



Universidad del Azuay

Facultad de Ciencia y Tecnología

Escuela de Ingeniería Electrónica

**“Diseño y Construcción de un Simulador Didáctico de Filtros
Analógicos Activos Basados en Microcontrolador PIC32”**

**Trabajo de Graduación previo a la obtención del título de
Ingeniero Electrónico**

Autores

Xavier Fernando Marín Tenorio

Juan Francisco Solano López

Director

Ing. Leonel Pérez

Cuenca, Ecuador

2010

DEDICATORIA

A Juan y Ritha, mis padres queridos.
por su esfuerzo, perseverancia y cariño,
cuyo ejemplo ha guiado
y guiará mis pasos toda la vida...

A mis buenos amigos y a todos quienes,
con su amistad y su palabra franca, leal y
sincerame impulsaron a llegar a la meta...

Juan Francisco

A mis padres, quienes me han guiado a lo
largo de estos años de estudio...

A mis compañeros de aula, algunos de ellos
me han acompañado y acompañarán
y con quienes se ha compartido muchas
experiencias...

A mi bisabuela, Ana María, quien siempre
estuvo pendiente hasta sus últimos
momentos de vida del "gran día" como
solía decir....

Xavier Fernando

AGRADECIMIENTO

Para Cuenca, su gente, su belleza y su paisaje,
escenario ideal para estudiar y aprender.

Para la Universidad del Azuay, entidad de Educación
Superior donde se crea ciencia y se desarrolla
conocimiento.

Para las autoridades y profesores universitarios,
cuyos saberes y enseñanzas
han dejado una huella imborrable
en nuestra mente y en nuestro pensamiento.

Para nuestros compañeros y amigos,
cuyas muestras de afecto y amistad
fueron parte fundamental de nuestra formación.

Para nuestro director de tesis,
Leonel Pérez,
distinguido Maestro Cubano
dilecto compañero y amigo...

Juan Francisco.

Xavier Fernando.

RESUMEN

Esta propuesta académica presenta un equipo que tendrá la facilidad de seleccionar automáticamente Filtros Pasa Alto, Pasa Bajo, Pasa Banda, Elimina Banda con Aproximación de Bessel, Butterworth y Tschebyshev y un selector de orden según se necesite.

Incluirá un teclado matricial para ingreso de datos y una pantalla LCD, para visualizar toda la información necesaria, en ella aparecerán valores de: frecuencias, resistencias, condensadores.

Para la realización de este trabajo se han utilizado métodos: investigativo, explicativo y descriptivo, también un software de programación basado en lenguaje C, así como diferentes recursos bibliográficos, teniendo como principal fuente la internet.

ABSTRACT

This academic proposal presents a device will can automatically select the type of Filter: Low-Pass, High-Pass, Band-Pass, Band-Stop with Approximation of Bessel, Butterworth and Tschhebyshev and order selector as needed.

Will include a LCD display to display all necessary information to the operator, it will appear values of: frequencies, resistors, capacitors. These data and their selection it will be done through a matrix keyboard.

Investigative, explanatory and descriptive methods have been used for the investigation of this work; we have been used C language-based programming software as well as different bibliographic, resources taking from internet as a primary source.

Índice de Contenidos

Dedicatoria.....	i
Agradecimientos.....	ii
Resumen.....	iii
Abstract.....	iv
Índice de Contenidos.....	v
Índice de Cuadros e Ilustraciones.....	viii
Índice de Anexos.....	xvi
INTRODUCCIÓN.....	1
CAPÍTULO 1: FILTROS ANALÓGICOS ACTIVOS.	
1.1 Introducción.....	4
1.2 Parámetros de diseño.....	6
1.2.1 Parámetros de diseño de un filtro en el dominio de la frecuencia.....	7
1.2.2 Parámetros de diseño de un filtro en el dominio del tiempo.....	9
1.3 Proceso de aproximación para el diseño de un filtro	10
1.3.1 Filtro de Butterworth.....	10
1.3.2 Filtro de Tschebychev.....	17
1.3.3 Filtro de Bessel.....	25
1.3.4 Cuadro comparativo entre los Aproximantes de Butterworth, Tschebychev y Bessel.....	28
1.4 Proceso de realización para el diseño de un filtro.....	29
1.4.1 Realización de un filtro Pasa Bajo.....	31
1.4.2 Realización de un filtro Pasa Alto.....	36
1.4.3 Realización de un filtro Pasa Banda.....	38
1.4.4 Realización de un filtro Elimina Banda.....	41

1.5 Conclusiones.....	43
-----------------------	----

CAPÍTULO 2: IMPLEMENTACIÓN DE HARDWARE.

2.1 Introducción.....	46
2.2 Estructura básica de un Microcontrolador.....	46
2.3 Arquitecturas.....	50
2.4 Familias de Microcontroladores.....	51
2.5 Microcontroladores PIC.....	52
2.5.1 Características principales.....	53
2.6 Microcontrolador Microchip PIC16F648A.....	53
2.6.1 Características generales.....	53
2.6.2 Esquema de bloques.....	54
2.7 Microcontrolador Microchip PIC32MX360F512F.....	54
2.7.1 Características generales.....	55
2.7.2 Esquema de bloques.....	55
2.8 Generador de Funciones.....	56
2.9 Conclusiones.....	67

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE DE PROGRAMACIÓN.

3.1 Introducción.....	68
3.2 Subrutina de programación para atención a la interrupción.....	71
3.3 Subrutina de programación para escritura de un comando, parámetro, o dato en el controlador del LCD.....	73
3.4 Subrutina de programación para la inicialización del LCD.....	76
3.5 Subrutina de programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden.....	79

3.6 Subrutina de programación para la selección de toda la información del programa general, mediante el teclado matricial y su visualización en la pantalla LCD.....	96
3.7 Conclusiones.....	115
Conclusiones Generales.....	117
Recomendaciones.....	119
Bibliografía.....	120
Anexos.....	123

Índice de Cuadros e Ilustraciones

Tabla 1.1:	Cuadro comparativo entre los Aproximantes de Butterworth, Tschebychev y Bessel.....	28
Tabla 2.1:	Fabricantes de Microcontroladores y sus respectivas familias.....	52
Tabla 2.2:	Rangos de frecuencia para señales senoidal, triangular y cuadrada en función de C_5 , R_4 Y VR_4	62
Tabla 2.3:	Rangos de frecuencia para señales diente de sierra y tren de pulsos en función de C_5 , R_4 , VR_4 y VR_5	63
Tabla 2.4:	Señales de salida en función de los switchs analógicos 4016B.....	64
Tabla A.1:	Polinomios de Butterworth.....	124
Tabla A.2:	Polinomios de Butterworth $B(s)$ para distintos valores de n	124
Tabla B.1:	Polinomios de Tschebychev para los distintos valores de n	125
Tabla C.1:	Polinomios de Bessel de orden n y ecuación de recurrencia.....	126
Tabla D.1:	Polinomios de Bessel factorizados.....	127
Tabla E.1:	Nomenclatura de los coeficientes para filtros tipo Sallen Key.....	128
Tabla F.1:	Coeficientes con aproximación de Butterworth.....	129
Tabla G.1:	Coeficientes con aproximación Tschebychev a 1-dB de rizado.....	130
Tabla G.2:	Coeficientes con aproximación Tschebychev a 2-dB de rizado.....	131
Tabla G.3:	Coeficientes con aproximación Tschebychev a 3-dB de rizado.....	131
Tabla H.1:	Coeficientes con aproximación de Bessel.....	132
Figura 1.1.a:	Filtro Pasa Bajo.....	5
Figura 1.1.b:	Filtro Pasa Alto.....	5
Figura 1.1.c:	Filtro Pasa Banda.....	5
Figura 1.1.d:	Filtro Elimina Banda.....	5
Figura 1.2:	Diagrama de bloques con los pasos a seguir para diseñar un Filtro Activo.....	7

Figura 1.3:	Parámetros para el diseño de un filtro en el dominio de la frecuencia.....	8
Figura 1.4:	Parámetros para el diseño de un filtro en el dominio del tiempo.....	9
Figura 1.5:	Procesos para la obtención de un filtro.....	10
Figura 1.6:	Respuesta de frecuencia y respuesta al escalón Para un filtro de Butterworth.....	11
Figura 1.7.a:	Distribución de polos para valores de $n=4$	14
Figura 1.7.b:	Distribución de polos para valores de $n=5$	14
Figura 1.8.a:	Especificaciones de partida para la obtención de un filtro.....	15
Figura 1.8.b:	Especificaciones normalizadas para el filtro de la figura 1.8(a).....	16
Figura 1.9:	Respuesta de frecuencia y respuesta al escalón para un filtro de Tschebychev.....	18
Figura 1.10.a:	Aproximante de Tschebychev tipo I.....	19
Figura 1.10.b:	Aproximante de Tschebychev tipo II.....	19
Figura 1.11:	Polos Tschebychev.....	22
Figura 1.12:	Especificaciones iniciales para la obtención de un filtro de Tschebychev.....	23
Figura 1.13:	Valor de las coordenadas de uno de los Polos de Tschebychev.....	24
Figura 1.14:	Respuesta de frecuencia y respuesta al escalón para un filtro de Bessel.....	25
Figura 1.15:	Comportamiento de de los Aproximantes de Butterworth, Tschebychev y Bessel.....	29
Figura 1.16:	Realización de un filtro utilizando estructura Sallen Key.....	29
Figura 1.17:	Realización de un filtro utilizando una estructura Rauch.....	30
Figura 1.18:	Ejemplo de una fuente de tensión controlada por voltaje (VCVS).....	31
Figura 1.19.a:	Filtro pasa bajo de primer orden no inversor.....	32
Figura 1.19.b:	Filtro Pasa bajo de primer orden inversor.....	32
Figura 1.20.a:	Filtro Pasa bajo no inversor de segundo orden con estructura Sallen Key.....	33

Figura 1.20.b: Filtro Pasa Bajo inversor de segundo orden con estructura Sallen Key.....	33
Figura 1.21.a: Filtro Pasa Bajo inversor de segundo orden con estructura Sallen Key.....	34
Figura 1.21.b: Filtro Pasa Bajo de segundo orden.....	35
Figura 1.22.a: Diagrama de bloques para el diseño de un filtro de orden superior.....	36
Figura 1.22.b: Diseño final para un filtro pasa bajo de Butterworth de orden 5 con frecuencia de corte de 50Khz.....	36
Figura 1.23.a: Filtro Pasa Bajo de primer orden con estructura Sallen Key.....	37
Figura 1.23.b: Filtro Pasa Alto de primer orden con estructura Salen Key.....	37
Figura 1.24: Esquema de un Filtro Pasa Alto de segundo orden.....	38
Figura 1.25: Esquema de un filtro Pasa Banda con estructura Sallen Key.....	40
Figura 1.26: Esquema de un filtro Elimina Banda.....	42
Figura 1.27: Ubicación de f_m y B	42
Figura 1.28: Esquema de un filtro Elimina Banda doble T.....	43
Figura 2.1: Esquema de un Microcontrolador PIC.....	47
Figura 2.2: Esquema de bloques general de un Microcontrolador PIC.....	48
Figura 2.3: Conexión entre un Microcontrolador y un periférico a través de un puerto de entrada y salida.....	49
Figura 2.4.a: Arquitectura Von Neumann.....	51
Figura 2.4.b: Arquitectura Harvard.....	51
Figura 2.5: Esquema de bloques de un Microcontrolador PIC16F648A.....	54
Figura 2.6: Esquema de bloques de un Microcontrolador PIC32MX3XX.....	56
Figura 2.7: Diagrama de un generador de funciones diseñado para el Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.....	57
Figura 2.8: Amplitud de Salida en función de VR1.....	58

Figura 2.9:	Partidor de tensión formado por R1 y R2.....	58
Figura 2.10:	Transferencia de corriente al colector en función de la corriente directa I_F y de la temperatura ambiente.....	65
Figura 2.11:	Circuito para el cálculo de R6 y R7.....	65
Figura 3.1:	Subrutina de programación para atención a la interrupción provocada por el módulo "Change Notice.".....	71
Figura 3.2:	Subrutina de programación para escritura de un comando, parámetro o dato en el controlador del LCD.....	74
Figura 3.3:	Subrutina de programación para la inicialización del LCD.....	77
Figura 3.4.a:	Subrutina de programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden.....	80
Figura 3.4.b:	Subrutina de programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden.....	81
Figura 3.4.c:	Subrutina de programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden.....	82
Figura 3.4.d:	Subrutina de programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden.....	83
Figura 3.4.e:	Subrutina de programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden.....	84
Figura 3.4.f:	Subrutina de programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden.....	85

Figura 3.5.a:	Subrutina de programación para la selección de toda la información del programa general, mediante el teclado matricial y su visualización en la pantalla LCD SH320240A (TaskTecladoMatricial).....	98
Figura 3.5.b:	Subrutina de programación para la selección de toda la información del programa general, mediante el teclado matricial y su visualización en la pantalla LCD SH320240A (TaskTecladoMatricial).....	99
Figura 3.5.c:	Subrutina de programación para la selección de toda la información del programa general, mediante el teclado matricial y su visualización en la pantalla LCD SH320240A (TaskTecladoMatricial).....	100
Figura 3.5.d:	Subrutina de programación para la selección de toda la información del programa general, mediante el teclado matricial y su visualización en la pantalla LCD SH320240A (TaskTecladoMatricial).....	101
Figura 3.5.e:	Subrutina de programación para la selección de toda la información del programa general, mediante el teclado matricial y su visualización en la pantalla LCD SH320240A (TaskTecladoMatricial).....	102
Figura 3.5.f:	Subrutina de programación para la selección de toda la información del programa general, mediante el teclado matricial y su visualización en la pantalla LCD SH320240A (TaskTecladoMatricial).....	103
Figura A.1:	Diagrama a bloques del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.....	134
Figura B.1:	Circuito para fuente de alimentación del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.....	136

Figura C.1:	Circuito para conexiones del PIC32MX360F512L del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.....	138
Figura D.1:	Diseño de circuito para el filtro Pasa Bajo del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32....	140
Figura E.1:	Diseño de circuito para el filtro Pasa Alto del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32...	142
Figura F.1:	Diseño de circuito para el filtro Pasa Banda del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32....	144
Figura G.1:	Diseño de circuito para el filtro Elimina Banda del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.....	146
Figura H.1:	Diseño de circuito para la ganancia de la señal filtrada del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.....	148
Figura I.1:	Diseño de circuito para la comunicación de datos seriales del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.....	150
Figura J.1:	Conexión para el interface usuario máquina del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.....	152
Figura K.1:	Diseño de circuito para el frecuencímetro multicanal del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.....	154

Figura L.1:	Conexión para el PIC16F648A del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.....	156
Figura M.1:	Diseño de circuito para el generador de funciones de señal pura del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.....	158
Figura N.1:	Diseño de circuito para el generador de funciones de señal de ruido del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.....	160
Figura O.1:	Diseño de circuito de señales acondicionadas para el ADC del PIC32MX360F512L del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.....	162
Diagrama 3.1:	Diagrama de bloques principal para el desarrollo del software para el Simulador Didáctico de Filtros Analógicos Activos Basados en Microcontrolador PIC32.....	70
Diagrama 3.2:	Subrutina de atención a la interrupción provocada por el módulo "Change Notice".....	72
Diagrama 3.3:	Subrutina para escritura de un comando, parámetro o dato en el controlador del LCD.....	75
Diagrama 3.4:	Subrutina de programación para la inicialización del LCD.....	78
Diagrama 3.5.a:	Cálculo de elementos pasivos para un filtro Pasa Bajo de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden (Primera Parte).....	86
Diagrama 3.5.b:	Cálculo de elementos pasivos para un filtro Pasa Bajo de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden (Segunda Parte).....	87
Diagrama 3.5.c:	Cálculo de elementos pasivos para un filtro Pasa Alto de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden (Primera Parte).....	88
Diagrama 3.5.d:	Cálculo de elementos pasivos para un filtro Pasa Alto de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden (Segunda Parte).....	89

Diagrama 3.5.e: Cálculo de elementos pasivos para un filtro Pasa Banda de 2 ^{do} , 4, 6 ^{to} y 8 ^{vo} orden (Primera Parte).....	90
Diagrama 3.5.f: Cálculo de elementos pasivos para un filtro Pasa Banda de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden (Segunda Parte).....	91
Diagrama 3.5.g: Cálculo de elementos pasivos para un filtro Elimina Banda de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden (Primera Parte).....	92
Diagrama 3.5.h: Cálculo de elementos pasivos para un filtro Elimina Banda de 2 ^{do} , 4 ^{to} , 6 y 8 ^{vo} orden (Segunda Parte).....	93
Diagrama 3.5.i: Cálculo de elementos pasivos para un filtro Pasa Todo de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden (Primera Parte).....	94
Diagrama 3.5.j: Cálculo de elementos pasivos para un filtro Pasa Todo de 2 ^{do} , 4 ^{to} , 6 ^{to} y 8 ^{vo} orden (Segunda Parte).....	95
Diagrama 3.6.a: Codificación del teclado matricial.....	104
Diagrama 3.6.b: Selección de la tecla presionada.....	105
Diagrama 3.6.c: Selección del tipo de filtro.....	106
Diagrama 3.6.d: Tipo de aproximación.....	107
Diagrama 3.6.e: Selección del orden del filtro.....	108
Diagrama 3.6.f: Frecuencia de corte baja.....	109
Diagrama 3.6.g: Frecuencia de corte alta.....	110
Diagrama 3.6.h: Selección del tipo de señal.....	111
Diagrama 3.6.i: Selección del tipo de ruido.....	112
Diagrama 3.6.j: Señales y ruido.....	113
Diagrama 3.6.k: Elementos pasivos.....	113
Diagrama 3.6.l: Habilitación del filtro.....	114
Diagrama 3.6.j: Adecuación de la señal y ruido.....	114

Índice de Anexos

CAPÍTULO 1.

Anexo A: Polinomios de Butterworth $B(s)$ para distintos valores de n	124
Anexo B: Polinomios de Tschebychev para los distintos valores de n	125
Anexo C: Polinomios de Bessel de orden n y ecuación de recurrencia....	126
Anexo D: Algunos polinomios de Bessel factorizados.....	127
Anexo E: Nomenclatura de los coeficientes para filtros tipo Sallen Key....	128
Anexo F: Tabla de coeficientes para filtros Sallen Key con aproximación de Butterworth.....	129
Anexo G: Tabla de coeficientes para filtros Sallen Key con aproximación de Tschebishev.....	130
Anexo H: Tabla de coeficientes para filtros Sallen Key con aproximación de Bessel.....	132

CAPÍTULO 2.

Anexo A: Diagrama de bloques del simulador.....	133
Anexo B: Circuito fuente de alimentación.....	135
Anexo C: Conexiones del PIC32MX360F512L.....	137
Anexo D: Circuito del filtro Pasa Bajo.....	139
Anexo E: Circuito del filtro Pasa Alto.....	141
Anexo F: Circuito del filtro Pasa Banda.....	143
Anexo G: Circuito del filtro Elimina Banda.....	145
Anexo H: Ganancia de la señal filtrada.....	147
Anexo I: Comunicación de datos seriales.....	149
Anexo J: Interface usuario maquina.....	151
Anexo K: Frecuencímetro multicanal.....	153
Anexo L: Conexiones del PIC16F648A.....	155
Anexo M: Generador de funciones de la señal pura.....	157

Anexo N: Generador de funciones de la señal de ruido.....	159
Anexo O: Señales acondicionadas para el ADC del PIC32.....	161

Marín Tenorio Xavier Fernando

Solano López Juan Francisco

Trabajo de Graduación

Ing. Leonel Pérez

Enero del 2010

Diseño y Construcción de un Simulador Didáctico de Filtros Analógicos Activos Basados en Microcontrolador PIC32

Introducción.

Un filtro electrónico es un elemento que discrimina una determinada frecuencia o gama de frecuencias de una señal eléctrica que pasa a través de él, pudiendo modificar tanto su amplitud, como su fase.

Las nuevas tecnologías de la información y la comunicación han empujado a la humanidad a la era de la comunicación universal, han hecho posible la eliminación de las distancias y han contribuido a forjar las sociedades del mañana.

La información más precisa y actual se puede poner a disposición de cualquier persona en la superficie del mundo, a menudo, en tiempo real y en las regiones más apartadas.

Muy pronto la “interactividad” permitirá no solo emitir y recibir información, sino también dialogar y transmitir información y conocimientos sin límite de distancia ni de tiempo de operación.

Esta libre circulación de la imagen y la palabra que prefigura el mundo del mañana, ha transformado tanto las relaciones internacionales como la comprensión del mundo que tienen las personas, constituyéndose en uno de los aceleradores de la mundialización y de la globalización, fenómenos sociales y culturales que, en cierto modo, condicionan la vida y el desarrollo de nuestros pueblos y culturas.

El avance científico y tecnológico, del mundo en que vivimos, es parte fundamental de la era de la comunicación; ha convertido al universo en una aldea global, en donde, casi todos los habitantes, del planeta tierra, tienen acceso a los bienes creados por el hombre.

Cada día se ponen en el mercado nuevos inventos, aparatos y equipos que facilitan la vida, el aprendizaje y, sobre todo, la posibilidad de estar informado y comunicado.

Esta explosión y desarrollo vertiginoso de las comunicaciones y las tecnologías de la información, ha generado una oferta y demanda que se incrementa progresivamente. Las naciones, las ciudades y los pueblos, necesitan estar preparados para:

- La adquisición, generación y procesamiento de señales, para lo cual es necesario el desarrollo de nuevos aparatos, recursos, equipos que, por un lado, estén en condiciones de dar respuesta a aquellas demandas y, por otro, que permitan, a los docentes universitarios, crear situaciones de enseñanza que faciliten a los estudiantes el desarrollo de procesos y experiencias de aprendizaje, como parte del programa de capacitación de los futuros profesionales.
- Por nuestra parte, proponemos la construcción de un equipo electrónico que facilite el desarrollo de prácticas y experiencias orientadas al estudio de diferentes tipos de señales, mediante el uso de filtros electrónicos.
- Nuestro anhelo es entregar a la Facultad, que nos abrió las puertas para nuestra formación, un recurso didáctico que sirva a los futuros estudiantes para la mejor comprensión y aprehensión del tema, motivo de esta tesis de grado.

La Universidad del Azuay, a través de su Facultad de Ciencia y Tecnología, fomenta la investigación, el desarrollo de la creatividad, la elaboración de trabajos finales de utilidad y aplicación práctica, que tengan funcionalidad, que sirvan como herramientas o instrumentos didácticos de aula y que sean el fiel reflejo de lo que la universidad ha logrado enseñar, forjar y dejar en la mente de los estudiantes de cada carrera.

Este recurso didáctico está pensado para el desarrollo de procesos formativos que, sin lugar a dudas, va a ayudar al futuro profesional a familiarizarse con el trabajo y la manipulación de señales basadas en diferentes tipos de filtros. Dicho proyecto estará a disposición en el laboratorio para que el estudiante realice sus prácticas en la (s) materia (s) asignadas para el efecto.

CAPÍTULO I FILTROS ANALÓGICOS ACTIVOS

1.1 Introducción.

Los filtros son usados para dejar pasar solamente las frecuencias que pudieran resultar ser de alguna utilidad y eliminar cualquier tipo de interferencia o ruido ajeno a ellas. Generalmente son utilizados para:

- **El acondicionamiento de señales de entrada**, eliminando ruido en dichas señales y seleccionando sus frecuencias útiles.
- **Digitalización de señales**, fijando el ancho de banda para compresión y transmisión y eliminado así las frecuencias superiores.
- **Acondicionamiento de señal producida**, mejorando la linealidad o el rendimiento, con la eliminación de picos y la supresión de interferencias.

Los filtros electrónicos son circuitos capaces de discriminar frecuencias. Existen diferentes tipos de circuitos que se clasifican por su comportamiento a la salida del mismo, ante una señal a la entrada, o por sus elementos constitutivos.

Para un mejor estudio, los filtros analógicos se clasifican en tres grupos según:

- **El tipo de señal procesada:** Filtros pasivos, filtros de capacidad conmutada y filtros activos.
- **Su aproximación matemática:** Filtro de Butterworth, filtro de Tshebyshev y filtro de Bessel.
- **Su función en frecuencia:** Filtro pasa bajo, filtro pasa alto, filtro pasa banda, filtro elimina banda.

- **Filtro Pasa Bajo:** permite el paso de frecuencias bajas, bloqueando frecuencias mayores a una frecuencia de corte (f_c) determinada.
- **Filtro Pasa Alto:** Impide el paso de las bajas frecuencias hasta una frecuencia de corte (f_c) permitiendo el paso de todas aquellas mayores a esta.
- **Filtro Pasa Banda:** Solo permite el paso de un limitado rango de frecuencias comprendidas entre la frecuencia de corte inferior (f_{ci}) y la frecuencia de corte superior (f_{cs}) e impiden el paso de las frecuencias restantes.
- **Filtro Elimina Banda:** Impide el paso de un rango de frecuencias comprendidas entre dos frecuencias de corte (inferior y superior) y permiten pasar las restantes.

Las figuras 1.1.a, 1.1.b, 1.1.c y 1.1.d muestran los distintos tipos de filtros en función de su frecuencia.

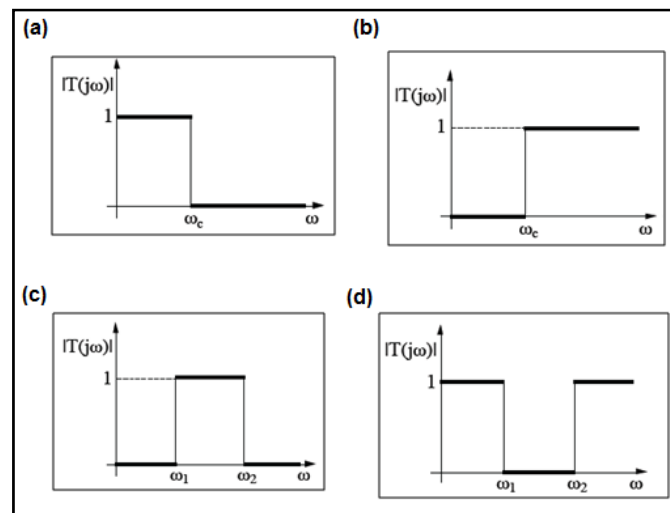


Figura 1.1.a: Filtro Pasa Bajo.

Figura 1.1.b: Filtro Pasa Alto.

Figura 1.1.c: Filtro Pasa Banda.

Figura 1.1.d: Filtro Elimina Banda.

Fuente: MIYARA, Federico; *Filtros Activos*. Argentina. Universidad Nacional de Rosario. 2004 [en línea] [ref. de mayo de 2009].

Disponible en <http://www.fceia.unr.edu.ar/enica3/filtros-t.pdf>

Los Filtros activos emplean dispositivos como transistores o amplificadores operacionales, junto con elementos R L C.

Características.

- Necesitan alimentación.
- No presentan pérdidas en la banda de paso, incluso pueden tener ganancias.
- Pueden distorsionar la señal si la entrada es elevada.
- Pueden ser implementados como integrados.
- Se utilizan en aplicaciones prácticas desde 0,1 Hz hasta 500 Khz.

Ventajas.

- Permiten el trabajo en aplicaciones de baja frecuencia y baja señal.
- Facilitan el encadenamiento de etapas aprovechando la baja impedancia de salida de los operacionales.

Desventajas.

- Ancho de banda limitado al operacional utilizado.
- Necesidad de alimentación.
- Ruido adicional provocado por el operacional.

1.2 Parámetros de diseño.

El funcionamiento de los filtros activos se basa en la interconexión en cascada de sus componentes y el proceso de diseño se inicia calculando un filtro pasa bajo que es considerado como filtro primario.

La figura 1.2 ilustra el procedimiento a seguir al momento de diseñar un filtro activo.

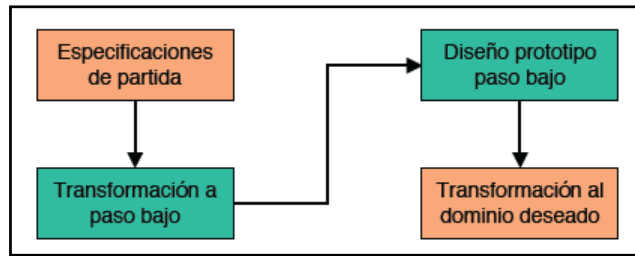


Figura 1.2: Diagrama de bloques con los pasos a seguir para diseñar un filtro activo.

Fuente: MOJÓN OJEA, Artemio. *Síntesis de Circuitos Eléctricos y Electrónicos*. España. Universidad de Vigo, 2005. [en línea] [ref. de 25 de junio de 2009]. Disponible en Web: <http://www.tsc.uvigo.es/BIO/Docencia/SCEE/Clases/Clase_07.pdf>

Se deben considerar dos parámetros para diseñar un filtro:

- **Parámetros en el dominio de frecuencia.**
- **Parámetros en el dominio de tiempo.**

1.2.1 Parámetros de diseño de un filtro en el dominio de la frecuencia.

Los parámetros principales que describen el comportamiento de un filtro en el dominio de la frecuencia son:

- **Banda de paso:** Son las frecuencias que se conservan en el filtro.
- **Banda Atenuada:** Es el intervalo frecuencial donde el módulo de la respuesta del filtro toma un valor constante, generalmente la unidad.
- **Zona de Transición:** Es la diferencia existente entre banda de paso y la banda atenuada.
- **Ganancia en ambas bandas:** Es la amplitud mínima o máxima en la que trabaja el filtro.
- **Nivel de rizado:** Se encuentra en la parte superior del filtro, comprende la zona ubicada entre la amplitud máxima y la banda de paso, haciendo iguales los picos de la señal filtrada.
- **Amplitud Máxima:** Rizado máximo de la banda de paso.

- **Amplitud Mínima:** Atenuación mínima de la banda atenuada (referida a la atenuación máxima de la banda de paso).
- f_p : frecuencia límite de la banda de paso.
- f_s : Frecuencia donde comienza la banda atenuada.
- **Frecuencia Normalizada:** $f_n=f/f_c$ $f_n=f/f_o$
- **Factor de Calidad (Q):** Es la respuesta en frecuencia normalizada, es decir, la relación entre la frecuencia central de paso-banda y las frecuencias de 3dB en un circuito paso-banda. Especifica la eficacia del filtro.

La figura1.3 indica los parámetros necesarios para el diseño de un filtro en el dominio de la frecuencia.

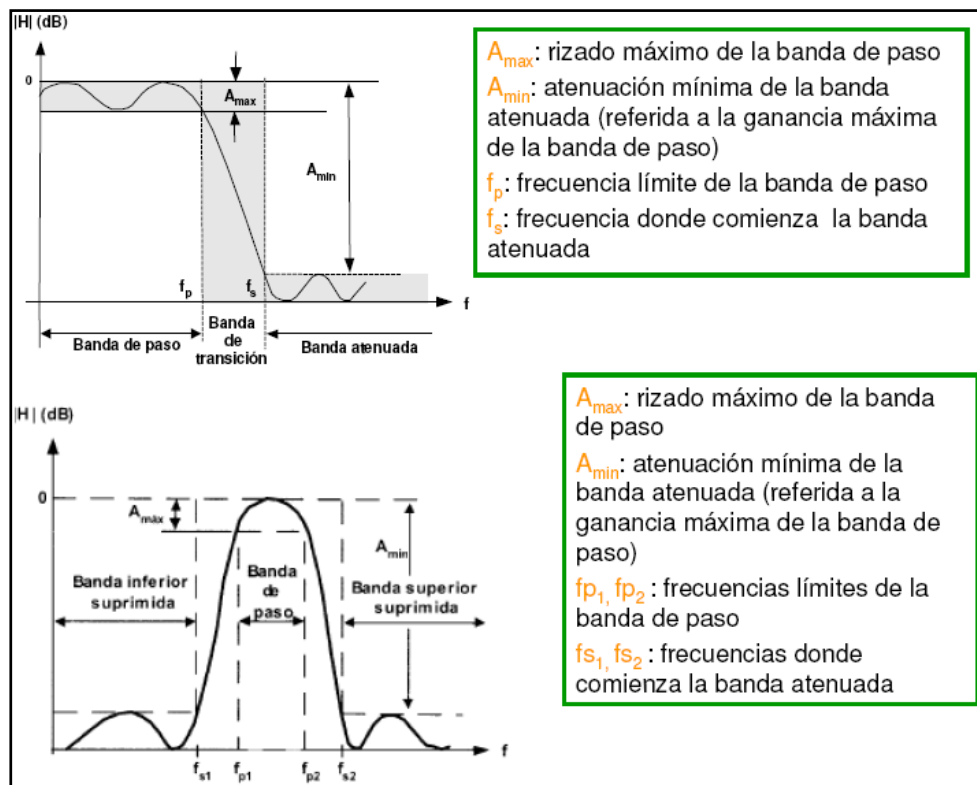


Figura 1.3: Parámetros para el diseño de un filtro en el dominio de la frecuencia.

Fuente: Arias Rodríguez; Julia. *Instrumentación Electrónica* [en línea]. España. Universidad Rey Juan Carlos. 2009 [ref. de mayo de 2009]. Disponible en Web: http://laimbio08.escet.urjc.es/assets/files/docencia/IE/TEMA2_filtros.pdf

1.2.2 Parámetros de diseño de un filtro en el dominio del tiempo.

El diseño de un filtro en el dominio del tiempo se relaciona con la respuesta de un filtro ante una señal de entrada en forma de escalón.

En respuesta a una señal de escalón se pueden medir en la salida los siguientes parámetros:

- **Tiempo de subida:** Es el tiempo requerido para que la respuesta pase del 10 al 90% del valor final. También puede definirse como el tiempo de paso del 5% al 95% o del 0% al 100%.
- **Tiempo de restablecimiento:** Es el tiempo necesario para que la respuesta del sistema esté dentro de un porcentaje del valor final.
- **Sobreoscilación:** Es el valor de pico máximo por unidad. Se suele expresar en porcentaje.
- **Suboscilación:** Es el valor pico mínimo por unidad. Suele expresarse en porcentaje.

La figura 1.4 indica los parámetros que hay que tener en cuenta al momento de diseñar un filtro en el dominio del tiempo.

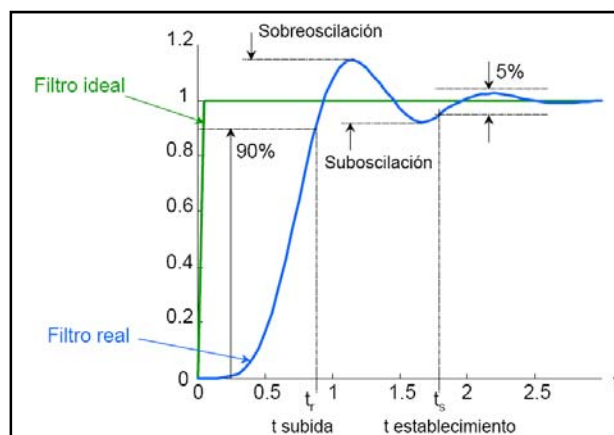


Figura 1.4: Parámetros para el diseño de un filtro en el dominio del tiempo.

Fuente: Arias Rodríguez; Julia. *Instrumentación Electrónica* [en línea]. España. Universidad Rey Juan Carlos. 2009 [ref. de mayo de 2009]. Disponible en Web: http://laimbio08.escet.urjc.es/assets/files/docencia/IE/TEMA2_filtros.pdf

Para la obtención de un filtro hay que realizar dos procesos. Aproximación y realización; como indica la figura 1.5.

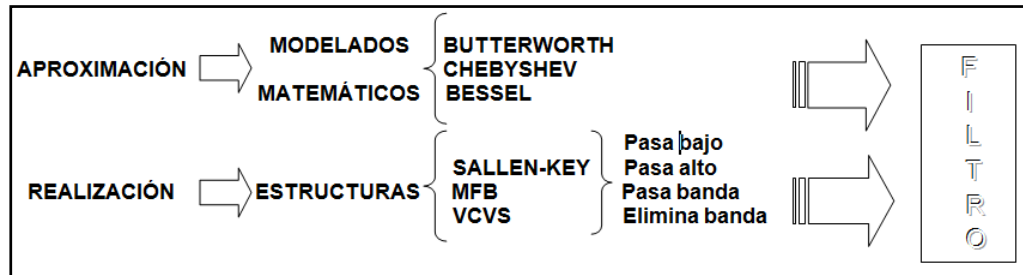


Figura 1.5: Procesos para la obtención de un filtro.

Fuente: Autores.

1.3 Proceso de aproximación para el diseño de un filtro.

El proceso de aproximación se refiere al desarrollo de los modelados matemáticos y tiene que ver básicamente con su aproximación matemática, normalmente dentro del diseño de los diversos tipos de filtros se nota que las aproximaciones aplicadas a estos diseños, generan dificultades al momento de ser comprobadas. Por esta razón existen varios tipos de aproximaciones siendo más conocidas las de Butterworth, Tschebychev y Bessel.

1.3.1 Filtro de Butterworth.

El filtro de Butterworth es diseñado para producir la respuesta más plana posible hasta la frecuencia de corte, es decir, su salida se mantiene constante casi hasta la frecuencia de corte y luego disminuye.

La figura 1.6 indica el comportamiento de un filtro de Butterworth:

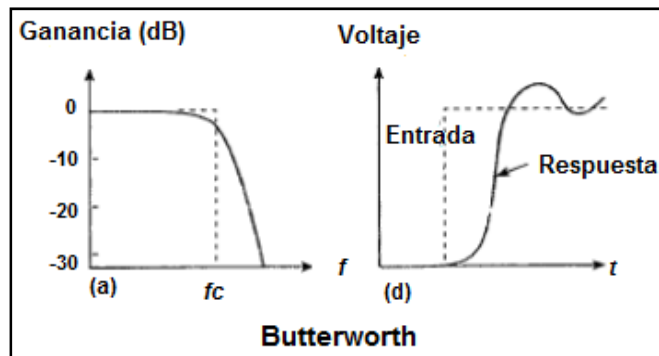


Figura 1.6: Respuesta de frecuencia y respuesta al escalón Para un filtro de Butterworth.

Fuente: *Instrumentación Electrónica* [en línea] 2008 - 2009 [ref. de mayo de 2009]. Disponible en Web:<http://laimbio08.escet.urjc.es/assets/files/docencia/IE/TEMA2_filtros.pdf>

Ventaja.

- Se optimiza una ganancia más plana en la banda de paso.

Desventaja.

- Presenta una región de transición de caída lenta y respuesta no lineal alrededor de la frecuencia de corte (f_c).

Diseño.

Se usan funciones de aproximación que se ajustan a:

$$|H(j\omega)|^2 = \frac{1}{1+D^2(\omega)} \quad (1.1)$$

$$|H(j\omega)| = \frac{1}{\sqrt{1+D^2(\omega)}} \quad (1.2)$$

Con $\omega_x = 1$

$$H(j\omega_x)_{dB} = -10 \log(1 + B_n) \quad (1.3)$$

Por comodidad, se supondrá que:

$$\omega_c = 1 \text{ rad/s}$$

$$D^2(\omega) = B_n \omega^{2n} \quad (1.4)$$

Sustituyendo la ecuación (1.1) por su equivalente en la ecuación (1.4):

$$|H(j\omega)|^2 = \frac{1}{1+B_n\omega^{2n}} \quad (1.5)$$

Suponiendo que: $B_n = 1$ y reemplazándolo en la ecuación (1.5):

$$|H(j\omega)|^2 = \frac{1}{1+\omega^{2n}} \quad (1.6)$$

El módulo al cuadrado se obtiene de acuerdo con las reglas de los números complejos:

$$|H(j\omega)|^2 = H(j\omega)H^*(j\omega) \quad (1.7)$$

Donde:

$$H^*(j\omega) = H(-j\omega)$$

$$|H(j\omega)|^2 = H(j\omega)H(-j\omega) \quad (1.8)$$

Considerando que:

$$j\omega = s \quad (1.9)$$

$$\omega = -js \quad (1.10)$$

Sustituyendo $j\omega$ de la ecuación (1.9) y ω de la ecuación (1.10).

$$H(s)H(-s) = \frac{1}{1+(-js)^{2n}}$$

$$H(s)H(-s) = \frac{1}{1+(-1)^n(s)^{2n}} \quad (1.11)$$

Donde:

Polos = Raíces del denominador

Ceros = Raíces del numerador

n = Orden del filtro

Cálculo del orden.

$$n = \left| \frac{\log \frac{(10^{A_{min}/10} - 1)}{(10^{A_{max}/10} - 1)}}{2 \log \left(\frac{\omega_s}{\omega_p} \right)} \right| \quad (1.12)$$

Localización de polos.

Igualando a cero el denominador en la ecuación (1.11) se obtiene los $2n$ polos de la función:

$$(s)^{2n} = \frac{1}{(-1)^n} \quad (1.13)$$

Si el valor de n es par, las raíces son:

$$s_k^{2n} = -1$$

$$s_k = \sqrt[2n]{-1} \quad (1.14)$$

Que son las $2n$ raíces de -1 y se obtienen a través de:

$$s_k = e^{j\left(\frac{2k\pi+\pi}{2n}\right)} \quad (1.15)$$

Con $k = 0, 1, 2, \dots, 2n-1$

Si el valor de n es impar, se obtiene la expresión:

$$s_k^{2n} = 1$$

$$s_k = \sqrt[2n]{-1} \quad (1.16)$$

Las $2n$ raíces se obtienen según:

$$s_k = e^{j\left(\frac{2k\pi}{2n}\right)} \quad (1.17)$$

Con $k = 0, 1, 2, \dots, 2n-1$

Para n par, como para n impar el módulo de las raíces es la unidad y solo se diferencian en el argumento por lo que el lugar geométrico en el que están situados los polos en el plano complejo¹ normalizado s es una circunferencia de radio unitario. Siempre habrá la misma cantidad de polos en el semiplano derecho que en el izquierdo y son simétricos.

Las figuras 1.7.a y 1.7.b muestran algunos ejemplos de ubicación de polos para valores de n en el plano complejo.

¹ En matemáticas, el **plano complejo** es una forma de visualizar el espacio de los números complejos. Puede entenderse como un plano cartesiano modificado, en el que la parte real está representada eje x y la parte imaginaria en el eje y . El eje x también recibe el nombre de **eje real** y el y **eje imaginario**. En teoría de control, uno de los usos del plano complejo se conoce como el 'plano s '

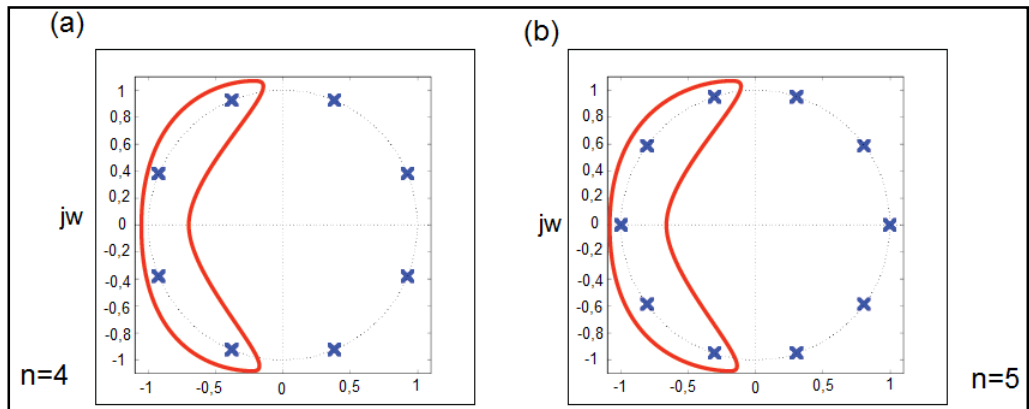


Figura 1.7.a:Distribución de polos para valores de n=4.
Figura 1.7.b: Distribución de polos para valores de n=5.

Fuente: MOJÓN OJEA, Artemio. *Síntesis de Circuitos Eléctricos y Electrónicos*. España. Universidad de Vigo. 2005. [en línea] [ref. de 25 de junio de 2009]. Disponible en Web: <http://www.tsc.uvigo.es/BIO/Docencia/SCEE/Clases/Clase_07.pdf>

Normalización:

Para cada valor de n existirá una distribución de polos fija, pudiendo obtener y tabular los polinomios que corresponden a cada distribución de polos. En las tablas A.1 y A.2 incluidas en el anexo A del capítulo 1; se indican los polinomios de Butterworth $B(s_x)$ para distintos valores de n.

La función de sistema será el inverso de los polinomios de Butterworth.

$$H(s_x) = \frac{1}{B(s_x)} \tag{1.18}$$

Los valores de los polinomios son válidos para $B_n=1$, por lo que la frecuencia de corte es ω_c es exactamente la frecuencia donde el módulo toma el valor de -3dB.

$$|H(j\omega_x)|_{dB} = -20 \log(2) = -3dB \tag{1.19}$$

$$\omega_{x3} = \frac{1}{2^n \sqrt{B_n}} \tag{1.20}$$

$$H(s) = H(s_x) \left| \begin{array}{l} \text{Función de sistema} \\ \text{normalizado } H(s_x) \end{array} \right. \begin{array}{l} s_x = \frac{s}{\omega_3} \end{array} \tag{1.21}$$

Ejemplo 1.1.

Obtener la función del sistema según Butterworth para:

$$n=2 \quad \text{y} \quad B_n=1: \quad B(s)=+\sqrt{2}s_x+1$$

Aplicando la ecuación (1.18) y analizando el anexo A para el valor de $n=2$, la función será:

$$\frac{1}{s_x^2\sqrt{2}s_x+1}$$

La función de sistema normalizado se obtiene aplicando la ecuación (1.21).

$$\frac{\omega_c}{s^2\sqrt{2}\omega_3s+\omega_3^2}$$

Ejemplo 1.2.

Se desea obtener una función de sistema de un filtro que tenga las siguientes especificaciones. Figura 1.8.a.

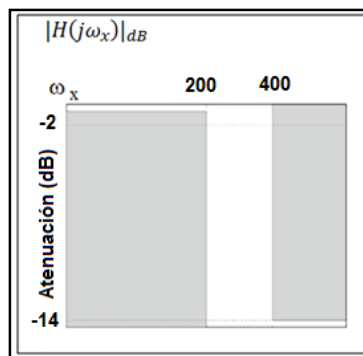


Figura 1.8.a: Especificaciones de partida para la obtención de un filtro.

Fuente: MOJÓN OJEA, Artemio. *Síntesis de Circuitos Eléctricos y Electrónicos*. España. Universidad de Vigo. 2005. [en línea] [ref. de 25 de junio de 2009]. Disponible en Web: <http://www.tsc.uvigo.es/BIO/Docencia/SCEE/Clases/Clase_07.pdf>

Autores.

Se normalizan las especificaciones en frecuencia considerando $\omega_c = 200$ rad/seg. Figura 1.8.b.

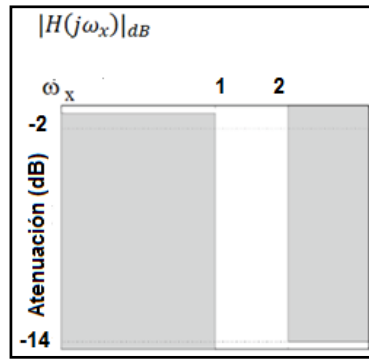


Figura 1.8.b: Especificaciones normalizadas para el filtro de la figura 1.8(a).

Fuente: MOJÓN OJEA, Artemio. *Síntesis de Circuitos Eléctricos y Electrónicos*. España. Universidad de Vigo. 2005. [en línea] [ref. de 25 de junio de 2009]. Disponible en Web: <http://www.tsc.uvigo.es/BIO/Docencia/SCEE/Clases/Clase_07.pdf>

Autores.

Partiendo de la ecuación (1.3).

$$|H(j\omega_x)|_{dB} = -10 \log(1 + B_n \omega_x^{2n})$$

Evaluando en el punto (1, -2) se obtiene el valor de B_n :

$$-2 = -10 \log(1 + B_n)$$

$$B_n = \log^{-1} 0,2 - 1 = 0,58$$

$$\mathbf{B_n = 0,58}$$

Evaluando ahora en el punto (2,-14):

$$-14 = -10 \log [1 + 0,58(2)^{2n}] \quad \Rightarrow \quad 1,4 = \log [1 + 0,58(2)^{2n}]$$

Aplicando algoritmos en ambos miembros y despejando:

$$2^{2n} = \frac{\log^{-1} 1,4 - 1}{0,58} = 41,58$$

Aplicando logaritmo y despejando:

$$n = \frac{\log 41,58}{2 \log 2} = 2,69$$

n debe de ser un número entero, entonces se escoge en $n=3$

$$B(s_x) = (s_x + 1)(s_x^2 + s_x + 1) \quad \Rightarrow \quad H(s_x) = \frac{1}{(s_x + 1)(s_x^2 + s_x + 1)}$$

Para des normalizar:

$$-3 = -10 \log[1 + B_n (\omega_{x3})^2]^{2n}$$

Sustituyendo en la ecuación (1.20):

$$\omega_{x3} = \frac{1}{\sqrt[6]{0,58}} \quad \Rightarrow \quad \frac{1}{(0,58)^{1/6}}$$

Despejando:

$$\log \omega_{x3} = -\frac{1}{6} \log 0,58 \quad \Rightarrow \quad \omega_{x3} = \log^{-1} \left[-\frac{1}{6} \log 0,58 \right]$$

$$\omega_{x3} = 1,095$$

La frecuencia des normalizada de -3dB se obtiene evaluando y despejando en:

$$\omega_x = \frac{\omega}{\omega_c} \tag{1.22}$$

$$\omega_3 = \omega_{x3} \omega_c \quad \Rightarrow \quad 219 \text{ rad/s}$$

Una vez calculado n, cuando $B_n=1$ se obtiene $-3\text{dB}(\omega_3)$ y se des normaliza la función de sistema con respecto a ella usando la ecuación (1.21).

$$H(s) = \frac{1}{\left[\left(\frac{s}{\omega_3}\right)^2 + \frac{s}{\omega_3} + 1\right]}$$

Resolviendo la ecuación de sistema.

$$H(s) = \frac{219^3}{(s+219)(s^2+219s+219^2)}$$

1.3.2 Filtro de Tschebychev.

Este aproximante, consigue una caída de la respuesta en frecuencia más pronunciada en frecuencias bajas, debido a que permiten más rizado que otros filtros en alguna de sus bandas.

La figura 1.9 indica el comportamiento del aproximante de Tschebychev:

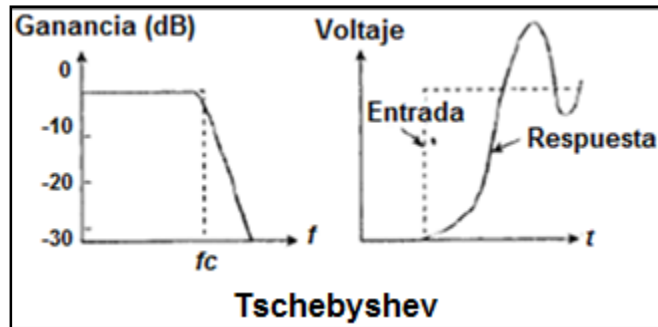


Figura 1.9: Respuesta de frecuencia y respuesta al escalón para un filtro de Tschebychev.

Fuente: *Instrumentación Electrónica* [en línea] 2008 - 2009 [ref. de mayo de 2009]. Disponible en Web: <http://laimbio08.escet.urjc.es/assets/files/docencia/IE/TEMA2_filtros.pdf>

Con el aproximante de Tschebychev se tiene dos clases de filtro:

- **Tipo I:** Se caracteriza por tener:
 - Solamente polos.
 - Rizado constante en la banda de paso.
 - Característica monótona en la banda atenuada.

- **Tipo II:** Se caracteriza por:
 - Tener polos y ceros
 - Presentar rizado constante en la banda de atenuación
 - Tener característica monótona en la banda de paso

Las figuras 1.10.a y 1.10.b muestran a los aproximantes de Tschebychev tipo I y tipo II con sus respectivas características.

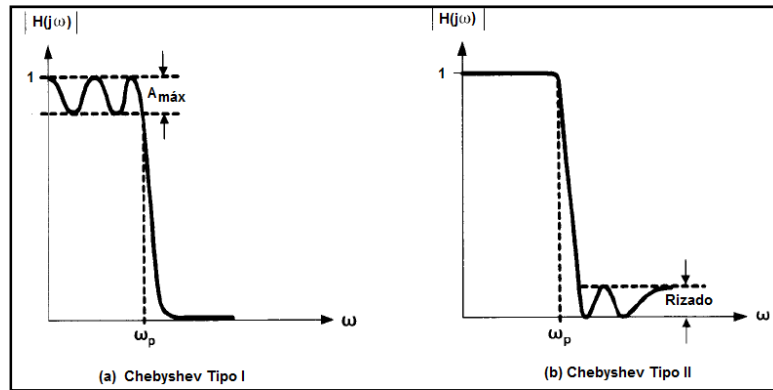


Figura 1.10.a Aproximante de Tschebychev tipo I.

Figura 1.10.b Aproximante de Tschebychev tipo II.

Fuente: *Instrumentación Electrónica* [en línea] 2008 - 2009 [ref. de mayo de 2009]. Disponible en Web: <http://laimbio08.escet.urjc.es/assets/files/docencia/IE/TEMA2_filtros.pdf>

Ventaja².

- Presenta una atenuación más rápida en la zona de transición.

Desventaja³.

- Presenta mayor rizado en la banda de paso.

Diseño⁴:

Se parte de una función que presenta todos los ceros en el infinito y está dada por la ecuación (1.1).

$$|H(j\omega)|^2 = \frac{1}{1+D^2(\omega)}$$

Donde:

$$D^2(\omega) = \varepsilon^2 C_n^2(\omega_x) \quad (1.23)$$

Reemplazando la ecuación (1.23) en la ecuación (1.1).

$$|H(j\omega)|^2 = \frac{1}{1+\varepsilon^2 C_n^2(\omega_x)} \quad (1.24)$$

ε es un parámetro a determinar que estará entre los valores:

$$0 < \varepsilon \leq 1 \quad (1.25)$$

^{2 3 4} Para un filtro de Tschebychev en general.

$C_n \omega_x$ Son los polinomios de Tschebychev de orden n. El valor de n es otro parámetro a determinar.

Los polinomios de Tschebychev están definidos por:

$$C_n(\omega_x) = \cos(n \cos^{-1} \omega_x) \quad \text{para } |\omega_x| \leq 1 \quad (1.26)$$

$$C_n(\omega_x) = \cosh(n \cosh^{-1} \omega_x) \quad \text{para } |\omega_x| > 1 \quad (1.27)$$

Si se sustituye para distintos valores de n, se obtiene la tabla B.1 para los polinomios de Tschebychev indicadas en el anexo B del capítulo 1.

Características generales de los polinomios de Tschebychev.

De las expresiones incluidas en el anexo B se deduce:

$$C_n^2(0) = \begin{cases} \pm 1 & \text{n par} \\ 0 & \text{n impar} \end{cases} \begin{matrix} \Rightarrow \\ \Rightarrow \end{matrix} \begin{matrix} |H(j0)| = \sqrt{\frac{1}{1+\varepsilon^2}} \\ |H(j0)| = 1 \end{matrix} \quad (1.28)$$

$$C_n^2(1) = 1 \quad \Rightarrow \quad |H(j1)| = \sqrt{\frac{1}{1+\varepsilon^2}} \quad (1.30)$$

$$\text{Si: } |\omega_x| \leq 1, C_n(\omega_x) \leq 1 \text{ Independientemente del valor de n.} \quad (1.31)$$

$$|\omega_x| > 1, C_n(\omega_x) \text{ Crece rápidamente con la frecuencia.} \quad (1.32)$$

A medida que aumenta el valor de n, mayor el crecimiento.

Cálculo de la atenuación:

$$A_{dB}(0) = -20 \log |H(j0)| = \begin{cases} 10 \log(1 + \varepsilon^2) & \text{n par} \\ 0 & \text{n impar} \end{cases} \quad (1.33)$$

$$A_{dB}(1) = -20 \log |H(j1)| = 10 \log(1 + \varepsilon^2) \quad (1.35)$$

Donde:

$$\varepsilon = \sqrt{10^{Am_{\acute{a}x}/10} - 1} \quad (1.36)$$

Cálculo del orden:

$$n = \left| \frac{\operatorname{arccosh}\left(\sqrt{\frac{10^{A_{\min}/10} - 1}{\varepsilon^2}}\right)}{\operatorname{arccos} h\left(\frac{\omega_c}{\omega_p}\right)} \right| \quad (1.37)$$

Localización de polos.

La función tendrá $2n$ polos y tendrá una abscisa (parte real) y una ordenada (parte imaginaria). Tanto la parte real como la imaginaria no solo dependen del valor de n , sino que también dependen de ε por lo que no es común encontrarse la posición de los polos. En la práctica se utiliza la computadora para la obtención de los mismos, hallando la abscisa y la ordenada.

Con la posición de los polos se obtiene la función del sistema normalizado $H(s_{xK})$:

$$\sigma_{xK} = \sinh a \sin\left(\frac{2K-1}{n}\right) \frac{\pi}{2} \quad \text{(Parte real)} \quad (1.38)$$

$$\omega_{xK} = \cosh a \cos\left(\frac{2K-1}{n}\right) \frac{\pi}{2} \quad \text{(Parte imaginaria)} \quad (1.39)$$

$$a = \frac{1}{n} \sinh^{-1} \frac{1}{\varepsilon} \quad \text{(Coeficiente de compresión)} \quad (1.40)$$

Con $K = 0, 1, 2, \dots, 2n-1$

$$s_k = \sigma_{xK} + j\omega_{xK} \quad (1.41)$$

Con $K = 1, 2, \dots, 2n$

Reemplazando, elevando al cuadrado y sumando ambas expresiones.

$$\frac{\sigma_k^2}{\sinh^2(a)} + \frac{\omega_k^2}{\cosh^2(a)} = 1 \quad (1.42)$$

La figura 1.11 comprueba que los polos Tschebychev forman una elipse aplicando la ecuación (1.42).

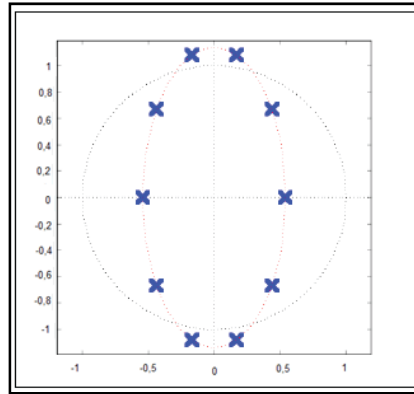


Figura 1.11: Polos Tschebychev.

Fuente: *Síntesis de Circuitos Eléctricos y Electrónicos*. España. Universidad de Vigo. 2005. [en línea]. [ref. de 25 de junio de 2009]. Disponible en Web: <http://www.tsc.uvigo.es/BIO/Docencia/SCEE/Clases/Clase_07.pdf>

Para definir $H(s)$.

$$s_k = -\sinh a \sin\left(\frac{2K-1}{n}\frac{\pi}{2}\right) + j\cosh a \cos\left(\frac{2K-1}{n}\frac{\pi}{2}\right) \quad (1.43)$$

$$H(s) = \frac{K}{(s-s_1)(s-s_2)\dots(s-s_n)} \quad (1.44)$$

$$H(s) = \frac{K}{s^n + b_{n-1}s^{n-1} + \dots + b_1s + b_0} \quad (1.45)$$

$$K_c = \frac{1}{\varepsilon 2^{n-1}} \quad (1.46)$$

$K = b_0$ para n impar

Normalización.

Para obtener la función del sistema en s , sólo hay que evaluar para s_x :

$$s_x = \frac{s}{\omega_c} \quad (1.47)$$

$$H(s) = H(s_x) \Bigg|_{s_x = \frac{s}{\omega_c}} \quad \text{Función del sistema normalizado } H(s_x) \quad (1.48)$$

Ejemplo 1.3

Para las especificaciones de la figura 1.12, obtener la función del sistema por el aproximante de Tschebychev.

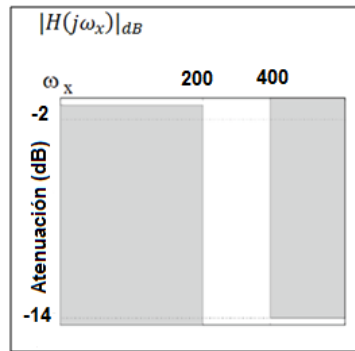


Figura 1.12: Especificaciones iniciales para la obtención de un filtro de Tschebychev.

Fuente: *Síntesis de Circuitos Eléctricos y Electrónicos*. [en línea]. Universidad de Vigo. [ref. de 25 de junio de 2009]. Disponible en Web: <http://www.tsc.uvigo.es/BIO/Docencia/SCEE/Clases/Clase_07.pdf>

Autores.

De la expresión del módulo normalizado en frecuencia se obtiene.

$$|H(j\omega_x)|_{dB} = -10 \log[1 + \varepsilon^2 c_n^2(\omega_x)] \quad (1.49)$$

Evaluando para el punto (1,-2) y aplicando la ecuación (1.36).

$$\varepsilon^2 = 0,58 \quad \Rightarrow \quad 0,76$$

Para el punto (2,-14) debe de cumplirse que la función $|H(\omega)|_{dB}$ sea:

$$|H(\omega)|_{dB} \leq -14$$

Reemplazando esta expresión en la ecuación (1.49).

$$-14 \geq -10 \log [1 + 0,58 c_n^2(2)]$$



$$c_n^2(2) \geq \frac{\log^{-1}(1,4)-1}{0,58} \Rightarrow c_n^2(2) \geq 41,58 \Rightarrow c_n(2) \geq 6,45$$

Pero se tiene que $c_n(2)$ son los polinomios de Tschebychev evaluados para $\omega_x = 2$, el valor de n será el primero que haga cumplir la desigualdad anterior, en este caso:

$$\text{Para } n=1 \quad c_1(2) = 2 < 6,45$$

$$\text{Para } n=2 \quad c_2(2) = 2(2)^2 - 1$$

$$c_2(2) = 7 > 6,45$$

El valor será $n=2$ y el número de polos será de 4, dos en el semiplano izquierdo y dos en el semiplano derecho.

Para hallar la posición de los polos se parte de las ecuaciones (1.38) y (1.39).

$$\sigma_{xK} = \sinh a \sin \left(\frac{2K-1}{n} \frac{\pi}{2} \right)$$

$$\omega_{xK} = \cosh a \cos \left(\frac{2K-1}{n} \frac{\pi}{2} \right)$$

De la ecuación (1.40) se obtendrá el valor de a .

$$a = \frac{\operatorname{arcsenh}(1/\varepsilon)}{n} \quad \Rightarrow \quad a = \frac{\operatorname{arcsenh}(1/0,76)}{2} \quad \Rightarrow \quad a = 0,545$$

Para $K=0$

$$\sigma_{x0} = \sinh 0,545 \operatorname{sen} \frac{\pi}{4} \quad \Rightarrow \quad \sigma_{x0} = 0,4$$

$$\omega_{x0} = \cosh 0,545 \operatorname{cos} \frac{\pi}{4} \quad \Rightarrow \quad \omega_{x0} = 0,8$$

Se obtuvieron las coordenadas de uno de los cuatro polos (figura 1.13), pero al presentar simetría ya se tienen las coordenadas de los cuatro polos.

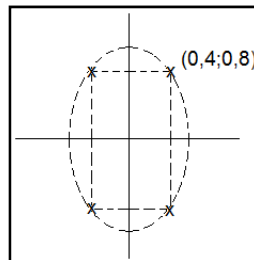


Figura 1.13: Valor de las coordenadas de uno de los Polos de Tscheychev.

Fuente: Autores.

El valor de K se obtiene reemplazando los valores obtenidos en la ecuación (1.46).

$$K_c = \frac{1}{\varepsilon^{2n-1}} \quad \Rightarrow \quad 0,66$$

La función de sistema se encuentra aplicando la ecuación (1.45).

$$H(s_x) = \frac{0,66}{(s_x+0,4+0,8j)(s_x+0,4-0,8j)} \quad \Rightarrow \quad \frac{0,66}{s_x^2+0,4s_x+0,8}$$

La ecuación (1.48) se utiliza para desnormalizar con respecto a la frecuencia de corte de 200 rad/seg .

$$H(s) = \frac{0,66}{\left(\frac{s}{200}\right)^2+0,8\frac{s}{200}+0,8} \quad \Rightarrow \quad \frac{0,66}{s^2+160s+32000}$$

1.3.3 Filtro de Bessel.

Esta clase de aproximante se distingue por tener un retardo constante en todo el ancho de banda deseado. Se caracteriza por ser completamente plano desde la frecuencia cero hasta aproximadamente la frecuencia de corte.

Ventaja.

- Presenta una mejor respuesta a un escalón.

Desventaja.

- Atenuación por encima de la banda de paso más alta.

La figura 1.14 indica el comportamiento del aproximante de Bessel.

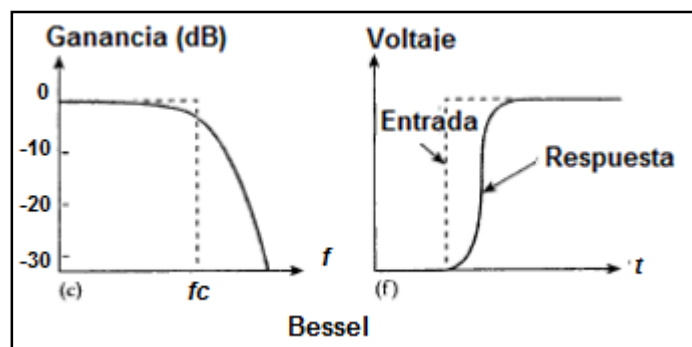


Figura 1.14: Respuesta de frecuencia y respuesta al escalón para un filtro de Bessel.

Diseño.

Se parte de la función:

$$H(s) = \frac{K}{B_n(s)} \quad (1.50)$$

$$K=0,1,2,3,\dots,n$$

$B_n(s)$ Es el polinomio de Bessel de orden n . Estos polinomios se encuentran en la tabla C.1 incluida en el anexo C del capítulo 1.

Un filtro de Bessel está formado por el producto de una o varias funciones cuadráticas del tipo:

$$H(s) = \frac{a_0}{b_2s^2+b_1s+1} \quad (1.51)$$

Es conveniente tener factorizados los polinomios de Bessel para su utilización en el diseño de filtros, la tabla D.1 incluida en el anexo D del capítulo 1 muestra algunos de estos polinomios.

La función retardo viene dada por:

$$\tau(\omega) = \frac{b_1(b_2\omega^2+1)}{b_2^2\omega^4+(b_1^2-2b_2)\omega^2+1} \quad (1.52)$$

Para $n/2$ factores cuadráticos, se tiene un filtro de orden n .

$$\tau(\omega) = \sum_{k=1}^{n/2} \left(\frac{b_1(b_2\omega^2+1)}{b_{2k}^2\omega^4+(b_{1k}^2-2b_{2k})\omega^2+1} \right) \quad (1.53)$$

El retardo en la banda pasante viene dado por:

$$\tau(0) = \sum_{k=1}^{n/2} b_{1k} \quad (1.54)$$

Ejemplo 1.4.

¿Cuál es la función de transferencia de Bessel normalizado a retardo unidad y con ganancia de 2 en la banda pasante?

Partiendo de la ecuación (1.50), se toma el polinomio de Bessel de orden 4 incluido en la tabla C.1 en el anexo C del capítulo 1.

$$H(s) = \frac{210}{s^4 + 10s^3 + 45s^2 + 105s + 105} \Rightarrow \text{Utilizando la función cuadrática:}$$

$$H(s) = \frac{(1,414213)}{(0,109408s^2 + 0,633735s + 1)} \cdot \frac{(1,414213)}{(0,087049s^2 + 0,366265s + 1)}$$

1.3.4 Cuadro comparativo entre los aproximantes de Butterworth, Tschebychev y Bessel.

La tabla 1.1 indica un resumen entre los aproximantes de Butterworth, Tschebychev y Bessel.

Aproximante	Ventajas	Inconvenientes
Butterworth	<p>Respuesta de ganancia plana en la banda de paso.</p> <p>Mejor respuesta ante el escalón que el Tschebychev.</p> <p>Atenuación mayor que Bessel por encima de la banda de paso.</p>	Oscilación y rizado en respuesta a un escalón
Tschebychev Tipo I	Atenuación mayor que Bessel por encima de la banda de paso.	<p>Rizado en la banda de paso.</p> <p>Peor respuesta a un escalón que Butterworth.</p>
Bessel	La mejor respuesta a un escalón de tensión.	Atenuación más lenta que Butterworth por encima de la banda de paso.

Tabla 1.1: Cuadro comparativo entre los aproximantes de Butterworth, Tschebychev y Bessel.

Fuente: *Instrumentación Electrónica* [en línea] 2008 - 2009 [ref. de mayo de 2009]. Disponible en Web: <http://laimbio08.escet.urjc.es/assets/files/docencia/IE/TEMA2_filtros.pdf>

La figura 1.15 indica el comportamiento de los aproximantes de Butterworth, Tschebychev y Bessel.

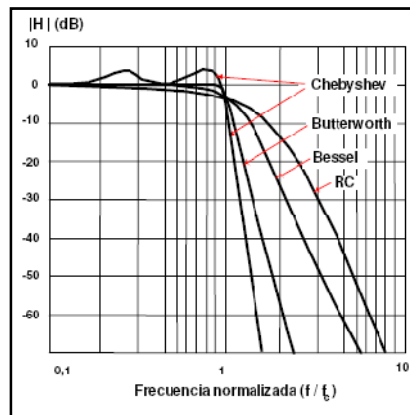


Figura 1.15: Comportamiento de de los aproximantes de Butterworth, Tschebychev y Bessel.

Fuente: *Instrumentación Electrónica* [en línea] 2008 - 2009 [ref. de mayo de 2009]. Disponible en Web: <http://laimbio08.escet.urjc.es/assets/files/docencia/IE/TEMA2_filtros.pdf>

1.4 Proceso de realización para el diseño de un filtro.

La realización en el diseño de filtros tiene que ver con el tipo de estructura a utilizar. Puede ser de tres tipos: Sallen Key, retro alimentación múltiple (estructura Rauch) y fuente de voltaje controlada por voltaje (VCVS).

La estructura Sallen Key es usada particularmente por su simplicidad. El circuito produce un filtro pasa bajo o pasa alto de dos polos usando dos resistencias, dos condensadores y un amplificador. Para obtener un filtro de orden mayor se puede poner en cascada varias etapas. La figura 1.16 indica un ejemplo de realización de un filtro utilizando estructura Sallen Key.

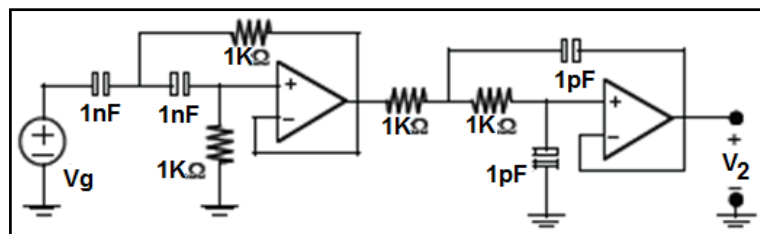


Figura 1.16: Realización de un filtro utilizando estructura Sallen Key.

Fuente: *Síntesis de Circuitos Eléctricos y Electrónicos*. [en línea]. Universidad de Vigo. [ref. de 25 de junio de 2009]. Disponible en Web: <http://www.tsc.uvigo.es/BIO/Docencia/SCEE/Clases/Clase_07.pdf>

La realización mediante retroalimentación múltiple o estructura Rauch se da mediante un filtro pasa banda, sencillo y de buen funcionamiento, para Q_s de bajos a moderados, hasta de 10 aproximadamente. La figura 1.17 muestra un ejemplo de realización mediante una estructura Rauch.

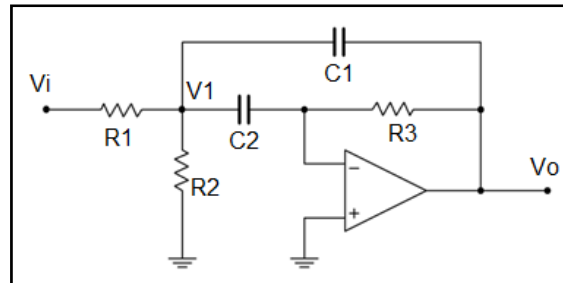


Figura 1.17: Realización de un filtro utilizando una estructura Rauch.

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009].
 Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

La realización de un filtro mediante fuente de voltaje controlada por voltaje (VCVS), se da mediante un dispositivo de dos puertos caracterizado por las siguientes propiedades:

- Impedancia de entrada infinita.
- Impedancia de salida cero.
- Una tensión de salida que es linealmente proporcional a la tensión de entrada.

La ganancia puede ser positiva, en cuyo caso, se dice que la VCVS es no inversora, o negativa, en el cual se dice inversora.

La figura 1.18 muestra un ejemplo de realización mediante una Fuente de tensión controlada por voltaje.

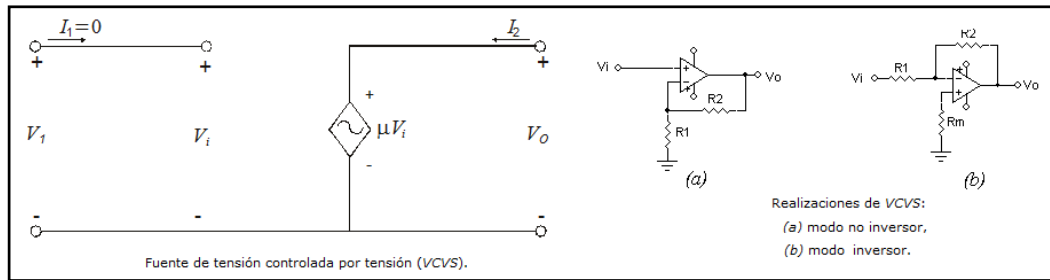


Figura 1.18: Ejemplo de una fuente de tensión controlada por voltaje (VCVS).

Fuente: AVENDAÑO, Luis Enrique. *Realización de Filtros*. [en línea] Universidad Nacional de Colombia Sede Manizales. 2009 [ref. de junio de 2009]. Disponible en Web:

<http://www.virtual.unal.edu.co/cursos/sedes/manizales/4040016/html/cap_5/cap5_1.html#el>

1.4.1 Realización de un filtro Pasa Bajo.

Para la realización de esta clase de filtro se requiere conocer las siguientes especificaciones:

$$\text{PB} \begin{cases} A_1 \leq |H(\omega)| \leq A & 0 \leq \omega \leq 2\pi f_c & (1.55) \\ A_2 \leq |H(\omega)| \leq A_1 & 2\pi f_c < \omega < 2\pi f_s & (1.56) \\ 0 \leq |H(\omega)| \leq A_2 & 2\pi f_s \leq \omega < \infty & (1.57) \end{cases}$$

Donde:

A_1 Rizado superior.

A_2 Rizado inferior.

f_c Frecuencia de corte.

f_s Frecuencia de supresión.

$$f = \frac{1}{2\pi RC} \quad (1.58)$$

$$H(s) = \frac{b_0}{s^2 + b_1 s + b_0} \quad (1.59)$$

Un filtro pasa bajo puede ser de primer orden, segundo orden y orden superior.

Realización de un filtro pasa bajo de primer orden con estructura Sallen Key.

Sallen Key.

Para diseñar un filtro pasa bajo de primer orden se debe de considerar la función de transferencia normalizada de un filtro e identificarla con la expresión matemática del filtro a diseñar.

La figura 1.19.a, indica un filtro pasa bajo no inversor con estructura Sallen Key.

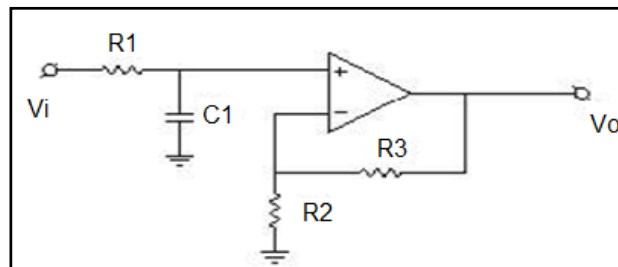


Figura 1.19.a: Filtro pasa bajo de primer orden no inversor.

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009].
 Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

$$F(s) = \frac{1 + \frac{R_2}{R_1}}{1 + \omega R_2 C_S} \quad (1.60)$$

La figura 1.19.b, indica un filtro pasa bajo inversor con estructura Sallen Key.

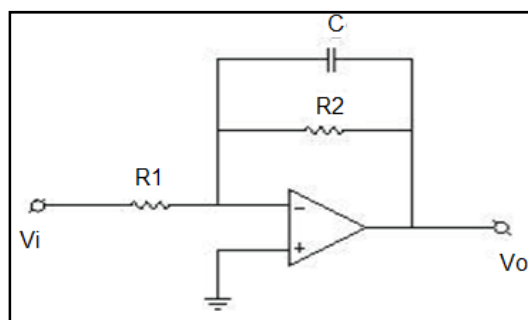


Figura 1.19.b: Filtro Pasa bajo de primer orden inversor.

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009].
 Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

$$F(s) = \frac{\frac{-R_2}{R_1}}{1 + \omega R_2 C_S} \quad (1.61)$$

Realización de un filtro pasa bajo de segundo orden con estructura

Sallen Key.

La Figura 1.20.a, indica un esquema general para un filtro activo pasa bajo no inversor de segundo orden.

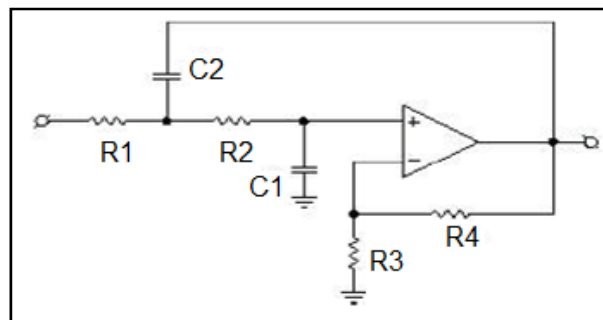


Figura 1.20.a: Filtro pasa bajo no inversor de segundo orden con estructura Sallen Key.

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009].
 Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

$$F(s) = \frac{A}{(\omega_c^2 R_1 R_2 C_1 C_2) s^2 + \omega_c [C_1 (R_1 + R_2) + (1-A) R_1 C_2] s + 1} \quad (1.62)$$

La Figura 1.20.b, indica el esquema de un filtro pasa bajo inversor de segundo orden.

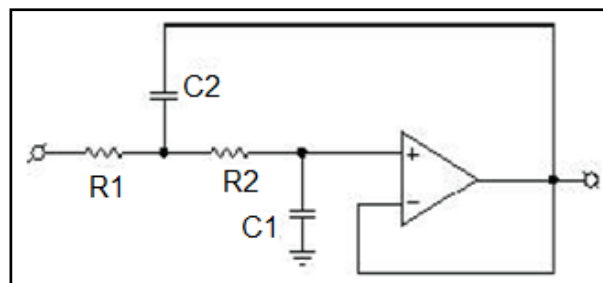


Figura 1.20.b: Filtro pasa bajo inversor de segundo orden con estructura Sallen Key.

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009].
 Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

$$F(s) = \frac{1}{(\omega_c^2 R_1 R_2 C_1 C_2) s^2 + \omega_c C_1 (R_1 + R_2) s + 1} \quad (1.63)$$

Para un circuito de ganancia unidad, los coeficientes serían:

$$A_0 = 1 \quad (1.64)$$

$$a_1 = \omega_c C_1 (R_1 + R_2) \quad (1.65)$$

$$b_1 = \omega_c^2 R_1 R_2 C_1 C_2 \quad (1.66)$$

Dando valores a C_1 y C_2 :

$$R_{1,2} = \frac{a_1 C_2 \pm \sqrt{a_1^2 C_2^2 - 4b_1 C_1 C_2}}{4\pi f_c C_1 C_2} \quad (1.67)$$

Para obtener valores reales:

$$C_2 \geq C_1 \cdot \frac{4b_1}{a_1^2} \quad (1.68)$$

Ejemplo 1.5.

Para el circuito de la figura 1.21.a, diseñar un filtro pasa bajo Tschebychev de segundo orden con una frecuencia de corte de 3Khz y un rizado de 3dB.

Los coeficientes para filtros Sallen Key con aproximación de Tschebychev se encuentran en las tablas G1, G2 y G3 incluidas en anexo G del capítulo 1 y serían:

$$a_1 = 1.065 \quad \text{y} \quad b_1 = 1.9305.$$

$$C_1 = 22\text{nF}.$$

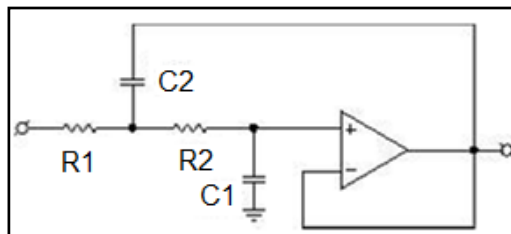


Figura 1.21.a: Filtro pasa bajo inversor de segundo orden con estructura Sallen Key.

Al aplicar la ecuación (1.68), se obtiene:

$$C_2 \geq 22 \times 10^{-9} \frac{4(1,9305)}{(1,065)^2} \quad \Rightarrow \quad 150 \text{ nF (normalizado)}$$

Los valores de C_1 y C_2 se reemplazan en la ecuación (1.67) y se obtiene R_1 y R_2 .

$$R_1 = \frac{1,065(150 \times 10^{-9}) - \sqrt{[(1,065)(150 \times 10^{-9})]^2 - 4(1,9305)(22 \times 10^{-9})(150 \times 10^{-9})}}{4\pi(3 \times 10^3)(22 \times 10^{-9})(150 \times 10^{-9})} = 1,26K$$

$$R_2 = \frac{1,065(150 \times 10^{-9}) + \sqrt{[(1,065)(150 \times 10^{-9})]^2 - 4(1,9305)(22 \times 10^{-9})(150 \times 10^{-9})}}{4\pi(3 \times 10^3)(22 \times 10^{-9})(150 \times 10^{-9})} = 1,3K$$

El circuito de la figura 1.21.b, indica el diseño de un filtro pasa bajo de segundo orden que cumple con las especificaciones requeridas en el ejemplo 1.5.

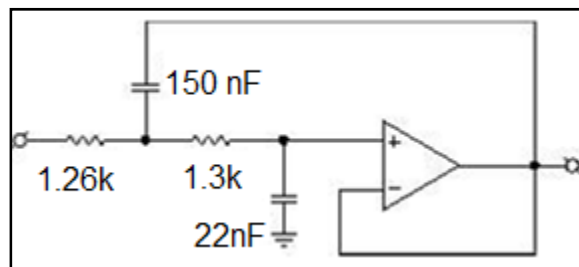


Figura 1.21.b: Filtro pasa bajo de segundo orden.

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009].
Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

Realización de un filtro pasa bajo de orden superior con estructura

Sallen Key.

Un filtro pasa bajo de orden superior se resuelve mediante colocación en serie de filtros de primer y segundo orden hasta obtener el orden necesario; como se indica en la figuras 1.22.a y 1.22.b.

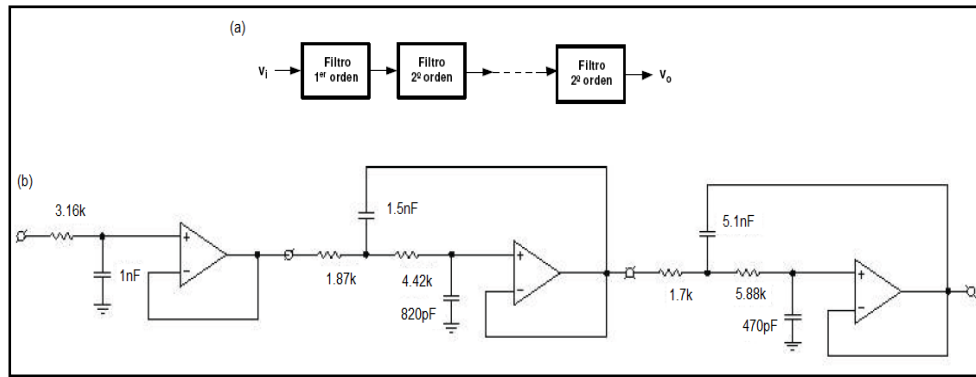


Figura 1.22.a: Diagrama de bloques para el diseño de un filtro de orden superior.
Figura 1.22.b: Diseño final para un filtro pasa bajo de Butterworth de orden 5 con frecuencia de corte de 50 KHz.

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009]. Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

Instrumentación Electrónica [en línea] 2008 - 2009 [ref. de mayo de 2009]. Disponible en Web: <http://laimbio08.escet.urjc.es/assets/files/docencia/IE/TEMA2_filtros.pdf>

1.4.2 Realización de un filtro Pasa Alto.

Para la realización de este tipo de filtro se requiere conocer las siguientes especificaciones:

$$\text{PA} \begin{cases} A_1 \leq |H(\omega)| \leq A & 2\pi f_c \leq \omega < \infty & (1.69) \\ A_2 \leq |H(\omega)| \leq A_1 & 2\pi f_c < \omega < 2\pi f_s & (1.56) \\ 0 \leq |H(\omega)| \leq A_2 & 0 \leq \omega \leq 2\pi f_s & (1.70) \end{cases}$$

Donde:

A_1 Rizado superior.

A_2 Rizado inferior.

f_c Frecuencia de corte.

f_s Frecuencia de supresión.

$$H(s) = \frac{s^2}{s^2 + b_1s + b_0} \tag{1.71}$$

Al igual que un filtro pasa bajo, un filtro pasa alto puede ser de primer orden, segundo orden y orden superior.

Realización de un filtro pasa alto de primer orden con estructura

Sallen Key.

Un filtro pasa alto de primer orden se obtiene reemplazando físicamente en un filtro pasa bajo R en lugar de C y viceversa. En la figuras 1.23.a y 1.23.b, se muestra a un filtro pasa bajo y a un filtro pasa alto de primer orden con estructura Sallen Key.

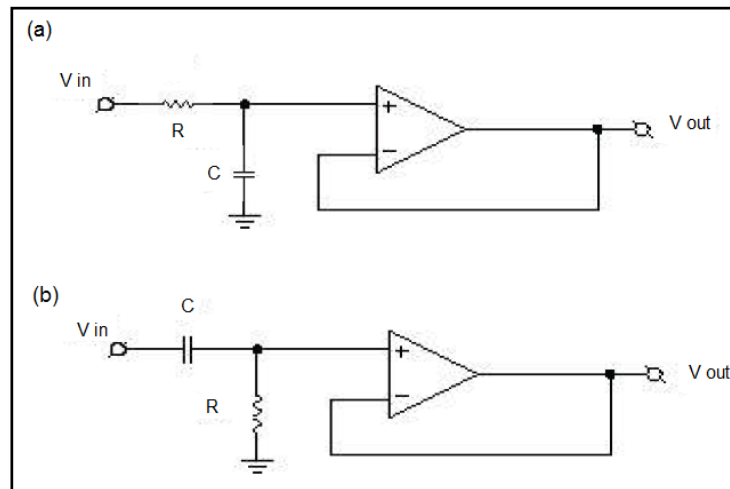


Figura 1.23.a: Filtro pasa bajo de primer orden con estructura Sallen Key.

Figura 1.23.b: Filtro pasa alto de primer orden con estructura Sallen Key.

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009]. Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

$$F(s) = \frac{A_{\infty}}{1 + \frac{s}{a_1}} \quad (1.72)$$

$$F(s) = \frac{1}{1 + \frac{1}{\omega_c RC} \left(\frac{1}{s}\right)} \quad (1.73)$$

$$a_1 = \frac{1}{\omega_c RC} \quad (1.74)$$

$$R = \frac{1}{2\pi f_c a_1 C} \quad (1.75)$$

Realización de un filtro pasa alto de segundo orden con estructura

Sallen Key.

La figura 1.24 indica el esquema de un filtro pasa alto de segundo orden con estructura Sallen Key.

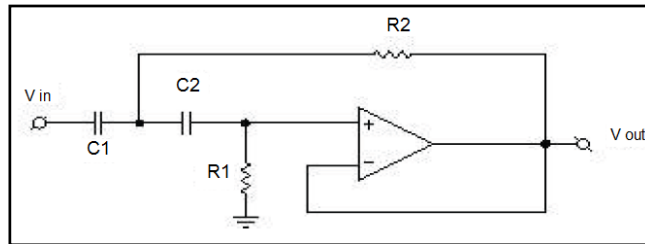


Figura 1.24: Esquema de un filtro pasa alto de segundo orden.

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009]. Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

$$F(s) = \frac{A_{\infty}}{\left(\frac{b_1}{s^2} + \frac{a_1}{s} + 1\right)} \quad (1.67)$$

$$F(s) = \frac{1}{\frac{1}{\omega_c^2 R_1 R_2 C_1 C_2} \left(\frac{1}{s^2}\right) + \frac{R_2 (C_1 C_2)}{\omega_c R_1 R_2 C_1 C_2} \left(\frac{1}{s}\right) + 1} \quad (1.77)$$

$$C_1 = C_2 \quad (1.78)$$

$$a_1 = \frac{2}{\omega_c R_1 C} \quad (1.79)$$

$$b_1 = \frac{1}{\omega_c^2 R_1 R_2 C^2} \quad (1.80)$$

$$R_1 = \frac{1}{\pi f_c C a_1} \quad (1.81)$$

$$R_2 = \frac{a_1}{4\pi f_c C b_1} \quad (1.82)$$

1.4.3. Realización de un filtro Pasa Banda.

Para la realización de esta clase de filtro se requiere conocer las siguientes especificaciones:

$$\text{PB} \left\{ \begin{array}{ll} 0 \leq |H(\omega)| \leq A_2 & 0 \leq \omega \leq \omega_1 & (1.83) \\ A_2 < |H(\omega)| < A_1 & \omega_1 < \omega < \omega_2 & (1.84) \\ A_1 \leq |H(\omega)| \leq A & \omega_2 \leq \omega \leq \omega_4 & (1.85) \\ A_2 < |H(\omega)| < A_1 & \omega_4 < \omega < \omega_5 & (1.86) \\ 0 \leq |H(\omega)| \leq A_2 & \omega_5 \leq \omega < \infty & (1.87) \end{array} \right.$$

$$\omega_3^2 = \omega_1 \cdot \omega_5 = \omega_2 \cdot \omega_4 \quad (1.88)$$

$$B_w = \omega_4 - \omega_2 \quad (1.89)$$

Donde:

A_1 Rizado superior.

A_2 Rizado inferior.

ω_2 Frecuencia de corte inferior.

ω_4 Frecuencia de corte superior.

ω_3 Promedio geométrico de la frecuencia central.

ω_1 Frecuencia de supresión inferior.

ω_5 Frecuencia de supresión superior.

B_w Ancho de banda.

Q Factor de calidad.

$$H(s) = \frac{\sqrt{b_0}s}{s^2 + b_1s + b_0} \quad (1.90)$$

Para diseñar y realizar un filtro pasa banda con estructura Sallen Key normalmente se usan filtros pasa bajo en serie con pasa alto de los órdenes adecuados, como se indica en la figura 1.25.

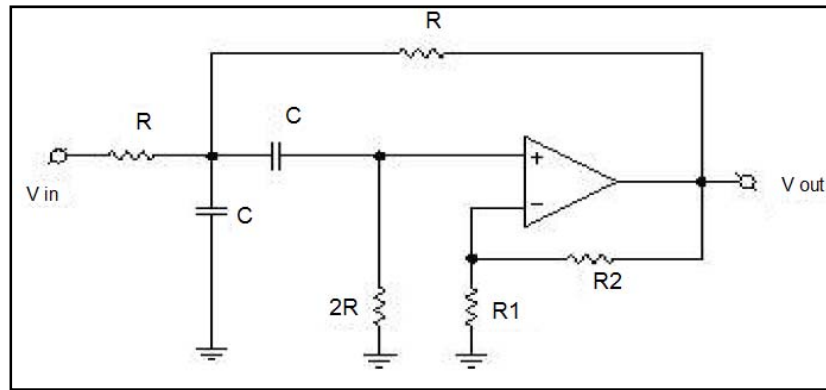


Figura 1.25: Esquema de un filtro pasa banda con estructura Sallen Key.

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009]. Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

$$F(s) = \frac{\frac{A_m s}{Q}}{s^2 + \frac{1}{Q}s + 1} \quad (1.91)$$

La Función de transferencia será:

$$F(s) = \frac{G.R.C.\omega_m s}{R^2 C^2 \omega_m^2 s^2 + R.C.\omega_m(3-G)s + 1} \quad (1.92)$$

La Frecuencia Central:

$$f_m = \frac{1}{2\pi RC} \quad (1.93)$$

La Ganancia Propia:

$$G = 1 + \frac{R_2}{R_1} \quad (1.94)$$

La Ganancia a Frecuencia Central:

$$A_m = \frac{G}{3-G} \quad (1.95)$$

Factor de Calidad del Filtro:

$$Q = \frac{1}{3-G} \quad (1.96)$$

Se debe de especificar f_m y C y entonces resolver R:

$$R = \frac{1}{2\pi f_m C} \quad (1.97)$$

Como R_2 depende de Q y de A_m , se tienen dos opciones a la hora de resolverlo:

Fijando la ganancia a frecuencias medias:

$$R_2 = \frac{2A_m - 1}{1 + A_m} \quad (1.98)$$

Especificando un factor de calidad determinado:

$$R_2 = \frac{2Q - 1}{Q} \quad (1.99)$$

1.4.4 Realización de un Filtro Elimina Banda.

Para realizar un filtro elimina banda se requiere conocer las siguientes especificaciones:

$$EB \left\{ \begin{array}{ll} A_1 \leq |H(\omega)| \leq A & 0 \leq \omega \leq \omega_1 & (1.100) \\ A_2 < |H(\omega)| < A_1 & \omega_1 < \omega < \omega_2 & (1.84) \\ 0 \leq |H(\omega)| \leq A_2 & \omega_2 \leq \omega \leq \omega_4 & (1.101) \\ A_2 < |H(\omega)| < A_1 & \omega_4 < \omega < \omega_5 & (1.86) \\ A_1 \leq |H(\omega)| \leq A & \omega_5 \leq \omega < \infty & (1.102) \end{array} \right.$$

$$\omega_3^2 = \omega_1 \cdot \omega_5 = \omega_2 \cdot \omega_4 \quad (1.88)$$

$$B_w = \omega_5 - \omega_1 \quad (1.104)$$

Donde:

A_1 Rizado superior.

A_2 Rizado inferior.

ω_2 Frecuencia de corte inferior.

ω_4 Frecuencia de corte superior.

ω_3 Promedio geométrico de la frecuencia central.

ω_1 Frecuencia de supresión inferior.

ω_5 Frecuencia de supresión superior.

B_w Ancho de banda.

Q Factor de calidad.

$$H(s) = \frac{s^2 + b_0}{s^2 + b_1s + b_0} \tag{1.105}$$

Para diseñar y realizar un filtro elimina banda con estructura Sallen Key sucede lo mismo que en el filtro pasa-banda; por lo que se puede hacer un filtro elimina banda en la combinación de un filtro pasa alto con filtro un pasa bajo; como se indica en la figura 1.26.

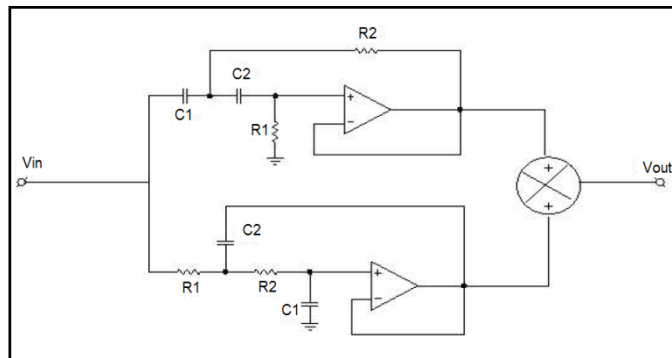


Figura 1.26: esquema de un filtro elimina banda.

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009]. Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

$$F(s) = \frac{A_0(1+s^2)}{s^2 + \frac{1}{Q}s + 1} \tag{1.106}$$

Donde:

$$Q = \frac{f_m}{B} \tag{1.107}$$

La figura 1.27 indica la ubicación de f_m y B .

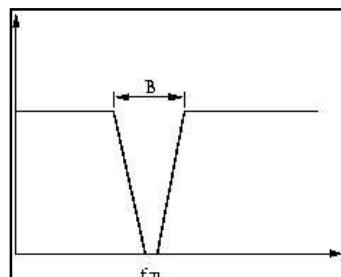


Figura 1.27: Ubicación de f_m y B

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009]. Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

La figura 1.28 indica un filtro elimina banda doble T.

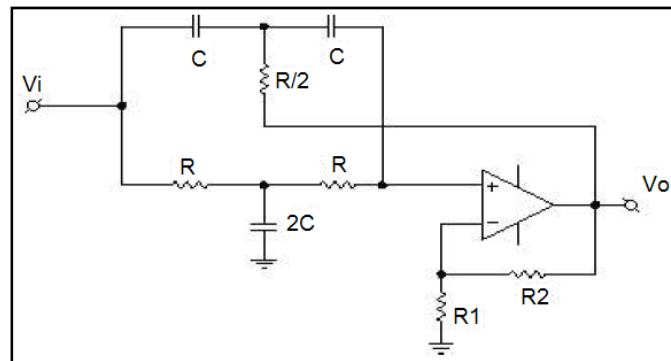


Figura 1.28: Esquema de un filtro elimina banda doble T.

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009]. Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

$$F(s) = \frac{k(1+s^2)}{s^2+2(2-k)s+1} \quad (1.108)$$

Donde:

$$k = 1 + \frac{R_2}{R_1} \quad (1.109)$$

$$A_0 = 1 + \frac{R_2}{R_1} \quad (1.110)$$

Frecuencia Central:

$$f_m = \frac{1}{2\pi RC} \quad (1.93)$$

La Ganancia Propia:

$$G = 1 + \frac{R_2}{R_1} \quad (1.94)$$

Factor de Calidad del Filtro:

$$Q = \frac{1}{2(2-G)} \quad (1.111)$$

1.5 Conclusiones.

1. El estudio en cuanto a filtros se refiere es extenso, como se ha analizado, existen diferentes clases de filtros y con todos ellos se debe de tener en

cuenta parámetros de diseño tanto en el dominio de la frecuencia, así como en el dominio del tiempo como especificaciones de partida.

2. Al momento de implementar una aplicación que involucre el trabajo con un filtro, hay que tener claros los procesos de aproximación y realización, como sus diferencias claramente especificadas dentro de este capítulo.
3. El proceso de aproximación trata sobre el desarrollo de los modelados matemáticos, siendo analizadas las aproximaciones de Butterworth, Tschebychev y Bessel.
4. La aproximación de Butterworth ofrece una respuesta de ganancia plana en la banda de paso, pero mayor oscilación y rizado en respuesta a un escalón.
5. El filtro de Tschebychev tiene dos variantes: Tipo I y tipo II. El filtro de Tschebychev del tipo I presenta como desventaja rizado en la banda de paso.
6. El filtro de Bessel, presenta una mejor respuesta a un escalón de tensión.
7. Al momento de hacer una comparación entre las tres aproximaciones, se determina que el filtro de Butterworth presenta una mejor respuesta ante el escalón que el filtro de Tschebychev.
8. Para la realización de un filtro se tienen tres tipos de estructuras: Sallen Key, Rauch y fuente de voltaje controlada por voltaje (VCVS).
9. La realización de un filtro con una estructura Sallen Key, Rauch y fuente de voltaje controlada por voltaje (VCVS); se puede obtener con la ayuda de un filtro pasa bajo, pasa alto, pasa banda o elimina banda.
10. Un filtro pasa bajo puede ser de primer orden, segundo orden y orden superior. Para diseñar un filtro pasa bajo de primer orden se debe de considerar la función de transferencia normalizada del filtro.
11. Un filtro pasa bajo de orden superior se resuelve mediante colocación en serie de filtros de primer y segundo orden hasta obtener el orden necesario.

12. Un filtro pasa alto puede ser de primer orden, segundo orden y orden superior.
13. Para diseñar y realizar un filtro pasa banda con estructura Sallen Key normalmente se usan filtros pasa bajo en serie con pasa alto de los órdenes adecuados.
14. Para diseñar y realizar un filtro elimina banda con estructura Sallen Key se puede hacer un filtro elimina banda en la combinación de un filtro pasa alto con filtro un pasa bajo.

CAPÍTULO II

IMPLEMENTACIÓN DE HARDWARE

2.1 Introducción.

Los Microcontroladores están concebidos fundamentalmente para ser utilizados en aplicaciones puntuales. En dichas aplicaciones, el Microcontrolador ejecuta un programa guardado permanente en su memoria, el cual trabaja con algunos datos almacenados temporalmente e interactúa con el exterior a través líneas de entrada y salida.

Los lenguajes de alto nivel más empleados para programar con Microcontroladores son el C y el BASIC de los que existen varias empresas que comercializan versiones de compiladores e intérpretes para diversas familias de micro controladores.

Según la aplicación a la que orienta el fabricante, cada modelo de micro controlador incorpora una diversidad de complementos que refuerzan la potencia y la flexibilidad del dispositivo.

2.2 Estructura básica de un Microcontrolador.

Los Microcontroladores son diseñados para disminuir el costo económico y el consumo de energía de un sistema en particular; por lo que el tamaño de la Unidad Central de Procesamiento (CPU), la cantidad de memoria y los periféricos incluidos dependerán de la aplicación.

Un Microcontrolador combina la unidad central de procesamiento (CPU), la memoria y los recursos de entrada y salida en un único circuito integrado.

En la figura 2.1, indica un esquema de un Microcontrolador metido dentro de un encapsulado de circuito integrado, con su procesador (CPU), buses, memoria,

periféricos y puertos de entrada salida. Fuera del encapsulado se ubican otros circuitos para completar periféricos internos y dispositivos que pueden conectarse a los pines de entrada/salida como convertidores de analógico a digital, temporizadores, UARTs y buses de interfaz serie.

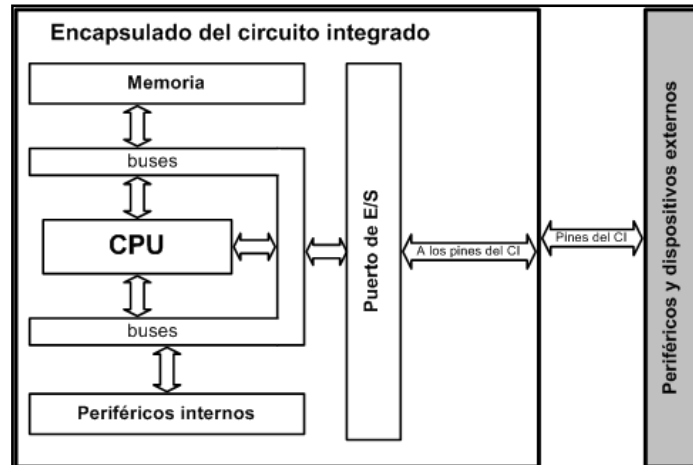


Figura 2.1: Esquema de un Microcontrolador PIC.

Fuente: [en línea] [ref. de noviembre de 2009]. Disponible en Web:

<http://images.google.com/ec/imgres?imgurl=http://www.ucontrol.com.ar/wiki/images/b/b7/EluC_02a.png&imgrefurl=http://www.ucontrol.com.ar/wiki/index.php/EI_microcontrolador&usg=__pthSwcTcvEoytIPKdx_r3mPy3ms=&h=193&w=346&sz=7&hl=es&start=14&um=1&itbs=1&tbnid=ogCZc2klx8aUvM:&tbnh=67&tbnw=120&prev=/images%3Fq%3Darquitectura%2Bvon%2Bnewman%26hl%3Des%26riz%3D1T4RNWN_esEC300EC301%26sa%3DN%26um%3D1>

Un Microcontrolador típico tiene un generador de reloj integrado y una pequeña cantidad de memoria RAM y ROM/EPROM/EEPROM/FLASH. Para hacerlo funcionar, todo lo que se necesita son unos pocos programas de control y un cristal de sincronización. Estos dispositivos integrados pueden ser controlados por instrucciones.

La figura 2.2 indica el diagrama de bloques general de un Microcontrolador.

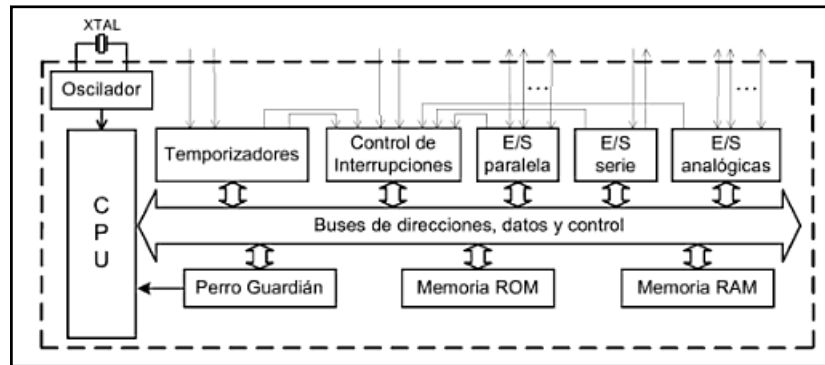


Figura 2.2: Esquema de bloques general de un Microcontrolador PIC.

Fuente: PALLÁS ARENY, Ramón; *Microcontroladores: fundamentos y aplicaciones con PIC* [en línea] [ref. de julio de 2009]. Disponible en Web: http://books.google.com.ec/books?id=ODenKGOHMRkC&dq=microcontroladores&source=gbs_navlinks_s

- **Oscilador:** Genera pulsos que sincronizan todas las operaciones.
- **CPU:** Cerebro del Microcontrolador; trae las instrucciones del programa una a una y las decodifica haciendo que se ejecuten.
- **Watchdog (perro guardián):** Garantiza la seguridad en el funcionamiento del Microcontrolador ante cualquier fallo. Su objetivo es provocar una re-iniciación cuando el programa queda bloqueado.
- **Control de interrupciones:** Los Microcontroladores tienen recursos para recibir y procesar las solicitudes de interrupción, tienen asociados dos bits: uno informativo y otro de control. Disponen además de un bit de control global, el mismo que permite el paso o no de cualquier interrupción hacia el CPU.
- **Entrada y salida paralela:** Permite que los bits de entrada y salida sean transferidos simultáneamente hacia y desde dispositivos periféricos tales como: Leds, teclados matriciales, switches, displays, etc.

Los Microcontroladores disponen de un bus para transportar el dato y un bus para el control de la transferencia de datos entre el periférico y el puerto. En la figura 2.3 se representa una conexión de un puerto bidireccional con su bus de datos y su bus de control.

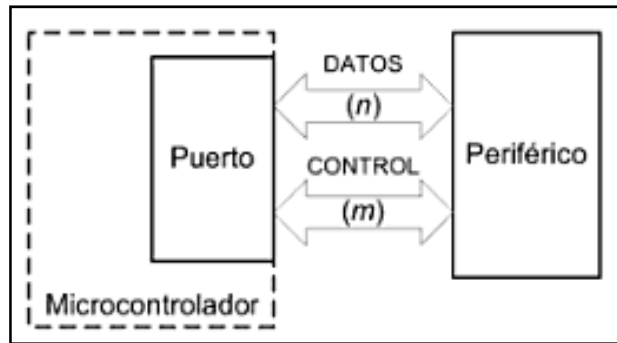


Figura 2.3: Conexión entre un Microcontrolador y un periférico a través de un puerto de entrada y salida.

Fuente: PALLÁS ARENY, Ramón; *Microcontroladores: fundamentos y aplicaciones con PIC* [en línea] [ref. de julio de 2009]. Disponible en Web:

<http://books.google.com.ec/books?id=ODenKGOHMRkC&dq=microcontroladores&source=gbs_navlinks_s>

- **Entrada y salida en serie:** Consiste en enviar y recibir de uno a uno y en forma sucesiva los bits de una palabra a través de terminales. Existen dos clases de comunicación: SÍNCRONA Y ASÍNCRONA.
- **Entrada y salida analógica:** Tiene que ver con la adquisición y generación de señales mediante dispositivos integrados en el Microcontrolador o en circuitos periféricos.
- **Temporizadores:** Se emplean para controlar períodos de tiempo y para llevar la cuenta de acontecimientos que suceden en el exterior.
- **Memoria:** Un Microcontrolador tiene dos tipos de memoria. La memoria de PROGRAMA o no volátil, básicamente de solo lectura; y la memoria de DATOS, fundamentalmente de lectura y escritura.

Existen diferentes clases de memorias, las más comunes se describen a continuación.

- **RAM:** Memoria de escritura y lectura. Existen dos tipos: estática donde la información almacenada está estable hasta q se corte la alimentación y dinámica que necesita refrescamiento periódico de la información.
- **ROM:** Es la memoria de solo lectura. Se utiliza para almacenar los programas que ponen en marcha el ordenador y realizan los diagnósticos.

- **EPROM y OTP:** Memoria donde se puede grabar y borrar información después de cierto tiempo (EPROM). Una vez programada la memoria no se puede borrar (OTP).
- **EEPROM:** Memoria no volátil de lectura y escritura. Se puede reprogramar en un número finito de veces.
- **Flash:** Este tipo de memoria se puede leer y escribir en celdas individuales. Para escribir en una celda hay que borrar primero su información.

2.3 Arquitecturas.

Las Arquitecturas de un Microcontrolador representan dos soluciones al problema de conexión entre el CPU con la memoria y a la organización de la misma.

En la memoria de un Microcontrolador, se almacenan instrucciones y datos. Las instrucciones deben pasar secuencialmente hacia el CPU para su decodificación y ejecución, en tanto que algunos datos en memoria son leídos por el CPU y otros son escritos en la memoria desde el CPU.

Un Microcontrolador puede tener dos tipos de arquitecturas:

- **Von Neumann:** Utiliza una sola memoria, donde coexisten las instrucciones de programa y los datos, accedidos con un bus de dirección, uno de datos y uno de control.
- **Harvard:** Utiliza, además de la memoria, un procesador que tiene los buses separados, de modo que cada tipo de memoria tiene un bus de datos, uno de direcciones y uno de control.

La memoria de programas que almacena instrucciones es de solo lectura, mientras que la memoria de datos se puede leer y escribir.

La figura 2.4.a, muestra el modelo de arquitectura Von Neumann. Con un mismo bus de direcciones se localizan instrucciones y datos y por un único bus de datos transitan tanto instrucciones como datos.

La figura 2.4.b, indica el modelo de arquitectura Harvard. La memoria de datos tiene sus propios buses de direcciones, datos y control independientes de los buses de la memoria de programa.

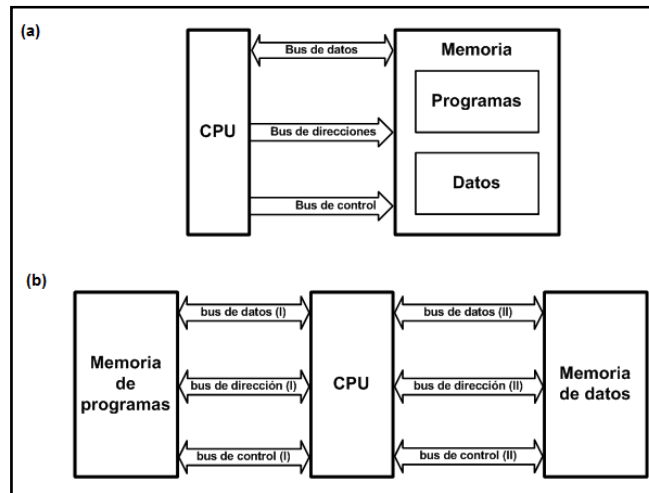


Figura 2.4.a: Arquitectura Von Neumann.

Figura 2.4.b: Arquitectura Harvard.

Fuente: [en línea] [ref. de noviembre de 2009]. Disponible en Web:

<http://images.google.com.ec/imgres?imgurl=http://www.ucontrol.com.ar/wiki/images/b/b7/EluC_02a.png&imgrefurl=http://www.ucontrol.com.ar/wiki/index.php/El_microcontrolador&usq=__pthSwcTcvEoytIPKDX_r3mPy3ms=&h=193&w=346&sz=7&hl=es&start=14&um=1&itbs=1&tbnid=ogCZc2klx8aUvM:&tbnh=67&tbnw=120&prev=/images%3Fq%3Darquitectura%2Bvon%2Bnewman%26hl%3Des%26rlz%3D1T4RNWN_esEC300EC301%26sa%3DN%26um%3D1>

2.4 Familias de Microcontroladores.

Los Microcontroladores de un mismo tipo forman una familia que se caracteriza por tener el mismo CPU y ejecutar el mismo repertorio de instrucciones. Es lo que se conoce como núcleo (core).

Los miembros de una misma familia de Microcontroladores tienen el mismo núcleo pero se diferencian en sus entradas, salidas y memoria.

Existen diferentes familias de Microcontroladores, la tabla 2.1 indica a las familias más comunes y a sus fabricantes.

Familias de Microcontroladores y sus fabricantes

Empresa	8 bits	12 bits	14 bits	16 bits	32 bits	64 bits
Atmel	ATmega8, 89Sxxxx familia similar 8051			ATmega16		
AVR						
Freescale (antes Motorola)	68HC05, 68HC08, 68HC11, HCS08	x	x	68HC12, 68HCS12, 68HCSX12, 68HC16	683xx, PowerPC Architecture, ColdFire	x
Hitachi, Ltd	H8	x	x	x	x	x
Holtek	HT8					
Intel	MCS-48 (familia 8048)	x	x	MCS96, MXS296	x	x
	MCS51 (familia 8051)					
	8xC251					
National Semiconductor	COP8	x	x	x	x	x
Microchip	Familia 10f2xx Familia 12Cxx Familia 12Fxx Familia 16Cxx Familia 16Fxx Familia 18Cxx Familia 18Fxx			PIC 24H PIC 24 F dsPIC30FXX dsPIC33F	PIC32	x
NEC	78K					
Parallax						
ST	ST 62, ST 7					
Texas Instruments	TMS370, MSP430					
Zilog	Z8, Z86E02					
Silabs	C8051					

Tabla 2.1: Fabricantes de Microcontroladores y sus respectivas familias.

Fuente: WIKIPEDIA. *Microcontrolador* [en línea] 2009 [ref. de noviembre de 2009]. Disponible en Web:
<<http://es.wikipedia.org/wiki/Microcontrolador#Memoria>>

2.5 Microcontrolador PIC.

Los pic son una familia de Microcontroladores fabricados por Microchip; los Pics actuales incorporan una amplia gama de mejoras con respecto a versiones anteriores. La arquitectura de un pic corresponde a la figura 2.1. En el cuadro 2.1 se tienen los diferentes tipos de familias de Microcontroladores de Microchip.

2.5.1 Características principales.

- Núcleo de CPU de 8/16/32 bits con arquitectura Harvard modificada.
- Memoria Flash y ROM.
- Temporizadores de 8/16/32 bits.
- Soporte de interfaz USB
- Controladores LCD.
- Memoria EEPROM interna con duración de hasta un millón de ciclos lectura/escritura.

2.6 Microcontrolador Microchip PIC16F648A.

En el cuadro 2.1 se observa que la familia de Microcontroladores 16FXX trabaja con un núcleo de 8 bits. El Microcontrolador pic16F684A además de contar con este tipo de núcleo de CPU, presenta una arquitectura de procesador Harvard, mejorando el ancho de banda de operación, obteniendo así diferentes anchos de buses.

2.6.1 Características generales⁵.

- Código de fuente compatible con los sistemas de la instrucción PIC16.
- **Memoria de PROGRAMA:**Flash
- **Memoria de DATOS**, dividida en 2 áreas:
 - Área RAM con capacidad de hasta:
 - 256 bytes.
 - Área EEPROM formada por:
 - 256 bytes.
- **Temporizador timer 0:** Contador de tiempo de 8 bits.
- **Temporizador timer 1:** Contador de tiempo de 16 bits.
- **Temporizador timer 2:** Contador de tiempo de 8 bits.
- **Número de pines:** 16 de entrada y salida.
- **Voltaje de operación:** 2 a 5,5V.
- **Frecuencia de operación:** Hasta 20MHz.

⁵ Más información acerca del PIC16F684A se puede obtener en <http://ww1.microchip.com/downloads/en/DeviceDoc/40044G.pdf>

2.6.2 Esquema de bloques.

La figura 2.5 indica mediante un esquema de bloques genérico el funcionamiento de un pic16F648A.

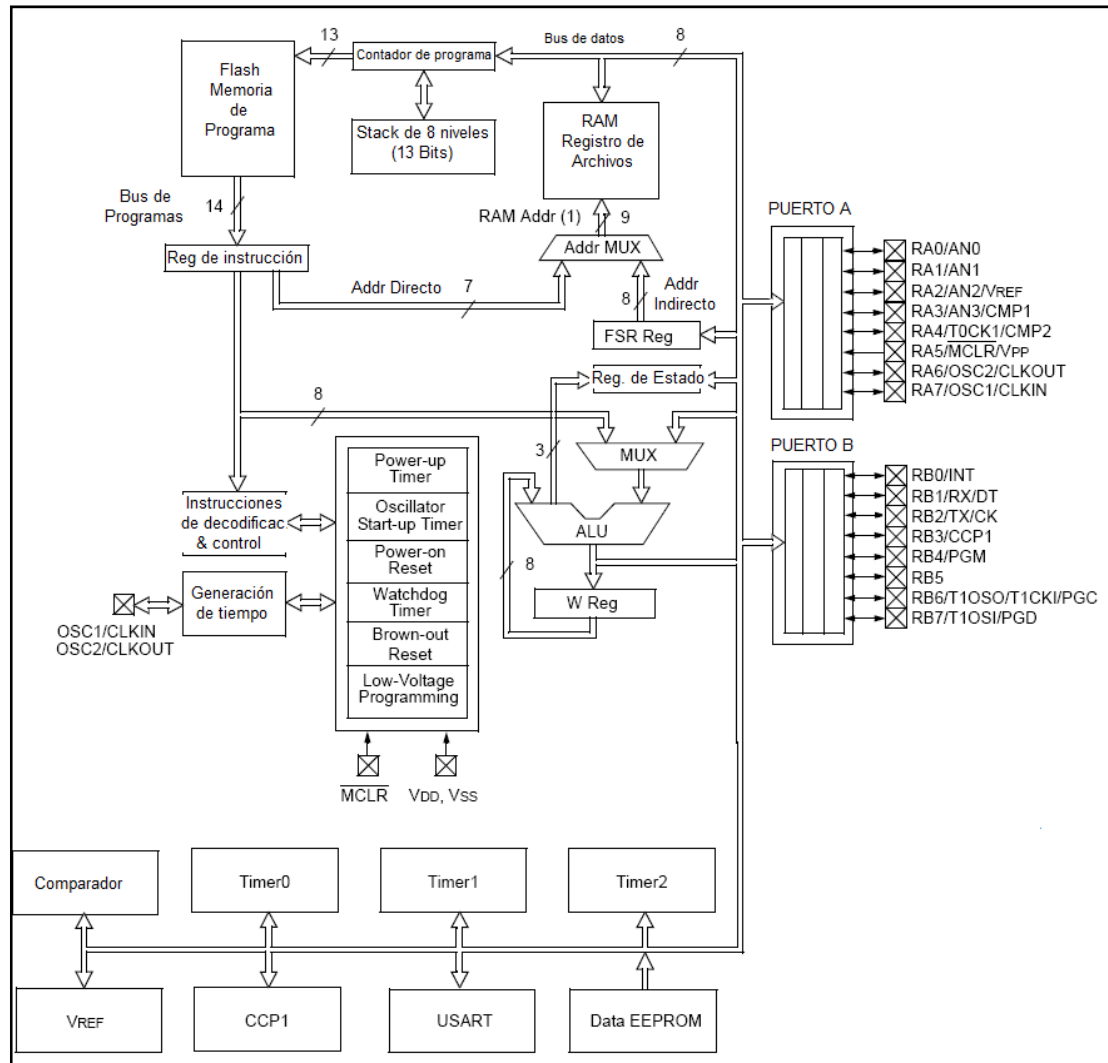


Figura 2.5: Esquema de bloques de un Microcontrolador PIC16F648A.

Fuente: MICROCHIP. *PIC16F627A/628A/648A Data Sheet*. [en línea] [ref. de noviembre de 2009].
 Disponible en Web: <<http://ww1.microchip.com/downloads/en/DeviceDoc/40044G.pdf>>

2.7 Microcontrolador Microchip PIC32MX360F512F.

El PIC32 viene de una familia nueva de Microcontroladores de Microchip, mismo que incorpora variaciones y mejoras con respecto a familias anteriores de PICS (cuadro 2.1).

Dichos avances tienen que ver con los siguientes aspectos:

- Rendimiento y memoria.

- Conectividad.
- Soporte de software y herramientas.

Con un aumento en la velocidad de procesado y su capacidad para ejecutar código generado con lenguajes de programación como C, los microcontroladores de 32 bits son recomendables para ser utilizados en diferentes aplicaciones.

2.7.1 Características generales⁶:

- Compatible con herramientas de desarrollo Microchip.
- Núcleo de 32-bits (MIPS32 M3K).
- **Memorias:**
 - RAM: interna de 32 Kb.
 - Flash: auto-programable 512Kb.
- **Arquitectura:** Harvard.
- **Bus:** Matricial de alto procesamiento, con soporte de alta velocidad para accesos concurrentes a la memoria y periféricos.
- **Convertidor USB-Serial:** Sirve para la interfaz USB con el computador
- **Puerto paralelo maestro:** 16-bits
- **Número de pines:** 100
- **Voltaje de Alimentación:** 3,3V.
- **Frecuencia de operación:** hasta 80 MHz

2.7.2 Esquema de bloques.

La figura 2.6 indica mediante un esquema de bloques genérico el funcionamiento de un PIC32MX3XX.

⁶ Más información acerca del pic32 se puede obtener en <http://ww1.microchip.com/downloads/en/DeviceDoc/39564c.pdf>

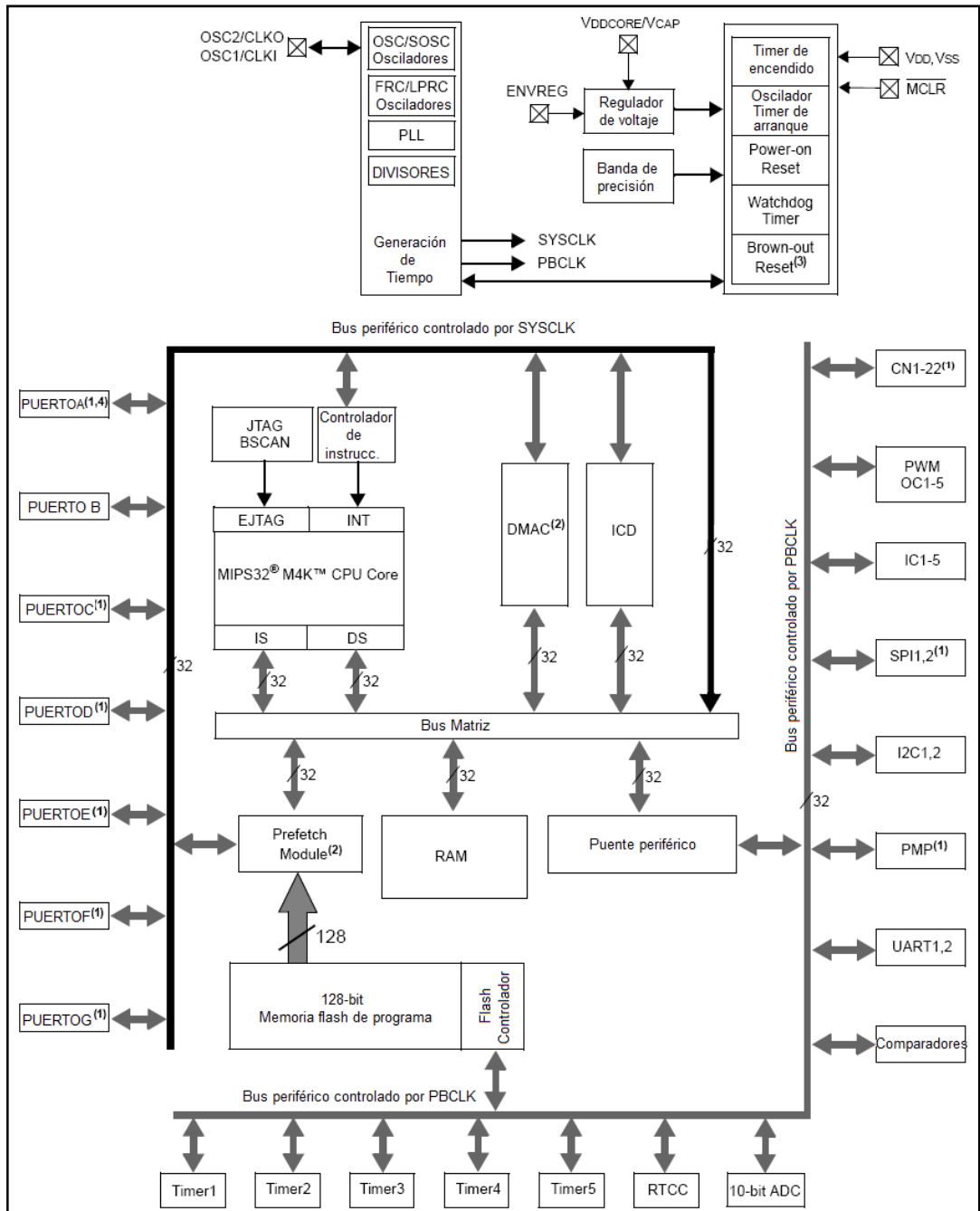


Figura 2.6: Esquema de bloques de un Microcontrolador PIC32MX3XX.

Fuente: MICROCHIP. *PIC32MX3XX/4XX Family Data Sheet*. [en línea] [ref. de julio de 2009].
 Disponible en Web: <<http://ww1.microchip.com/downloads/en/DeviceDoc/39564c.pdf>>

2.8 Generador de funciones.

El generador de funciones tanto para señal pura como para señal de ruido, es diseñado y basado en el circuito integrado generador de funciones monolítico XR-2206, capaz de generar señales con forma de onda senoidal, triangular,

cuadrada, diente de sierra y tren de pulsos de acuerdo a la configuración que se asigne.

La figura 2.7⁷ indica el diagrama de un generador de funciones diseñado para el Simulador Didáctico de Filtros Analógicos Activos Basado En Microcontrolador PIC32.

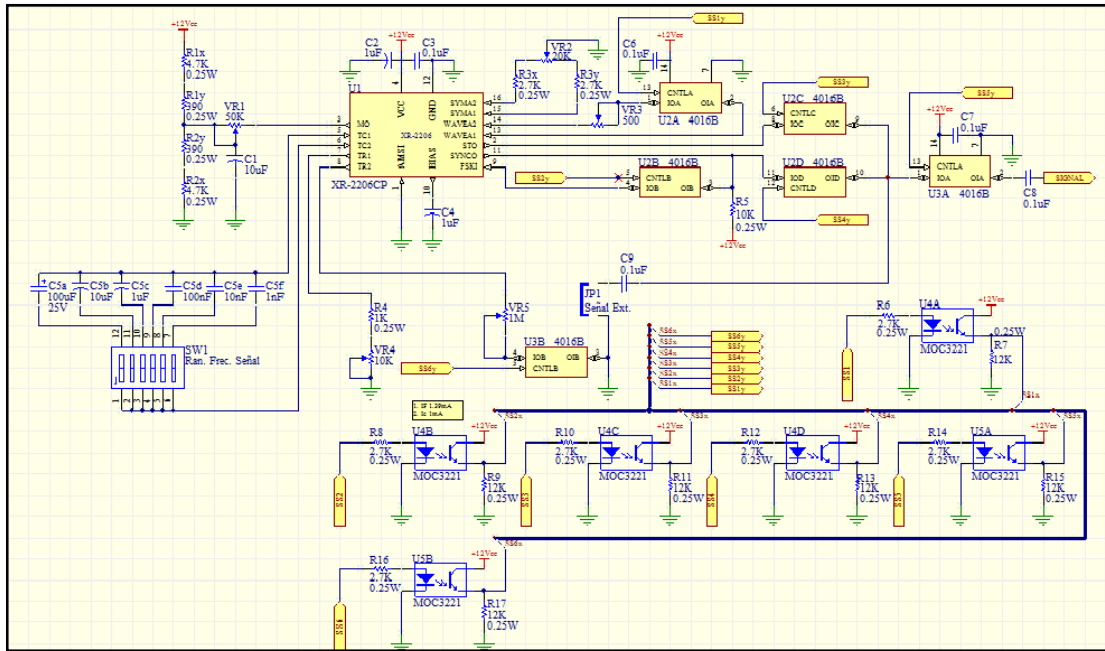


Figura 2.7: Diagrama de un generador de funciones diseñado para el Simulador Didáctico de Filtros Analógicos Activos Basado En Microcontrolador PIC32.

Fuente: Autores

En la figura A.1 (anexo A del capítulo 2) se observa los dispositivos con el que cuenta el generador de funciones; cada uno de estos componentes dispone de funciones definidas que se detallan a continuación.

- **Amplitud:** La amplitud de la señal de salida de este generador se ajusta variando la resistencia (potenciómetro VR₁) según las curvas indicadas en la figura 2.8 y dado que la amplitud máxima que soporta el PIC32MX360F512L como entrada en sus canales analógicos es de 3,3Vpp, se ha escogido un potenciómetro de 50KΩ lo cual produce un voltaje pico-pico de 3V en salida de señal senoidal, mientras que para el resto de señales se deberá atenuar las mismas ajustando dicho potenciómetro.

⁷ Un diagrama más amplio se encuentra en el anexo M correspondiente al capítulo 2

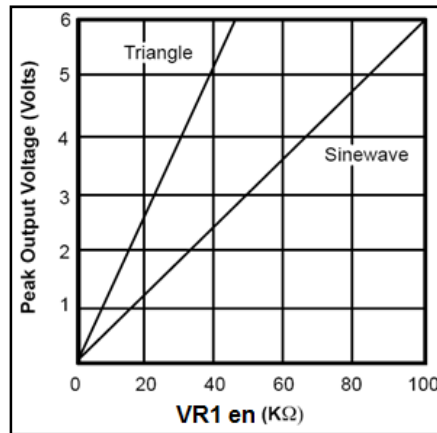


Figura 2.8: Amplitud de Salida en función de VR1.

Fuente: *Instrumentación Electrónica* [en línea] 2008 - 2009 [ref. de mayo de 2009]. Disponible en Web: <http://laimbio08.escet.urjc.es/assets/files/docencia/IE/TEMA2_filtros.pdf>

La figura 2.9 indica un partidor de tensión formado por R_1 y R_2 con un voltaje Thevening igual a $V_{CC}/2$ y una corriente máxima de 2,4mA ($VR_1=0K\Omega$) con lo cual se tiene:

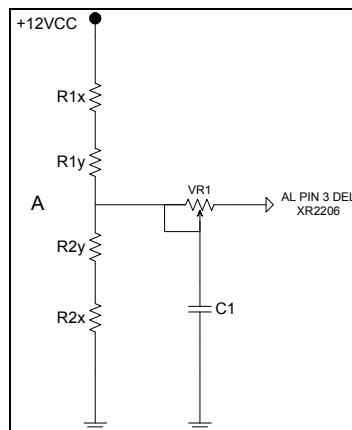


Figura 2.9: Partidor de tensión formado por R_1 y R_2 .

Fuente: Autores

$$V_A = V_{CC} \frac{R_2}{R_1 + R_2} \quad 2.1$$

$$R_1 = R_2 \quad 2.2$$

$$V_A = V_{CC} (0,5) \quad 2.3$$

$$V_A = 12 (0,5)$$

$$V_A = 6V$$

$$V_{TH} = V_A \quad 2.4$$

$$R_{TH} = \frac{R_1 R_2}{R_1 + R_2} \quad 2.5$$

$$I_{THm\acute{a}x} = \frac{V_{TH}}{R_{TH} + VR_1} \quad 2.6$$

$I_{THm\acute{a}x}$ se da con $VR_1 = 0\Omega$; aplicando la ecuaci3n 2.6:

$$I_{THm\acute{a}x} = 2,4mA$$

Pero se escoge como m\acute{a}ximo 2,35 mA.

$$R_{THm\acute{i}n} = \frac{V_{TH}}{I_{THm\acute{a}x}} \quad 2.7$$

Aplicando la ecuaci3n 2.7 y usando los valores obtenidos en la ecuaci3n 2.3 y escogiendo 2,35 mA para $I_{THm\acute{a}x}$.

$$\mathbf{R_{THm\acute{i}n} = 2,553K\Omega}$$

Aplicando la ecuaci3n 2.2.

$$R_1 = R_2 = 2R_{THm\acute{i}n} \quad 2.8$$

$$\mathbf{R_1 = 5,106K\Omega}$$

$$\mathbf{R_2 = 5,106K\Omega}$$

$$R_1 = R_{1x} + R_{1y} \quad 2.9$$

R_{1x} se escoge a 4,7K Ω y R_{1y} a 390 Ω .

Aplicando la ecuaci3n 2.9 se obtiene:

$$\mathbf{R_1 = 5,09K\Omega}$$

$$R_2 = R_{2x} + R_{2y} \quad 2.10$$

R_{2x} se escoge a 4,7K Ω y R_{2y} a 390 Ω .

Aplicando la ecuación 2.10:

$$R_2 = 5,09K\Omega$$

Ahora se comprueba que no se exceda $I_{THm\acute{a}x}$ usando la ecuación 2.5:

$$R_{TH} = 2,545K\Omega$$

$$I_{TH} = \frac{V_{TH}}{R_{TH}} \quad 2.11$$

$$I_{TH} = 2,358mA$$

$$2,358mA \approx 2,36mA \text{ por lo tanto no excede de } 2,4mA$$

Finalmente, como se aprecia en la figura 2.8, el capacitor C_1 de $10\mu F$ sirve para mantener estable el nivel de voltaje de entrada al pin 3 seleccionado de acuerdo a la salida pico-pico que se desee.

- **Simetría:** Para obtener una mejor simetría $V_{(+)}=V_{(-)}$, es recomendable colocar una resistencia ajustable $VR_2 = \frac{(VR_1)}{2}$, es decir $VR_2 = 25K\Omega$ entre los pines 15 y 16 del Cl. XR-2206 mientras que el pin ajustable de la resistencia variable (potenciómetro) debe ir conectado a tierra (GND).

Por lo tanto, se ha colocado un potenciómetro VR_2 de $20K\Omega$ en serie a cada uno de los extremos del potenciómetro con una resistencia de $2,7K\Omega$, así:

$$VR_2 = 20K\Omega$$

$$R_3 = R_{3x} + R_{3y} \quad 2.12$$

$$R_{3x} = R_{3y} = 2,7K\Omega$$

$$R_T = R_{3x} + VR_2 + R_{3y} \quad 2.13$$

$$R_T = 25,4K\Omega$$

- **Distorsión armónica total (thd):** Según la hoja de datos del fabricante, se recomienda colocar una resistencia ajustable VR_3 de 500Ω entre los pines 14 y 13 del Cl. XR-2206 para calibrar la Distorsión armónica total, esto

siempre y cuando la señal de salida seleccionada sea senoidal, ya que para el resto de señales el pin 13 del CI. XR-2206 quedará desconectado de la resistencia variable VR3.

- **Bías:** Según la hoja de datos del fabricante, se recomienda colocar un capacitor C_4 de $1\mu\text{F}$ al pin 10 del CI. XR-2206.
- **Modulación de amplitud:** Como el CI. XR-2206 está siendo utilizado únicamente como generador de señales, mas no como modulador de amplitud o frecuencia, entonces el pin 1 del generador monolítico debe ser conectado a tierra (GND).
- **Filtro de alimentación:** El fabricante recomienda colocar un capacitor C_2 de $1\mu\text{F}$, lo más próximo al pin 4 para la estabilidad del voltaje de alimentación del integrado. Adicionalmente se ha colocado un capacitor C_3 de $0,1\mu\text{F}$ conocido como capacitor de bypass (filtro de ruido proveniente de la fuente de poder) así mismo lo más próximo al pin 4 del CI. XR-2206.
- **Frecuencia:** Para seleccionar la frecuencia de la señal de salida, se tiene que operar los pines 5 (TC_1), 6 (TC_2), 7 (TR_1), 8 (TR_2), de los cuales TC_1 y TC_2 son las entradas del capacitor C_5 que es el encargado directo de seleccionar el rango de frecuencia seleccionable ($F_{\text{mín.}} - F_{\text{máx.}}$), por otro lado TR_1 y TR_2 son las salidas a resistencias ($R_4 + \text{VR}_4$) y VR_5 , las mismas que son encargadas del ajuste fino de la frecuencia de la señal de salida.

En el diseño del simulador didáctico, se ha aplicado las recomendaciones del fabricante, el cual señala que el valor del capacitor C_5 debe ser escogido entre $1\mu\text{F}$ y $100\mu\text{F}$, en tanto que $R_4 + \text{VR}_4$ no puede ser menor a $1\text{K}\Omega$ y $R_4 + \text{VR}_4 + \text{VR}_5$ no puede ser mayor a $2\text{M}\Omega$.

La frecuencia para las señales senoidal, triangular y cuadrada (VR_5 desconectado) viene dada por la siguiente expresión:

$$F = \frac{1}{(R_4 + \text{VR}_4)C_5}$$

En tanto que para las señales Diente de sierra y tren de pulsos (VR_5 conectado), la frecuencia viene dada por:

$$F = \frac{2}{C} \left(\frac{1}{R_4 + VR_4 + VR_5} \right) \quad 2.15$$

Mientras que el Duty Cycle para el Tren de Pulsos (VR_5 conectado) viene dado por:

$$Duty\ Cycle = \frac{R_4 + VR_4}{R_4 + VR_4 + VR_5} \quad 2.16$$

Así, para obtener mayores rangos de frecuencia y un ajuste más preciso, se han escogido 6 valores para C_5 ($C_{5a}=100\mu F$, $C_{5b}=10\mu F$, $C_{5c}=1\mu F$, $C_{5d}=100nF$, $C_{5e}=10nF$, $C_{5f}=1nF$) seleccionables a través del Dip Switch SW_1 , en tanto que para los timing resistors outputs se ha escogido $R_4=1K\Omega$, $VR_4=10K\Omega$ y $VR_5=1M\Omega$ con lo cual se obtienen los siguientes rangos de frecuencia mostrados en las Tablas 2.2 y 2.3.

Tabla 2.2: Rangos de Frecuencia para las Señales Senoidal, Triangular y Cuadrada en función de C_5 , R_4 y VR_4 .

C_5	$F_{min} = (R_4 + VR_4) = 11K\Omega$	$F_{max} = (R_4 + VR_4) = 1K\Omega$
$100\mu F$	0,909 Hz	10 Hz
$10\mu F$	9,091 Hz	100 Hz
$1\mu F$	90,909 Hz	1 Khz
$100nF$	909,091 Hz	10 Khz
$10nF$	9,091 Khz	100 Khz
$1nF$	90,909 Khz	1 Mhz

Tabla 2.2: Rangos de frecuencia para señales senoidal, triangular y cuadrada en función de C_5 , R_4 Y VR_4

Fuente: Autores

Tabla 2.3: Rangos de Frecuencia para las Señales Diente de Sierra y Tren de Pulsos en función de C_5 , R_4 , VR_4 y VR_5 .

C_5	$F_{\min} = (R_4 + VR_4) = 11K\Omega$	$F_{\max} = (R_4 + VR_4) = 1K\Omega$
$100\mu F$	0,909 Hz	10 Hz
$10\mu F$	9,091 Hz	100 Hz
$1\mu F$	90,909 Hz	1 KHz
100nF	909,091 Hz	10 KHz
10 nF	9,091 KHz	100 KHz
1 nF	90,909 KHz	1 Mhz

Tabla 2.3: Rangos de frecuencia para señales diente de sierra y tren de pulsos en función de C_5 , R_4 , VR_4 y VR_5 .

Fuente: Autores

Los valores máximo y mínimo que se pueden seleccionar para el duty cycle del tren de pulsos son:

$$Duty\ Cycle_{\max} = \frac{R_4 + VR_4}{R_4 + VR_4 + VR_5} 100\% \quad 2.17$$

$$Duty\ Cycle_{\max} = \frac{1K\Omega}{1K\Omega} 100\%$$

$$Duty\ Cycle_{\max} = 100\%$$

$$Duty\ Cycle_{\min} = \frac{R_4 + VR_4}{R_4 + VR_4 + VR_5} 100\% \quad 2.17$$

$$Duty\ Cycle_{\min} = \frac{11000}{1011000} 100\%$$

$$Duty\ Cycle_{\min} = 1,088\%$$

- **Señal de salida:** La señal de salida del C.I. XR-2206 es seleccionable automáticamente a través del teclado matricial y del Microcontrolador PIC32MX360F512 que se comunica por un puerto serial con el PIC16F648A quien se encarga de conmutar los distintos switches analógicos 4016B a través de los optotransistores MOC3221.

Según el estado (activo - inactivo) de los switches 4016B se obtendrá uno u otro tipo de salida como se muestra en la Tabla 2.4.

Tabla 2.4 Señales de Salida en función de los Switchs Analógicos 4016B

SS_{1Y}	SS_{2Y}	SS_{3Y}	SS_{4Y}	SS_{5Y}	SS_{6Y}	SEÑAL DE SALIDA
1	0	1	0	1	0	Senoidal
0	0	1	0	1	0	Triangular
X	0	0	1	1	0	Cuadrada
X	1	1	0	1	1	Diente de sierra
X	1	1	1	1	1	Tren de pulsos

Tabla 2.4: Señales de salida en función de los switchs analógicos 4016B

Fuente: Autores

El diseño del generador de señal y ruido del simulador didáctico, tiene la posibilidad de salida de señales generadas para otras aplicaciones, esto se consigue deshabilitando el switch analógico U3_A (SS_{5y}) y conectando (cables-terminales) al jumper JP₁.

Si se desea utilizar señales externas para probar un filtro determinado, entonces se debe deshabilitar los switchs U2_C (SS_{3y}), U2_D (SS_{4y}) y habilitar el switch U3_A (SS_{5y}) mientras que el estado de los switchs U2_A (SS_{1y}), U2_B (SS_{2y}), U3_B (SS_{6y}) no importa.

Finalmente cualquiera que sea el caso: señal interna (entrada o salida) o señal externa (entrada) se ha conectado a través de capacitores (C₈ y C₉) de 0,1μF para eliminar la componente continua de la señal, debiendo considerar el usuario este detalle para posterior diseño y aplicación.

- **Control de los switchs analógicos 4016B:** Los switchs analógicos 4016B son los encargados directos de la conmutación entre las diferentes configuraciones para los diferentes tipos de señales, sin embargo las señales de control de estos CI. es de 12V, por lo que se los ha habilitado a través de optotransistores MOC3221 polarizándolos como se indica en la figura 2.10 y basados en las curvas de transferencia de corriente de la hoja de datos del CI. MOC3221 (PS2501L-2) para el respectivo cálculo:

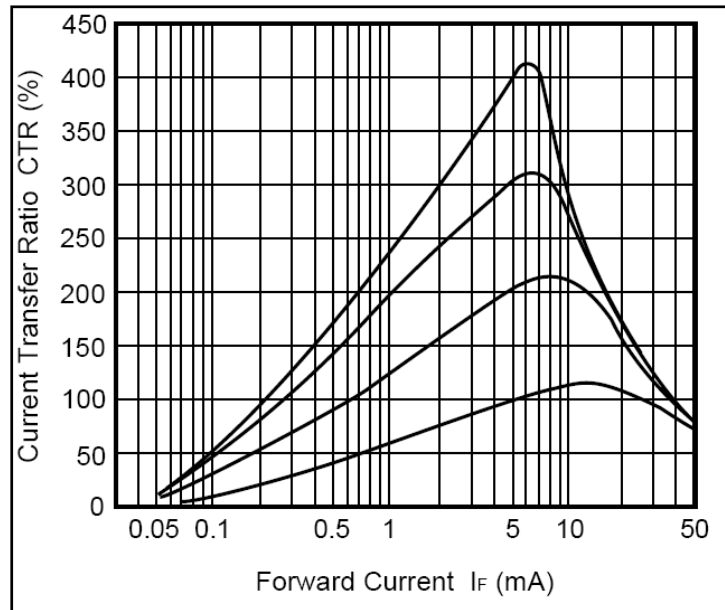


Figura 2.10: Transferencia de corriente al colector en función de la corriente directa I_F y de la temperatura ambiente.

Fuente: *Instrumentación Electrónica* [en línea] 2008 - 2009 [ref. de mayo de 2009]. Disponible en Web: <http://laimbio08.escet.urjc.es/assets/files/docencia/IE/TEMA2_filtros.pdf>

La figura 2.11 indica el circuito para el cálculo de valores de las resistencias R_6 y R_7 .

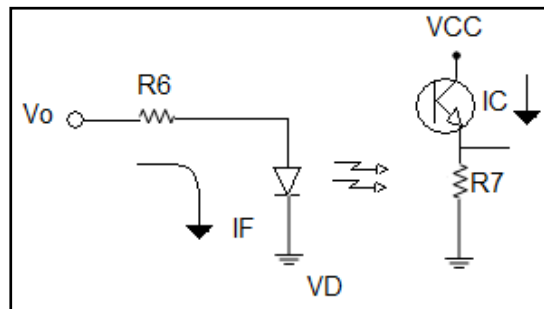


Figura 2.11: Circuito para el cálculo de R_6 y R_7 .

Fuente: Autores

Datos:

$$V_D = 1,2V$$

$$V_{CC} = 12V$$

$$V_0 \begin{cases} V_{0H} = 5V \\ V_{0L} = 0V \end{cases}$$

$$V_{CE\ sat} = 0V$$

$$I_C = 1mA$$

$$C_{TR} = 70\%$$

$$C_{TR} = \frac{I_C}{I_F} 100 \quad 2.18$$

$$I_F = \frac{I_C (100)}{C_{TR}} \quad 2.19$$

$$I_F \approx \mathbf{1,429mA}$$

$$R_6 = \frac{V_{OH} - V_D}{I_F} \quad 2.20$$

$$R_6 = \mathbf{2,659K\Omega}$$

Se toma el valor de $R_6 = 2,7K\Omega$, con lo cual, se reemplaza este valor en la ecuación 2.20 y se obtiene I_F .

$$I_F = \mathbf{1,407mA}$$

$$PR_6 = I_F^2 R_6 \quad 2.21$$

$$PR_6 = \mathbf{5,348mW}$$

$$R_6 = \frac{2,7K\Omega}{0,25W} \quad 2.22$$

$$R_7 = \frac{V_{CC} - V_{CE_{SAT}}}{I_C} \quad 2.23$$

$$R_7 = \mathbf{12K\Omega}$$

$$PR_7 = I_C^2 R_7 \quad 2.24$$

$$PR_7 = 12mW$$

$$R_7 = \frac{12K\Omega}{0,25W} \quad 2.25$$

De esta manera se tiene que para $R_6, R_8, R_{10}, R_{12}, R_{14}$ y R_{16} se escogen de valores indicados en la ecuación 2.22.

Para $R_7, R_9, R_{11}, R_{13}, R_{15}$ y R_{17} se escogen de valores indicados en la ecuación 2.25.

En los C.I. 4016B se han colocado filtros de bypass (capacitores cerámicos C_6 y C_7 de $0.1\mu F$) para eliminar el ruido eléctrico proveniente de la fuente de poder.

2.9 Conclusiones.

1. Un Microcontrolador combina la unidad central de procesamiento (CPU), la memoria y los recursos de entrada y salida en un único circuito integrado.
2. Un Microcontrolador sirve para aplicaciones puntuales, funciona sobre la base de un programa guardado en su memoria, usando los lenguajes C, BASIC o ENSAMBLADORA según se requiera.
3. Está diseñado para reducir el consumo de electricidad; incluyen un reloj (generadores en milisegundos para interrupciones) interno y diferentes tipos de memoria RAM / ROM / EPROM / EEPROM / FLASH, etc. que controlan las entradas y salidas en series analógicas o digitales e incluyen en algunos, temporizadores.
4. Los Microcontroladores de un mismo tipo forman una familia que tienen el mismo núcleo (core) y se diferencian en sus entradas, salidas y puertos de habilitación.
5. Microcontroladores PIC, fabricados por MICROCHIP tienen diferentes tipos de familias según el tipo de aplicación a utilizar.
6. Microcontroladores MICROCHIP PIC32MX mejoran el rendimiento, la conectividad, el soporte del software y sus herramientas de programación.
7. El generador de funciones tanto para señal pura, como para señal con ruido, es capaz de generar señales: senoidales, triangulares, cuadradas, diente de sierra, tren de pulsos, entre otros; de acuerdo a la configuración que se asigne y a las necesidades de estudio y aprendizaje.

CAPÍTULO III

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE DE PROGRAMACIÓN.

3.1 Introducción.

La programación tiene como objetivo el tratamiento de la información correctamente, con lo que se espera que un programa de el resultado correcto y no erróneo.

Existen lenguajes de programación que utilizan compilador; la ejecución de un programa con compilador requiere de dos etapas: Traducción del programa simbólico a código máquina y la ejecución del procesamiento de datos.

Un programa, se puede dividir en subprogramas llamados subrutinas. Una subrutina es una porción de código que forma parte de un programa más grande y realiza una tarea específica, relativamente independiente del resto del código. La mayoría de los lenguajes de programación soportan la creación de subrutinas y métodos para llamarlas y retornarlas.

Existen varias ventajas al momento de trabajar con subrutinas, las mismas que tienen que ver con la reducción de códigos duplicados, el uso del código en múltiples programas, descomposición de problemas complejos permitiendo la legibilidad del código en un programa.

Un lenguaje de programación es diseñado para describir un conjunto de acciones consecutivas que permiten crear herramientas y software que controlen el comportamiento físico y lógico de una máquina.

Un software de programación es el conjunto de herramientas que permiten al programador desarrollar programas informáticos, usando diferentes alternativas y lenguajes de programación de una manera práctica.

El proceso de creación de software puede llegar a ser muy complejo, dependiendo de su tamaño y características del mismo. Normalmente puede ser pequeño, mediano y de gran porte según su tamaño y/o costo.

El proceso de desarrollo puede involucrar numerosas y variadas tareas, pasando por lo administrativo, lo técnico y hasta la gestión y el gerenciamiento. Pero casi rigurosamente siempre se cumplen ciertas etapas mínimas que a continuación se describen.

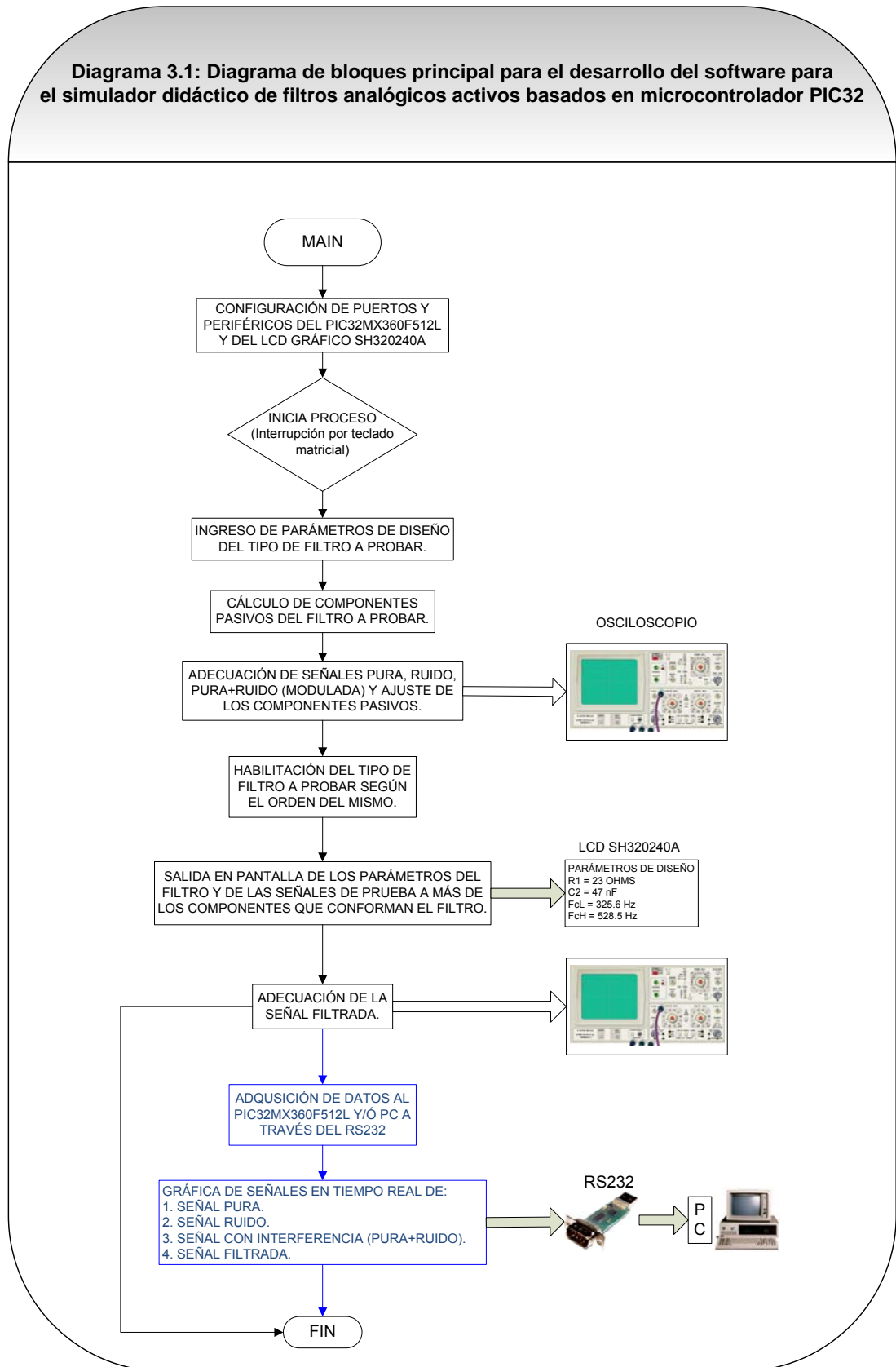
- Análisis
- Diseño.
- Implementación.
- Pruebas (unitarias y de integración).
- Instalación y paso a producción.
- Mantenimiento.

Para desarrollar un programa, de manera ordenada, clara y precisa, es necesario, utilizar cierto tipo de flujo grama o diagrama de flujo que consiste en representar gráficamente hechos, situaciones, movimientos o relaciones de todo tipo, por medio de símbolos para tener una visión conceptual de los pasos a seguir dentro de una estructura de programación.

Cada estructura de programación constará de un cuerpo principal, varias aclaraciones, y cierto tipo de lasos, que permiten el desarrollo ordenado de puntos específicos e importantes para la conclusión, sin errores, de cada subrutina.

El diagrama 3.1 indica un flujo grama general de programación y función del simulador didáctico de filtros analógicos activos basado en microcontrolador PIC32 y las subrutinas más importantes citadas para la construcción de dicho equipo.

Diagrama 3.1: Diagrama de bloques principal para el desarrollo del software para el simulador didáctico de filtros analógicos activos basados en microcontrolador PIC32

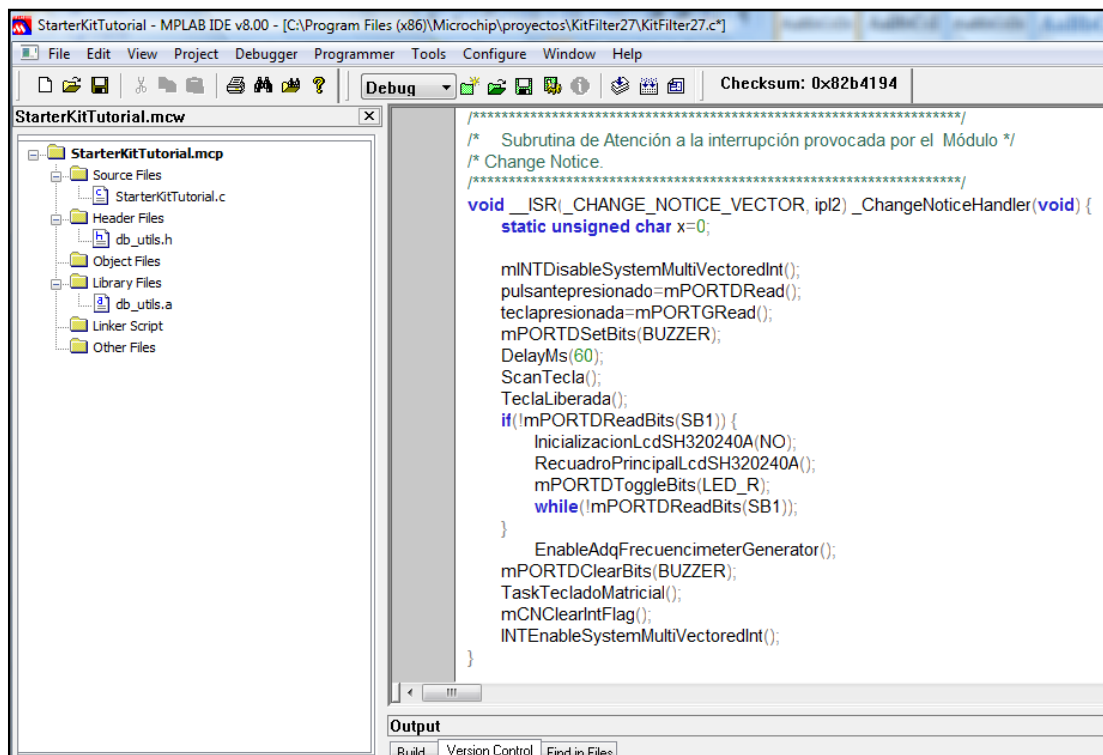


3.2 Subrutina de programación para atención a la interrupción provocada por el módulo “Change Notice”.

Las instrucciones programadas para esta subrutina son las siguientes:

1. Deshabilitar todas las interrupciones Multi Vector.
2. Leer los puertos para terminar la condición de mismatch.
3. Encender el buzzer piezoeléctrico.
4. Retardar antirebote.
5. Verificar que pulsante del Teclado Matricial ha sido presionado.
6. Esperar que libere la tecla presionada del teclado matricial.
7. Verificar que pulsante del Pic Starter Kit ha sido presionado.
8. Tarea a realizar en función de la tecla presionada.
9. Borrar la bandera de interrupción del Módulo Change Notice.
10. Habilitar todas las interrupciones Multi Vector.

La subrutina de programación se describe en la figura 3.1.



```

StarterKitTutorial - MPLAB IDE v8.00 - [C:\Program Files (x86)\Microchip\proyectos\KitFilter27\KitFilter27.c]
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug Checksum: 0x82b4194
StarterKitTutorial.mcw
  StarterKitTutorial.mcp
    Source Files
      StarterKitTutorial.c
    Header Files
      db_utils.h
    Object Files
      db_utils.a
    Library Files
    Linker Script
    Other Files

/* ***** */
/* Subrutina de Atención a la interrupción provocada por el Módulo */
/* Change Notice.
/* ***** */
void __ISR(_CHANGE_NOTICE_VECTOR, ip2) _ChangeNoticeHandler(void) {
    static unsigned char x=0;

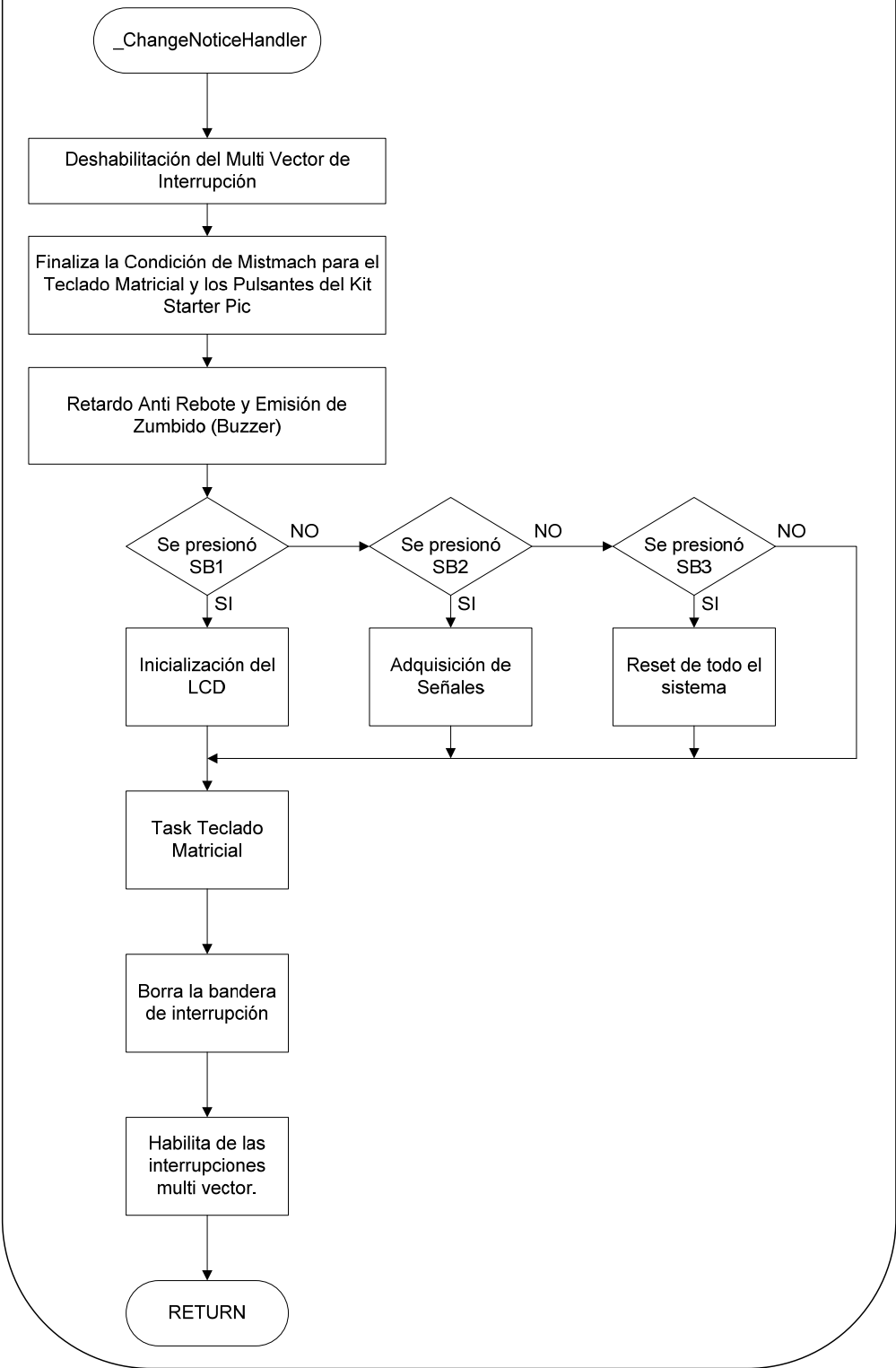
    mINTDisableSystemMultiVectoredInt();
    pulsantepresionado=mPORTDRead();
    teclapresionada=mPORTGRead();
    mPORTDSetBits(BUZZER);
    DelayMs(60);
    ScanTecla();
    TeclaLiberada();
    if(!mPORTDReadBits(SB1)) {
        InicializacionLcdSH320240A(NO);
        RecuadroPrincipalLcdSH320240A();
        mPORTDToggleBits(LED_R);
        while(!mPORTDReadBits(SB1));
    }
    EnableAdqFrecuencimetroGenerator();
    mPORTDClearBits(BUZZER);
    TaskTecladoMatricial();
    mCNClearIntFlag();
    INTEnableSystemMultiVectoredInt();
}
  
```

Figura 3.1: Subrutina de programación para atención a la interrupción provocada por el módulo “Change Notice.”

Fuente: Autores.

El diagrama 3.2 indica el flujo grama para la subrutina de atención a la interrupción provocada por el módulo “Change Notice”.

Diagrama 3.2: Subrutina de atención a la interrupción provocada por el módulo "Change Notice"



Fuente: Autores.

3.3 Subrutina de programación para escritura de un comando, parámetro, o dato en el controlador del LCD.

Las instrucciones programadas para esta subrutina son las siguientes:

1. Leer directamente el registro del puerto "b" (Enmascaramiento).
2. Mandar a 0 "cero" el pin de escritura del controlador "WR=0"
3. Preguntar si se va a ingresar un comando, parámetro ó dato.
4. Si es comando, mandar el pin A_0 del controlador a 0 "cero"
5. Si es parámetro o dato hacer que el pin A_0 del controlador esté recibiendo un 1 lógico.
6. Cargar el valor en el puerto "B".
7. Mandar a los pines CS "chipset" y WR "write/escritura" a 0 "cero"
8. Generar un retardo para cargar los valores en la memoria.
9. Cargar los pines CS "chipset" y WR "write/escritura" con un 1 lógico.
10. Salir de la subrutina "RETURN".

La subrutina de programación se describe en la figura 3.2.

```

/*****
/* Esta subrutina inicializa las variables utilizadas para el mane-*/
/* jo de datos, control y eventos del LCD SH320240A. */
*****/
void InicializacionLcdSH320240A(unsigned char clean_screen) {

    mPORTBSetBits(CS);
    mPORTBSetBits(WR);
    mPORTASetBits(RD);
    mPORTASetBits(RST);
    mPORTCClearBits(Ao);

    mPORTCClearBits(DISPOFF);

    WriteCommandLcdSH320240A(CMD_SYSTEM_SET);
    WriteParameterLcdSH320240A(P1_SYSTEM_SET);
    WriteParameterLcdSH320240A(P2_SYSTEM_SET);
    WriteParameterLcdSH320240A(P3_SYSTEM_SET);
    WriteParameterLcdSH320240A(P4_SYSTEM_SET);
    WriteParameterLcdSH320240A(P5_SYSTEM_SET);
    WriteParameterLcdSH320240A(P6_SYSTEM_SET);
    WriteParameterLcdSH320240A(P7_SYSTEM_SET);
    WriteParameterLcdSH320240A(P8_SYSTEM_SET);
    WriteCommandLcdSH320240A(CMD_SCROLL);
    WriteParameterLcdSH320240A(P1_SCROLL);
    WriteParameterLcdSH320240A(P2_SCROLL);
    WriteParameterLcdSH320240A(P3_SCROLL);
    WriteParameterLcdSH320240A(P4_SCROLL);
    WriteParameterLcdSH320240A(P5_SCROLL);
    WriteParameterLcdSH320240A(P6_SCROLL);
    WriteParameterLcdSH320240A(P7_SCROLL);
    WriteParameterLcdSH320240A(P8_SCROLL);
    WriteParameterLcdSH320240A(P9_SCROLL);
    WriteParameterLcdSH320240A(P10_SCROLL);
    WriteCommandLcdSH320240A(CMD_HDOT_SCR);
    WriteParameterLcdSH320240A(P1_HDOT_SCR);
    WriteCommandLcdSH320240A(CMD_OVLAY);
    WriteParameterLcdSH320240A(P1_OVLAY);
    WriteCommandLcdSH320240A(CMD_DISP_OFF);
    WriteParameterLcdSH320240A(P1_DISP_ON);
    if(clean_screen==YES) {

        Clear1eraPantallaTextoLcdSH320240A();

        Clear2ndaPantallaGraficaLcdSH320240A();

    }

    WriteCommandLcdSH320240A(CMD_CSR_FORM);
    WriteParameterLcdSH320240A(P1_CSR_FORM);
    WriteParameterLcdSH320240A(P2_CSR_FORM);
    WriteCommandLcdSH320240A(CMD_CSRDIR_RIGHT);
    WriteCommandLcdSH320240A(CMD_DISP_ON);
    WriteParameterLcdSH320240A(P1_DISP_ON);
    mPORTCSetBits(DISPOFF);

}

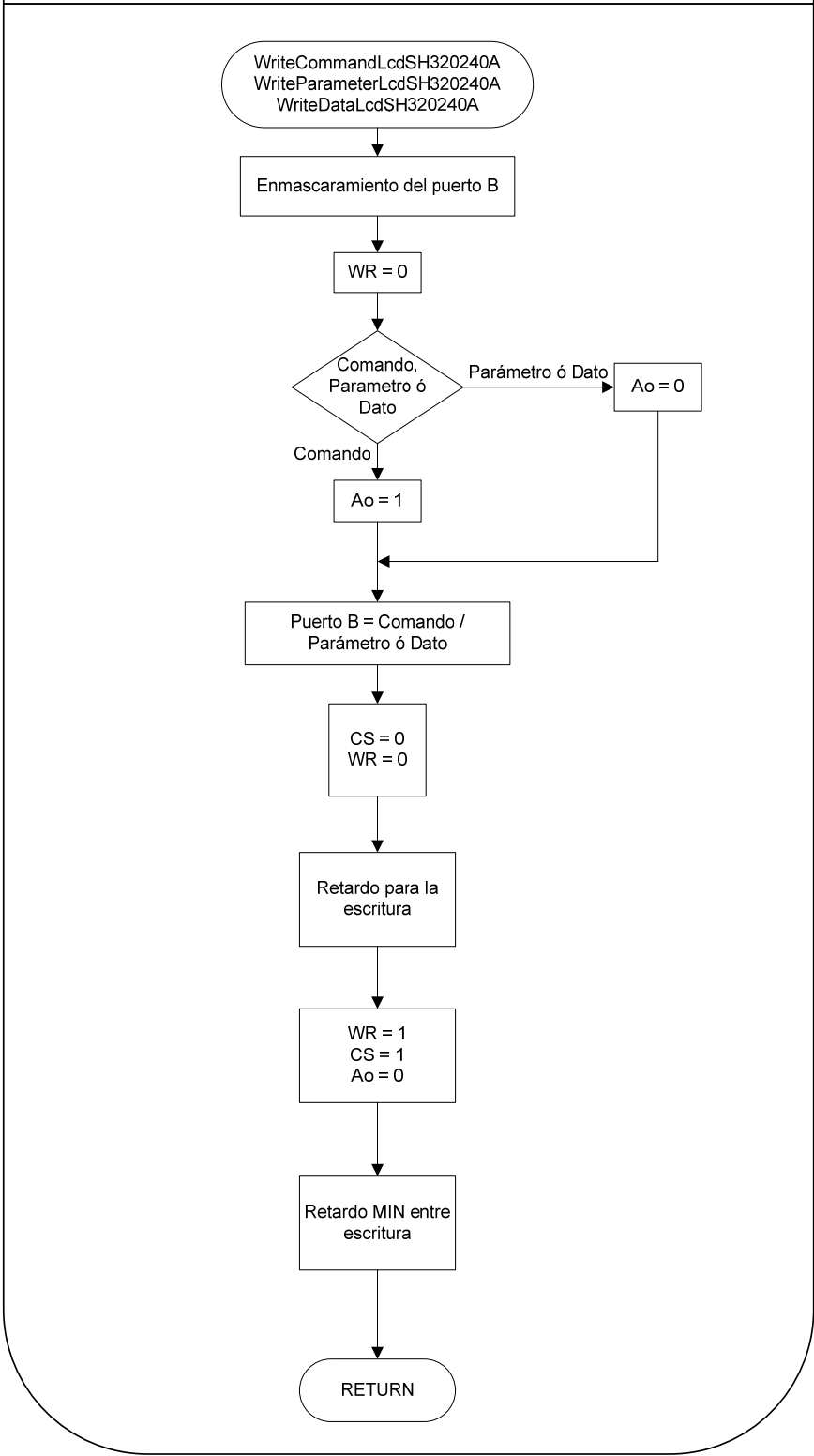
```

Figura 3.2: Subrutina de programación para escritura de un comando, parámetro, o dato en el controlador del LCD.

Fuente: Autores.

El diagrama 3.3 indica el flujo grama para la subrutina de programación para escritura de un comando, parámetro o dato en el controlador del LCD.

Diagrama 3.3: Subrutina para escritura de un comando, parámetro o dato en el controlador del LCD.



Fuente: Autores.

3.4 Subrutina de programación para la inicialización del LCD.

Las instrucciones programadas para esta subrutina son las siguientes:

1. Deshabilitar todas las líneas de control del LCD, CS, WR, RD, RST, Ao: Las señales de reset RST, habilitación del chip CS, escritura WR y lectura RD son activas a nivel lógico 0, por lo tanto inicialmente todas estas líneas se encuentran activadas, razón por la cual se deshabilitan.
2. Retardo para estabilización de la fuente.
3. Escribir en la memoria de programa del LCD SH320240A, todos los comandos y parámetros que van a ser utilizados.
4. Limpiar la pantalla al finalizar el ingreso de los datos para que esté lista y limpia al comenzar a ejecutar el programa.

La subrutina programación se describe en la figura 3.3.

```

StarterKitTutorial - MPLAB IDE v8.00 - [C:\Program Files (x86)\Microchip\proyectos\KitFilter27\KitFilter27.c*]
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug Checksum: 0x82b4194
StarterKitTutorial.mcp
  StarterKitTutorial.mcp
    Source Files
      StarterKitTutorial.c
    Header Files
      db_utils.h
    Object Files
    Library Files
      db_utils.a
    Linker Script
    Other Files
}

/*****
/* Esta subrutina inicializa las variables utilizadas para el mane- */
/* jo de datos, control y eventos del LCD SH320240A. */
*****/

void InicializacionLcdSH320240A(unsigned char clean_screen) {

    mPORTBSetBits(CS);
    mPORTBSetBits(WR);
    mPORTASetBits(RD);
    mPORTASetBits(RST);
    mPORTCClearBits(Ao);

    mPORTCClearBits(DISPOFF);

    WriteCommandLcdSH320240A(CMD_SYSTEM_SET);
    WriteParameterLcdSH320240A(P1_SYSTEM_SET);
    WriteParameterLcdSH320240A(P2_SYSTEM_SET);
    WriteParameterLcdSH320240A(P3_SYSTEM_SET);
    WriteParameterLcdSH320240A(P4_SYSTEM_SET);
    WriteParameterLcdSH320240A(P5_SYSTEM_SET);
    WriteParameterLcdSH320240A(P6_SYSTEM_SET);
    WriteParameterLcdSH320240A(P7_SYSTEM_SET);
    WriteParameterLcdSH320240A(P8_SYSTEM_SET);
    WriteCommandLcdSH320240A(CMD_SCROLL);
    WriteParameterLcdSH320240A(P1_SCROLL);
    WriteParameterLcdSH320240A(P2_SCROLL);
    WriteParameterLcdSH320240A(P3_SCROLL);
    WriteParameterLcdSH320240A(P4_SCROLL);
    WriteParameterLcdSH320240A(P5_SCROLL);
    WriteParameterLcdSH320240A(P6_SCROLL);
    WriteParameterLcdSH320240A(P7_SCROLL);
    WriteParameterLcdSH320240A(P8_SCROLL);
    WriteParameterLcdSH320240A(P9_SCROLL);
    WriteParameterLcdSH320240A(P10_SCROLL);
    WriteCommandLcdSH320240A(CMD_HDOT_SCR);
    WriteParameterLcdSH320240A(P1_HDOT_SCR);
    WriteCommandLcdSH320240A(CMD_OVLAY);
    WriteParameterLcdSH320240A(P1_OVLAY);
    WriteCommandLcdSH320240A(CMD_DISP_OFF);
    WriteParameterLcdSH320240A(P1_DISP_ON);
    Clear1eraPantallaTextoLcdSH320240A();
    Clear2ndaPantallaGraficaLcdSH320240A();
}

    WriteCommandLcdSH320240A(CMD_CSR_FORM);
    WriteParameterLcdSH320240A(P1_CSR_FORM);
    WriteParameterLcdSH320240A(P2_CSR_FORM);
    WriteCommandLcdSH320240A(CMD_CSRDIR_RIGHT);
    WriteCommandLcdSH320240A(CMD_DISP_ON);
    WriteParameterLcdSH320240A(P1_DISP_ON);
    mPORTCSetBits(DISPOFF);
}

Output
Build Version Control Find in Files
PIC32MX360F512L Ln 2516, Col 51 INS WR

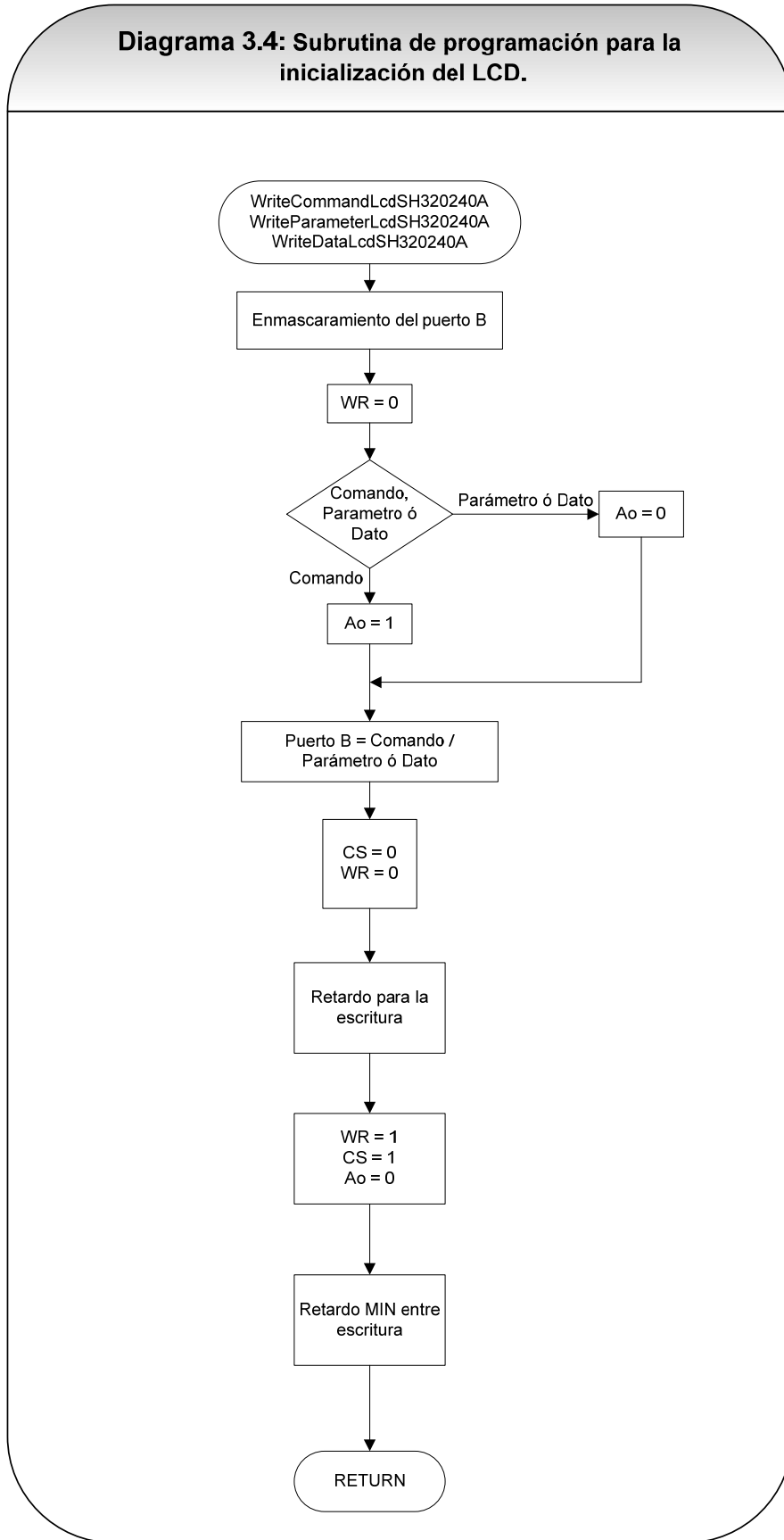
```

Figura 3.3: Subrutina de programación para la inicialización del LCD.

Fuente: Autores.

El diagrama 3.4 indica el flujo grama para la subrutina de programación para la inicialización del LCD.

Diagrama 3.4: Subrutina de programación para la inicialización del LCD.



Fuente: Autores.

3.5 Subrutina de programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden.

Las instrucciones programadas para esta subrutina son las siguientes.

1. Poner en cero todos los valores de resistencias y condensadores.
2. Limpiar los espacios de visualización del display de los valores de los elementos pasivos.
3. Seleccionar el tipo de filtro con el que se va a trabajar: Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda o Pasa Todo.
 - 3.1 Seleccionar el orden del filtro: 2^{do} orden, 4^{to} orden, 6^{to} orden y 8^{vo} orden.
 - 3.2 En caso de escoger un filtro de 2^{do} orden se ingresan valores para los componentes pasivos.
 - 3.3 En caso de no escoger el filtro de 2^{do} orden, se optará por la opción de 4^{to} orden y se ingresan valores para los componentes pasivos.
 - 3.4 Si no se desea un filtro de 4^{to} orden, se elige la opción de 6^{to} orden y se ingresan valores para los componentes pasivos.
 - 3.5 Si no se desea un filtro de 6^{to} orden, se cargan al sistema valores de componentes pasivos para un filtro de 8^{vo} orden.
4. Seleccionar el capacitor C_1 según la frecuencia de corte ingresada.
5. Calcular el valor del capacitor C_2
6. Calcular el valor de las resistencias R_1 y R_2 .
7. Re calcular los parámetros de la ecuación característica.
8. Cargar los valores de los componentes pasivos para ser mostradas en el LCD.

Una vez seleccionado el tipo de filtro en el ítem 3, para cualquiera de ellos (Pasa Alto, Pasa Bajo, Pasa Banda, Elimina Banda O Pasa Todo) se repetirá el mismo procedimiento desde el ítem 3.1 hasta el ítem 8 manteniendo la estructura de datos para un filtro pasa bajo, pasa alto, pasa banda, elimina banda o pasa todo, de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden.

La subrutina programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden se describen en las figuras 3.4.a, 3.4.b, 3.4.c, 3.4.d, 3.4.e y 3.4.f.

```

=====*)
/*****
/* Esta subrutina calcula los valores de los elementos pasivos de - */
/* los diferentes Tipos de filtros activos */
/*****
//void __attribute__((mips16)) CalculoElementosPasivos(void) {
void CalculoElementosPasivos(void) {
    unsigned char i, j;
    for(i=0; i<4; i++) {
        R1[i]=0;
        R2[i]=0;
        R3[i]=0;
        R4[i]=0;
        Cq1[i]=0;
        Cq2[i]=0;
        for(j=0; j<8; j++) {
            PDF_R1[i][j]='';
            PDF_R2[i][j]='';
            PDF_R3[i][j]='';
            PDF_R4[i][j]='';
        }
    }
    R4min=0;
    if(PDF_FilTip==T_FPB) {
        for(i=0; i<=(PDF_FilOrd-1); i++) {
            switch(PDF_FilOrd) {
                case O_2DO:
                    a1=CFA Ctes_2do_Orden[i][0]((PDF_FilApr-1));
                    b1=CFA Ctes_2do_Orden[i][1]((PDF_FilApr-1));
                    fci=PDF_FilFcLow*CFA Ctes_2do_Orden[i][2]((PDF_FilApr-1));
                    break;
                case O_4TO:
                    a1=CFA Ctes_4to_Orden[i][0]((PDF_FilApr-1));
                    b1=CFA Ctes_4to_Orden[i][1]((PDF_FilApr-1));
                    fci=PDF_FilFcLow*CFA Ctes_4to_Orden[i][2]((PDF_FilApr-1));
                    break;
                case O_6TO:
                    a1=CFA Ctes_6to_Orden[i][0]((PDF_FilApr-1));
                    b1=CFA Ctes_6to_Orden[i][1]((PDF_FilApr-1));
                    fci=PDF_FilFcLow*CFA Ctes_6to_Orden[i][2]((PDF_FilApr-1));
                    break;
                case O_8VO:
                    a1=CFA Ctes_8vo_Orden[i][0]((PDF_FilApr-1));
                    b1=CFA Ctes_8vo_Orden[i][1]((PDF_FilApr-1));
                    fci=PDF_FilFcLow*CFA Ctes_8vo_Orden[i][2]((PDF_FilApr-1));
                    break;
            }
        }
        if(PDF_FilApr>=A_TSC_5dB && PDF_FilApr<=A_BES) {
            if(fci<=1)
                Cq1[i]=C[30];
            if(fci>1 && fci<=2)
                Cq1[i]=C[28];
            if(fci>2 && fci<=5)
                Cq1[i]=C[26];
            if(fci>5 && fci<=10)
                Cq1[i]=C[24];
            if(fci>10 && fci<=20)
                Cq1[i]=C[22];
            if(fci>20 && fci<=50)
                Cq1[i]=C[20];
            if(fci>50 && fci<=100)
                Cq1[i]=C[18];
            if(fci>100 && fci<=200)
                Cq1[i]=C[16];
            if(fci>200 && fci<=500)
                Cq1[i]=C[14];
            if(fci>500 && fci<=1000)
                Cq1[i]=C[12];
            if(fci>1000 && fci<=2000)
                Cq1[i]=C[10];
            if(fci>2000 && fci<=5000)
                Cq1[i]=C[8];
            if(fci>5000 && fci<=10000)
                Cq1[i]=C[6];
            if(fci>10000 && fci<=20000)
                Cq1[i]=C[4];
        }
    }
}
=====*)

```

Output window: PIC32MX360F512L, Ln 1847, Col 1, INS WR

Figura 3.4.a: Subrutina de programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden.

Fuente: Autores.

```

StarterKitTutorial - MPLAB IDE v8.00 - [C:\Program Files (x86)\Microchip\projectos\KitFilter27\KitFilter27.c]
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug Checksum: 0x82b4194

StarterKitTutorial.mcw
├── StarterKitTutorial.mcw
├── Source Files
│   └── StarterKitTutorial.c
├── Header Files
│   └── db_utils.h
├── Object Files
├── Library Files
│   └── db_utils.a
├── Linker Script
└── Other Files

    if(fci>20000 && fci<=50000)
        Cq1[i]=C[2];
    if(fci>50000)
        Cq1[i]=C[0];
    }
    Cq2[i]=(4*b1/(a1*a1))*Cq1[i];
    j=0;
    while(Cq2[i]>=C[j] && j<49)
        j++;
    Cq2[i]=C[j];
    R1[i]=(a1*Cq2[i]-sqrt(a1*a1*Cq2[i]*Cq2[i]-4*b1*Cq1[i]*Cq2[i]))/(4*PI*fci*Cq1[i]*Cq2[i]);
    R2[i]=(a1*Cq2[i]+sqrt(a1*a1*Cq2[i]*Cq2[i]-4*b1*Cq1[i]*Cq2[i]))/(4*PI*fci*Cq1[i]*Cq2[i]);
    a[i]=Cq1[i]*(R1[i]+R2[i]);
    b[i]=R1[i]*R2[i]*Cq1[i]*Cq2[i];
    }
    switch(PDF_FilOrd) {
        case O_2DO:

            break;
        case O_4TO:

            break;
        case O_6TO:

            break;
        case O_8VO:

            break;
    }
}
if(PDF_FilTip==T_FPA) {
    for(i=0, i<=(PDF_FilOrd-1); i++) {
        switch(PDF_FilOrd) {
            case O_2DO:
                a1=CFA Ctes_2do_Orden[i][0][(PDF_FilApr-1)];
                b1=CFA Ctes_2do_Orden[i][1][(PDF_FilApr-1)];
                fci=PDF_FilFcHigh*CFA Ctes_2do_Orden[i][2][(PDF_FilApr-1)];
                break;
            case O_4TO:
                a1=CFA Ctes_4to_Orden[i][0][(PDF_FilApr-1)];
                b1=CFA Ctes_4to_Orden[i][1][(PDF_FilApr-1)];
                fci=PDF_FilFcHigh*CFA Ctes_4to_Orden[i][2][(PDF_FilApr-1)];
                break;
            case O_6TO:
                a1=CFA Ctes_6to_Orden[i][0][(PDF_FilApr-1)];
                b1=CFA Ctes_6to_Orden[i][1][(PDF_FilApr-1)];
                fci=PDF_FilFcHigh*CFA Ctes_6to_Orden[i][2][(PDF_FilApr-1)];
                break;
            case O_8VO:
                a1=CFA Ctes_8vo_Orden[i][0][(PDF_FilApr-1)];
                b1=CFA Ctes_8vo_Orden[i][1][(PDF_FilApr-1)];
                fci=PDF_FilFcHigh*CFA Ctes_8vo_Orden[i][2][(PDF_FilApr-1)];
                break;
        }
    }
    if(PDF_FilApr>=A_TSC_5dB && PDF_FilApr<=A_BES) {
        if(fci<=1)
    }
    if(PDF_FilApr>=A_TSC_5dB && PDF_FilApr<=A_BES) {
        if(fci<=1)
            Cq1[i]=C[35];
        if(fci>1 && fci<=2)
            Cq1[i]=C[33];
        if(fci>2 && fci<=5)
            Cq1[i]=C[31];
        if(fci>5 && fci<=10)
            Cq1[i]=C[29];
        if(fci>10 && fci<=20)
            Cq1[i]=C[27];
        if(fci>20 && fci<=50)
            Cq1[i]=C[25];
        if(fci>50 && fci<=100)
            Cq1[i]=C[23];
        if(fci>100 && fci<=200)
            Cq1[i]=C[21];
        if(fci>200 && fci<=500)
            Cq1[i]=C[19];
    }
}
Files Symbols Output

```

Figura 3.4.b: Subrutina de programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden.

Fuente: Autores.

```

StarterKitTutorial - MPLAB IDE v8.00 - [C:\Program Files (x86)\Microchip\projectos\KitFilter27\KitFilter27.c]
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug Checksum: 0x82b4194

StarterKitTutorial.mcw
  StarterKitTutorial.mcp
    Source Files
    StarterKitTutorial.c
    Header Files
    db_utils.h
    Object Files
    Library Files
    db_utils.a
    Linker Script
    Other Files

if(fci>500 && fci<=1000)
  Cq1[i]=C[17];
if(fci>1000 && fci<=2000)
  Cq1[i]=C[15];
if(fci>2000 && fci<=5000)
  Cq1[i]=C[13];
if(fci>5000 && fci<=10000)
  Cq1[i]=C[11];
if(fci>10000 && fci<=20000)
  Cq1[i]=C[9];
if(fci>20000 && fci<=50000)
  Cq1[i]=C[7];
if(fci>50000)
  Cq1[i]=C[5];
}
Cq2[i]=Cq1[i];
R1[i]=1/(PI*fci*Cq1[i]*a1);
R2[i]=a1/(4*PI*fci*Cq1[i]*b1);
aij=2*(Cq1[i]*R1[i]);
b[i]=1/(pow(Cq1[i],2)*R1[i]*R2[i]);
}
switch(PDF_FilOrd) {
  case O_2DO:

    break;
  case O_4TO:

    break;
  case O_6TO:

    break;
  case O_8VO:

    break;
}
if(PDF_FilTip==T_FPbN) {
  PDF_FilFn=sqrt(PDF_FilFcLow*PDF_FilFcHigh);
  for(i=0; i<=(PDF_FilOrd-1); i++) {
    switch(PDF_FilOrd) {
      case O_2DO:
        a1=CFA_Ctes_2do_Orden[i][0]((PDF_FilApr-1));
        b1=CFA_Ctes_2do_Orden[i][1]((PDF_FilApr-1));
        fci=PDF_FilFn*CFA_Ctes_2do_Orden[i][2]((PDF_FilApr-1));
        break;
      case O_4TO:
        a1=CFA_Ctes_4to_Orden[i][0]((PDF_FilApr-1));
        b1=CFA_Ctes_4to_Orden[i][1]((PDF_FilApr-1));
        fci=PDF_FilFn*CFA_Ctes_4to_Orden[i][2]((PDF_FilApr-1));
        break;
      case O_6TO:
        a1=CFA_Ctes_6to_Orden[i][0]((PDF_FilApr-1));
        b1=CFA_Ctes_6to_Orden[i][1]((PDF_FilApr-1));
        fci=PDF_FilFn*CFA_Ctes_6to_Orden[i][2]((PDF_FilApr-1));
        break;
      case O_8VO:
        a1=CFA_Ctes_8vo_Orden[i][0]((PDF_FilApr-1));
        b1=CFA_Ctes_8vo_Orden[i][1]((PDF_FilApr-1));
        fci=PDF_FilFn*CFA_Ctes_8vo_Orden[i][2]((PDF_FilApr-1));
        break;
    }
  }
if(PDF_FilApr>=A_TSC_5dB && PDF_FilApr<=A_BES) {
  if(fci<=1)
    Cq1[i]=C[35];
  if(fci>1 && fci<=2)
    Cq1[i]=C[33];
  if(fci>2 && fci<=5)
    Cq1[i]=C[31];
  if(fci>5 && fci<=10)
    Cq1[i]=C[29];
  if(fci>10 && fci<=20)
    Cq1[i]=C[27];
  if(fci>20 && fci<=50)
    Cq1[i]=C[25];
  if(fci>50 && fci<=100)
    Cq1[i]=C[23];
}
}

```

PIC32MX360F512L Ln 1847, Col 1 IN

Figura 3.4.c: Subrutina de programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden.

Fuente: Autores.

```

StarterKitTutorial - MPLAB IDE v8.00 - [C:\Program Files (x86)\Microchip\proyectos\KitFilter27\KitFilter27.c*]
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug Checksum: 0x82b4194
StarterKitTutorial.mcw
StarterKitTutorial.mcp
Source Files
StarterKitTutorial.c
Header Files
db_utils.h
Object Files
Library Files
db_utils.a
Linker Script
Other Files

if(fci>100 && fci<=200)
    Cq1[i]=C[21];
if(fci>200 && fci<=500)
    Cq1[i]=C[19];
if(fci>500 && fci<=1000)
    Cq1[i]=C[17];
if(fci>1000 && fci<=2000)
    Cq1[i]=C[15];
if(fci>2000 && fci<=5000)
    Cq1[i]=C[13];
if(fci>5000 && fci<=10000)
    Cq1[i]=C[11];
if(fci>10000 && fci<=20000)
    Cq1[i]=C[9];
if(fci>20000 && fci<=50000)
    Cq1[i]=C[7];
if(fci>50000)
    Cq1[i]=C[5];
}
Cq2[i]=Cq1[i];
alfa=1-(pow(a1,2)/(8*b1));
R1[i]=a1/(8*PI*fci*alfa*Cq1[i]);
R2[i]=(2*b1)/(PI*fci*a1*Cq1[i]);
R3[i]=(alfa*R1[i])/(1-alfa);
a[i]=(R2[i]*Cq2[i]*(1-alfa)+alfa*R1[i]*(Cq1[i]+Cq2[i]));
b[i]=pow(Cq1[i],2)*alfa*R1[i]*R2[i];
c[i]=(1-alfa)*R2[i]*Cq2[i];
}
switch(PDF_FilOrd) {
    case O_2DO:
        break;
    case O_4TO:
        break;
    case O_6TO:
        break;
    case O_8VO:
        break;
}
}
if(PDF_FilTip==T_FEBn) {
    PDF_FilFn=sqrt(PDF_FilFcLow*PDF_FilFcHigh);
    for(i=0; i<=(PDF_FilOrd-1); i++) {
        switch(PDF_FilOrd) {
            case O_2DO:
                a1=CFA_Ctes_2do_Orden[i][0]((PDF_FilApr-1));
                b1=CFA_Ctes_2do_Orden[i][1]((PDF_FilApr-1));
                fci=PDF_FilFn*CFA_Ctes_2do_Orden[i][2]((PDF_FilApr-1));
                R4min=CFA_Ctes_2do_Orden[i][4]((PDF_FilApr-1));
                break;
            case O_4TO:
                a1=CFA_Ctes_4to_Orden[i][0]((PDF_FilApr-1));
                b1=CFA_Ctes_4to_Orden[i][1]((PDF_FilApr-1));
                fci=PDF_FilFn*CFA_Ctes_4to_Orden[i][2]((PDF_FilApr-1));
                R4min=CFA_Ctes_4to_Orden[i][4]((PDF_FilApr-1));
                break;
            case O_6TO:
                a1=CFA_Ctes_6to_Orden[i][0]((PDF_FilApr-1));
                b1=CFA_Ctes_6to_Orden[i][1]((PDF_FilApr-1));
                fci=PDF_FilFn*CFA_Ctes_6to_Orden[i][2]((PDF_FilApr-1));
                R4min=CFA_Ctes_6to_Orden[i][4]((PDF_FilApr-1));
                break;
            case O_8VO:
                a1=CFA_Ctes_8vo_Orden[i][0]((PDF_FilApr-1));
                b1=CFA_Ctes_8vo_Orden[i][1]((PDF_FilApr-1));
                fci=PDF_FilFn*CFA_Ctes_8vo_Orden[i][2]((PDF_FilApr-1));
                R4min=CFA_Ctes_8vo_Orden[i][4]((PDF_FilApr-1));
                break;
        }
    }
}
if(PDF_FilApr>=A_TSC_5dB && PDF_FilApr<=A_BES) {
    if(fci<=1)
        Cq2[i]=C[30];
}
}
PIC32MX360F512L Ln1847, Col1 INS WR

```

Figura 3.4.d: Subrutina de programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden.

Fuente: Autores.

```

StarterKitTutorial - MPLAB IDE v8.00 - [C:\Program Files (x86)\Microchip\proyectos\KitFilter27\KitFilter27.c]
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug Checksum: 0x82b4194
StarterKitTutorial.mcw
  StarterKitTutorial.mcp
    Source Files
    StarterKitTutorial.c
    Header Files
    db_utils.h
    Object Files
    Library Files
    db_utils.a
    Linker Script
    Other Files

if(fci>1 && fci<=2)
  Cq2[i]=C[28];
if(fci>2 && fci<=5)
  Cq2[i]=C[26];
if(fci>5 && fci<=10)
  Cq2[i]=C[24];
if(fci>10 && fci<=20)
  Cq2[i]=C[22];
if(fci>20 && fci<=50)
  Cq2[i]=C[20];
if(fci>50 && fci<=100)
  Cq2[i]=C[18];
if(fci>100 && fci<=200)
  Cq2[i]=C[16];
if(fci>200 && fci<=500)
  Cq2[i]=C[14];
if(fci>500 && fci<=1000)
  Cq2[i]=C[12];
if(fci>1000 && fci<=2000)
  Cq2[i]=C[10];
if(fci>2000 && fci<=5000)
  Cq2[i]=C[8];
if(fci>5000 && fci<=10000)
  Cq2[i]=C[6];
if(fci>10000 && fci<=20000)
  Cq2[i]=C[4];
if(fci>20000 && fci<=50000)
  Cq2[i]=C[2];
if(fci>50000)
  Cq2[i]=C[0];
}
Cq1[i]=(4*b1/(a1*a1))*Cq2[i];
j=0;
while(Cq1[i]>=C[j] && j<49)
  j++;
Cq1[i]=C[j];
R1[i]=(a1*Cq1[i]-sqrt(a1*a1*Cq1[i]*Cq1[i]-4*b1*Cq1[i]*Cq2[i]))/(4*PI*fci*Cq1[i]*Cq2[i]);
R2[i]=(a1*Cq1[i]+sqrt(a1*a1*Cq1[i]*Cq1[i]-4*b1*Cq1[i]*Cq2[i]))/(4*PI*fci*Cq1[i]*Cq2[i]);
a[i]=Cq2[i]*(R1[i]+R2[i]);
b[i]=R1[i]*R2[i]*Cq1[i]*Cq2[i];
R4[i]=R4min*1000;
R3[i]=R4[i]*((1+(R2[i]/R1[i]))*(Cq2[i]/Cq1[i]));
}
switch(PDF_FilOrd) {
  case O_2DO:
    break;
  case O_4TO:
    break;
  case O_6TO:
    break;
  case O_8VO:
    break;
}
if(PDF_FilTip==T_FPT) {
  PDF_FilFn=1000;
  for(i=0; i<=(PDF_FilOrd-1); i++) {
    switch(PDF_FilOrd) {
      case O_2DO:
        a1=CFA_Ctes_2do_Orden[i][0]((PDF_FilApr-1));
        b1=CFA_Ctes_2do_Orden[i][1]((PDF_FilApr-1));
        fci=PDF_FilFn*CFA_Ctes_2do_Orden[i][2]((PDF_FilApr-1));
        R4min=CFA_Ctes_2do_Orden[i][4]((PDF_FilApr-1));
        break;
      case O_4TO:
        a1=CFA_Ctes_4to_Orden[i][0]((PDF_FilApr-1));
        b1=CFA_Ctes_4to_Orden[i][1]((PDF_FilApr-1));
        fci=PDF_FilFn*CFA_Ctes_4to_Orden[i][2]((PDF_FilApr-1));
        R4min=CFA_Ctes_4to_Orden[i][4]((PDF_FilApr-1));
        break;
      case O_6TO:
        a1=CFA_Ctes_6to_Orden[i][0]((PDF_FilApr-1));
        b1=CFA_Ctes_6to_Orden[i][1]((PDF_FilApr-1));
    }
  }
}

```

Figura 3.4.e: Subrutina de programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden.

Fuente: Autores.

```

StarterKitTutorial - MPLAB IDE v8.00 - [C:\Program Files (x86)\Microchip\proyectos\KitFilter27\KitFilter27.c]
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug Checksum: 0x82b4194

StarterKitTutorial.mcw
Source Files
StarterKitTutorial.c
Header Files
db_utls.h
Object Files
Library Files
db_utls.a
Linker Script
Other Files

fci=PDF_FilFn*CFA Ctes_6to_Orden[i][2]((PDF_FilApr-1));
R4min=CFA Ctes_6to_Orden[i][4]((PDF_FilApr-1));
break;
case O_8VO:
a1=CFA Ctes_8vo_Orden[i][0]((PDF_FilApr-1));
b1=CFA Ctes_8vo_Orden[i][1]((PDF_FilApr-1));
fci=PDF_FilFn*CFA Ctes_8vo_Orden[i][2]((PDF_FilApr-1));
R4min=CFA Ctes_8vo_Orden[i][4]((PDF_FilApr-1));
break;
}
if(PDF_FilApr>=A_TSC_5dB && PDF_FilApr<=A_BES) {
if(fci<=1)
Cq2[i]=C[30];
if(fci>1 && fci<=2)
Cq2[i]=C[28];
if(fci>2 && fci<=5)
Cq2[i]=C[26];
if(fci>5 && fci<=10)
Cq2[i]=C[24];
if(fci>10 && fci<=20)
Cq2[i]=C[22];
if(fci>20 && fci<=50)
Cq2[i]=C[20];
if(fci>50 && fci<=100)
Cq2[i]=C[18];
if(fci>100 && fci<=200)
Cq2[i]=C[16];
if(fci>200 && fci<=500)
Cq2[i]=C[14];
if(fci>500 && fci<=1000)
Cq2[i]=C[12];
if(fci>1000 && fci<=2000)
Cq2[i]=C[10];
if(fci>2000 && fci<=5000)
Cq2[i]=C[8];
if(fci>5000 && fci<=10000)
Cq2[i]=C[6];
if(fci>10000 && fci<=20000)
Cq2[i]=C[4];
if(fci>20000 && fci<=50000)
Cq2[i]=C[2];
if(fci>50000)
Cq2[i]=C[0];
}
Cq1[i]=(4*b1/(a1*a1))*Cq2[i];
j=0;
while(Cq1[i]>=C[j] && j<49)
j++;
Cq1[i]=C[j];
R1[i]=(a1*Cq1[i]-sqrt(a1*a1*Cq1[i]*Cq1[i]-4*b1*Cq1[i]*Cq2[i]))/(4*PI*fci*Cq1[i]*Cq2[i]);
R2[i]=(a1*Cq1[i]+sqrt(a1*a1*Cq1[i]*Cq1[i]-4*b1*Cq1[i]*Cq2[i]))/(4*PI*fci*Cq1[i]*Cq2[i]);
a[i]=Cq2[i]*(R1[i]+R2[i]);
b[i]=R1[i]*R2[i]*Cq1[i]*Cq2[i];
R4[i]=R4min*1000;
R3[i]=2*R4[i]*((1+(R2[i]/R1[i]))*(Cq2[i]/Cq1[i]));
}
switch(PDF_HilOrd) {
case O_2DO:
break;
case O_4TO:
break;
case O_6TO:
break;
case O_8VO:
break;
}
}
}

```

Output

PIC32MX360F512L Ln 1847, Col 1 INS WR

Figura 3.4.f: Subrutina de programación para el cálculo de elementos pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda y Pasa Todo de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden.

Fuente: Autores.

Los diagramas 3.5(a), 3.5(b), 3.5(c), 3.5(d), 3.5(e), 3.5(f), 3.5(g), 3.5(h), 3.5(i), 3.5(j), indican los flujo gramas de la subrutina para el cálculo de elementos

pasivos para un filtro Pasa Bajo, Pasa Alto, Pasa Banda, Elimina Banda Y Pasa Todo de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden.

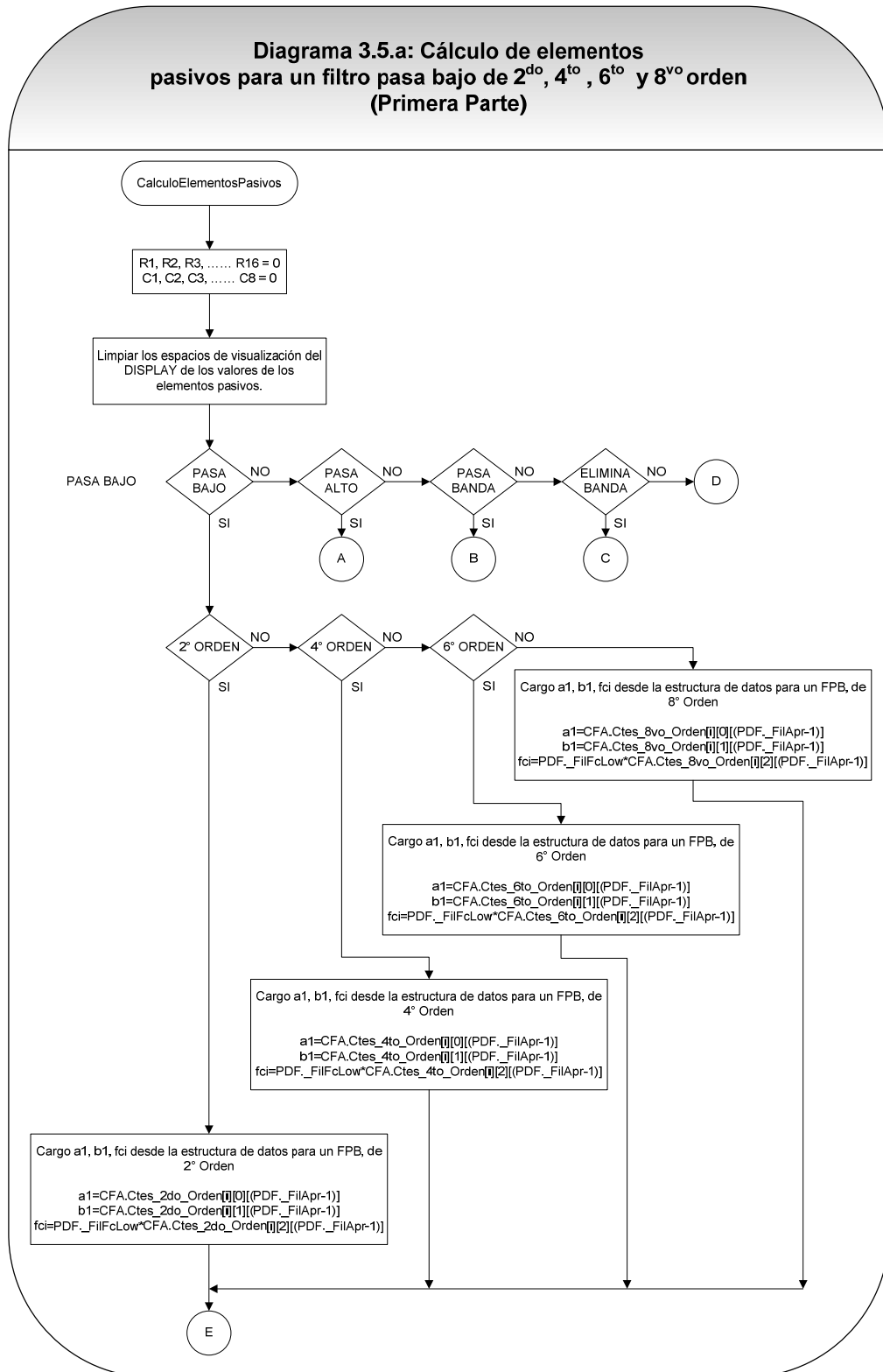


Diagrama 3.5.b: Cálculo de elementos pasivos para un filtro pasa bajo de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden (Segunda Parte)

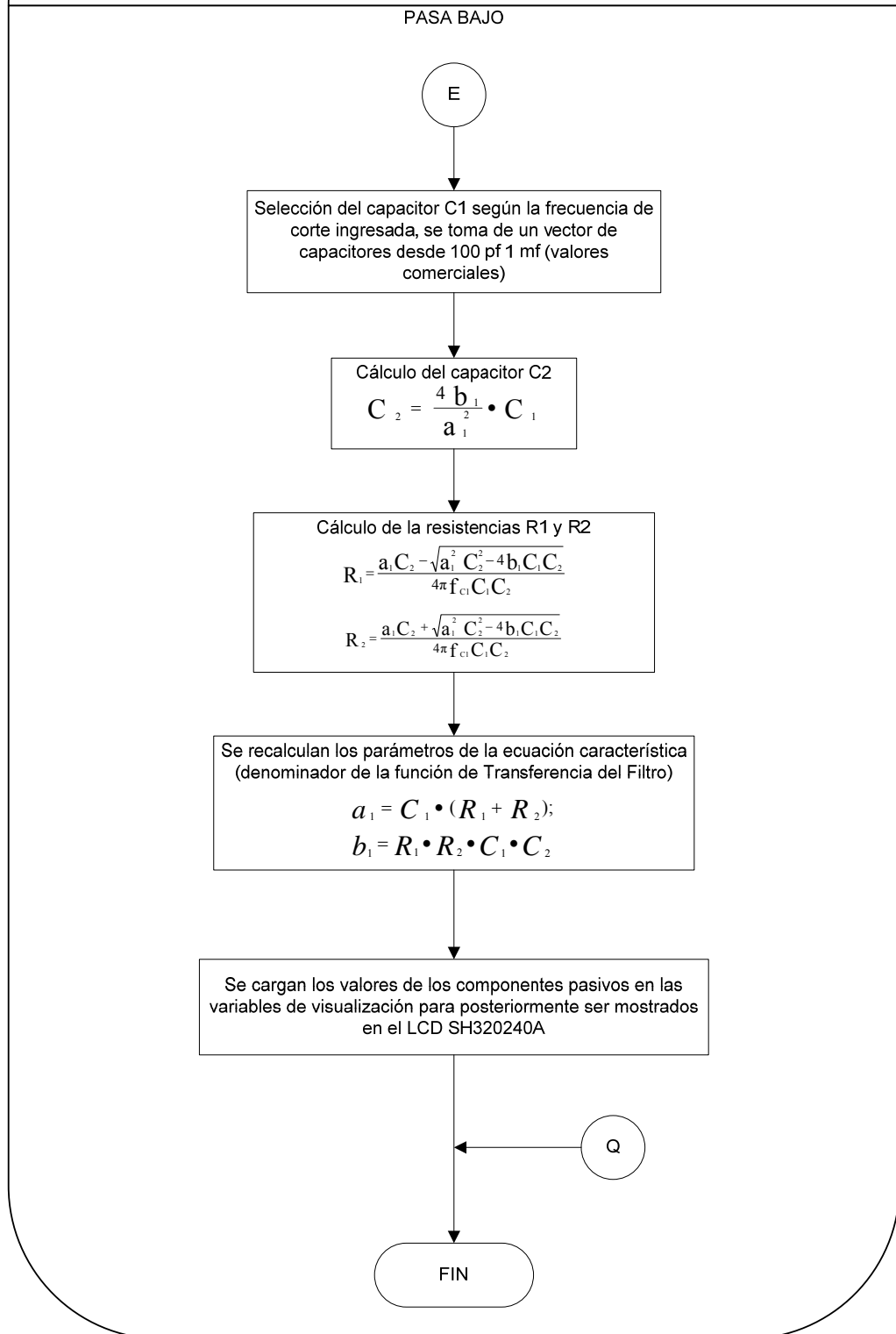
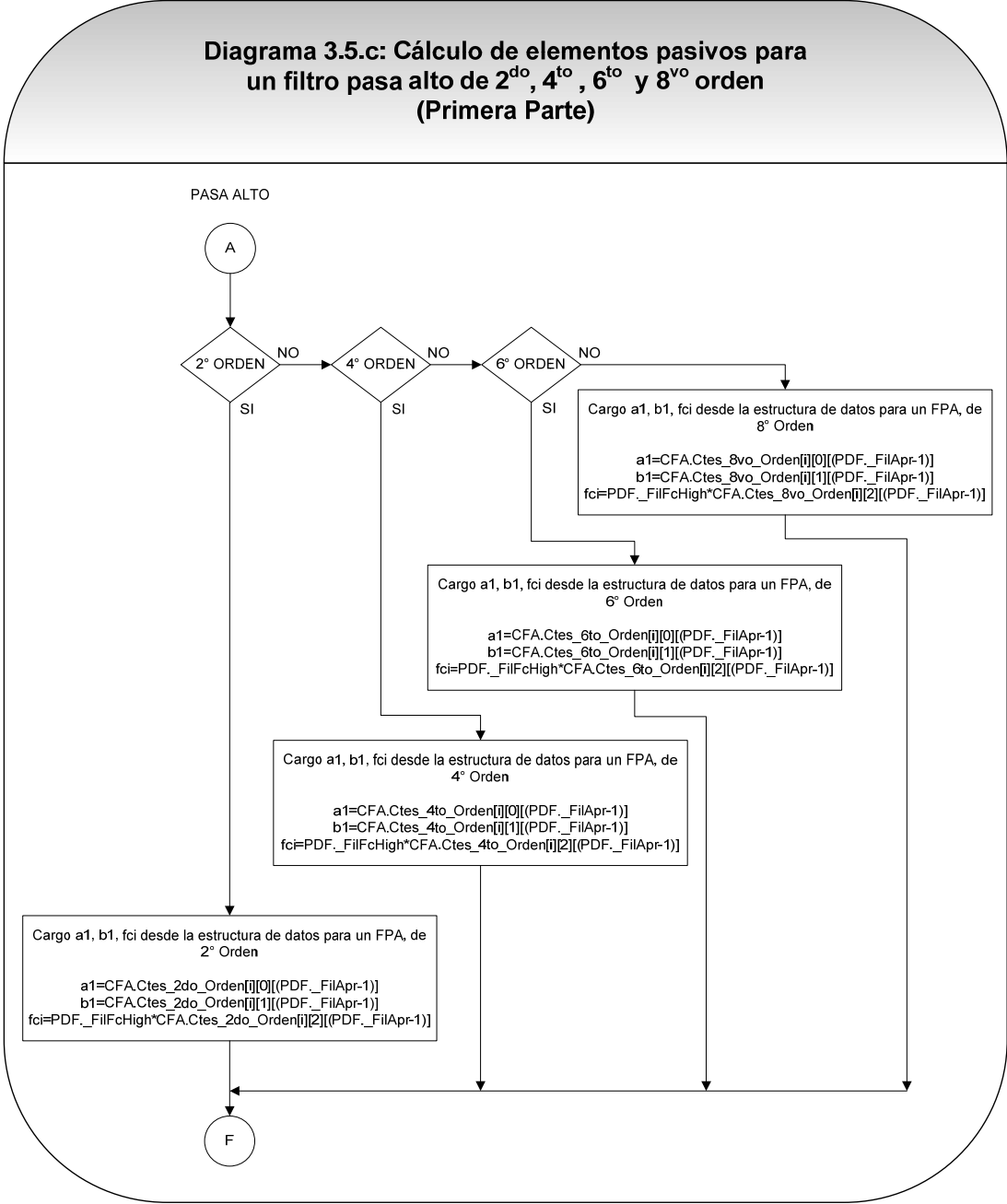
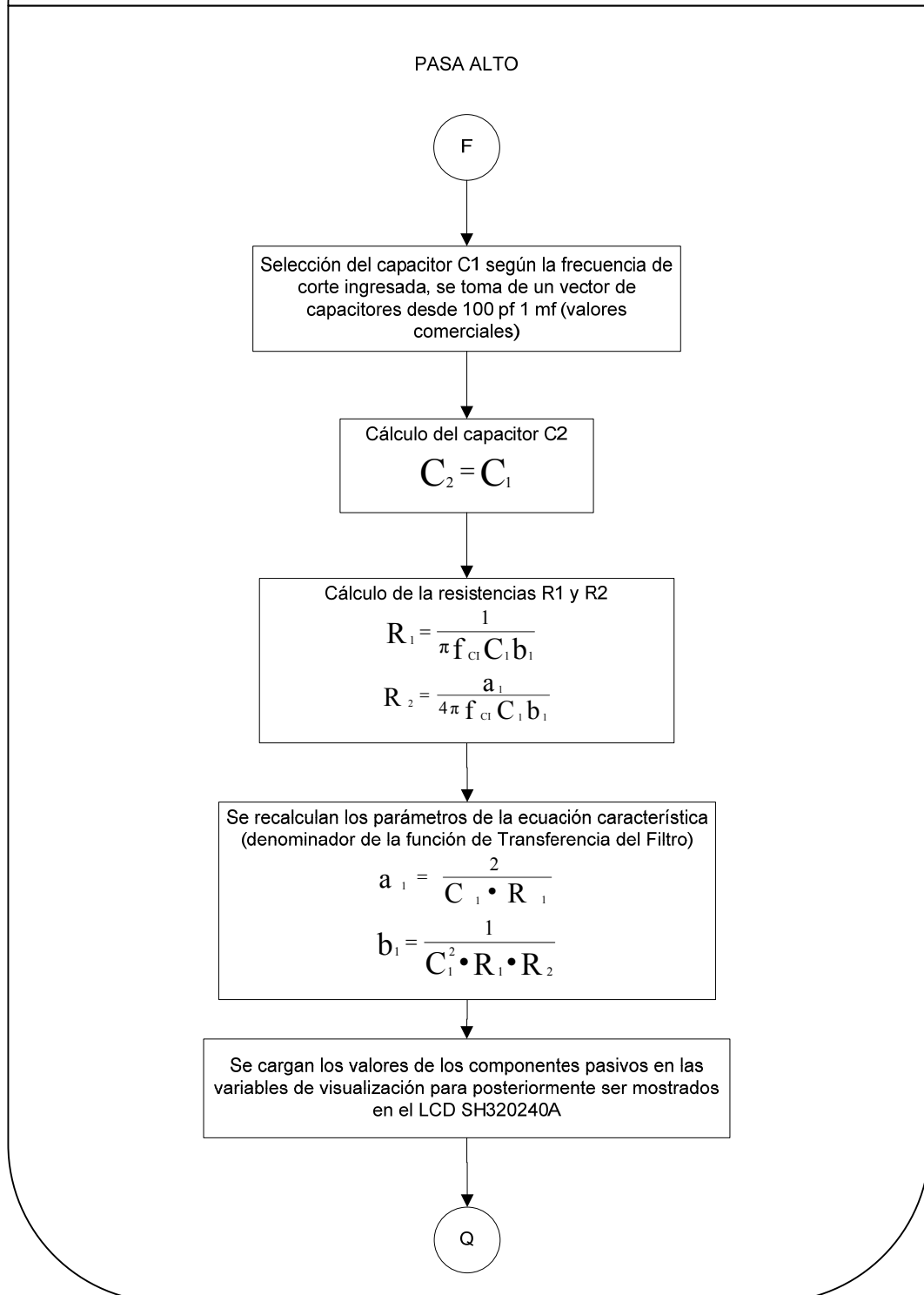


Diagrama 3.5.c: Cálculo de elementos pasivos para un filtro pasa alto de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden (Primera Parte)



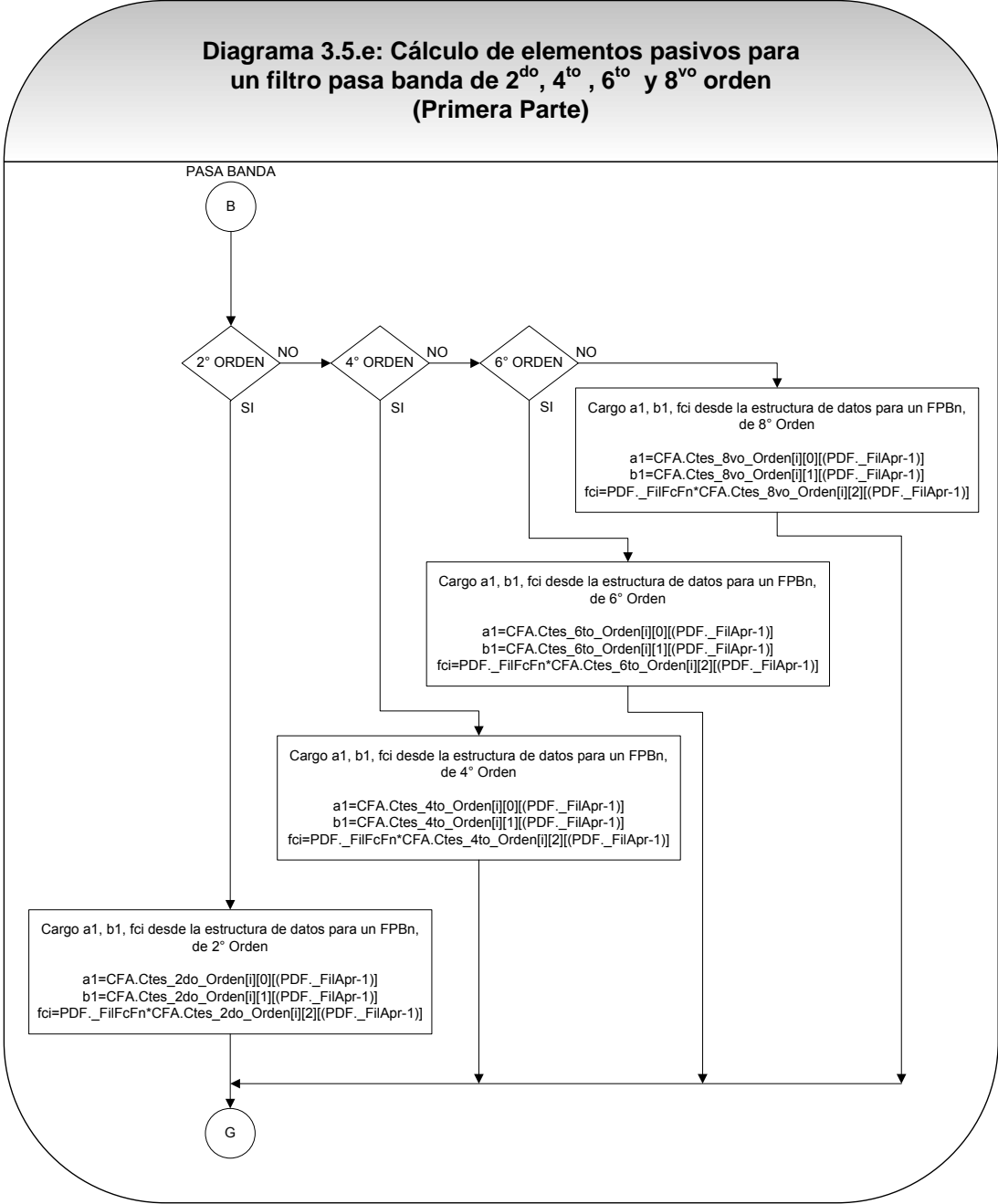
Fuente: Autores.

Diagrama 3.5.d: Cálculo de elementos pasivos para un filtro pasa alto de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden (Segunda Parte)



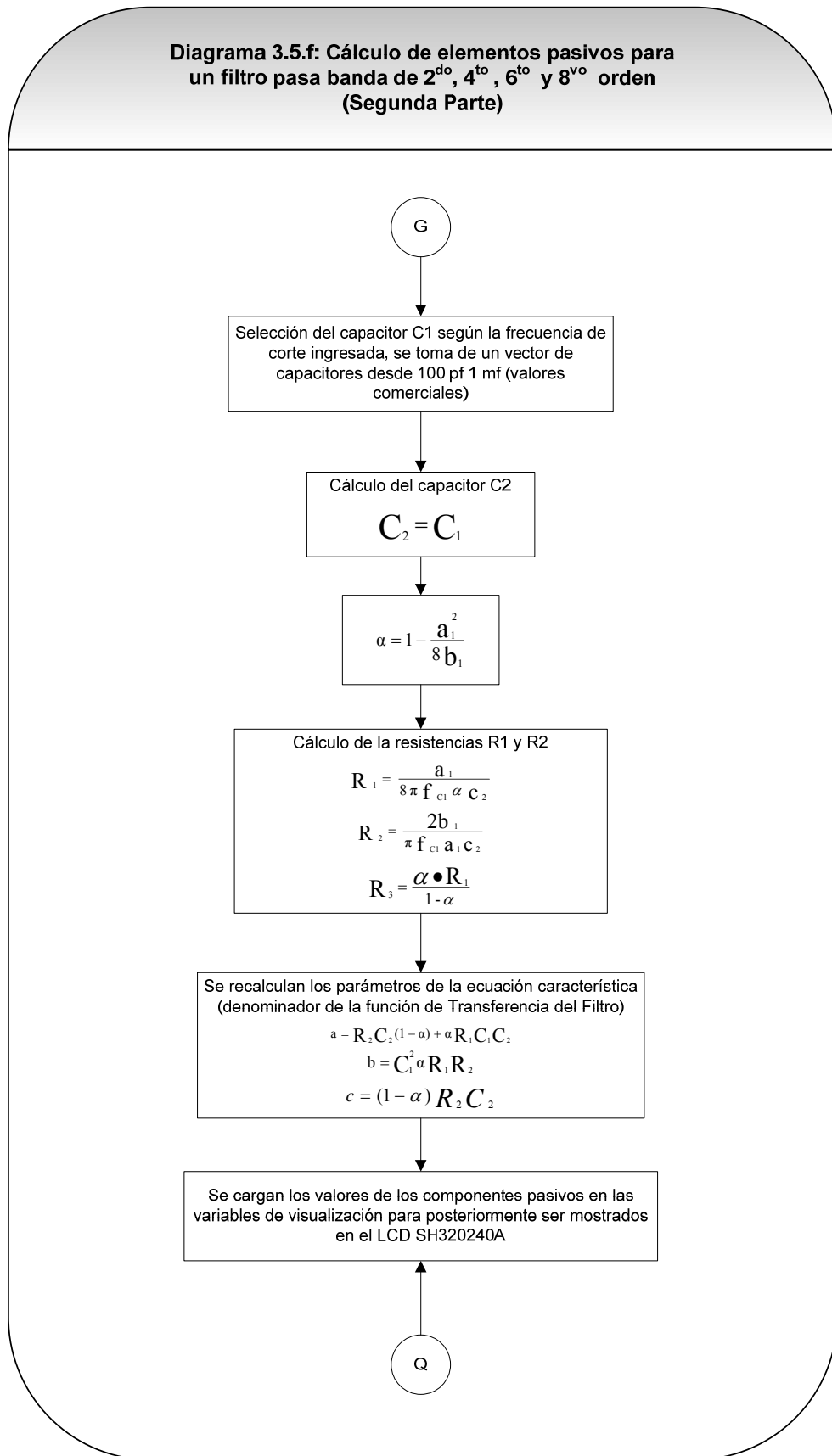
Fuente: Autores.

Diagrama 3.5.e: Cálculo de elementos pasivos para un filtro pasa banda de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden (Primera Parte)



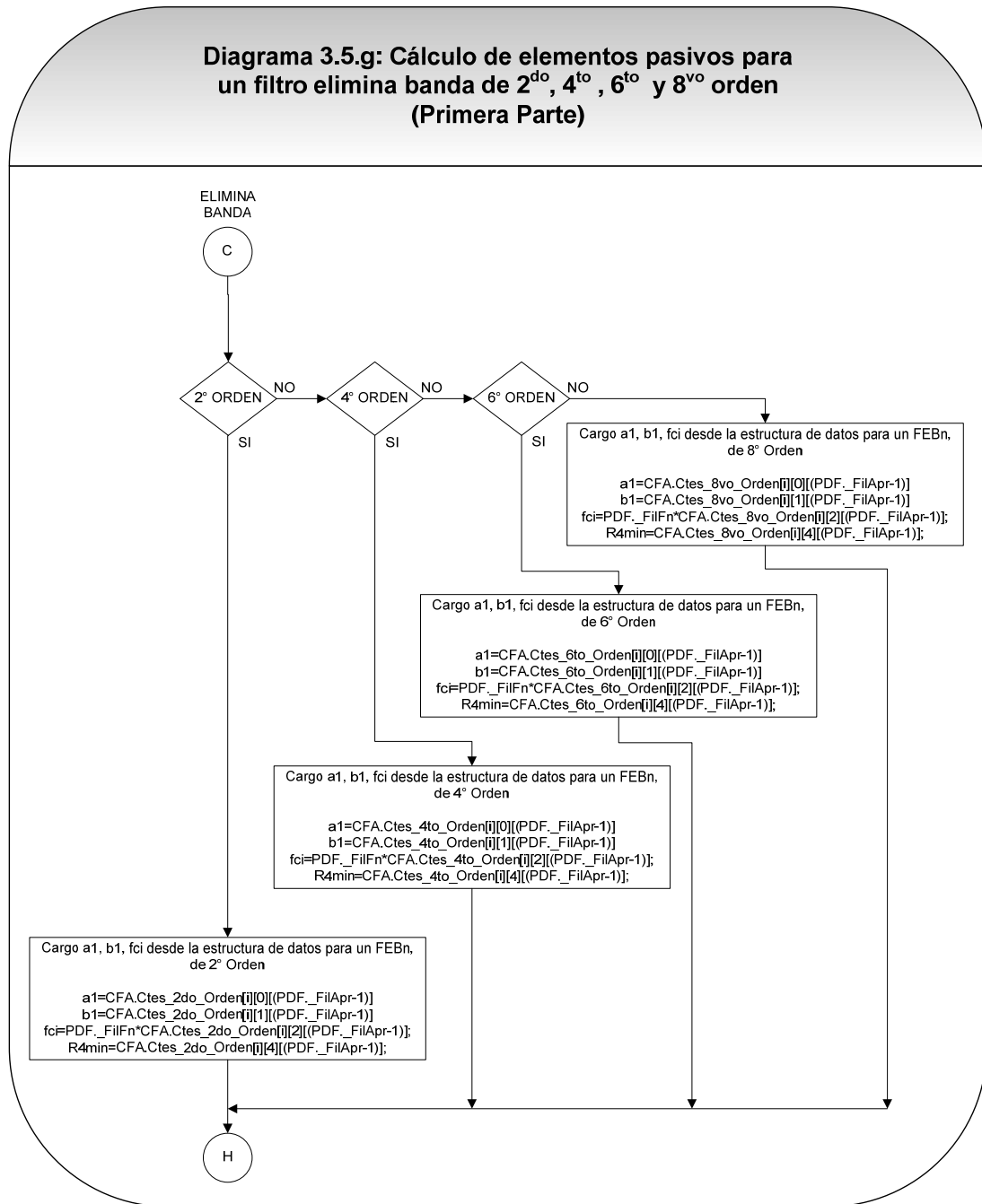
Fuente: Autores.

Diagrama 3.5.f: Cálculo de elementos pasivos para un filtro pasa banda de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden (Segunda Parte)



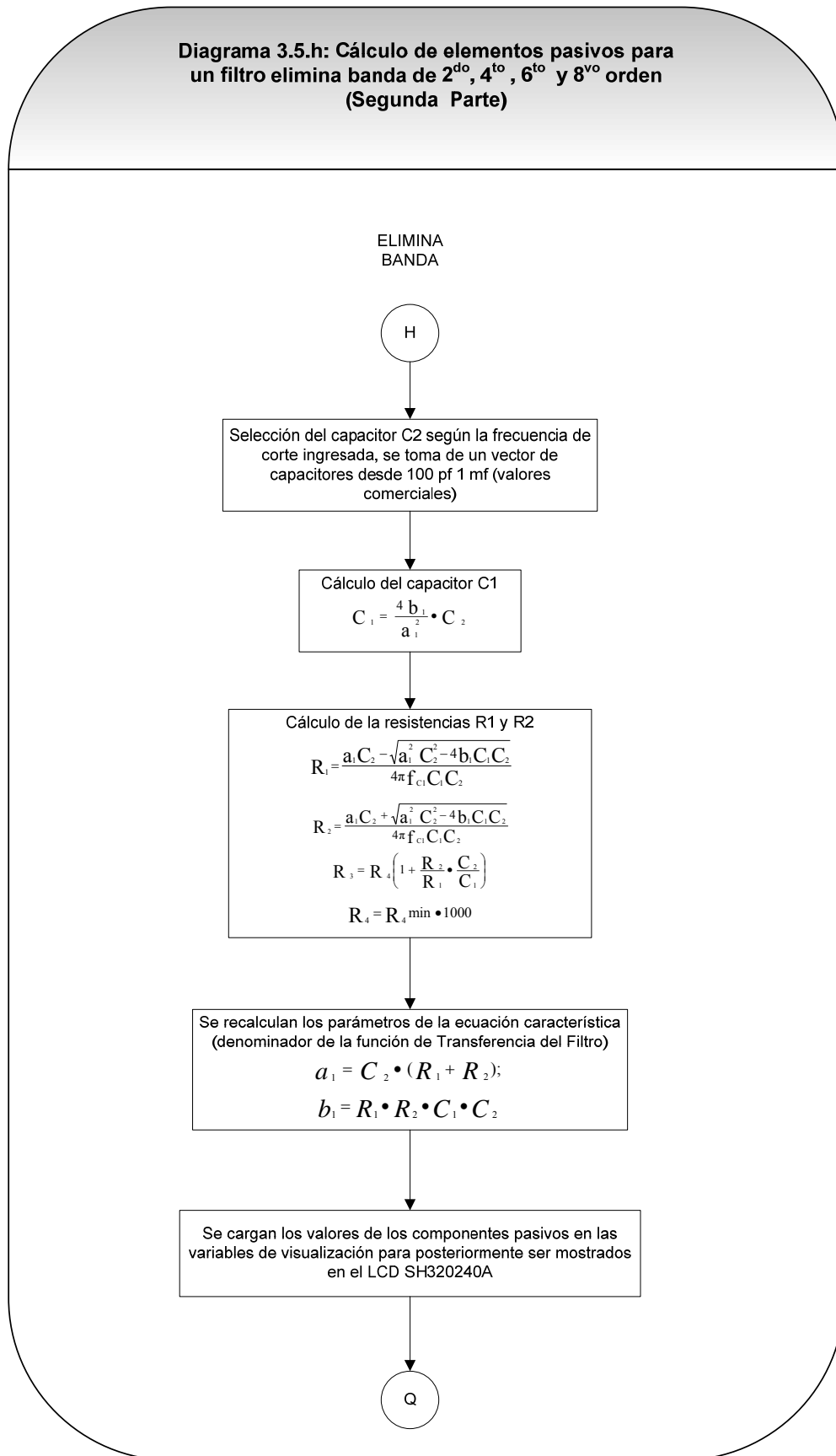
Fuente: Autores.

Diagrama 3.5.g: Cálculo de elementos pasivos para un filtro elimina banda de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden (Primera Parte)



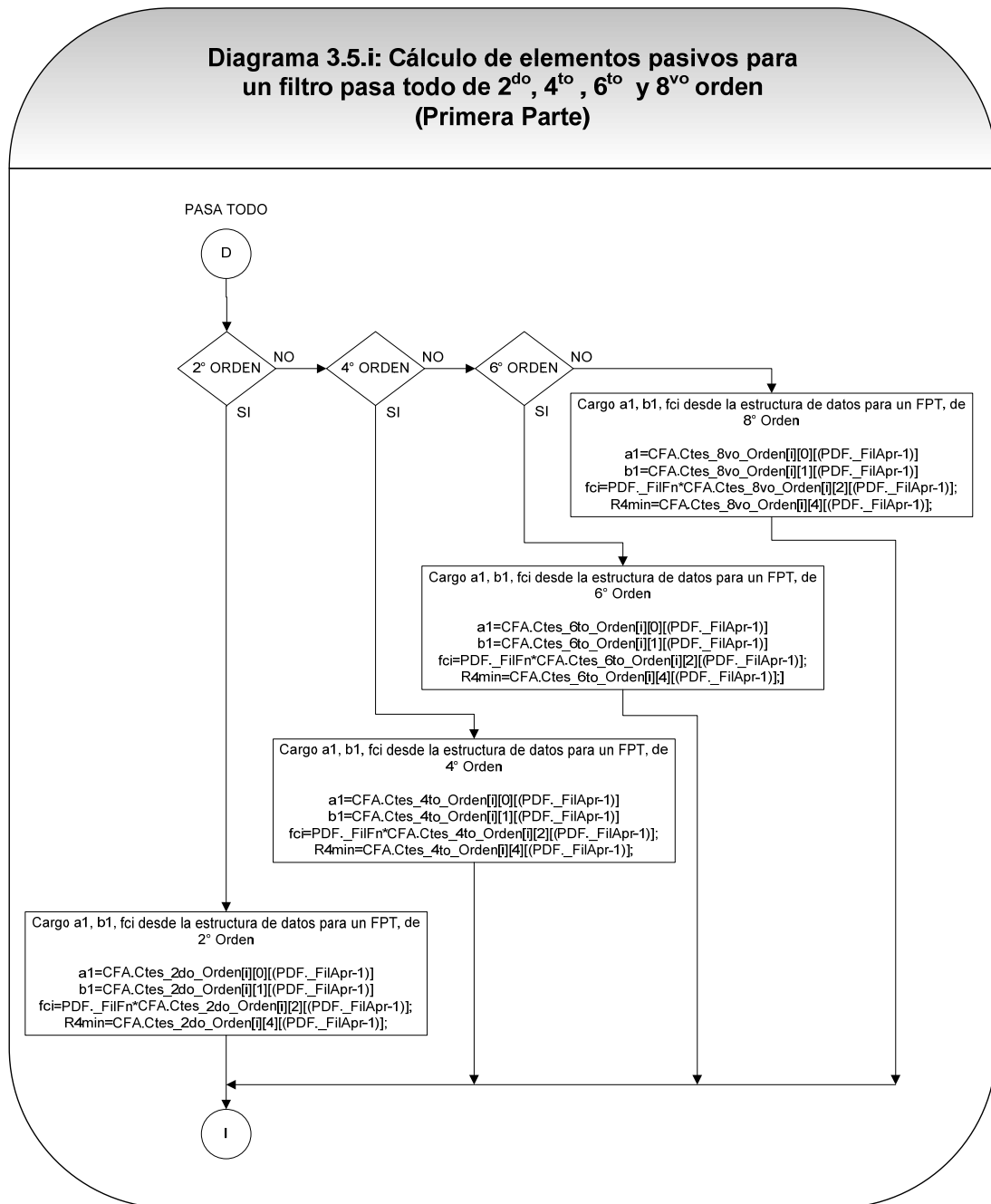
Fuente: Autores.

Diagrama 3.5.h: Cálculo de elementos pasivos para un filtro elimina banda de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden (Segunda Parte)



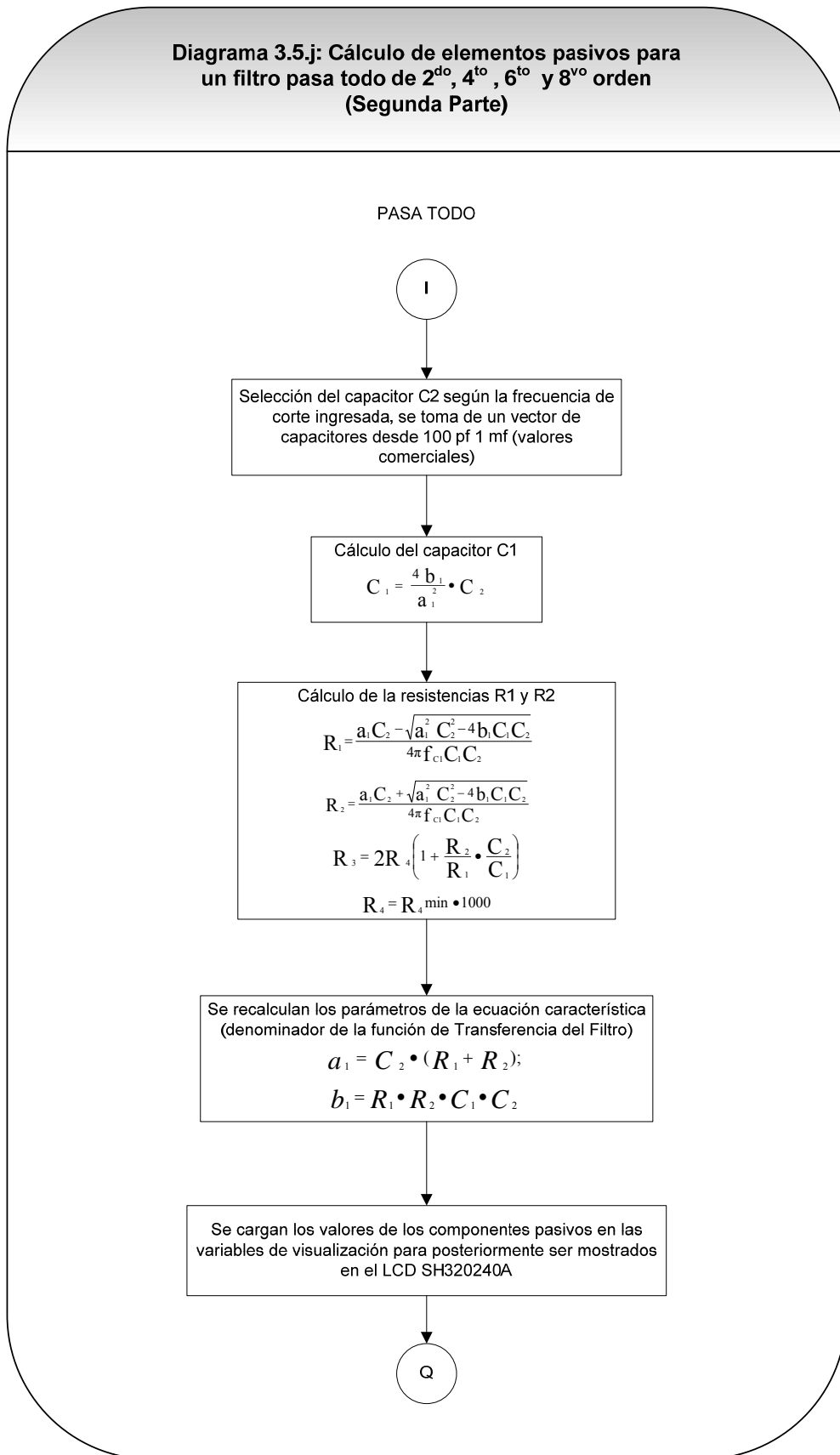
Fuente: Autores.

Diagrama 3.5.i: Cálculo de elementos pasivos para un filtro pasa todo de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden (Primera Parte)



Fuente: Autores.

Diagrama 3.5.j: Cálculo de elementos pasivos para un filtro pasa todo de 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden (Segunda Parte)



3.6 Subrutina de programación para la selección de toda la información del programa general, mediante el teclado matricial y su visualización en la pantalla LCD SH320240A (TaskTecladoMatricial).

1. Al momento de presionar una tecla, para empezar a operar el equipo, se borra la pantalla de texto. Capturar el ASCII de la tecla presionada.
2. Mostrar en pantalla un mensaje de texto sobre el tipo de filtro a seleccionar.
3. Inicializar a 0 la estructura de los parámetros del filtro.
4. Deshabilitar los filtros de: 2^{do}, 4^{to}, 6^{to} y 8^{vo} orden.
5. Banderas de operación a ejecutar a 0.
6. Según la tecla presionada, seleccionar el tipo de filtro.
 - 6.1. Tecla 1: Filtro Pasa Bajo.
 - 6.2. Tecla 2: Filtro Pasa Alto
 - 6.3. Tecla 3: Filtro Pasa Banda
 - 6.4. Tecla 4: Filtro Elimina Banda
 - 6.5. Tecla 5: Filtro Pasa Todo.
7. Visualizar en pantalla el dígito del filtro escogido.
8. Visualizar en pantalla el tipo de aproximación a escoger.
9. Repetir el paso 6, y seleccionar mediante el teclado:
 - 9.1. Aproximación Tschebychev 0.5dB
 - 9.2. Aproximación Tschebychev 1 dB
 - 9.3. Aproximación Tschebychev 2dB
 - 9.4. Aproximación Tschebychev 3dB
 - 9.5. Aproximación Butterworth
 - 9.6. Aproximación Bessel.
10. Una vez seleccionado el tipo de aproximante, visualizar en pantalla el dígito de la aproximación escogido.
11. Visualizar en pantalla el orden del filtro a escoger.
12. Repetir el paso 6, y seleccionar mediante el teclado:
 - 12.1. 2^{do} Orden.
 - 12.2. 4^{to} Orden.
 - 12.3. 6^{to} Orden.
 - 12.4. 8^{vo} Orden.
13. Una vez seleccionado el orden del filtro, visualizar en pantalla el dígito del orden escogido.
14. Limpiar la pantalla de texto.

15. Según el tipo de filtro, visualizar en pantalla sobre la selección de frecuencias, ya sea de corte baja o de corte alta, dependiendo del filtro seleccionado.
16. Visualizar en pantalla el tipo de señal a utilizar:
 - 16.1. Senoidal.
 - 16.2. Triangular.
 - 16.3. Cuadrada.
 - 16.4. Diente de sierra.
 - 16.5. Tren de pulsos.
17. Una vez seleccionado el tipo de señal, visualizar en pantalla el dígito de la señal escogida.
18. Cálculo de elementos pasivos.
19. Conversión de datos numéricos.
20. Visualizar en la pantalla LCD el tipo de señal de ruido con la que el operador va a trabajar.
 - 20.1. Senoidal.
 - 20.2. Triangular.
 - 20.3. Cuadrada.
 - 20.4. Diente de sierra.
 - 20.5. Tren de pulsos.
21. Borrar pantalla de texto.
22. Mensaje sobre ajuste de elementos pasivos.
23. Visualizar componentes pasivos.
24. Mostrar en LCD, el ajuste para los parámetros de las señales pura y ruido, tanto de amplitud, frecuencia y distorsión.
25. Mostrar en LCD indicaciones de adecuación de las señales: pura, ruido, pura + ruido y filtrada.
26. Visualizar en pantalla mensaje de confirmación de parámetros configurados.
27. Visualizar en pantalla los parámetros para el filtro diseñado.

Las figuras 3.5.a, 3.5.b, 3.5.c, 3.5.d, 3.5.e y 3.5.f, muestran la subrutina de programación para la selección de toda la información del programa general, mediante el teclado matricial y su visualización en la pantalla LCD SH320240A (TaskTecladoMatricial).

```

StarterKitTutorial - MPLAB IDE v8.00 - [C:\Program Files (x86)\Microchip\proyectos\KitFilter27\KitFilter27.c]
File Edit View Project Debugger Programmer Tools Configure Window Help
Debug Checksum: 0x82b4194
StarterKitTutorial.mcp
Source Files
  StarterKitTutorial.c
Header Files
  db_utilis.h
Object Files
Library Files
  db_utilis.a
Linker Script
Other Files

/*
 * Esta subrutina selecciona tareas a realizar según datos ingresa- */
/* por el Teclado Matricial. */
/*
 */
void TaskTecladoMatricial(void) {
    static unsigned char i, j, ptr, c;
    static unsigned char ptr_lowfreq;
    static unsigned char ptr_highfreq;
    static unsigned int value, coc, res;
    c=teclapresionada+48;
    if(habfilpru==TRUE){
        if(teclapresionada==BACK){
            ClearPantallaTextoLcdSH320240A();
            WriteCadenaCaracteresLcdSH320240A(" PARAMETROS DE DISEÑO DEL FILTRO. ", 0, 2); //4,2
            switch(PDF_FilTip) {
                case T_FPB:
                    WriteCadenaCaracteresLcdSH320240A(" TIPO: Pasa Bajo FcL: Hz. ", 0, 4); //2,4
                    break;
                case T_FPA:
                    WriteCadenaCaracteresLcdSH320240A(" TIPO: Pasa Alto FcL: Hz. ", 0, 4); //2,4
                    break;
                case T_FPBN:
                    WriteCadenaCaracteresLcdSH320240A(" TIPO: Pasa Banda FcL: Hz. ", 0, 4); //2,4
                    break;
                case T_FEBN:
                    WriteCadenaCaracteresLcdSH320240A(" TIPO: Elimina Banda FcL: Hz. ", 0, 4); //2,4
                    break;
                case T_FPT:
                    WriteCadenaCaracteresLcdSH320240A(" TIPO: Pasa Todo FcL: Hz. ", 0, 4); //2,4
                    break;
            }
            switch(PDF_FilApr) {
                case A_TSC_5dB:
                    WriteCadenaCaracteresLcdSH320240A(" APROX: Tscheb./0.5dB FcH: Hz. ", 0, 5); //2,5
                    break;
                case A_TSC_1dB:
                    WriteCadenaCaracteresLcdSH320240A(" APROX: Tscheb./1.0dB FcH: Hz. ", 0, 5); //2,5
                    break;
                case A_TSC_2dB:
                    WriteCadenaCaracteresLcdSH320240A(" APROX: Tscheb./2.0dB FcH: Hz. ", 0, 5); //2,5
                    break;
                case A_TSC_3dB:
                    WriteCadenaCaracteresLcdSH320240A(" APROX: Tscheb./3.0dB FcH: Hz. ", 0, 5); //2,5
                    break;
                case A_BUT:
                    WriteCadenaCaracteresLcdSH320240A(" APROX: Butterworth FcH: Hz. ", 0, 5); //2,5
                    break;
                case A_BES:
                    WriteCadenaCaracteresLcdSH320240A(" APROX: Bessel FcH: Hz. ", 0, 5); //2,5
                    break;
            }
            switch(PDF_FilOrd) {
                case O_2DO:
                    WriteCadenaCaracteresLcdSH320240A(" ORDEN: Segundo Fn: Hz. ", 0, 6); //2,6
                    break;
                case O_4TO:
                    WriteCadenaCaracteresLcdSH320240A(" ORDEN: Cuarto Fn: Hz. ", 0, 6); //2,6
                    break;
                case O_6TO:
                    WriteCadenaCaracteresLcdSH320240A(" ORDEN: Sexto Fn: Hz. ", 0, 6); //2,6
                    break;
                case O_8VO:
                    WriteCadenaCaracteresLcdSH320240A(" ORDEN: Octavo Fn: Hz. ", 0, 6); //2,6
                    break;
            }
            if(PDF_FilFcLow!=0){
                for(i=0; i<ptr_lowfreq; i++){
                    screen_text[35-ptr_lowfreq+i][4]=((unsigned char)(input_lowfreq[i]+48));
                }
            }
            else {
                for(i=0; i<ptr_lowfreq; i++){
                    screen_text[35-ptr_lowfreq+i][4]=" ";
                }
            }
            if(PDF_FilFcHigh!=0){
                for(i=0; i<ptr_highfreq; i++){
                    ...
                }
            }
        }
    }
}
PIC32MX360FS12L Output Ln 884, Col 1 INS WR

```

Figura 3.5.a: Subrutina de programación para la selección de toda la información del programa general, mediante el teclado matricial y su visualización en la pantalla LCD SH320240A (TaskTecladoMatricial).

Fuente: Autores.

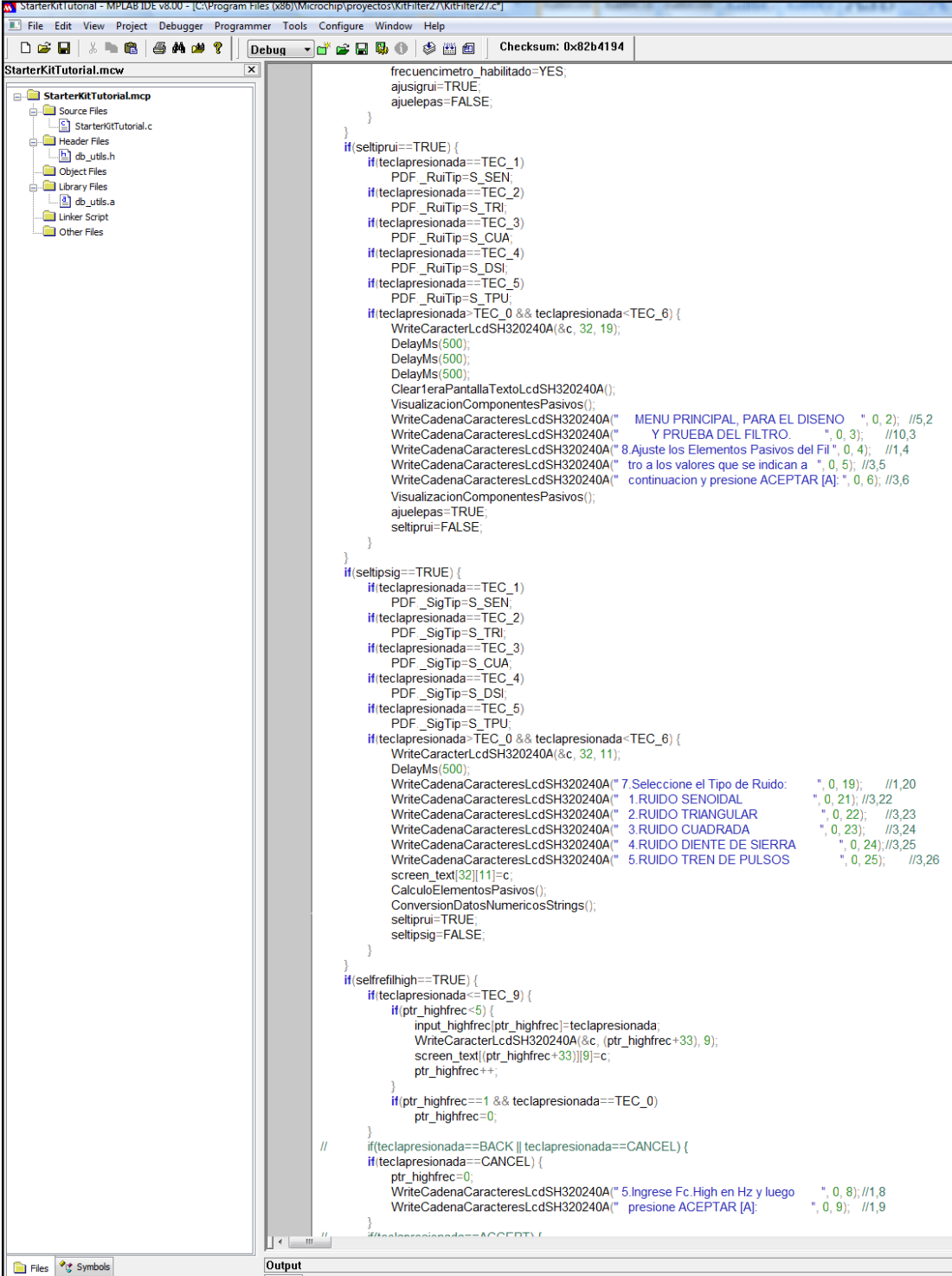


Figura 3.5.c: Subrutina de programación para la selección de toda la información del programa general, mediante el teclado matricial y su visualización en la pantalla LCD SH320240A (TaskTecladoMatricial).

Fuente: Autores.

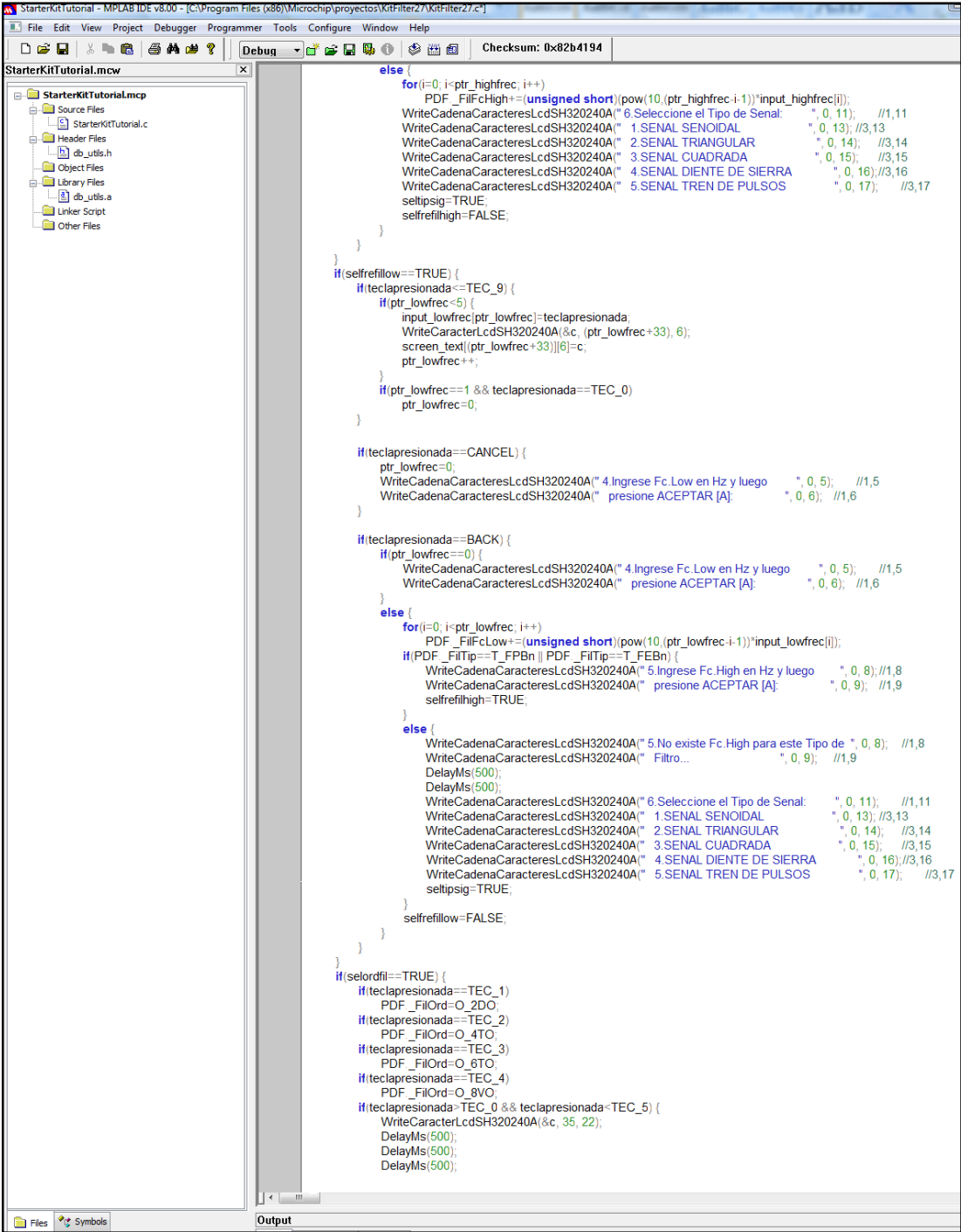


Figura 3.5.d: Subrutina de programación para la selección de toda la información del programa general, mediante el teclado matricial y su visualización en la pantalla LCD SH320240A (TaskTecladoMatricial).

Fuente: Autores.

```

Clear1eraPantallaTextoLcdSH320240A();
WriteCadenaCaracteresLcdSH320240A(" MENU PRINCIPAL, PARA EL DISENO ", 0, 2); //5,2
WriteCadenaCaracteresLcdSH320240A(" Y PRUEBA DEL FILTRO. ", 0, 3); //10,3
if(PDF_FilTip==T_FPB || PDF_FilTip==T_FPBN || PDF_FilTip==T_FEBN) {
    WriteCadenaCaracteresLcdSH320240A(" 4.Ingrese Fc.Low en Hz y luego ", 0, 5); //1,5
    WriteCadenaCaracteresLcdSH320240A(" presione ACEPTAR [A]; ", 0, 6); //1,6
    selfrefill=TRUE;
    selordfil=FALSE;
}
if(PDF_FilTip==T_FPA) {
    WriteCadenaCaracteresLcdSH320240A(" 4.No existe Fc.Low para este Tipo de ", 0, 5); //1,5
    WriteCadenaCaracteresLcdSH320240A(" Filtro... ", 0, 6); //1,6
    DelayMs(500);
    DelayMs(500);
    WriteCadenaCaracteresLcdSH320240A(" 5.Ingrese Fc.High en Hz y luego ", 0, 8); //1,8
    WriteCadenaCaracteresLcdSH320240A(" presione ACEPTAR [A]; ", 0, 9); //1,9
    selfrefillhigh=TRUE;
    selordfil=FALSE;
}
if(PDF_FilTip==T_FPT) {
    WriteCadenaCaracteresLcdSH320240A(" 4.No existe Fc.Low para este Tipo de ", 0, 5); //1,5
    WriteCadenaCaracteresLcdSH320240A(" Filtro... ", 0, 6); //1,6
    DelayMs(500);
    DelayMs(500);
    WriteCadenaCaracteresLcdSH320240A(" 5.No existe Fc.High para este Tipo de ", 0, 8); //1,8
    WriteCadenaCaracteresLcdSH320240A(" Filtro... ", 0, 9); //1,9
    DelayMs(500);
    DelayMs(500);
    WriteCadenaCaracteresLcdSH320240A(" 6.Seleccione el Tipo de Senal: ", 0, 11); //1,11
    WriteCadenaCaracteresLcdSH320240A(" 1.SENAL SENOIDAL ", 0, 13); //3,13
    WriteCadenaCaracteresLcdSH320240A(" 2.SENAL TRIANGULAR ", 0, 14); //3,14
    WriteCadenaCaracteresLcdSH320240A(" 3.SENAL CUADRADA ", 0, 15); //3,15
    WriteCadenaCaracteresLcdSH320240A(" 4.SENAL DIENTE DE SIERRA ", 0, 16); //3,16
    WriteCadenaCaracteresLcdSH320240A(" 5.SENAL TREN DE PULSOS ", 0, 17); //3,17
    selftipsig=TRUE;
    selordfil=FALSE;
}
}
}
if(selaprfil==TRUE) {
    if(teclapresionada==TEC_1)
        PDF_FilApr=A_TSC_5dB;
    if(teclapresionada==TEC_2)
        PDF_FilApr=A_TSC_1dB;
    if(teclapresionada==TEC_3)
        PDF_FilApr=A_TSC_2dB;
    if(teclapresionada==TEC_4)
        PDF_FilApr=A_TSC_3dB;
    if(teclapresionada==TEC_5)
        PDF_FilApr=A_BUT;
    if(teclapresionada==TEC_6)
        PDF_FilApr=A_BES;
    if(teclapresionada==TEC_0 && teclapresionada<TEC_7) {
        WriteCaracterLcdSH320240A(&c, 31, 13);
        DelayMs(500);
        WriteCadenaCaracteresLcdSH320240A(" 3. Seleccione el Orden del Filtro: ", 0, 22); //1,22
        WriteCadenaCaracteresLcdSH320240A(" 1.SEGUNDO ", 0, 24); //3,24
        WriteCadenaCaracteresLcdSH320240A(" 2.CUARTO ", 0, 25); //3,25
        WriteCadenaCaracteresLcdSH320240A(" 3.SEXTO ", 0, 26); //3,26
        WriteCadenaCaracteresLcdSH320240A(" 4.OCTAVO ", 0, 27); //3,27
        screen_text[31][13]=c;
        selordfil=TRUE;
        selaprfil=FALSE;
    }
}
if(seltipfil==TRUE) {
    if(teclapresionada==TEC_1)
        PDF_FilTip=T_FPB;
    if(teclapresionada==TEC_2)
        PDF_FilTip=T_FPA;
    if(teclapresionada==TEC_3)
        PDF_FilTip=T_FPBN;
    if(teclapresionada==TEC_4)
        PDF_FilTip=T_FEBN;
    if(teclapresionada==TEC_5)
        PDF_FilTip=T_FEBN;
    if(teclapresionada==TEC_6)
        PDF_FilTip=T_FPT;
}
}
}

```

Figura 3.5.e: Subrutina de programación para la selección de toda la información del programa general, mediante el teclado matricial y su visualización en la pantalla LCD SH320240A (TaskTecladoMatricial).

Fuente: Autores.


```

WriteCaracterLcdSH320240A(&c, 33, 5);
DelayMs(500);
WriteCadenaCaracteresLcdSH320240A(" 2. Seleccione la Aproximacion: ", 0, 13); //1,13
WriteCadenaCaracteresLcdSH320240A(" 1. TSCHEBYSCHIEFF DE 0.5dB ", 0, 15); //3,15
WriteCadenaCaracteresLcdSH320240A(" 2. TSCHEBYSCHIEFF DE 1.0dB ", 0, 16); //3,16
WriteCadenaCaracteresLcdSH320240A(" 3. TSCHEBYSCHIEFF DE 2.0dB ", 0, 17); //3,17
WriteCadenaCaracteresLcdSH320240A(" 4. TSCHEBYSCHIEFF DE 3.0dB ", 0, 18); //3,18
WriteCadenaCaracteresLcdSH320240A(" 5. BUTTERWORTH ", 0, 19); //3,19
WriteCadenaCaracteresLcdSH320240A(" 6. BESSEL ", 0, 20); //3,20
screen_text[33][5]=c;
setlapfil=TRUE;
setlpru=FALSE;
}
}

if(teclapresionada==TEC_E) {
Clear1eraPantallaTextoLcdSH320240A();
WriteCadenaCaracteresLcdSH320240A(" MENU PRINCIPAL, PARA EL DISENO ", 0, 2); //5,2
WriteCadenaCaracteresLcdSH320240A(" Y PRUEBA DEL FILTRO: ", 0, 3); //10,3
WriteCadenaCaracteresLcdSH320240A(" 1. Seleccione el Tipo de Filtro: ", 0, 5); //1,5
WriteCadenaCaracteresLcdSH320240A(" 1. FILTRO PASA BAJO ", 0, 7); //3,7
WriteCadenaCaracteresLcdSH320240A(" 2. FILTRO PASA ALTO ", 0, 8); //3,8
WriteCadenaCaracteresLcdSH320240A(" 3. FILTRO PASA BANDA ", 0, 9); //3,9
WriteCadenaCaracteresLcdSH320240A(" 4. FILTRO ELIMINA BANDA ", 0, 10); //3,10
WriteCadenaCaracteresLcdSH320240A(" 5. FILTRO PASA TODO ", 0, 11); //3,11
PDF_FilTip=0;
PDF_FilApr=0;
PDF_FilOrd=0;
PDF_FilFcLow=0;
PDF_FilFcHigh=0;
PDF_FilFn=0;

for(i=0; i<5; i++) {
input_lowfreq[i]=0;
input_highfreq[i]=0;
}

frecuencimetro_habilitado=NO;

mPORTAClearBits(FPB2N1 | FPB2N2 | FPB4N1 | FPB4N2 | FPB6N1 | FPB6N2 | FPB8N1 | FPB8N2);
mPORTDClearBits(FPA2N1 | FPA2N2 | FPA4N1 | FPA4N2 | FPA6N1);
mPORTEClearBits(FPA6N2 | FPA8N1);
mPORTFClearBits(FPA6N2);
mPORTEClearBits(FPBn2N1 | FPBn2N2 | FPBn4N1 | FPBn4N2 | FPBn6N1 | FPBn6N2 | FPBn8N1 | FPBn8N2);
mPORTGClearBits(FEBn2N1 | FEBn2N2 | FEBn4N1 | FEBn4N2 | FEBn6N1);
mPORTGClearBits(FEBn6N2 | FEBn8N1 | FEBn8N2);

setlpru=TRUE;
setlapfil=FALSE;
selordfil=FALSE;
selfrefillow=FALSE;
selfrefilhigh=FALSE;
ajuetepas=FALSE;
setlipsig=FALSE;
ajusigrui=FALSE;
setlprui=FALSE;
ajusigrui=FALSE;
adesigrui=FALSE;
habfilpru=FALSE;
}
}

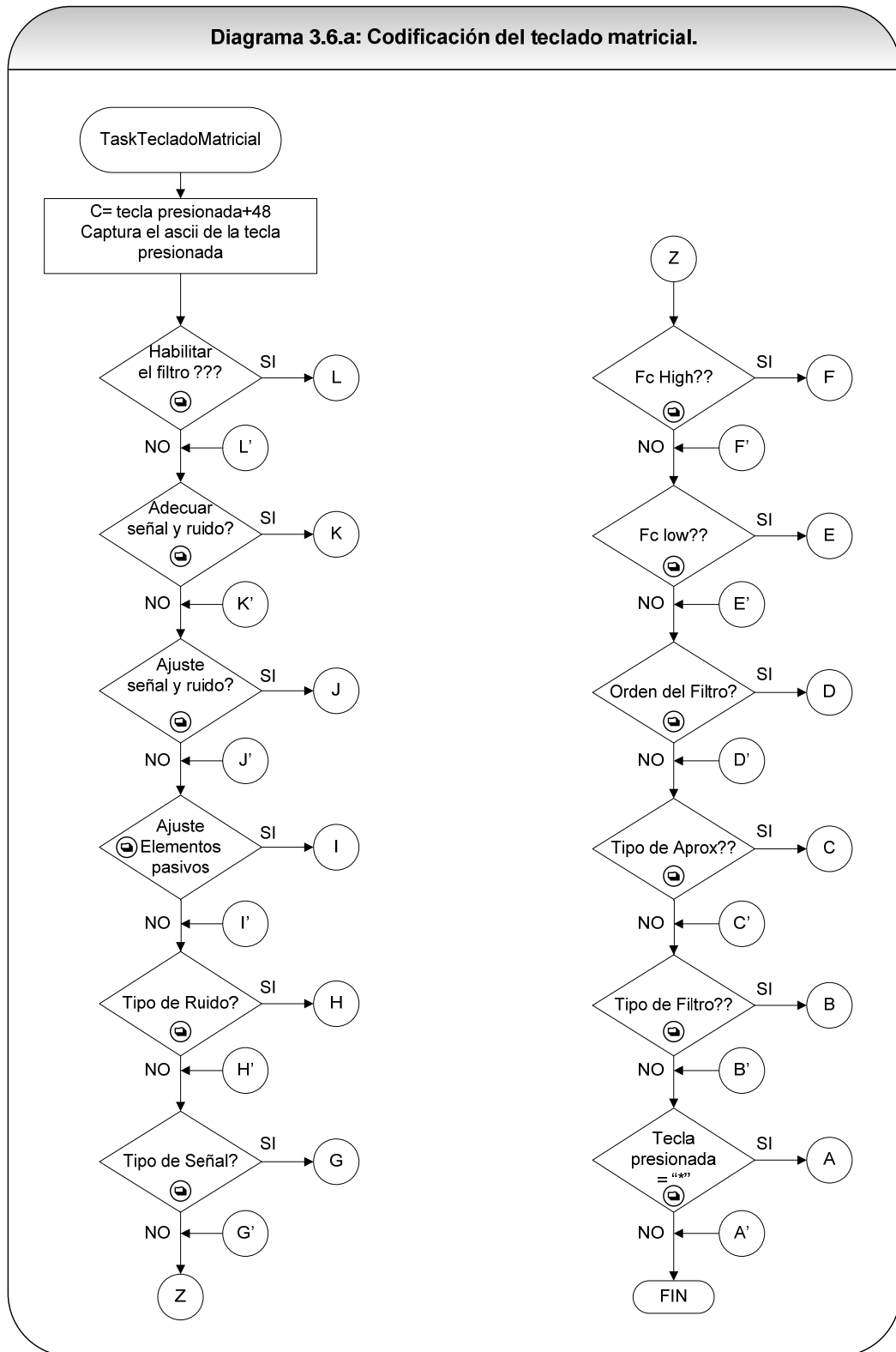
```

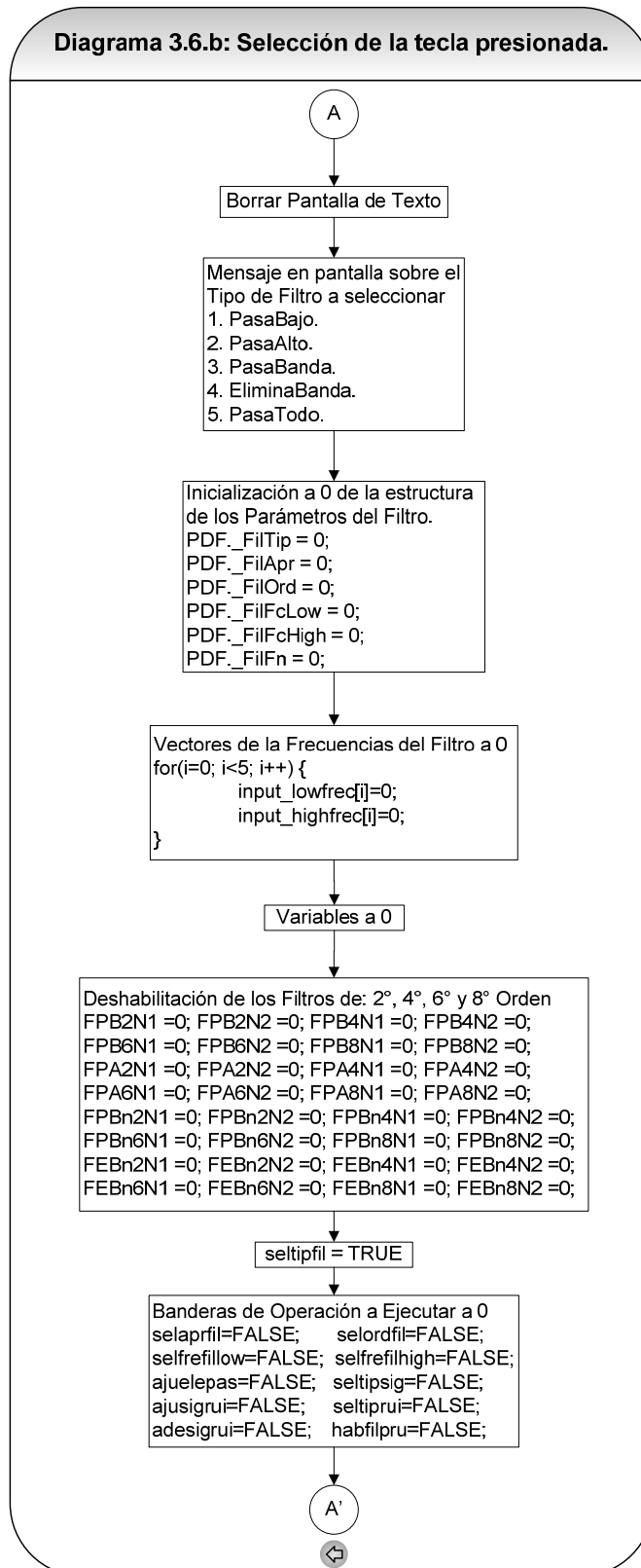
Figura 3.5.f: Subrutina de programación para la selección de toda la información del programa general, mediante el teclado matricial y su visualización en la pantalla LCD SH320240A (TaskTecladoMatricial).

Fuente: Autores.

Los diagramas 3.6.a, 3.6.b, 3.6.c, 3.6.d, 3.6.e, 3.6.f, 3.6.g, 3.6.h, 3.6.i, 3.6.j, 3.6.k, 3.6.l, 3.6.m, muestran los flujo gramas para la subrutina de programación

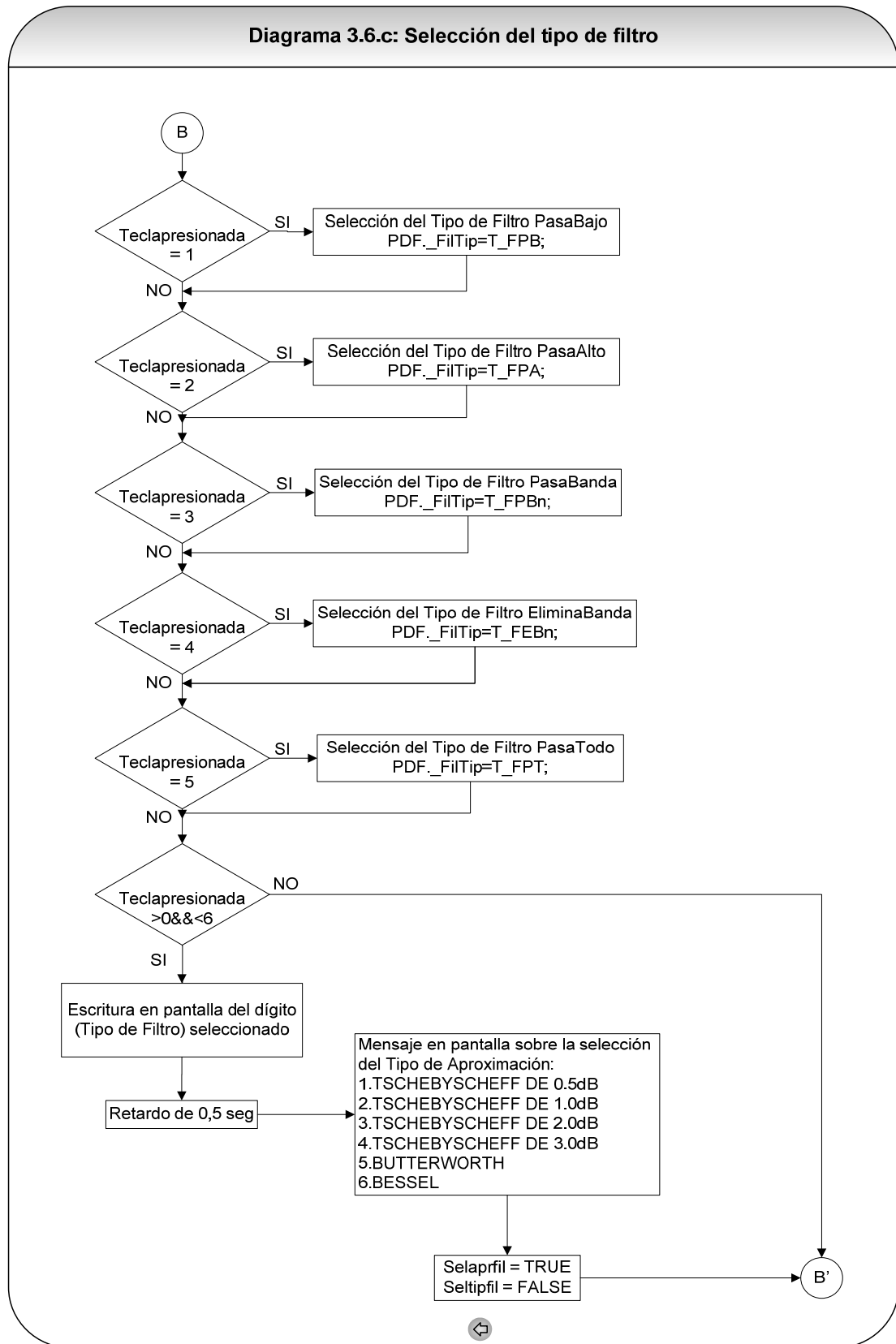
para el reconocimiento de teclas del teclado matricial y su visualización en la pantalla LCD.





Fuente: Autores.

Diagrama 3.6.c: Selección del tipo de filtro



Fuente: Autores.

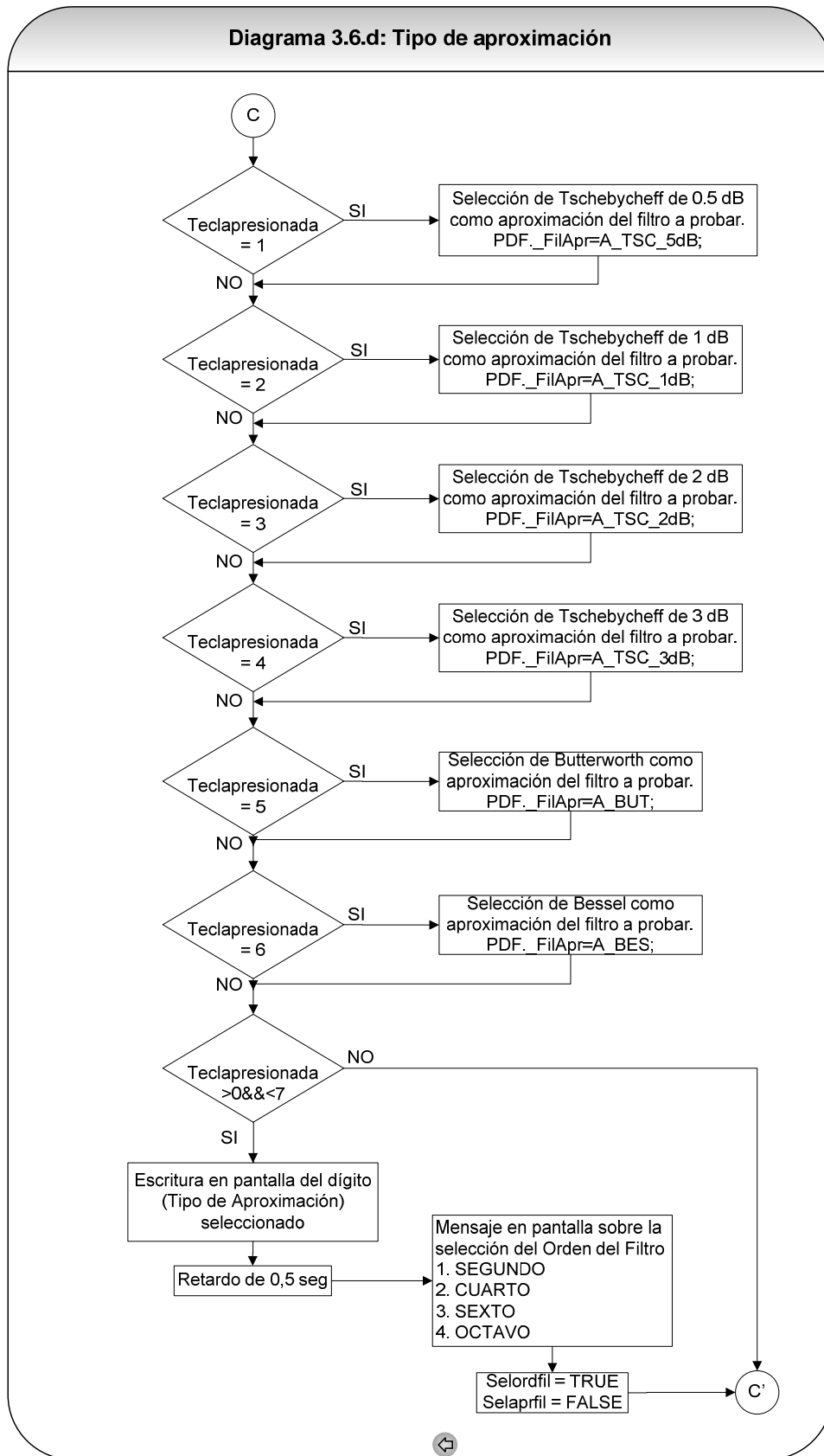


Diagrama 3.6.e: Selección del orden del filtro

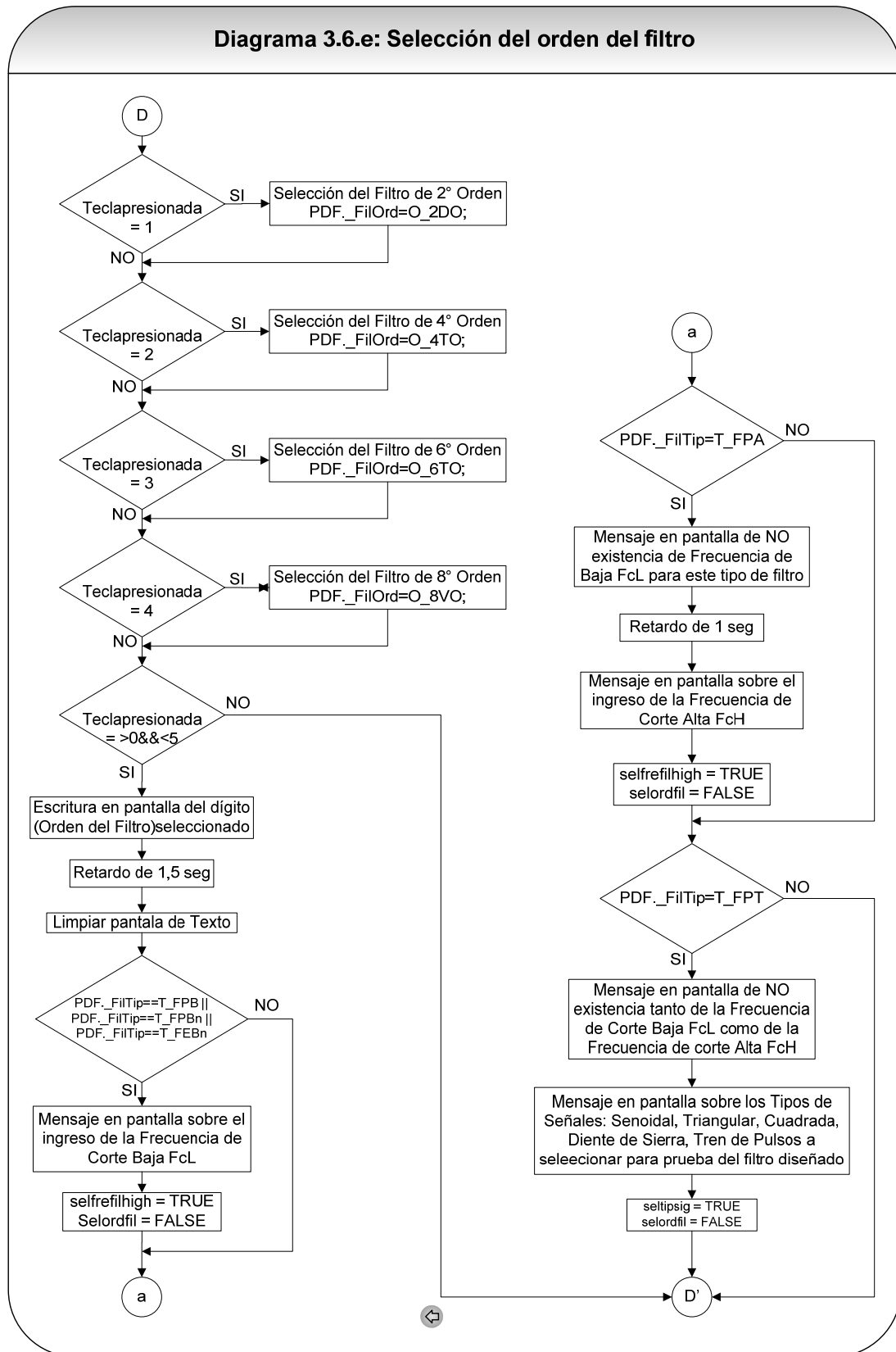


Diagrama 3.6.f:Frecuencia de corte baja.

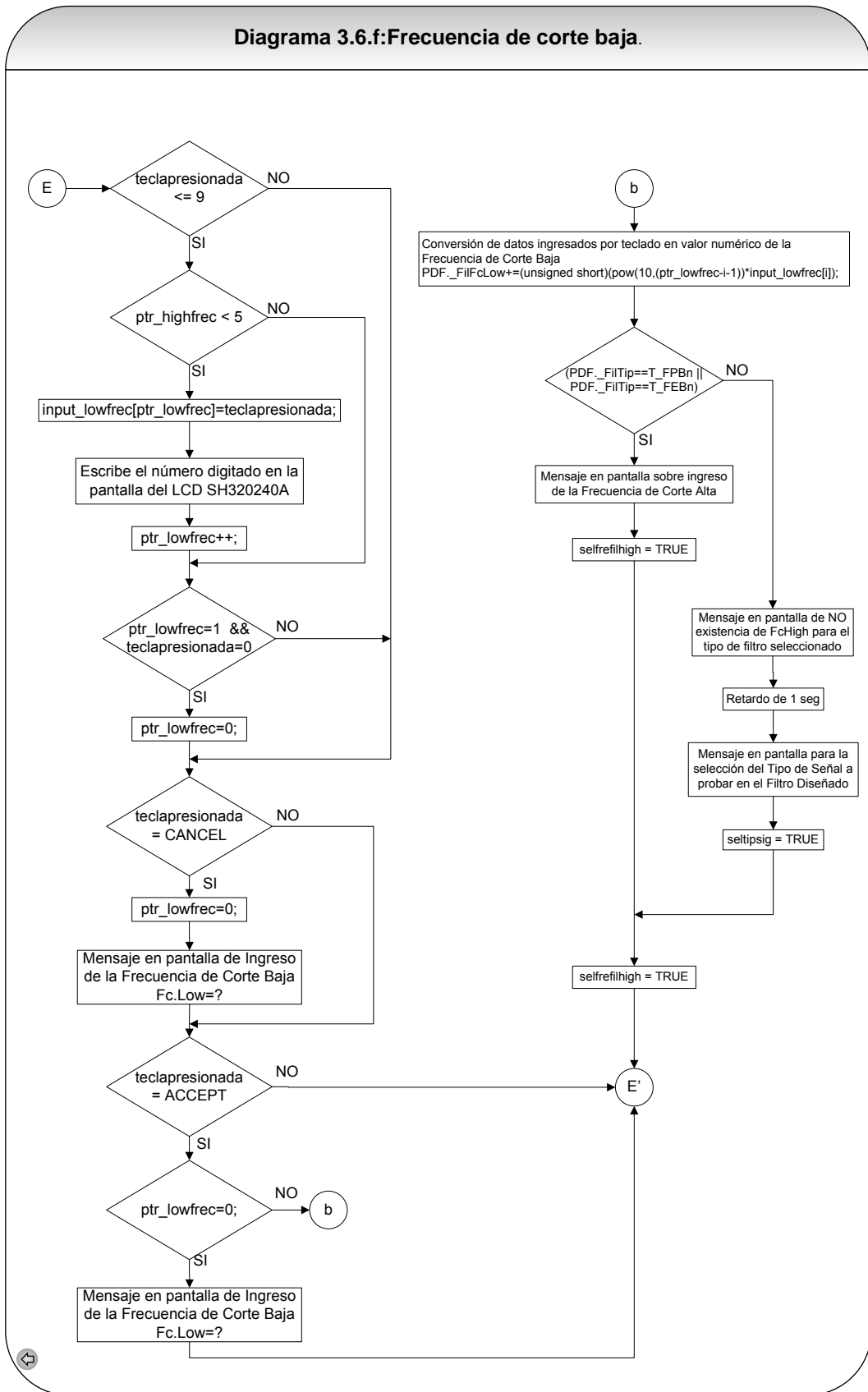
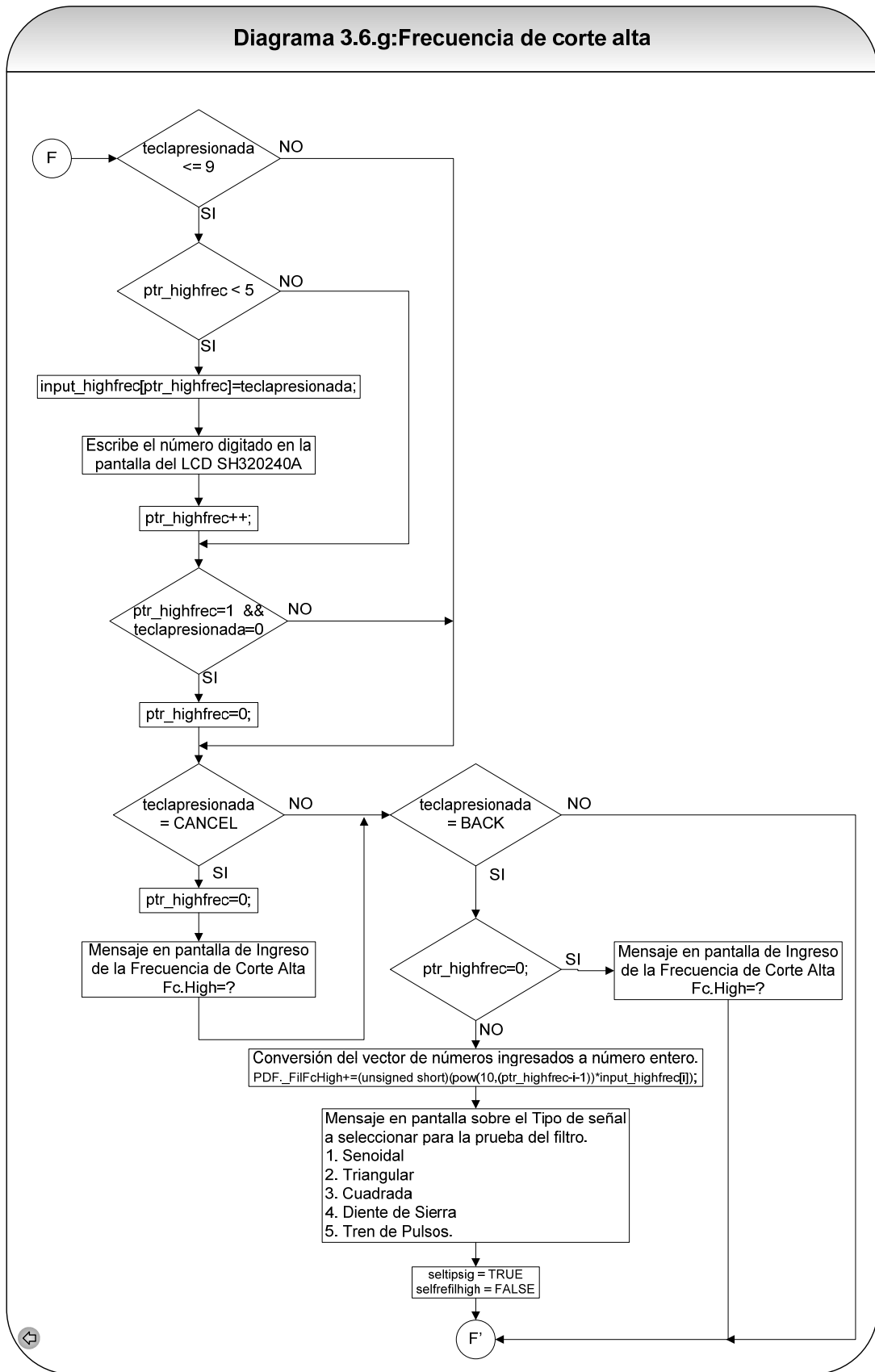


Diagrama 3.6.g:Frecuencia de corte alta



Fuente: Autores.

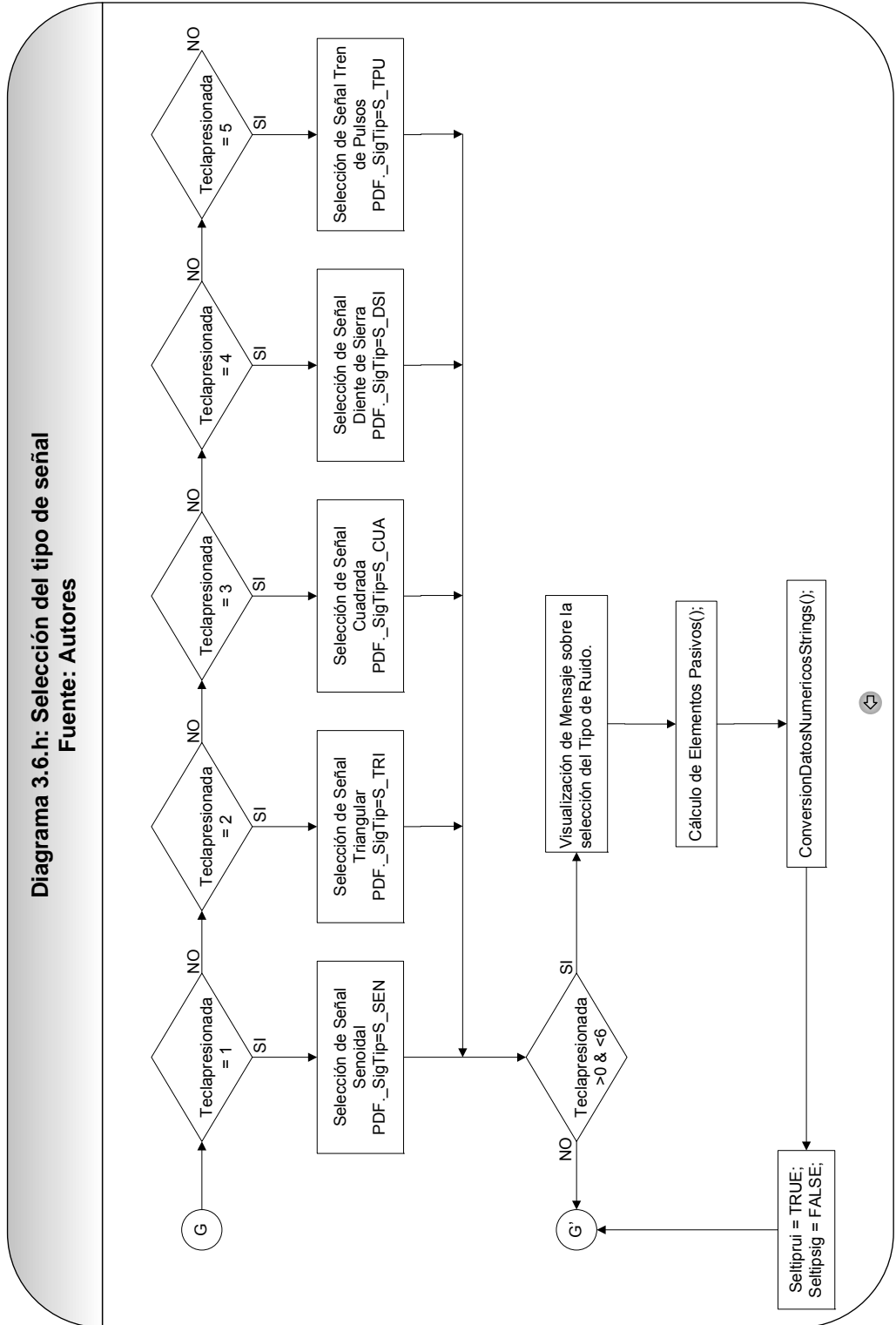
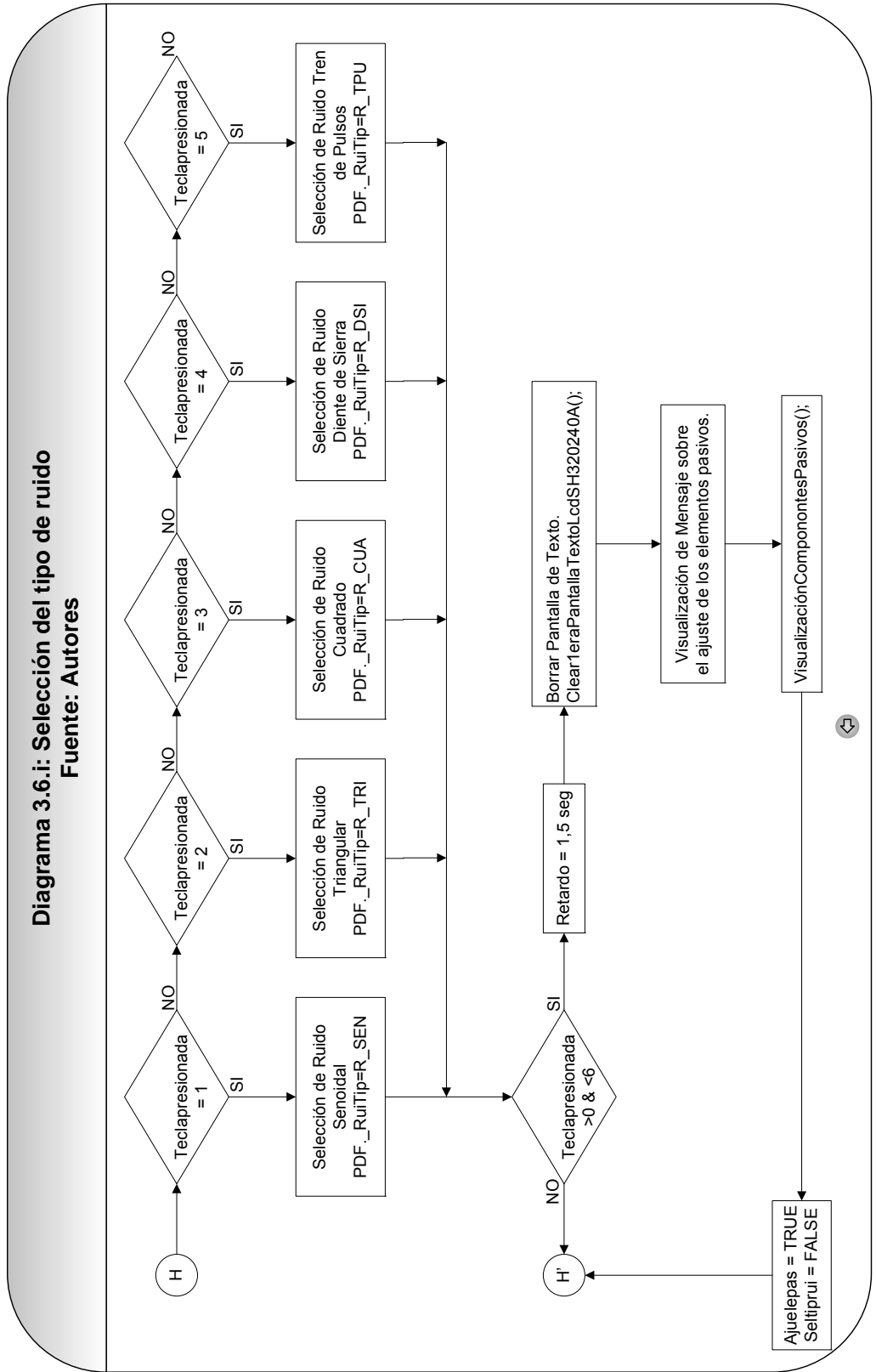
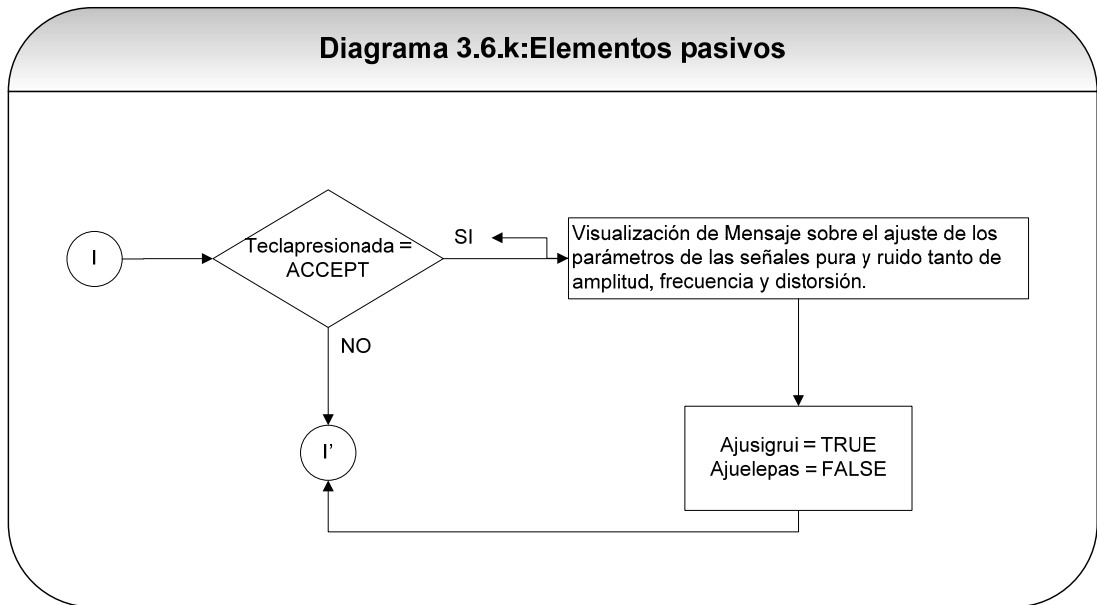
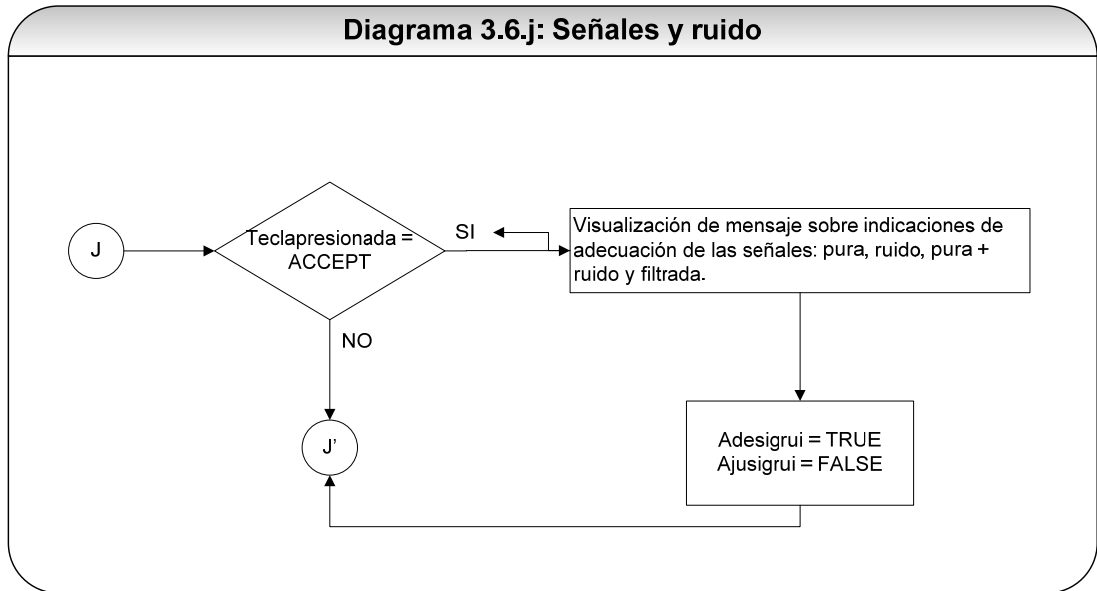
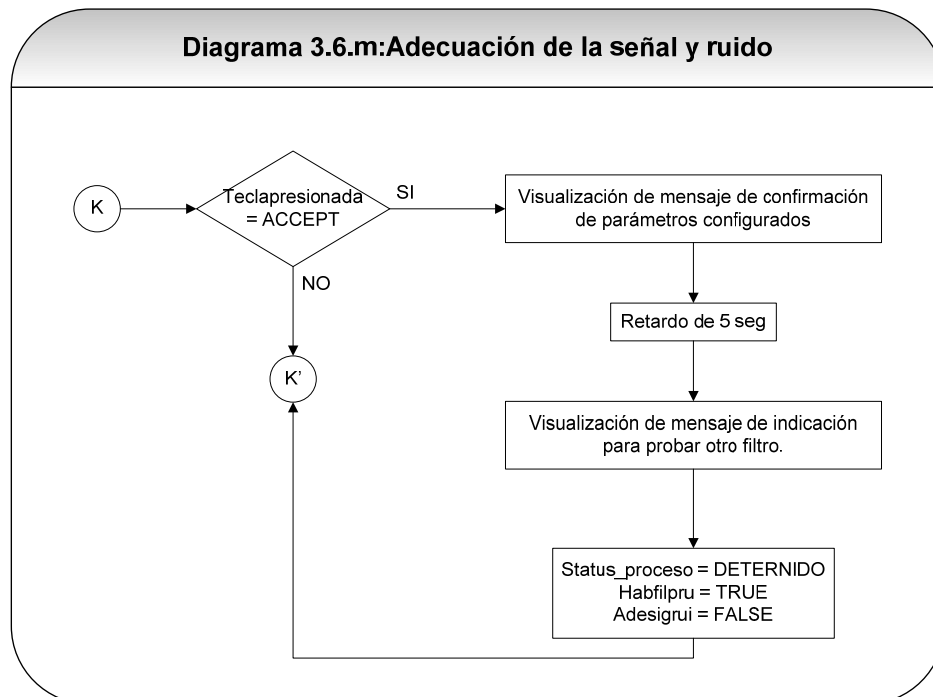
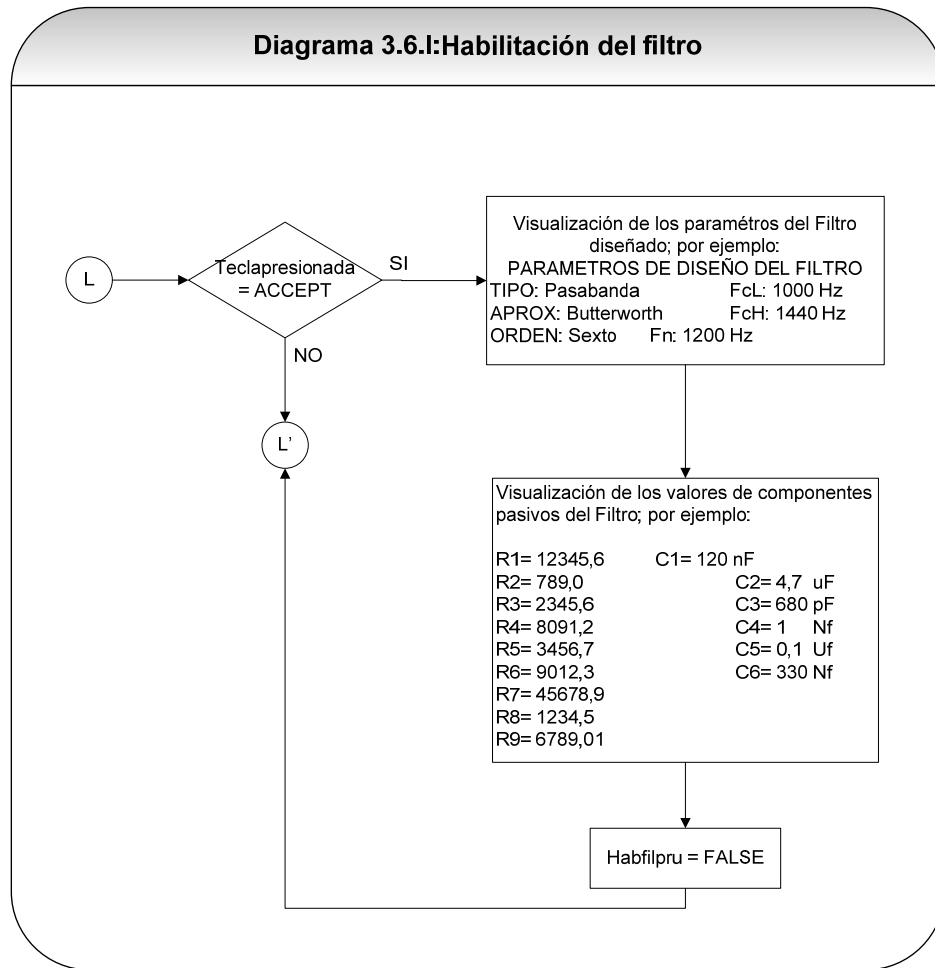


Diagrama 3.6.i: Selección del tipo de ruido
Fuente: Autores







3.7 Conclusiones.

1. Para determinar las direcciones a programa a utilizar se deben seguir, ordenamientos estipulados, de forma general, para cualquier tipo de lenguaje que se va a utilizar.
2. Las subrutinas de programación sirven para ordenar los ingresos y direccionamientos que son necesarios para realizar un cuerpo de programa.
3. Siempre que se sigan todos los ordenamientos necesarios para programar, se obtendrá de manera rápida, la solución o automatización de cada subrutina.
4. Los diagramas de flujo sirven para orientar al programador, a cargar sucesivamente los direccionamientos de una subrutina, demostrando su importancia al momento de programar.
5. Cada proceso tiene su importancia para ser codificado, estos se representan mediante figuras geométricas dentro de su diagrama de flujo específico y ayudan al programador u operador a explicar que es lo que se va a direccionar.
6. Siempre que se necesite lenguaje de programación o flujo grama el operador debe tener presente toda la información sobre el tipo de “software” que desee aplicar, como fue demostrado en las diferentes subrutinas anexadas.
7. Toda información que se desee ingresar debe ser primero comprobada matemáticamente y lógicamente analizada para realizar diagramas de flujo y posterior a esto su codificación en el lenguaje utilizado.
8. Todo microcontrolador soporta su respectivo lenguaje de programación, y siempre que se siga un diagrama ordenado, su ejecución se hará en tiempo corto y de una manera eficaz.

9. Los microcontroladores soportan cierta cantidad de direcciones, debido a su espacio en la memoria interna, pero permiten la comunicación serial con otros microcontroladores para su ejecución.

CONCLUSIONES

Generales

1. La universidad ha hecho grandes esfuerzos por dotar de recursos y materiales que faciliten el proceso de enseñanza – aprendizaje.

En esta perspectiva, el estudio de los filtros realizado por nosotros en las aulas de nuestra Facultad, en su momento, nos motivó para trabajar un producto que facilite la comprensión y la aprehensión del conocimiento por parte de los alumnos y que cumpla además, entre otras, estas funciones:

- que sea didáctico,
 - accesible a los estudiantes,
 - que ponga en práctica lo estudiado en la teoría
 - que el estudiante pueda investigar, verificar, comprobar, las veces que sea necesario, sus ejercicios, sus cálculos, sus problemas relacionados con los filtros electrónicos.
2. Nuestro objetivo: el desarrollo de una herramienta de simulación de circuitos que permita hacer las simulaciones que necesitan los alumnos en su estudio, como paso previo al montaje experimental de estos en el laboratorio.
 3. Para el diseño del simulador se han tomado en cuenta las características particulares de la actividad estudiantil y docente.
 4. El simulador didáctico desarrollado por nosotros, reúne una serie de características que lo hacen ideal para su utilización en entornos docentes.

Acerca al educando a los procesos involucrados en la simulación y diseño de circuitos electrónicos, así como, el acceso a variables internas que facilitan la comprensión de los cálculos involucrados en una simulación.

Permite resolver gran cantidad de ejemplos y manejar el sistema de control de versiones y errores.

5. Tiene, además, algunas funcionalidades que posibilitan el acceso a datos internos de la simulación, así como, su análisis e interpretación.

6. El aprendizaje de cualquier tema o materia se facilita cuando un proceso se lo descompone en partes, esto facilita la percepción del conocimiento por parte del alumno y eso, precisamente, lo que hemos tratado de conseguir con el simulador de filtros electrónicos..
7. El estudio, relacionado con los filtros es extenso; como lo hemos analizado, existen diferentes clases de filtros y con todos ellos se debe tener en cuenta parámetros de diseño, tanto en el dominio de la frecuencia, como en el dominio del tiempo, así como en las especificaciones de partida.
8. Al momento de implementar una aplicación que involucre el trabajo con un filtro, hay que tener claros los procesos de aproximación y realización y sus diferencias claramente especificadas, dentro de este capítulo.
9. Este trabajo es una contribución al desarrollo de la ciencia, que hace posible, la vinculación de la teoría estudiada y desarrollada en clases, con la práctica que facilita la comprobación y demostración de nuestros aprendizajes y saberes.
10. Este simulador se convierte en un elemento formativo y pasa a integrarse al conjunto de auxiliares didácticos que, nosotros creemos, debería ser considerado como un aporte al desarrollo de las TICs (tecnologías de la información y la comunicación)
11. EL ESTUDIANTE tiene, en el simulador electrónico, un recurso tecnológico apropiado para el desarrollo del pensamiento lógico y la eficiente percepción de sus aprendizajes, en materia de filtros electrónicos.
12. EL PROFESOR tiene en sus manos una herramienta tecnológica que, a no dudarlo, le va a proveer de nuevos escenarios para la enseñanza y una metodología moderna y novedosa, desarrollada en la propia universidad.
13. EL PROFESOR dispone de un instrumento tecnológico que le va a servir para descubrir y formar talentos.

El diseño e incorporación de una tecnología, en la formación académica, abre las puertas al saber, al desarrollo del pensamiento y de la ciencia; fomenta la investigación y estimula a los alumnos a poner en juego sus conocimientos, sus capacidades y sobre todo, su creatividad.

Recomendaciones

- En vista de que, desde el Estado, no se promueven políticas de promoción y fomento para el desarrollo de la ciencia, en la rama de la electrónica, creemos que la universidad es la llamada a crear los mecanismos y las condiciones adecuadas para el fomento de la investigación y la aplicación del conocimiento electrónico.
- El Estado y / o la Universidad deben crear incentivos y estímulos que impulsen a los estudiantes, del nivel superior, al emprendimiento de proyectos innovadores orientados a la creación de equipos, aparatos y herramientas que fortalezcan los procesos formativos e instruccionales.
- Las Universidad podría promover el desarrollo y la creación de un banco de tecnologías y recursos auxiliares para la enseñanza y desarrollo de la ciencia, elaborados por sus maestros y educandos.

BIBLIOGRAFÍA

ANGULO, José; ANGULO, Ignacio. *Microcontroladores PIC: Diseño Práctico De Aplicaciones*. España. Editorial McGraw-Hill/Interamericana de España S.A, 2003.

ANGULO USATEGUI, José María; GARCÍA ZAPIRAIN, Begoña; ANGULO MARTÍNEZ, Ignacio. *Microcontroladores Avanzados DsPIC: Controladores Digitales De Señales. Arquitectura, Programación Y Aplicaciones*. España. Editorial Thomson Paraninfo, 2006.

CAMPS VALLS, Gustavo; MUÑOZ MARÍ, Jordi. *Fundamentos De Electrónica Analógica*. España. Universidad de Valencia, 2006.

JAMES, Glyn; BURLEY, David; *Matemáticas Avanzadas Para Ingeniería*. España. Editorial Pearson Educación, 2002.

MANDADO PEREZ, Enrique; MARIÑO ESPIÑEIRA, Perfecto; LAGO FERREIRO Alfonso. *Instrumentación Electrónica*. España. Editorial Marcombo S.A, 1995.

MANDADO PEREZ, Enrique; MENENDEZ FUERTES, Luis Manuel; FERNANDEZ FERREIRA, Manuel; LOPEZ MATOS, Emilio. *Microcontroladores PIC. Sistema Integrado Para El Auto aprendizaje*. España. Editorial Marcombo S.A, 2007.

MONTÓ OLIVERA, Américo. *Diseño de Filtros Analógicos*. Cuba. ISP JAE, 1986.

PINDADO RICO, Rafael; *Electrónica Analógica Integrada: Introducción Al Diseño Mediante Problemas*. España. Editorial Marcombo S.A, 1997.

REMIRO DOMÍNGUEZ, Fernando. *Curso De Programación De Microcontroladores PIC: En MPLAB*. España. Editorial Resistor, 2000.

REFERENCIAS ELECTRÓNICAS.

ALECSA. *Diccionario Informático*. [en línea] Argentina. alecsa.com.ar, 2009 [ref. de noviembre 22 de 2009]. Disponible en Web:
<<http://www.alegsa.com.ar/Dic/subrutina.php>>

ARIAS RODRÍGUEZ, Julia. *Instrumentación Electrónica*. [en línea] España. Universidad Rey Juan Carlos. 2009 [ref. de mayo de 2009]. Disponible en Web:
<http://laimbio08.escet.urjc.es/assets/files/docencia/IE/TEMA2_filtros.pdf>

AVENDAÑO, Luis Enrique. *Realización de Filtros*. [en línea] Universidad Nacional de Colombia Sede Manizales. 2009 [ref. de junio de 2009]. Disponible en Web:
<http://www.virtual.unal.edu.co/cursos/sedes/manizales/4040016/html/cap_5/cap5_1.html#el>

CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea]. España. Universidad de las Palmas de Gran Canaria, 2001. [ref. de junio de 2009]. Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

ESPÍ LÓPEZ, José; ESPI, José; CAMPS VALLS, Gustavo; MAGDALENA BENEDITO, José Rafael. *Síntesis de Redes: Impedancias y Filtros* [en línea]. España. Delta Publicaciones, 2008 [ref. de mayo de 2009]. Disponible en Web: <http://books.google.com.ec/books?id=YbUpcy_14H8C&pg=PA97&dq=filtros+de+bessel&ei=eIBKSsG4GJ-EzASyvl3DBg>

GABIOLA, Francisco J; AL-HADITY, Basil. *Análisis Y Diseño De Circuitos Electrónicos Analógicos. Teoría y Ejercicios Resueltos* [en línea]. España. Visión Net, 2007 [ref. de mayo de 2009]. Disponible en Web: <<http://books.google.com.ec/books?id=BUT9ljPgjRUC&pg=PA388&dq>>

Lenguajes de programación. [en línea] España. lenguajes-de-programación.com, 2009 [ref. de noviembre 22 de 2009]. Disponible en Web: <<http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>>

MICROCHIP. *PIC32MX3XX/4XX Family Data Sheet.* [en línea] U.S.A. Microchip Technology Inc, 2008 [ref. de julio de 2009]. Disponible en Web: <<http://ww1.microchip.com/downloads/en/DeviceDoc/39564c.pdf>>

MICROCHIP. *PIC16F627A/628A/648A Data Sheet.* [en línea] U.S.A. Microchip Technology Inc, 2009 [ref. de noviembre de 2009]. Disponible en Web: <<http://ww1.microchip.com/downloads/en/DeviceDoc/40044G.pdf>>

MIYARA, Federico; *Filtros Activos* [en línea]. Argentina. Universidad Nacional de Rosario, 2004 [ref. de mayo de 2009]. Disponible en Web: <<http://www.fceia.unr.edu.ar/enica3/filtros-t.pdf>>

MOJÓN OJEA, Artemio. *Síntesis de Circuitos Eléctricos y Electrónicos.* [en línea]. España. Universidad de Vigo, 2004. [ref. de 25 de junio de 2009]. Disponible en Web: <http://www.tsc.uvigo.es/BIO/Docencia/SCEE/Clases/Clase_07.pdf> <http://www.tsc.uvigo.es/BIO/Docencia/SCEE/Clases/Clase_04.pdf>

PALLÁS ARENY, Ramón; *Microcontroladores: fundamentos y aplicaciones con PIC* [en línea]. España. Editorial Marcombo, 2007 [ref. de julio de 2009]. Disponible en Web: <http://books.google.com.ec/books?id=ODenKGOHMRkC&dq=microcontroladores&source=gbs_navlinks_s>

SANZ,A; ARTIGAS, J.I. *Diseño de Filtros Activos 1.* [en línea] España. Universidad de Zaragoza, 2003. [ref. de junio de 2009]. Disponible en Web: <http://www.cps.unizar.es/~te/Docencia_archivos/ea_archivos/t4_filtros.pdf>

UCONTROL. *El Microcontrolador*. [en línea] Argentina. ucontrol.com.ar, 2007. [ref. de noviembre de 2009]. Disponible en Web: <http://images.google.com/ec/imgres?imgurl=http://www.ucontrol.com.ar/wiki/images/b/b7/EluC_02a.png&imgrefurl=http://www.ucontrol.com.ar/wiki/index.php/El_microcontrolador&usg=__pthSwcTcvEoytIPKDX_r3mPy3ms=&h=193&w=346&sz=7&hl=es&start=14&um=1&itbs=1&tbnid=ogCZc2klx8aUvM:&tbnh=67&tbnw=120&prev=/images%3Fq%3Darquitectura%2Bvon%2Bnewman%26hl%3Des%26rlz%3D1T4RNWN_esEC300EC301%26sa%3DN%26um%3D1>

ULLOA ROJAS, José Antonio. *Filtros: Aproximación y Síntesis* [en línea]. México. Universidad Iberoamericana, 2005 [ref. de mayo de 2009]. Disponible en Web: <books.google.com/ec/books?id=23HkliM8280C&pg=PP20&dq=filtros+activos+de+bessel&ei=QO5KSsbFMIPcygSrwaXtDQ>

WIKIPEDIA. *Análisis de la Respuesta Temporal de un Sistema* [en línea] U.S.A. wikipedia.org, 2009 [ref. de 23 de junio de 2009]. Disponible en Web: <http://es.wikipedia.org/wiki/An%C3%A1lisis_de_la_respuesta_temporal_de_un_sistema>

WIKIPEDIA. *Microcontrolador Memorias*. [en línea] U.S.A. wikipedia.org, 2009 [ref. de noviembre de 2009]. Disponible en Web: <<http://es.wikipedia.org/wiki/Microcontrolador#Memoria>>

ANEXOS

CAPÍTULO 1
FILTROS ANALÓGICOS ACTIVOS

ANEXO A

Polinomios de Butterworth $B(s)$
para distintos valores de n

n	$B(s_x)$
1	(s_x+1)
2	$(s_x^2+\sqrt{2}s_x+1)$
3	$(s_x+1)(s_x+\sqrt{2}s_x+1)$
4	$(s_x^2+0,7654s_x+1)(s_x^2+1,8478s_x+1)$
5	$(s_x+1)(s_x^2+0,618s_x+1)(s_x^2+1,618s_x+1)$
6	$(s_x^2+0,5176s_x+1)(s_x^2+\sqrt{2}s_x+1)(s_x^2+1,9319s_x+1)$
7	$(s_x+1)(s_x^2+0,445s_x+1)(s_x^2+1,247s_x+1)(s_x^2+1,8019s_x+1)$
8	$(s_x^2+0,3902s_x+1)(s_x^2+1,1111s_x+1)(s_x^2+1,1663s_x+1)(s_x^2+1,9616s_x+1)$
9	$(s_x+1)(s_x^2+0,3473s_x+1)(s_x^2+s_x+1)(s_x^2+1,5321s_x+1)(s_x^2+1,8794s_x+1)$
10	$(s_x^2+0,3129s_x+1)(s_x^2+0,908s_x+1)(s_x^2+\sqrt{2}s_x+1)(s_x^2+1,782s_x+1)(s_x^2+1,9754s_x+1)$

Tabla A.1: Polinomios de Butterworth

Fuente: MONTÓ OLIVERA, Américo. *Diseño de Filtros Analógicos*. Cuba, ISP JAE.1986.

n	b_0	$b_1s_x^1$	$b_2s_x^2$	$b_3s_x^3$	$b_4s_x^4$	$b_5s_x^5$	$b_6s_x^6$	$b_7s_x^7$	$b_8s_x^8$	$b_9s_x^9$	$b_{10}s_x^{10}$
2	1	$\sqrt{2}$	2								
3	1	2	2	1							
4	1	2,6131	3,4142	2,6131	1						
5	1	3,2361	5,2361	5,2361	3,2361	1					
6	1	3,8337	7,4641	9,1416	7,4641	3,8337	1				
7	1	4,4940	10,0978	14,5920	14,5920	10,0978	4,4940	1			
8	1	5,1528	13,1371	21,8262	25,6884	21,8262	13,1371	5,1528	1		
9	1	5,7588	16,5817	31,1634	41,9864	41,9864	31,1634	16,5817	5,7588	1	
10	1	6,3925	20,4317	42,8021	64,8824	74,2334	64,8824	42,8021	20,4317	6,3925	1

Tabla A.2: Polinomios de Butterworth $B(s)$ para distintos valores de n

Fuente: MONTÓ OLIVERA, Américo. *Diseño de Filtros Analógicos*. Cuba, ISP JAE.1986.

CAPÍTULO 1
FILTROS ANALÓGICOS ACTIVOS

ANEXO B

Polinomios de Tschebychev para los distintos valores de n

n	$C_n(\omega_x)$
0	1
1	ω_x
2	$2\omega_x^2 - 1$
3	$4\omega_x^3 - 3\omega_x$
4	$\omega_x^4 - 8\omega_x^2 + 1$
5	$16\omega_x^5 - 20\omega_x^3 + 5\omega$
6	$32\omega_x^6 - 48\omega_x^4 + 18\omega_x^2 - 1$
7	$64\omega_x^7 - 112\omega_x^5 + 56\omega_x^3 - 7\omega_x$
8	$128\omega_x^8 - 256\omega_x^6 + 160\omega_x^4 - 32\omega_x^2 + 1$
9	$256\omega_x^9 - 276\omega_x^7 + 432\omega_x^5 - 120\omega_x^3 + 9\omega$
10	$512\omega_x^{10} - 1280\omega_x^8 + 1120\omega_x^6 - 400\omega_x^4 + 50\omega_x^2 - 1$
.	.
.	.
.	.
n	$C_n(\omega_x) = 2\omega_x C_{n-1}(\omega_x) - C_{n-2}(\omega_x)$

Tabla B.1: Polinomios de Tschebychev para los distintos valores de n

Fuente: MONTÓ OLIVERA, Américo. *Diseño de Filtros Analógicos*. Cuba, ISP JAE.1986.

CAPÍTULO 1
FILTROS ANALÓGICOS ACTIVOS

ANEXO C

Polinomios de Bessel de orden n y ecuación de recurrencia

$B_n(s)$

$B_0(s) = 1$
$B_1(s) = s + 1$
$B_2(s) = s^2 + 3s + 3$
$B_3(s) = s^3 + 6s^2 + 15s + 15$
$B_4(s) = s^4 + 10s^3 + 45s^2 + 105s + 105$
$B_5(s) = s^5 + 15s^4 + 105s^3 + 420s^2 + 495s + 945$
$B_6(s) = s^6 + 210s^4 + 1260s^3 + 4725s^2 + 10395s + 10395$
.
.
.
$B_{n+1}(s) = (2n + 1)B_n(s) + s^2B_{n-1}(s)$

Tabla C.1: Polinomios de Bessel de orden n y ecuación de recurrencia

Fuente: ESPÍ LÓPEZ, José ;ESPI, José; CAMPS VALLS, Gustavo; MAGDALENA BENEDITO, José Rafael.
Síntesis de Redes: Impedancias y Filtros [en línea] [ref. de mayo de 2009].

Disponible en Web:

<http://books.google.com.ec/books?id=YbUpcy_14H8C&pg=PA97&dq=filtros+de+bessel&ei=eIBKSsG4GJ-EzASyvl3DBg>

CAPÍTULO 1
FILTROS ANALÓGICOS ACTIVOS

ANEXO D

Algunos polinomios de Bessel factorizados

$B_2(s) = \frac{1}{3}S^2 + S + 1$
$B_4(s) = (0,109408s^2 + 0,633735s + 1)(0,087049s^2 + 0,366265s + 1)$
$B_6(s) = (0,053188s^2 + 0,451926s + 1)(0,047955s^2 + 0,358293s + 1)(0,037716s^2 + 0,189781s + 1)$

Tabla D.1: Polinomios de Bessel factorizados

Fuente: ESPÍ LÓPEZ, José ;ESPI, José; CAMPS VALLS, Gustavo; MAGDALENA BENEDITO, José Rafael.
Síntesis de Redes: Impedancias y Filtros [en línea] [ref. de mayo de 2009].

Disponible en Web:

<http://books.google.com.ec/books?id=YbUpcy_14H8C&pg=PA97&dq=filtros+de+bessel&ei=eIBKSsG4GJ-EzASyvl3DBg>

CAPÍTULO 1
FILTROS ANALÓGICOS ACTIVOS

ANEXO E

**Nomenclatura de los coeficientes para
filtros tipo Sallen Key.**

n	Orden del filtro
i	Número del filtro parcial
a _i , b _i	Coeficientes del filtro
K _i	Cociente entre la frecuencia de corte de cada filtro parcial con respecto a la frecuencia de corte del filtro total.
Q _i	Factor de calidad de cada filtro parcial
T _{gro}	Retardo normalizado para los filtros pasa-todo

Tabla E.1: Nomenclatura de los coeficientes para filtros tipo Sallen Key

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009]. Disponible en Web:
<<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

CAPÍTULO 1
FILTROS ANALÓGICOS ACTIVOS

ANEXO F

Tabla de coeficientes para filtros Sallen Key
con aproximación de Butterworth.

n	i	ai	bi	ki	Qi
1	1	1,0000	0,0000	1,0000	-----
2	1	1,4142	1,0000	1,0000	0,71
3	1	1,0000	0,0000	1,0000	-----
	2	1,0000	1,0000	1,2720	1,00
4	1	1,8478	1,0000	0,7190	0,54
	2	0,7654	1,0000	1,3900	1,31
5	1	1,0000	0,0000	1,0000	-----
	2	1,6180	1,0000	0,8590	0,62
	3	0,6180	1,0000	1,4480	1,62

Tabla F.1: Coeficientes con aproximación de Butterworth

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009]. Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

CAPÍTULO 1
FILTROS ANALÓGICOS ACTIVOS

ANEXO G

Tabla de coeficientes para filtros Sallen Key

con aproximación de Tschebychev.

n	i	ai	bi	ki	Qi
1	1	1,0000	0,0000	1,0000	-----
2	1	1,3022	1,5515	1,0000	0,96
3	1	2,2156	0,0000	0,4510	-----
	2	0,5442	1,2057	1,3530	2,02
4	1	2,5904	4,1301	0,5400	0,78
	2	0,3039	1,1697	1,4170	3,56
5	1	3,5711	0,0000	0,2800	-----
	2	1,1280	2,4896	0,8940	1,40
	3	0,1872	1,0814	1,4860	5,56

Tabla G.1: Coeficientes con aproximación Tschebychev a 1-dB de rizado

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009]. Disponible en Web:
<<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

n	i	ai	bi	ki	Qi
1	1	1,0000	0,0000	1,0000	-----
2	1	1,1813	1,7775	1,0000	1,13
3	1	2,7994	0,0000	0,3570	-----
	2	0,4300	1,2036	1,3780	2,55
4	1	2,4025	4,9862	0,5500	0,93
	2	0,2374	1,1896	1,4130	4,59
5	1	4,6345	0,0000	0,2160	-----
	2	0,9090	2,6036	0,9080	1,78
	3	0,1434	1,0750	1,4930	7,23

Tabla G.2: Coeficientes con aproximación Tschebychev a 2-dB de rizado

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009]. Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

n	i	ai	bi	ki	Qi
1	1	1,0000	0,0000	1,0000	-----
2	1	1,065	1,9305	1,0000	1,3
3	1	3,3496	0,0000	1,2990	-----
	2	0,3559	1,1923	1,3960	3,07
4	1	2,1853	5,5339	0,5570	1,08
	2	0,1964	1,2009	1,4100	5,58
5	1	5,6334	0,0000	0,1780	-----
	2	0,7620	2,6530	0,9170	2,14
	3	0,1172	1,0686	1,5000	8,82

Tabla G.3: Coeficientes con aproximación Tschebychev a 3-dB de rizado

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009]. Disponible en Web: <<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

CAPÍTULO 1
FILTROS ANALÓGICOS ACTIVOS

ANEXO H

Tabla de coeficientes para filtros Sallen Key
con aproximación de Bessel.

n	i	ai	bi	ki	Qi
1	1	1,0000	0,0000	1,0000	-----
2	1	1,3617	0,6180	1,0000	0,58
3	1	0,7560	0,0000	1,3230	-----
	2	0,9996	0,4772	1,4140	0,69
4	1	1,3397	0,4889	0,9780	0,52
	2	0,7743	0,3890	1,7970	0,81
5	1	0,6656	0,0000	1,5020	-----
	2	1,1402	0,4128	1,1840	0,56
	3	0,6216	0,3245	2,1380	0,92

Tabla H.1: Coeficientes con aproximación de Bessel

Fuente: CABRERA PEÑA, José. *Filtros Activos, Teoría* [en línea] [ref. de junio de 2009]. Disponible en Web:
<<http://www.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>>

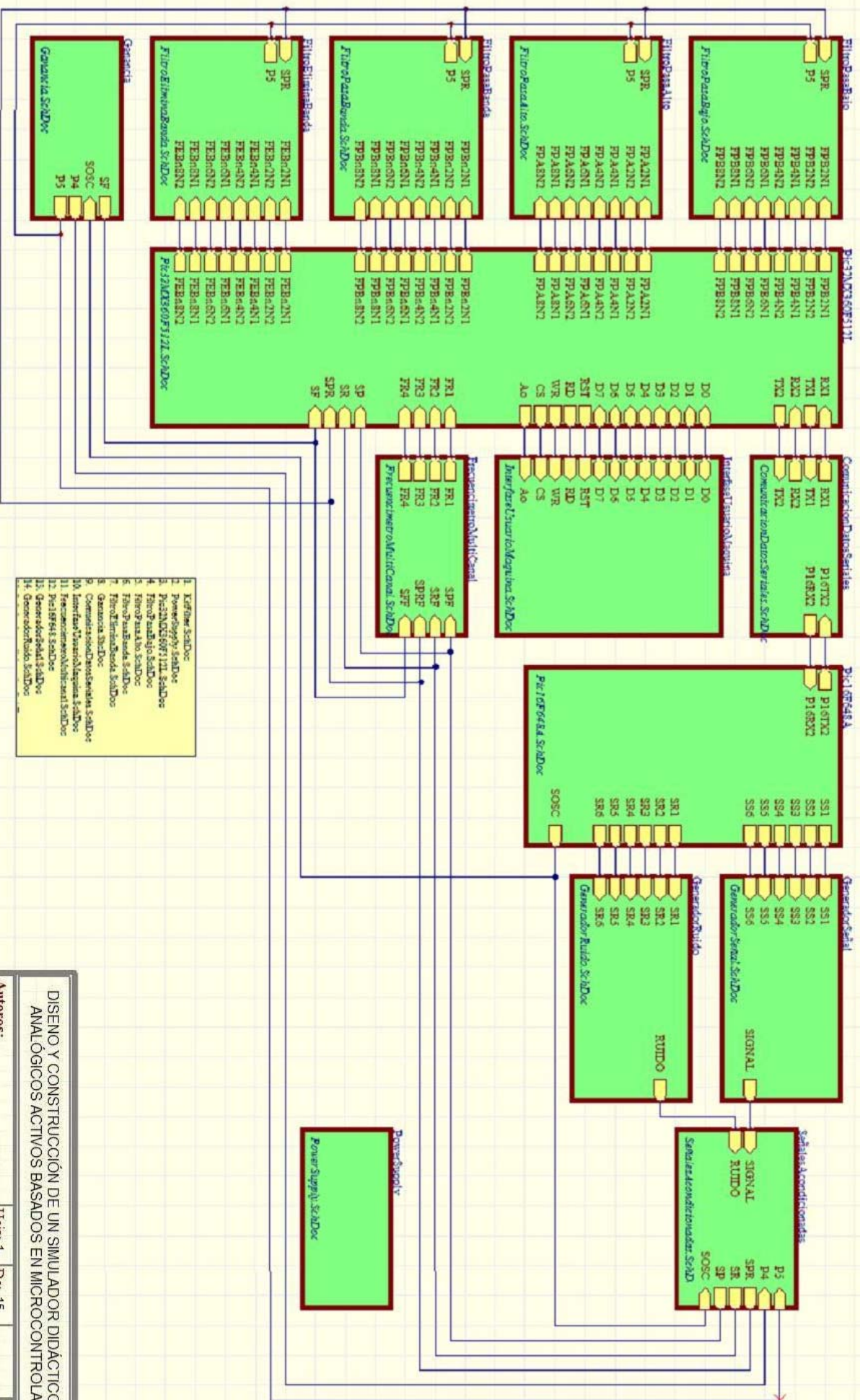
CAPÍTULO 2
IMPLEMENTACIÓN DE HARDWARE.

ANEXO A

Figura A.1: Diagrama a bloques del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

DIAGRAMA DE BLOQUES PARA UN SIMULADOR DIDÁCTICO DE FILTROS ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32



DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32

Autores:
Xavier Fernando Marin Tenorio
Juan Francisco Solano López

Hoja: 1 De: 15
ENERO DE 2010



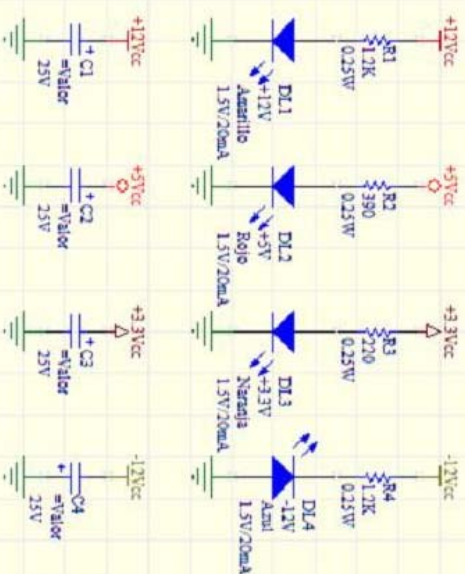
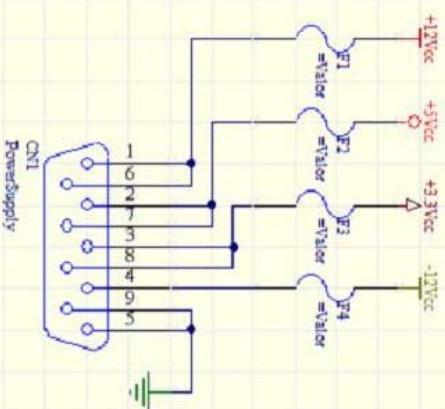
CAPÍTULO 2
IMPLEMENTACIÓN DE HARDWARE

ANEXO B

Figura B.1: Circuito para fuente de alimentación del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

FUENTE DE ALIMENTACIÓN +12Vcc, +5Vcc, +3.3Vcc, -12Vcc



- | | | |
|---|-----|---------|
| 1 | DL1 | 8.175mA |
| 2 | DL2 | 8.97mA |
| 3 | DL3 | 8.185mA |
| 4 | DL4 | 8.175mA |

DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS
ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32

Autores:

Xavier Fernando Marin Tenorio
Juan Francisko Solano López

Hoja: 2 | De: 15

ENERO DE 2010

FUENTE DE ALIMENTACIÓN



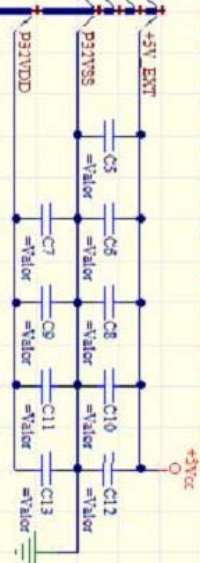
CAPÍTULO 2
IMPLEMENTACIÓN DE HARDWARE

ANEXO C

Figura C.1: Circuito para conexiones del PIC32MX360F512L del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

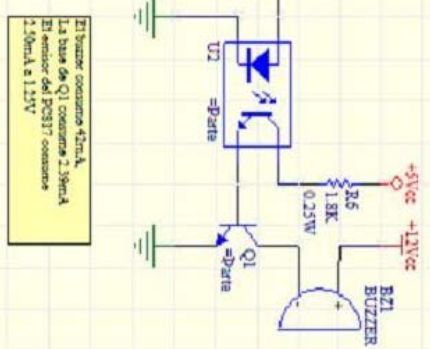
PIC32MX360F512L



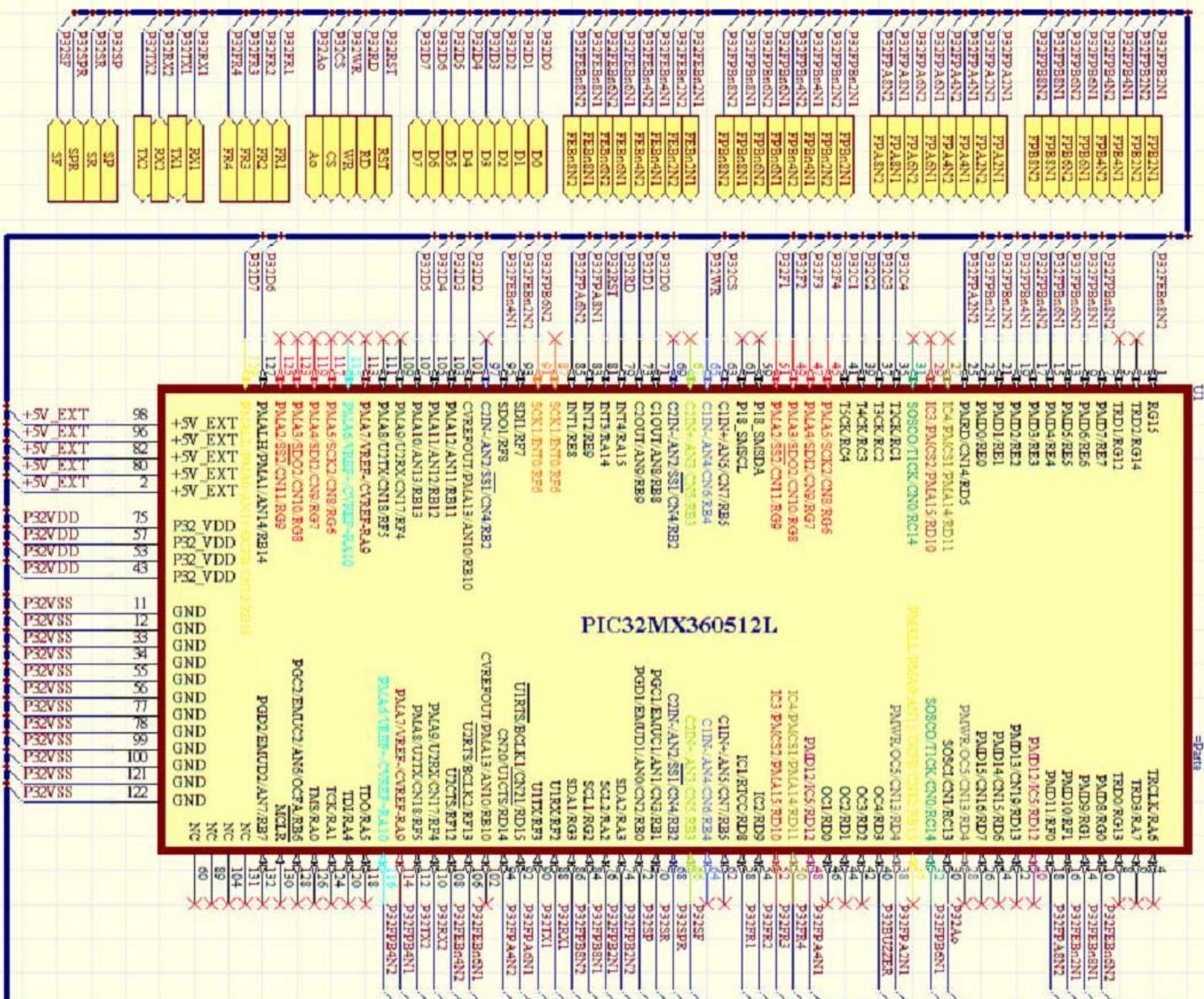
Una pila de 4 x LR se llama
Inerc. 210mA a 1.5V

C1	Teclado
C2	Memorial
C3	
C4	
F1	6 5 4
F2	7 8 9
F3	0
F4	1

TECLADO NUMERICAL



El buzzer consume 40mA.
La base de Q1 consume 2.5mA.
El sensor de PIC32 consume
2.50mA a 1.5V.



DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS
ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32

Autores:
Xavier Ferrando Marín Tenorio
Juan Francisco Solano López

Hoja: 3 Dec: 15
ENERO DE 2010



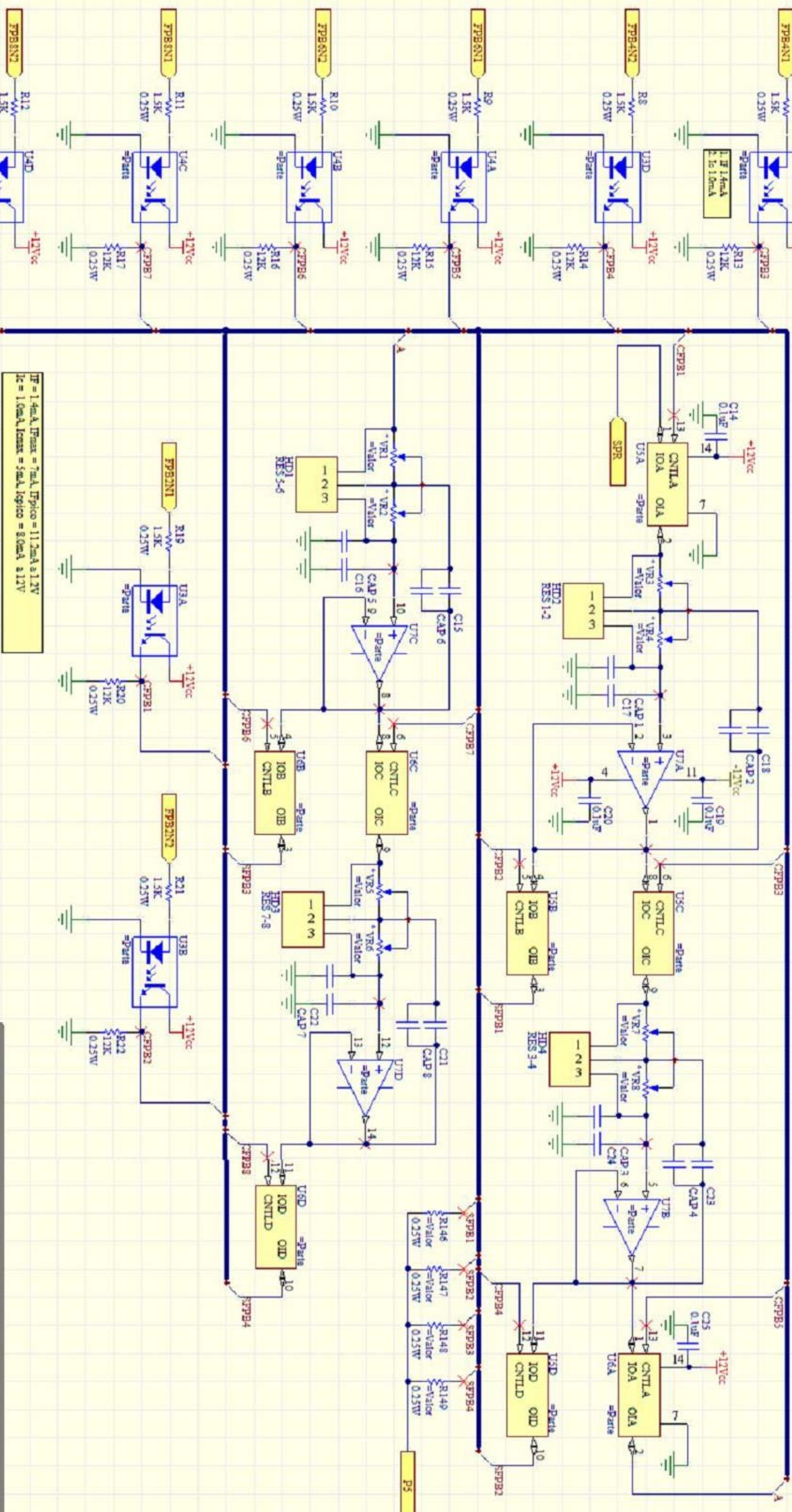
CAPÍTULO 2
IMPLEMENTACIÓN DE HARDWARE

ANEXO D

Figura D.1: Diseño de circuito para el filtro Pasa Bajo del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

FILTROS PASA BAJO Células No Inversoras de 2do - 4to - 6to - 8vo Orden



$I_F = 1.4mA$, $I_{max} = 7mA$, $I_{tipico} = 11.2mA$ a $1.1V$
 $I_c = 1.0mA$, $I_{max} = 5mA$, $I_{logico} = 80mA$ a $1.1V$

DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32
Autores:
 Xavier Fernando Marín Tenorio
 Juan Francisco Solano López

Hoja: 4	Dec 15
ENERO DE 2010	

FILTRO PASA BAJO DE 2do, 4to, 6to, 8vo ORDEN



CAPÍTULO 2
IMPLEMENTACIÓN DE HARDWARE

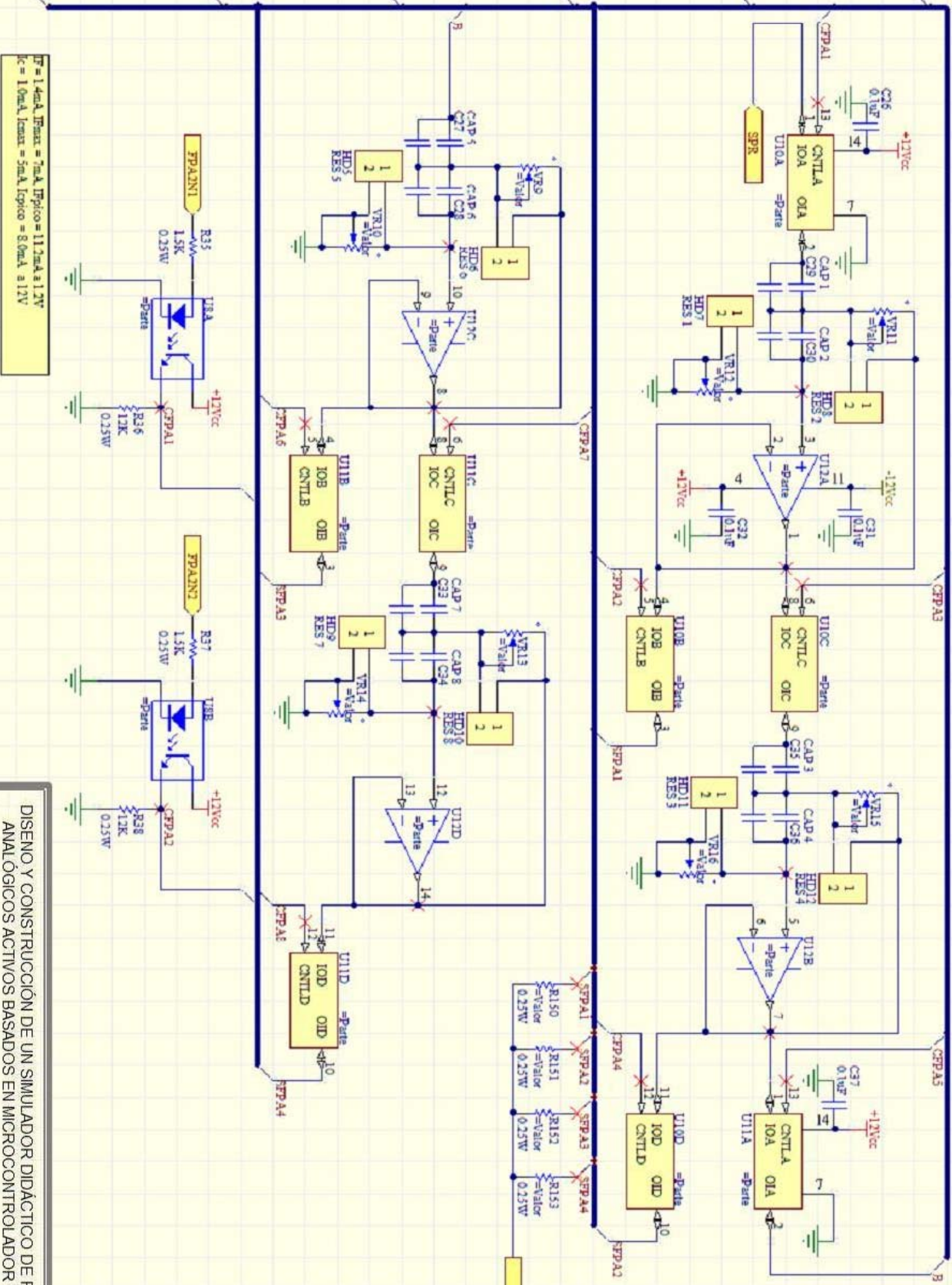
ANEXO E

Figura E.1: Diseño de circuito para el filtro Pasa Alto del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

FILTROS PASA ALTO

Células No Inversoras de 2do - 4to - 6to - 8vo Orden



$f_c = 1 \text{ kHz}$, $f_{max} = 7 \text{ kHz}$, $f_{pico} = 11.7 \text{ kHz}$, 12 V
 $I_c = 10 \text{ mA}$, $I_{max} = 5 \text{ mA}$, $I_{pico} = 8 \text{ mA}$, a 12 V

DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS
 ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32

Autores:
 Xavier Fernando Marín Tenorio
 Juan Francisco Solano López

Hoja: 5 Dec: 15
 ENERO DE 2010

FILTRO PASA ALTO DE 2do, 4to, 6to, 8vo ORDEN



CAPÍTULO 2
IMPLEMENTACIÓN DE HARDWARE

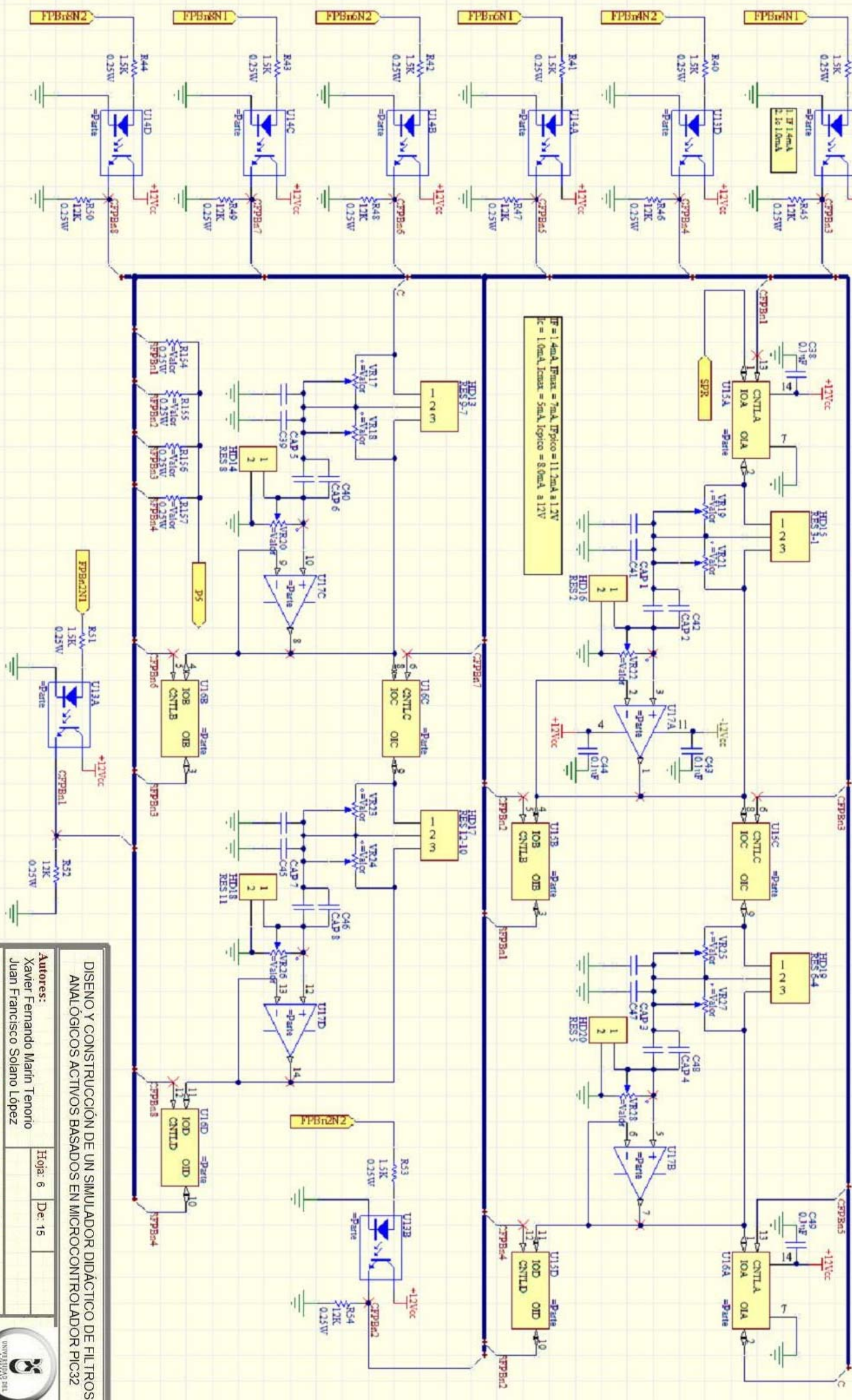
ANEXO F

Figura F.1: Diseño de circuito para el filtro Pasa Banda del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

FILTROS PASA BANDA

Células No Inversoras de 2do - 4to - 6to - 8vo Orden



DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS
 ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32

Autores:
 Xavier Ferrando Marin Tenorio
 Juan Francisco Solano Lopez

Hoja: 6 De: 15

FILTRO PASA BANDA DE 2do, 4to, 6to, 8vo ORDEN



CAPÍTULO 2
IMPLEMENTACIÓN DE HARDWARE

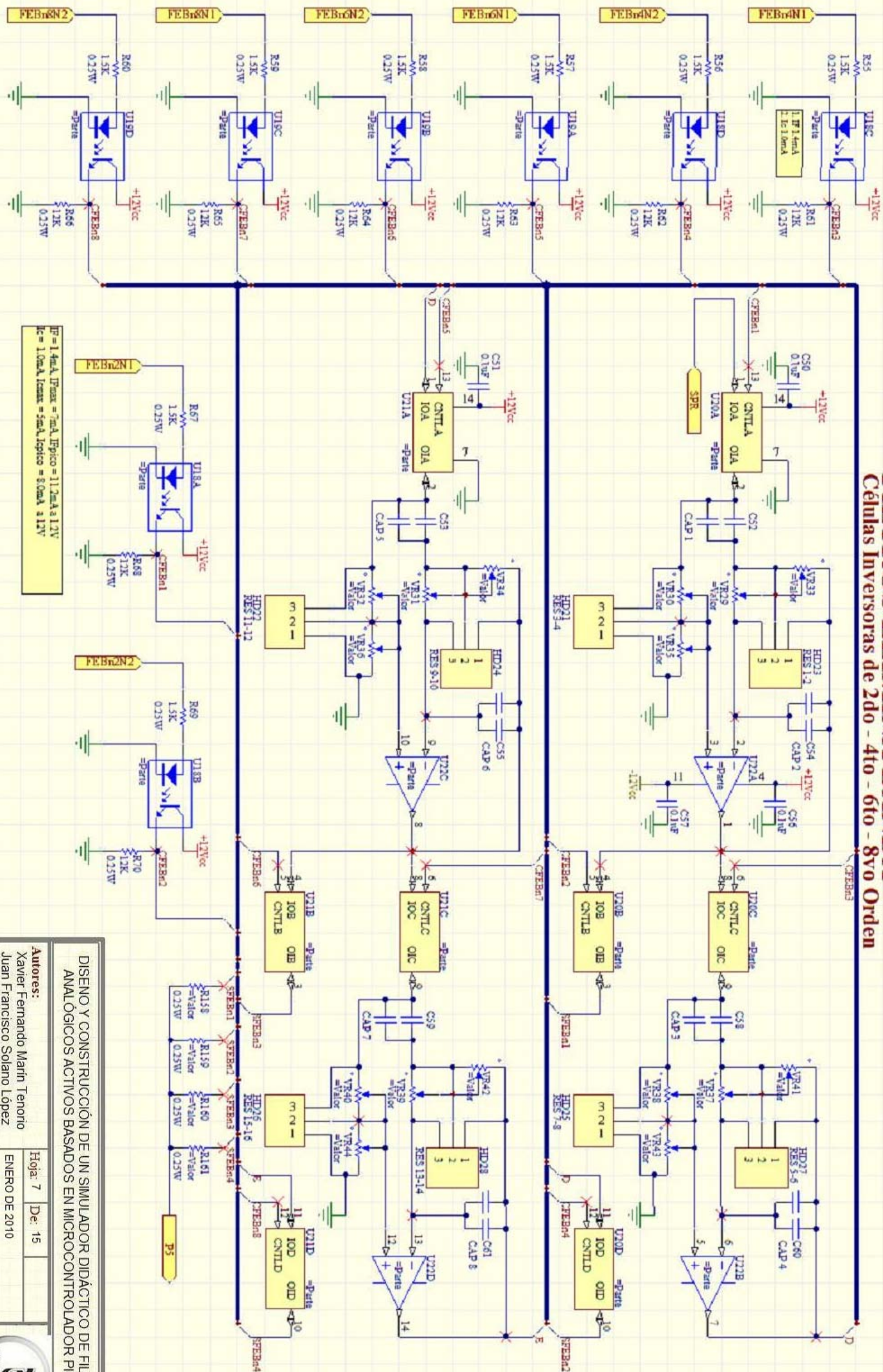
ANEXO G

Figura G.1: Diseño de circuito para el filtro Elimina Banda del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

FILTROS ELIMINA BANDA

Células Inversoras de 2do - 4to - 6to - 8vo Orden



DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32

Autores:
 Xavier Fernando Marín Tenorio
 Juan Francisco Solano López

Hoja: 7 Dec. 15
 ENERO DE 2010

FILTROS ELIMINA BANDA DE 2do, 4to, 6to, 8vo ORDEN



CAPÍTULO 2

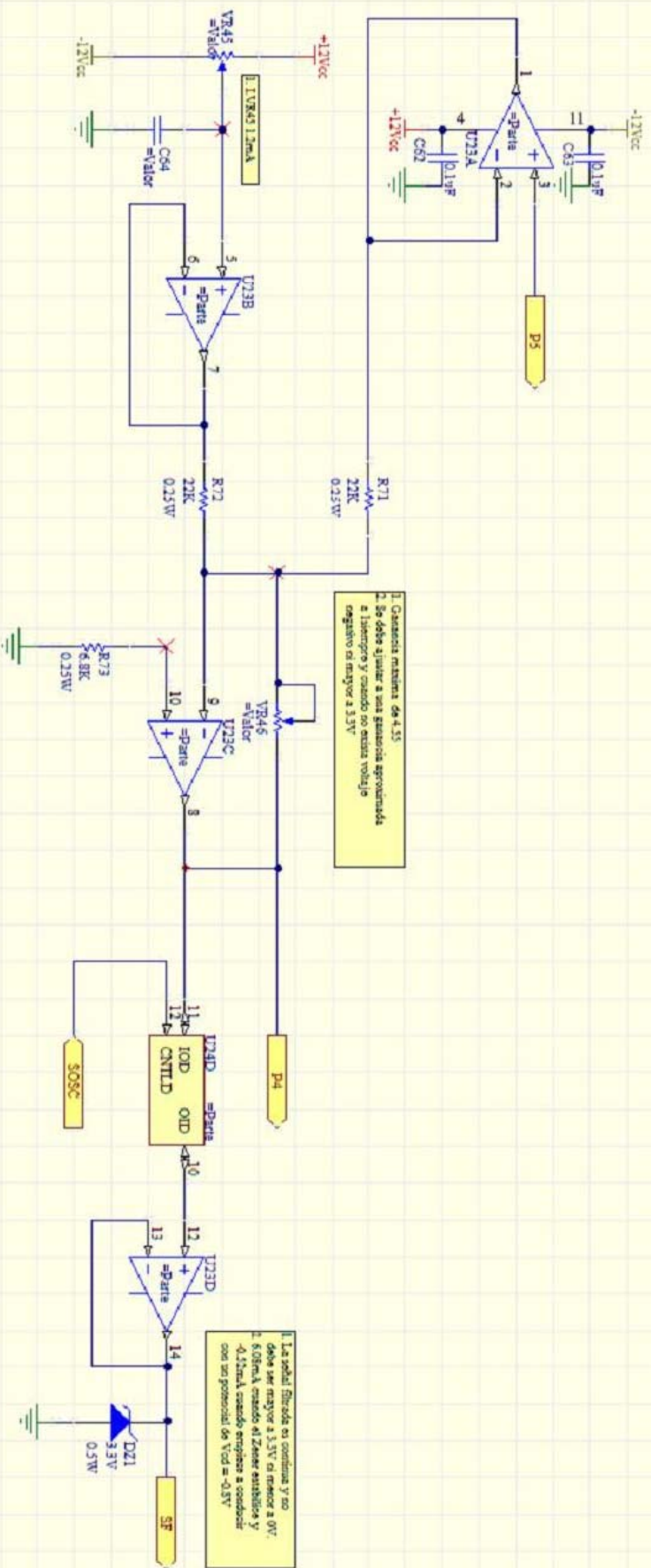
IMPLEMENTACIÓN DE HARDWARE

ANEXO H

Figura H.1: Diseño de circuito para ganancia de la señal filtrada del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

GANANCIA DE LA SEÑAL FILTRADA



DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS
ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32

Autores:
Xavier Fernando Marin Tenorio
Juan Francisco Solano López

Hoja: 8 Dec: 15
ENERO DE 2010



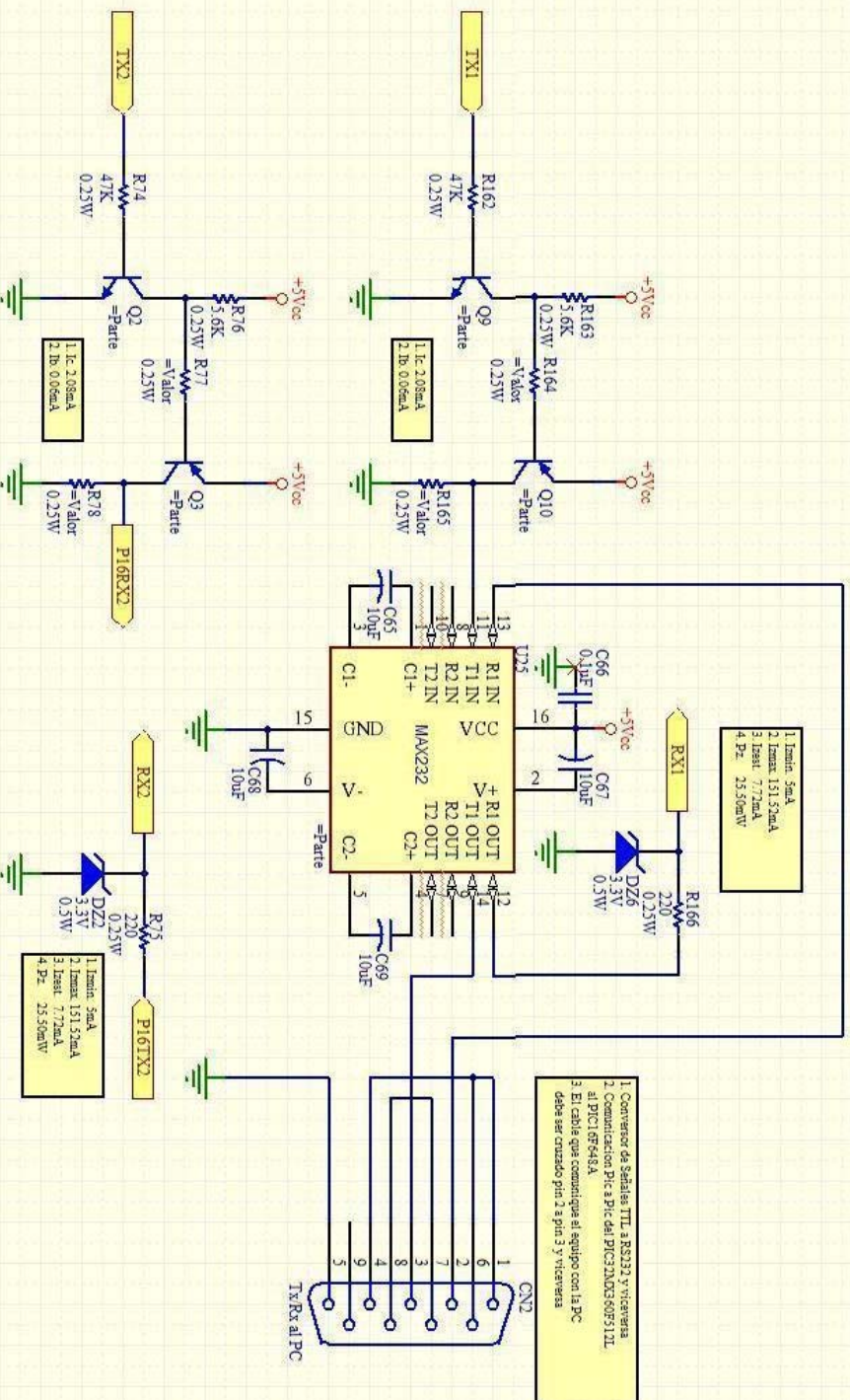
CAPÍTULO 2
IMPLEMENTACIÓN DE HARDWARE

ANEXO I

Figura I.1: Diseño de circuito para la comunicación de datos seriales del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

COMUNICACIÓN DE DATOS SERIALES



- 1 Conector de Salida TTL a RS232 configurada
- 2 Conector de Pic a Pic del PIC32MDS60P31L
- 3 El cable que comunica el equipo con la PC debe ser cruzado pin 2 a pin 3 y viceversa

DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32

Autores:
Xavier Ferrando Marín Tenorio
Juan Francisco Solano López

Hoja: 9 De: 15
ENERO DE 2010



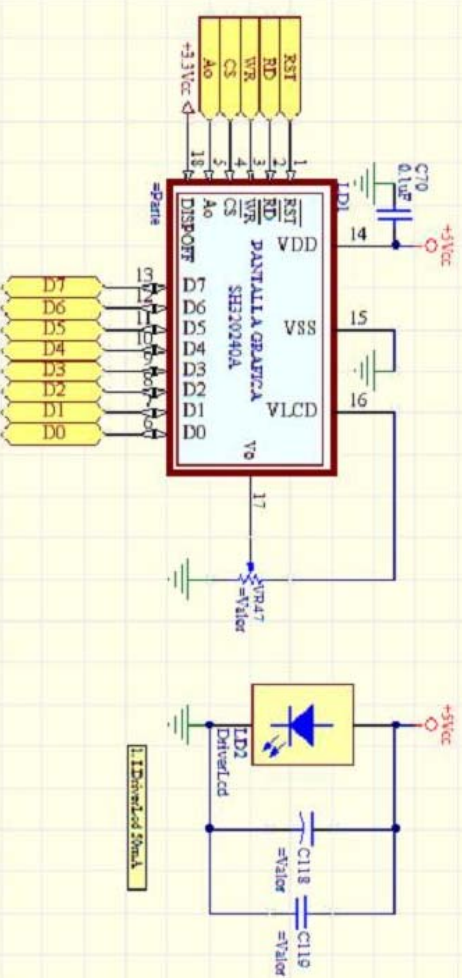
CAPÍTULO 2
IMPLEMENTACIÓN DE HARDWARE

ANEXO J

Figura J.1: Conexión para el interface usuario máquina del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

INTERFASE USUARIO MÁQUINA



DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS
ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32

Autores:

Xavier Fernando Marin Tenorio
Juan Francisco Solano López

Hoja: 10 De: 15

ENERO DE 2010

INTERFASE USUARIO MÁQUINA



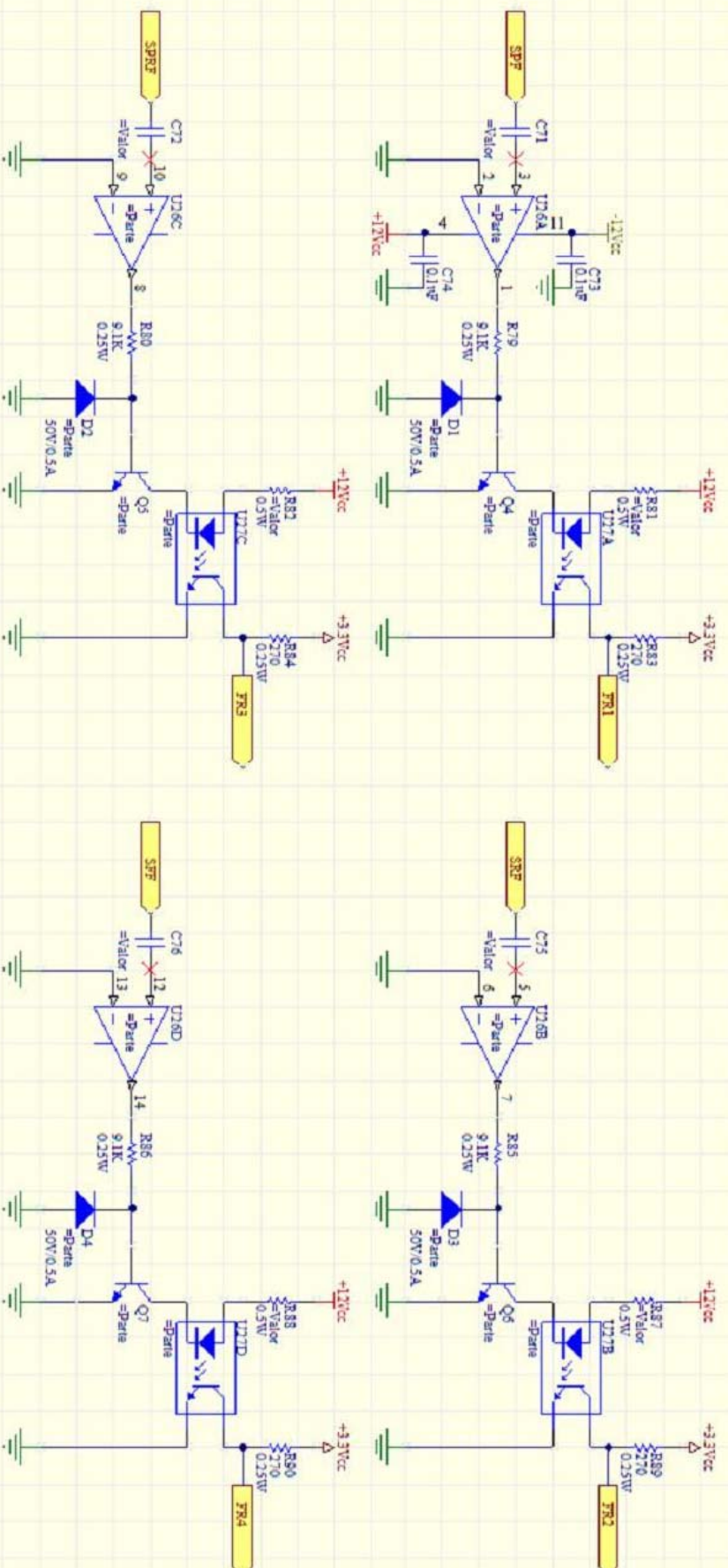
CAPÍTULO 2
IMPLEMENTACIÓN DE HARDWARE

ANEXO K

Figura K.1: Diseño de circuito para el frecuencímetro multicanal del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

FRECUENCIÓMETRO MULTICANAL 52.08KHZ max.



Resistor	Value	Power	Frequency	Transistor	V _{BE}	R _{TH}
R79	20k	25W	50	9.09mA	45.45kHz	48.6%
R80	20k	25W	50	9.09mA	45.45kHz	48.6%
R81	20k	25W	50	9.09mA	45.45kHz	48.6%
R82	20k	25W	50	9.09mA	45.45kHz	48.6%
R83	20k	25W	50	9.09mA	45.45kHz	48.6%
R84	20k	25W	50	9.09mA	45.45kHz	48.6%
R85	20k	25W	50	9.09mA	45.45kHz	48.6%
R86	20k	25W	50	9.09mA	45.45kHz	48.6%
R87	20k	25W	50	9.09mA	45.45kHz	48.6%
R88	20k	25W	50	9.09mA	45.45kHz	48.6%

DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32

Autores:
 Xavier Ferrando Marin Tenorio
 Juan Francisco Solano López

Hoja: 11 Dec: 15
 ENERO DE 2010

FRECUENCIÓMETRO MULTICANAL



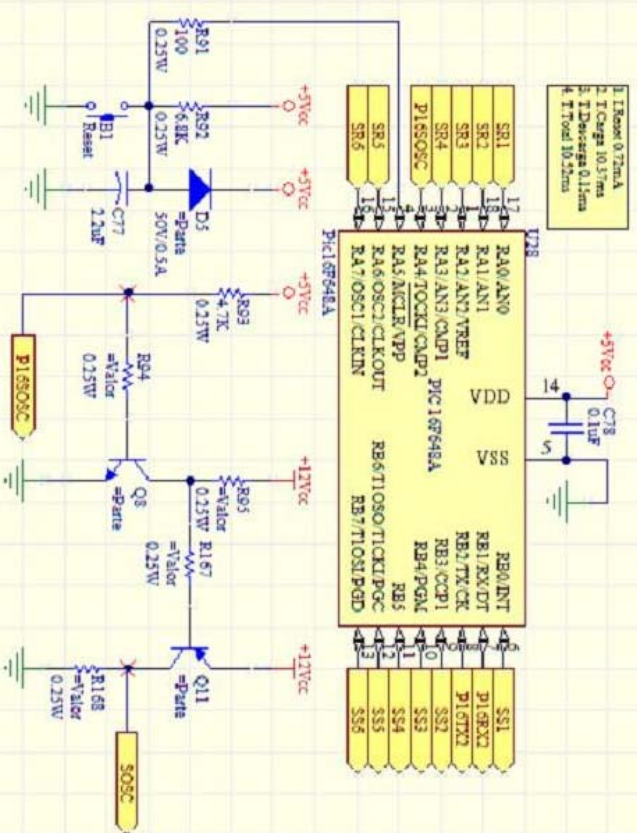
CAPÍTULO 2
IMPLEMENTACIÓN DE HARDWARE

ANEXO L

Figura L.1: Conexión para el PIC16F648A del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

PIC16F648A



- 1. I Bias: 0.75mA
- 2. I Cmp: 10.57mA
- 3. I De: 0.012mA
- 4. I Tot: 10.27mA

DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32

Autores:
Xavier Fernando Marín Tenorio
Juan Francisco Solano López

Hoja: 12 De: 15
ENERO DE 2010



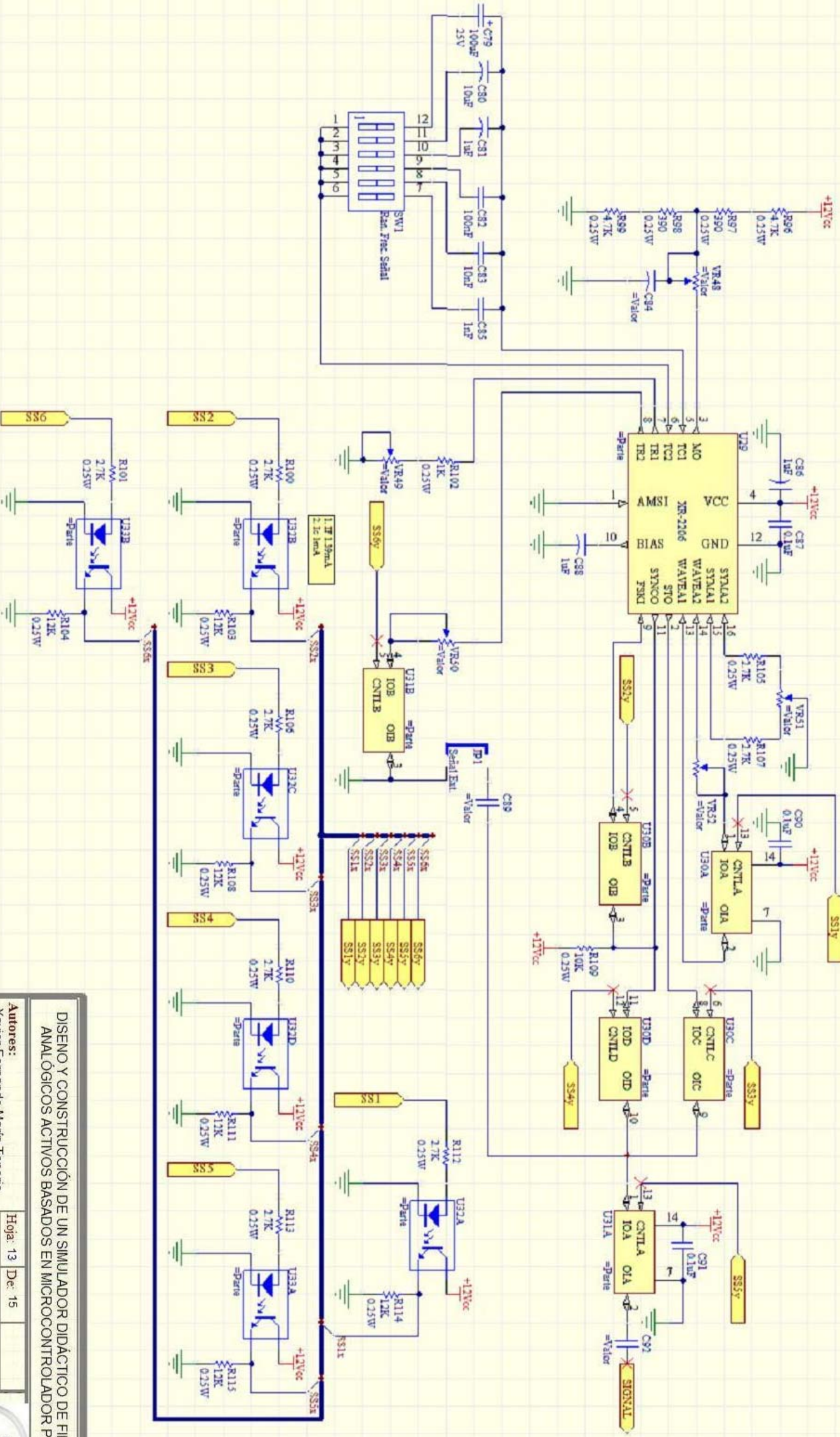
CAPÍTULO 2
IMPLEMENTACIÓN DE HARDWARE

ANEXO M

Figura M.1: Diseño de circuito para el generador de funciones de señal pura del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

GENERADOR DE FUNCIONES DE LA SEÑAL PURA



DISENO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32

Autores:

Xavier Ferrando Marín Tenorio
Juan Francisco Solano López

Hoja: 13 De: 15

ENERO DE 2010

GENERADOR DE FUNCIONES DE LA SEÑAL PURA



CAPÍTULO 2

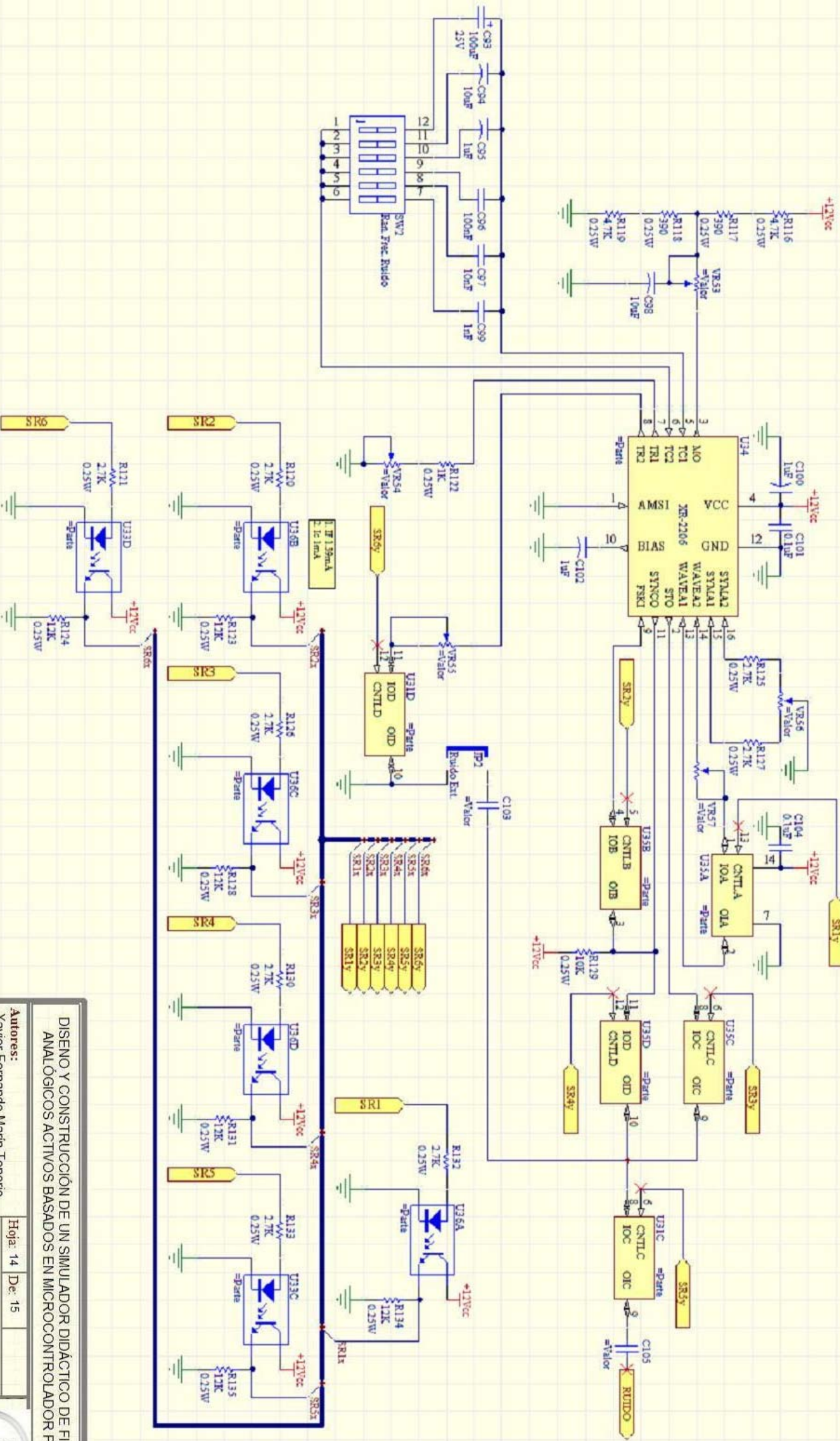
IMPLEMENTACIÓN DE HARDWARE

ANEXO N

Figura N.1: Diseño de circuito para el generador de funciones de señal de ruido del Simulador Didáctico de filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

GENERADOR DE FUNCIONES DE LA SEÑAL DE RUIDO



DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32

Autores:
Xavier Ferrando Marin Tenorio
Juan Francisco Solano López

Hoja: 14 | Dec: 15
ENERO DE 2010

GENERADOR DE FUNCIONES DE LA SEÑAL DE RUIDO



CAPÍTULO 2

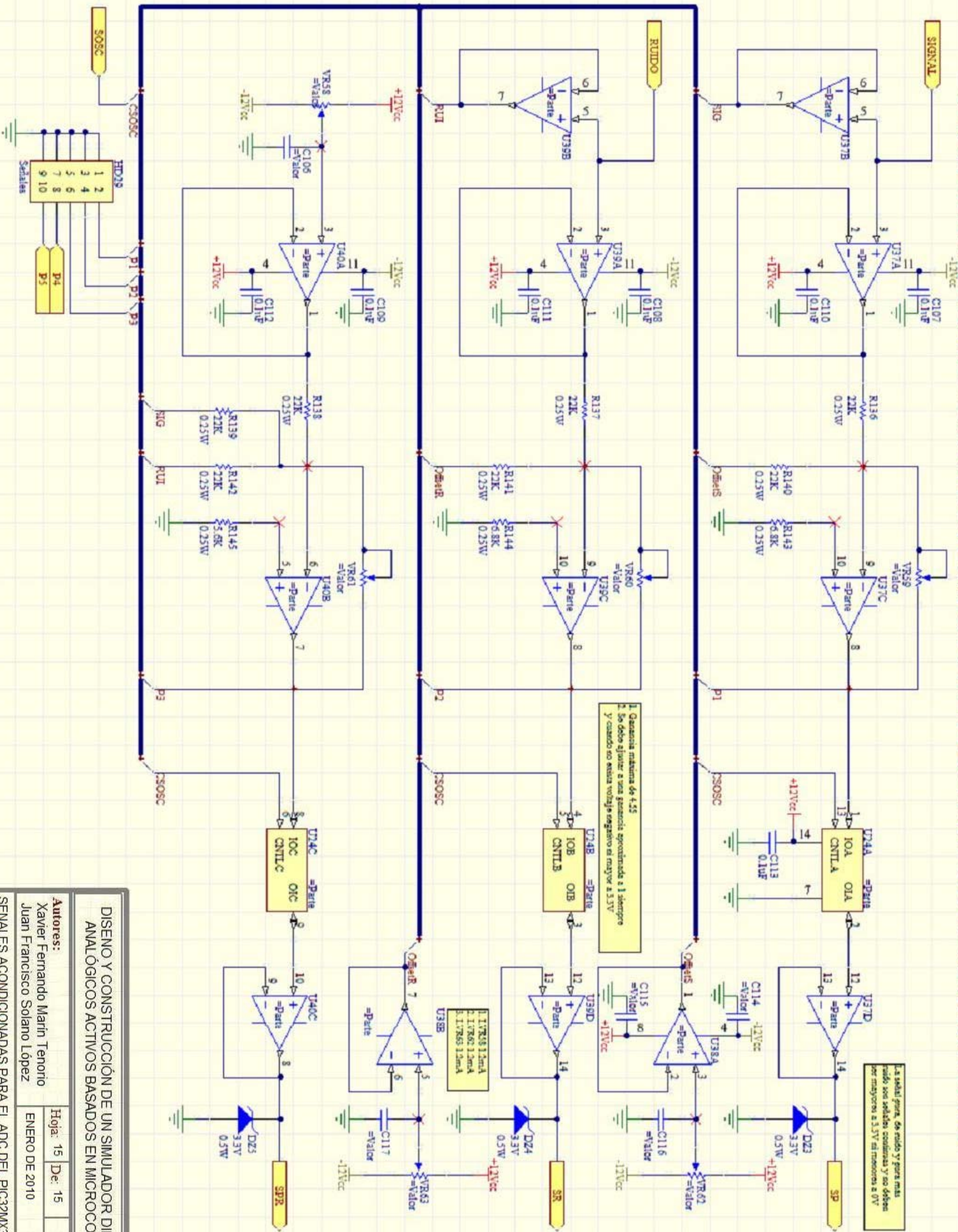
IMPLEMENTACIÓN DE HARDWARE

ANEXO O

Figura O.1: Diseño de circuito de señales acondicionadas para el ADC del PIC32MX360F512L del Simulador Didáctico de Filtros Analógicos Activos Basado en Microcontrolador PIC32.

Fuente: Autores.

SEÑALES ACONDICIONADAS PARA EL ADC DEL PIC32MX360F512L



DISEÑO Y CONSTRUCCIÓN DE UN SIMULADOR DIDÁCTICO DE FILTROS
 ANALÓGICOS ACTIVOS BASADOS EN MICROCONTROLADOR PIC32

Autores:
 Xavier Fernando Marín Tenorio
 Juan Francisco Solano López

Hoja: 15 De: 15
 ENERO DE 2010

SENALES ACONDICIONADAS PARA EL ADC DEL PIC32MX360F512L

