entre otras cosas. Lo cual es de utilidad para probar los filtros.

Debido a que el conversor trabaja con tensiones positivas, se agregó una tensión continua a la señal alterna, mediante un divisor de tensión variable y colocando un capacitor en su punto medio. Básicamente la señal entra por el capacitor y en el punto medio del divisor tenemos la señal montada al nivel de continua obtenida por el mismo.

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales



Las demostraciones se encuentran en esta carpeta de drive:

[Demostraciones en video](#)

Código fuente utilizado:

```

/**
// * @file Tp_N2.c
// * @brief Realizar un programa aplicativo que sea capaz de aplicar un filtro FIR,
// * por muestras, a un buffer de memoria de 512 muestras, que es adquirido con
// * el Laboratorio #1 teniendo en cuenta las frecuencias de muestreo de 8KHz,

```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```
// * 16 KHz, 22 KHz, 44 KHz y 48 KHz. Con una de las teclas de la placa de
// * evaluación FRDM-K64F, se habilitara la aplicación del filtro o se hace
// * bypass del filtro a un buffer de salida distinto del buffer de entrada. De este
// * buffer de salida se enviaran las muestras al DAC de la palca FRDM-K64F.
// * Visualizar el resultado utilizando un osciloscopio y comparar los resultados.
// *
// * Requerimientos de los filtros:
// * Pasa Bajos: Fc= 3600Hz Astop=30db
// * Pasa Altos: Fc=35 Hz Astop=30db
// * Pasa Banda Fc1=35Hz, Fc2=3500Hz Astop=30db
// * Elimina Banda Fr= 50 Hz Bw=15 Hz Astop=25db
// */

#include <stdio.h>

#include "board.h"

#include "peripherals.h"

#include "pin_mux.h"

#include "clock_config.h"

#include "MK64F12.h"

#include "fsl_debug_console.h"

/* TODO: insert other include files here. */

#include "fsl_gpio.h"

#include "fsl_common.h"

#include "fsl_dac.h"

#include "stdbool.h"
```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```
#include "stdint.h"

#include "arm_math.h"

/* TODO: insert other definitions and declarations here. */

/// low pass

#define NUM_COEFF_8k_3600 64

#define NUM_COEFF_16k_3600 146

#define NUM_COEFF_22k_3600 130

#define NUM_COEFF_44k_3600 228

#define NUM_COEFF_48k_3600 46

///high pass

#define NUM_COEFF_8K_100 90

#define NUM_COEFF_16k_100 126

#define NUM_COEFF_22k_100 236

#define NUM_COEFF_44k_100 90

#define NUM_COEFF_48k_100 102

///band pass

#define NUM_COEFF_8k_100_3500 310 //faltan exportar

#define NUM_COEFF_16k_200_3500 124

#define NUM_COEFF_22k_100_3500 272

#define NUM_COEFF_44k_100_3500 76

#define NUM_COEFF_48k_200_3500 88

///band stop

#define NUM_COEFF_8k_35_135 320
```


Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```
#define NUM_COEFF_16k_35_135      316

#define NUM_COEFF_22k_35_135      294

#define NUM_COEFF_44k_40_170      78

#define NUM_COEFF_48k_40_170      84


#define BLOCK_SIZE                1      //consultar el limite
#define BLOCK_MEMORI              512

/*-----
 * buffer de entrada y salida
-----*/

static q15_t input                 [BLOCK_MEMORI];
static q15_t output               [BLOCK_MEMORI];


/*-----
 * buffer para ir guardando los datos de salida
-----*/

///low pass

static q15_t state_buffer_8k_3600 [2+NUM_COEFF_8k_3600-1];
static q15_t state_buffer_16k_3600 [2+NUM_COEFF_16k_3600-1];
static q15_t state_buffer_22k_3600 [2+NUM_COEFF_22k_3600-1];
static q15_t state_buffer_44k_3600 [2+NUM_COEFF_44k_3600-1];
static q15_t state_buffer_48k_3600 [2+NUM_COEFF_48k_3600-1];


///high pass mayores a 100hrz
```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```
static q15_t state_buffer_8k_100      [2+NUM_COEFF_8K_100-1];
static q15_t state_buffer_16k_100     [2+NUM_COEFF_16k_100-1];
static q15_t state_buffer_22k_100     [2+NUM_COEFF_22k_100-1];
static q15_t state_buffer_44k_100     [2+NUM_COEFF_44k_100-1];
static q15_t state_buffer_48k_100     [BLOCK_MEMORI+NUM_COEFF_48k_100-1];
```

///
band pass

```
static q15_t state_buffer_8k_100_3500 [2+NUM_COEFF_8k_100_3500-1];
static q15_t state_buffer_16k_100_3500 [2+NUM_COEFF_16k_200_3500-1];
static q15_t state_buffer_22k_100_3500 [2+NUM_COEFF_22k_100_3500-1];
static q15_t state_buffer_44k_100_3500 [2+NUM_COEFF_44k_100_3500-1];
static q15_t state_buffer_48k_200_3500 [2+NUM_COEFF_48k_200_3500-1];
```

///
band stop

```
static q15_t state_buffer_8k_35_135   [2+NUM_COEFF_8k_35_135-1];
static q15_t state_buffer_16k_35_135 [2+NUM_COEFF_16k_35_135-1];

static q15_t state_buffer_22k_35_135 [2+NUM_COEFF_22k_35_135-1];
static q15_t state_buffer_44k_40_170 [2+NUM_COEFF_44k_40_170-1];

static q15_t state_buffer_48k_35_135 [2+NUM_COEFF_48k_40_170-1];
```

/*-----

* coeficientes

----- */

///
low pass

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```
extern const q15_t fir_coeffs_8k_3600[NUM_COEFF_8k_3600];  
extern const q15_t fir_coeffs_16k_3600[NUM_COEFF_16k_3600];  
extern const q15_t fir_coeffs_22k_3600[NUM_COEFF_22k_3600];  
extern const q15_t fir_coeffs_44k_3600[NUM_COEFF_44k_3600];  
extern const q15_t fir_coeffs_48k_3600[NUM_COEFF_48k_3600];
```

///
//high pass mayores a 100hrz

```
extern const q15_t fir_coeffs_8k_100[NUM_COEFF_8K_100];  
extern const q15_t fir_coeffs_16k_100[NUM_COEFF_16k_100];  
extern const q15_t fir_coeffs_22k_100[NUM_COEFF_22k_100];  
extern const q15_t fir_coeffs_44k_100[NUM_COEFF_44k_100];  
extern const q15_t fir_coeffs_48k_100[NUM_COEFF_48k_100];
```

///
//band pass

```
extern const q15_t fir_coeffs_8k_100_3500[NUM_COEFF_8k_100_3500];  
extern const q15_t fir_coeffs_16k_100_3500[NUM_COEFF_16k_200_3500];  
extern const q15_t fir_coeffs_22k_100_3500[NUM_COEFF_22k_100_3500];  
extern const q15_t fir_coeffs_44k_100_3500[NUM_COEFF_44k_100_3500];  
extern const q15_t fir_coeffs_48k_200_3500[NUM_COEFF_48k_100];
```

///
//band stop

```
extern const q15_t fir_coeffs_8k_40_170[NUM_COEFF_8k_35_135];  
extern const q15_t fir_coeffs_16k_40_170[NUM_COEFF_16k_35_135];
```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```

extern const q15_t fir_coeffs_22k_35_135[NUM_COEFF_22k_35_135];
extern const q15_t fir_coeffs_44k_40_170[NUM_COEFF_44k_40_170];
extern const q15_t fir_coeffs_48k_40_170[NUM_COEFF_48k_40_170];

/*-----
* instancias de fir
-----*/

///low pass
arm_fir_instance_q15 fir_8k_3600;
arm_fir_instance_q15 fir_16k_3600;
arm_fir_instance_q15 fir_22k_3600;
arm_fir_instance_q15 fir_44k_3600;
arm_fir_instance_q15 fir_48k_3600;

///high pass mayores a 100hrz
arm_fir_instance_q15 fir_8k_100;
arm_fir_instance_q15 fir_16k_100;
arm_fir_instance_q15 fir_22k_100;
arm_fir_instance_q15 fir_44k_100;
arm_fir_instance_q15 fir_48k_100;

///band pass
arm_fir_instance_q15 fir_8k_100_3500;
arm_fir_instance_q15 fir_16k_200_3500;

arm_fir_instance_q15 fir_22k_100_3500;

```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```
arm_fir_instance_q15 fir_44k_200_3500;
```

```
arm_fir_instance_q15 fir_48k_200_3500;
```

```
///  
band stop
```

```
arm_fir_instance_q15 fir_8k_40_170;
```

```
arm_fir_instance_q15 fir_16k_35_135;
```

```
arm_fir_instance_q15 fir_22k_35_135;
```

```
arm_fir_instance_q15 fir_44k_40_170;
```

```
arm_fir_instance_q15 fir_48k_40_170;
```

```
/*-----*/
```

```
* variables para instanciar el fir
```

```
-----*/
```

```
uint32_t block_size = BLOCK_SIZE;
```

```
uint16_t numBlocks = BLOCK_MEMORI/BLOCK_SIZE; //para trabajar sub  
dividiendo el filtro
```

```
/*-----*/
```

```
* tiempos de muestreo
```

```
* 8kHz USEC_TO_COUNT(125,PIT_CLK_FREQ)
```

```
* 16kHz USEC_TO_COUNT(63,PIT_CLK_FREQ)
```

```
* 22kHz USEC_TO_COUNT(45,PIT_CLK_FREQ)
```

```
* 44kHz USEC_TO_COUNT(23,PIT_CLK_FREQ)
```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```

* 48kHz          USEC_TO_COUNT(21,PIT_CLK_FREQ)
----- */

long time[5]={  USEC_TO_COUNT(125,PIT_CLK_FREQ),
                USEC_TO_COUNT(63,PIT_CLK_FREQ),
                USEC_TO_COUNT(45,PIT_CLK_FREQ),
                USEC_TO_COUNT(23,PIT_CLK_FREQ),
                USEC_TO_COUNT(21,PIT_CLK_FREQ));

/*-----

* banderas de estados
----- */

volatile bool enable_pit = false; //bande para para apagar el pit

volatile bool enable_led      = false; //badera para habilitar los led cada vez que pulso para
cambiar el periodo

volatile bool adc_convert     = false;

volatile bool bypass          = true;

volatile bool fir_enable = false;

/*-----

* ADC
----- */

q15_t adc_value;

/*-----

* variable para encender los led en base a la frecuencia del clock

* variable para el indice del buffer

* variables para tener un estimado de tiempos de filtrado y conversion del adc

* variable para elegir los distintos filtros
-----*/

```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```
volatile uint8_t count;

volatile uint32_t i;

float t,t1,t2,t3,t_adc_1,t_adc_2,t_adc_3,t_adc_4,t_total;

char seleccion;

/*-----
 * punteros a los tiempos de muestreo
----- */

long *direc,*lim_direc;

/*-----
 * punteros de auxiliares
-----*/

q15_t *fin_input,*output_p,*adc_value_p,*fin_output,*input_p;

/*-----
 * Interrupciones
----- */

void GPIOA_IRQHANDLER(void){
    enable_led=true;

    GPIO_PortClearInterruptFlags(BOARD_SW3_GPIO, 1U << BOARD_SW3_GPIO_PIN);

    if (direc==lim_direc) {
        direc  = &time[0];

        count =0;
    }

    else {
        direc++;
    }
}
```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```

    count++;

}

PIT_SetTimerPeriod(PIT_PERIPHERAL,PIT_CHANNEL_0,(uint32_t)*direc;

__DSB();

}

void GPIOC_IRQHANDLER(void){

    GPIO_PortClearInterruptFlags(BOARD_SW2_GPIO,1U<<BOARD_SW2_GPIO_PIN);

    // if(!enable_pit){
    //     PIT_StartTimer(PIT_PERIPHERAL, PIT_CHANNEL_0);
    //     enable_pit= true;
    // }

    // else {

        /*-----
        *    cambio del buffer de salida o de entrada en funcion del indice
        *    que se va guardando para no perder fase
        -----*/

        if(!bypass){

            output_p = &input[i-1];

            bypass=true;

            fir_enable=false;

        }

        else {

            output_p = &output[i-1];

            bypass=false;

```


Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```

        fir_enable=true;

    }

//    PIT_StopTimer(PIT_PERIPHERAL, PIT_CHANNEL_0);
//    enable_pit=false;
// }

__DSB();
}

/* TODO: PIT */
void PIT_CHANNEL_0_IRQHANDLER(void){

    PIT_ClearStatusFlags(PIT_PERIPHERAL, PIT_CHANNEL_0, kPIT_TimerFlag);

    PIT_StartTimer(PIT_PERIPHERAL, PIT_CHANNEL_2);

    ADC16_SetChannelConfig(ADC0_PERIPHERAL, ADC0_CH0_CONTROL_GROUP,
    &ADC0_channelsConfig[0]);

    t_adc_1 = PIT_GetCurrentTimerCount(PIT_PERIPHERAL,PIT_CHANNEL_2);

    __DSB();
}

/* TODO: ADC */
void ADC0_IRQHANDLER(void){

    t_adc_2 = PIT_GetCurrentTimerCount(PIT_PERIPHERAL,PIT_CHANNEL_2);

    PIT_StopTimer(PIT_PERIPHERAL, PIT_CHANNEL_2);

    adc_convert=true;

    ADC16_ClearStatusFlags(ADC0_PERIPHERAL, kADC16_ChannelConversionDoneFlag);

    /*-----
    * tomo el dato

```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```

-----*/

adc_value = ADC16_GetChannelConversionValue(ADC0_PERIPHERAL,
ADC0_CH0_CONTROL_GROUP)-1250u;

__DSB();
}

/*-----
* TP_N2
-----*/

int main(void) {

    /* Init board hardware. */

    BOARD_InitButtonsPins();

    BOARD_InitLEDsPins();

    BOARD_InitBootClocks();

    BOARD_InitBootPeripherals();

#ifdef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
    /* Init FSL debug console. */

    BOARD_InitDebugConsole();
#endif

    /*-----
    * variables del fir
    -----*/

    //uint32_t o; // por si quiero usar un block_size distinto de 512

    /*-----
    * assignacion de punteros

```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```

-----*/

direc          = &time[0];          //guardo la direccion de memoria de
periodos siempre se guarda la primera direccion

lim_direc      = &time[4];          //guardo la posicion de memoria del
ultimo periodo

input_p        = &input[0];

fin_input      = &input[511];

output_p       = &input[0];          //inicilizo que apunte a al buffer de entrada
fin_output     = &output[511];

adc_value_p = &adc_value;

/*-----
* inicializacion de distintos fir
-----*/

///low pass

arm_fir_init_q15(&fir_8k_3600, NUM_COEFF_8k_3600,&fir_coeffs_8k_3600[0],
&state_buffer_8k_3600[0], block_size);

arm_fir_init_q15(&fir_16k_3600, NUM_COEFF_16k_3600,&fir_coeffs_16k_3600[0],
&state_buffer_16k_3600[0], block_size);

arm_fir_init_q15(&fir_22k_3600, NUM_COEFF_22k_3600,&fir_coeffs_22k_3600[0],
&state_buffer_22k_3600[0], block_size);

arm_fir_init_q15(&fir_44k_3600, NUM_COEFF_44k_3600,(q15_t *)&fir_coeffs_44k_3600[0],
&state_buffer_44k_3600[0], block_size);

arm_fir_init_q15(&fir_48k_3600, NUM_COEFF_48k_3600,&fir_coeffs_48k_3600[0],
&state_buffer_48k_3600[0], block_size);

```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```
///high pass

arm_fir_init_q15(&fir_8k_100, NUM_COEFF_8K_100,(q15_t *) &fir_coeffs_8k_100[0],
&state_buffer_8k_100[0], block_size);

arm_fir_init_q15(&fir_16k_100, NUM_COEFF_16k_100,(q15_t *) &fir_coeffs_16k_100[0],
&state_buffer_16k_100[0], block_size);

arm_fir_init_q15(&fir_22k_100, NUM_COEFF_22k_100, &fir_coeffs_22k_100[0],
&state_buffer_22k_100[0], block_size);

arm_fir_init_q15(&fir_44k_100, NUM_COEFF_44k_100,(q15_t *) &fir_coeffs_44k_100[0],
&state_buffer_44k_100[0], block_size);

arm_fir_init_q15(&fir_48k_100, NUM_COEFF_48k_100, &fir_coeffs_48k_100[0],
&state_buffer_48k_100[0], block_size);


///band pass

arm_fir_init_q15(&fir_8k_100_3500, NUM_COEFF_8k_100_3500,(q15_t *)
&fir_coeffs_8k_100_3500[0], &state_buffer_8k_100_3500[0], block_size);

arm_fir_init_q15(&fir_16k_200_3500, NUM_COEFF_16k_200_3500,(q15_t *)
&fir_coeffs_16k_100_3500[0], &state_buffer_16k_100_3500[0], block_size);

arm_fir_init_q15(&fir_22k_100_3500, NUM_COEFF_22k_100_3500,(q15_t *)
&fir_coeffs_22k_100_3500[0], &state_buffer_22k_100_3500[0], block_size);

arm_fir_init_q15(&fir_44k_200_3500, NUM_COEFF_44k_100_3500,(q15_t *)
&fir_coeffs_44k_100_3500[0], &state_buffer_44k_100_3500[0], block_size);

arm_fir_init_q15(&fir_48k_200_3500, NUM_COEFF_48k_200_3500,(q15_t *)
&fir_coeffs_48k_200_3500[0], &state_buffer_48k_200_3500[0], block_size);


///band stop

arm_fir_init_q15(&fir_8k_40_170, NUM_COEFF_8k_35_135,(q15_t *)
&fir_coeffs_8k_40_170[0], &state_buffer_8k_35_135[0], block_size);

arm_fir_init_q15(&fir_16k_35_135, NUM_COEFF_16k_35_135,(q15_t *)
&fir_coeffs_16k_40_170[0], &state_buffer_16k_35_135[0], block_size);
```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```

    arm_fir_init_q15(&fir_22k_35_135, NUM_COEFF_22k_35_135, (q15_t *)
&fir_coefs_22k_35_135[0], &state_buffer_22k_35_135[0], block_size);

    arm_fir_init_q15(&fir_44k_40_170, NUM_COEFF_44k_40_170, (q15_t *)
&fir_coefs_44k_40_170[0], &state_buffer_44k_40_170[0], block_size);

    arm_fir_init_q15(&fir_48k_40_170, NUM_COEFF_48k_40_170, (q15_t *)
&fir_coefs_48k_40_170[0], &state_buffer_48k_35_135[0], block_size);

/*-----
* seteo antes la primera prefuencia de muestreo para arrancar con la primera frecuencia de
muestreo

* y inicializo el timer

-----*/

count = 0;

PIT_SetTimerPeriod(PIT_PERIPHERAL, PIT_CHANNEL_0, (uint32_t) * direc);

PIT_StartTimer(PIT_PERIPHERAL, PIT_CHANNEL_0);

enable_led = true; //habilito el led del primer periodo

i = 0;

printf("Seleccione el tipo de filtro\n");

printf("1: low pass\n");

printf("2: high pass\n");

printf("3: band pass\n");

printf("4: band stop\n");

seleccion = GETCHAR();

while(1) {

/*-----

* hasta que no cambie de frecuencia de muestreo no cambia el color de los led

-----*/

if (enable_led){

```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```
switch(count){  
  
case 0:  
  
    LED_BLUE_ON();  
  
    LED_GREEN_OFF();  
  
    LED_RED_OFF();  
  
    break;  
  
case 1:  
  
    LED_BLUE_OFF();  
  
    LED_GREEN_ON();  
  
    LED_RED_OFF();  
  
    break;  
  
case 2:  
  
    LED_BLUE_OFF();  
  
    LED_GREEN_OFF();  
  
    LED_RED_ON();  
  
    break;  
  
case 3:  
  
    LED_BLUE_ON();  
  
    LED_GREEN_ON();  
  
    LED_RED_OFF();  
  
    break;  
  
case 4:  
  
    LED_BLUE_ON();  
  
    LED_GREEN_OFF();  
  
    LED_RED_ON();
```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```

        break;

default:

    LED_BLUE_OFF();

    LED_GREEN_OFF();

    LED_RED_OFF();

}

enable_led = false;

}

/*-----
* si convierte se guarda el valor en el buffer de entrada y se hace en forma circular
-----*/

if(adc_convert){
    input[i]      = adc_value;
    adc_convert   = false;
    if(fir_enable){
        switch(seleccion){
            case 49: //low pass
                switch(count){
                    case 0: arm_fir_q15(&fir_8k_3600, adc_value_p, output_p, 1);
                        break;

                    case 1: arm_fir_q15(&fir_16k_3600, adc_value_p, output_p, 1);
                        break;

                    case 2: arm_fir_q15(&fir_22k_3600, adc_value_p, output_p, 1);
                        break;

                    case 3: arm_fir_q15(&fir_44k_3600, adc_value_p, output_p, 1);

```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```

        break;

        case 4: arm_fir_q15(&fir_48k_3600, adc_value_p, output_p, 1);

        break;

    }

    break;

case 50:

    switch(count){

        case 0: arm_fir_q15(&fir_8k_100, adc_value_p, output_p, 1);

            break;

        case 1: arm_fir_q15(&fir_16k_100, adc_value_p, output_p, 1);

            break;

        case 2: arm_fir_q15(&fir_22k_100, adc_value_p, output_p, 1);

            break;

        case 3: arm_fir_q15(&fir_44k_100, adc_value_p, output_p, 1);

            break;

        case 4: arm_fir_q15(&fir_48k_100, adc_value_p, output_p, 1);

            break;

    }

    break;

case 51:

    switch(count){

        case 0: arm_fir_q15(&fir_8k_100_3500, adc_value_p, output_p,
1);

            break;

        case 1: arm_fir_q15(&fir_16k_200_3500, adc_value_p,
output_p, 1);

```


Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```

                                break;
                                case 2: arm_fir_q15(&fir_22k_100_3500, adc_value_p,
output_p, 1);

                                break;
                                case 3: arm_fir_q15(&fir_44k_200_3500, adc_value_p,
output_p, 1);

                                break;
                                case 4: arm_fir_q15(&fir_48k_200_3500, adc_value_p,
output_p, 1);

                                break;
                                }
                                break;
                                case 52:
                                switch(count){
                                case 0: arm_fir_q15(&fir_8k_40_170, adc_value_p, output_p,
1);

                                break;
                                case 1: arm_fir_q15(&fir_16k_35_135, adc_value_p, output_p,
1);

                                break;
                                case 2: arm_fir_q15(&fir_22k_35_135, adc_value_p, output_p,
1);

                                break;
                                case 3: arm_fir_q15(&fir_44k_40_170, adc_value_p, output_p,
1);

                                break;
                                case 4: arm_fir_q15(&fir_48k_40_170, adc_value_p, output_p,
1);

```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```

        break;

    }

    break;

}

/* TODO: Prueba de tiempos de filtrados y adc */
//      t_adc_3= (t_adc_1-t_adc_2);
//      t_adc_4= t_adc_3/PIT_CHANNEL_1_TICKS;
//
//      //uint32_t interrupciones = DisableGlobalIRQ();
//      PIT_StartTimer(PIT_PERIPHERAL, PIT_CHANNEL_1);
//
//      t = PIT_GetCurrentTimerCount(PIT_PERIPHERAL,PIT_CHANNEL_1);
//
//      arm_fir_q15(&fir_44k_200_3500, adc_value_p, output_p, 1);
//
//      t1 = PIT_GetCurrentTimerCount(PIT_PERIPHERAL,PIT_CHANNEL_1);
//      t2= (t-t1);
//      t3= t2/PIT_CHANNEL_1_TICKS;
//      PIT_StopTimer(PIT_PERIPHERAL, PIT_CHANNEL_1);
//      t_total= t3 + t_adc_4;
//      //EnableGlobalIRQ(interrupciones);
}

DAC_SetBufferValue(DAC0,0U, *output_p+1250u); //al obtener una salida en
q15 y como podria arrojar valores negativos le sumo un valor fijo para no tener una salida saturada

i++;

output_p++;

```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```

        input_p++;

        if(input_p==fin_input) input_p = &input[0];

        /*-----
        * reinicio a la primera posicion si ya recorrio el buffer circular de salida
        -----*/

        if(output_p==fin_output || output_p==fin_input) {
            if(fir_enable)output_p = &output[0];
            else output_p=&input[0];
        }

        if (i>=511) {
            i=0;
        }

    }

    return 0 ;

}

/* -----
* notas
* 1.121ms tiempo que tarda en adquirir datos filtrar
*
-----*/

///low pass
//          arm_fir_q15(&fir_8k_3600, adc_value_p, output_p, 1);
//          arm_fir_q15(&fir_16k_3600, adc_value_p, output_p, 1);

```

Facultad de Ciencia Exactas, Físicas y Naturales

Procesamiento digital de señales

```
//          arm_fir_q15(&fir_22k_3600, adc_value_p, output_p,1);
//          arm_fir_q15(&fir_44k_3600, adc_value_p, output_p, 1);
//          arm_fir_q15(&fir_48k_3600, adc_value_p, output_p, 1);

///high pass
//          arm_fir_q15(&fir_8k_100, adc_value_p, output_p, 1);
//          arm_fir_q15(&fir_16k_100, adc_value_p, output_p, 1);
//          arm_fir_q15(&fir_22k_100, adc_value_p, output_p, 1);
//          arm_fir_q15(&fir_44k_100, adc_value_p, output_p, 1);
//          arm_fir_q15(&fir_48k_100, adc_value_p, output_p, 1);

///band pass
//          arm_fir_q15(&fir_8k_100_3500, adc_value_p, output_p, 1);
//          arm_fir_q15(&fir_16k_200_3500, adc_value_p, output_p, 1);
//          arm_fir_q15(&fir_22k_100_3500, adc_value_p, output_p, 1);
//          arm_fir_q15(&fir_44k_200_3500, adc_value_p, output_p, 1);
//          arm_fir_q15(&fir_48k_200_3500, adc_value_p, output_p, 1);

///band stop
//          arm_fir_q15(&fir_8k_40_170, adc_value_p, output_p, 1);
//          arm_fir_q15(&fir_16k_35_135, adc_value_p, output_p, 1);
//          arm_fir_q15(&fir_22k_35_135, adc_value_p, output_p, 1);
//          arm_fir_q15(&fir_44k_40_170, adc_value_p, output_p, 1);
//          arm_fir_q15(&fir_48k_40_170, adc_value_p, output_p, 1);
```

Facultad de Ciencia Exactas, Físicas y Naturales
Procesamiento digital de señales