

DISEÑO Y FABRICACIÓN DE UN MÓDULO DIDÁCTICO BASADO EN dsPIC
COMO HERRAMIENTA DE APOYO EN EL APRENDIZAJE DE SISTEMAS DE
CONTROL DIGITAL CON ÉNFASIS EN LA IMPLEMENTACIÓN DE FILTROS
DIGITALES.

YESID ERASMO MENESES YARURO



UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERIAS
FACULTAD DE INGENIERA ELECTRÓNICA
BUCARAMANGA
2011

DISEÑO Y FABRICACIÓN DE UN MÓDULO DIDÁCTICO BASADO EN dsPIC
COMO HERRAMIENTA DE APOYO EN EL APRENDIZAJE DE SISTEMAS DE
CONTROL DIGITAL CON ÉNFASIS EN LA IMPLEMENTACIÓN DE FILTROS
DIGITALES.

YESID ERASMO MENESES YARURO

PROYECTO DE GRADO

CARLOS GERARDO HERNÁNDEZ CAPACHO
DIRECTOR DEL PROYECTO

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERIAS
FACULTAD DE INGENIERÍA ELECTRÓNICA
BUCARAMANGA
2011

Nota de Aceptación

Firma del Jurado

Firma del Jurado

Bucaramanga, Julio del 2011

DEDICATORIA

Dedico este proyecto y mi carrera universitaria a Dios y a la Virgen, por enseñarme el camino correcto de la vida, la salud, la esperanza, las capacidades y herramientas para hacerlo todo posible, estando a mi lado guiándome y dándome la fortaleza de seguir adelante cada día rompiendo todas las barreras que se presenten.

A toda mi familia, en especial a mis padres, Teresa Yaruro y Erasmo Meneses, por creer y confiar siempre en mí, quienes con su motivación, educación y apoyo incondicional me enseñaron a luchar para alcanzar mis metas; por estar siempre pendientes a lo largo de la vida velando por mi bienestar. Es por ellos que soy lo que soy ahora. ¡Los amo!

A mis hermanas, María Teresa y Liceth Karine, por permitirme estar a su lado compartiendo espacios y momentos significativos, con su cariño, consejos, amor y ejemplo me enseñaron a luchar por lo que queremos. María T. Con su gran entusiasmo y alegría, tú me has traído momentos alegres en mi vida y con tu ejemplo de dedicación me has enseñado que todo con entrega y esfuerzo se pueden hacer realidad, Liceth mi hermanita menor, con todo su amor irradia seguridad, con sus consejos, apoyo y ánimo siempre da el mejor ejemplo a seguir.

A mi Tia Ilva Yaruro, Carlos Arturo, Carlos Andrés y Juan Felipe, por brindarme apoyo y soporte de familia que con cariño y consejos me dieron su ayuda motivándome para seguir adelante, gracias por permitirme ser un integrante más de su familia y por estar pendientes de mi.

A mi Tía Alcira Yaruro por brindarme sus consejos, apoyo y orientación en los momentos más difíciles, por su labor como segunda mamá siempre buscando que los que están a su alrededor estén bien. A mi Tia Elva Yaruro y abuelos, quienes han sido los angelitos que han iluminado mi camino. Aunque no estén acá, se que desde el cielo me han enviado fuerzas para verme obtener esta meta, siempre me quisieron ver como un gran profesional, gracias por sus bendiciones.

También agradezco a mis amigos más cercanos, con los cuales he adquirido muchas experiencias a lo largo de la vida y que siempre he contado desde que los conocí, Zoraida Amado, Fabio Antonio Hernández, Erika Liliana Gonzales, Andrea Amado, Carlos Adrian y a todos aquellos amigos que no menciono por lo extenso que sería la lista.

Y finalmente también agradezco a mis profesores que me han apoyado en esta vida universitaria y fueron parte esencial en mi formación como profesional.

YESID ERASMO MENESES YARURO

AGRADECIMIENTOS

Al Ingeniero Carlos Gerardo Hernández Capacho, director del proyecto, por su apoyo y dedicación, durante el desarrollo y ejecución de este proceso.

A la Universidad Pontificia Bolivariana, y a todos los profesores que me apoyaron y me brindaron sus valioso conocimientos durante mi carrera.

A todas y cada una de las personas que de una u otra forma estuvieron involucradas directa o indirectamente en el desarrollo del proyecto, brindándome incondicionalmente apoyo para su ejecución.

TABLA DE CONTENIDO

	Pàg.
INTRODUCCIÓN.....	16
OBJETIVOS.....	18
OBJETIVO GENERAL.....	18
OBJETIVOS ESPECÍFICOS.....	18
1. MARCO TEÓRICO.....	19
1.1. ESTADO DEL ARTE.....	19
1.2. dsPIC.....	20
1.2.1. dsPIC33F.....	21
1.2.2. Descripción del dispositivo dsPIC33FJ128GP802.....	21
1.2.3. Módulos del dsPIC33FJ128GP802.....	22
1.2.3.1. Temporizadores.....	23
1.2.3.2. Módulo SPI (Interfaz Serial Periférica).....	24
1.2.3.3. Convertidor Analógico a Digital (ADC).....	24
1.2.3.4. Convertidor Digital Analógico (DAC).....	24
1.3. HERRAMIENTAS DE DESARROLLO PARA dsPIC.....	25
1.3.1. Compilador C30 – Microchip Technologies.....	25
1.3.1.1. Herramienta de diseño de filtros dsPICFdLite.....	26
1.3.2. Compilador MikroC Pro for dsPIC – Mikroelektronika.....	30
1.3.2.1. Herramienta de Diseño Filter Designer Tool.....	31
1.4. COMPARACIÓN DE LOS COMPILADORES C30 DE MICROCHIP Y MIKROC DE MIKROELEKTRONIKA.....	36
2. FILTROS DIGITALES.....	38
2.1. FUNDAMENTO DE FILTROS DIGITALES.....	38
2.1.1. Etapas de diseño de un filtro digital.....	39
2.2. FILTROS FIR (RESPUESTA AL IMPULSO FINITO):.....	40
2.2.1. Respuesta a la frecuencia ideal.....	42
2.2.2. Diseño de filtros FIR.....	44
2.2.2.1. Método de ventanas.....	45
2.3. FILTROS IIR (RESPUESTA AL IMPULSO INFINITO):.....	55
2.3.1. Diseño de filtros IIR.....	56
2.3.1.1. Diseño de filtros IIR mediante transformada Bilineales.....	56
3. ENTRENADOR DE AUDIO.....	61

3.1. PRE-AMPLIFICADOR DE ENTRADA AUXILIAR.....	62
3.3. AMPLIFICADOR DIFERENCIAL	66
3.4. FILTRO PASA-BAJOS ANALÓGICO	67
3.5. CONVERTOR DIGITAL ANALÓGICO MCP4921.....	68
3.6. MICROCONTROLADOR dsPIC33FJ128GP802	69
3.6.1. Oscilador.....	69
3.6.2. Oscilador Secundario.....	69
3.6.3. Selección de entrada análoga.....	69
3.6.4. Condensador estabilizador del Voltaje de núcleo.	70
3.6.5. Circuito de Reset.	70
3.6.6. Indicadores o Testigos:	70
3.6.7. Conexión ICSP o interfaz para PICKIT2.	70
3.6.8. Puerto SPI.....	70
3.6.9. Puerto USART.	70
3.6.10. Salida PWM.	71
3.7. FUENTE DE ALIMENTACIÓN.....	71
4. MANUAL DE USUARIO DEL ENTRENADOR DE AUDIO dsPIC33FJ128GP802	73
4.1. RECONOCIMIENTO DEL ENTRENADOR DE AUDIO	74
4.1.1. Filtro pasa-bajos.	75
4.1.2. Reset.....	76
4.1.3. Interfaz PICKIT2 -ICSP-.....	76
4.1.4. Interfaz USB.....	76
4.1.5. Fuente de alimentacion.....	77
4.1.7. Memoria	78
4.1.8. Diferencial.	78
4.1.9. Selección de entradas y osciladores.....	79
4.1.10. Indicadores.	80
4.1.11. Entrada analógica.	80
4.1.12. Pre-amplificador y micrófono	81
4.1.13. Amplificador de salida.	81
4.2. COMPILADOR C30 DE MICROCHIP	82
4.2.1. Procedimiento para utilizar Mplab C30 – Microchip.	82

4.2.1. Asistente para creación de coeficientes para C30 – dsPICFD LITE.....	85
4.3. COMPILADOR MIKROC	91
4.3.1. Procedimiento para utilizar MikroC Pro for dsPIC –Mikroelektronika.....	91
4.3.1. Asistente para la creación de coeficientes para MikroC Filter Designer Tool. 94	
5. MANUAL DE PRÁCTICAS DEL ENTRENADOR DE AUDIO	
dsPIC33FJ128GP802.....	101
5.1. PRÁCTICAS DEL ENTRENADOR DE AUDIO dsPIC33FJ128GP802 UTILIZANDO EL COMPILADOR C30 DE MICROCHIP	101
5.1.1. Práctica No. 1 ADC/DAC LOOPBACK	101
5.1.2. Práctica No. 2 FIR Pasa-bajos.....	103
5.1.3. Práctica No. 3 FIR Final.....	106
5.1.4. Práctica No. 4 IIR Pasa-alto	108
5.1.5. Práctica No. 5 IIR Final.....	111
5.1.6. Práctica NO.6 AUDIO FIR	114
5.2. PRÁCTICAS DEL ENTRENADOR DE AUDIO dsPIC33FJ128GP802 UTILIZANDO EL COMPILADOR MIKROC PRO FOR DSPIC.....	118
5.2.1. Práctica No. 1 TIMER.....	119
5.2.2. Práctica N.2 MCP4921	120
5.2.3. PRÁCTICA No. 3 ADC/DAC LOOPBACK	122
5.2.4. Práctica No. 4 FIR Pasa-alto	124
5.2.5. Práctica No. 5 IIR Final.....	127

Lista Figuras

Figura 1. Software dsPICFD Lite	26
Figura 2. Ventana de selección dsPICFD Lite a. Diseño del Filtro, b. Tipo de Filtro	26
Figura 3. Ventana de Especificación del Filtro del software dsPICFD Lite	27
Figura 4. Tipo de Implementación del Filtro del software dsPICFD Lite, a.IIR b .FIR	27
Figura 5. Análisis del Filtro del software dsPICFD Lite	28
Figura 6. dsPICFD Lite a. Menú para Generar Código Coeficientes y ubicación en Memoria, b. localización del Archivo, c. Código Generado.....	29
Figura 7. Herramienta de Diseño Filter Designer Tool.....	31
Figura 8. Parámetros de configuración filtros FIR e IIR Filter Designer Tool.	32
Figura 9. Herramienta Filter Designer Tool.....	32
Figura 10. Tipo de Filtro del Filter Designer Tool.....	33
Figura 11. software Filter Designer Tool a. Tipo de Ventana Filtro FIR, b. Método del Filtro IIR.....	33
Figura 12. Software Filter Designer Tool a. Frecuencia Limite del Filtro FIR, b. Frecuencia limite, atenuación de banda de Paso Filtro IIR.....	34
Figura 13. Software Filter Designer Tool a. Tipo de Ventana y Pulso de Respuesta del Filtro FIR, b. Característica de la Ventana y Diseño del Filtro FIR, c. Función de Transferencia del Filtro Análogo y Digital IIR, d. Amplitud en Frecuencia del Filtro.....	35
Figura 14. Código Fuente del Programa para el diseño del filtro software Filter Designer Tool.....	36
Figura 15. Diagrama de Bloques Filtro Digital Simplificado	38
Figura 16. Especificaciones de un Filtro Digital	39
Figura 17. Estructura de Filtro FIR.....	41
Figura 18. Filtros Ideales	42
Figura 19. a. Ventana rectangular en el dominio del tiempo. b. ventana rectangular en el dominio de la frecuencia.	46
Figura 20 a. Ventana triangular en el dominio del tiempo. b. ventana triangular en el dominio de la frecuencia.	46
Figura 21 a. Ventana Hann en el dominio del tiempo. b. ventana Hann en el dominio de la frecuencia.	47
Figura 22 a. Ventana Hamming en el dominio del tiempo. b. ventana Hamming en el dominio de la frecuencia.	48

Figura 23 a. Ventana Blackman en el dominio del tiempo. b. ventana Blackman en el dominio de la frecuencia.	49
Figura 24. Respuesta al impulso del filtro ideal pasa bajas	52
Figura 25. Respuesta al impulso del filtro ideal pasa bajas.	53
Figura 26. Coeficientes Ventana Hamming Pasa Bajos	55
Figura 27. Estructura de Filtro IIR	56
Figura 28. Filtro digital IIR directo	60
Figura 29. Diagrama de Bloques Entrenador de Audio.....	61
Figura 30. Entrenador de Audio	62
Figura 31. Plano esquemático Pre-amplificador.	62
Figura 32. Comportamiento del pre-amplificador $F_{in}=500\text{Hz}$ $G\approx 2$	63
Figura 33. Comportamiento en frecuencia del pre-amplificador	64
Figura 34. Plano esquemático amplificador de salida.....	65
Figura 35. Comportamiento del Amplificador de Salida $F_{in}=500\text{Hz}$ $G\approx 5$	65
Figura 36. Comportamiento en frecuencia del Amplificador de Salida.....	66
Figura 37. Plano esquemático amplificador diferencial.....	66
Figura 38. Plano esquemático Filtro Pasa-bajos.....	67
Figura 39. Configuración de un filtro pasa-bajos Sallen Key Orden 2.....	67
Figura 40. Comportamiento en frecuencia del Filtro Pasa-bajos	68
Figura 41. Diagrama esquemático conversor Digital Analógico MCP4921	68
Figura 42. Diagrama esquemático del dsPIC33F en la tarjeta entrenadora	69
Figura 43. Fuente de alimentación del entrenador de audio para dsPIC	71
Figura 44 . Plano Esquemático Entrenador de Audio	72
Figura 45. Entrenador de audio dsPIC33FJ128GP802.....	74
Figura 46. Diagrama General del Entrenador de Audio	75
Figura 47. Diagrama General del Entrenador de Audio Filtro Pasa-Bajos.....	75
Figura 48. Diagrama General del Entrenador de Audio – Reset.....	76
Figura 49. Diagrama General del Entrenador de Audio – Interfaz PICKIT2-ICSP.	76
Figura 50. Diagrama General del Entrenador de Audio – Interfaz USB.....	77
Figura 51. Diagrama General del Entrenador de Audio – Fuente de alimentación.	77
Figura 52. Diagrama General del Entrenador de Audio – D/A MCP4921.	78
Figura 53. Diagrama General del Entrenador de Audio – Memoria.	78
Figura 54. Diagrama General del Entrenador de Audio – Diferencial.	79
Figura 55. Diagrama General del Entrenador de Audio – Selección de entradas y audio.	79
Figura 56. Diagrama General del Entrenador de Audio – Indicadores.....	80
Figura 57. Diagrama General del Entrenador de Audio – Entradas analógicas.....	80
Figura 58. Diagrama General del Entrenador de Audio – Pre-amplificador.	81
Figura 59. Diagrama General del Entrenador de Audio – Amplificador de salida.	81

Figura 60. Asistente para Crear un Proyecto Mplab Idle	82
Figura 61. Selección del dispositivo Mplab Idle.	82
Figura 62. Selección del lenguaje a utilizar Mplab Idle.	83
Figura 63. Asignación del nombre y ruta del proyecto Mplab Idle.....	83
Figura 64. Selección de ficheros del proyecto Mplab Idle	84
Figura 65. Selección Ventana finalización de creación de proyecto Mplab Idle.....	84
Figura 66. Software dsPICFD Lite	85
Figura 67. Ventana de Selección dsPICFD Lite filtro FIR. a. Diseño del Filtro, b. Tipo de Filtro.	86
Figura 68. Ventana de Especificación del Filtro dsPICFD Lite filtro FIR.	86
Figura 69. Tipo de Implementación del Filtro FIR dsPICFD Lite filtro FIR.	87
Figura 70. Análisis del Filtro dsPICFD Lite filtro FIR	87
Figura 71. Selección de la Ubicación de los coeficientes y Código Generado dsPICFD Lite filtro FIR.	88
Figura 72. Ventana de Selección dsPICFD Lite filtro IIR. a. Diseño del Filtro, b. Tipo de Filtro.	89
Figura 73. Ventana de Especificación del Filtro dsPICFD Lite filtro IIR.	89
Figura 74. Tipo de Implementación del Filtro IIR dsPICFD Lite filtro IIR.....	90
Figura 75. Análisis del Filtro dsPICFD Lite filtro IIR.	90
Figura 76. Ventana de aplicación nuevo proyecto Mikroelektronika.	91
Figura 77. Interfaz de selección del dispositivo MikroC Pro for dsPIC.....	92
Figura 78. Interfaz de selección del reloj MikroC Pro for dsPIC.....	92
Figura 79. Interfaz de asignación del nombre y ruta del proyecto MikroC Pro for dsPIC	93
Figura 80. Interfaz de selección de ficheros del proyecto MikroC Pro for dsPIC ...	93
Figura 81. Ventana de selección de librerías MikroC Pro for dsPIC	94
Figura 82. Ventana finalización de creación de proyecto.....	94
Figura 83. Herramienta de Diseño Filter Designer Tool.....	95
Figura 84. Parámetros de configuración filtros FIR e IIR Filter Designer Tool	95
Figura 85. Herramienta Filter Designer Tool.....	96
Figura 86. Tipo de Filtro Filter Designer Tool.....	97
Figura 87. Filter Designer Tool a. Tipo de Ventana Filtro FIR, b. Método del Filtro IIR.	97
Figura 88. Filter Designer Tool. a. Frecuencia Límite del Filtro FIR, b. Frecuencia límite, atenuación de banda de Paso del filtro IIR.....	98
Figura 89. Filter Designer Tool. a. Tipo de Ventana y Pulso de Respuesta del Filtro FIR, b. Característica de la Ventana y Diseño del Filtro FIR, c. Función de Transferencia del Filtro Análogo y Digital IIR, d. Amplitud en Frecuencia del Filtro	99

Figura 90. Código Fuente del Programa para el diseño del filtro. Filter Designer Tool.....	100
Figura 91. Módulo entrenador de Audio “ADC/DAC LOOPBACK” Mplab C30	102
Figura 92. Diagrama de Flujo “01-AD-DA LOOPBACK” Mplab C30	103
Figura 93. Entrenador de audio 02-FIR-LOWPASS Mplab C30	104
Figura 94. Diagrama de flujo “02-FIR-LOWPASS” Mplab C30	105
Figura 95. Entrenador de audio 03-FIR-FINAL Mplab C30.....	106
Figura 96. Diagrama de flujo “03-FIR-FINAL”	107
Figura 97. Entrenador de audio IIR 04-IIR-HIGHPASS Mplab C30	109
Figura 98. Diagrama de flujo “04-IIR-HIGHPASS” Mplab C30.....	110
Figura 99. Entrenador de audio 05-IIR-FINAL. Mplab C30	111
Figura 100. Diagrama de flujo “05-IIR-FINAL” Mplab C30	113
Figura 101. Entrenador de audio 06-AUDIO-FIR Mplab C30.....	115
Figura 102. Diagrama de flujo “06-AUDIO-FIR” Mplab C30.....	117
Figura 103. Entrenador de audio 01-TIMER1 PRB MikroC	119
Figura 104. Diagrama de flujo “01-TIMER1 PRB” MikroC	120
Figura 105. Entrenador de audio 02-MCP4921 PRB MikroC.....	121
Figura 106. Diagrama de flujo “02-MCP4921 PRB” MikroC.....	122
Figura 107. Entrenador de audio 03-ADC-DAC MikroC.....	123
Figura 108. Diagrama de flujo “03-ADC-DAC” MikroC.....	124
Figura 109. Entrenador de audio 04-FIR PALTO MikroC	125
Figura 110. Diagrama de flujo “04-FIR PALTO” MikroC	126
Figura 111. Entrenador de audio 05-IIR TODOS” MikroC	127
Figura 112 Diagrama de flujo “07-IIR TODOS” MikroC.....	128

Lista de Tablas

Tabla 1. Características Importantes de la Familia dsPIC33F	21
Tabla 2. Características Principales de dsPIC33FJ128GP802	23
Tabla 3. Características Principales de dsPIC33FJ128GP802	37
Tabla 4. Filtros Ideales Respuesta en frecuencia	43
Tabla 5. Secuencias Simétricas.....	44
Tabla 6. Aplicaciones a Tipos de Filtros	44
Tabla 7. Características Frecuenciales de Algunas funciones de Ventana.	49
Tabla 8. Coeficientes ventana Hamming Pasa-Bajo.....	54
Tabla 9. Respuestas de los distintos tipos de filtros analógicos	58

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: DISEÑO Y FABRICACIÓN DE UN MÓDULO DIDÁCTICO BASADO EN dsPIC COMO HERRAMIENTA DE APOYO EN EL APRENDIZAJE DE SISTEMAS DE CONTROL DIGITAL CON ÉNFASIS EN LA IMPLEMENTACIÓN DE FILTROS DIGITALES.

AUTOR: YESID ERASMO MENESES YARURO

FACULTAD: FACULTAD DE INGENIERÍA ELECTRÓNICA

DIRECTOR: CARLOS GERARDO HERNÁNDEZ CAPACHO

RESUMEN

En este proyecto se brinda una alternativa en soluciones de control digital mediante el uso de filtros digitales, ya que constituyen un método moderno, robusto, flexible, estable en el tiempo y son muy apropiados para el tratamiento y procesamiento de señales.

El módulo entrenador realiza implementación de filtros digitales respuesta al impulso finito FIR y respuesta al impulso infinita IIR (pasa-bajos, pasa-alta, pasa-banda), utilizando un microcontrolador de la familia dsPIC33F, el cual se encarga de hacer el respectivo procesamiento de señales por medio de un convertidor analógico a digital y digital a analógico. También cuenta con un Pre-amplificador de entrada cuya función principal es acoplar una señal de audio para ser procesada por el dsPIC, además de un amplificador de salida cuyo propósito es aumentar la potencia de la señal para ser escuchada por un parlante tipo desktop. Así mismo dispone de un sistema de configuración para establecer el tipo de entrada y salida que se vaya a procesar.

Para la implementación de filtros digitales se utilizaron dos compiladores en lenguaje C: el C30 de Microchip Technologies y el MikroC Pro for dsPIC de Mikroelektronika.

PALABRAS CLAVES: Filtros Digitales, Filtros FIR, Filtros IIR, Microcontrolador, dsPIC,

V° B° DIRECTOR DE TRABAJO DE GRADO

ABSTRACT OF THESIS PROJECT

TITLE: DESIGN AND MANUFACTURE OF AN EDUCATIONAL MODULE BASEDON A dsPIC AS SUPPORT TOOL IN THE LEARNING OF DIGITAL CONTROL SYSTEMS WITH EMPHASIS ON THE IMPLEMENTATION OF DIGITAL FILTERS.

AUTOR: YESID ERASMO MENESES YARURO

FACULTY: FACULTAD DE INGENIERÍA ELECTRÓNICA

DIRECTOR: CARLOS GERARDO HERNÁNDEZ CAPACHO

ABSTRACT

This project provides an alternative on digital control solutions using digital filters; which are modern, robust, flexible and stable over time methods and very suitable for the treatment and processing of signals.

The trainer module implement digital filters FIR finite impulse response and IIR infinite impulse resposnse (low-pass, high-pass, band-pass) using a microcontroller of the dsPIC33F family, which is responsible of doing the signal processing using an analog digital and digital analog converters. It also has a pre-amplifier input; its function is to couple the audio signal to be processed by the dsPIC, in addition an output amplifier which purpose is to increase the signal strength to be heard by a desktop type speaker. It also has a configuration system to set the type of input and output to be processed.

For the implementation of digital filters two C compilers were used, C30 from Microchip Technologies and MikroC Pro for dsPIC from Mikroelektronika.

KEYWORDS: Digital Filters, FIR Filters, IIR Filters, Microcontroller, dsPIC.

V° B° THESIS DIRECTOR

INTRODUCCIÓN

Los filtros digitales constituyen un método moderno para el tratamiento de señales. Dada su robustez, flexibilidad, estabilidad en el tiempo se consideran herramientas muy apropiadas para el procesamiento de señales de audio, video, biométricas, entre otras.

Hace unos años su implementación no era tan popular debido a la gran velocidad de procesamiento que demanda, por esta razón la mayoría de aplicaciones estaban basadas solo con DSPs.

Con la aparición de dispositivos híbridos entre microcontroladores y DSPs, aplicaciones como el control de motores, el procesamiento de señales y la conversión digital de potencia fueron más populares y accesibles gracias a la fácil adquisición por sus bajos costos.

Estos dispositivos desarrollados por la empresa Microchip en el año 2002 recibieron el nombre de DSC (Digital Signal Controller), comúnmente conocidos como dsPIC, los cuales son potentes microcontroladores de 16 bits con los principales recursos de un DSP.

La propuesta del trabajo de grado surge de la necesidad de proveer tanto a estudiantes como docentes de una herramienta hardware sobre la que puedan experimentar los diferentes conceptos teóricos aprendidos sobre el tema de procesamiento digital de señales, aplicando específicamente al tratamiento de una señal filtrada digitalmente. Para esto se plantea la elaboración de una tarjeta la que tiene como objetivo principal el filtrado digital de una señal de audio en tiempo real.

En este proyecto se diseñó una tarjeta electrónica cuyo dispositivo principal es el microcontrolador dsPIC33Fj128GP802 que por medio de sus módulos permiten comunicarse con el exterior y realizar diversas acciones de control. Para el funcionamiento de la tarjeta se diseñó un pre-amplificador de entrada para acoplar la señal de audio para ser procesada por el microcontrolador, un amplificador de salida que proporciona una buena potencia a los parlantes. Los principales módulos implementados del microcontrolador son ADC (Conversor Analógico Digital) permite convertir una señal analógica a una señal digital de salida para que ser procesada, el DAC (Conversor Digital Analógico) que permite convertir los datos digitales a una señal analógica que será enviada al amplificador de salida para ser escuchada por medio de los altavoces. Las señales de audio adquiridas por medio de un dispositivo MP3 o micrófono, son procesadas en tiempo real a través de filtros digitales FIR e IIR implementados mediante el software que configura y supervisa el desempeño del filtro programado previamente diseñado. Generando un aprendizaje dinámico, permitiendo al estudiante afianzar los conocimientos adquiridos en esta área mediante la

realización de prácticas que le ayudarán a visualizar de una manera más explícita la aplicación de estos conceptos en aplicaciones reales, mostrando como aplicación inicial el uso de filtros digitales empleados en señales de audio.

OBJETIVOS

OBJETIVO GENERAL

Diseñar y construir un módulo didáctico basado en dsPIC33FJXXXGPXXX, que permita soportar los fundamentos teóricos sobre filtros digitales en el área de control digital, mostrando su aplicación en señales de audio.

OBJETIVOS ESPECÍFICOS

- Diseñar un circuito electrónico acondicionador de señal de audio para acoplar la señal de entrada al conversor analógico digital del dsPIC.
- Diseñar un circuito electrónico que permita acondicionar los resultados de la conversión digital analógica para la reproducción de sonidos generados en parlantes tipo desktop.
- Documentar y comparar las herramientas para diseñar filtros digitales contenidos en el compilador C30 (Microchip) y MikroC (Mikroelektronika).
- Implementar los filtros FIR: paso-bajo, paso-alto y pasa-banda en un dsPIC utilizando las herramientas C30 y MikroC.
- Implementar los filtros IIR: paso-bajo, paso-alto y pasa-banda en un dsPIC utilizando las herramientas C30 y MikroC.
- Diseñar e implementar un dispositivo que permita aplicar filtros digitales a señales de audio adquiridas en tiempo real, además que sirva de soporte para el diseño de nuevos desarrollos que involucren el uso de un dsPIC.
- Elaborar un manual de uso de la herramienta donde se detallarán las prácticas propuestas en este proyecto que podrán ser usadas por estudiantes y docentes de la Facultad.

1. MARCO TEÓRICO

1.1. ESTADO DEL ARTE

Hoy en día, con el crecimiento industrial y el aumento de complejidad en procesos de control y automatización, han implementado una serie de adelantos tecnológicos teniendo como objetivo indagar una serie de soluciones que permitan dar como resultado una solución rápida y eficiente.

Gracias a los avances en los últimos tiempos de la microelectrónica han evolucionado los microcontroladores. Por ello, el avance de la tecnología ha llevado al estudio de desarrollos de tarjetas basadas en microcontroladores dsPIC, de tal forma de establecer diversas estrategias para el desarrollo de aplicaciones.

A continuación se dan a conocer una serie de trabajos, publicaciones, proyectos y tesis, que implican desarrollos de módulos didácticos con microcontroladores dsPIC.

En Venezuela, el Instituto Universitario de tecnología de Valencia, realizó un control didáctico de un motor, utilizando el dsPIC33F12MC202, llamado “Control basado en lógica borrosa en dsPIC33 para un motor Brushless”, por José Daniel Florez Rodríguez; logrando ilustrar el diseño de los controladores borrosos de manera sencilla y eficaz en dsPIC los cuales, pueden trabajar con operaciones de punto flotante logrando un alto desempeño en la aplicación de estos sistemas.¹

En Argentina, en la Universidad Nacional de Mar del Plata, se encontró un estudio llamado “Sistema de compresión de voz portátil basado en un dsPIC”, por Francisco Denk, Pablo Daniel Agüero, Alejandro José Uriz, Juan Carlos Tulli, y Esteban Lucio González, donde se desarrolló un dispositivo portátil que comprime la voz en tiempo real, con el cual se mejora la calidad de vida de las personas con impedimento auditivo ya que es similar a los que se encuentran en el mercado pero con un costo más bajo.²

En México, en el Instituto Politécnico Nacional la Escuela Superior de Ingeniería Mecánica y Eléctrica, unidad Culhuacán, se realizó una tesis llamada “Diseño de un sistema de monitoreo electrónico de parámetros eléctricos de una línea monofásica usando el dsPIC33FL256GP701-I” por Felipe Cortés Ávila, donde se

¹ FLOREZ RODRIGUEZ, José Daniel. Control basado en lógica borrosa en dsPIC33 para un motor Brushless. Republica Bolivariana de Venezuela: Instituto Universitario de tecnología de Valencia.

² DENK Francisco, AGÜERO Pablo daniel, URIZ Alejandro José, TULLI Juan Carlos, GONZALEZ Esteban Lucio. Sistema de compresión de voz portátil basado en un dsPIC. Argentina: Universidad Nacional de Mar del Plata

obtuvo como resultado el monitoreo del factor de potencia, corriente, voltaje y armónicos de una línea de alimentación monofásica residencial.³

En Estados Unidos, en la Universidad de Iowa, se realizó un estudio llamado “Data acquisition unit for an implantable multi-channel optical glucose sensor” por Kiran Kanukurthya, Mathew B. Covera y David R. Andersenb, donde se desarrolló un dispositivo de implantación subcutánea, que por medio de un sensor detecta los niveles de glucosa y los envía al dsPIC, donde son convertidos y enviados al usuario final para poder controlar la inyección de insulina a un paciente.⁴

En Ecuador, en la Escuela Politécnica del Ejercito, se realizó un estudio llamado “Diseño e implementación de un prototipo de un afinador digital de instrumentos musicales con dsPIC30F3014” por Helena Dolores Ibarra Ruiz, donde se desarrolló el cálculo de la frecuencia de una señal de audio ingresada al dsPIC, este saca el espectro de frecuencias por medio de la FFT y luego se determina el valor con la mayor energía, el mismo que será comparado con una tabla de valores de frecuencia de cuerdas conocidas de acuerdo a la Escala Temperada.⁵

1.2. dsPIC

Microchip Technology, el mayor fabricante actual de microcontroladores en el mundo, ha iniciado su incursión en el mercado de dispositivos para el procesamiento digital de señales, supliendo la necesidad de los consumidores que necesitan trabajar con funciones de procesamiento digital a mediana escala y bajo costo; ha desarrollado un híbrido entre un DSP y un microcontrolador de 16 bits, a este nuevo producto lo han llamado DSC por sus siglas (Digital Signal Controller) ®, y comúnmente se conocen como dsPIC.

Los dsPIC, son el mejoramiento de los DSP. Su diseño ha sido desarrollado por expertos especialistas de diferentes áreas.

Este microcontrolador ofrece velocidad, potencia, manejo flexible de interrupciones, un amplio campo de funciones periféricas analógicas y digitales, opciones de reloj, protección Brown out, seguridad del código, simulación en tiempo real, etc.

³ CORTES AVILA Felipe. Diseño de un sistema de monitoreo electrónico de parámetros eléctricos de una línea monofásica usando el dsPIC33FL256GP701-I. México.:Instituto Politécnico Nacional, escuela Superior de Ingeniería Mecánica y Eléctrica, unidad Culhuacán.

⁴ KANUKARTHYA kiran, COVERA Mathew B., ANDERSEN David R. Data acquisition unit for an implantable multi-channel optical glucose sensor. Estados Unidos: Universidad de Iowa.

⁵ Ibarra Ruiz, Helena Dolores . Diseño e implementación de un prototipo de un afinador digital de instrumentos musicales con DSPIC30F3014. Facultad de Ingeniería Electrónica. ESPE. Sede Sangolquí 2007

El dsPIC posee un núcleo RISC con arquitectura Harvard modificada, con un juego optimizado para el lenguaje C, un bus de datos de 16 bits, bus de instrucciones de 24 bits, banco de registros de propósito general de 16 bits, multiplicación para enteros y fraccionales.

1.2.1. dsPIC33F. A continuación se darán a conocer las principales características del dsPIC33F:

El dsPIC cuenta con una arquitectura de 16 bits, de velocidad 40 MIPS, la capacidad de memoria es tipo flash 256kb, mantiene los periféricos, SRAM de hasta 30kb (Tabla 1). Algunos de ellos poseen reloj y calendario en tiempo real. No disponen de memoria EEPROM. El rango de temperatura es de – 40 a 125°C.⁶

Tabla 1. Características Importantes de la Familia dsPIC33F

RECURSO	RANGO DE VALORES
Memoria de Programa FLASH	Hasta 256 KB
Memoria de Datos RAM	Hasta 30 KB
Memoria de Datos EEPROM	No Dispone
Temporizadores de 16 bits	Hasta 9
Modulo de Captura	Hasta 8 Entradas
Módulo Comparador / PWM	Hasta 8 Salidas
Convertor A/D de 10 bits	2,2 Mbps, hasta 32 canales
Convertor A/D de 12 bits	1Mbps, hasta 32 canales
UART	Hasta 2 módulos
SPI™ (8-16 bits)	Hasta 2 módulos
I ² C™	Hasta 2 módulos
QEI	1
Interfaz CODEC	1
CAN	Hasta 2 módulos

Fuente: Datasheet Familia dsPIC33 Microchip

1.2.2. Descripción del dispositivo dsPIC33FJ128GP802. A continuación se describen las características del microcontrolador que será el propósito de esta tesis. Este microcontrolador es de la familia de propósito general, el cual fue seleccionado ya que posee un convertor digital analógico especialmente para la aplicación de audio.

⁶ <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en532298>

Un dsPIC33FJ128GP802 cuenta con un rango de funcionamiento de hasta 40 MIPS (a 3.0-3.6V), un rango de temperatura industrial (-40 ° C a +85 ° C) y un rango de temperatura extendida (-40 ° C a +125 ° C) de alto rendimiento DSC CPU. Cuenta con una arquitectura Harvard modificada, un juego de instrucciones optimizado para el lenguaje C, Bus de datos de 16 bits y un Bus de instrucciones de 24 bits. Repertorio de instrucciones hasta 4M. Modo de direccionamiento de memoria hasta 64 Kbytes. Repertorio de 83 instrucciones en su mayoría una palabra / 1 ciclo. Dos acumuladores de 40 bits con redondeo y saturación. Modos de direccionamiento indirecto: Modular- Bit-invertida. Con manejo de pila con software. Multiplicador para fraccionarios de 16 x 16, multiplicador de 32/16 y 16/16. Operación de multiplicación y acumulación de ciclo.

Memoria de programa de flash, memoria de datos SRAM, de arranque seguro y seguridad general para el programa de acceso directo a la memoria flash (DMA). Módulo de entrada de 8 canales DMA hardware. Hasta 2 kbytes de doble zona de amortiguamiento portado DMA para almacenar los datos transferidos a través de DMA. Permite transferencia de datos entre la RAM y un periférico, mientras la CPU ejecuta el código. Modo PWM de 16 bits, hasta 5 temporizadores de 16 bits pero se puede enlazar hasta hacer 2 temporizadores de 32 bits y pudiendo trabajar en tiempo real con oscilador externo de 32,768KHZ. Módulo de entrada de 4 canales y 4 de salida o doble de 16 bits. Flexiones opciones de reloj en tiempo real, calendario, alarma. Funciones de controlador de interrupción de 5 ciclos de latencia. 118 vectores de interrupción. Hasta 49 fuentes de interrupción disponibles. Hasta 3 interrupciones externas. 7 niveles de prioridad, programables. 5 excepciones especiales de E/S Digital. Hasta 35 puntos digitales programables / O. Wake-up/Interrupt-on-Change para un máximo de 21 pines. Los pines de salida pueden manejar desde 3.0V a 3.6V. hasta 5 V de salida con la configuración de salida abierta. Todos los pines de entrada digital son tolerantes a 5V. 4 mA en todos los pines I / O de administración del sistema (Ver Tabla 2)⁷.

1.2.3. Módulos del dsPIC33FJ128GP802. A continuación se darán a conocer los principales módulos utilizados en este proyecto. La descripción es muy general, para mayor información de configuración y más módulos, referirse al manual del dsPIC33FJ128GP802.

⁷ <http://ww1.microchip.com/downloads/en/DeviceDoc/70292E.pdf>

Tabla 2. Características Principales de dsPIC33FJ128GP802

Nombre del parámetro	Valor
Arquitectura	16-bit
Velocidad de la CPU (MIPS)	40
Tipo de memoria	Flash
Memoria de programa (KB)	128
Bytes de RAM	16.384
Rango de temperatura C	-40 A 125
De tensión de funcionamiento (V)	3 a 3,6
Pines I / O	21
Pines totales	28
Características del sistema de gestión	PBOR
Oscilador interno	7,37 MHz, 512 kHz
Comunicación Digital Periféricos	2-UART, SPI-2, 1 de I2C-
Periféricos Análogos	1-A / D 10x12 bits @ 500 (KSPS) 1-D / A de 2x16 bits @ 100 (KSPS)
Comparadores	2
CAN (#, tipo)	1 ECAN
Captura / Comparación / PWM Periféricos	4 / 4
De 16-bit de resolución PWM	16
Temporizadores	5 x 16-bit 2 x de 32-bit
Puerto paralelo	PMP
Hardware RTCC	Sí
DMA	8

Fuente: datasheet Familia dsPIC33 Microchip

1.2.3.1. Temporizadores. La familia de los dispositivos dsPIC33F ofrece una serie de módulos de temporizadores, que trabajan a 16 bits, el número de Timers depende del dispositivo. Cada módulo tiene asociado una serie de registros de 16 bits que son: Registro de conteo (TMRx) guarda el valor que establece el tiempo a controlar, Registro que almacena el periodo asociado al temporizador (PRx) y Registro de control (TxCON).

Los temporizadores se clasifican en tres tipos de acuerdo a sus diferentes funcionalidades. El temporizador de tipo A (Timer1) puede ser accionado cuando el dispositivo trabaja con el oscilador desde el bajo consumo de 32 kHz, trabaja en modo asíncrono cuando el reloj proviene de una fuente externa. El temporizador de tipo B (Timer2, Timer4) puede ser conectado con un temporizador de de Tipo C para formar un contador de 32 bits. El temporizador de tipo C

(Timer3, Timer5) tiene la capacidad de desencadenar una conversión de análogo a digital (A / D), lo que indica que cuando ocurre un desbordamiento por Timer, finaliza el proceso de muestreo y empieza la conversión, de igual forma se pueden conectar con un temporizador de tipo B para formar un contador de 32 bits.

1.2.3.2. Módulo SPI (Interfaz Serial Periférica). Es un módulo de interfaz en serie síncrono, útil para comunicarse con otros dispositivos periféricos o con un microcontrolador. Los dispositivos periféricos pueden ser memorias EEPROM, registros de desplazamiento, controladores de video, convertidores A/D, etc. Dependiendo de la variante, la familia dsPIC33F ofrece uno o dos módulos de SPI en un solo dispositivo. Los módulos, designados SPI1 y SPI2, son funcionalmente idénticos.

El módulo SPI utiliza una serie de modos de operación flexible como son: transmisión y recepción de datos a 8 bits y 16 bits, modo maestro y esclavo, modo marco (Framed), operación únicamente de recepción y tratamiento del error.

1.2.3.3. Convertidor Analógico a Digital (ADC). El módulo Conversor Analógico Digital, se puede configurar para que trabaje con 10 bits o 12 bits. Este conversor A/D puede tener hasta 13 canales de entrada. El número real de pines de entrada analógica y externo de referencia del voltaje de entrada de configuración depende del dispositivo específico.

Las principales características que tiene para configuración de 10 bits son: velocidades de conversión hasta 1,1 Msps. hasta 13 pines de entrada analógica, muestreo simultáneo de la entrada analógica hasta cuatro pines, modo de búsqueda de canales automática, conversión de fuente de disparo seleccionable, funcionamiento de la CPU durante el modo sleep/espera y cuatro opciones de resultado de la alineación (fraccionaria/entero, sin signo/con signo). La configuración de ADC de 12 bits son compatibles con todas las anteriores excepto con la velocidad de conversión que son compatibles hasta 500KSPS y el muestreo simultáneo en canales múltiples no es compatible.

1.2.3.4. Convertidor Digital Analógico (DAC). El módulo Convertidor Digital Analógico (DAC) es un convertidor de señal Delta-Sigma de 16 bits diseñado para aplicaciones de audio. Dos canales de salida soportan las operaciones estéreo. La entrada de datos tiene forma de un valor digital de 16 bits desde el programa de aplicación a través del módulo DMA o de los registros de datos y control del DAC. El dato de salida es un voltaje análogo, que es proporcional al valor digital de entrada.

El módulo DAC está diseñado específicamente para aplicaciones de Audio, que provee estas características claves: Resolución de 16 bits (14 bits de precisión), Modulador digital Delta-Sigma de segundo orden, Radio de sobre muestreo de 256x, 100 ksps tasa máxima de muestreo, Reloj de Muestreo controlable por el

usuario, Frecuencia máxima de entrada de 45 kHz, Salidas análogas diferenciales, Búfer de entrada de 4 palabras de profundidad, Procesador de 16 bits de Entrada/Salida e interfaces DMA.

1.3. HERRAMIENTAS DE DESARROLLO PARA dsPIC

Debido al gran volumen de ventas que maneja Microchip, muchas empresas desarrolladoras ven interés en ofrecer herramientas software para programar los dispositivos, es el caso de Custom Computer Service (CCS), que desarrolló uno de los compiladores más populares para PIC y dsPIC basado en lenguaje C (PCW); la empresa recientemente adquirida por Microchip, HITECH, desarrolló un robusto compilador (Hitech PICC) llamando la atención de la casa matriz Microchip que no descansó hasta lograr obtener los derechos de este software. Mikroelektronika con su amplia gama de compiladores para PIC y dsPIC basado en lenguajes Basic (mikrobasic), Pascal (mikropascal) y C (mikroc), ofrece alternativas didácticas para hacer herramientas indicadas para estos dispositivos.

Microchip también ofrece herramientas de desarrollo para sus productos, integradas en un solo IDE llamado MPLAB IDE con soporte del lenguaje ensamblador para cada una de las familias y opción de compiladores en C para las familias: PIC18 (C18), PIC24 (C30), dsPIC30 (C30), dsPIC33 (C30) y PIC32 (C32).

En el presente trabajo se hará énfasis en 2 herramientas de desarrollo para dsPIC basadas en ANSI C, las cuales se ilustraran con detalle a continuación:

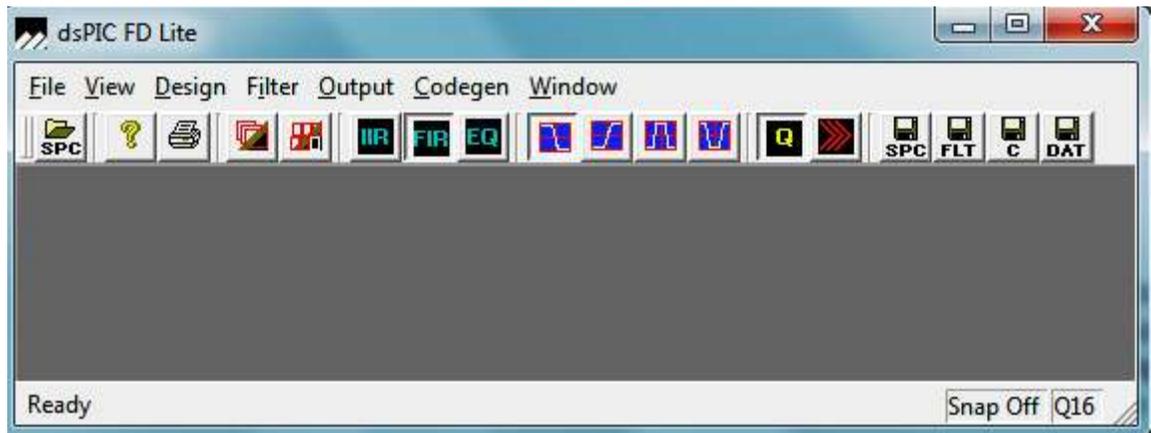
1.3.1. Compilador C30 – Microchip Technologies. Este compilador está basado en el estándar ANSI C, y soporta las familias de PIC de 16 bits PIC24 y de DSC dsPIC30 y dsPIC33. Es compatible con la interfaz de usuario MPLAB IDE.

Las características principales del C30 de acuerdo con microchip son:

- Compatible con estándar ANSI C, conversiones de datos y librerías matemáticas.
- Genera módulos objeto para optimizar el uso del código.
- Optimizado para generar mínimo un 30% menos de código que otros compiladores para dispositivos de 16 bits.
- Soporte robusto que permite el uso de código ensamblador dentro del código C.
- Soporte de librerías para los periféricos.
- Permite que tanto código como datos sean ubicados en cualquier lugar de la memoria.
- Soporta niveles avanzados de optimización de código.
- Soporta el trabajo con registros acumuladores que tienen los dsPIC.
- Soporta funciones intrínsecas de los DSP.

1.3.1.1. Herramienta de diseño de filtros dsPICFDLite. Este programa es de gran ayuda a la hora de análisis, aplicaciones y diseño de filtros FIR e IIR, mediante la utilización de interfaz grafica de usuario. Esta herramienta mediante cálculos matemáticos para determinar los coeficientes del filtro deseado, por otro lado permite visualizar la respuesta en frecuencia, en donde se puede observar gráficamente magnitud, respuesta al impulso, respuesta al escalón, polos y ceros, fase y retardo de grupo (Figura 1).

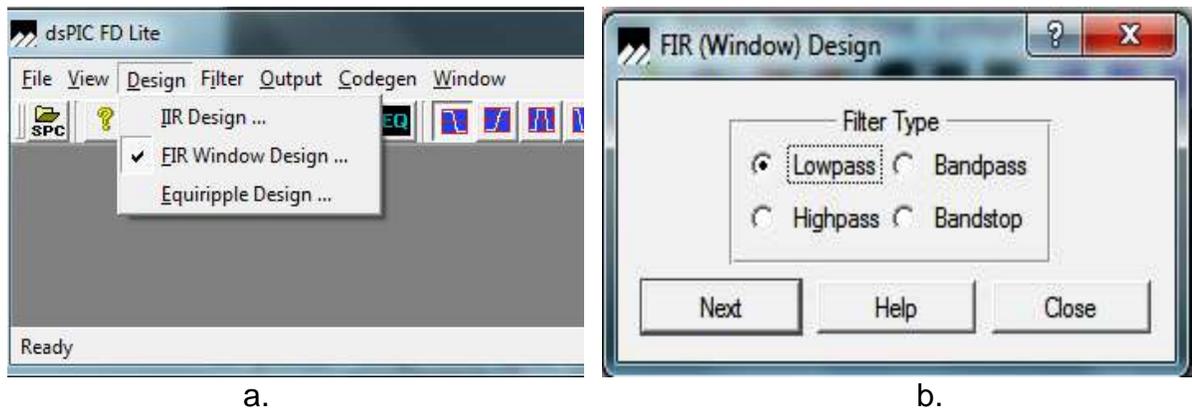
Figura 1. Software dsPICFD Lite



Fuente: Imagen tomada por el autor del texto.

Este software dispone de un asistente para el usuario en la construcción de un filtro FIR e IIR, siguiendo los pasos de la figura 2 y eligiendo el tipo de filtro que se requiera utilizar (pasa baja, pasa alta, pasa banda y rechaza banda).

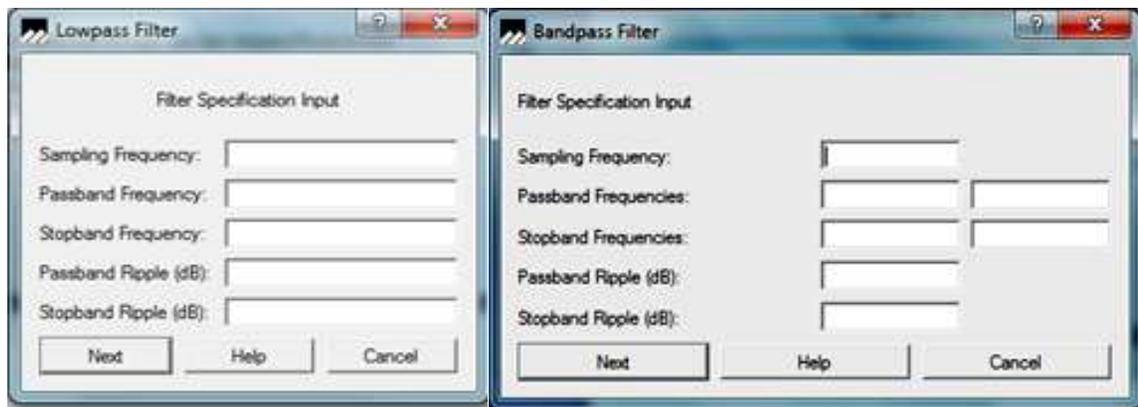
Figura 2. Ventana de selección dsPICFD Lite a. Diseño del Filtro, b. Tipo de Filtro



Fuente: Imagen tomada por el autor del texto

En la figura 3 se observa las especificaciones de las características de los filtros, en donde se pueden configurar la frecuencia de muestreo, frecuencias de pasa banda, frecuencias rechaza banda, atenuación pasa banda y atenuación rechaza banda. Esto es aplicable para los tres tipos de filtros IIR, FIR e EQUIRIPLE, pasa-bajos, pasa-altos, pasa-banda y rechaza-banda.

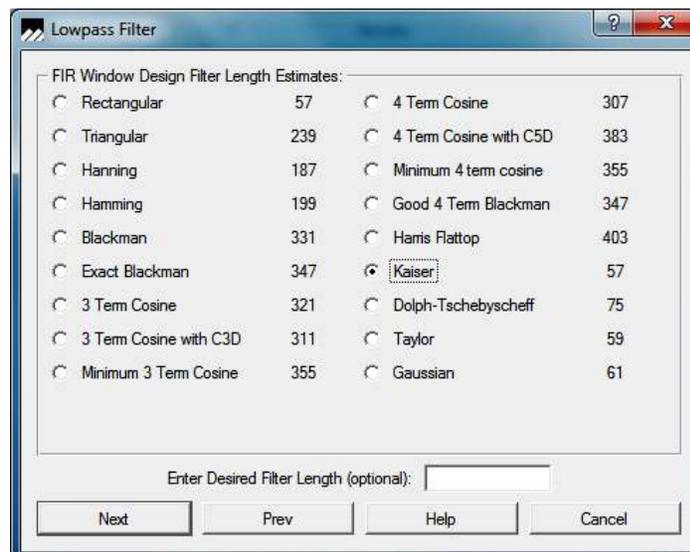
Figura 3. Ventana de Especificación del Filtro del software dsPICFD Lite



Fuente: Imagen tomada por el autor del texto

Al finalizar las especificaciones, se dispone a seleccionar el tipo de implementación indicando cual será el orden óptimo para el filtro (Figura 4).

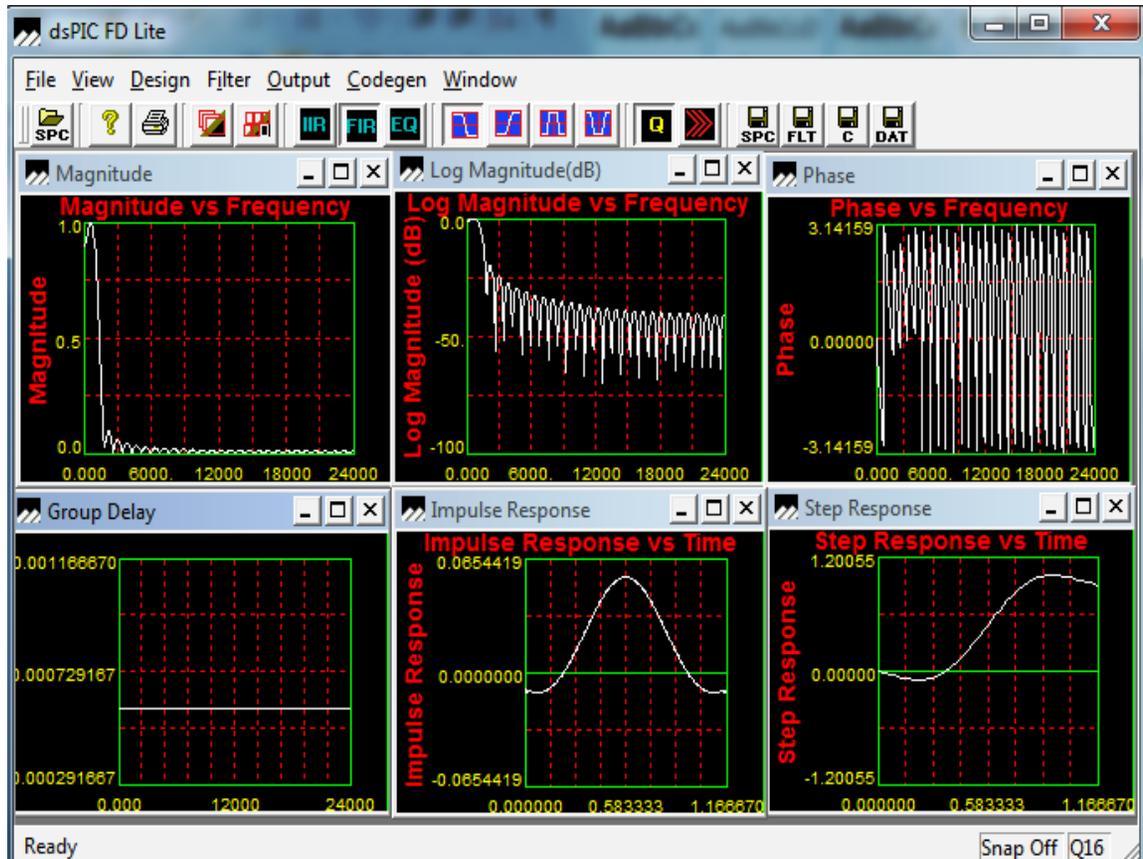
Figura 4. Tipo de Implementación del Filtro del software dsPICFD Lite, a.IIR b .FIR



Fuente: Imagen tomada por el autor del texto

Una vez seleccionado el tipo de filtro, se finaliza el diseño en donde se puede observar gráficamente magnitud, respuesta al impulso, respuesta al escalón, polos y ceros, fase y retardo de grupo, características muy importantes a la hora de implementar filtros (Figura 5).

Figura 5. Análisis del Filtro del software dsPICFD Lite



Fuente: Imagen tomada por el autor del texto del software dsPICFD Lite.

A continuación se observaran de manera mas detalladas las diferentes ventanas de la figura 5

Respuesta del módulo en escala lineal: visualiza la respuesta del módulo en escala lineal con respecto a la frecuencia introducida al sistema.

Respuesta del módulo en dB: visualiza la información del modulo en escala logarítmica.

Respuesta al impulso: visualiza la salida del sistema aplicando una entrada un impulso.

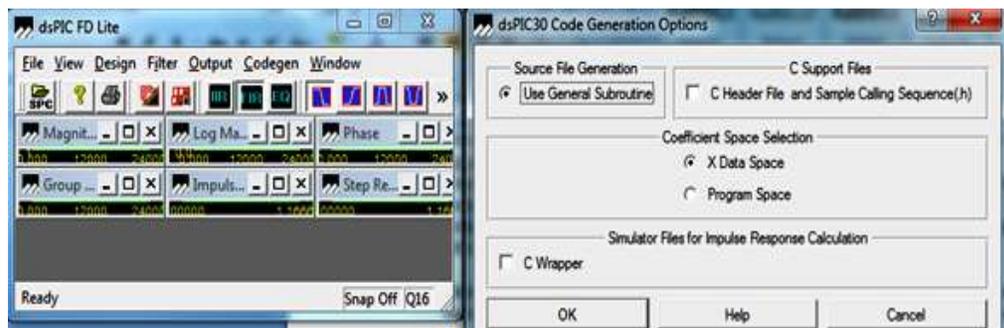
Respuesta al escalón: visualiza la salida del sistema aplicando una entrada escalon.

Fase: visualiza el desfase de la señal con respecto a la frecuencia de entrada, deduciendo la linealidad del sistema.

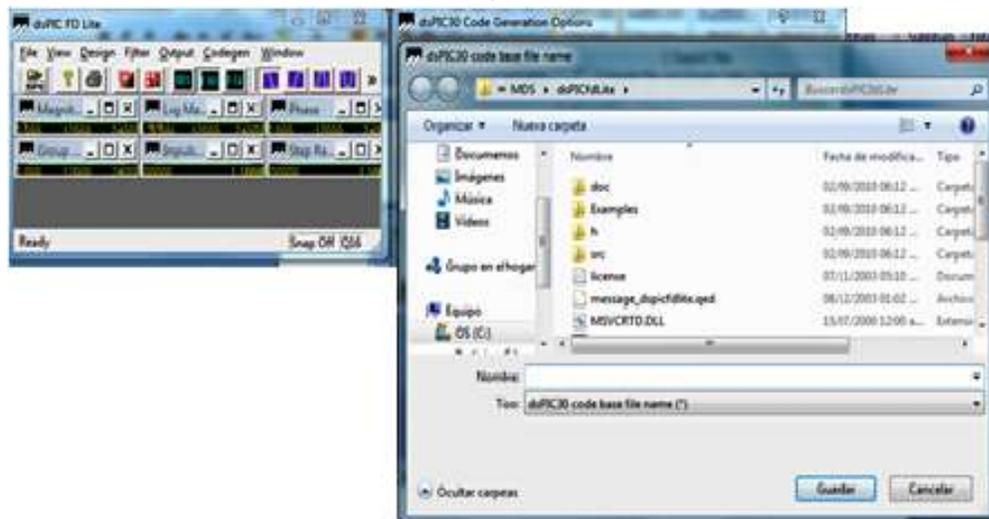
Retardo de grupo: visualiza el retardo de grupo dentro de la banda de paso del filtro.

Una vez determinado que el filtro diseñado es el adecuado se procede a generar los coeficientes del filtro para importarlos al programa principal. Para generar este archivo se debe ir a la barra de herramientas del menú Codegen y seleccionar Microchip. Se debe ubicar la sección en donde quiere ubicar los coeficientes en la memoria y así generar el archivo con dicha estructura de configuración (Figura 6).

Figura 6. dsPICFD Lite a. Menú para Generar Código Coeficientes y ubicación en Memoria, b. localización del Archivo, c. Código Generado.



a.



b.

```

1  /
2  /-----
3  / File pasabanda.s
4  /-----
5  /
6  /
7  /-----
8  / Allocate and initialize filter taps
9  /-----
10 /
11 /
12 /
13 pasabandaTaps:
14 .hword 0x01D2, 0x016F, 0x00FA, 0x0075, 0xFFE4, 0xFF4B, 0xFEAE, 0xFE14, 0xFD81
15 .hword 0xFCF8, 0xFC88, 0xFC2A, 0xFB88, 0xFB33, 0xFB8F, 0xFBDC, 0xFC1A, 0xFC79
16 .hword 0xFCF6, 0xFD8E, 0xFE3C, 0xFEFC, 0xFFC9, 0x0099, 0x0169, 0x0231, 0x02EA
17 .hword 0x038F, 0x041A, 0x0487, 0x04D2, 0x04F7, 0x04F7, 0x04D2, 0x0487, 0x041A
18 .hword 0x038F, 0x02EA, 0x0231, 0x0169, 0x0099, 0xFFC9, 0xFEFC, 0xFE3C, 0xFD8E
19 .hword 0xFCF6, 0xFC79, 0xFC1A, 0xFBDC, 0xFB8F, 0xFB33, 0xFB88, 0xFC2A, 0xFC88
20 .hword 0xFCF8, 0xFD81, 0xFE14, 0xFEAE, 0xFF4B, 0xFFE4, 0x0075, 0x00FA, 0x016F
21 .hword 0x01D2
22 /
23 /-----
24 / Allocate delay line in (uninitialized) Y data space
25 /-----
26 /
27 /
28 /
29 /
30 /
31 /

```

C.

Fuente: Imagen tomada por el autor del texto

1.3.2. Compilador MikroC Pro for dsPIC – Mikroelektronika. Este compilador tiene su principal ventaja en la facilidad de uso y las herramientas didácticas, haciendo muy práctico para personas que inician su proceso en el mundo de los dsPIC. Maneja funciones de alto nivel que permiten que se genere un código funcional con muy pocas líneas.

Posee una gran variedad de librerías elaboradas para los periféricos, procesamiento digital, estándar C, y misceláneas. Se enumeraran las que tienen que ver con el procesamiento de señales siguiendo con el objeto de este proyecto:

- Librería filtros FIR.
- Librería filtros IIR.
- Librería FFT.
- Librería para Vectores.
- Librería para Matrices.

La interfaz de usuario es muy intuitiva y concentra todas las herramientas y opciones que este compilador posee.

Los listados estadísticos que genera el IDE son los siguientes:

- Indicador gráfico de uso de RAM.
- Listado de locaciones de RAM utilizadas.
- Listado de registros especiales.
- Indicador gráfico de uso de ROM.
- Listado de constantes almacenadas en ROM.

- Listado de las funciones ordenado por tamaño.
- Listado de funciones ordenado por nombre.
- Listado de funciones ordenado por dirección en ROM.
- Árbol de funciones
- Resumen de memoria

Las herramientas incluidas son:

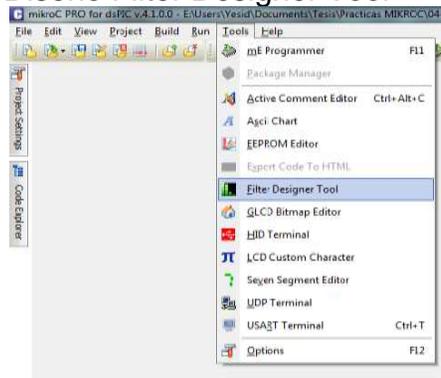
- Tabla ASCII.
- Terminal USART.
- Editor de memoria EEPROM.
- Terminal Protocolo UDP.
- Generador de imágenes para LCD gráfico.
- Decodificador para display de 7 segmentos.
- Asistente para crear caracteres para pantallas LCD alfanuméricas.
- Herramienta para el diseño de filtros.

1.3.2.1. Herramienta de Diseño Filter Designer Tool. Es una herramienta sencilla a la hora del diseño de filtros digitales ya sean FIR (Respuesta al Impulso Finito) e IIR (Respuesta al Impulso Infinito).

Para el diseño de un filtro FIR e IIR se inicia mediante la herramienta Filter Designer Tool, para esto se debe ir al menú principal del compilador, en tools se halla una serie de utilidades en donde se encuentra el acceso para iniciar la aplicación (Figura 7).

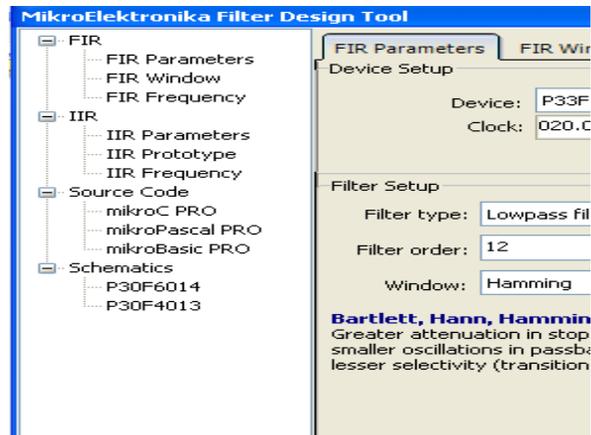
Una vez ejecutada esta herramienta, en la parte izquierda del asistente se localizan los filtros FIR e IIR con sus respectivas propiedades de configuración. Así mismo se encuentra el código fuente en algunos lenguajes de programación y el esquemático para dos dsPIC específicos de la familia 30F (Figura 8).

Figura 7. Herramienta de Diseño Filter Designer Tool



Fuente: Imagen tomada por el autor del texto.

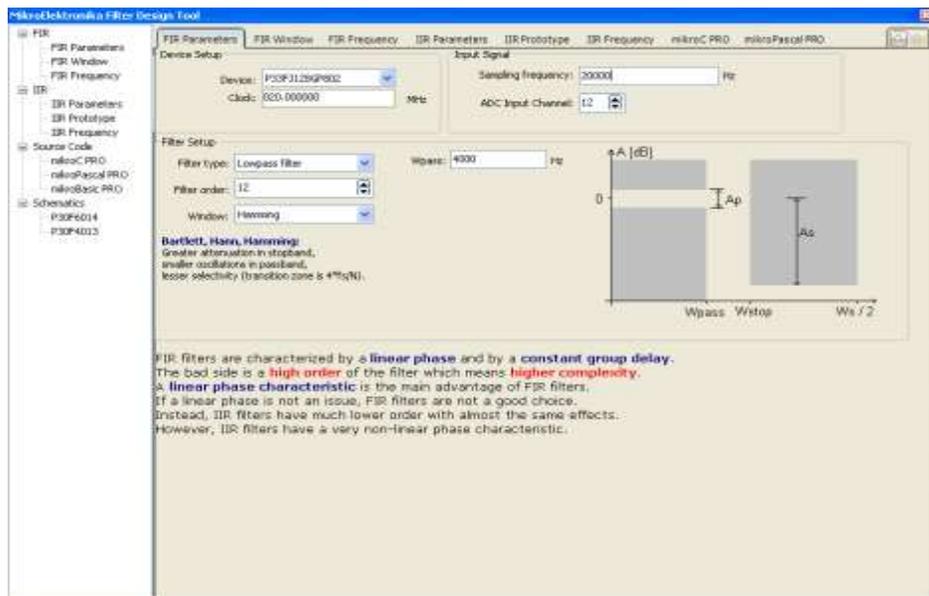
Figura 8. Parámetros de configuración filtros FIR e IIR Filter Designer Tool.



Fuente: Imagen tomada por el autor del texto.

En los parámetros para los filtros FIR e IIR se encuentran la configuración de dispositivos, como son el reloj y la designación del microcontrolador a utilizar. En las señales de entrada, localizamos dos parámetros, la frecuencia de muestreo y el canal de entrada del convertidor A/D. Cabe resaltar que la frecuencia de muestreo debe cumplir con el teorema de Nyquist, es decir se debe muestrear por lo menos al doble de la frecuencia máxima de la señal de entrada (Figura 9).

Figura 9. Herramienta Filter Designer Tool

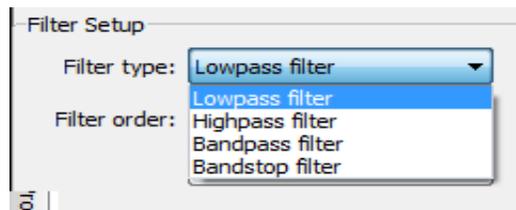


Fuente: Imagen tomada por el autor del texto del software **Filter Designer Tool**

Dentro de las opciones de tipo de filtros para ambas clases se encuentran las siguientes (Figura 10):

- Filtro de paso bajo,
- Filtro de paso alto,
- Filtro de paso de banda,
- Rechaza Banda.

Figura 10. Tipo de Filtro del Filter Designer Tool

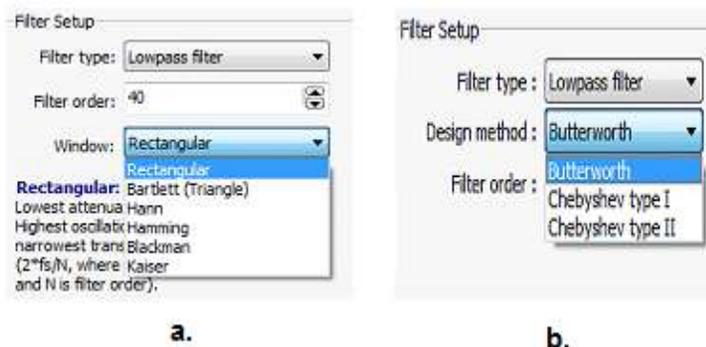


Fuente: Imagen tomada por el autor del texto

A continuación se procede a seleccionar el orden del filtro. En los FIR con método ventana Kaiser no se aplica esta opción porque se ajusta automáticamente de acuerdo a los parámetros especificados en el filtro. A medida que aumenta el orden, será mayor su selectividad esto tiene como consecuencia un incremento en la complejidad del filtro es decir se necesitará más memoria para la recolección y procesamiento de las muestras.

Después de seleccionar el tipo de filtro y el orden, se escoge el método a utilizar para obtener el código fuente y sus respectivas gráficas. Para el diseño de Filtros FIR se utiliza el método de ventanas, en donde se encuentran las siguientes opciones: ventana Rectangular, Bartlett, Hann, Hamming, Blackman y Kaiser. En el caso de Filtros IIR los métodos básicos de diseño son: Butterworth, Chebyshev I, Chebyshev II (Figura 11).

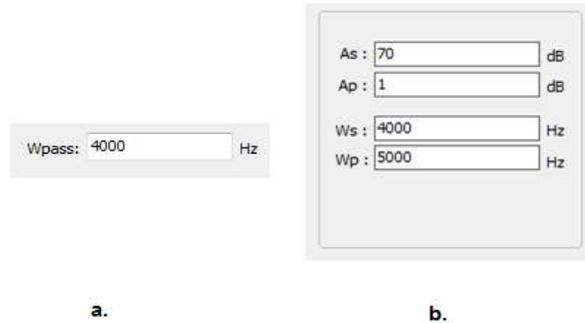
Figura 11. software Filter Designer Tool a. Tipo de Ventana Filtro FIR, b. Método del Filtro IIR



Fuente: Imagen tomada por el autor del texto del software **Filter Designer Tool**

Una vez seleccionado el tipo de filtro y dependiendo el método escogido se introducen los parámetros correspondientes, en el caso de los FIR se deben ingresar la frecuencia límite del filtro. En los IIR según método elegido se introducen parámetros tales como atenuación de banda de paso, límite de frecuencia de paso entre otros (Figura 12).

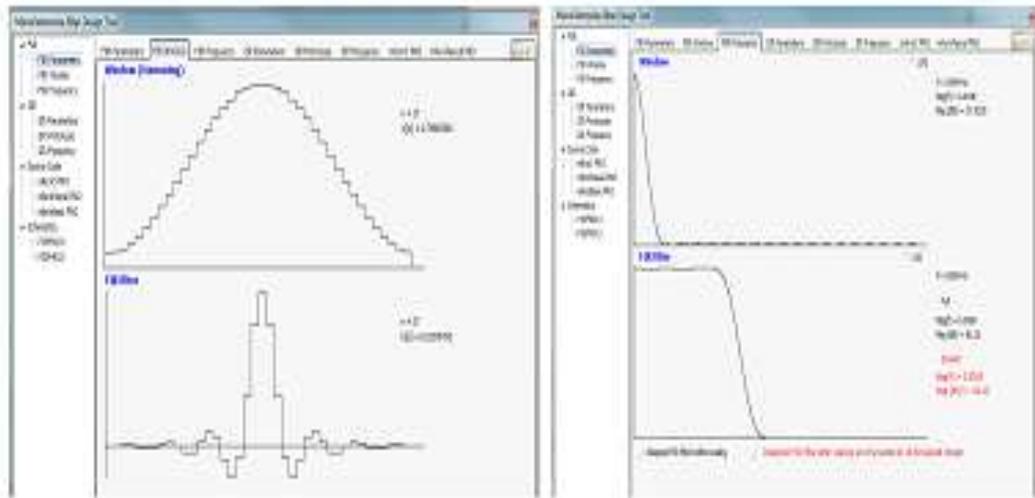
Figura 12. Software Filter Designer Tool a. Frecuencia Limite del Filtro FIR, b. Frecuencia limite, atenuación de banda de Paso Filtro IIR



Fuente: Imagen tomada por el autor del texto del software **Filter Designer Tool**

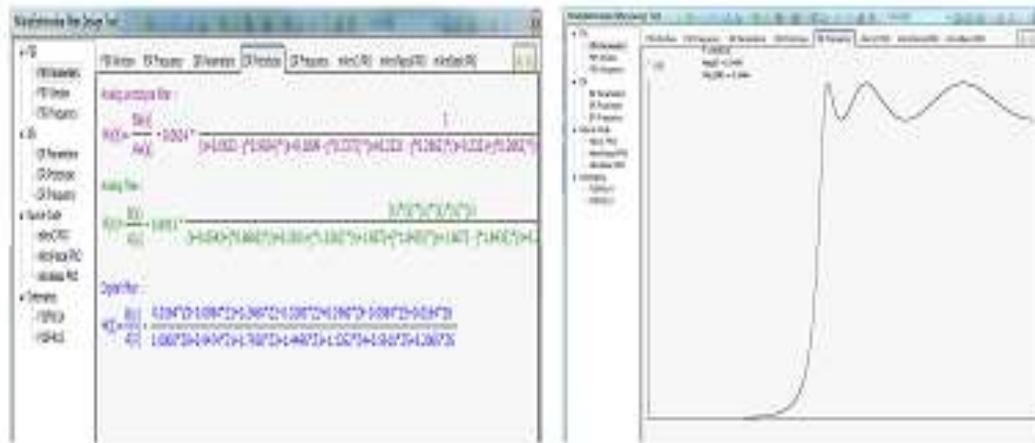
Cuando todos los parámetros se encuentran establecidos, se selecciona una de las opciones del asistente del lado izquierdo, para FIR se encuentran el filtro diseñado con el tipo de ventana, pulso de respuesta, características de la ventana y diseño del filtro FIR. En IIR se encuentran el prototipo de la función de transferencia del filtro análogo y digital, a demás de la característica de amplitud en frecuencia del filtro (Figura 13).

Figura 13. Software Filter Designer Tool a. Tipo de Ventana y Pulso de Respuesta del Filtro FIR, b. Característica de la Ventana y Diseño del Filtro FIR, c. Función de Transferencia del Filtro Análogo y Digital IIR, d. Amplitud en Frecuencia del Filtro



a.

b.



c.

d.

Fuente: Imagen tomada por el autor del texto del software **Filter Designer Tool**

Por último se obtiene el código fuente en los compiladores mikroPascal, mikroBasic o mikroC (Figura 14).

Figura 14. Código Fuente del Programa para el diseño del filtro software Filter Designer Tool.

```

MikroElektronika Filter Design Tool
FIR Window  FIR Frequency  IIR Parameters  IIR Prototype  IIR Frequency  mikroC PRO  mikroPascal PRO  mikroBasic PRO
// Device clock: 020.000000 MHz
// Sampling Frequency: 22050 Hz
// Filter setup:
// Filter kind: IIR
// Filter type: Highpass filter
// Filter order: 6
// Design method: Chebyshev type I

const unsigned int
    BUFFER_SIZE = 8;
const unsigned int
    FILTER_ORDER = 6;

const unsigned int
    COEFF_B[FILTER_ORDER+1] = {0x0433, 0xE6CF, 0x3EFA, 0xACC6, 0x3EFA, 0xE6CF, 0x0433, 0x0000};
const unsigned int
    COEFF_A[FILTER_ORDER+1] = {0x4000, 0x3CA1, 0x72CE, 0x5CC6, 0x48A8, 0x25E7, 0x0DB7};

const unsigned int
    SCALE_B = 1;
const unsigned int
    SCALE_A = -1;

sbit loadPin at LATC2_bit; // DAC load pin
sbit loadPinDir at TRISC3_bit; // DAC load pin
sbit csPin at LATC1_bit; // DAC CS pin
sbit csPinDir at TRISC1_bit; // DAC CS pin

unsigned int inext; // Input buffer index
ydata unsigned int input[BUFFER_SIZE]; // Input buffer
ydata unsigned int output[BUFFER_SIZE]; // Output buffer
    
```

Fuente: Imagen tomada por el autor del texto del software **Filter Designer Tool**

1.4. COMPARACIÓN DE LOS COMPILADORES C30 DE MICROCHIP Y MIKROC DE MIKROELEKTRONIKA.

En la tabla 3 se observan la comparación entre los compiladores C30 de Microchip y MikroC de MikroElektronika.

Tabla 3. Características Principales de dsPIC33FJ128GP802

COMPILADOR C30 DE MICROCHIP	COMPILADOR MIKROC DE MIKROELECTRONIKA
Soporta niveles avanzados de optimización de códigos.	Entorno grafico agradable y maneja un costo elevado.
Soporta numerosos módulos de memoria y niveles de optimización para flexibilidad en el diseño.	El plano que arroja solo es para 2 dsPIC específicos y además no coincide con la configuración en el código.
Permite la escritura de módulos mixtos en C y Esamblador.	Para dsPICs que poseen PLL configurable en software no genera el código necesario para que funcione correctamente.
Proporciona extensiones que permiten soportar propiedades de dsPIC, como las interrupciones y los periféricos.	Para dsPICs con pines remapeables no genera el código necesario para la configuración del modulo SPI.
Posee librerías como son las funciones matemáticas y funciones de control, las cuales son implementadas en esta tesis.	La configuración del formato del A/D que elije la aplicación (unsigned int), no es la adecuada para aplicar el filtro digital, se debe pasar a formato Q15 (signed fraccional).
La aplicación de filtros digitales es mas adecuada en la implementación de audio ya que permite una comunicación optima entre los periféricos y el microcontrolador, permitiendo asi un excelente procesamiento de dicha señal.	Cuando se quiere escoger un filtro IIR, la aplicación presenta inconvenientes, como tergiversación de datos solicitados y generación de coeficientes extras.

2. FILTROS DIGITALES

2.1. FUNDAMENTO DE FILTROS DIGITALES

Los filtros digitales tienen como entrada una señal analógica o digital, un proceso de filtrado o proceso matemático el cual realiza las variaciones de amplitud y fase con respecto a la frecuencia de la señal de entrada y por último una señal de salida que es una señal analógica o digital. En el siguiente diagrama se observa el proceso básico de un filtro digital. (Figura 15)

Figura 15. Diagrama de Bloques Filtro Digital Simplificado



Fuente: El autor del texto

El proceso de filtrado se caracteriza por recolectar muestras actuales y/o posteriores de la señal de entrada y multiplicarlas por coeficientes definidos, recolectar valores de salida en tiempos anteriores y multiplicarlos por coeficientes, y por último realizar la suma respectiva de las señales de entradas y salidas y así obtener una salida actual con valores instantáneos a las señales de entrada.

En los filtros digitales se observan ciertas ventajas y desventajas tecnológicas respecto a los filtros analógicos; entre las ventajas más conocidas son:

- **Son programables:** La operación de un filtro digital está determinada por un programa residente en una memoria. De esta forma el filtro puede modificarse fácilmente sin afectar el hardware.
- **Son estables:** Las características de los filtros digitales no varían con el tiempo ni con la temperatura.
- **Son versátiles:** Los filtros digitales tienen la posibilidad de adaptarse a cambios en las características de la señal de entrada.
- **Son flexibles:** A diferencia de los filtros análogos, los filtros digitales pueden trabajar con señales de baja frecuencia con precisión. Con la evolución de la tecnología, los filtros digitales van siendo aplicados a señales de muy alta frecuencia (RF) que antiguamente era solo un campo designado para los filtros analógicos.

Las desventajas de los filtros digitales:

- **Error de cuantización:** En la etapa de conversión Analógico/Digital se crea el error de cuantización debido a las limitaciones de la memoria numérica y del computo.
- **Velocidad:** La velocidad del proceso de filtrado depende únicamente de la rapidez del procesador en la cual se este trabajando.

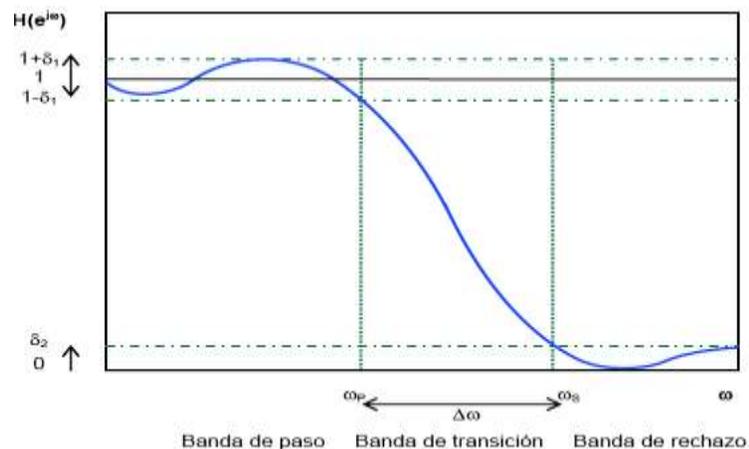
Existen diferentes tipos de filtros digitales y diversas clasificaciones de ellos; los filtros se clasifican de acuerdo con la parte del espectro que dejan pasar y la atenuación que tengan, entre los más comunes se encuentran los filtros Pasa Bajo, Pasa Alto y Pasa Banda. De manera adicional según su orden los filtros se clasifican en Primer y Segundo orden.

De acuerdo con el tipo de respuesta ante entrada unitaria están los filtros: FIR (Respuesta finita al impulso), IIR (Respuesta infinita al impulso), TIIR (Respuesta infinita al impulso truncado). Además con la estructura con que se implementa se encuentran los filtros: Laticce, en cascada, en paralelo.

2.1.1. Etapas de diseño de un filtro digital. Para el diseño de un filtro se deben tener en cuenta las siguientes etapas:

- **Especificaciones:** Las especificaciones que se deben tener en cuenta a la hora de la implementación de un filtro digital son: frecuencia de corte de banda pasante, frecuencia corte de banda supresora, rizado de banda pasante y rizado de banda supresora las cuales se observan en la Figura 16.

Figura 16. Especificaciones de un Filtro Digital



Fuente: <http://dsp1.materia.unsl.edu.ar/Filtros%20FIR%20Resumen.pdf>

En la figura 16 se derivan los siguientes parámetros: Ancho de la región de transición, atenuación a la banda suprimida y atenuación a la banda pasante.

$$\Delta W = W_S - W_P \quad (2.1)$$

$$A_P = 20 * \log\left(\frac{2 + \delta_1}{2 - \delta_1}\right) \quad (2.2)$$

$$A_S = 20 * \log(\delta_2) \quad (2.3)$$

- **Cálculo de coeficientes:** Para realizar el cálculo respectivo de los coeficientes se debe tener claro que tipo de filtro digital se desea implementar en la aplicación deseada, ya sean filtros FIR (Método de ventanas, método de muestreo en frecuencia, método de optimización de error) o filtros IIR (Método de transformación de filtros analógicos, método de minimización de error).

Para el diseño se deben calcular los coeficientes de la función de transferencia del sistema lineal, cuya respuesta debe ser muy cercana a la del filtro ideal deseado. Si la respuesta de la función de transferencia es una expresión Polinómica se implementa un FIR, y si es una expresión racional Polinómica es un IIR.

- **Estructura:** Para poder asignar la estructura del filtro se debe tener en cuenta las diferentes aplicaciones entre ellas se encuentran: estructura directa (sencillez y posible ahorro de retardos), estructura cascada (mejor precisión de palabra de longitud finita), estructura celosía (estabilidad).

- **Implementación:** Los filtros digitales se pueden implementar de manera física mediante: el soporte electrónico (ASIC, FPGA), procesador aplicado (DSP) y soporte aplicado (PC y microcontroladores).

2.2. FILTROS FIR (RESPUESTA AL IMPULSO FINITO):

Los filtros FIR, son conocidos como filtros transversales. Son aquellos donde la respuesta al impulso es de una duración finita, es decir, termina cuando la señal de entrada finaliza. La respuesta al impulso de un filtro FIR de orden n es igual a n+1, luego la salida pasa a ser cero. Se caracterizan por ser muy estables, no son recursivos y mantienen la fase lineal en la mayoría de aplicaciones. Como única desventaja se dirá que su implementación consume mucho más recursos de máquina que los filtros IIR.

La ecuación en diferencias de los filtros FIR es:

$$y[n] = b_0x[n] + b_1x[n - 1] + b_2x[n - 2] + \dots + b_{M-1}[n - M + 1] \quad (2.4)$$

$$y[n] = \sum_{K=0}^{M-1} b_K x[n - k] \quad (2.5)$$

Donde:

M= Longitud del Filtro.

X[n]= Señal de entrada del filtro

Y [n]= Señal de salida del filtro

b_k =coeficiente del filtro

$n-1 \dots n-M+1$ =retardos en el tiempo de la señal ya sea de entrada

Al aplicar la transformada Z queda:

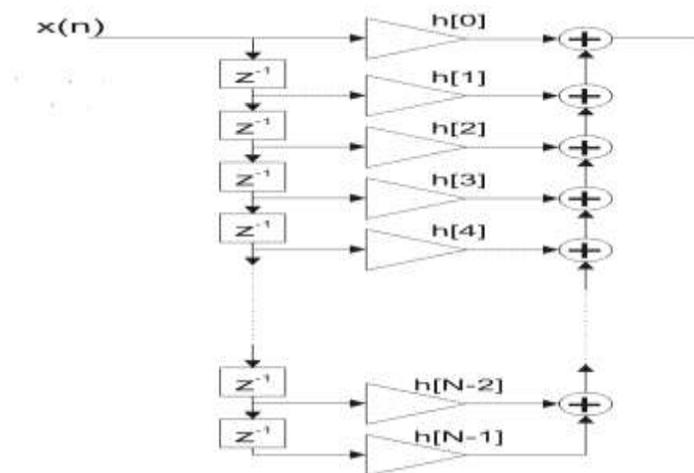
$$H(z) = \sum_{n=0}^{N-1} b(n) z^{-n} \quad (2.6)$$

Los filtros FIR todos los polos de la función de transferencia se encuentran en el origen y agrupaciones pares equitativas de ceros para establecer la linealidad de la fase.

Teniendo la ventaja que pueden diseñarse para ser de fase lineal, es decir, no introducen desfases en la señal, a diferencia de los IIR o los filtros análogos. Por tal motivo son de interés en audio.

En la figura 17 se podrá detallar la estructura de un filtro digital.

Figura 17. Estructura de Filtro FIR

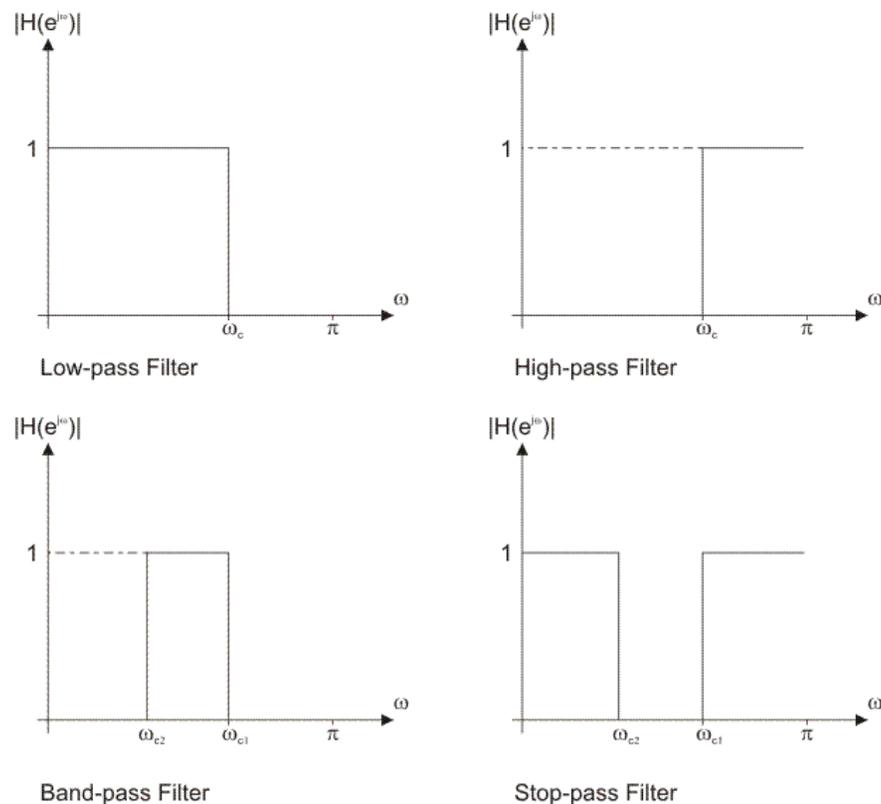


Fuente: Libro digital de Diseño de Filtros Digitales-MiKroelectroniKa

Los filtros FIR se pueden diseñar con diferentes métodos, pero la mayoría de ellos se basan en la aproximación del filtro ideal.

2.2.1. Respuesta a la frecuencia ideal. Los filtros ideales tienen como función eliminar una determinada banda o margen de frecuencias, las cuales pueden ser filtros paso bajo (Low Pass), paso alto (High Pass), paso de banda (Band Pass) y filtros eliminabanda (Band stop o Notch). (Figura 18)

Figura 18. Filtros Ideales



Fuente: Libro digital de Diseño de Filtros Digitales-MiKroelectroniKa

Los filtros ideales se clasifican según su respuesta espectral en donde el tipo de banda determina la forma y la linealidad el retardo define la fase. Una fase lineal es simplemente un desplazamiento temporal que no afecta la forma de la señal de entrada y se comporta en forma general como la ecuación 2.7.

$$Hd(e^{j\omega}) = |H(e^{j\omega})| e^{-j\omega\alpha}, \text{ para } |\omega| < \pi \quad (2.7)$$

Lo que indica que la salida $Y[n]$ de un sistema LTI de fase lineal es la convolución de la señal de entrada $x[n]$ con la magnitud de la respuesta al impulso $|H(e^{j\omega})|$ del sistema, más un desplazamiento en tiempo α muestras.

La respuesta en frecuencia de los diferentes tipos de filtros FIR se determina a través de la transformada de Fourier (Tabla 4).

Tabla 4. Filtros Ideales Respuesta en frecuencia

Tipo de Filtro	Respuesta en frecuencia $hd[n]$
Filtro pasa bajo	$hd[n] = \begin{cases} \frac{\sin[\omega_c(n - M)]}{\pi(n - m)}; & n \neq M \\ \frac{\omega_c}{\pi}; & n = M \end{cases}$
Filtro Pasa alto	$hd[n] = \begin{cases} 1 - \frac{\omega_c}{\pi}; & n \neq M \\ -\frac{\sin[\omega_c(n - M)]}{\pi(n - m)}; & n = M \end{cases}$
Filtro Pasa Banda	$hd[n] = \begin{cases} \frac{\sin[\omega_{c2}(n - M)]}{\pi(n - m)} - \frac{\sin[\omega_{c1}(n - M)]}{\pi(n - m)}; & n \neq M \\ \frac{\omega_{c2} - \omega_{c1}}{\pi}; & n = M \end{cases}$
Filtro Rechaza banda	$hd[n] = \begin{cases} \frac{\sin[\omega_{c1}(n - M)]}{\pi(n - m)} - \frac{\sin[\omega_{c2}(n - M)]}{\pi(n - m)}; & n \neq M \\ \frac{\omega_{c2} - \omega_{c1}}{\pi}; & n = M \end{cases}$

Fuente: Libro digital de Diseño de Filtros Digitales-MiKroelectroniKa

En los filtros Fir se debe tener en cuenta si la fase lineal al impulso satisface la condición de simetría de sus coeficientes.

La secuencia simétrica puede ser par o impar como se observa a continuación:

Simetría par: $h(n) = h[N - n]$

Este tipo de simetría las muestras coinciden en su valor absoluto y signo.

Simetría impar: $h(n) = -h[N - n]$

Este tipo de simetría las muestras coinciden en su valor absoluto, pero su signo es contrario.

En conclusión se puede decir que la simetría impar es cuando el punto medio cae en el punto de la secuencia y es par si cae entre dos puntos.

De acuerdo a las Tablas 5 y 6 se observa las secuencias y aplicaciones para cada tipo de filtro Fir en general.

Tabla 5. Secuencias Simétricas

Secuencias Simétricas L=1/2(N-1), M=1/2N, F es la frecuencia digital =f/f_s don f_s es la frecuencia de muestreo					
TIP O	SIMETR ÍA	N	H(F)	H(0)	H(1/2)
1	Par	Imp ar	$h[0] + 2 \sum_{K=1}^L h[K] \cos(2K\pi f)$	$h[0] + 2 \sum_{K=1}^L h[K]$	$h[0] + 2 \sum_{K=1}^L (-1)^K h[K]$
2	Par	Par	$2 \sum_{K=1}^M h[K] \cos[2\pi F (L + \frac{1}{2})]$	$2 \sum_{K=1}^L h[K]$	0
3	Impar	Imp ar	$-j2 \sum_{K=1}^M h[K] \sin(2K\pi f)$	0	0
4	Impar	Par	$-j2 \sum_{K=1}^M h[K] \sin[2\pi F (L + \frac{1}{2})]$	0	$-2 \sum_{K=1}^M (-1)^K h[K]$

Fuente: <http://www.tecnun.es/asignaturas/tratamiento%20digital/tema9.pdf>

Tabla 6. Aplicaciones a Tipos de Filtros

Aplicaciones de las Secuencias Simétricas LP: Filtro Pasa Bajas, HP: Filtro Pasa Altas, BP: Filtro Pasa Banda, BS. Filtro Para Banda		
TIPO	OBSERVACIONES H(F)	APLICACIÓN
1		Todo tipo de filtros
2	 H(1/2)=0 	Sólo LP, BP y BS
3	 H(0) =0= H(1/2) 	Sólo LP y BS
4	 H(0)=0 	Sólo HP , BP y BS

Fuente: <http://www.tecnun.es/asignaturas/tratamiento%20digital/tema9.pdf>

2.2.2. Diseño de filtros FIR. En el diseño de los filtros FIR existen tres métodos los cuales son el método de ventanas, método de muestreo en frecuencia y

método por optimación de error. Debido a la extensión del tema se ara énfasis en el diseño de filtros digitales FIR con el método de ventanas.

2.2.2.1. Método de ventanas. Este método es uno de los más utilizados, comunes y simples a la hora de diseñar filtros digitales FIR. Esta técnica se caracteriza por obtener los coeficientes de un filtro realizable mediante la respuesta de un filtro ideal $Hd(n)$ con el producto de una función de truncamiento que es conocida como ventana $w[n]$.

Las ventanas se diferencian una de las otras por sus determinadas características como son la banda de transición, frecuencias con respecto a ancho de banda, amplitud de los lóbulos, etc. Se debe tener en cuenta las especificaciones del filtro para poder determinar la ventana adecuada.

A continuación se darán a conocer las principales tipos de ventanas en donde se observan gráficamente el comportamiento en el tiempo como en frecuencia.

Ventana Rectangular. La ventana rectangular es poco utilizada por su baja atenuación en su banda de rechazo ya que permite reducir la zona de transición al aumentar los números de coeficientes, es decir consiguiendo una mínima región de transición a un costo de atenuación de banda de rechazo lo cual no tiene efecto sobre el rizado. Los coeficientes se definen según la relación de la ecuación 2., el comportamiento en frecuencia y en el tiempo se observan en la figura 19 en donde se puede ver la región más estrecha con una atenuación de 13dB en su primer lóbulo.

$$W[n] = 1 \text{ para } 0 \leq n \leq N - 1$$

$$W[n] = 0 \text{ para cualquier otro valor} \quad (2.8)$$

Ventana Triangular (Barlett). La ventana triangular disminuyen los efectos de las últimas muestras. Obteniendo como resultado la atenuación de rechazo de la banda, donde la selectividad es menor.

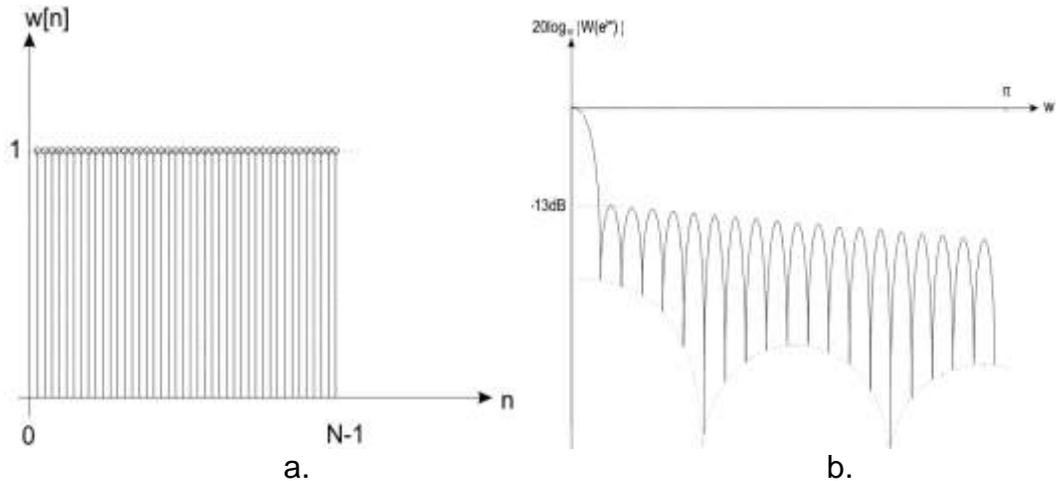
Una de sus ventajas es la simplicidad en el cómputo de coeficientes de acuerdo con la fórmula 2.9, el comportamiento en frecuencia y en el tiempo se observan en la figura 20.

$$W[n] = \frac{2n}{N-1} \text{ para } 0 \leq n \leq \frac{N-1}{2}$$

$$W[n] = 2 - \frac{2n}{N-1} \text{ para } \frac{N-2}{2} \leq n \leq N-1 \quad (2.9)$$

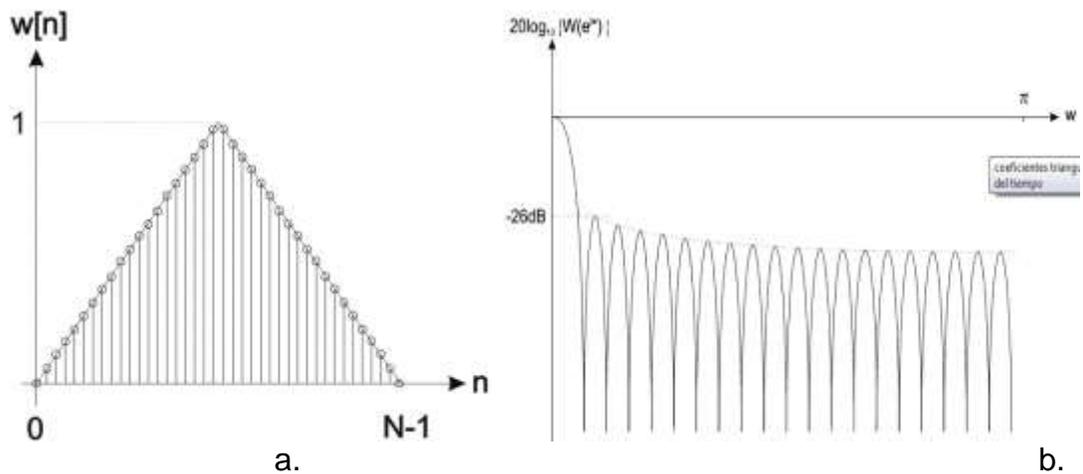
$$W[n] = 0 \text{ para cualquier otro valor}$$

Figura 19. a. Ventana rectangular en el dominio del tiempo. b. ventana rectangular en el dominio de la frecuencia.



Fuente: Libro digital de Diseño de Filtros Digitales-MiKroelectroniKa

Figura 20 a. Ventana triangular en el dominio del tiempo. b. ventana triangular en el dominio de la frecuencia.



Fuente: Libro digital de Diseño de Filtros Digitales-MiKroelectroniKa

Ventana Hann. La ventana Hann se utiliza para disminuir los malos efectos de la característica de frecuencia producidos por las últimas muestras de una señal que se esté filtrando.

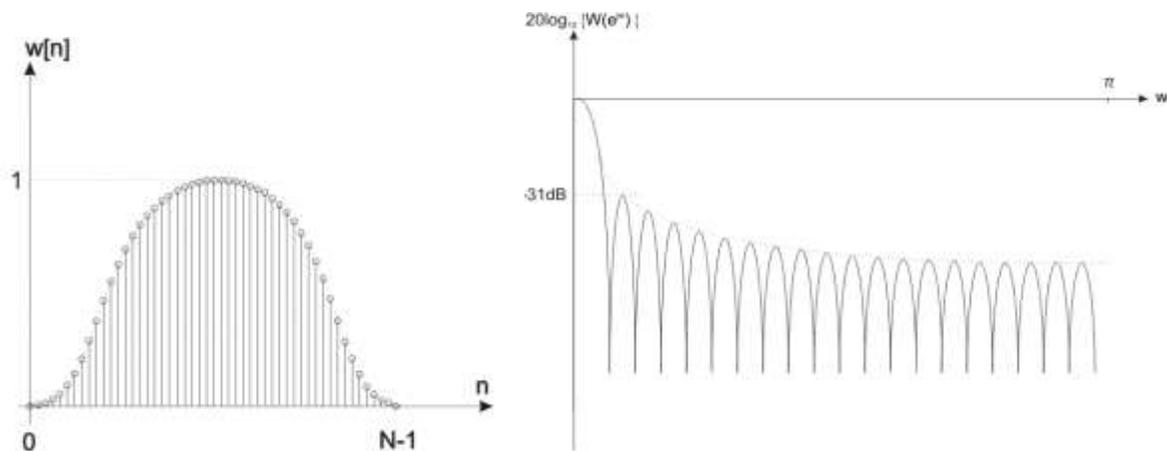
Esta ventana permite truncar gradualmente la respuesta ideal.

Los coeficientes del filtro Hann, pueden expresarse de acuerdo con la ecuación 2.10, el comportamiento en frecuencia y en el tiempo se observan en la figura 21 En donde los lóbulos laterales en el dominio de la frecuencia tienen 31dB de atenuación. Su grande ventaja es la capacidad de aumento rápido en la atenuación de la banda de rechazo de los lóbulos siguientes.

$$W[n] = \frac{1}{2} \left[1 - \cos \left(\frac{2\pi n}{N-1} \right) \right] \text{ para } 0 \leq n \leq N-1$$

$$W[n] = 0 \text{ para cualquier otro valor} \quad (2.10)$$

Figura 21 a. Ventana Hann en el dominio del tiempo. b. ventana Hann en el dominio de la frecuencia.



Fuente: Libro digital de Diseño de Filtros Digitales-MiKroelectroniKa

Ventana Hamming. Es una ventana de las más populares y de las más comúnmente usadas. Este tipo de ventana combina la Hann con una ventana uniforme, teniendo como resultado una mejor definición de los lóbulos laterales en respuesta en frecuencia.

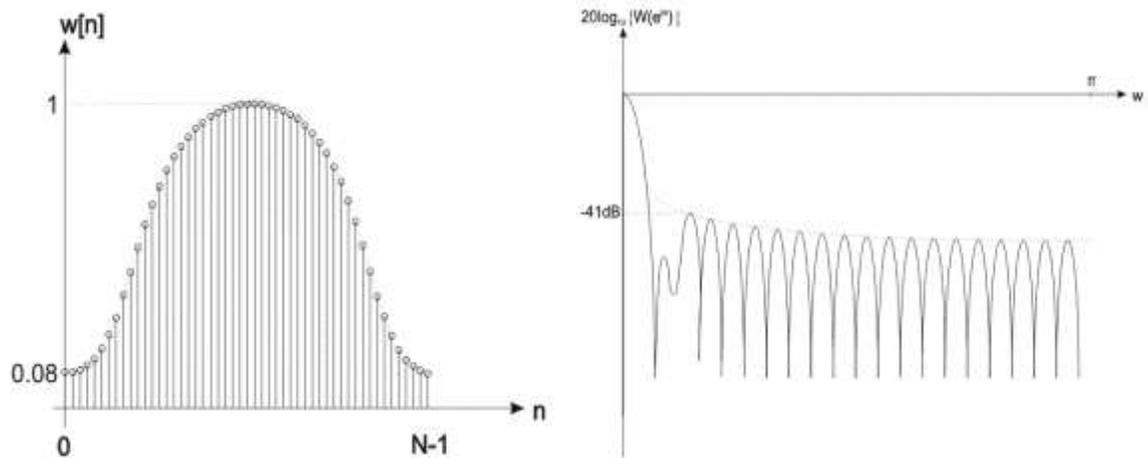
Los coeficientes están dados por la fórmula de la ecuación 2.11, el comportamiento en frecuencia y en el tiempo se observan en la figura 22.

En donde el primer lóbulo se atenúa ya que los lóbulos laterales en el dominio de la frecuencia tienen 41dB de atenuación mínima de banda de rechazo.

$$W[n] = 0.54 - 0.46 \cos \left(\frac{2\pi n}{N-1} \right) \text{ para } 0 \leq n \leq N-1$$

$$W[n] = 0 \text{ para cualquier otro valor} \quad (2.11)$$

Figura 22 a. Ventana Hamming en el dominio del tiempo. b. ventana Hamming en el dominio de la frecuencia.



Fuente: Libro digital de Diseño de Filtros Digitales-MiKroelectroNiKa

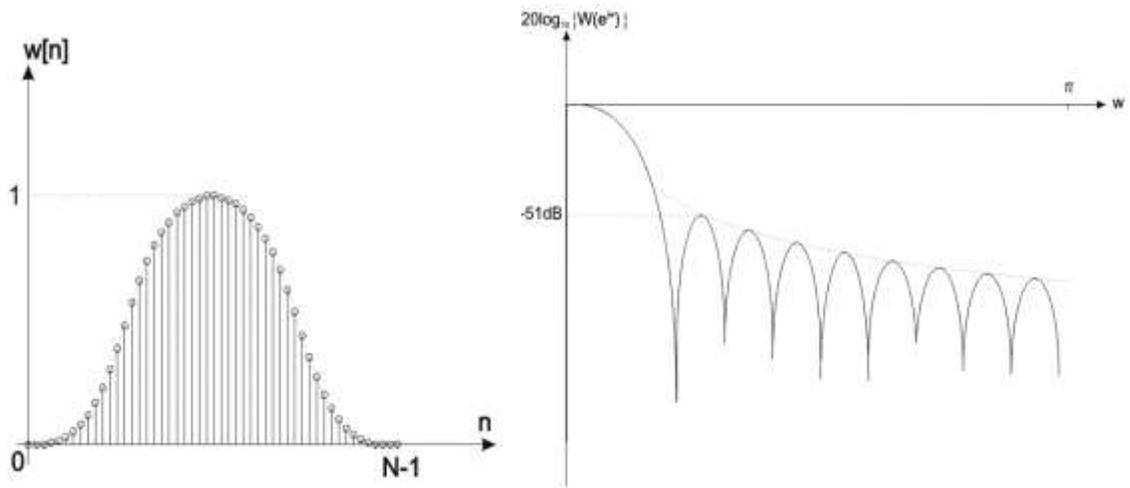
Ventana Blackman. Es una de las más populares y comúnmente usadas. Con una atenuación relativamente alta, lo que permite que sea utilizada en la mayoría de aplicaciones. Esta ventana en el dominio de la frecuencia el lóbulo principal y los laterales asciende a 51dB.

Los coeficientes de esta ventana están dados por la fórmula de la ecuación 2.12, el comportamiento en frecuencia y en el tiempo se observan en la figura 23.

$$W[n] = 0.42 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right) + 0.08 \cos\left(\frac{4\pi n}{N-1}\right) \text{ para } 0 \leq n \leq N-1$$

$$W[n] = 0 \text{ para cualquier otro valor} \quad (2.12)$$

Figura 23 a. Ventana Blackman en el dominio del tiempo. b. ventana Blackman en el dominio de la frecuencia.



Fuente: Libro digital de Diseño de Filtros Digitales-MiKroelectroniKa

De acuerdo a la Tabla 7 se determinan las características generales de algunas funciones de ventanas en el diseño de filtros FIR.

Tabla 7. Características Frecuenciales de Algunas funciones de Ventana.

Tipo de Ventana	Ancho de Transición Aproximado del Lóbulo principal	Atenuación en Lóbulos Laterales (dB)
Rectangular	$4\pi/M$	-13
Triangular (Bartlett)	$8\pi/M$	-26
Hanning	$8\pi/M$	-31
Hamming	$8\pi/M$	-41
Blackman	$12\pi/M$	-57

Fuente: dsPIC Diseño Práctico de Aplicaciones, Jose M. Angulo, p606

Ventana Kaiser. Este tipo de ventana es catalogada como la más óptima ya que utiliza la función de Bessel modificada de primera especie. Una sus principales características es la concentración de la mayor energía en su lóbulo principal.

La ventana Kaiser se define por la expresión:

$$W[n] = \frac{I_0\left(\beta \sqrt{1 - \left(\frac{n - N/2}{N/2}\right)^2}\right)}{I_0(\beta)}; \text{ para } 0 \leq n \leq N - 1 \quad (2.13)$$

Donde I_0 es la función de Bessel modificada de primer tipo, definida por la expresión 2.14.

$$I_0(X) = 1 + \sum_{k=1}^{\infty} \left(\frac{1}{k!} \left(\frac{X}{2}\right)^k\right)^2 \quad (2.14)$$

El parámetro de β se calcula mediante la ecuación 2.15.

$$\beta = \begin{cases} 0.1102(A - 8.7), & A > 50 \\ 0.5842(A - 21)^{0.4} + 0.07886(A - 21), & 21 \leq A \leq 50, \\ 0, & A < 21 \end{cases} \quad (2.15)$$

Donde A es la atenuación de la banda de rechazo deseada. El orden del filtro es determinado mediante la ecuación 2.16

$$N = \begin{cases} \frac{A - 7.95}{2.285 \Delta w} + 1, & A > 21dB \\ \frac{5.794}{\Delta w} + 1, & A \leq 21dB \end{cases} \quad (2.16)$$

De igual forma se deben tener en cuenta la aproximación del filtro pasa bajos y la anchura de la región de transición (normalizado), que se verán a continuación.

$$A = -20 \log_{10} \delta$$

$$\Delta w = \frac{w_p - w_s}{2w_n} \quad (2.17)$$

Una de las principales características de este es tipo de venta, es que al variar el orden del filtro M y el parámetro β se pueden ajustar el ancho del lóbulo central y la amplitud de los lóbulos laterales.

De acuerdo a las etapas mencionadas anteriormente se enunciarán las pautas para el diseño de un filtro FIR con el método de ventanas.

Especificaciones: los datos inicialmente en el diseño son: frecuencia de corte de banda pasante, frecuencia corte de banda supresora, rizado de banda pasante y rizado de banda supresora.

Elección de la ventana: se selecciona el tipo de ventada de acuerdo con las especificaciones del filtro deseado, entre ellas se encuentran la ventana rectangular, Barlett, Hanning, Haming, Blackman y káiser.

Estimación de Parámetros: los parámetros que se deben estimar son la longitud del filtro en función del ancho de transición.

Enventanado: para determinar los coeficientes se debe calcular la repuesta al impulso del filtro ideal $hd[n]$, luego los valores de la función de la venta $w[n]$, mediante la convolucion o el pructo de estos dos parametos se obtienen los coefcients del filtro deseado.

Ejemplo 1:

Diseñar un filtro FIR paso-bajo, Ventana Hamming de orden 64, con frecuencia de corte de la banda pasante de 1[KHz], frecuencia de corte de la banda suprimida 2[KHz] y frecuencia de muestreo de 48[KHz].

1. Especificación del filtro:

Cuando no se especifica el orden filtro, se debe determinar mediante la fórmula del ancho de transición aproximado del Lóbulo principal, estas especificaciones se encuentran en la tabla 2.4. En este caso no se tendrá en cuenta ya que el orden es dado.

Las frecuencias y pulsaciones digitales de corte para la banda suprimida y banda pasante son:

$$f_s = \frac{F_s}{F_m} = \frac{2 \text{ KHz}}{48 \text{ KHz}} = 0.041 \rightarrow \omega_s = 2\pi f_s = 0.083\pi$$

$$f_p = \frac{F_p}{F_m} = \frac{1 \text{ KHz}}{48 \text{ KHz}} = 0.020 \rightarrow \omega_p = 2\pi f_p = 0.041\pi$$

La frecuencia de corte es:

$$\omega_c = \frac{\omega_s + \omega_p}{2} = \frac{0.083 + 0.041}{2} = 0.062\pi$$

La respuesta espectral del filtro ideal se muestra a continuación, en donde la fase en este ejemplo es $\alpha=(M-1)/2= 32$ y la frecuencia de corte $\omega_c=0.062\pi$.

$$Hd(e^{jm}) = e^{-jma\pi} \left(\frac{w}{2wc} \right) = \begin{cases} e^{-jm\frac{M-1}{2}} & \text{si } |w| \leq wc \\ 0 & \text{si } 0 < |w| \end{cases} = \begin{cases} e^{-j27m} & \text{si } |w| \leq 0.062\pi \\ 0 & \text{si } 0 < |w| \end{cases}$$

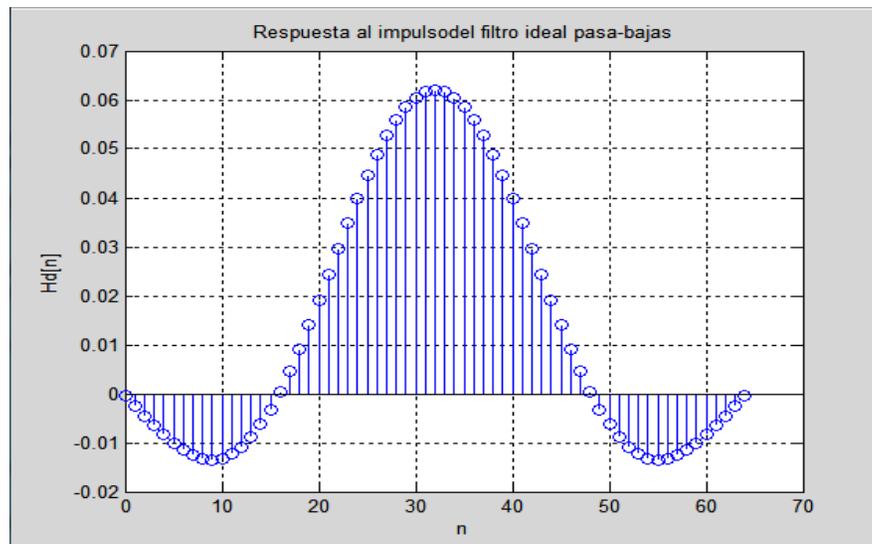
Realizando la transformada inversa de Fourier discreta a $Hd(e^{jm})$ se obtendría una respuesta impulsiva con una cantidad infinita de muestras.

$$Hd[n] = \frac{wc}{\pi} \text{sinc} \left[\left(n - \frac{M-1}{2} \right) \frac{wc}{\pi} \right]$$

Reemplazando los datos suministrados en este ejemplo en el filtro ideal pasa bajos $Hd[n]$ se obtiene la respuesta al impulso como se ilustra en la Figura 24.

$$Hd[n] = 0.062 \text{sinc}[(n - 32)0.062] = \frac{\text{sen}[0.062\pi(n - 32)]}{\pi(n - 32)}$$

Figura 24. Respuesta al impulso del filtro ideal pasa bajas



Fuente: El autor del texto

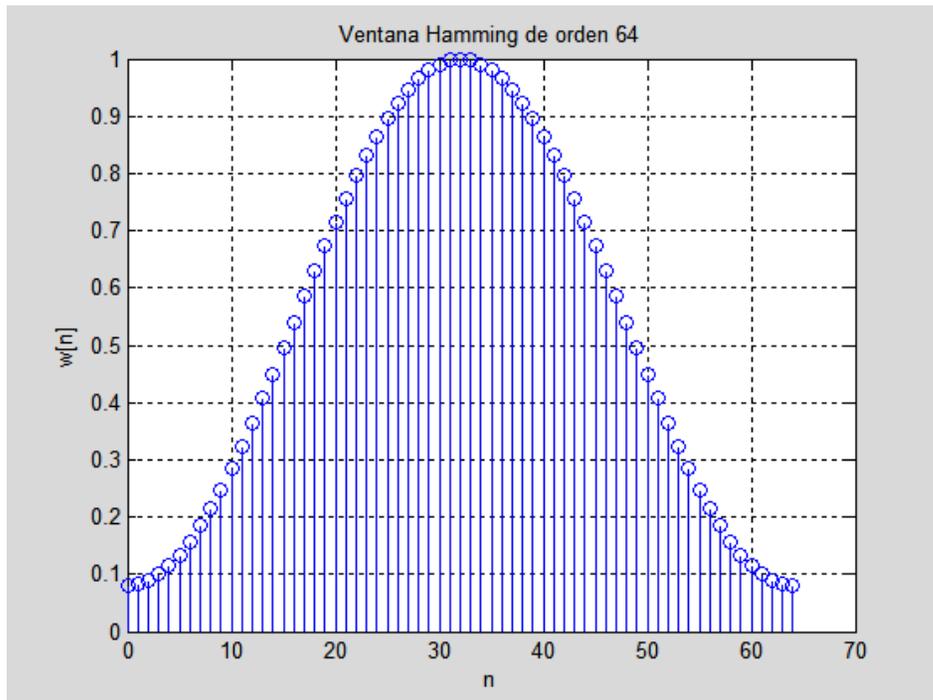
La ventana Hamming está dada por siguiente función:

$$w[n] = 0.54 - 0.64 \cos \left[\frac{2\pi n}{M-1} \right] \text{ para } 0 \leq n \leq M-1$$

Reemplazando los valores del ejemplo se obtiene una ventana de 65 muestras como se ilustra en la Figura 25.

$$w[n] = 0.54 - 0.64\cos\left[\frac{2\pi n}{64}\right] \text{ para } 0 \leq n \leq 64$$

Figura 25. Respuesta al impulso del filtro ideal pasa bajas.



Fuente: El autor del texto

A continuación se debe realizar un truncamiento de $h[n]$, lo que equivale a multiplicar $hd[n]$ por una ventana $w[n]$ de longitud N , en este caso sería la ventana hamming

$$h[n] = hd[n]w[n] = \left(\frac{\text{sen}[0.062\pi(n - 32)]}{\pi(n - 32)}\right) * 0.54 - 0.64\cos\left[\frac{2\pi n}{64}\right]$$

Los coeficientes resultantes son calculados mediante Matlab y son reflejados en la Tabla 8, de igual forma se ve ilustrado en la Figura 26 el comportamiento de los coeficientes.

```

close all
clear all
clc
m=65;
n=0:1:64;
hd=0.062*sinc((n-32)*0.062);           //filtro ideal pasa bajos
wn=(0.54-0.46*cos((2*pi*n)/(m-1)));    //Ventana hamming
hn=hd.*wn;
stem (n,hn)
grid on

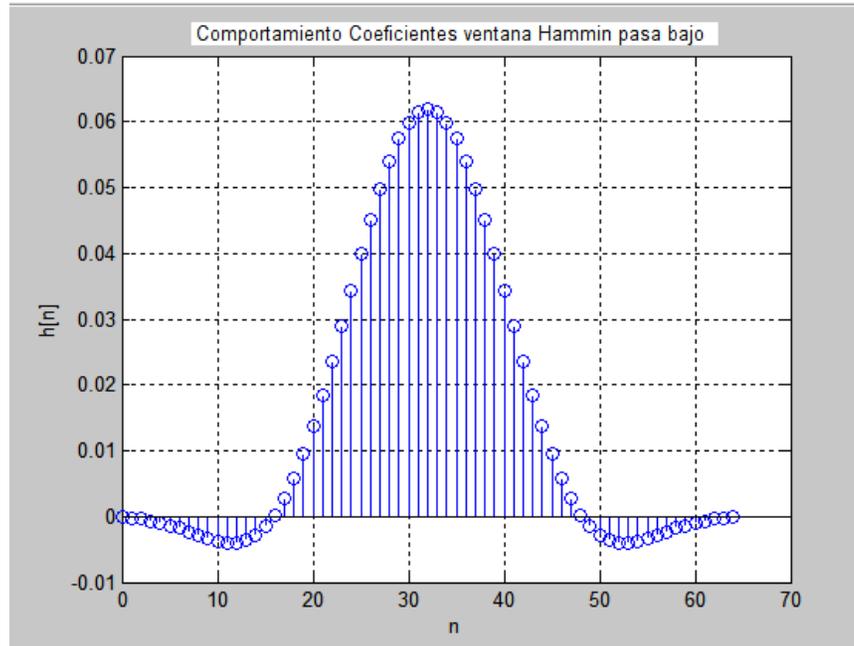
```

$$h[n] = h_d[n] * w[n]$$

Tabla 8. Coeficientes ventana Hamming Pasa-Bajo

h[n]	hd[n]	w[n]	h[n]= hd[n]*w[n]
h[0]=h[64]	-0,0004997894750370	0,0800000000000000	-0,00003998315800296
h[1]=h[63]	-0,0024910236626697	0,082215025730789	-0,00020479957452239
h[2]=h[62]	-0,0045176585945831	0,088838771014514	-0,00040134323740592
h[3]=h[61]	-0,0065073167913362	0,099807445563183	-0,00064947866641368
h[4]=h[60]	-0,0083842041587618	0,115015415044808	-0,00096431272114040
h[5]=h[59]	-0,0100712945205031	0,134316218399757	-0,00135273819438417
h[6]=h[58]	-0,0114926256687183	0,157523978340829	-0,00181036411691844
h[7]=h[57]	-0,0125756385311958	0,184415191453141	-0,00231913878737597
h[8]=h[56]	-0,0132534882625369	0,214730880654188	-0,00284593320635450
h[9]=h[55]	-0,0134672554902755	0,248179089284723	-0,00334229120274127
h[10]=h[54]	-0,0131679876338150	0,284437692810983	-0,00374547202152588
h[11]=h[53]	-0,0123185041420442	0,323157501060041	-0,00398081701534078
h[12]=h[52]	-0,0108949055687802	0,363965621112059	-0,00396537107229832
h[13]=h[51]	-0,0088877344576791	0,406469048462947	-0,00361258896800421
h[14]=h[50]	-0,0063027458082092	0,450258451872581	-0,00283786457015068
h[15]=h[49]	-0,0031612561512075	0,494912115448402	-0,00156454396926838
h[16]=h[48]	0,0004999473637722	0,5400000000000000	0,00026997157643703
h[17]=h[47]	0,0046291433560464	0,585087884551598	0,00270845569347530
h[18]=h[46]	0,0091606501210523	0,629741548127419	0,00576884198908513
h[19]=h[45]	0,0140162157496892	0,673530951537053	0,00944035513083679
h[20]=h[44]	0,0191067903107689	0,716034378887941	0,01368111873271360
h[21]=h[43]	0,0243346340252914	0,756842498939959	0,01841748522649091
h[22]= h[42]	0,0295957047399454	0,795562307189017	0,02354522714579591
h[23]=h[41]	0,0347822590300064	0,831820910715277	0,02893261038307462
h[24]=h[40]	0,0397855942216617	0,865269119345812	0,03442524607482714
h[25]=h[39]	0,0444988537539246	0,895584808546859	0,0398524974197632
h[26]=h[38]	0,0488198157662883	0,922476021659171	0,0450351094262193
h[27]=h[37]	0,0526535846924320	0,945683781600243	0,0497936410867478
h[28]=h[36]	0,0559151079784289	0,964984584955192	0,0539572172652890
h[29]=h[35]	0,0585314447644979	0,980192554436816	0,0573720863585906
h[30]=h[34]	0,0604437203352637	0,991161228985486	0,0599094721319550
h[31]=h[33]	0,0616087091462026	0,997784974269211	0,0614722442702030
h[32]	0,0620000000000000	1	0,0620000000000000

Figura 26. Coeficientes Ventana Hamming Pasa Bajos



Fuente: El autor del texto

2.3. FILTROS IIR (RESPUESTA AL IMPULSO INFINITO):

Son aquellos que al aplicarle en la entrada un impulso, su salida no volverá al reposo, es decir, será infinita. Se denominan filtros con memoria ya que su salida dependerá de la entrada actual, entradas pasadas y salidas pasadas. Los filtros IIR son recursivos, pueden no ser estables y su fase no es lineal. Su principal ventaja radica en que pueden tener las mismas características de los FIR con un orden menor.

El diseño de filtros IIR está ligado a la forma analógica de diseñar filtros, por lo que existe un modelo IIR para cada configuración de filtro analógico.

La ecuación 2.18 en diferencias para los filtros IIR es:

$$y(n) = b(0)x(n) + b(1)x(n - 1) + b(n)x(n - N) - a(1)y(n - 1) - a(2)y(n - 2) - a(M)y(n - M) \quad (2.18)$$

Donde

a y b : son los coeficientes del filtro.

M y N : el orden del filtro es el máximo.

Al aplicar la transformada Z a la ecuación 2.18 obtendremos como resultado la ecuación 2.19.

$$H(Z) = \frac{\sum_{k=0}^N b[k].z^{-k}}{1 + \sum_{k=1}^N a[k].z^{-k}} \quad (2.19)$$

Donde

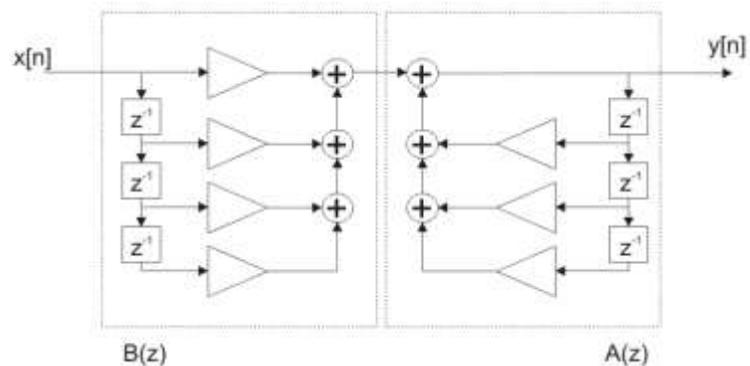
N es el orden del filtro.

b_k es el coeficiente no recursivo del filtro IIR.

a_k es el coeficiente de la parte recursiva del filtro IIR.

A continuación en la figura 27 se podrá detallar la estructura de un filtro digital IIR.

Figura 27. Estructura de Filtro IIR



Fuente: Libro digital de Diseño de Filtros Digitales-MiKroelectroNiKa

2.3.1. Diseño de filtros IIR. El método más común a la hora de diseñar filtro IIR es mediante la transformación de filtros analógicos filtro digital que cumpla con los parámetros establecidos.

2.3.1.1. Diseño de filtros IIR mediante transformada Bilineales. La transformación bilineal es un proceso que permite pasar de un sistema de tiempo discreto a tiempo continuo, mediante un cambio matemático que transforma el eje $j\omega$ en la circunferencia unidad del plano z. Para esto es de suma importancia el conocimiento de la función de transferencia del filtro que se desea diseñar.

Los coeficientes obtenidos mediante la transformación forman los coeficientes del filtro IIR.

Esta transformación de variables esta descrita por la ecuación 2.20:

$$S = \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}} = \frac{2}{T} \frac{z-1}{z+1} \quad (2.20)$$

Y su relacion inversamente esta dad por la ecuación 2.21.

$$z = \frac{1 + \frac{T}{2}s}{1 - \frac{T}{2}s} \quad (2.21)$$

En el plano Z y S, estas variables tienen un equivalente en función de la frecuencia angular $s = \sigma + j\Omega$ y $z = e^{j\omega}$. Si lo sustituimos en la ecuación (2.21) obtendremos:

$$z = \frac{1 + \sigma \left(\frac{T}{2}\right) + j\Omega \left(\frac{T}{2}\right)}{1 - \sigma \left(\frac{T}{2}\right) - j\Omega \left(\frac{T}{2}\right)} \quad (2.22)$$

En la transformación bilineal, para obtener la relación de sus respuestas en frecuencia se sustituye en las variables S por $j\Omega$ y Z por $e^{j\omega}$ en las ecuaciones (2.20) y (2.21) obteniendo como resultado:

$$\Omega = \frac{2}{T} \tan\left(\frac{\omega}{2}\right) \quad (2.23)$$

$$\omega = 2 \arctan\left(\frac{\Omega T}{2}\right) \quad (2.24)$$

El intervalo de la frecuencia digital es de $0 \leq \omega \leq \pi/T$, lo cual equivale de frecuencia analógica de $0 \leq \Omega \leq \infty$.

Este tipo de diseños para filtros IIR mediante transformada bilineal, se establece mediante las especificaciones requeridas del filtro digital en dominio en la frecuencia, las cuales son transformadas a frecuencia continuas, con el fin de obtener el filtro analógico.

Los filtros analógicos se eligen de acuerdo a la respuesta de cada uno y dependiendo de la aplicación. Dicha respuesta debe satisfacer las características requeridas. Una vez obtenido el diseño del filtro analógico se procesa a hacer la transformación digital. En la tabla 9 se muestran las características de cada filtro.

Tabla 9. Respuestas de los distintos tipos de filtros analógicos

Butterworth	Chebyshev tipo I
Solo Polos Máximamente plana para $\Omega \rightarrow 0$ Monótona decreciente $0 < \Omega$	Sólo polos Rizado constante $\Omega < \Omega_0$ Monótona decreciente $\Omega_0 < \Omega$
Chebyshev tipo II	Cauer
Polos sobre elipse Ceros sobre $s = j\Omega$ Rizado constante en $\Omega_0 < \Omega$ Monótona decreciente en $\Omega < \Omega_0$	Zona de transición más estrecha para $=N$ Rizado constante en $\Omega < \Omega_0$ y en $\Omega_0 < \Omega$

Ejemplo 2:

Diseñar un filtro IIR paso-alto, de orden 3 con frecuencia de corte de la banda pasante de 1[KHz], frecuencia de corte de la banda suprimida 5[KHz] y frecuencia de muestreo de 20[KHz].⁸

1. Especificaciones

Orden del filtro $N=3$

Frecuencia de corte de la banda pasante de 1[KHz]

Frecuencia de corte de la banda suprimida 5[KHz]

Frecuencia de muestreo de 20[KHz].

2. Filtro Análogo tipo Butterwork

3. Para calcular los polos y ceros en el filtro Butterwork, se determinan mediante la expresión siguiente:

$$S_k = e^{j\pi\left(\frac{1}{2} + \frac{2K+1}{2N}\right)} = \cos\left(\pi\left(\frac{1}{2} + \frac{2K+1}{2N}\right)\right) + j\sin\left(\pi\left(\frac{1}{2} + \frac{2K+1}{2N}\right)\right)$$

Se evalúa $N=3$, obteniendo los Polos del sistema.

$$S_1 = -0,5 + j 0,86$$

$$S_2 = -1$$

$$S_3 = -5 - j 0,86$$

Obteniendo así la función de transferencia del prototipo de filtro análogo.

$$H(s) = \frac{1}{(s + 0.5 + j 0.866) * (s + 1) * (s + 0.5 - j 0.866)}$$

⁸ MILIVOJEVIC Zorann, Digital Filter Design, Mikroelektronika, capitulo 3 ejemplo 2.

3. La frecuencia de corte del prototipo del filtro análogo

$$\Omega_c = \operatorname{tg}\left(\pi \cdot \frac{f_c}{f_s}\right) = \operatorname{tg}\left(\pi \cdot \frac{5000}{20000}\right) = \operatorname{tg}(0.25 \cdot \pi) = 1$$

4. Función de transferencia del filtro Butterworth.

$$H(s) = \frac{1}{(-p_1) * (-p_2) * (-p_3)} * \frac{s^3}{(s - \Omega_c * p_1) * (s - \Omega_c * p_2) * (s - \Omega_c * p_3)}$$

$$H(s) = \frac{1}{(0,5 + j0.86) * (1) * (0,5 - j0.86)} * \frac{s^3}{(s - 1 * (-0,5 + j0.86)) * (s - 1 * (-1)) * (s - 1 * (-0,5 - j0.86))}$$

$$H(s) = \frac{s^3}{(s + 0.5 + j0.866) * (s + 1) * (s + 0.5 - j0.866)}$$

5. Aplicando la transformada Bilineal se obtiene.

$$S = \frac{1 - z^{-1}}{1 + z^{-1}}$$

$$H(z) = H_0 * (-1)^{N-M} * \frac{\prod_{k=1}^M (1 - z_k)}{\prod_{k=1}^M (1 - p_k)} * (1 + z^{-1})^{N-M} * \frac{\prod_{k=1}^M (1 - \frac{1 + z_k}{1 + z_k} * z^{-1})}{\prod_{k=1}^M (1 - \frac{1 + p_k}{1 + p_k} * z^{-1})}$$

$$H(z) = H_0 * (-1)^{3-3} * \frac{1 * 1 * 1}{(1 - p_1) * (1 - p_2) * (1 - p_2)} * (1 + z^{-1})^{3-3} * \frac{(1 - z^{-1}) * (1 - z^{-1}) * (1 - z^{-1})}{(1 - \frac{1 + p_1}{1 - p_1} * z^{-1}) * (1 - \frac{1 + p_2}{1 - p_2} * z^{-1}) * (1 - \frac{1 + p_3}{1 - p_3} * z^{-1})}$$

$$H(z) = \frac{1}{(1 + 0,5 + j0.86) * (1 + 1) * (1 + 0,5 - j0.86)} * \frac{1}{(1 - z^{-1})^3} * \frac{1}{(1 - \frac{1 - 0,5 - j0.86}{1 + 0,5 + j0.86} * z^{-1}) * (1 - 0) * (1 - \frac{1 - 0,5 + j0.86}{1 + 0,5 - j0.86} * z^{-1})}$$

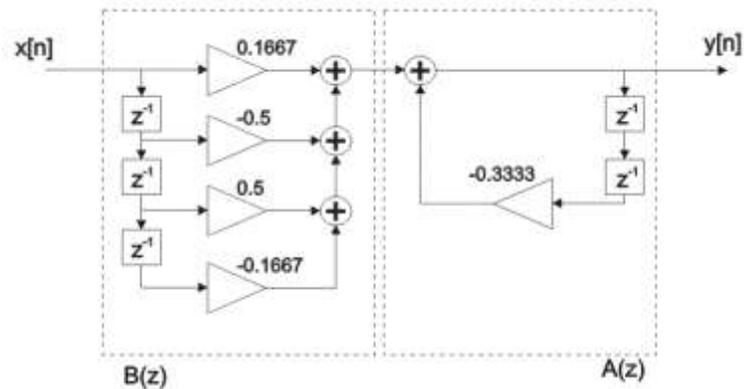
$$H(z) = \frac{0.1667 - 0.5 * z^{-1} + 0.5 * z^{-2} - 0.1667 * z^{-3}}{1 + 0.3333 * z^{-2}}$$

6. Tendríamos como resultado al filtro IIR :

Numerador: a1=0.1667
a2=-0.5
a3=0.1667
a4=0.5

Denominador: b=-0.3333

Figura 28. Filtro digital IIR directo



Fuente: Libro digital de Diseño de Filtros Digitales-MiKroelectroniKa

3. ENTRENADOR DE AUDIO

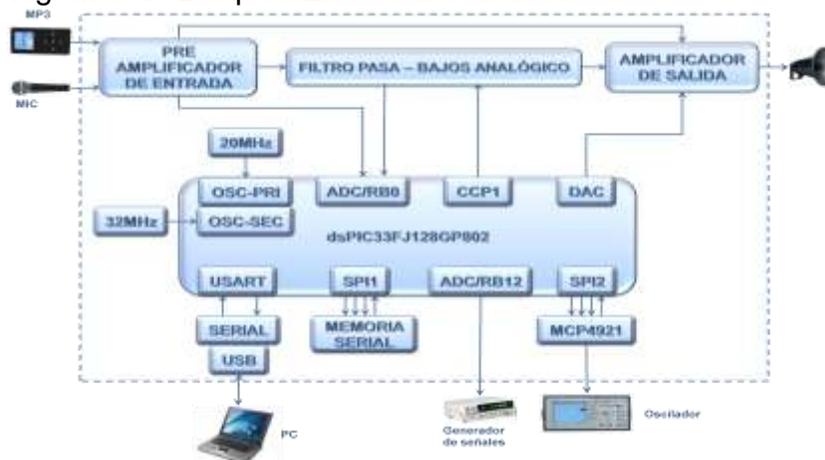
En este capítulo se mencionarán las etapas del entrenador de audio, en donde se pretende mostrar cada una de ellas con su funcionamiento.

La tarjeta electrónica diseñada proporciona un sistema que sirve como herramienta de aplicación para filtros digitales, ésta se encuentra conformada por las siguientes partes o secciones principales:

- Pre-amplificador de entrada.
- Amplificador de salida.
- Amplificador diferencial.
- Filtro Pasa-Bajos analógico.
- Conversor Digital Analógico MCP 4921.
- Microcontrolador dsPIC33FJ128GP802.
- Fuente de alimentación.

El elemento más importante de esta tarjeta es el microcontrolador dsPIC33FJ128GP802 que por medio de sus módulos permiten comunicarse con el exterior y realizar diversas acciones de control, los principales implementados son: ADC (Conversor Analógico Digital) que permite convertir una señal analógica a una señal digital de salida para ser procesada, el DAC (Conversor Digital Analógico) que permite convertir los datos digitales a una señal analógica que será enviada al amplificador de salida para ser escuchada por medio de los altavoces, SPI (Interfaz Serial Periférica) comunica el microcontrolador con el Conversor Digital Analógico externo MCP 4921, mediante esta interfaz se comunica el dsPIC con la memoria flash y UART (Transmisión y Receptor Asíncrono Universal) comunica la tarjeta entrenadora y el programa terminal en el computador. Las etapas del entrenador de audio se ilustran en la Figura 29 y la tarjeta de desarrollo final se observa en la Figura 30.

Figura 29. Diagrama de Bloques Entrenador de Audio



Fuente: Autor del texto.

Figura 30. Entrenador de Audio

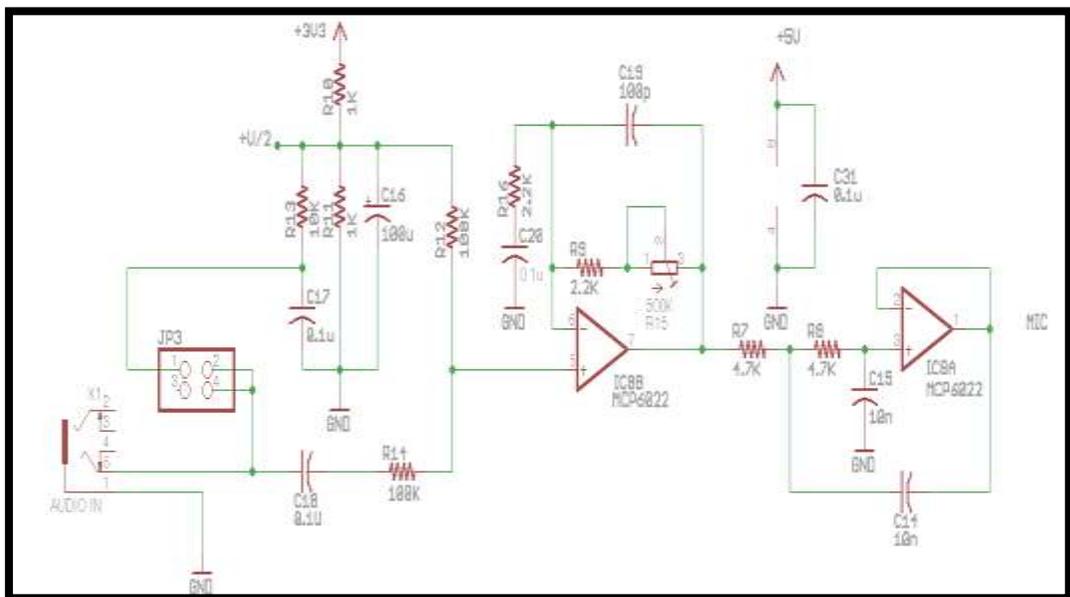


Fuente: Autor del texto.

3.1. PRE-AMPLIFICADOR DE ENTRADA AUXILIAR

Este circuito tiene como función principal acoplar la señal de audio de entrada a los niveles óptimos para ser procesada por el dsPIC, para esto cuenta con un control de ganancia y un realce en las frecuencias de audio entre 100Hz y 10KHz donde se encuentra el espectro de las señales de la voz humana. Como se puede observar en la Figura 31.

Figura 31. Plano esquemático Pre-amplificador.



Fuente: Autor del texto.

La implementación se basa en el amplificador operacional MCP6022 de la empresa Microchip debido a que se puede alimentar con fuente sencilla desde 2.5 a 5.5 voltios y tiene la característica Rail-to-Rail que permite que la excursión del amplificador este entre los niveles Vcc y Gnd.

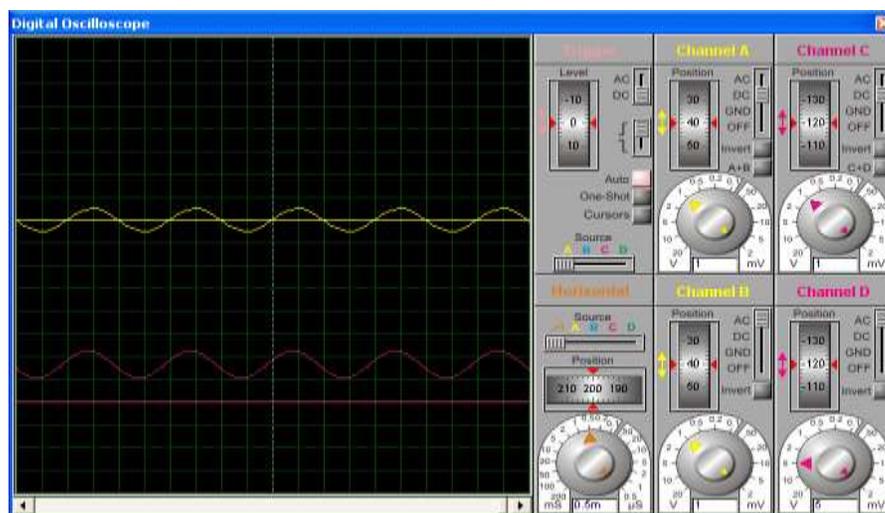
La señal de audio es introducida por medio del conector X1 (Audio In) y pasa al selector de entrada que permite escoger el acople para dos diferentes tipos de señal: micrófono o señal auxiliar (reproductor de audio MP3 o similar). La diferencia es que mientras la señal auxiliar se lleva directamente a la entrada del pre-amplificador, la señal de un micrófono se debe polarizar para que éste funcione adecuadamente, mediante el jumper JP3 se logra polarizar.

La siguiente etapa la compone un amplificador sumador que garantiza a la señal de entrada ser ubicarla sobre una componente DC de 1.65V y la multiplicada por un factor de ganancia seleccionado por el potenciómetro R15. Para la amplificación selectiva de frecuencias de audio se instaló la red RC conformada por R16 y C20, la cual varía su impedancia dependiendo de la frecuencia de la señal de entrada variando a su vez la ganancia del amplificador sumador.

Por último la señal pasa a través de un filtro pasa-bajos que funciona como filtro anti-aliasing para anular las señales que están por encima de las señales de audiofrecuencia.

La figura 32 muestra el comportamiento del pre-amplificador en el tiempo y en ella se aprecia cómo la señal de entrada que variaba entre -0.5 y 0.5 voltios, se convierte en una señal variable entre 3.3 y 1.0 voltios.

Figura 32. Comportamiento del pre-amplificador $f_{in}=500\text{Hz}$ $G\approx 2$



Fuente: Imagen tomada por el autor del texto usando el simulador de circuitos PROTEUS

A continuación se expone la figura 33 que describe el comportamiento del pre-amplificador en frecuencia con la máxima ganancia ($R_{15}=500K$) la cual produce una ganancia de 20dB sobre la señal de entrada. También es visible la amplificación selectiva de las frecuencias de audio y la incidencia del filtro anti-aliasing en el corte de las frecuencias altas.

Figura 33. Comportamiento en frecuencia del pre-amplificador



Fuente: Imagen tomada por el autor del texto usando el simulador de circuitos PROTEUS

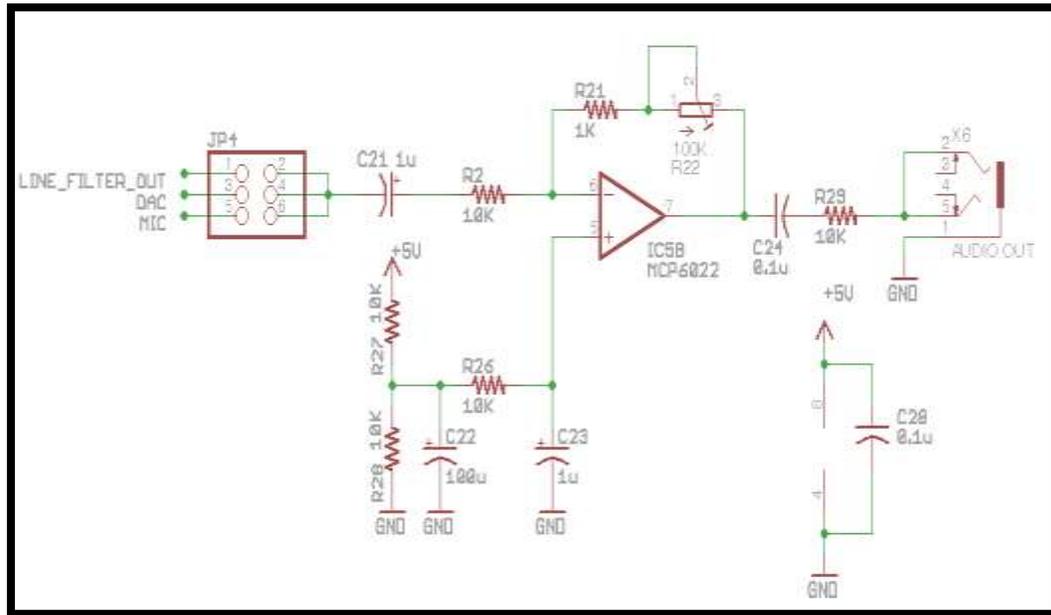
3.2. AMPLIFICADOR DE SALIDA

El amplificador de salida (figura 34) tiene como objetivo suministrar a los parlantes una señal con buena potencia para que esta pueda ser reproducida. Al igual que la etapa anterior, se desarrolló con un amplificador operacional MCP6022, esta vez configurado como amplificador restador. Cuenta con un control de ganancia y un selector de la señal de entrada por medio de un jumper; de esta manera se puede comparar en tiempo real las señales provenientes del pre-amplificador, salida del filtro análogo o la salida del convertor Analógico Digital del dsPIC.

Una vez se ha seleccionado la fuente de entrada por medio del jumper JP4, la señal entra al amplificador restador, el cual resta de un nivel de DC la señal de entrada para garantizar que la señal de salida varíe alrededor del nivel de DC para un máximo de 0 a 5 Voltios. El control de ganancia es fijado por la resistencia R22, la señal de salida se acopla por medio de la red C24, R29 y está disponible como salida en un conector de audio estéreo de 35mm X6 para que pueda ser fácilmente adaptada al sistema externo de amplificación determinado en el actual

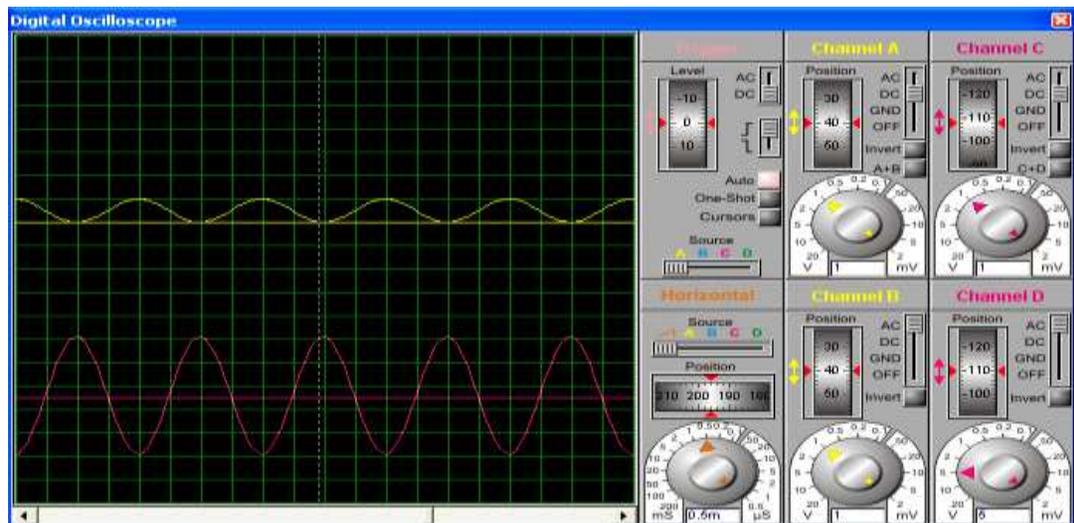
proyecto como parlante tipo desktop. En la figura 35 se aprecia el comportamiento de las señales en el amplificador de salida.

Figura 34. Plano esquemático amplificador de salida



Fuente: Autor del texto.

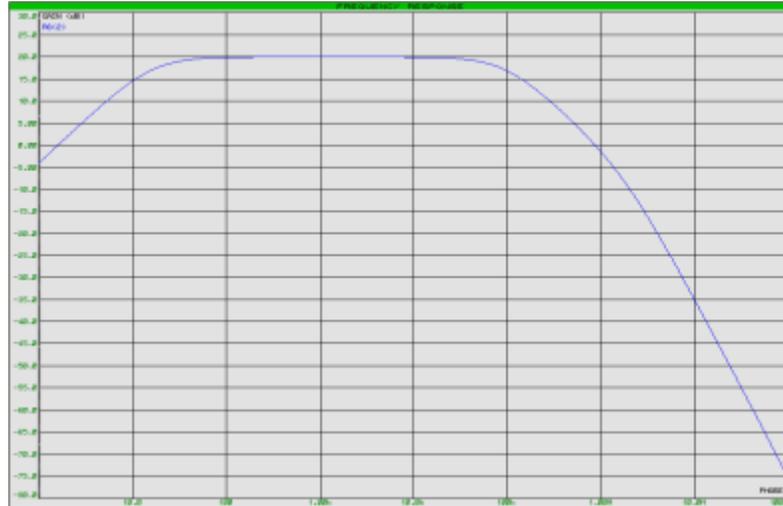
Figura 35. Comportamiento del Amplificador de Salida $f_{in}=500\text{Hz}$ $G \approx 5$



Fuente: Imagen tomada por el autor del texto usando el simulador de circuitos PROTEUS

El comportamiento en frecuencia (figura 36) muestra cómo toda la gama de señales de audio son amplificadas de igual manera. Cuando R22 es 100K, se genera una ganancia de 20dB sobre la señal de entrada.

Figura 36. Comportamiento en frecuencia del Amplificador de Salida



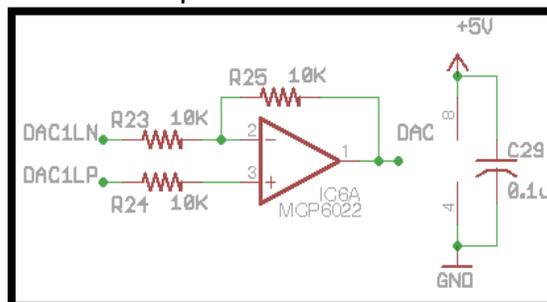
Fuente: Imagen tomada por el autor del texto usando el simulador de circuitos PROTEUS

3.3. AMPLIFICADOR DIFERENCIAL

Debido a la naturaleza de la señal que arroja el convertor Analógico Digital del dsPIC33F, se diseñó esta etapa, la cual tiene como objetivo convertir la señal diferencial en una señal común con referencia GND.

El amplificador operacional (figura 37) tiene una configuración típica de un restador con ganancia unitaria, la señal inversora se resta de la no inversora y se multiplica por la unidad.

Figura 37. Plano esquemático amplificador diferencial

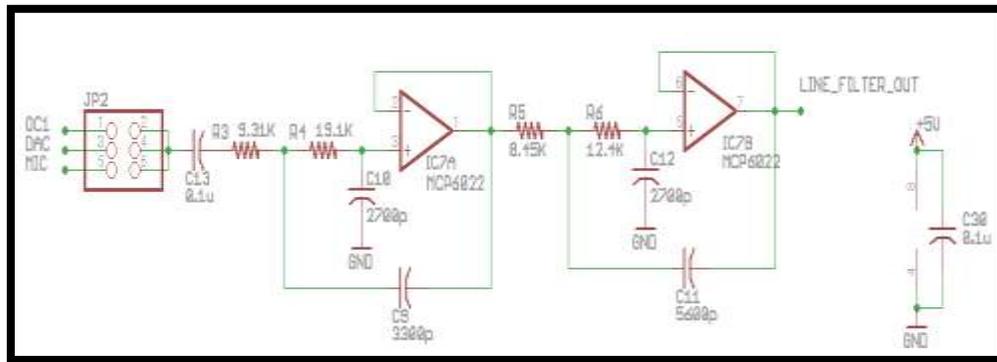


Fuente: Autor del texto.

3.4. FILTRO PASA-BAJOS ANALÓGICO

El amplificador pasa-bajos se diseñó pensando en ofrecer más utilidad al entrenador. Es posible implementarlo como filtro previo al muestreo de datos, filtro posterior a la salida analógica del dsPIC y especialmente como filtro para una señal modulada en ADPCM por medio del módulo PWM del dsPIC. Estas señales pueden configurarse como entrada del filtro configurando el jumper de entrada JP2. Figura 38.

Figura 38. Plano esquemático Filtro Pasa-bajos



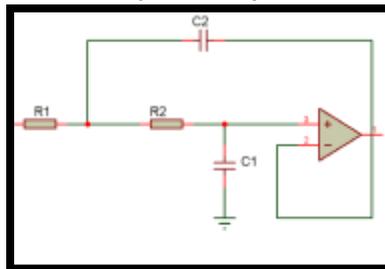
Fuente: Autor del texto.

Las características técnicas del filtro se escogieron tomando como base su función principal, convertir la señal de salida PWM en una señal análoga que represente el sonido. Por esto la frecuencia de corte se eligió en 4KHz y el orden del filtro en 4.

Para el diseño se usó la configuración Sallen-Key (figura 39) que usa 2 amplificadores operacionales para un filtro de orden 4. La frecuencia de corte para un filtro Sallen-Key pasa-bajo está dada por la formula 3.1:

$$F_c = \frac{1}{2 * \pi * \sqrt{R_1 * C_1 * R_2 * C_2}} \quad (3.1)$$

Figura 39. Configuración de un filtro pasa-bajos Sallen Key Orden 2

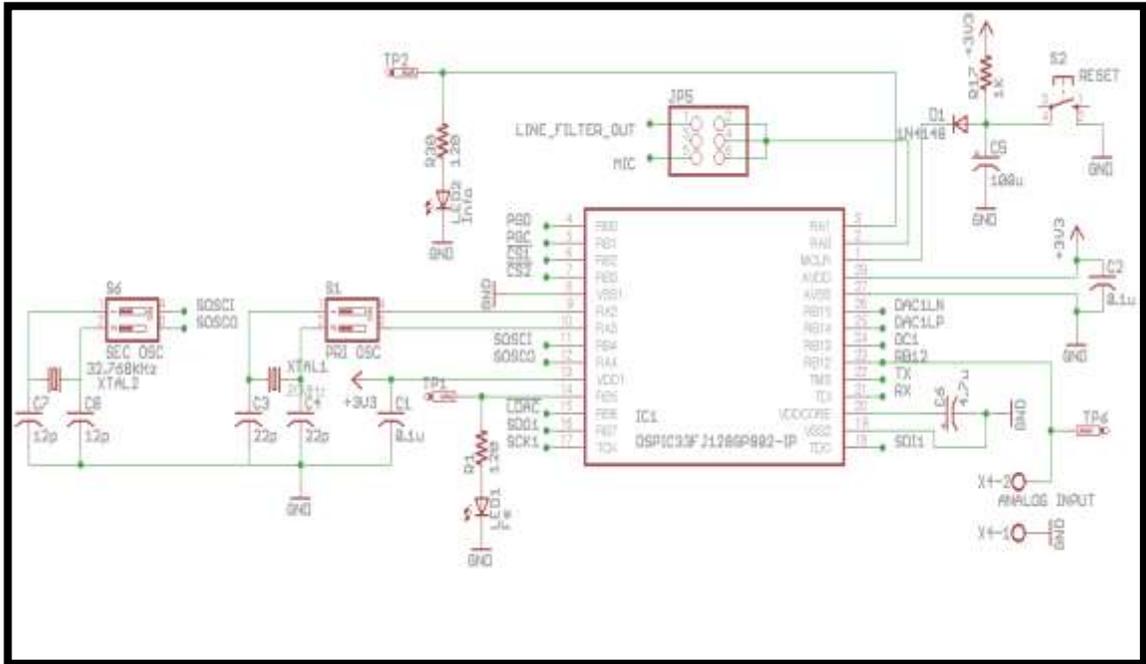


Fuente: Autor del texto.

3.6. MICROCONTROLADOR dsPIC33FJ128GP802

Esta parte de la tarjeta electrónica (figura 42) contiene el microcontrolador propiamente dicho (descrito en el capítulo 1), los componentes electrónicos auxiliares y el conexionado con los módulos y dispositivos de salida.

Figura 42. Diagrama esquemático del dsPIC33F en la tarjeta entrenadora



Fuente: Autor del texto.

Los componentes electrónicos y elementos auxiliares para el funcionamiento de esta parte de la tarjeta se describen a continuación:

3.6.1. Oscilador. Consiste en un cristal de 20MHz que puede ser usado como oscilador primario activando el dip-switch marcado como S1.

3.6.2. Oscilador Secundario. Es un cristal de 32.768KHz seleccionable por medio del dip-switch S6 para que se puedan desarrollar prácticas con el módulo reloj de tiempo real (RTCC) del dsPIC33F.

3.6.3. Selección de entrada análoga. Mediante el jumper JP5 es posible elegir qué fuente se desea usar como señal de entrada para el dsPIC. Las posibles opciones son: Salida del pre-amplificador o salida del filtro análogo.

3.6.4. Condensador estabilizador del Voltaje de núcleo. Para estabilizar el voltaje interno del microcontrolador que es una protección de detección de bajo voltaje, se instaló un condensador de $4.7\mu/16V$ (C6), sugerido por el fabricante, en el pin 20 del dsPIC marcado como VCAP o VDDCORE.

A continuación se describe el conexionado con los módulos y dispositivos de salida, los cuales varios de ellos están marcados en la tarjeta electrónica como secciones diferentes a la del microcontrolador:

3.6.5. Circuito de Reset. Es muy importante tener un buen diseño del circuito de Reset o MCLR, para ello en el entrenador se usó el circuito sugerido por el fabricante, el cual garantiza que el dsPIC no se reiniciará por falsos picos bajos en el voltaje de alimentación y una adecuada estabilización al momento de encendido del sistema.

3.6.6. Indicadores o Testigos: En la parte inferior de la tarjeta se encuentran dos LEDs (LED1 y LED2) los cuales están conectados a los pines RB5 y RA1. En el desarrollo de los códigos de ejemplo se usó el pin RB5 como testigo de la frecuencia de muestreo. Estos pines logran ser visualizados mediante el osciloscopio gracias a que han sido ubicados también como puntos de prueba (TP1 y TP2) que logran ser fácilmente accesibles con la punta de prueba del instrumento de medida.

3.6.7. Conexión ICSP o interfaz para PICKIT2. Se implementó una conexión siguiendo el estándar ICSP de Microchip para lograr interconectar un programador PICKIT2, PICKIT3 o similar, esta conexión permite también usar la opción de debugger de estos programadores.

3.6.8. Puerto SPI. El módulo de comunicación SPI del dsPIC lo conforman el SPI2 y el SPI1, los cuales están conectados a su vez en la tarjeta diseñada a dos dispositivos externos: el SPI2 al conversor Digital Analógico MCP4921 (descrito en el numeral 3.5) y el SPI1 a la memoria Flash la cual, se encuentra en la tarjeta electrónica como IC10, siendo un circuito integrado con 8 pines de montaje superficial. El conversor Digital Analógico es el que permitirá realizar prácticas de control digital y filtros digitales aplicados a señales diferentes a las de audio. La memoria Flash de 8MB permitirá realizar prácticas para el almacenamiento de señales de manera digital.

3.6.9. Puerto USART. El módulo de comunicación USART del dsPIC está conectado a un convertidor serie-USB comercial ubicado en la tarjeta diseñada como Interfaz USB. Por medio de éste se puede establecer una comunicación entre la tarjeta entrenadora y el computador mediante una aplicación para intercambio de información serial.

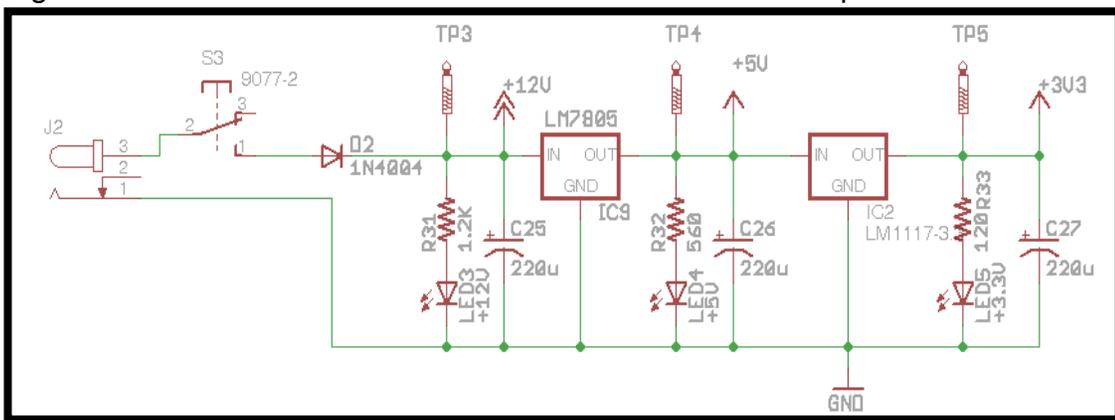
3.6.10. Salida PWM. La salida PWM del dsPIC está como opción de entrada para el filtro pasa-bajos analógico (numeral 3.4). Así se podrían realizar prácticas que involucren la modulación ADPCM y la reproducción de muestras de audio usando el PWM.

3.7. FUENTE DE ALIMENTACIÓN

La fuente de alimentación (figura 43) que se diseñó para el entrenador utiliza reguladores de voltaje de precisión que están desarrollados con la tecnología LDO (Low drop-out) que significa bajo nivel de voltaje de diferencia entre entrada y salida, la cual ofrece ventajas en cuanto al voltaje de entrada para la regulación permitiendo que con un voltaje ligeramente mayor al de regulación puedan empezar a trabajar. También mejora el consumo de corriente interno del regulador minimizando la disipación de potencia con respecto a un regulador lineal.

La entrada de corriente es suministrada por un adaptador externo de corriente de 12 Voltios a 1 Amperio el cual entra por el conector J2 y es interrumpido por el switch ON-OFF S3. Cuando el interruptor esta activo, la corriente pasa por el diodo D2 que actúa como protección de polarización. Ingresando al regulador de 5V LM7805V que va a servir como fuente estable para los amplificadores operacionales, y entrada al regulador de 3 Voltios LM1117-3.3V la salida de este regulador será la fuente del dsPIC, la memoria FLASH y el MCP4921.

Figura 43. Fuente de alimentación del entrenador de audio para dsPIC

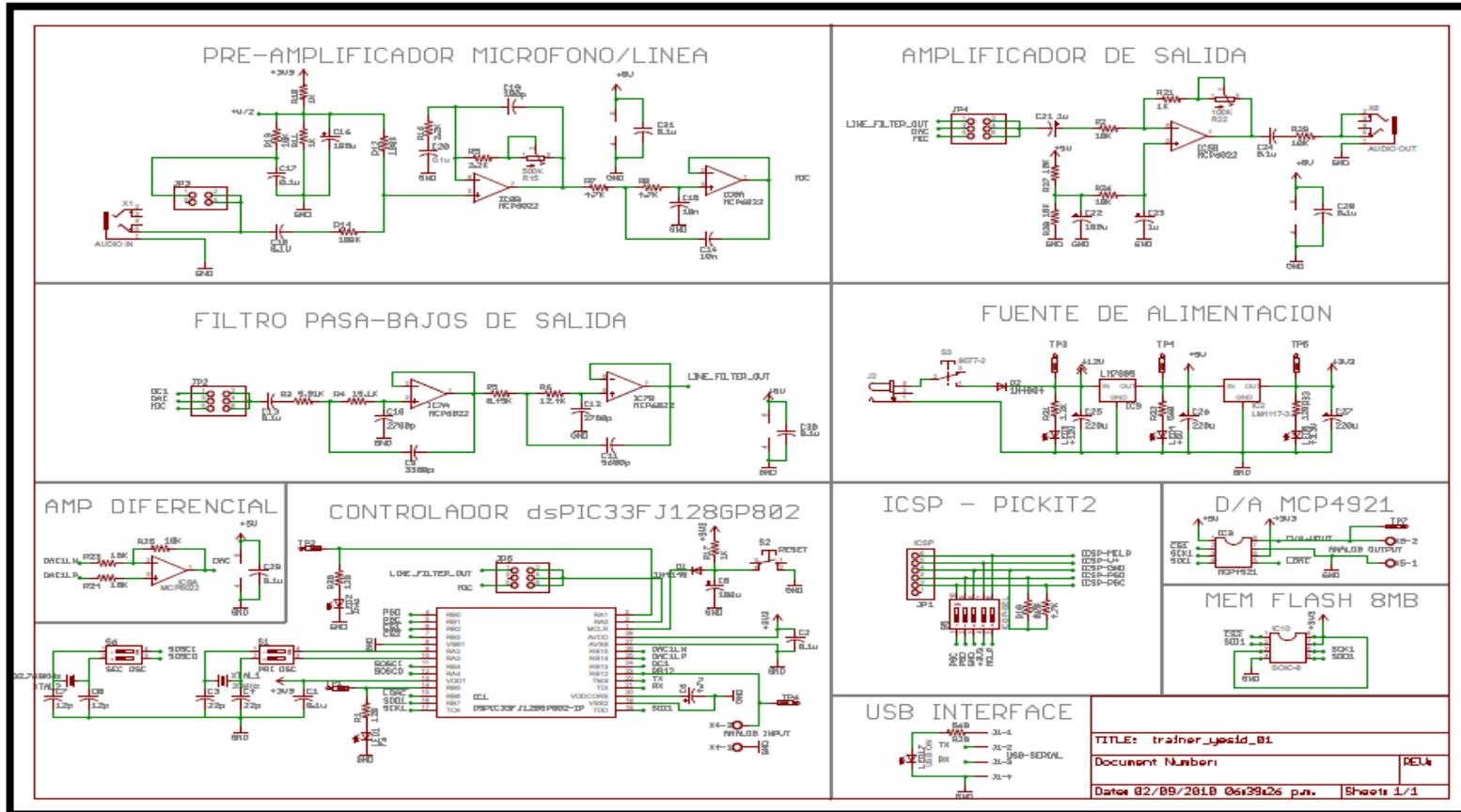


Fuente: Autor del texto.

Los condensadores C25, C26 Y C27 estabilizan el voltaje de salida de cada regulador. Cada nivel de voltaje tiene un LED testigo que dará una indicación visual que el voltaje está presente en la tarjeta.

La figura 44 muestra el diseño esquemático completo de la tarjeta electrónica diseñada.

Figura 44 . Plano Esquemático Entrenador de Audio



Fuente: Autor del texto.

4. MANUAL DE USUARIO DEL ENTRENADOR DE AUDIO dsPIC33FJ128GP802

El Manual de Usuario del Entrenador de Audio dsPIC33fj128GP802 proporciona un ambiente en el cual se muestra la información necesaria a los usuarios, qué se deben tener en cuenta antes y durante la utilización del entrenador, además un enfoque práctico de la utilización de C30 de Microchip y MikroC de Mikroelektronika en la implementación de filtros digitales. A pesar de ser una parte del libro puede ser usado de manera independiente del mismo como material de apoyo en el manejo del Entrenador de Audio.

El presente capítulo lo componen tres parte principales precedidas de unas recomendaciones necesarias para el inicio con el manejo de la tarjeta. La primera parte es un previo reconocimiento del módulo en donde se observa los diferentes tipos de configuración y cómo esta comprendido; para así posteriormente realizar la metodología en la aplicación de filtros digitales mediante las herramientas C30 de Microchip y MikroC de Mikroelektronika. La segunda parte es una guía para el uso del compilador MPLAB C30 de Microchip y dsPIC FD LITE los cuales, son de gran utilidad para las aplicaciones posteriormente planteadas. Finalmente, la tercera parte se propone una guía de uso de MikroC de Mikroelektronika, utilizando la herramienta Filter designer tools.

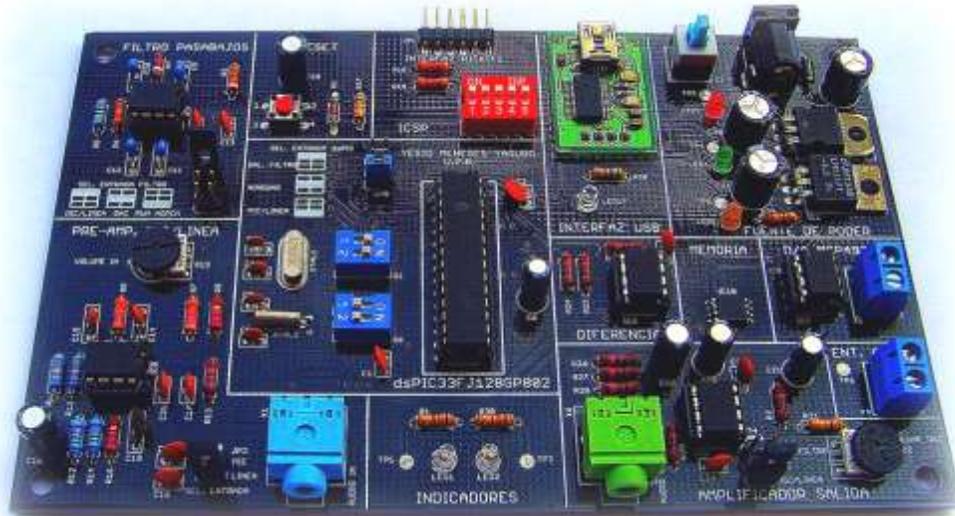
Recomendaciones generales a la hora de conexiones del módulo:

- Antes de conectar la tarjeta de desarrollo, es necesario instalar los controladores para el funcionamiento del programador; en donde es necesario instalar el programa del PICKIT2 para cargar el código con extensión .hex en el microcontrolador, esto se hace cuando se desea trabajar con MikroC de Mikroelektronika.
- Al encender el sistema de desarrollo se debe tener en cuenta que el rango de voltaje de alimentación de entrada debe estar entre 6VDC y 12VDC. Una vez asegurado el voltaje de alimentación se enciende mediante el interruptor correspondiente. Los diodos LED se encienden automáticamente e indica que la fuente diseñada está operando para su uso.
- Configurar el generador de señales analógicas de entradas la cual deben trabajar a un máximo de 3.3 Vpp positivos; lo que indica que debe tener un OFFSET de 1.5 Voltios y 1.5 Voltios pico de amplitud la señal como valor máximo.

4.1. RECONOCIMIENTO DEL ENTRENADOR DE AUDIO

El módulo entrenador de audio que se muestra en la figura 45, es una tarjeta para el desarrollo de aplicaciones de hardware y software basado en el microcontrolador dsPIC33FJ128GP802, el cual es muy versátil a la hora de ser implementado, sirve como herramienta de aplicación de filtros digitales.

Figura 45. Entrenador de audio dsPIC33FJ128GP802



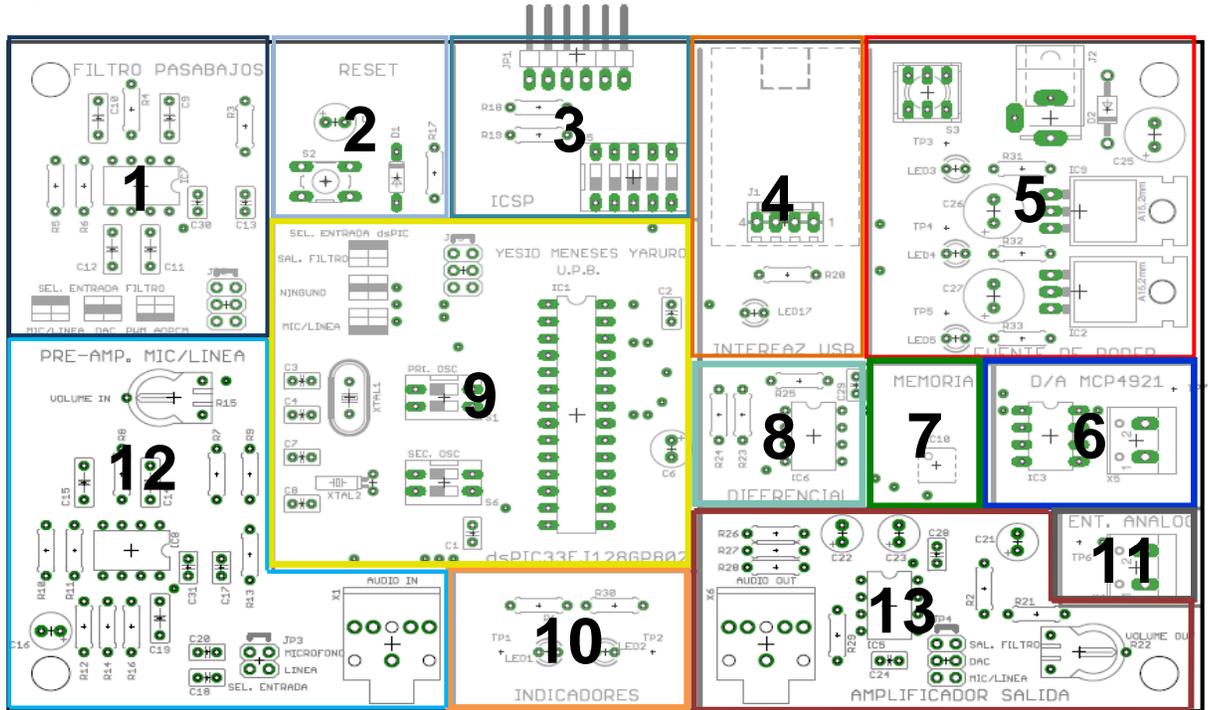
Fuente: El autor del texto.

A continuación se ilustra la división de la tarjeta en cada una de sus partes, para identificar su ubicación físicamente y estas partes serán descritas explicando su funcionamiento.

Las secciones principales del Entrenador de Audio que se muestran en la figura y siguiendo su numeración son:

1. Filtro pasa-bajos.
2. Reset.
3. Interfaz PICKIT2 -ICSP-
4. Interfaz USB.
5. Fuente de alimentación.
6. D/A MCP4921.
7. Memoria.
8. Diferencial.
9. Selección de entradas y osciladores.
10. Testigos.
11. Entrada analógica.
12. Pre-Amplificador de entrada.
13. Amplificador de salida.

Figura 46. Diagrama General del Entrenador de Audio



Fuente: El autor del texto

4.1.1. Filtro pasa-bajos. Se utiliza en conjunto para codificación ADPCM por medio del módulo PWM del dsPIC. También puede ser utilizado en cascada con el filtro digital del microcontrolador ya sea previo al muestreo de datos o posterior a la salida analógica del dsPIC. Estas señales pueden configurarse como entrada del filtro configurando el jumper de entrada JP2.

Figura 47. Diagrama General del Entrenador de Audio Filtro Pasa-Bajos



Fuente: El autor del texto.

4.1.2. Reset. Para el diseño del entrenador se empleó el circuito sugerido por el fabricante. El pulsador normalmente abierto rojo es quien activa o permite el valor lógico bajo en el pin de reset del microcontrolador generándose de esta manera la condición de reset externa, dando la reiniciación del sistema.

Figura 48. Diagrama General del Entrenador de Audio – Reset



Fuente: El autor del texto.

4.1.3. Interfaz PICKIT2 -ICSP-. La conexión sigue el estándar ICSP de Microchip para conectar el programador, el estado de los dip-switch debe ser on.

Figura 49. Diagrama General del Entrenador de Audio – Interfaz PICKIT2-ICSP.



Fuente: El autor del texto.

4.1.4. Interfaz USB. Realiza una comunicación serial con un computador al que se encuentre conectado. Se envían los datos desde el computador, y el sistema se configura para ejecutar un filtro deseado.

Figura 54. Diagrama General del Entrenador de Audio – Diferencial.



Fuente: El autor del texto.

4.1.9. Selección de entradas y osciladores. Mediante el jumper JP5 es posible elegir qué fuente se desea usar como señal de entrada para el dsPIC. Las posibles opciones son: Salida del pre-amplificador ó salida del filtro análogo. Se selecciona el oscilador primario de 20Mhz activando el dip-switch marcado como S1. Se selecciona el oscilador secundario de 32.768KHz activando el dip-switch S6 para que se puedan hacer prácticas con el módulo reloj de tiempo real (RTCC) del dsPIC33F.

Figura 55. Diagrama General del Entrenador de Audio – Selección de entradas y audio.



Fuente: El autor del texto.

4.1.10. Indicadores. Se han dejado dos LEDs conectados a los pines RB5 y RA1. En el desarrollo de los códigos de ejemplo se usó el pin RB5 como testigo de la frecuencia de muestreo.

Figura 56. Diagrama General del Entrenador de Audio – Indicadores.



Fuente: El autor del texto.

4.1.11. Entrada analógica. Esta entrada se utiliza en forma adicional si se desea que la fuente a muestrear sea diferente a audio. **ADVERTENCIA:** Cualquier señal introducida en esta entrada debe tener un OFFSET de 1.5 Voltios, y amplitud de 1.5 Voltios pico (es decir, debe encontrarse en el rango de 0 a 3,3 Voltios) de lo contrario el microcontrolador puede sufrir daños.

Figura 57. Diagrama General del Entrenador de Audio – Entradas analógicas.



Fuente: El autor del texto.

4.1.12. Pre-amplificador y micrófono. Adecuación de la entrada de la señal de audio o micrófono con un Pre-amplificador.

Figura 58. Diagrama General del Entrenador de Audio – Pre-amplificador.



Fuente: El autor del texto.

4.1.13. Amplificador de salida. Este amplificador tiene la posibilidad de seleccionar la señal que se desea amplificar y ser llevada al exterior, mediante su jumper respectivo.

Figura 59. Diagrama General del Entrenador de Audio – Amplificador de salida.



Fuente: El autor del texto.

4.2. COMPILADOR C30 DE MICROCHIP

4.2.1. Procedimiento para utilizar Mplab C30 – Microchip. Para realizar un proyecto en el cual se utilice el compilador MPLAB C30 se deben seguir una serie de pasos que se ilustraran con detalle a continuación:

1. Nuevo Proyecto: Se ingresa a MPLAB IDE, para crear un nuevo proyecto. Mediante la opción menú Project->Project Wizard en donde se iniciará un asistente con una serie de pasos para la creación del proyecto.

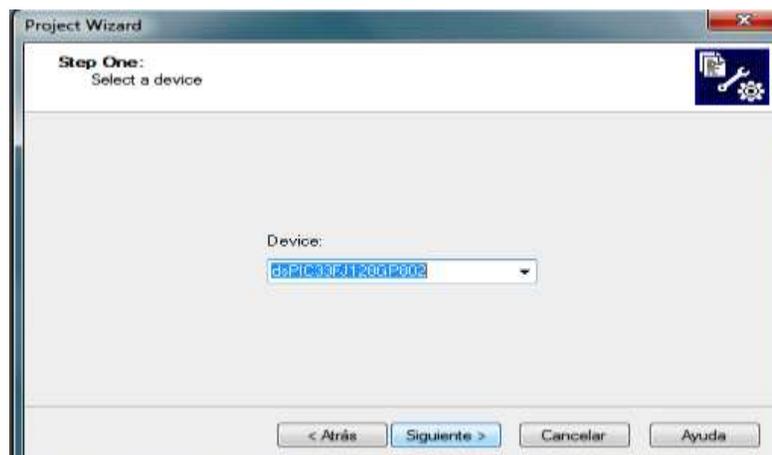
Figura 60. Asistente para Crear un Proyecto Mplab Idle



Fuente: Imagen tomada por el autor del texto

2. Elección del dispositivo. Se selecciona el tipo de microcontrolador que se va a utilizar, en éste caso el dsPIC33FJ128GP802.

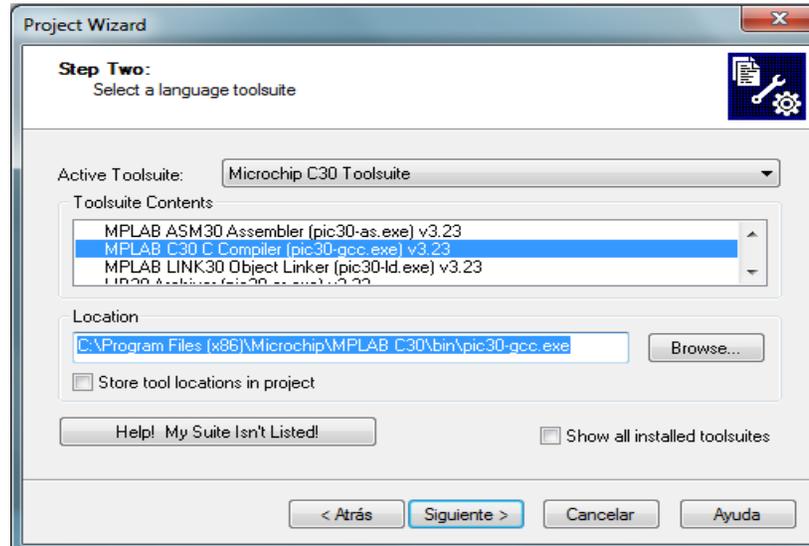
Figura 61. Selección del dispositivo Mplab Idle.



Fuente: Imagen tomada por el autor del texto

3. Elección del compilador. Se especifica el paquete de herramientas del lenguaje con que se va a trabajar, en éste caso se trabaja con el Compilador C30.

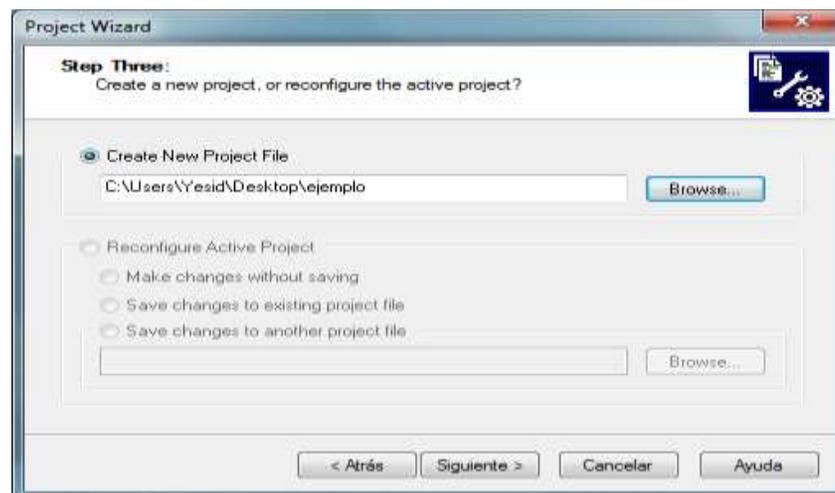
Figura 62. Selección del lenguaje a utilizar Mplab Idle.



Fuente: Imagen tomada por el autor del texto usando Mplab Idle

4. Especificaciones del proyecto. Se especifica el nombre y la ruta en donde se desea guardar el proyecto.

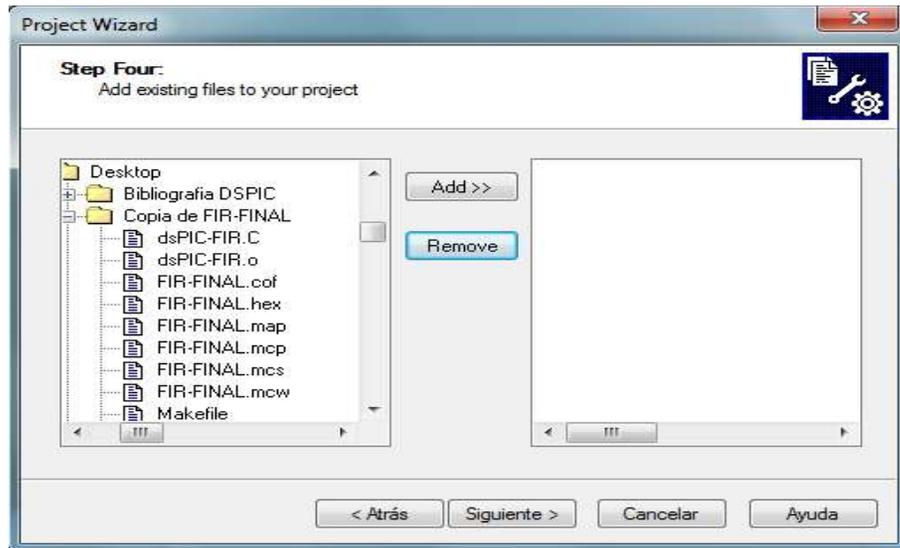
Figura 63. Asignación del nombre y ruta del proyecto Mplab Idle



Fuente: Imagen tomada por el autor del texto usando Mplab Idle

5. Selección de los ficheros del proyecto. Se especifica los archivos que contienen el proyecto y se adicionan para que sean cargados.

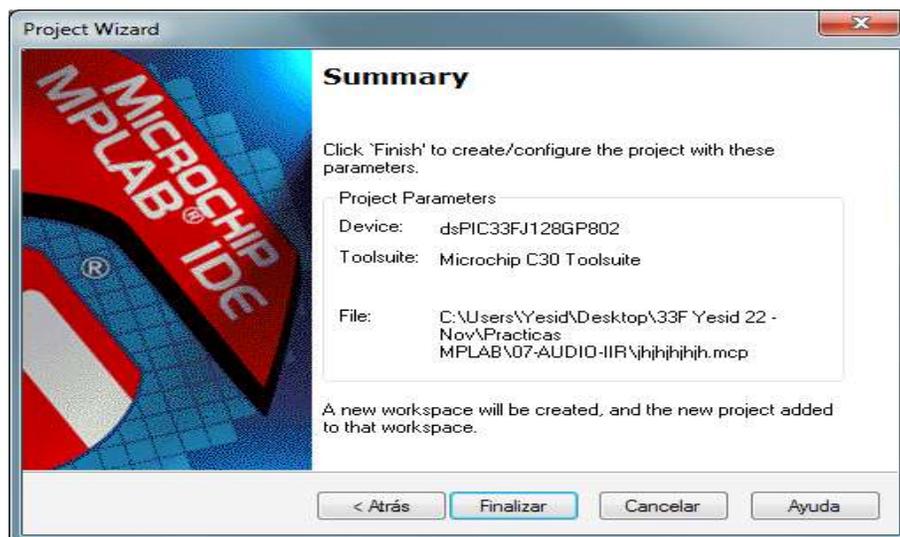
Figura 64. Selección de ficheros del proyecto Mplab Idle



Fuente: Imagen tomada por el autor del texto usando Mplab Idle

6. Resumen del proyecto que se ha creado

Figura 65. Selección Ventana finalización de creación de proyecto Mplab Idle

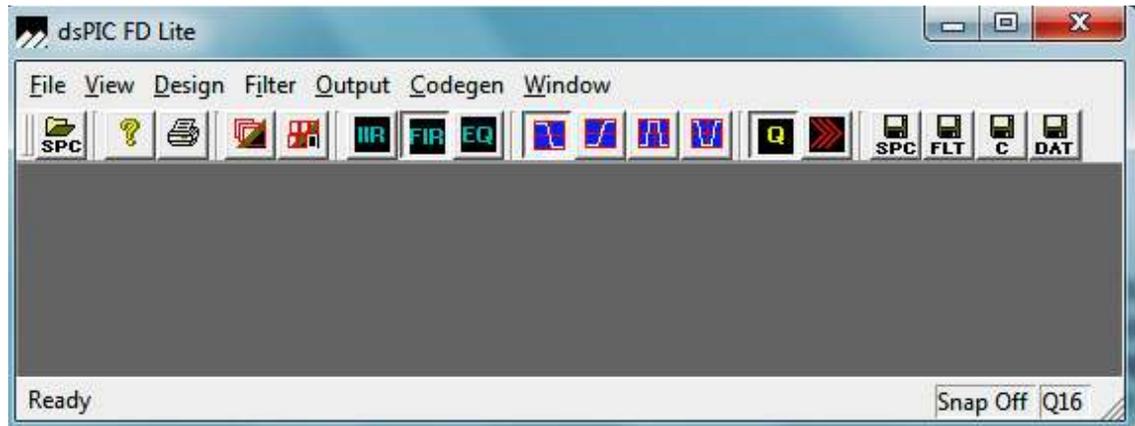


Fuente: Imagen tomada por el autor del texto usando Mplab Idle

4.2.1. Asistente para creación de coeficientes para C30 – dsPICFD LITE.

Se abre el dsPIC FD Lite

Figura 66. Software dsPICFD Lite



Fuente: Imagen tomada por el autor del texto usando dsPICFD Lite.

Este software dispone de un asistente para el usuario en la construcción de un filtro FIR e IIR. Para iniciar el proceso de diseño debe ir al menú Design en donde se puede elegir de tres tipos de filtros como son: IIR, FIR usando método de ventanas y EQUIRIPLE que es un filtro FIR que utiliza el método de Remez. Cada uno puede ser de diferente tipo (pasa-bajos, pasa-altos, pasa-banda, rechaza-banda) según las necesidades de diseño.

A continuación se darán a conocer una serie de ejemplos para la creación de filtros FIR e IIR.

- Filtro FIIR Pasa Bajo

Especificaciones: Frecuencia de Muestreo = 48KHz

Frecuencia de Corte = 800Hz

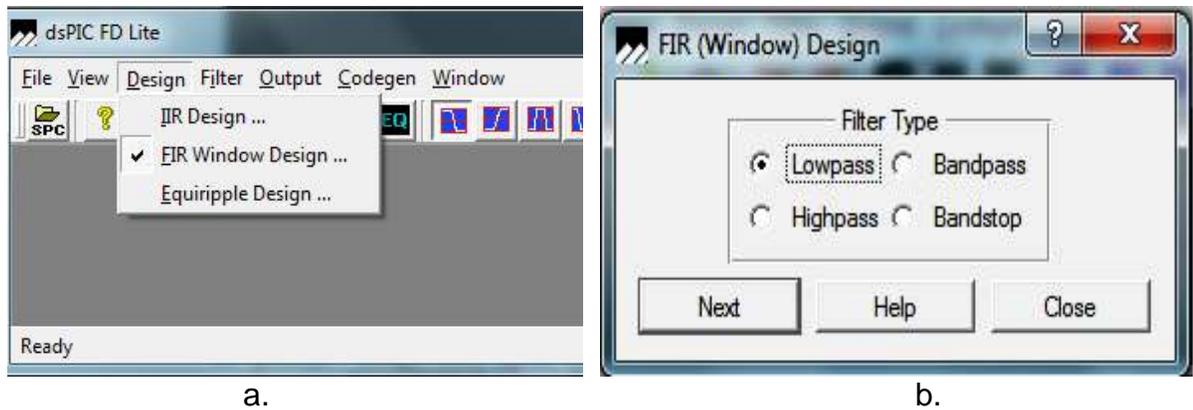
Frecuencia de Rechazo = 1.5KHz

Atenuación de Banda de Paso = 1dB

Atenuación de Banda de rechazo = 3dB

Con estas especificaciones se procede a calcular los coeficientes del Filtro FIR Pasa Baja, por medio del Programa dsPICLiteFD el cual es explicado en el capítulo uno.

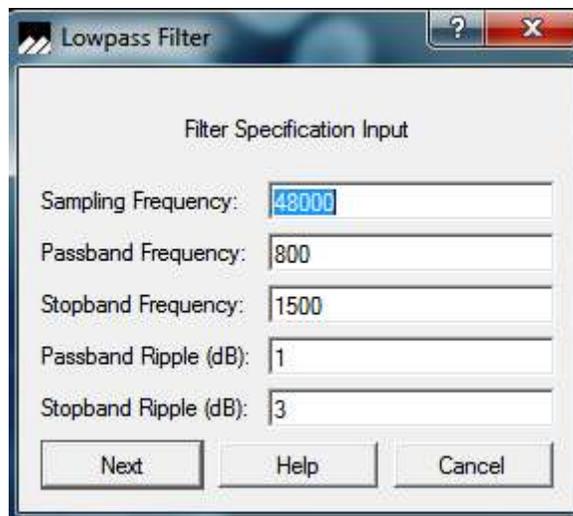
Figura 67. Ventana de Selección dsPICFD Lite filtro FIR. a. Diseño del Filtro, b. Tipo de Filtro.



Fuente: Imagen tomada por el autor del texto usando dsPICFD Lite.

A continuación se observa la ventana de características de las especificaciones del filtro, en donde se pueden configurar la frecuencia de muestreo, frecuencias de pasa banda, frecuencias rechaza banda, atenuación pasa banda y atenuación rechaza banda. Esto es aplicable para los tres tipos de filtros IIR, FIR e EQUIRIPLE, pasa-bajos, pasa-altos, pasa-banda y rechaza-banda.

Figura 68. Ventana de Especificación del Filtro dsPICFD Lite filtro FIR.



Fuente: Imagen tomada por el autor del texto usando dsPICFD Lite.

Al finalizar las especificaciones, se dispone a seleccionar el tipo de implementación del filtro FIR, donde se indica cual será el orden óptimo para el filtro.

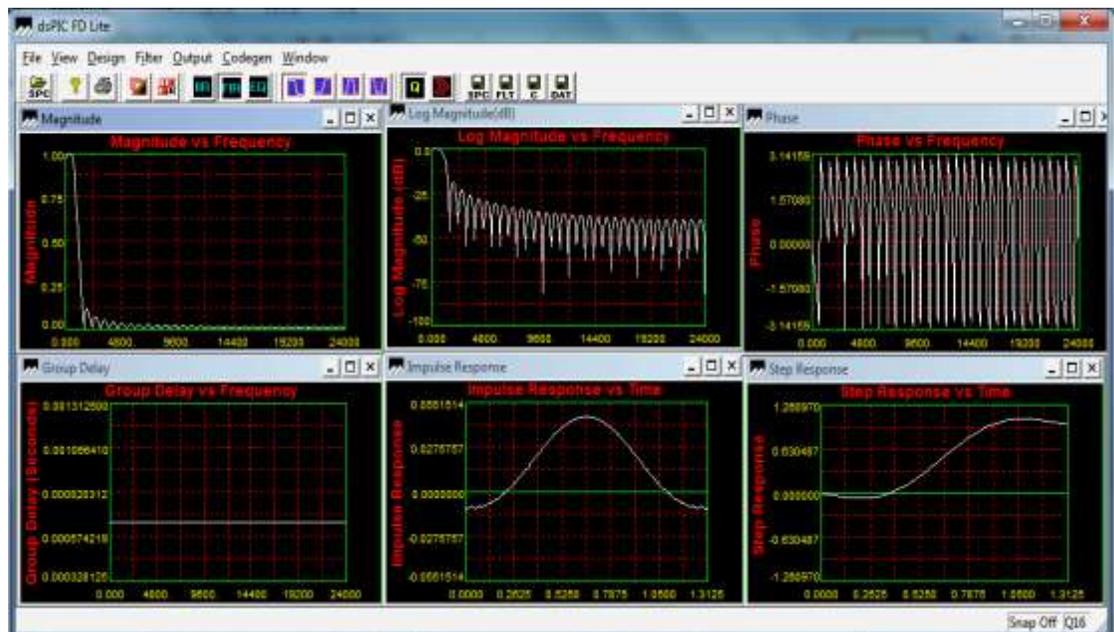
Figura 69. Tipo de Implementación del Filtro FIR dsPICFD Lite filtro FIR.



Fuente: Imagen tomada por el autor del texto usando dsPICFD Lite.

Una vez seleccionado el tipo de filtro, se finaliza el diseño en donde se observa gráficamente la respuesta en magnitud, respuesta al impulso, respuesta al escalón, fase y retardo de grupo, características muy importantes a la hora de implementar filtros.

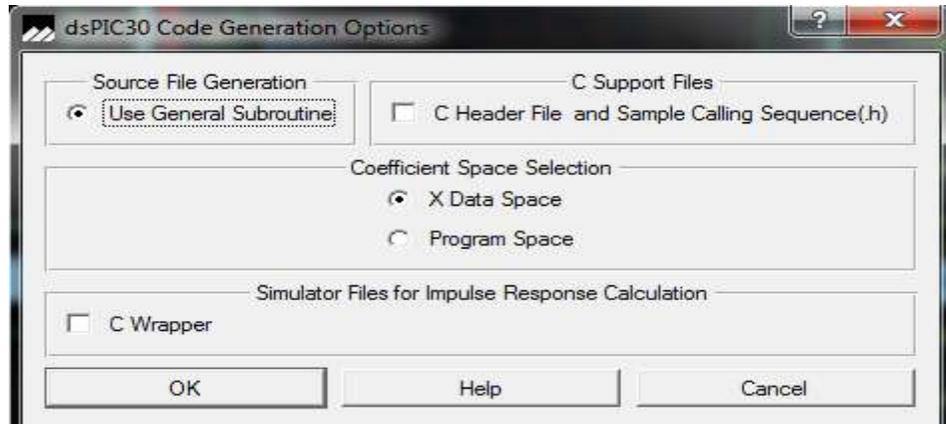
Figura 70. Análisis del Filtro dsPICFD Lite filtro FIR



Fuente: Imagen tomada por el autor del texto usando dsPICFD Lite.

Tras observar que el filtro FIR diseñado es el adecuado se procede a generar los coeficientes obtenidos en los parámetros anteriores; en donde los coeficientes son situados en la memoria X del microcontrolador.

Figura 71. Selección de la Ubicación de los coeficientes y Código Generado dsPICFD Lite filtro FIR.



Fuente: Imagen tomada por el autor del texto usando dsPICFD Lite.

Una vez generado el fichero de los coeficientes que contiene las rutinas necesarias para lograr ser implementado el proyecto en la plantilla del filtro FIR pasa-bajos. En este caso el fichero es llamado <<pasabajos.s>> que contiene las el orden del filtro y sus coeficientes.

- Filtro IIR Pasa- Banda

Especificaciones: Frecuencia de Muestreo = 48KHz

Frecuencia de Corte inferior= 1000Hz

Frecuencia de Corte superior= 1200Hz

Frecuencia de Rechazo inferior= 800Hz

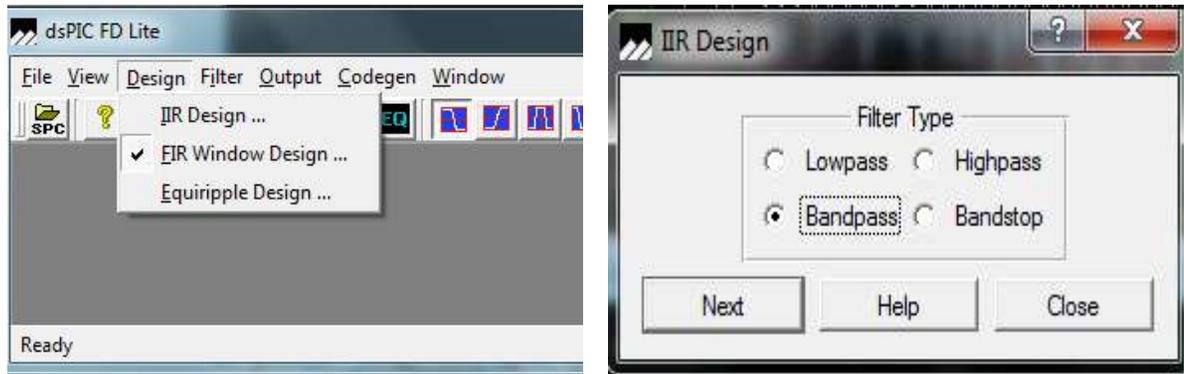
Frecuencia de Rechazo superior= 1.5KHz

Atenuación de Banda de Paso = 1dB

Atenuación de Banda de rechazo = 3dB

Para el diseño del filtro IIR se realiza un procedimiento muy parecido al diseño de los filtros FIR; dicho procedimiento se ve a continuación:

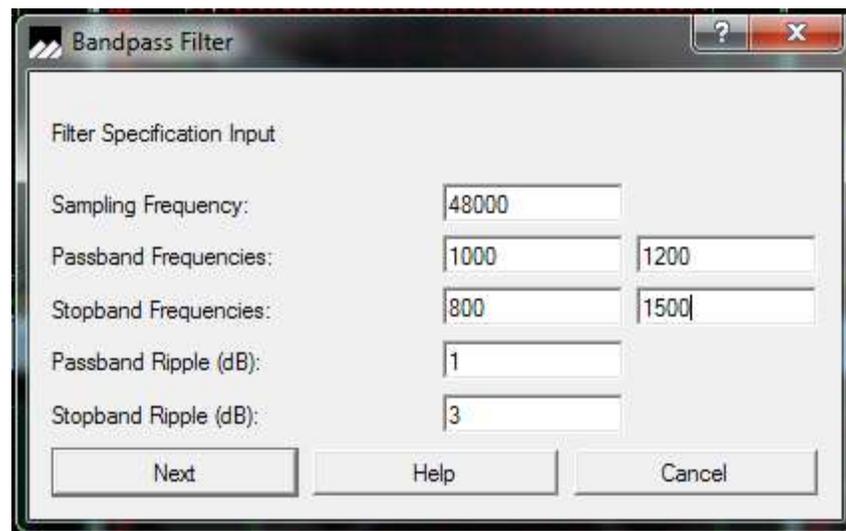
Figura 72. Ventana de Selección dsPICFD Lite filtro IIR. a. Diseño del Filtro, b. Tipo de Filtro.



Fuente: Imagen tomada por el autor del texto usando dsPICFD Lite.

A continuación se observa la ventana de características de las especificaciones del filtro IIR, en donde se configuraran los principales parámetros del filtro a diseñar.

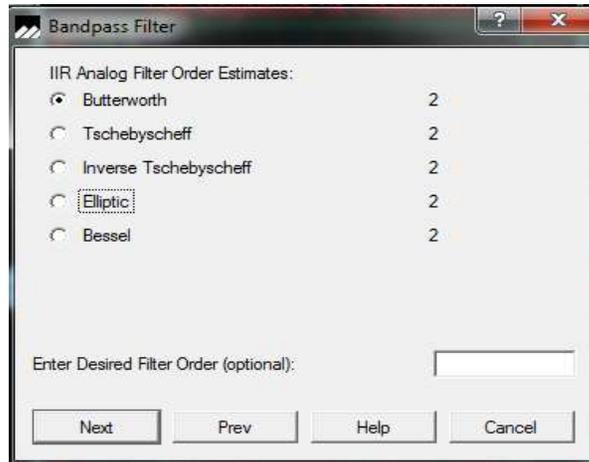
Figura 73. Ventana de Especificación del Filtro dsPICFD Lite filtro IIR.



Fuente: Imagen tomada por el autor del texto usando dsPICFD Lite.

Al finalizar las especificaciones, se dispone a seleccionar el tipo de implementación del filtro IIR, donde se indica cual será el orden óptimo para el filtro.

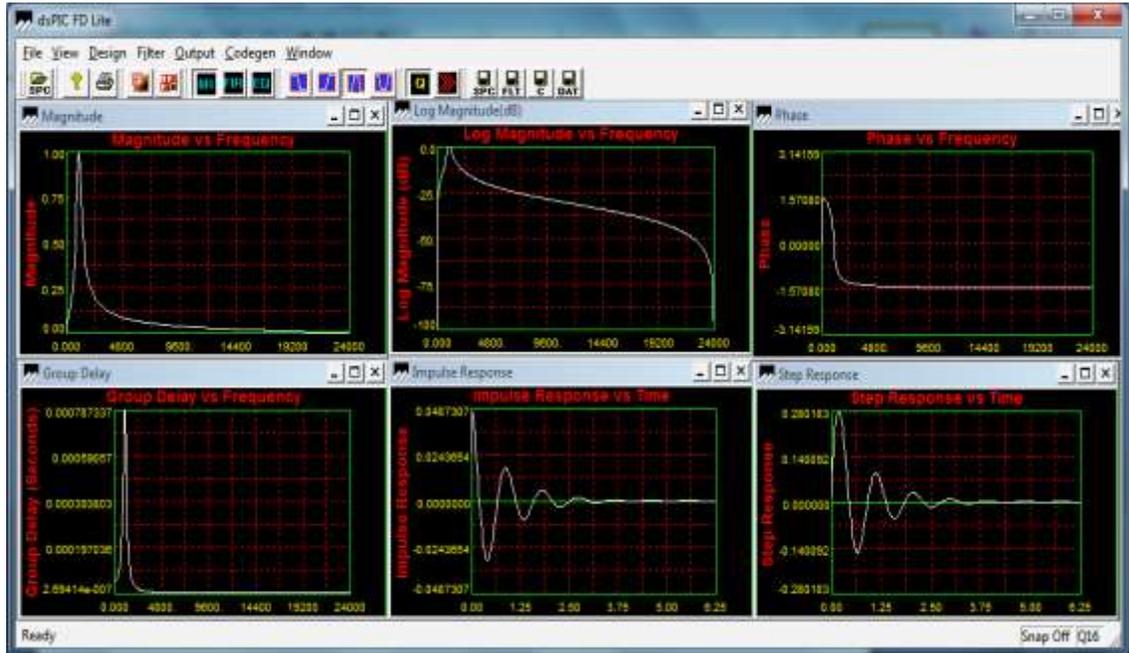
Figura 74. Tipo de Implementación del Filtro IIR dsPICFD Lite filtro IIR.



Fuente: Imagen tomada por el autor del texto usando dsPICFD Lite.

Una vez seleccionado el tipo de filtro, se finaliza el diseño en donde se observa gráficamente las características del filtro a implementar.

Figura 75. Análisis del Filtro dsPICFD Lite filtro IIR.



Fuente: Imagen tomada por el autor del texto usando dsPICFD Lite.

Luego de observar que el filtro IIR diseñado es el adecuado se procede a generar los coeficientes obtenidos en los parámetros anteriores; en donde los coeficientes son situados en la memoria X del microcontrolador.

Una vez generado el fichero de los coeficientes que contiene las rutinas necesarias para poder ser implementado el proyecto en la plantilla del filtro FIR pasa-bajos. En este caso el fichero es llamado <<pasabanda.s>> que contiene las el orden del filtro y sus coeficientes.

4.3. COMPILADOR MIKROC

4.3.1. Procedimiento para utilizar MikroC Pro for dsPIC –Mikroelektronika.

Para realizar un proyecto se deben seguir una serie de pasos que se ilustrarán con detalle a continuación:

1. Nuevo Proyecto: Mediante la opción menú Project-> New Project, se iniciara un asistente denominado Project Wizard con una serie de pasos para la creación del nuevo proyecto.

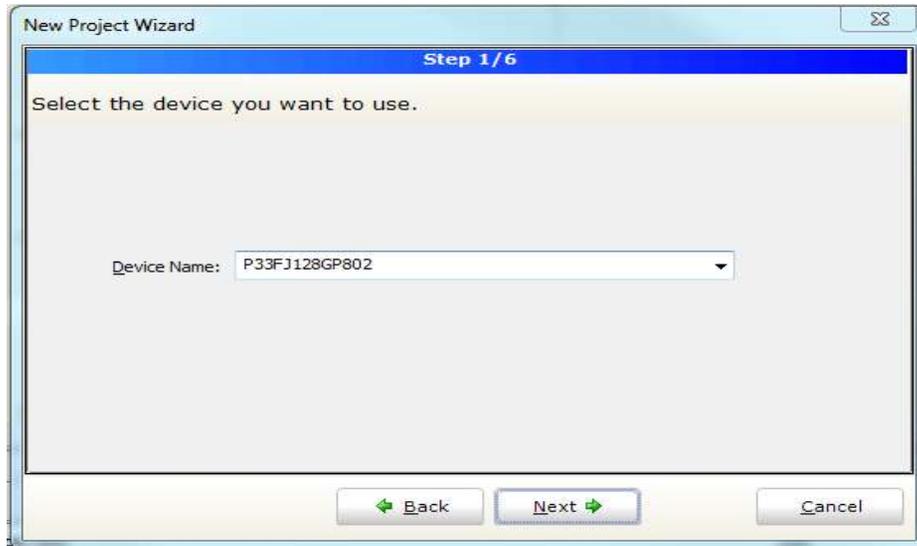
Figura 76. Ventana de aplicación nuevo proyecto Mikroelektronika.



Fuente: Imagen tomada por el autor del texto usando MikroC Pro for dsPIC.

2. Elección del dispositivo. Se selecciona el tipo de microcontrolador que vayamos a utilizar, en nuestro caso el dsPIC33FJ128GP802.

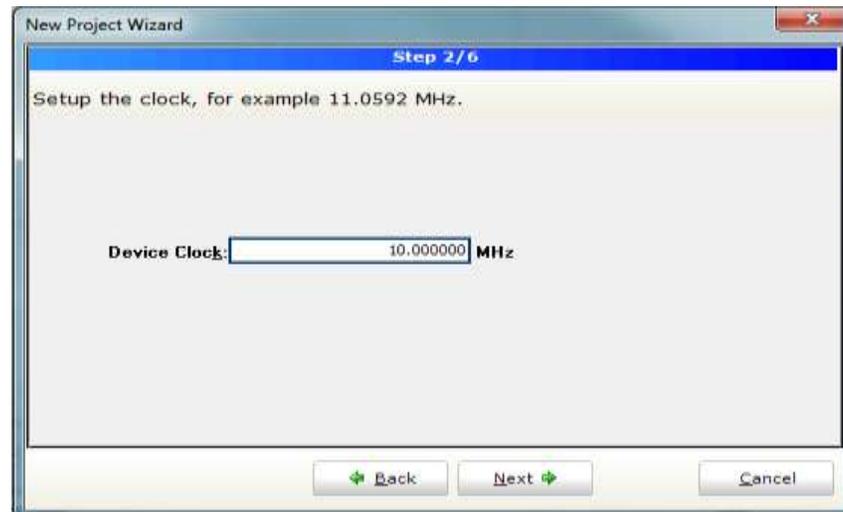
Figura 77. Interfaz de selección del dispositivo MikroC Pro for dsPIC.



Fuente: Imagen tomada por el autor del texto usando MikroC Pro for dsPIC.

3. Selección de la frecuencia de reloj. Se especifica la frecuencia de reloj que se va a utilizar.

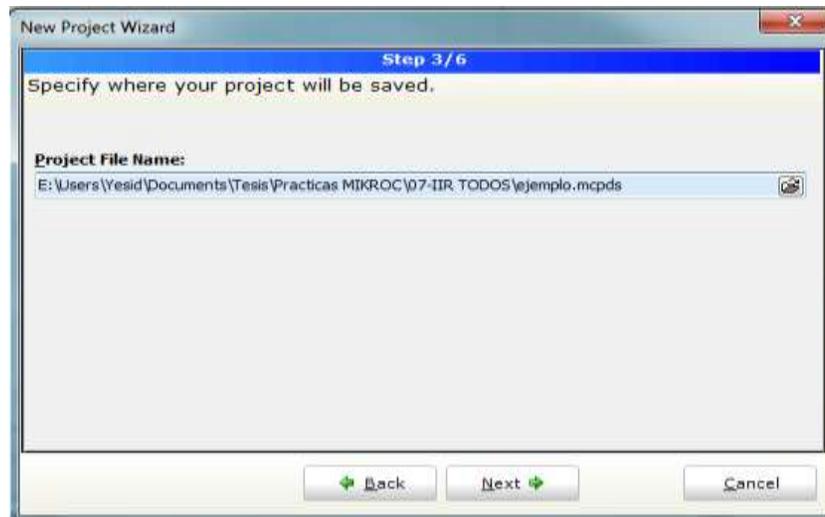
Figura 78. Interfaz de selección del reloj MikroC Pro for dsPIC.



Fuente: Imagen tomada por el autor del texto usando MikroC Pro for dsPIC

4. Especificaciones del proyecto. Se selecciona el nombre y la ruta para el proyecto.

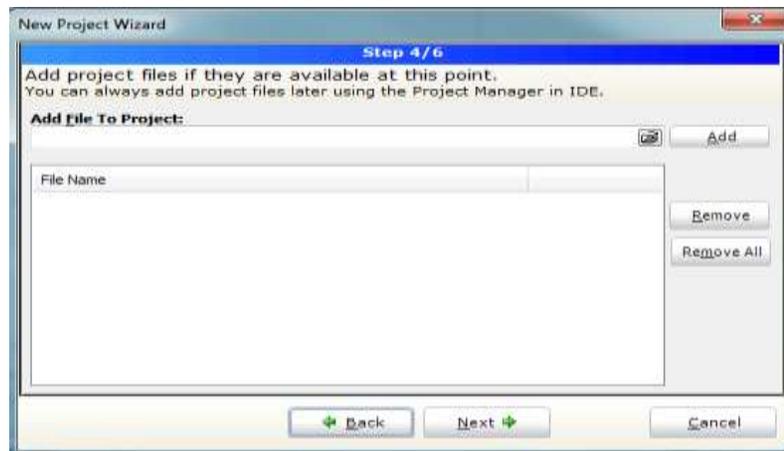
Figura 79. Interfaz de asignación del nombre y ruta del proyecto MikroC Pro for dsPIC



Fuente: Imagen tomada por el autor del texto usando MikroC Pro for dsPIC

5. Selección de los ficheros del proyecto. Se especifica los archivos que contienen el proyecto y se adicionan para que sean cargados.

Figura 80. Interfaz de selección de ficheros del proyecto MikroC Pro for dsPIC



Fuente: Imagen tomada por el autor del texto usando MikroC Pro for dsPIC

6. Selección de las librerías deseadas.

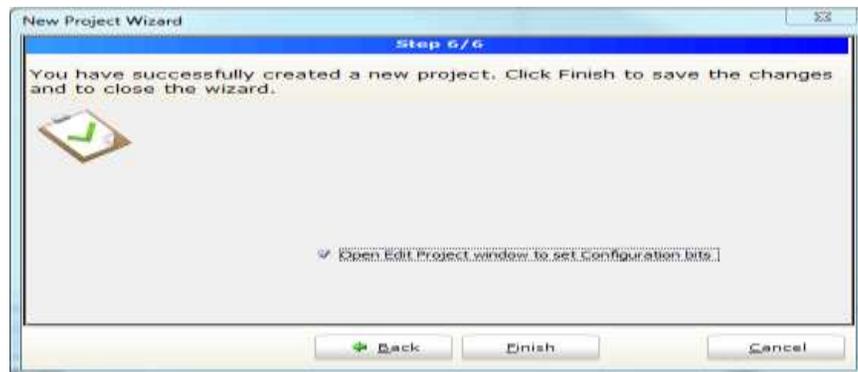
Figura 81. Ventana de selección de librerías Mikroc Pro for dsPIC



Fuente: Imagen tomada por el autor del texto usando Mikroc Pro for dsPIC

7. Creación del nuevo proyecto.

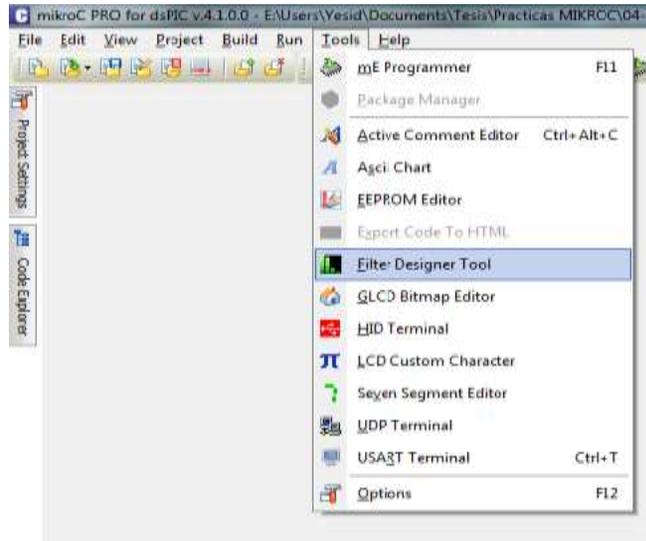
Figura 82. Ventana finalización de creación de proyecto



Fuente: Imagen tomada por el autor del texto usando Mikroc Pro for dsPIC

4.3.1. Asistente para la creación de coeficientes para Mikroc Filter Designer Tool. Para el diseño de un filtro FIR e IIR se inicia mediante la herramienta Filter para esto se debe ir al menú principal del compilador, en tools se halla una serie de utilidades en donde se encuentra el acceso para iniciar la aplicación.

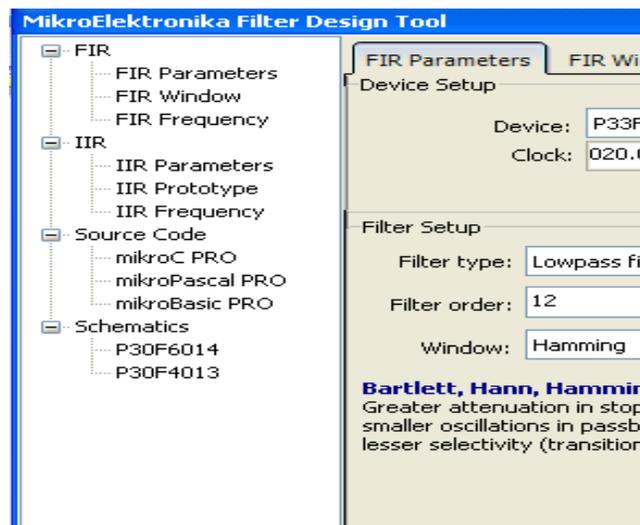
Figura 83. Herramienta de Diseño Filter Designer Tool.



Fuente: Imagen tomada por el autor del texto usando Filter Designer Tool

Una vez ejecutada esta herramienta, en la parte izquierda del asistente se localizan los filtros FIR e IIR con sus respectivas propiedades de configuración. Así mismo se encuentra el código fuente en algunos lenguajes de programación y el esquemático para dos dsPIC específicos de la familia 30F.

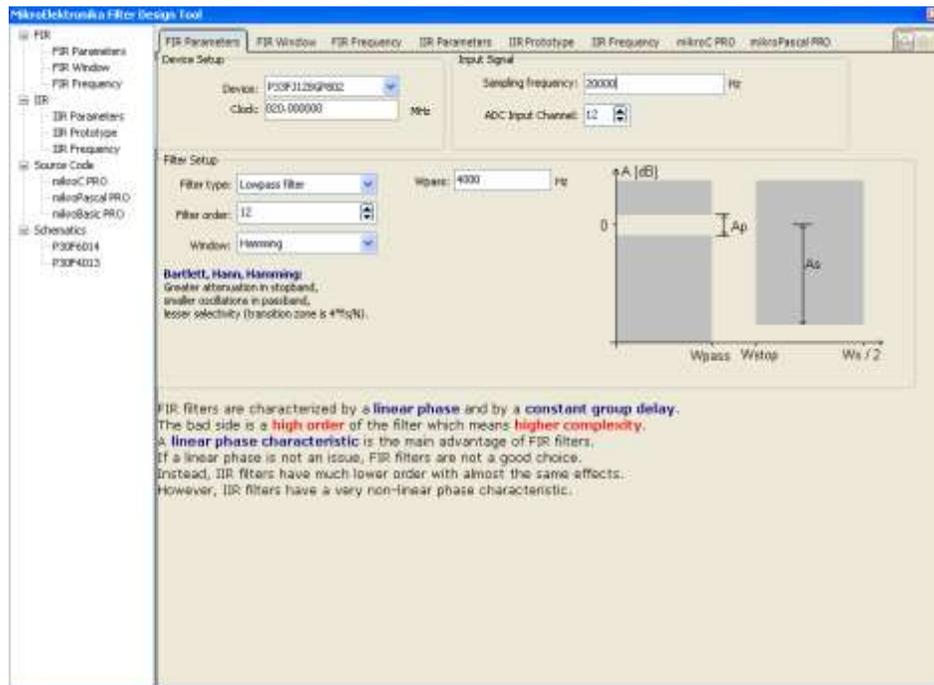
Figura 84. Parámetros de configuración filtros FIR e IIR Filter Designer Tool



Fuente: Imagen tomada por el autor del texto usando Filter Designer Tool.

En los parámetros para los filtros FIR e IIR se encuentra la configuración de dispositivos, como son el reloj y la designación del microcontrolador a utilizar. En las señales de entrada, localizamos dos parámetros, la frecuencia de muestreo y el canal de entrada del convertidor A/D. Cabe resaltar que la frecuencia de muestreo debe cumplir con el teorema de Nyquist, es decir se debe muestrear por lo menos al doble de la frecuencia máxima de la señal de entrada.

Figura 85. Herramienta Filter Designer Tool

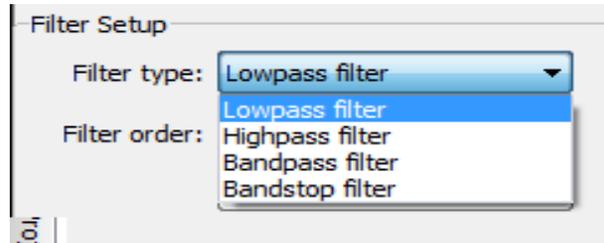


Fuente: Imagen tomada por el autor del texto usando Filter Designer Tool.

Dentro de las opciones de tipo de filtros para ambas clases se encuentran las siguientes:

- Filtro de paso bajo,
- Filtro de paso alto,
- Filtro de paso de banda,
- Rechaza Banda.

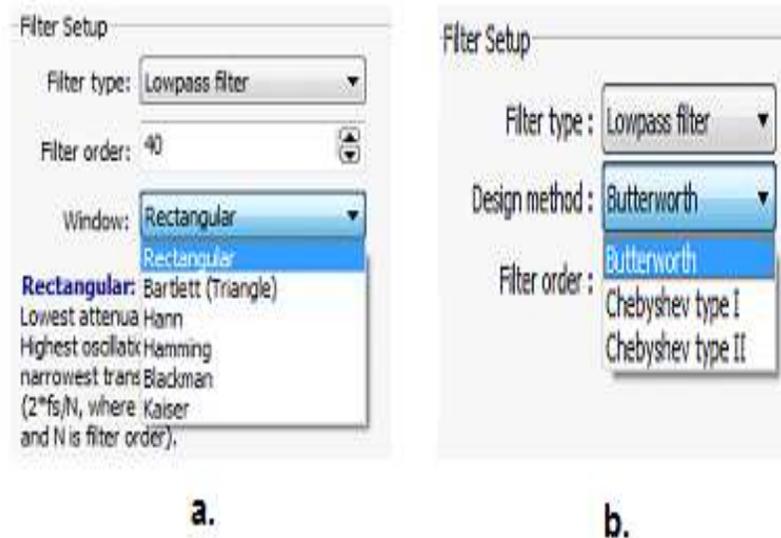
Figura 86. Tipo de Filtro Filter Designer Tool.



Fuente: Imagen tomada por el autor del texto usando Filter Designer Tool.

Después de seleccionar el tipo de filtro y el orden, se selecciona el método a utilizar para obtener el código fuente y sus respectivas gráficas. Para el diseño de Filtros FIR se utiliza el método de ventanas, en donde se encuentran las siguientes opciones: ventana Rectangular, Bartlett, Hann, Hamming, Blackman y Kaiser. En el caso de Filtros IIR los métodos básicos de diseño son: Butterworth, Chebyshev I, Chebyshev II.

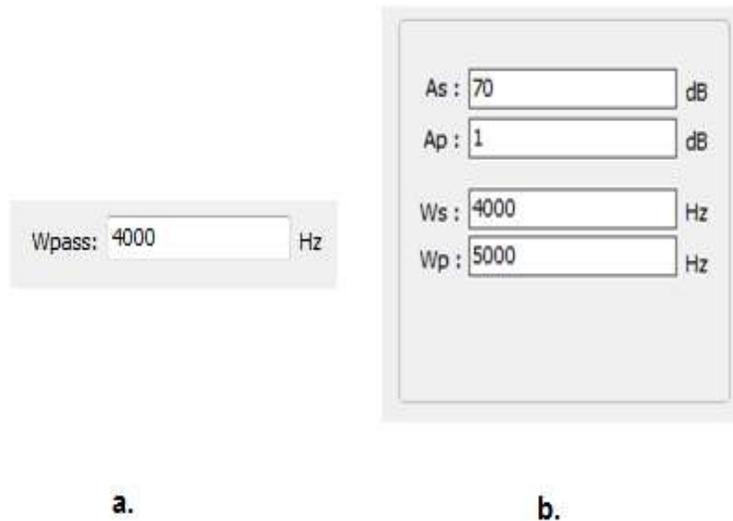
Figura 87. Filter Designer Tool a. Tipo de Ventana Filtro FIR, b. Método del Filtro IIR.



Fuente: Imagen tomada por el autor del texto usando Filter Designer Tool.

Una vez seleccionado el tipo de filtro y dependiendo el método escogido se introducen los parámetros correspondientes, en el caso de los FIR se deben ingresar la frecuencia límite del filtro. En los IIR según método elegido se introducen parámetros tales como atenuación de banda de paso, límite de frecuencia de paso entre otros.

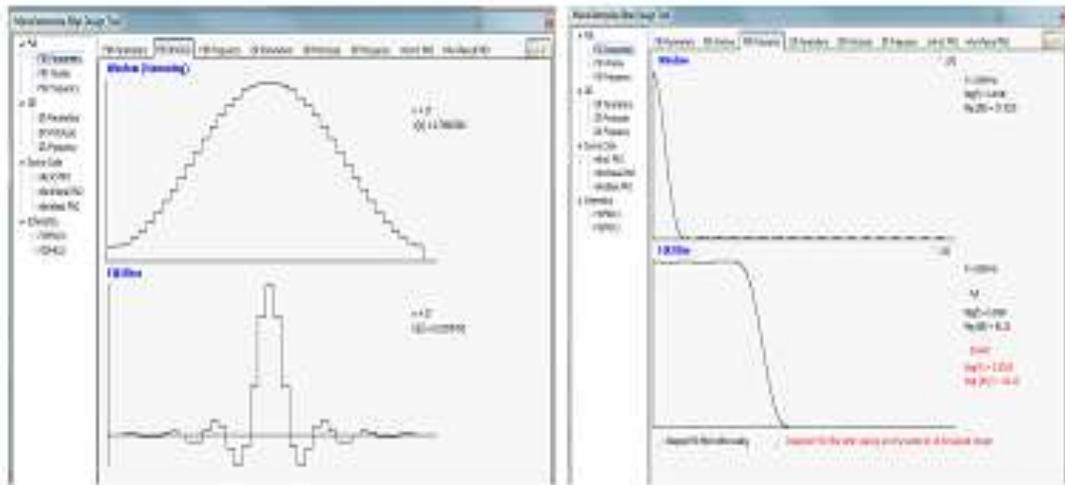
Figura 88. Filter Designer Tool. a. Frecuencia Límite del Filtro FIR, b. Frecuencia límite, atenuación de banda de Paso del filtro IIR



Fuente: Imagen tomada por el autor del texto usando Filter Designer Tool.

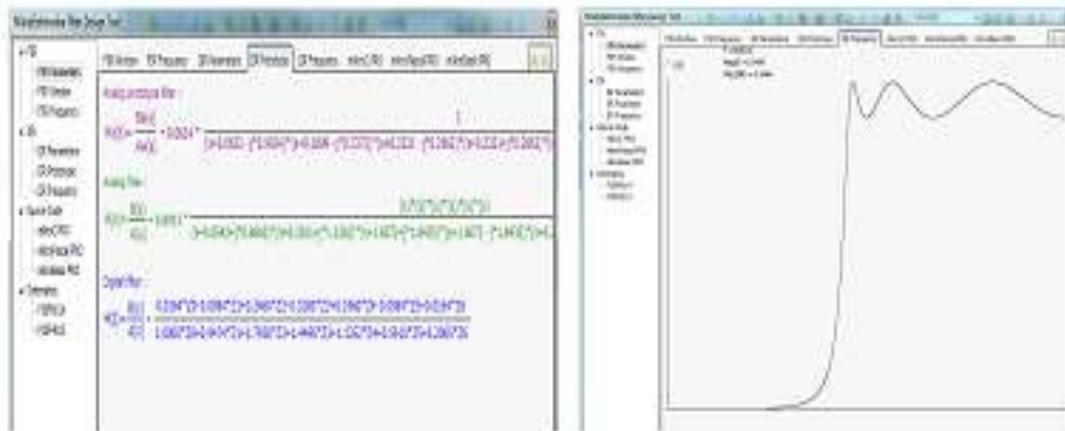
Cuando todos los parámetros se encuentran establecidos, se selecciona una de las opciones del asistente del lado izquierdo, para FIR se encuentra el filtro diseñado con el tipo de ventana, pulso de respuesta, características de la ventana y diseño del filtro FIR. En IIR se tiene el prototipo de la función de transferencia del filtro análogo y digital, a demás de la característica de amplitud en frecuencia del filtro.

Figura 89. Filter Designer Tool. a. Tipo de Ventana y Pulso de Respuesta del Filtro FIR, b. Característica de la Ventana y Diseño del Filtro FIR, c. Función de Transferencia del Filtro Análogo y Digital IIR, d. Amplitud en Frecuencia del Filtro



a.

b.



c.

d.

Fuente: Imagen tomada por el autor del texto usando Filter Designer Tool.

Por último se obtiene el código fuente en los compiladores mikroPascal, mikroBasic o mikroC.

Figura 90. Código Fuente del Programa para el diseño del filtro. Filter Designer Tool.

```

MikroElektronika Filter Design Tool
├─ FIR
│  └─ FIR Parameters
│  └─ FIR Window
│  └─ FIR Frequency
├─ IIR
│  └─ IIR Parameters
│  └─ IIR Prototype
│  └─ IIR Frequency
├─ Source Code
│  └─ mikroC PRO
│  └─ mikroPascal PRO
│  └─ mikroBasic PRO
├─ Schematics
│  └─ P30F6014
│  └─ P30F4013
└─ FIR Window  FIR Frequency  IIR Parameters  IIR Prototype  IIR Frequency  mikroC PRO  mikroPascal PRO  mikroBasic PRO

// Device clock: 020.000000 MHz
// Sampling Frequency: 22050 Hz
// Filter setup:
// Filter kind: IIR
// Filter type: Highpass filter
// Filter order: 6
// Design method: Chebyshev type I

const unsigned int
    BUFFER_SIZE = 8;
const unsigned int
    FILTER_ORDER = 6;

const unsigned int
    COEFF_B[FILTER_ORDER+1] = {0x0433, 0xE6CF, 0x3EFA, 0xAC08, 0x3EFA, 0x26CF, 0x0433, 0x0000};
const unsigned int
    COEFF_A[FILTER_ORDER+1] = {0x4000, 0x3CAl, 0x72CE, 0x5CC6, 0x48AB, 0x23E7, 0x0037};

const unsigned int
    SCALE_B = 1;
const unsigned int
    SCALE_A = -1;

sbit loadPin at LATC2_bit;           // DAC load pin
sbit loadPinDir at TRISC2_bit;       // DAC load pin
sbit csPin at LATC1_bit;             // DAC CS pin
sbit csPinDir at TRISC1_bit;         // DAC CS pin

unsigned int inext;                  // Input buffer index
ydata unsigned int input[BUFFER_SIZE]; // Input buffer
ydata unsigned int output[BUFFER_SIZE]; // Output buffer
    
```

Fuente: Imagen tomada por el autor del texto usando Filter Designer Tool.

5. MANUAL DE PRÁCTICAS DEL ENTRENADOR DE AUDIO dsPIC33FJ128GP802

El presente manual de prácticas del entrenador de audio, describe un serie de prácticas fundamentales en la aplicación de filtros digitales, en donde se manejan una serie de módulos y periféricos implementados en la tarjeta, los software usados para el desarrollo de estas prácticas son MikroC de Mikroelektronika y MPLAB C30 de Microchip, cada una de ellas en su versión estudiantil, para la programación del dsPIC implementado en la tarjeta se hará uso del programador PICKIT2. Todas las practicas han sido implementadas en el entrenador de audio, cada una de ellas encontraran el objetivo, digrama de conexiones, procedimiento y digrama de flujo.

Con lo anterior, éste manual de prácticas intenta satisfacer las necesidades del estudiante y del profesor con la finalidad de llevar a cabo lo más eficientemente posible el proceso enseñanza-aprendizaje.

Se debe resaltar que el alumno para esto debe tener conceptos fundamentales en programación en lenguaje C, señales, y filtros digitales.

5.1.PRÁCTICAS DEL ENTRENADOR DE AUDIO dsPIC33FJ128GP802 UTILIZANDO EL COMPILADOR C30 DE MICROCHIP

A continuación se propone una serie de prácticas aplicando los conceptos de filtros digitales y manejo del dsPIC implementados en esta tarjeta, en donde se darán a conocer los diferentes tipos de configuración del módulo en la implemetacion de filtros digitales en audio en tiempo real.

5.1.1. Práctica No. 1 ADC/DAC LOOPBACK

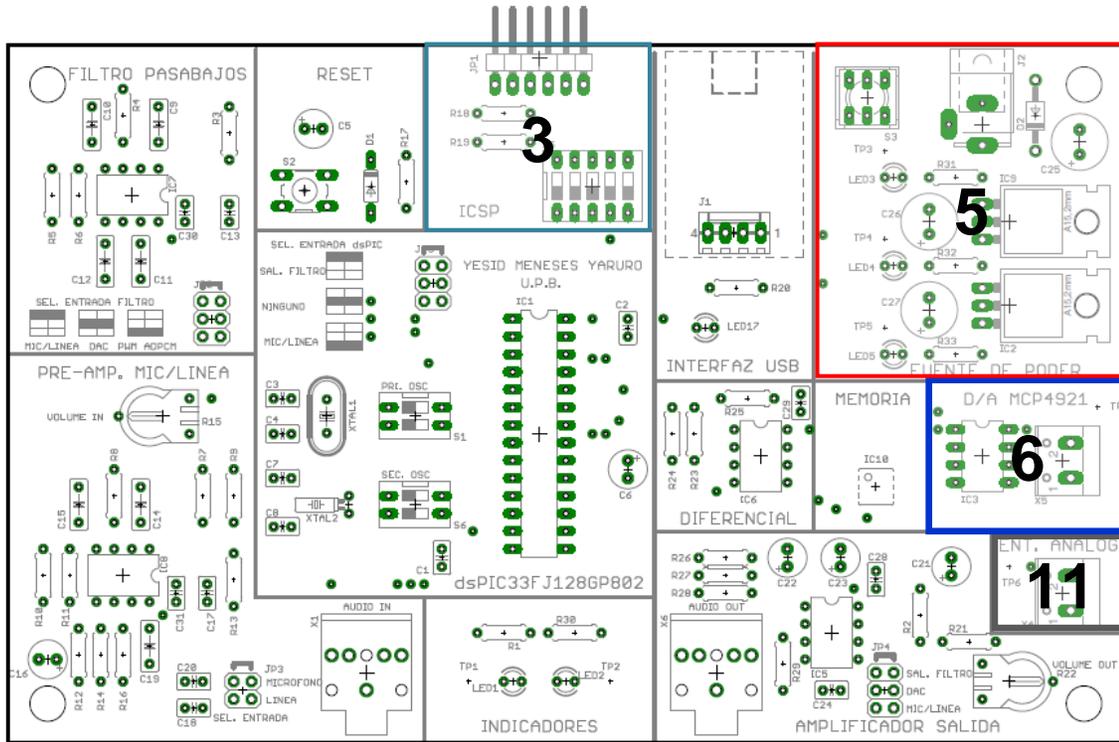
La presente práctica nos muestra el comportamiento de la entrada ADC con respecto a una señal obtenida por el generador de señales, teniendo como resultado la retroalimentación del sistema.

Objetivo:

Aprender a configurar los periféricos y dispositivos que van a hacer parte de la adquisición, procesamiento y reconstrucción de la señal.

Diagrama de Conexiones:

Figura 91. Módulo entrenador de Audio “ADC/DAC LOOPBACK” Mplab C30



Fuente: El autor del texto.

Procedimiento:

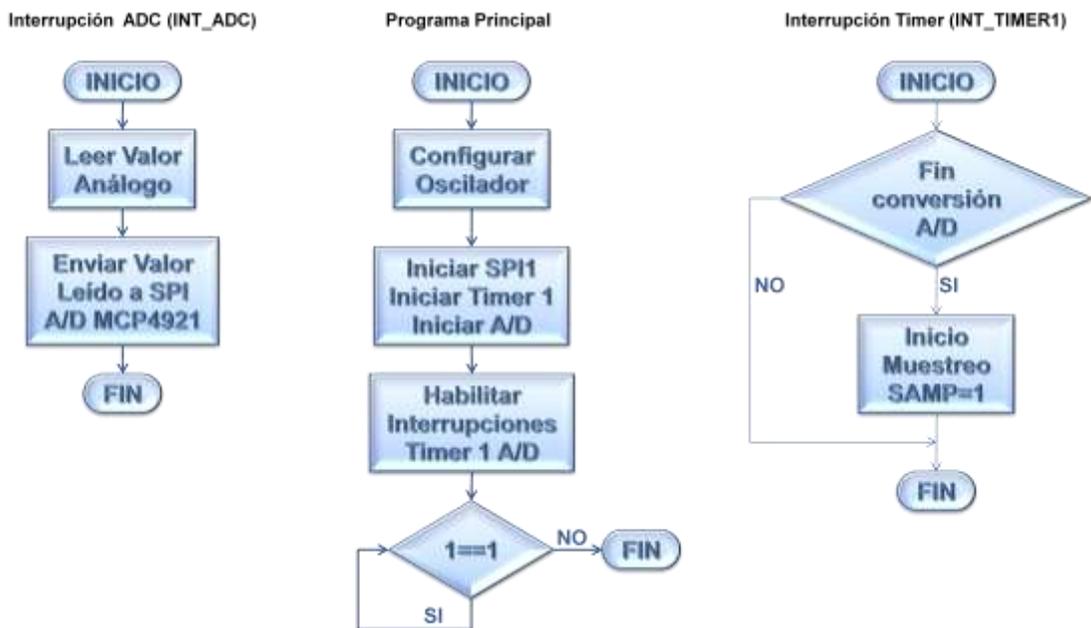
- Abra el proyecto de MPLAB y realice la programación propuesta en el diagrama de flujo (Figura 92) “01-AD-DA LOOPBACK”.
- Conecte el programador *PICKIT2* a la tarjeta entrenadora. (bloque 3)
- Conecte el adaptador de voltaje a la tarjeta de desarrollo, encienda la tarjeta, compile el código y grabe el dsPIC.
- Configure el generador de señales para que produzca una onda senoidal de frecuencia 2KHz, con OFFSET de 1.5V, y 1.5Vp de amplitud (**ADVERTENCIA:** si se envía la señal sin offset se quemará el microcontrolador).
- Inyecte la señal en el conector marcado como “ENT ANALOG” en la tarjeta, teniendo en cuenta la polaridad del mismo.
- Conecte el osciloscopio en el conector marcado como “D/A MCP4921” en la tarjeta, teniendo en cuenta la polaridad del mismo.
- Habilite el pin MCLR desde el MPLAB para iniciar el programa.

Observaciones:

El programa ejecutado en funcionamiento, deberá sacar la misma señal de entrada en la salida.

La utilidad de esta función es verificar el correcto funcionamiento del microcontrolador, encargado de realizar una conversión Análogo/Digital y el convertidor Digital Análogo.

Figura 92. Diagrama de Flujo “01-AD-DA LOOPBACK” Mplab C30



Fuente: Autor del texto.

5.1.2. Práctica No. 2 FIR Pasa-bajos

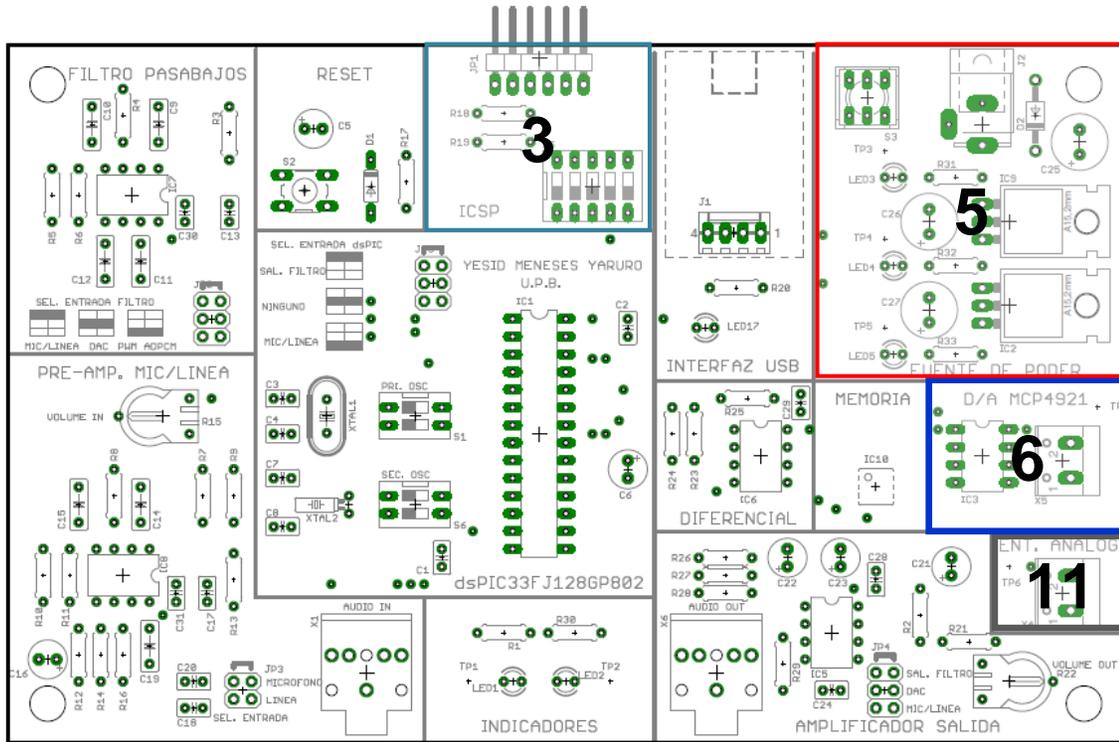
La práctica No. 2 muestra la implementación de un Filtro FIR pasa-bajo mediante el uso de la herramienta dsPICLiteFD y la obtención de los coeficientes del filtro. De igual forma se deben tener claro los conceptos respectivos en el diseño y desarrollo de filtros FIR.

Objetivo:

Implementar un filtro FIR pasa-bajo, de acuerdo a unas condiciones preestablecidas.

Diagrama de Conexiones:

Figura 93. Entrenador de audio 02-FIR-LOWPASS Mplab C30

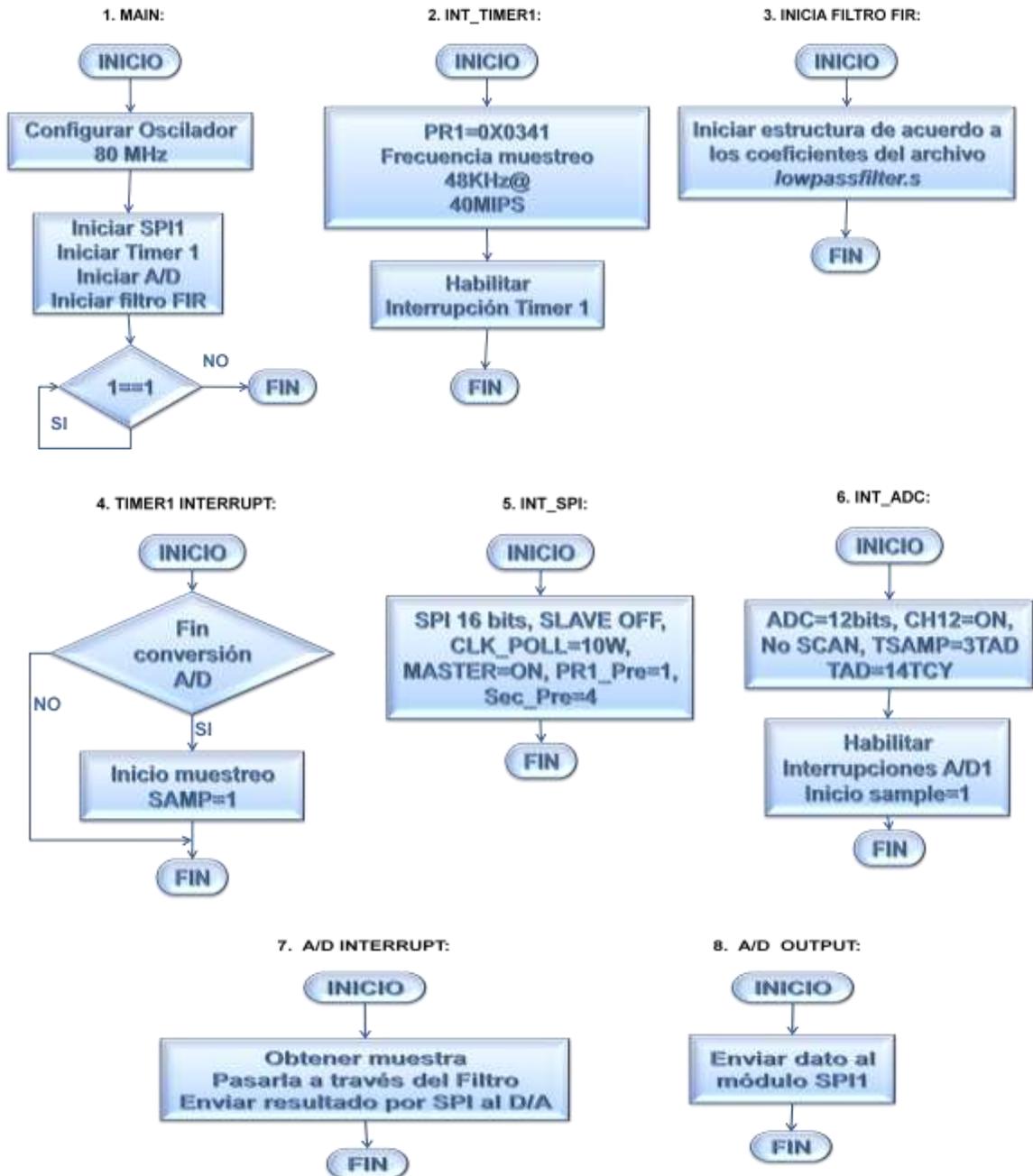


Fuente: Autor del texto.

Procedimiento:

- Abra un nuevo proyecto de MPLAB y realice la programación propuesta en el diagrama de flujo (Figura 94) "02-FIR-LOWPASS".
- Conecte el programador PICKIT2 a la tarjeta entrenadora.
- Encienda la tarjeta, compile el código y grabe el dsPIC.
- Configure el generador de señales para que produzca una onda senoidal en un rango de 0 a 1.5 KHz, con OFFSET de 1.5V, y 1.5Vp de amplitud (**ADVERTENCIA:** si se envía la señal sin offset, o con polaridad negativa se quemará el microcontrolador).
- Inyecte la señal en el conector marcado como "ENT ANALOG" en la tarjeta, teniendo en cuenta la polaridad del mismo.
- Conecte el osciloscopio en el conector marcado como "D/A MCP4921" en la tarjeta, teniendo en cuenta la polaridad del mismo.
- Habilite el pin MCLR desde el MPLAB para iniciar el programa.
- Varíe la frecuencia del generador de señales observando la salida en el osciloscopio.

Figura 94. Diagrama de flujo "02-FIR-LOWPASS" Mplab C30



Fuente: Autor del texto.

Observaciones:

El código implementado es un filtro pasa-bajo, con frecuencia de paso de 1 Hz hasta 800 Hz. Se aprecia que al variar el generador de señales las frecuencias debajo de 800Hz son las que pasan y no son atenuadas. Estos parámetros son previamente utilizados en la herramienta dsPICLiteFD en el cual se dan los coeficientes y las respuestas en frecuencia del filtro deseado.

5.1.3. Práctica No. 3 FIR Final.

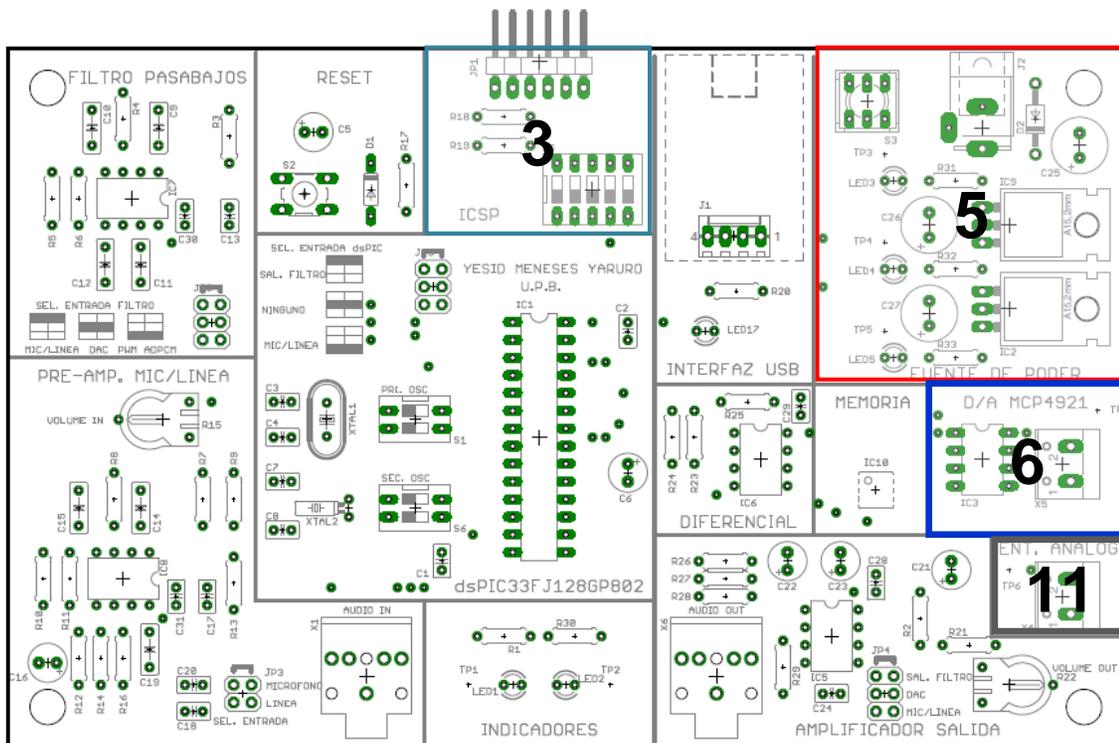
En la práctica No. 3 se implementa los Filtros FIR ya sea en pasa-bajo, pasa-alto y pasa-banda en el cual se deben tener en cuenta los principales parámetros para cada uno de los filtros anteriores.

Objetivo:

Implementar un filtro FIR, bien sea, pasa-banda, pasa-bajo o pasa-alta con un solo algoritmo, utilizando la entrada analoga.

Diagrama de Conexiones:

Figura 95. Entrenador de audio *03-FIR-FINAL* Mplab C30

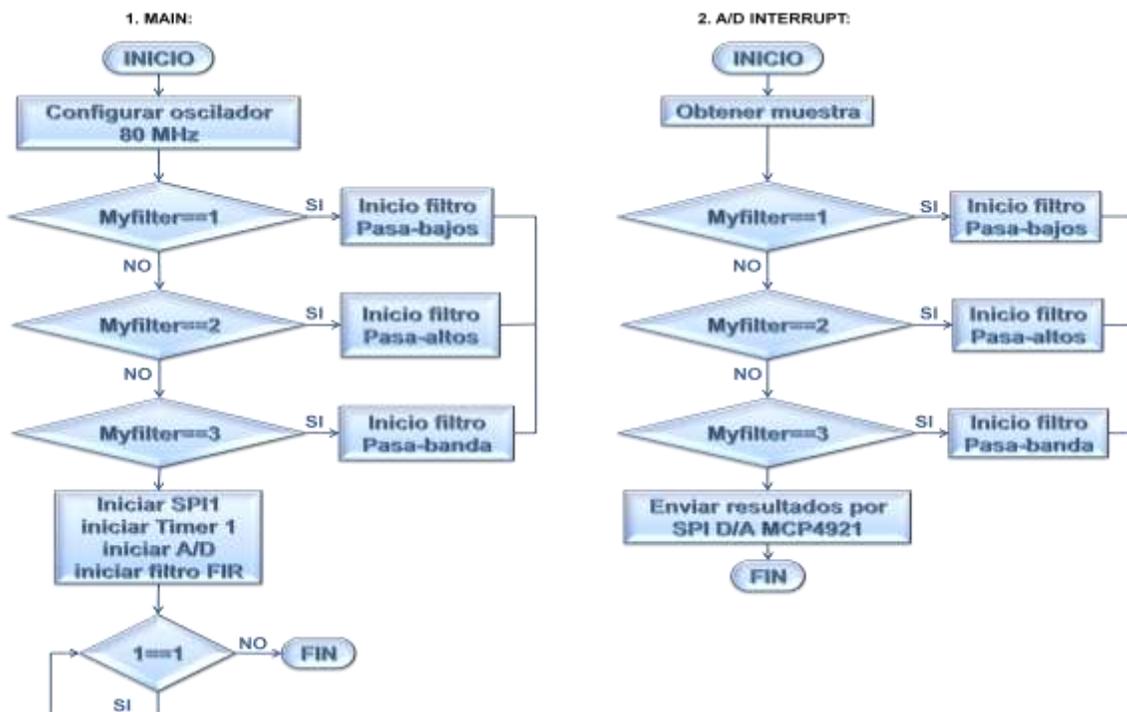


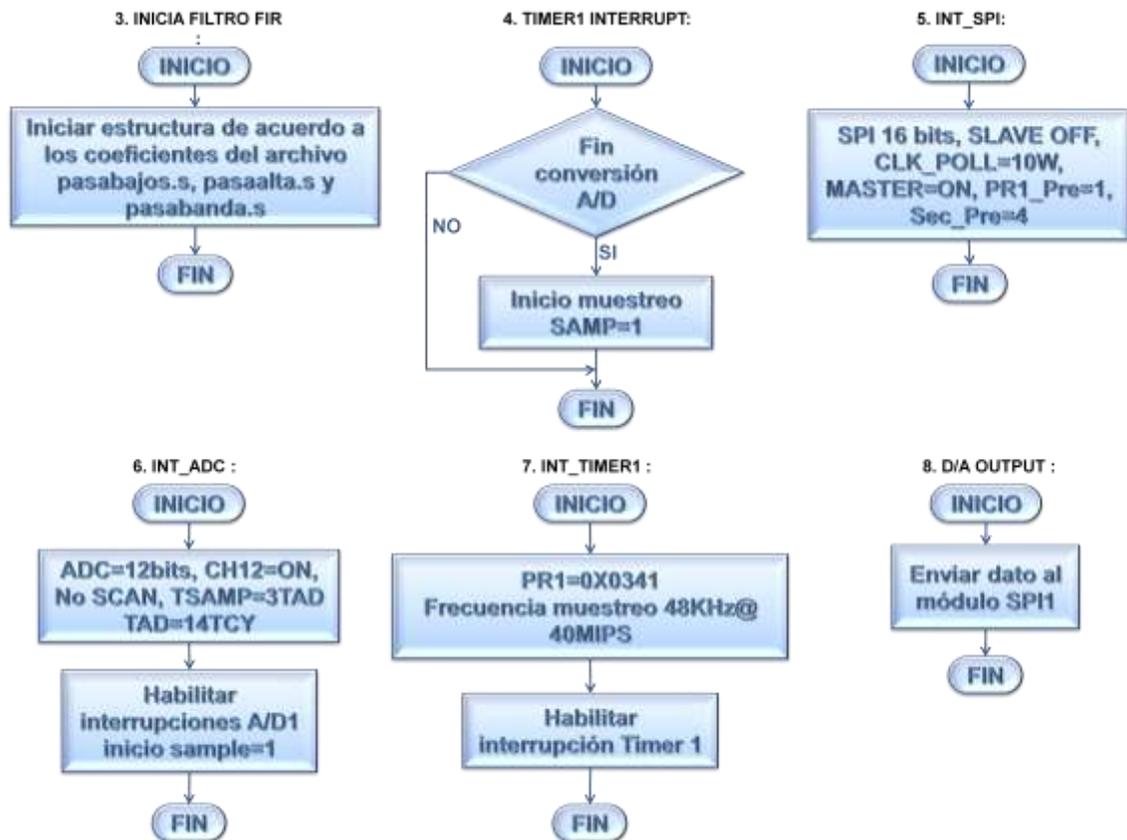
Fuente: El autor del texto

Procedimiento:

- Abra un nuevo proyecto de MPLAB y realice la programación propuesta en el diagrama de flujo (Figura 96) "03-FIR-FINAL".
- Conecte el programador *PICKIT2* a la tarjeta entrenadora.
- Exporte el fichero en el cual se encuentre los coeficientes del filtro utilizando la herramienta dsPICLiteFD especificado por el profesor.
- Encienda la tarjeta, compile el código y grabe el dsPIC.
- Configure el generador de señales para que produzca una onda senoidal en un rango de 0 a 1.5 KHz, con OFFSET de 1.5V, y 1.5Vp de amplitud (**ADVERTENCIA:** si se envía la señal sin offset, o con polaridad negativa se quemará el microcontrolador).
- Inyecte la señal en el conector marcado como "ENT ANALOG" en la tarjeta, teniendo en cuenta la polaridad del mismo.
- Conecte el osciloscopio en el conector marcado como "D/A MCP4921" en la tarjeta, teniendo en cuenta la polaridad del mismo.
- Habilite el pin MCLR desde el MPLAB para iniciar el programa.
- Ajuste las ganancias de entrada y de salida de acuerdo a las señales procesadas.
- Varíe la frecuencia del generador de señales observando la salida en el osciloscopio.

Figura 96. Diagrama de flujo "03-FIR-FINAL".





Fuente: Autor del texto.

Observaciones:

El código implementado tiene tres opciones: filtro pasa-bajo, pasa-alta, pasa-banda, Estos parámetros son previamente utilizados en la herramienta dsPICLiteFD en el cual se dan los coeficientes y las respuestas en frecuencia del filtro deseado. Verificar cada uno de los tipos de filtro mediante la selección MyFilter dentro del código propuesto en el diagrama de flujo.

5.1.4. Práctica No. 4 IIR Pasa-alto

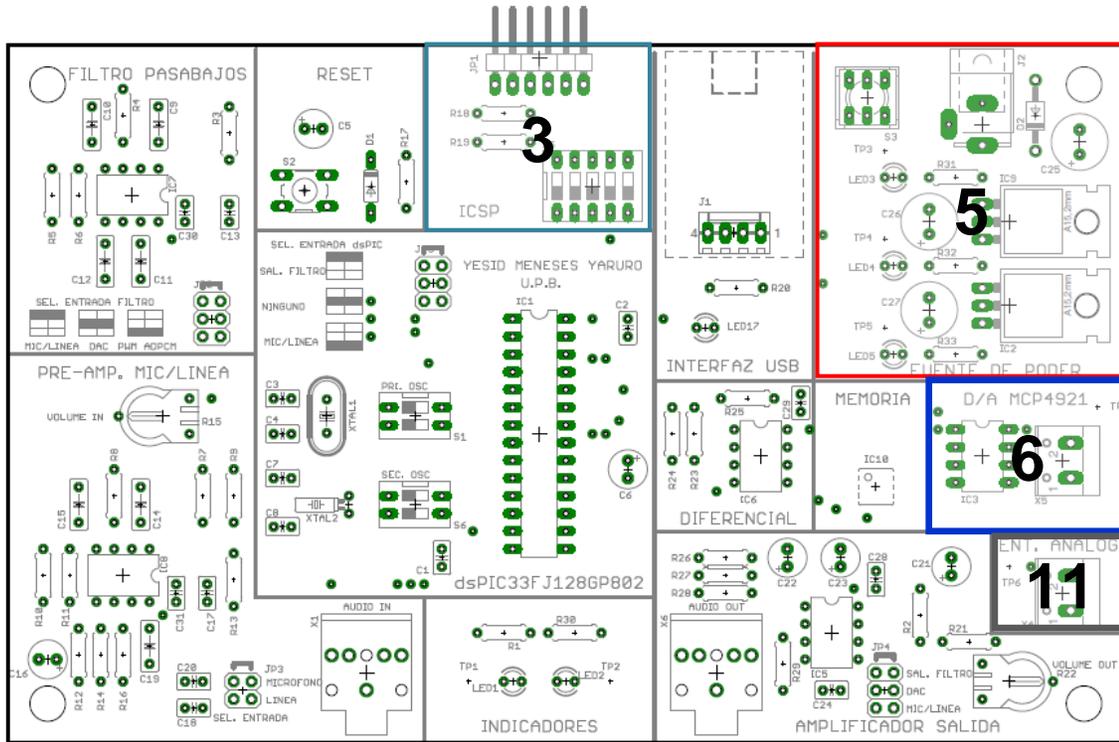
La práctica No.4 implementa un Filtro IIR pasa alto en donde se observan sus principales características en el osciloscopio donde se determinaran la frecuencia de corte y su comportamiento con la frecuencia del sistema.

Objetivo:

Implementar un filtro IIR pasa-alto.

Diagrama de Conexiones:

Figura 97. Entrenador de audio IIR 04-IIR-HIGHPASS Mplab C30

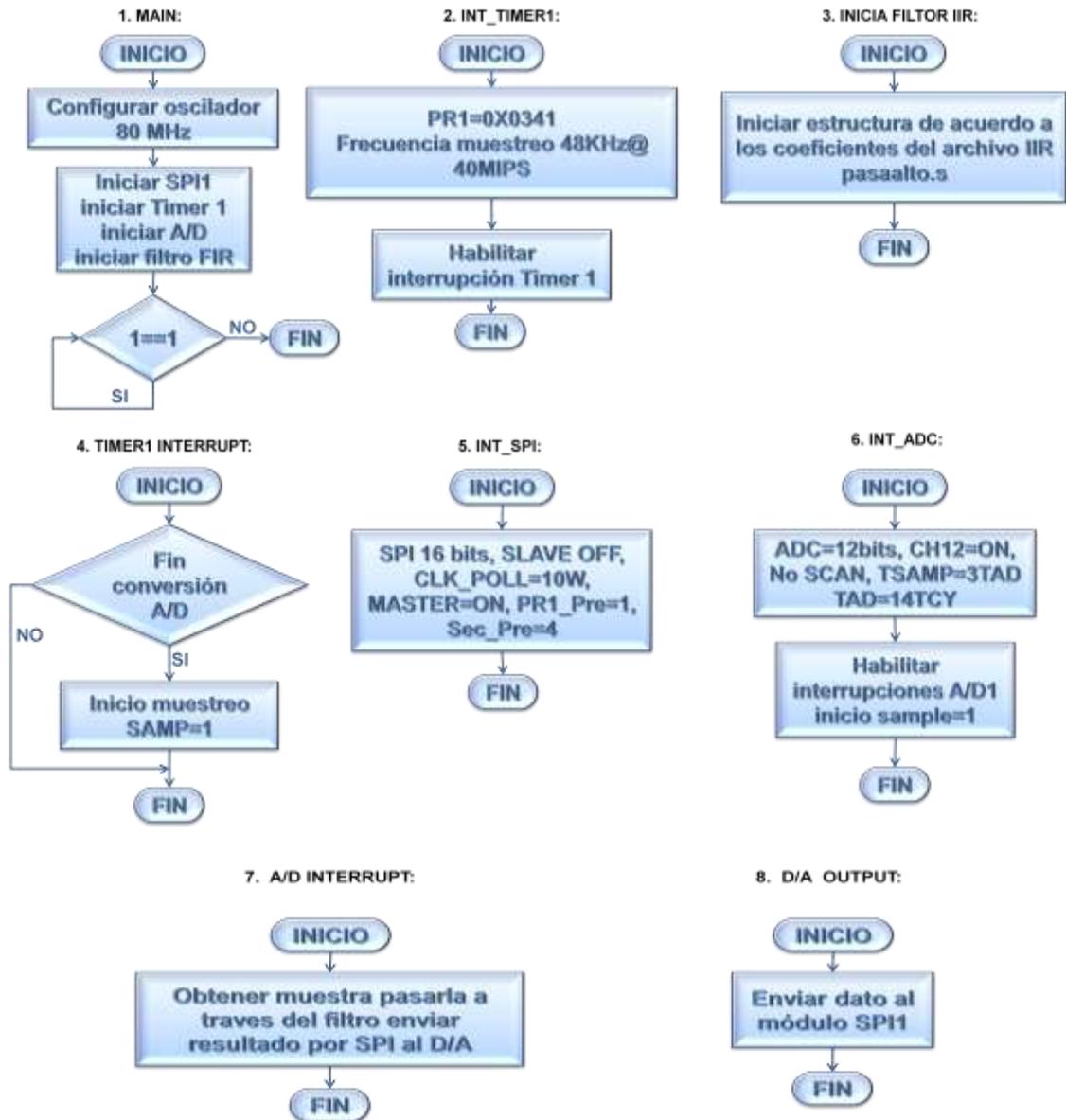


Fuente: Autor del texto

Procedimiento:

- Abra un nuevo proyecto de MPLAB realice la programación propuesta en el diagrama de flujo (Figura 98) "04-IIR-HIGHPASS".
- Conecte el programador PICKIT2 a la tarjeta entrenadora.
- Encienda la tarjeta, compile el código y grabe el dsPIC.
- Configure el generador de señales para que produzca una onda senoidal en un rango de 0 a 1.5 KHz, con OFFSET de 1.5V, y 1.5Vp de amplitud (**ADVERTENCIA:** si se envía la señal sin offset, o con polaridad negativa se quemará el microcontrolador).
- Inyecte la señal en el conector marcado como "ENT ANALOG" en la tarjeta, teniendo en cuenta la polaridad del mismo.
- Conecte el osciloscopio en el conector marcado como "D/A MCP4921" en la tarjeta, teniendo en cuenta la polaridad del mismo.
- Habilite el pin MCLR desde el MPLAB para iniciar el programa.
- Varíe la frecuencia del generador de señales observando la salida en el osciloscopio

Figura 98. Diagrama de flujo "04-IIR-HIGHPASS" Mplab C30



Fuente: Autor del texto.

Observaciones: Este código se encarga de realizar un filtro pasa alto, con una frecuencia de corte establecida. Al variar el generador de señales, se puede observar que en la salida del sistema.

5.1.5. Práctica No. 5 IIR Final

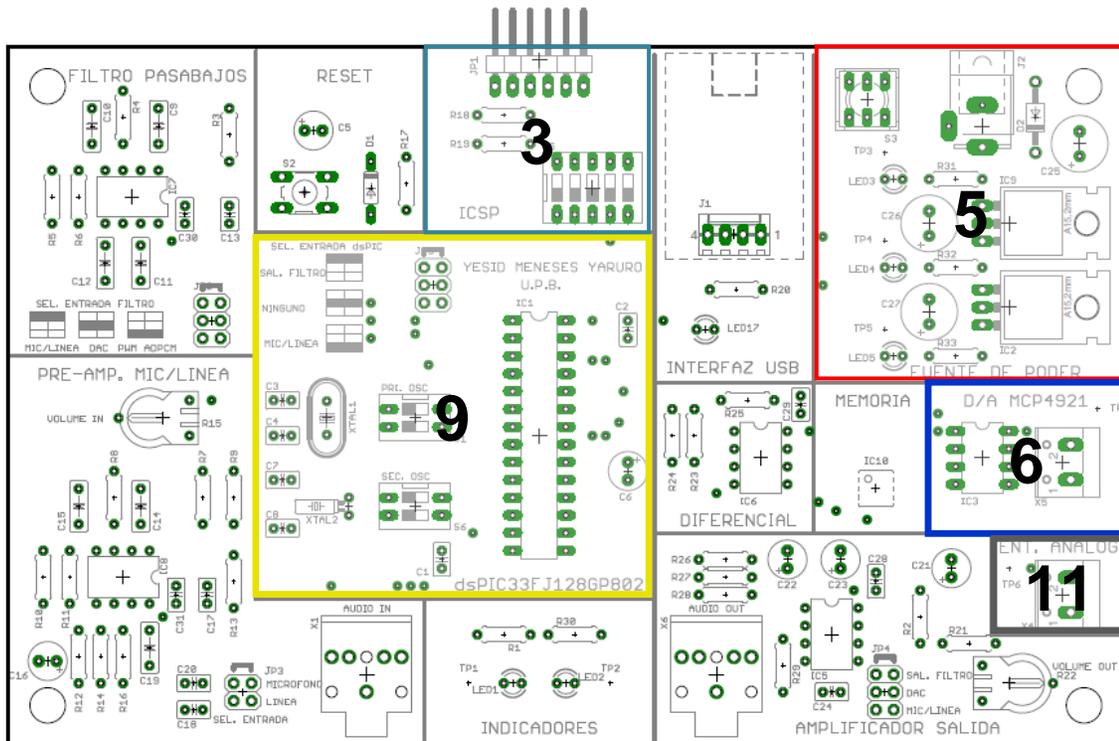
En la práctica No.5 se implementa los Filtros IIR ya sea en pasa-bajo, pasa-alto y pasa-banda en el cual se deben tener en cuenta los principales parámetros para cada uno de los filtros anteriores.

Objetivo:

Implementar un filtro IIR, bien sea tipo pasa-banda, pasa-bajo o pasa-alta con un solo algoritmo, utilizando la entrada analógica.

Diagrama de Conexiones:

Figura 99. Entrenador de audio 05-IIR-FINAL. Mplab C30



Fuente: Autor del texto.

Procedimiento Primera Parte:

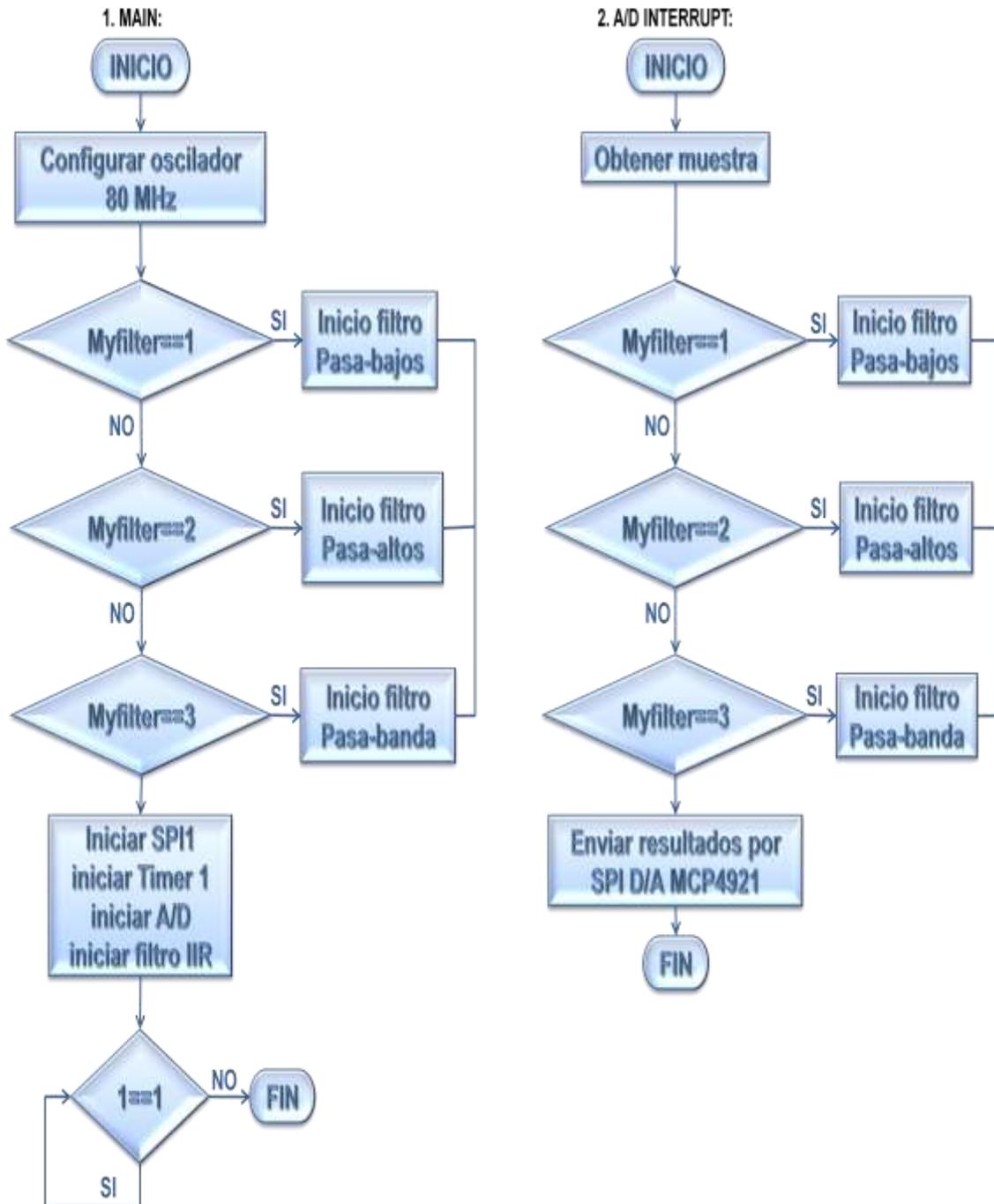
- Abra un nuevo proyecto de MPLAB realice la programación propuesta en el diagrama de flujo (Figura 100) "05-IIR-FINAL".
- Conecte el programador PICKIT2 a la tarjeta entrenadora.
- En el código a compilar, seleccione un tipo de filtro con la opción MYFILTER.

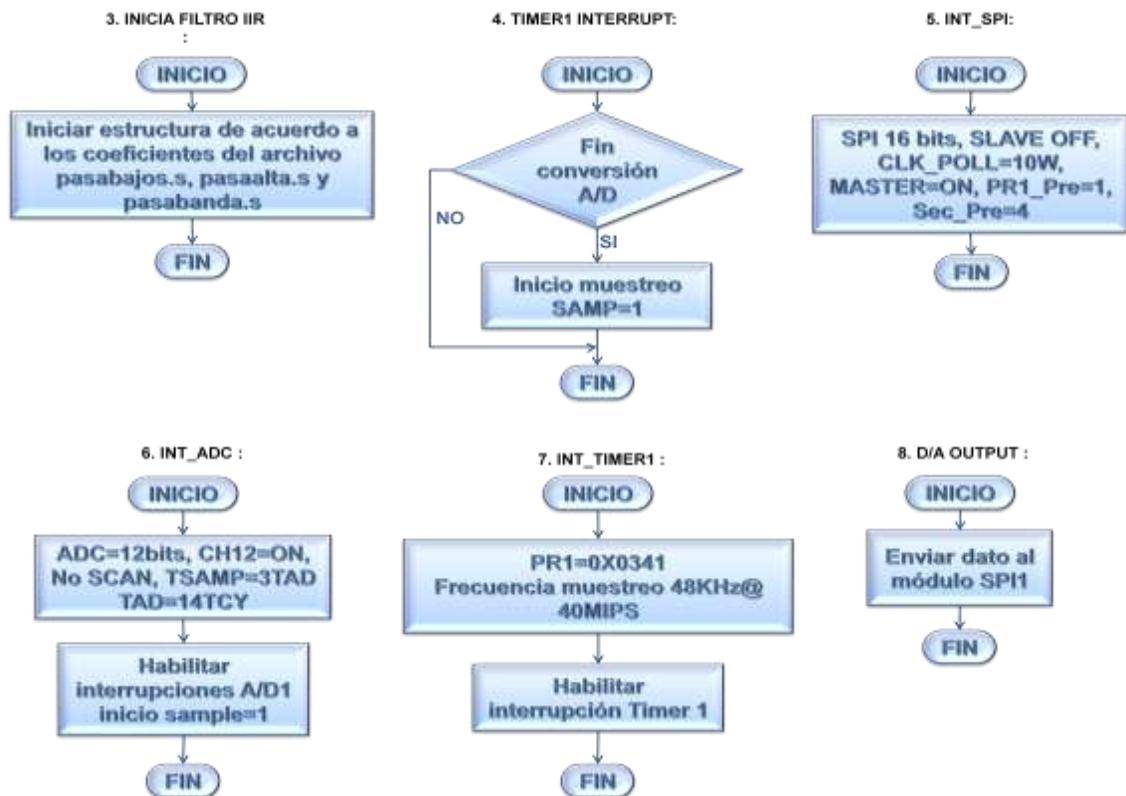
- d) Encienda la tarjeta, compile el código y grabe el dsPIC.
- e) Configure el generador de señales para que produzca una onda senoidal en un rango de acuerdo al filtro seleccionado, con OFFSET de 1.5V, y 1.5Vp de amplitud (**ADVERTENCIA:** si se envía la señal sin offset, o con polaridad negativa se quemará el microcontrolador).
- f) Inyecte la señal en el conector marcado como “*ENT ANALOG*” en la tarjeta, teniendo en cuenta la polaridad del mismo.
- g) Conecte el osciloscopio en el conector marcado como “*D/A MCP4921*” en la tarjeta, teniendo en cuenta la polaridad del mismo.
- h) Habilite el pin MCLR desde el MPLAB para iniciar el programa.
- i) Varíe la frecuencia del generador de señales observando la salida en el osciloscopio.

Procedimiento Segunda Parte:

- a) Manteniendo las conexiones de la primera parte, energizar el sistema.
- b) Escoja en el código, el tipo de filtro especificado por el profesor (definición MYFILTER dentro del código)
- c) Con la ayuda del dsPIC FD lite, genere un nuevo archivo de coeficientes de acuerdo a las especificaciones del filtro.
- d) Incluya el archivo en el Proyecto de MPLAB, reemplazando el anterior archivo.
- e) Compile el proyecto y programe el Microcontrolador.
- f) Inyecte la señal en el conector marcado como “*ENT ANALOG*” en la tarjeta, teniendo en cuenta la polaridad del mismo.
- g) Conecte el osciloscopio en el conector marcado como “*D/A MCP4921*” en la tarjeta, teniendo en cuenta la polaridad del mismo.
- h) Habilite el pin MCLR desde el MPLAB para iniciar el programa.
- i) Varíe la frecuencia del generador de señales observando la salida en el osciloscopio

Figura 100. Diagrama de flujo "05-IIR-FINAL" Mplab C30





Fuente: Autor del texto.

Observaciones: Al cambiar los coeficientes, se rediseña totalmente el tipo de filtro que se estaba trabajando anteriormente. Dando la posibilidad al usuario de poder implementar cualquier filtro que desee, de acuerdo a las condiciones establecidas en el programa del entrenador de audio (frecuencia máxima de entrada).

5.1.6. Práctica NO.6 AUDIO FIR

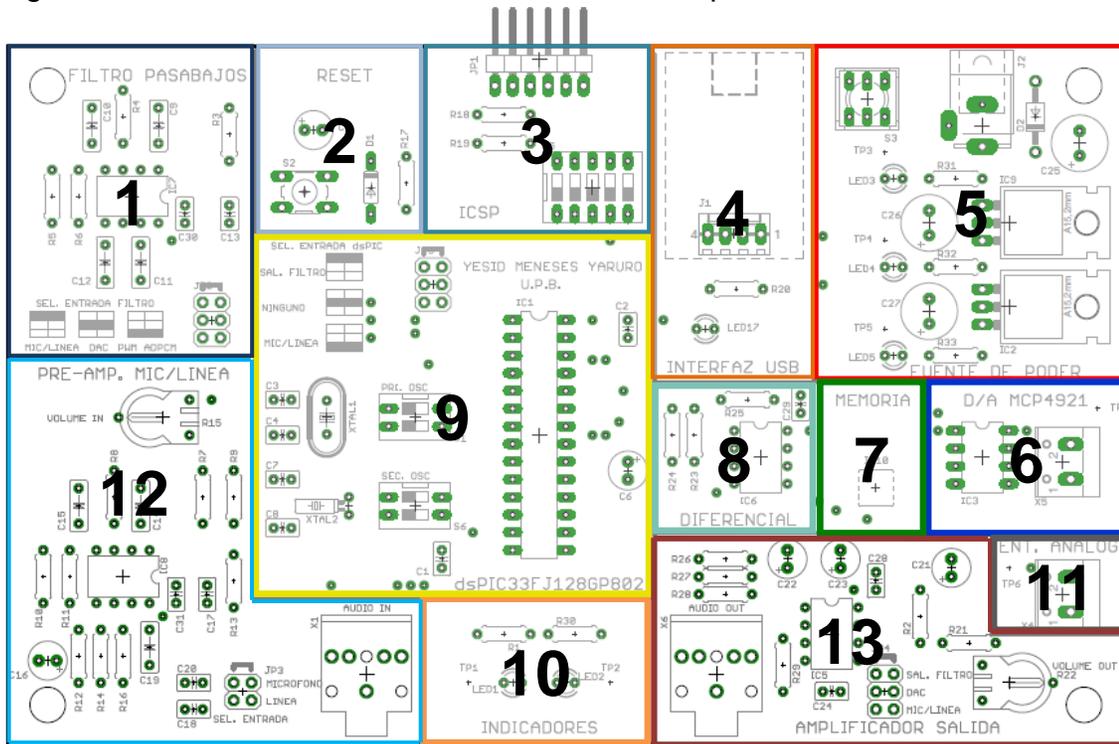
La práctica No. 6 muestra la implementación de un filtro FIR en el módulo de audio en tiempo real, con sus diferentes características y diferentes configuraciones de dicha tarjeta.

Objetivo:

Implementar un filtro FIR, ya sea tipo pasa-banda, pasa-bajo o pasa-alta con un solo algoritmo, utilizando como entrada una señal MP3 o Microfono.

Diagrama de Conexiones:

Figura 101. Entrenador de audio 06-AUDIO-FIR Mplab C30



Fuente: Autor del texto.

Procedimiento Primera Parte:

- Abra un nuevo proyecto de MPLAB realice la programación propuesta en el diagrama de flujo (Figura 102) "06-AUDIO-FIR".
- Conecte el programador PICKIT2 a la tarjeta entrenadora.
- Conecte el jumper (bloque 9) del microcontrolador para configurar la entrada de audio.
- Escoja en el código, el tipo de filtro especificado por el profesor (definición MYFILTER)
- Encienda la tarjeta, compile el código y grabe el dsPIC.
- Conecte en la entrada de línea, o micrófono una fuente de audio (celular, computador, MP3 o micrófono), teniendo en cuenta de ajustar el jumper si la fuente es diferente a un micrófono (bloque 12).
- Conecte la salida del amplificador de salida (Bloque 13) a un parlante tipo desktop.
- Habilite el pin MCLR desde el MPLAB para iniciar el programa.
- Ajuste las ganancias de entrada y de salida de acuerdo a las señales procesadas.

Observaciones:

Esta primera parte se puede implementar los diferentes tipos de filtros FIR mediante la opción MYFILTER, en donde se puede escoger el filtro deseado. De igual forma se pueden tomar muestras por el conversor digital analógico MPC4921 para ser observadas en el osciloscopio.

En esta sección se puede configurar el módulo en cinco opciones de entradas y salidas del módulo para observar los diferentes comportamientos de la tarjeta; las cinco opciones de entradas y salidas son:

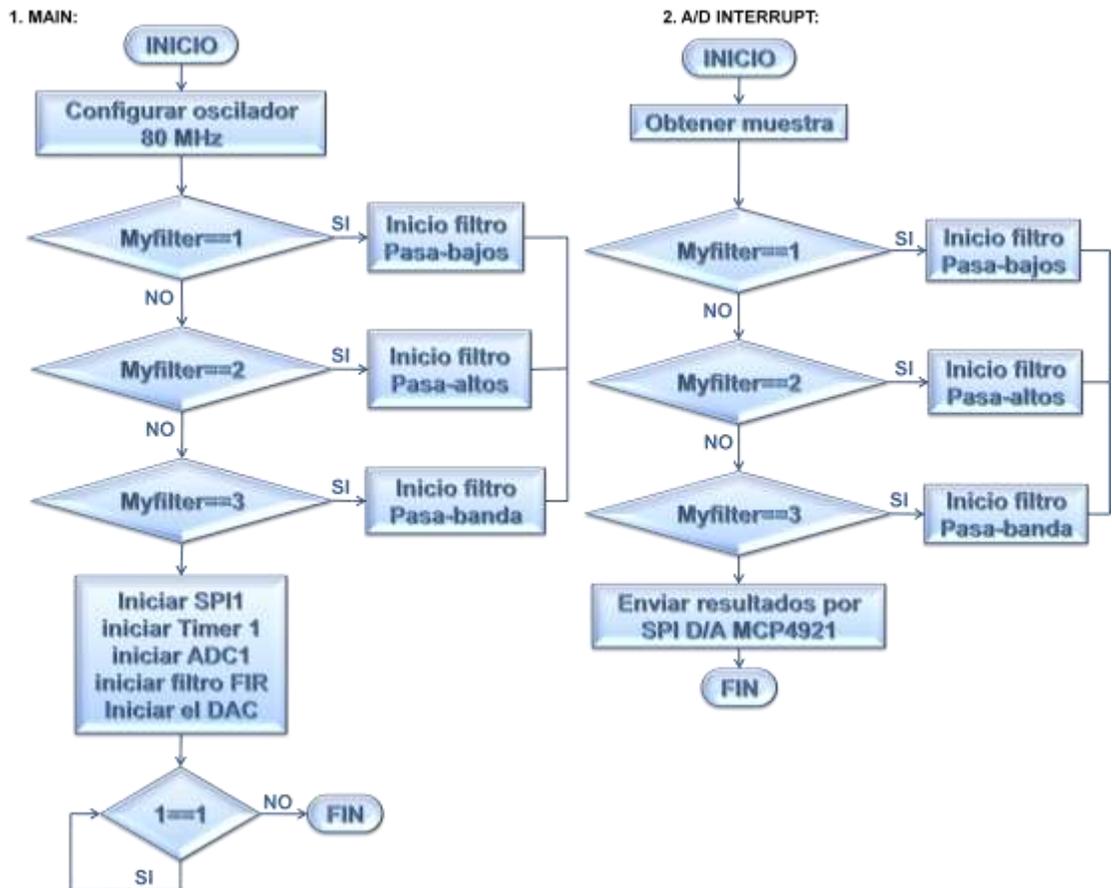
1. Entrada (MIC, línea auxiliar) – Salida (MIC/línea): Produciendo que la entrada sea igual a la salida, esto ajustado por los jumper del amplificador de entrada y salida.
2. Entrada (MIC, línea auxiliar) – dsPIC(MIC/línea) – Salida (DAC): Produciendo que la entrada sea una señal de audio procesada previamente por el filtro FIR programado y reproducido a la salida mediante el DAC interno del microcontrolador y siendo ajustado por los jumper del amplificador de entrada, salida y el dsPIC33FJ128GP802.
3. Entrada (MIC, línea auxiliar) – Filtro Analógico(MIC/línea) – Salida (Filtro): La señal de audio de entrada es procesada por el filtro analógico pasa bajo produciendo una salida del filtro análogo.
4. Entrada (MIC, línea auxiliar) – Filtro Analógico(MIC/línea) – dsPIC(salida de filtro analógico) – Salida (DAC): La señal de audio de entrada es procesada por el filtro analógico pasa bajo el cual es conectado en casacada con el filtro digital implementado en el dsPIC y reproducido a la salida mediante el DAC interno del microcontrolador.
5. Entrada (MIC, línea auxiliar) – dsPIC(MIC/línea) – Filtro Analógico(DAC) – Salida (filtro analógico): La señal de audio de entrada es procesada por el filtro digital implementado en el dsPIC el cual es conectado con el filtro analógico pasa bajo en casacada y reproducido a la salida mediante la configuración del jumper del amplificador de salida del filtro.

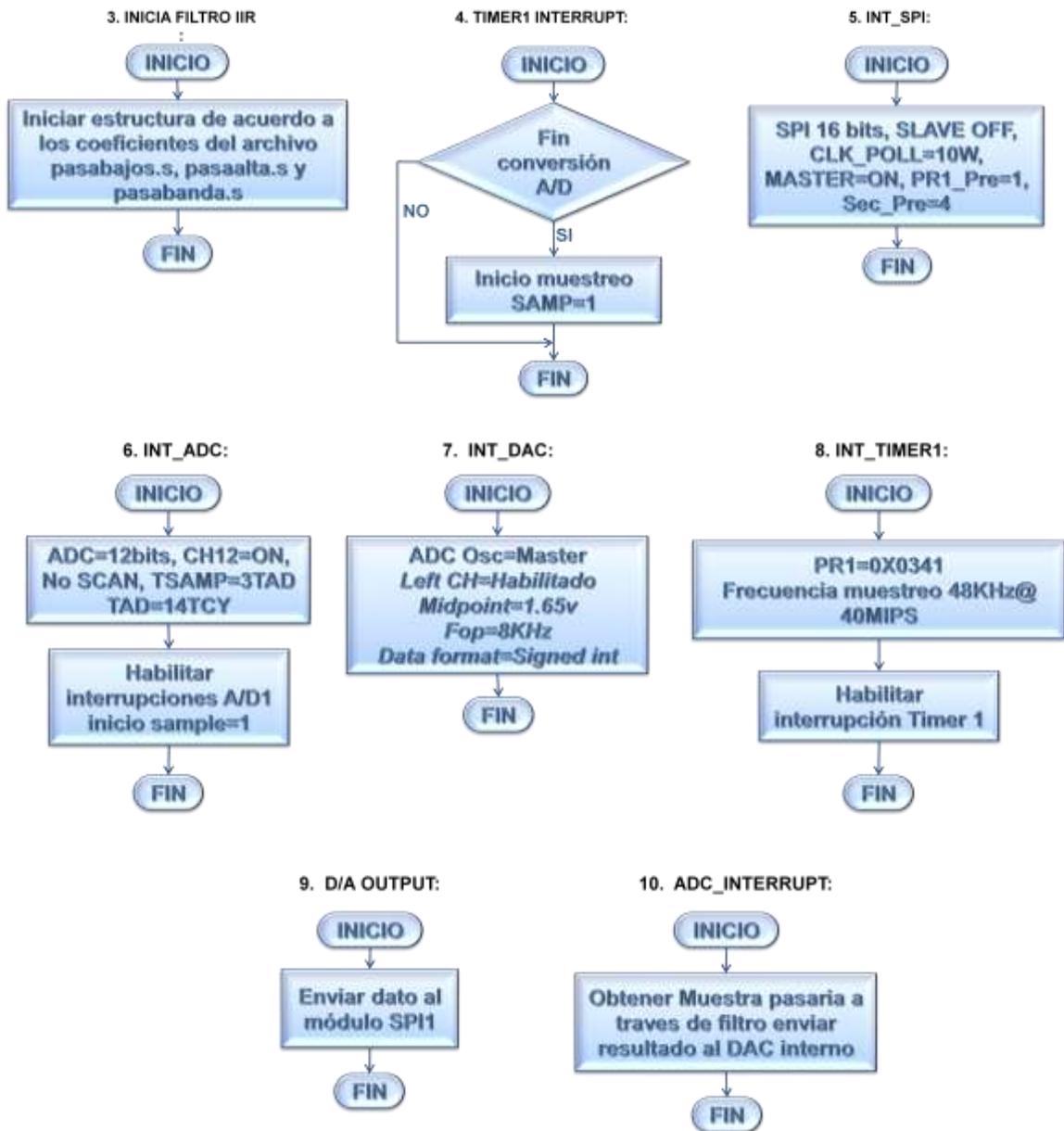
Procedimiento Segunda Parte:

- a) Manteniendo las conexiones de la primera parte, energizar el sistema.
- b) Escoja en el código, el tipo de filtro especificado por el profesor (definición MYFILTER dentro del código)
- c) Con la ayuda del dsPIC FD lite, genere un nuevo archivo de coeficientes de acuerdo a las especificaciones del filtro.
- d) Incluya el archivo en el Proyecto de MPLAB
- e) Compile el proyecto y programe el Microcontrolador.
- f) Conecte en la entrada de línea, o micrófono una fuente de audio (celular, computador, MP3 o micrófono), teniendo en cuenta de ajustar el jumper si la fuente es diferente a un micrófono (bloque 12).

- g) Conecte la salida del amplificador de salida (Bloque 13) a un parlante tipo desktop.
- h) Habilite el pin MCLR desde el MPLAB para iniciar el programa.
- i) Ajuste las ganancias de entrada y de salida de acuerdo a las señales procesadas.

Figura 102. Diagrama de flujo "06-AUDIO-FIR" Mplab C30





Fuente: Al autor del texto

5.2.PRÁCTICAS DEL ENTRENADOR DE AUDIO dsPIC33FJ128GP802 UTILIZANDO EL COMPILADOR MIKROC PRO FOR DSPIC

A continuación se implimentarán una serie de experiencias donde se afinciará el manejo del módulo en el diseño de filtros digitales, utilizando el compilador MikroC PRO for DSPIC.

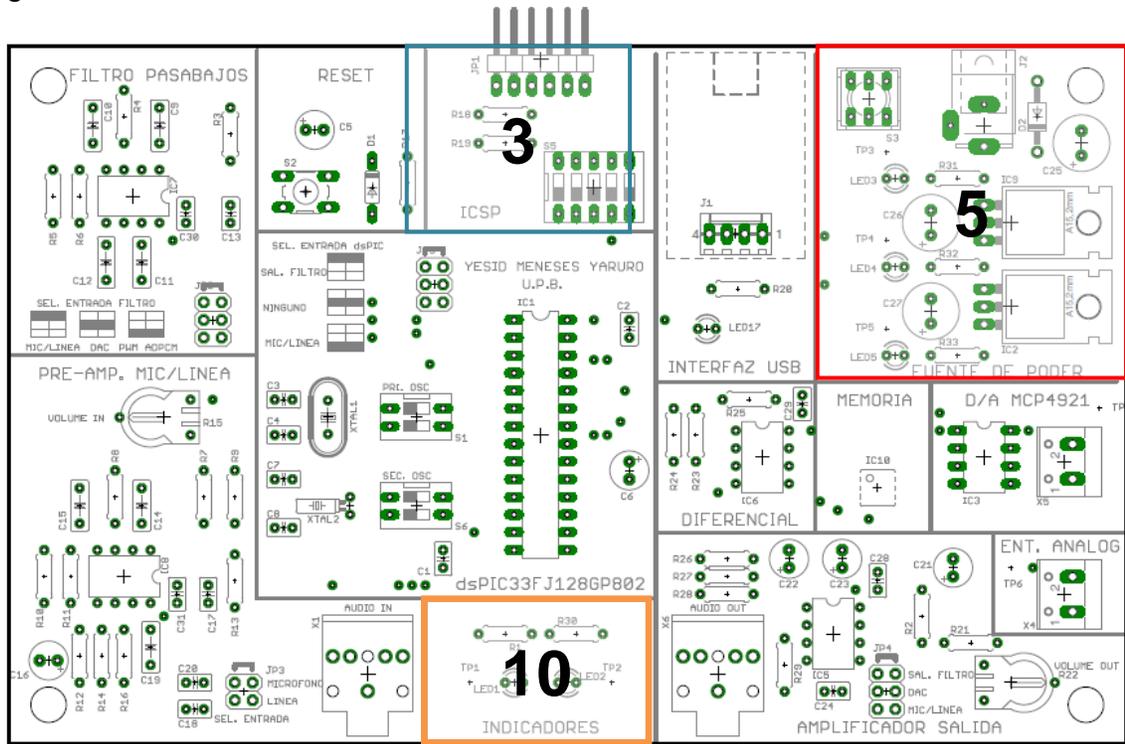
5.2.1. Práctica No. 1 TIMER

Objetivo:

Familiarizarse con el compilador mikroC, implementando una conmutación por timer.

Diagrama de Conexiones:

Figura 103. Entrenador de audion01-TIMER1 PRB MikroC



Fuente: Autor del texto.

Procedimiento:

- Abra un nuevo proyecto de mikroC realice la programación propuesta en el diagrama de flujo (Figura 104) "01-TIMER1 PRB".
- Conecte el programador PICKIT2 a la tarjeta entrenadora.
- Encienda la tarjeta, compile el código y grabe el dsPIC con la ayuda del software Pickit 2.

Figura 104. Diagrama de flujo "01-TIMER1 PRB" MikroC



Fuente: Autor del texto.

Observaciones: El LED oscilará a una frecuencia previamente dispuesta en el código, mediante una interrupción por timer.

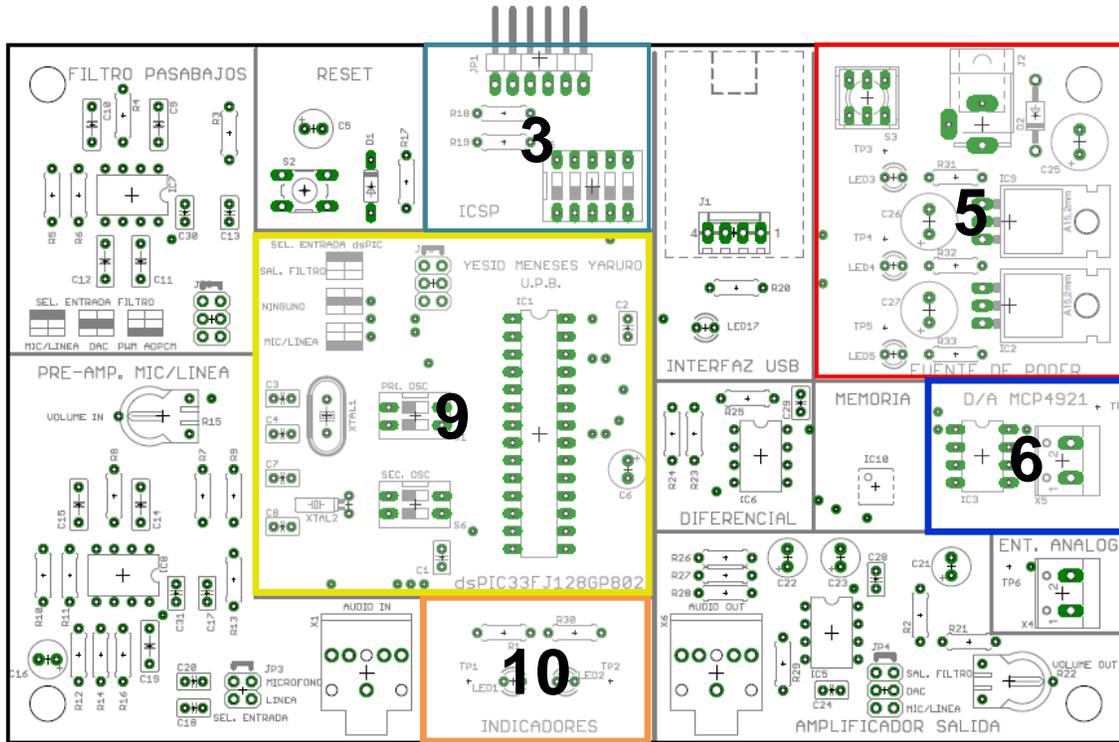
5.2.2. Práctica N.2 MCP4921

Objetivo:

Implementar la generación de una onda preestablecida utilizando el compilador mikroC de mikroelektronika

Diagrama de Conexiones:

Figura 105. Entrenador de audio *02-MCP4921 PRB* MikroC

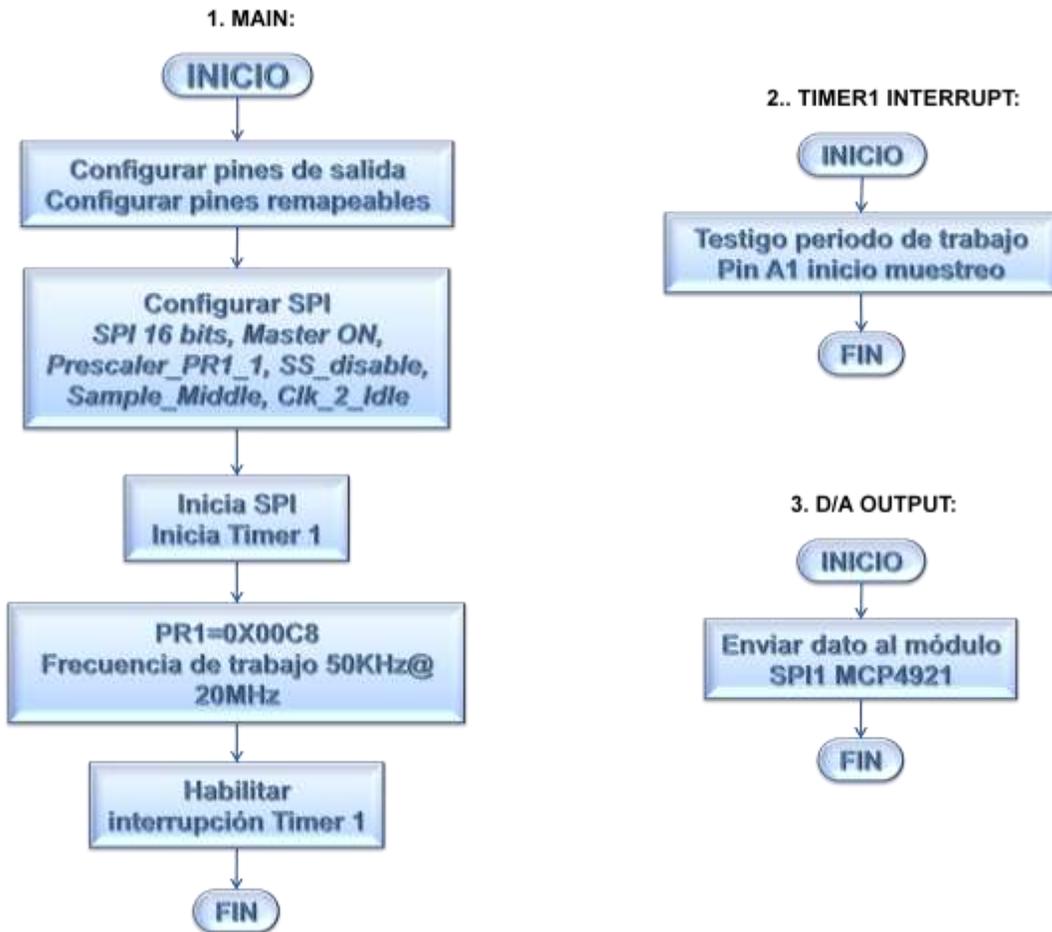


Fuente: Autor del texto.

Procedimiento:

- Abra un nuevo proyecto de mikroC realice la programación propuesta en el diagrama de flujo (Figura 106) "*02-MCP4921 PRB*".
- Conecte el programador *PICKIT2* a la tarjeta entrenadora.
- Encienda la tarjeta, compile el código y grabe el dsPIC con la ayuda del software Pickit 2.
- Conecte el osciloscopio en la salida del DAC, teniendo en cuenta la polaridad de la conexión.

Figura 106. Diagrama de flujo “02-MCP4921 PRB” MikroC



Fuente: El autor del taxto.

Observaciones:

En el osciloscopio a la salida se observará una señal con forma de diente de sierra. Con este código se verifica el correcto funcionamiento del convertidor Digital/Análogo.

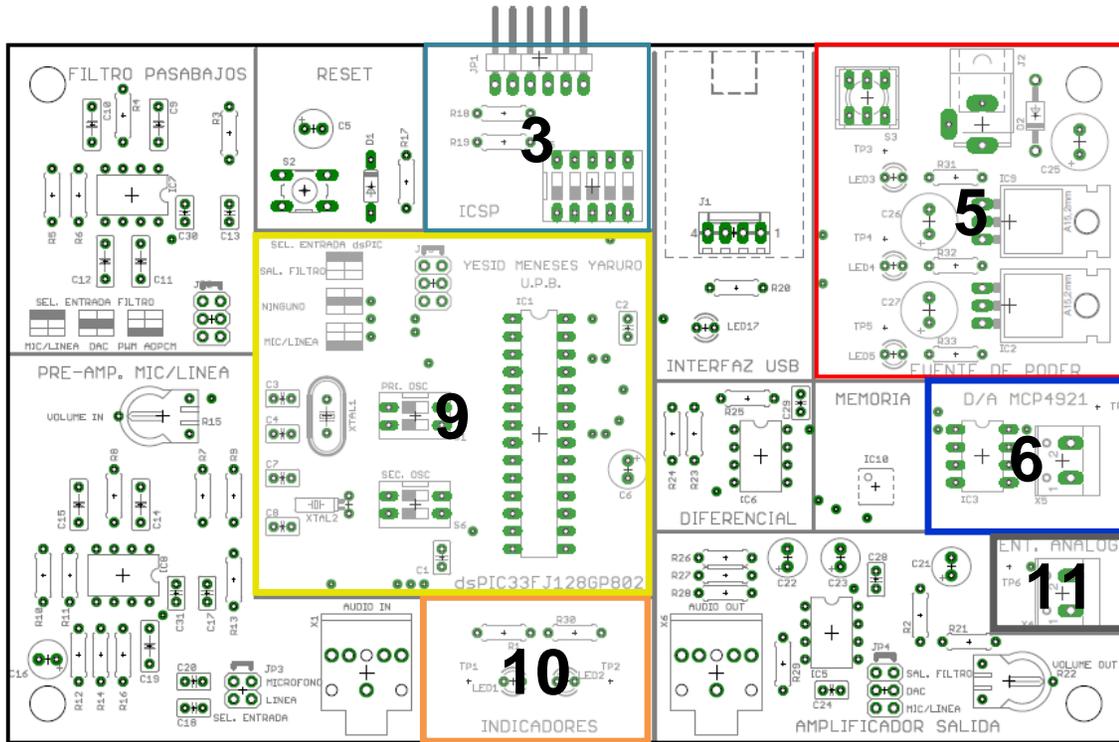
5.2.3. PRÁCTICA No. 3 ADC/DAC LOOPBACK

Objetivo:

Aprender a configurar los periféricos y dispositivos que van a hacer parte de la adquisición, procesamiento y reconstrucción de la señal.

Diagrama de Conexiones:

Figura 107. Entrenador de audio 03-ADC-DAC MikroC

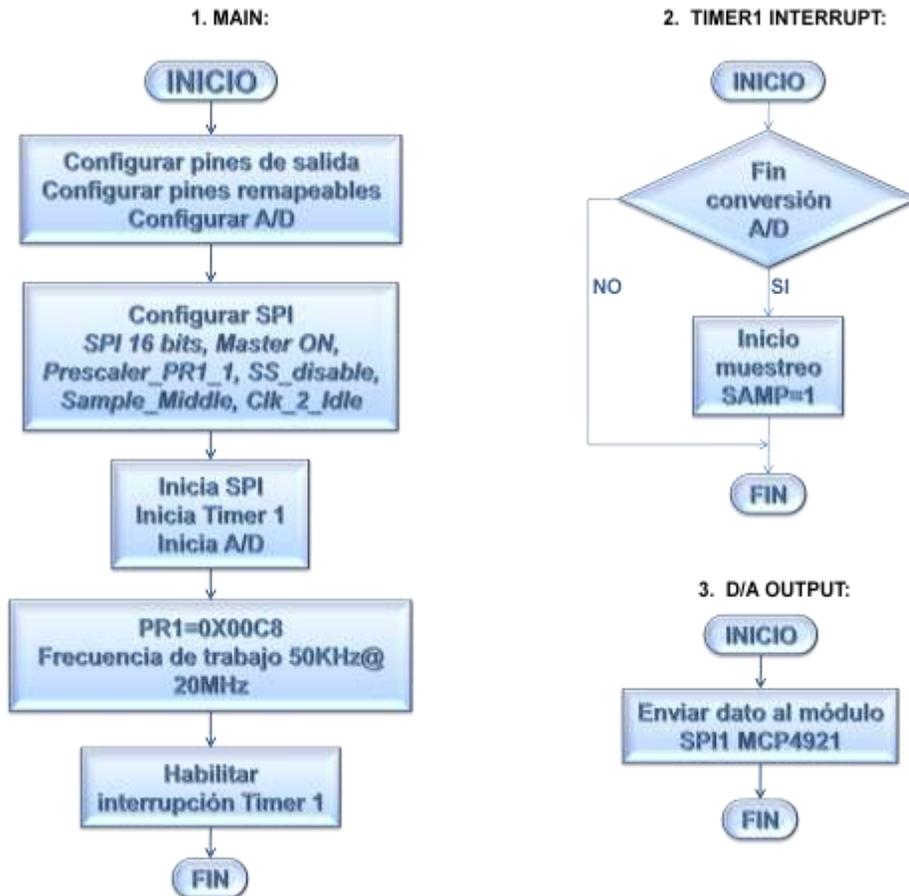


Fuente: Autor del texto.

Procedimiento:

- Abra un nuevo proyecto de mikroC realice la programación propuesta en el diagrama de flujo (Figura 108) "03-ADC-DAC".
- Conecte el programador *PICKIT2* a la tarjeta entrenadora. (bloque 3)
- Encienda la tarjeta, compile el código y grabe el dsPIC.
- Configure el generador de señales para que produzca una onda senoidal de frecuencia 2KHz, con OFFSET de 1.5V, y 1.5Vp de amplitud (**ADVERTENCIA:** si se envía la señal sin offset se quemará el microcontrolador).
- Inyecte la señal en el conector marcado como "ENT ANALOG" en la tarjeta, teniendo en cuenta la polaridad del mismo.
- Conecte el osciloscopio en el conector marcado como "D/A MCP4921" en la tarjeta, teniendo en cuenta la polaridad del mismo.
- Habilite el pin MCLR para iniciar el programa.

Figura 108. Diagrama de flujo “03-ADC-DAC” MikroC



Fuente: Autor del texto.

Observaciones:

El código implementado mostrará cualquier señal a la entrada en la salida del DAC del sistema. Esto tiene la ventaja que realiza la prueba del sistema de adquisición de datos y posterior entrega análoga, verificando con ello el correcto funcionamiento del dsPIC y del convertor D/A.

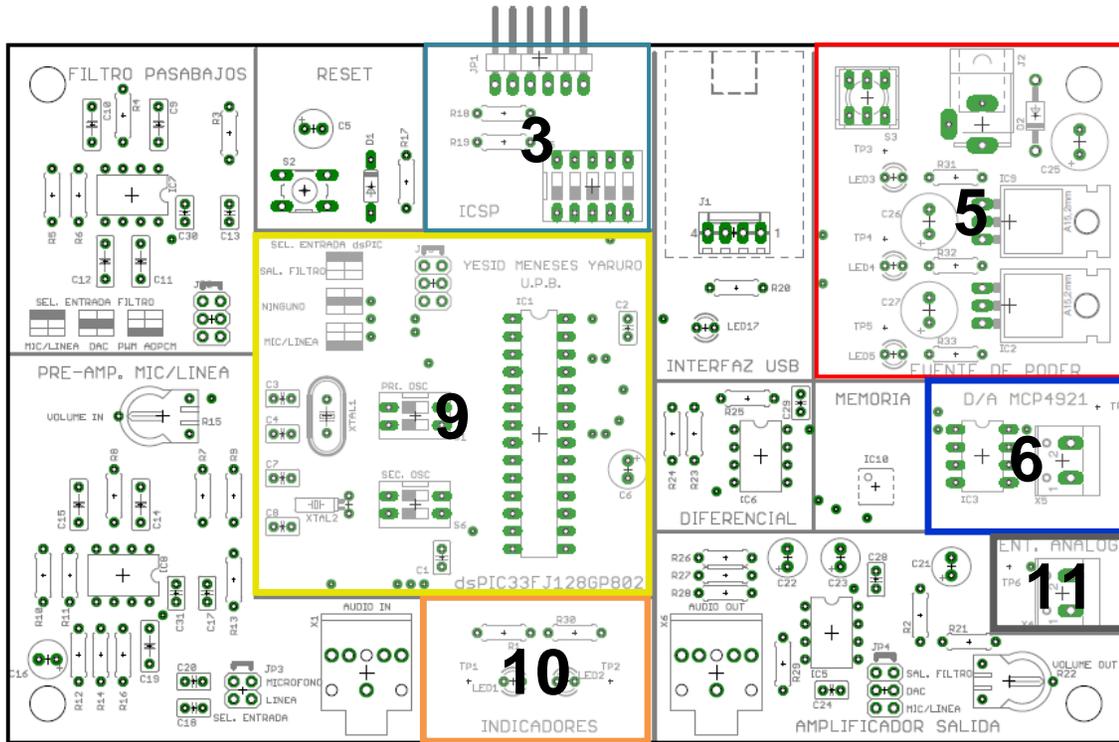
5.2.4. Práctica No. 4 FIR Pasa-alto

Objetivo:

Implementar un filtro FIR, pasa-alto utilizando el compilador mikroC.

Diagrama de Conexiones:

Figura 109. Entrenador de audio *04-FIR PALTO* MikroC

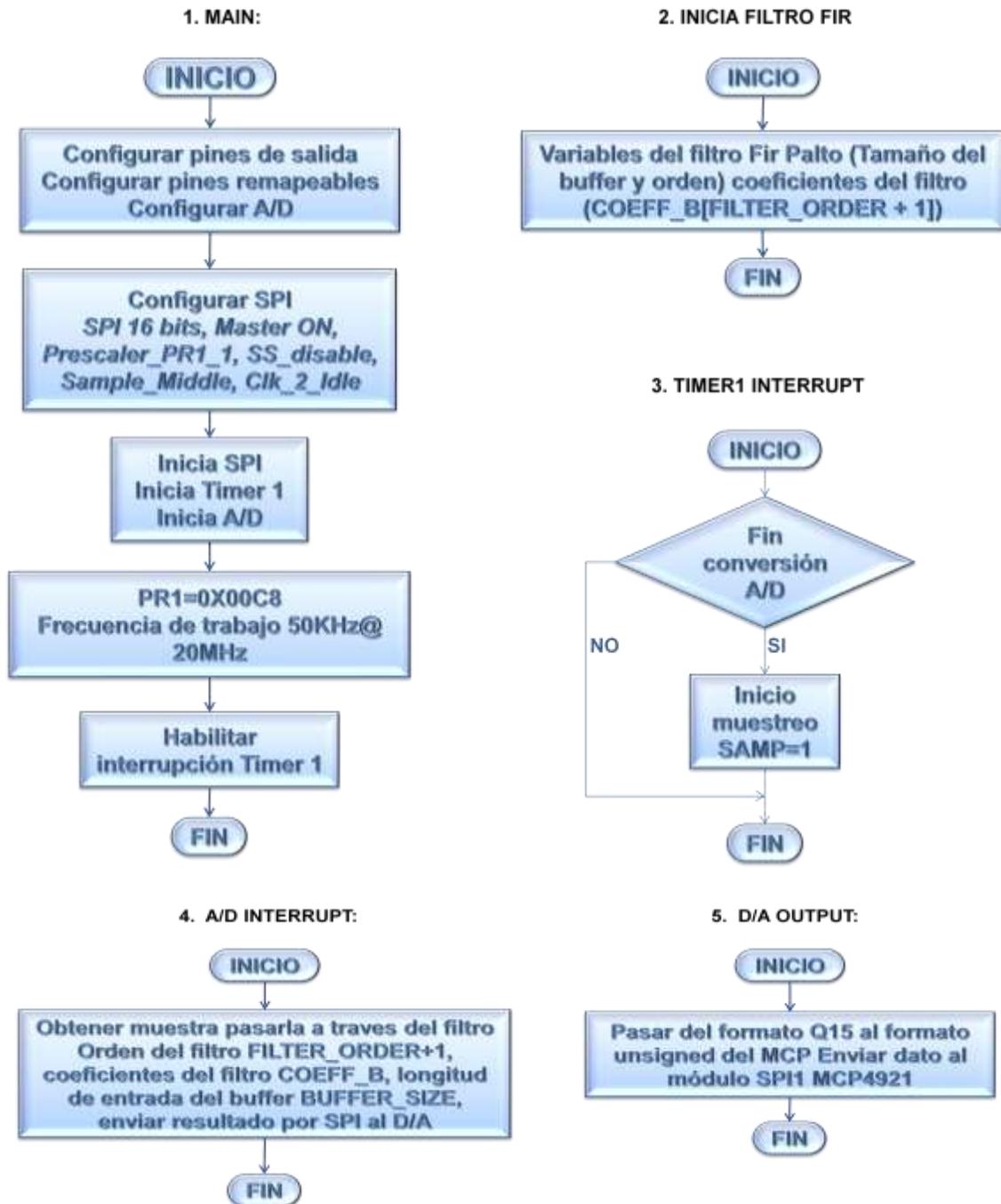


Fuente: Autor del texto.

Procedimiento:

- Abra un nuevo proyecto de mikroC realice la programación propuesta en el diagrama de flujo (Figura 110) “*04-FIR PALTO*”.
- Conecte el programador *PICKIT2* a la tarjeta entrenadora.
- Configure el generador de señales para que produzca una onda senoidal en un rango de 0 a 1.5 KHz, con OFFSET de 1.5V, y 1.5Vp de amplitud (**ADVERTENCIA:** si se envía la señal sin offset, o con polaridad negativa se quemará el microcontrolador).
- Compile el proyecto y programe con la ayuda del software Pickit2.
- Inyecte la señal en el conector marcado como “*ENT ANALOG*” en la tarjeta, teniendo en cuenta la polaridad del mismo.
- Conecte el osciloscopio en el conector marcado como “*D/A MCP4921*” en la tarjeta, teniendo en cuenta la polaridad del mismo.
- Varíe la frecuencia del generador de señales observando la salida en el osciloscopio

Figura 110. Diagrama de flujo "04-FIR PALTO" MikroC



Fuente: Autor del texto.

Observaciones:

Esta práctica pretende implementar un filtro FIR pasa-alto en otro compilador, mostrando así otra herramienta que hace la implementación de un filtro digital. Al incrementarse la frecuencia, se observa la atenuación en la salida del sistema. En el momento que se alcanza la frecuencia configurada de acuerdo a las constantes calculadas en la herramienta, se muestra la frecuencia del generador, iniciando con una leve atenuación para posteriormente alcanzar un grado mayor, dejando pasar la frecuencia alta ingresada.

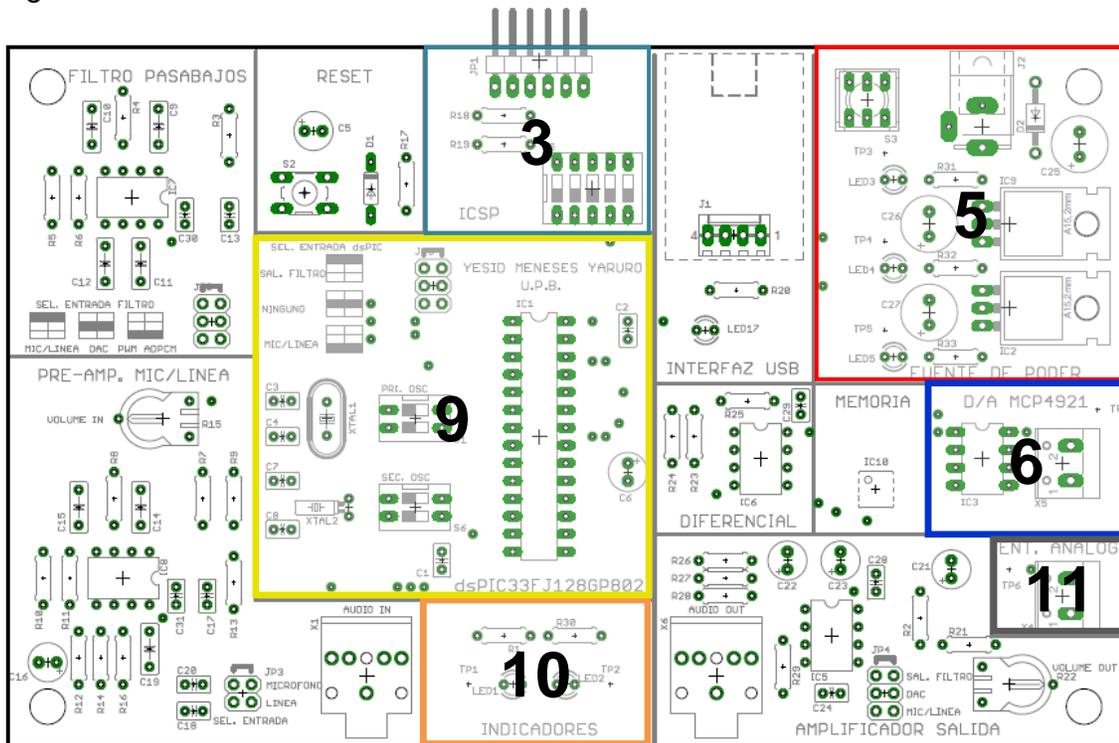
5.2.5. Práctica No. 5 IIR Final

Objetivo:

Implementación de un filtro IIR, de acuerdo a unos parámetros establecidos

Diagrama de Conexiones:

Figura 111. Entrenador de audio "05-IIR TODOS" MikroC



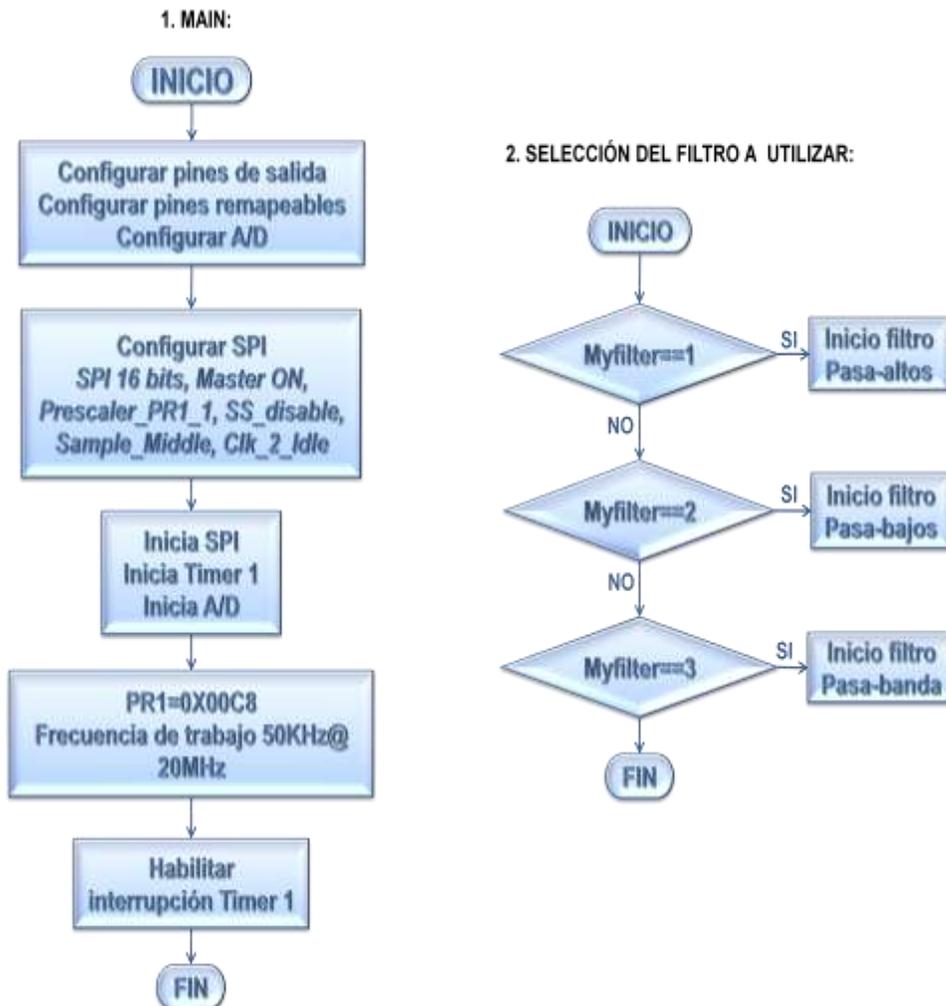
Fuente: Autor del texto.

Procedimiento:

- Abra un nuevo proyecto de MikroC realice la programación propuesta en el diagrama de flujo (Figura 112) "05-IIR TODOS".
- Conecte el programador PICKIT2 a la tarjeta entrenadora.

- c) En el código a compilar, seleccione un tipo de filtro con la opción Filter Designer Tool.
- d) Encienda la tarjeta, compile el código y grabe el dsPIC con la ayuda del software Pickit2.
- e) Configure el generador de señales para que produzca una onda senoidal en un rango de acuerdo al filtro seleccionado, con OFFSET de 1.5V, y 1.5Vp de amplitud (**ADVERTENCIA:** si se envía la señal sin offset, o con polaridad negativa se quemará el microcontrolador).
- f) Inyecte la señal en el conector marcado como “ENT ANALOG” en la tarjeta, teniendo en cuenta la polaridad del mismo.
- g) Conecte el osciloscopio en el conector marcado como “D/A MCP4921” en la tarjeta, teniendo en cuenta la polaridad del mismo.
- h) Varíe la frecuencia del generador de señales observando la salida del sistema en el osciloscopio

Figura 112 Diagrama de flujo “07-IIR TODOS” MikroC

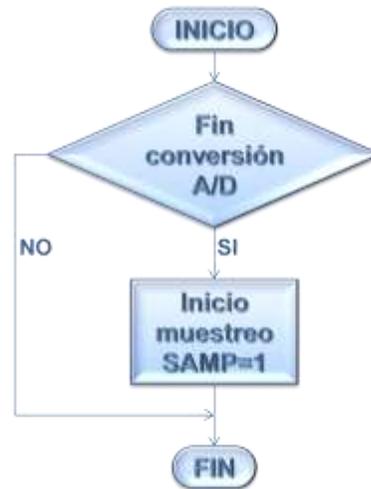


Fuente: Autor del texto.

3. INICIA FILTRO IIR



4. TIMER1 INTERRUPT



5. A/D INTERRUPT:



6. D/A OUTPUT:



Fuente: Autor del texto.

Observaciones:

De acuerdo al filtro seleccionado en MYFILTER se podrán observar diferentes frecuencias de corte y paso. Este algoritmo en mikroC, tiene una frecuencia de muestreo de 20KHz.

NOTA: La solución de las prácticas propuestas se encuentran en los anexos del libro.

RECOMENDACIONES

- Para los futuros proyectos basados en dsPIC33F, revisar las opciones de aplicaciones especializadas que ofrece el fabricante. En la actualidad existe un grupo de procesadores de propósito general (GP), uno para el control de motores (MC) y otro para fuentes de poder digitales (GS).
- Promover entre la comunidad académica de la Universidad Pontificia de Bucaramanga el empleo de dsPIC en proyectos que impliquen de tratamiento digital de señales.
- Proveer el laboratorio de electrónica de la Facultad de Ingenierías de módulos entrenadores similares al desarrollado en este proyecto para que sean usados como apoyo en el proceso de enseñanza de asignaturas como control digital y microcontroladores.
- Tomar como base los alcances del presente proyecto para continuar la línea de investigación en audio, desarrollando aplicaciones como: cancelación de eco, ecualizadores, control automático de ganancia (AGC), codificación y decodificación de audio.

CONCLUSIONES

- La invención de nuevas tecnologías en electrónica y la masificación de las mismas, permiten que la implementación de filtros digitales sea cada vez más sencilla, económica y práctica que la de filtros analógicos.
- El proceso de digitalización y tratamiento de la señal de audio utilizado para este proyecto es válido para todos los tipos de señales, solo hay que tener en cuenta que la velocidad de procesamiento de la tarjeta desarrollada logre soportar el ancho de banda requerido por la señal a procesar, que según el teorema de NYQUIST debe ser de al menos dos veces la frecuencia máxima.
- El desarrollo de aplicaciones con C30 presenta un esfuerzo inicial mientras se adquiere el dominio del compilador, pero los beneficios que ofrece trabajar con el compilador del fabricante se ven representados principalmente en la optimización del código generado y el reconocimiento por parte del diseñador de la arquitectura del procesador utilizado.
- La implementación de la herramienta didáctica para el aprendizaje de filtros digitales constituye un elemento valioso en el desarrollo de prácticas y para la comparación de los diferentes métodos usados en el cálculo de los coeficientes de los filtros.
- La integración de periféricos análogos en el mismo encapsulado de los procesadores y la implementación de tecnologías de bajo consumo de corriente, han generado que se fabriquen dispositivos cada vez más pequeños y portables como los teléfonos celulares, ipods, cámaras fotográficas entre otros.
- El compilador MikroC para dsPICs es una herramienta práctica y útil cuando se requiere la elaboración de un firmware para un modelo antiguo de procesador, pero se vuelve complejo e incierto para los modelos más recientes.

BIBLIOGRAFÍA

- [1] OGATA Katsuhiko. Sistemas de Control en Tiempo Discreto. Segunda Edición. Pearson Educación. Pág. 135-136. 1996.
- [2] BAUTISTA, Ricardo Valerio. Matlab y El Diseño de Filtros Digitales. Revista Digital "Investigación y Educación" No. 19. Vol II. Sep. 2005.
- [3] Procesamiento Digital de Señales. Introducción a los Filtros Digitales – Filtros FIR. Universidad Nacional de Córdoba.
- [4] PEREZ, Carlos Andrés. TOSCANO, Diana Patricia. Tesis: "EFECTO DE LA PALABRA FINITA EN LA IMPLEMENTACIÓN DE FILTRO DIGITAL". Universidad Pontificia Bolivariana. 2008
- [5] Tesis de Graduación. García Castillo Francisco J. Procesamiento Digital de Señales o DSP. 2007
- [6] TOVAR, Cesar Armando. Tesis "Implementación de Filtros Digitales Utilizando PC". Universidad Industrial de Santander. 2005.
- [7] SOBRINO, José, Teledetección, Ed. Universidad de Valencia, 2000. 468p
- [8] OROPEZA, Rodríguez José Luis, Computación y sistemas Vol. 9, 2006
- [9] OPPENHEIM, Alan y otros, Señales y sistemas. Ed. Pearson, 1998. 956p.
- [10] ANGULO, Usategui José y otros, Microcontroladores avanzados dsPIC, Controladores digitales de señales arquitectura, programación y aplicaciones. Ed Thomson, 2006
- [11] ANGULO, Usategui José y otros, dsPIC Diseño Práctico de Aplicaciones. Mc Graw Hill, 2007
- [12] GARCIA Lopez Juan, Procesamiento Digital de Señales, Martínez García Moreno, Ciro Andrés, 2007
- [13] MILIVOJEVIC Zorann, Digital Filter Design, Mikroelektronika
- [14] LEIS, Laboratorio de electrónica, investigación y servicios, Procesamiento Digital de Señales, Universidad Nacional de San Luis, Argentina <http://dsp.materia.unsl.edu.ar>
- [15] <http://www.fwdconcepts.com/dsp5409.htm>; noviembre 2010

[16] http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2629¶m=en533465; noviembre 2010

[17] <http://www.mikroe.com/en/compilers/mikroc/pro/dspic/>; noviembre 2010

[18] <http://www.microchip.com/>; noviembre 2010

ANEXOS

ANEXO A. Características y Diagrama de Pines dsPIC33FJ128GP802



dsPIC33FJ32GP302/304, dsPIC33FJ64GPX02/X04, AND dsPIC33FJ128GPX02/X04

High-Performance, 16-Bit Digital Signal Controllers

Operating Range:

- Up to 40 MIPS operation (at 3.0-3.6V):
 - Industrial temperature range (-40°C to +85°C)
 - Extended temperature range (-40°C to +125°C)
- Up to 20 MIPS operation (at 3.0-3.6V):
 - High temperature range (-40°C to +140°C)

High-Performance DSC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set
- 16-bit wide data path
- 24-bit wide instructions
- Linear program memory addressing up to 4M instruction words
- Linear data memory addressing up to 64 Kbytes
- 83 base instructions: mostly 1 word/1 cycle
- Two 40-bit accumulators with rounding and saturation options
- Flexible and powerful addressing modes:
 - Indirect
 - Modulo
 - Bit-Reversed
- Software stack
- 16 x 16 fractional/integer multiply operations
- 32/16 and 16/16 divide operations
- Single-cycle multiply and accumulate:
 - Accumulator write back for DSP operations
 - Dual data fetch
- Up to ± 16 -bit shifts for up to 40-bit data

Direct Memory Access (DMA):

- 8-channel hardware DMA
- Up to 2 Kbytes dual ported DMA buffer area (DMA RAM) to store data transferred via DMA:
 - Allows data transfer between RAM and a peripheral while CPU is executing code (no cycle stealing)
- Most peripherals support DMA

Timers/Capture/Compare/PWM:

- Timer/Counters, up to five 16-bit timers:
 - Can pair up to make two 32-bit timers
 - One timer runs as a Real-Time Clock with an external 32.768 kHz oscillator
 - Programmable prescaler
- Input Capture (up to four channels):
 - Capture on up, down or both edges
 - 16-bit capture input functions
 - 4-deep FIFO on each capture
- Output Compare (up to four channels):
 - Single or Dual 16-bit Compare mode
 - 16-bit Glitchless PWM mode
- Hardware Real-Time Clock/Calendar (RTCC):
 - Provides clock, calendar and alarm functions

Interrupt Controller:

- 5-cycle latency
- Up to 49 available interrupt sources
- Up to three external interrupts
- Seven programmable priority levels
- Five processor exceptions

Digital I/O:

- Peripheral pin Select functionality
- Up to 35 programmable digital I/O pins
- Wake-up/Interrupt-on-Change for up to 31 pins
- Output pins can drive from 3.0V to 3.6V
- Up to 5V output with open drain configuration
- All digital input pins are 5V tolerant
- 4 mA sink on all I/O pins

On-Chip Flash and SRAM:

- Flash program memory (up to 128 Kbytes)
- Data SRAM (up to 16 Kbytes)
- Boot, Secure and General Security for program Flash

dsPIC33FJ32GP302/304, dsPIC33FJ64GPX02/X04, AND dsPIC33FJ128GPX02/X04

System Management:

- Flexible clock options:
 - External, crystal, resonator, internal RC
 - Fully integrated Phase-Locked Loop (PLL)
 - Extremely low jitter PLL
- Power-up Timer
- Oscillator Start-up Timer/Stabilizer
- Watchdog Timer with its own RC oscillator
- Fail-Safe Clock Monitor
- Reset by multiple sources

Power Management:

- On-chip 2.5V voltage regulator
- Switch between clock sources in real time
- Idle, Sleep, and Doze modes with fast wake-up

Analog-to-Digital Converters (ADCs):

- 10-bit, 1.1 Msps or 12-bit, 500 ksps conversion:
 - Two and four simultaneous samples (10-bit ADC)
 - Up to 13 input channels with auto-scanning
 - Conversion start can be manual or synchronized with one of four trigger sources
 - Conversion possible in Sleep mode
 - ± 2 LSB max integral nonlinearity
 - ± 1 LSB max differential nonlinearity

Audio Digital-to-Analog Converter (DAC):

- 16-bit Dual Channel DAC module
- 100 ksps maximum sampling rate
- Second-Order Digital Delta-Sigma Modulator

Data Converter Interface (DCI) module:

- Codec interface
- Supports I²S and AC'97 protocols
- Up to 16-bit data words, up to 16 words per frame
- 4-word deep TX and RX buffers

Comparator Module:

- Two analog comparators with programmable input/output configuration

CMOS Flash Technology:

- Low-power, high-speed Flash technology
- Fully static design
- 3.3V ($\pm 10\%$) operating voltage
- Industrial and Extended temperature
- Low power consumption

Communication Modules:

- 4-wire SPI (up to two modules):
 - Framing supports I/O interface to simple codecs
 - Supports 8-bit and 16-bit data
 - Supports all serial clock formats and sampling modes
- I²C™:
 - Full Multi-Master Slave mode support
 - 7-bit and 10-bit addressing
 - Bus collision detection and arbitration
 - Integrated signal conditioning
 - Slave address masking
- UART (up to two modules):
 - Interrupt on address bit detect
 - Interrupt on UART error
 - Wake-up on Start bit from Sleep mode
 - 4-character TX and RX FIFO buffers
 - LIN bus support
 - IrDA® encoding and decoding in hardware
 - High-Speed Baud mode
 - Hardware Flow Control with CTS and RTS
- Enhanced CAN (ECAN™ module) 2.0B active:
 - Up to eight transmit and up to 32 receive buffers
 - 16 receive filters and three masks
 - Loopback, Listen Only and Listen All
 - Messages modes for diagnostics and bus monitoring
 - Wake-up on CAN message
 - Automatic processing of Remote Transmission Requests
 - FIFO mode using DMA
 - DeviceNet™ addressing support
- Parallel Master Slave Port (PMP/EPSP):
 - Supports 8-bit or 16-bit data
 - Supports 16 address lines
- Programmable Cyclic Redundancy Check (CRC):
 - Programmable bit length for the CRC generator polynomial (up to 16-bit length)
 - 8-deep, 16-bit or 16-deep, 8-bit FIFO for data input

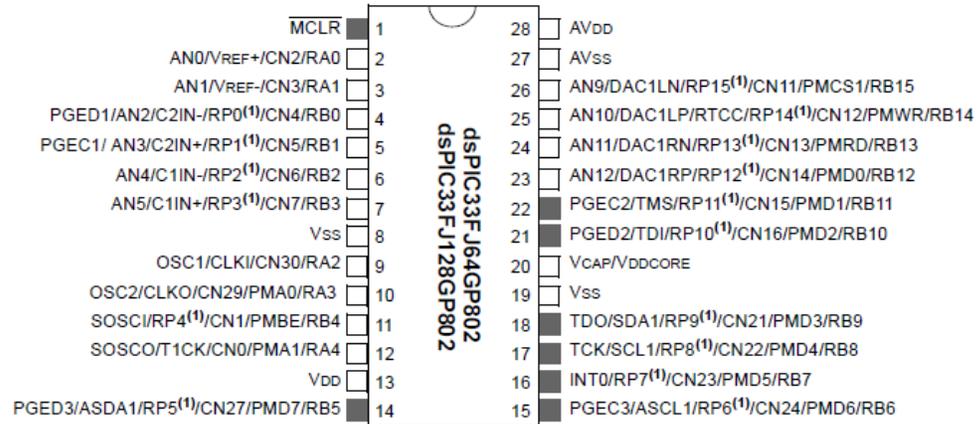
Packaging:

- 28-pin SDIP/SOIC/QFN-S
- 44-pin TQFP/QFN

Note: See the device variant tables for exact peripheral features per device.

28-Pin SDIP, SOIC

■ = Pins are up to 5V tolerant



Anexo A. Convertidor Digital-Análogo MCP4921



MICROCHIP MCP4901/4911/4921

8/10/12-Bit Voltage Output Digital-to-Analog Converter with SPI Interface

Features

- MCP4901: 8-Bit Voltage Output DAC
- MCP4911: 10-Bit Voltage Output DAC
- MCP4921: 12-Bit Voltage Output DAC
- Rail-to-Rail Output
- SPI Interface with 20 MHz Clock Support
- Simultaneous Latching of the DAC Output with $\overline{\text{LDAC}}$ Pin
- Fast Settling Time of 4.5 μs
- Selectable Unity or 2x Gain Output
- External Voltage Reference Input
- External Multiplier Mode
- 2.7V to 5.5V Single-Supply Operation
- Extended Temperature Range: -40°C to $+125^{\circ}\text{C}$

Applications

- Set Point or Offset Trimming
- Precision Selectable Voltage Reference
- Motor Control Feedback Loop
- Digitally-Controlled Multiplier/Divider
- Calibration of Optical Communication Devices

Related Products

P/N	DAC Resolution	No. of Channels	Voltage Reference (V_{REF})
MCP4801	8	1	Internal (2.048V)
MCP4811	10	1	
MCP4821	12	1	
MCP4802	8	2	
MCP4812	10	2	External
MCP4822	12	2	
MCP4901	8	1	
MCP4911	10	1	
MCP4921	12	1	
MCP4902	8	2	
MCP4912	10	2	
MCP4922	12	2	

Note: The products listed here have similar AC/DC performances.

Description

The MCP4901/4911/4921 devices are single channel 8-bit, 10-bit and 12-bit buffered voltage output Digital-to-Analog Converters (DACs), respectively. The devices operate from a single 2.7V to 5.5V supply with an SPI compatible Serial Peripheral Interface. The user can configure the full-scale range of the device to be V_{REF} or $2 \cdot V_{\text{REF}}$ by setting the gain selection option bit (gain of 1 of 2).

The user can shut down the device by setting the Configuration Register bit. In Shutdown mode, most of the internal circuits are turned off for power savings, and the output amplifier is configured to present a known high resistance output load (500 k Ω , typical).

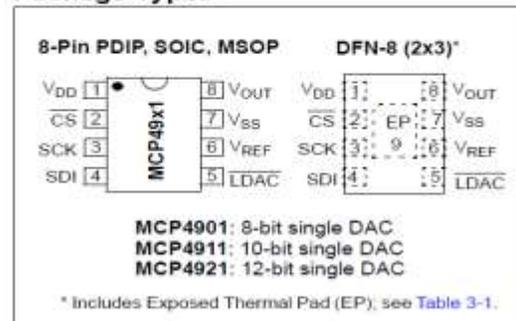
The devices include double-buffered registers, allowing synchronous updates of the DAC output using the $\overline{\text{LDAC}}$ pin. These devices also incorporate a Power-on Reset (POR) circuit to ensure reliable power-up.

The devices utilize a resistive string architecture, with its inherent advantages of low Differential Non-Linearity (DNL) error and fast settling time. These devices are specified over the extended temperature range ($+125^{\circ}\text{C}$).

The devices provide high accuracy and low noise performance for consumer and industrial applications where calibration or compensation of signals (such as temperature, pressure and humidity) are required.

The MCP4901/4911/4921 devices are available in the PDIP, SOIC, MSOP and DFN packages.

Package Types



ANEXO B. Amplificador Operacional MCP6021



MCP6021/1R/2/3/4

Rail-to-Rail Input/Output, 10 MHz Op Amps

Features

- Rail-to-Rail Input/Output
- Wide Bandwidth: 10 MHz (typical)
- Low Noise: 8.7 nV/√Hz, at 10 kHz (typical)
- Low Offset Voltage:
 - Industrial Temperature: ±500 μV (maximum)
 - Extended Temperature: ±250 μV (maximum)
- Mid-Supply V_{REF} : MCP6021 and MCP6023
- Low Supply Current: 1 mA (typical)
- Total Harmonic Distortion:
 - 0.00053% (typical, $G = 1$ V/V)
- Unity Gain Stable
- Power Supply Range: 2.5V to 5.5V
- Temperature Range:
 - Industrial: -40°C to +85°C
 - Extended: -40°C to +125°C

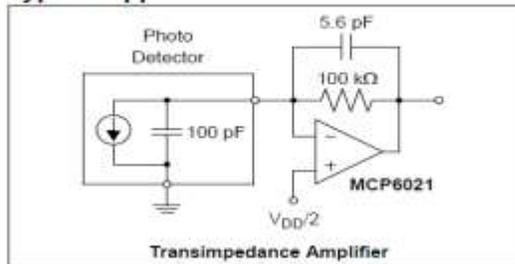
Applications

- Automotive
- Multi-Pole Active Filters
- Audio Processing
- DAC Buffer
- Test Equipment
- Medical Instrumentation

Design Aids

- SPICE Macro Models
- FilterLab® Software
- Mindi™ Circuit Designer & Simulator
- Microchip Advanced Part Selector (MAPS)
- Analog Demonstration and Evaluation Boards
- Application Notes

Typical Application



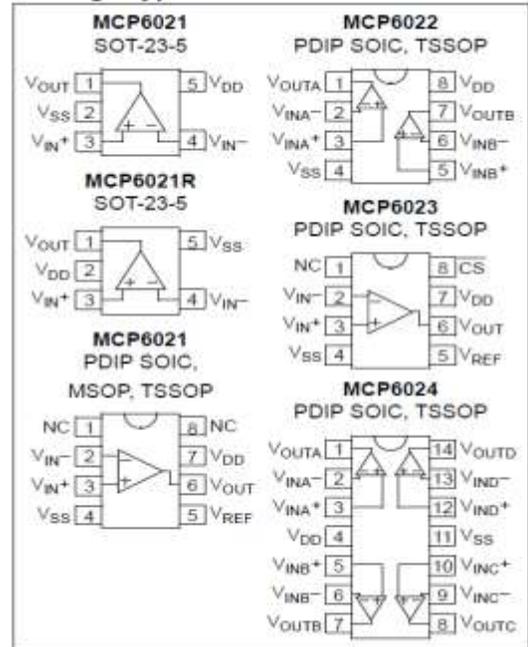
Description

The MCP6021, MCP6021R, MCP6022, MCP6023 and MCP6024 from Microchip Technology Inc. are rail-to-rail input and output op amps with high performance. Key specifications include: wide bandwidth (10 MHz), low noise (8.7 nV/√Hz), low input offset voltage and low distortion (0.00053% THD+N). The MCP6023 also offers a Chip Select pin (\overline{CS}) that gives power savings when the part is not in use.

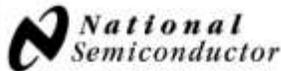
The single MCP6021 and MCP6021R are available in SOT-23-5. The single MCP6021, single MCP6023 and dual MCP6022 are available in 8-lead PDIP, SOIC and TSSOP. The Extended Temperature single MCP6021 is available in 8-lead MSOP. The quad MCP6024 is offered in 14-lead PDIP, SOIC and TSSOP packages.

The MCP6021/1R/2/3/4 family is available in Industrial and Extended temperature ranges. It has a power supply range of 2.5V to 5.5V.

Package Types



ANEXO C. Regulador de Voltaje LM1117- 3.3V



April 2006

LM1117/LM1117I 800mA Low-Dropout Linear Regulator

General Description

The LM1117 is a series of low dropout voltage regulators with a dropout of 1.2V at 800mA of load current. It has the same pin-out as National Semiconductor's industry standard LM317.

The LM1117 is available in an adjustable version, which can set the output voltage from 1.25V to 13.8V with only two external resistors. In addition, it is also available in five fixed voltages, 1.8V, 2.5V, 2.85V, 3.3V, and 5V.

The LM1117 offers current limiting and thermal shutdown. Its circuit includes a zener trimmed bandgap reference to assure output voltage accuracy to within $\pm 1\%$.

The LM1117 series is available in LLP, TO-263, SOT-223, TO-220, and TO-252 D-PAK packages. A minimum of 10 μ F tantalum capacitor is required at the output to improve the transient response and stability.

Features

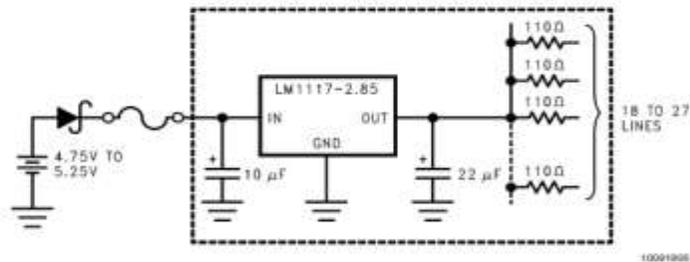
- Available in 1.8V, 2.5V, 2.85V, 3.3V, 5V, and Adjustable Versions
- Space Saving SOT-223 and LLP Packages
- Current Limiting and Thermal Protection
- Output Current 800mA
- Line Regulation 0.2% (Max)
- Load Regulation 0.4% (Max)
- Temperature Range
 - LM1117 0°C to 125°C
 - LM1117I -40°C to 125°C

Applications

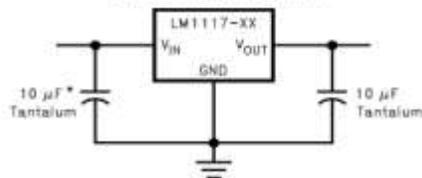
- 2.85V Model for SCSI-2 Active Termination
- Post Regulator for Switching DC/DC Converter
- High Efficiency Linear Regulators
- Battery Charger
- Battery Powered Instrumentation

Typical Application

Active Terminator for SCSI-2 Bus



Fixed Output Regulator



* Required if the regulator is located far from the power supply filter.

LM1117/LM1117I 800mA Low-Dropout Linear Regulator

ANEXO D. Regulador de Voltaje LM7805 – 5 V



www.fairchildsemi.com

MC78XX/LM78XX/MC78XXA

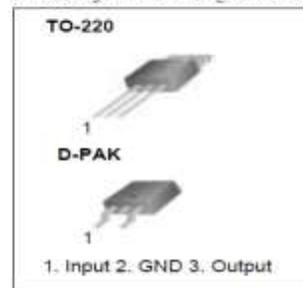
3-Terminal 1A Positive Voltage Regulator

Features

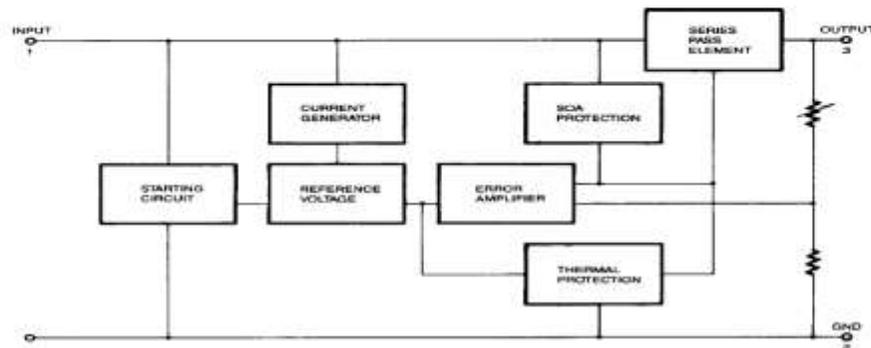
- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The MC78XX/LM78XX/MC78XXA series of three terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



Internal Block Diagram



Rev. 1.0.1

Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Input Voltage (for $V_O = 5V$ to $18V$) (for $V_O = 24V$)	V_I	35	V
	V_I	40	V
Thermal Resistance Junction-Cases (TO-220)	$R_{\theta JC}$	5	$^{\circ}C/W$
Thermal Resistance Junction-Air (TO-220)	$R_{\theta JA}$	65	$^{\circ}C/W$
Operating Temperature Range	T_{OPR}	0 ~ +125	$^{\circ}C$
Storage Temperature Range	T_{STG}	-65 ~ +150	$^{\circ}C$

Electrical Characteristics (MC7805/LM7805)

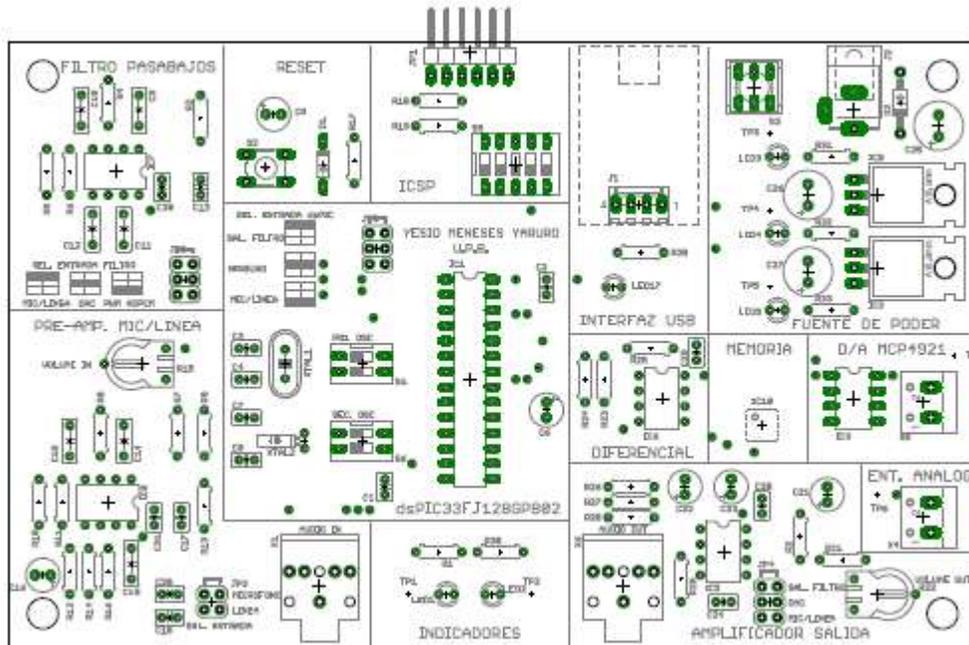
(Refer to test circuit, $0^{\circ}C < T_J < 125^{\circ}C$, $I_O = 500mA$, $V_I = 10V$, $C_I = 0.33\mu F$, $C_O = 0.1\mu F$, unless otherwise specified)

Parameter	Symbol	Conditions	MC7805/LM7805			Unit	
			Min.	Typ.	Max.		
Output Voltage	V_O	$T_J = +25^{\circ}C$	4.8	5.0	5.2	V	
		$5.0mA \leq I_O \leq 1.0A$, $P_O \leq 15W$ $V_I = 7V$ to $20V$	4.75	5.0	5.25		
Line Regulation (Note 1)	Regline	$T_J = +25^{\circ}C$	$V_O = 7V$ to $25V$	-	4.0	100	mV
			$V_I = 8V$ to $12V$	-	1.6	50	
Load Regulation (Note 1)	Regload	$T_J = +25^{\circ}C$	$I_O = 5.0mA$ to $1.5A$	-	9	100	mV
			$I_O = 250mA$ to $750mA$	-	4	50	
Quiescent Current	I_Q	$T_J = +25^{\circ}C$	-	5.0	8.0	mA	
Quiescent Current Change	ΔI_Q	$I_O = 5mA$ to $1.0A$	-	0.03	0.5	mA	
		$V_I = 7V$ to $25V$	-	0.3	1.3		
Output Voltage Drift	$\Delta V_O / \Delta T$	$I_O = 5mA$	-	-0.8	-	mV/ $^{\circ}C$	
Output Noise Voltage	V_N	$f = 10Hz$ to $100KHz$, $T_A = +25^{\circ}C$	-	42	-	$\mu V/V_O$	
Ripple Rejection	RR	$f = 120Hz$ $V_O = 8V$ to $18V$	62	73	-	dB	
Dropout Voltage	V_{Drop}	$I_O = 1A$, $T_J = +25^{\circ}C$	-	2	-	V	
Output Resistance	r_O	$f = 1KHz$	-	15	-	$m\Omega$	
Short Circuit Current	I_{SC}	$V_I = 35V$, $T_A = +25^{\circ}C$	-	230	-	mA	
Peak Current	I_{PK}	$T_J = +25^{\circ}C$	-	2.2	-	A	

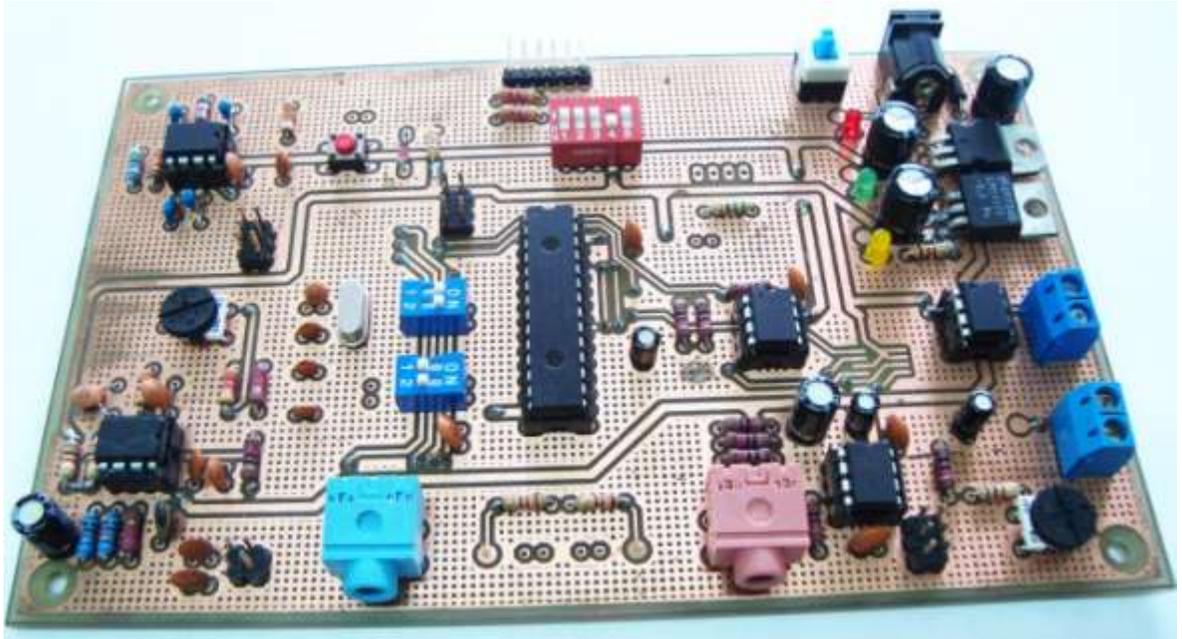
Note:

1. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

ANEXO E. Diseño Eagle



ANEXO F. Tarjeta Prototipo



ANEXO G. Tarjeta Final

