

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA
AUTOMATIZACION Y CONTROL

PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO
DE INGENIERÍA

DISEÑO E IMPLEMENTACION DE UN PROTOTIPO DE UN
AFINADOR DIGITAL DE INSTRUMENTOS MUSICALES CON
DSPIC30F3014

HELENA DOLORES IBARRA RUIZ

Sangolquí – Ecuador
2007

CERTIFICACION

Certificamos que el presente proyecto de grado fue realizado en su totalidad por el Srta. Helena Dolores Ibarra Ruiz bajo nuestra dirección.

Ing. Julio Larco
DIRECTOR

Ing Víctor Proaño
CODIRECTOR

DEDICATORIA

*A mi madre, que aunque no se encuentre
con nosotros , sus enseñanzas
siempre serán mi guía.*

AGRADECIMIENTOS

A mi padre y hermanos por su apoyo incondicional.

Al Ing. Julio Larco por toda la ayuda que me brindó durante el desarrollo de este proyecto.

Al Ing. Byron Navas por todo el conocimiento que compartió con sus alumnos durante el tiempo que impartió sus clases.

A mis amigos que siempre estuvieron a mi lado cuando necesité su ayuda y consejos.

PROLOGO

Hoy en día existen en el mercado productos portátiles para la afinación, que en su gran mayoría tiene funciones básicas como: pantalla LCD, bocina interna que emite sonido o led's de referencia para afinar, auto power off, jack de entrada y salida para ingreso de la señal de audio. Existe también software para computadora que tiene las funciones de un afinador con características similares a los afinadores portátiles, pero con la desventaja, que se hace necesario el uso de un computador. El objetivo de este proyecto es construir un afinador digital basado en un dsPIC, que ofrezca nuevas opciones a los músicos por las características personalizadas que se irían desarrollando de acuerdo a la continuidad que se de a este proyecto.

Un dsPIC es un híbrido innovador, que combina las características de control de un Microcontrolador de 16-bits con la funcionalidad de un procesador digital de señales. Este dispositivo permite trabajar en una plataforma de arquitectura simple, versátil y aplicable a cualquier tipo de sistema. Su arquitectura se ajusta a un esquema de servicio de interrupciones periódicas, capturando datos de múltiples sensores y entradas de control, además comparte datos con otros módulos de control en un sistema distribuido.

Usar una arquitectura relativamente nueva es hoy en día una de las mayores ventajas en cuanto al desarrollo de una aplicación, siendo uno de los últimos avances el uso del dsPIC, estos microcontroladores digitales son utilizados para realizar tareas de control y procesamiento digital de señales. Los DSP's son utilizados para aplicaciones como sistemas de audio, compresión, codificación y filtraje de señales e imágenes.

Este proyecto trata sobre el diseño y construcción de un prototipo de afinador digital de instrumentos musicales de cuerdas con dsPIC30F3014 que tendrá como características principales, un botón de encendido y apagado, dos teclas de control, un LCD para visualización y una entrada de audio, con la capacidad de afinar 4 instrumentos diferentes (guitarra, violín, bajo y chelo), para cada uno de los cuales existirá un menú de selección de cuerdas para su afinación.

Este proyecto está dividido en 7 capítulos, en el primero se abordará toda la teoría musical, dentro de este se encuentran las propiedades del sonido, que es una escala musical y los tipos de escalas más importantes, en el capítulo dos recopilará la teoría de afinación y los tipos de afinadores existentes en el mercado.

El tercer capítulo tratará sobre el dsPIC y su funcionamiento, en el cuarto capítulo encontraremos teoría sobre el cálculo de la FFT y sus algoritmos. En el capítulo cinco se detallará el diseño del afinador digital de instrumentos, en el capítulo seis se podrá encontrar las pruebas realizadas para verificar el funcionamiento del dispositivo diseñado y en el último capítulo tendrá las conclusiones y recomendaciones a las que se llegó después del desarrollo del proyecto.

INDICE DE CONTENIDO

CAPITULO 1	1
TEORÍA MUSICAL	1
1.1 Propiedades del Sonido	1
1.1.1 La altura o tono.....	3
1.1.2 La intensidad.....	4
1.1.3 Amplitud.....	4
1.1.4 Frecuencia.....	5
1.1.5 El timbre o calidad.....	5
1.1.6 Resonancia.....	6
1.2 Altura.....	7
1.3 Armónicos	8
1.3.1 Progresiones armónicas.....	8
1.3.2 Armonía y Estructura.....	9
1.3.3 Ritmo musical.....	9
1.3.4 Pulso y Compás.....	9
1.3.5 Unidades de tiempo más largas.....	11
1.3.6 La Serie Armónica.....	11
1.4 Escalas	12
1.4.1 Los intervalos musicales.....	15
1.4.2 Las escalas musicales.....	15
1.4.2.1 La escala diatónica.....	15
1.4.2.2 La escala cromática.....	16
1.4.2.3 La escala temperada.....	16
1.4.2.4 Otras escalas.....	16
1.4.3 Sonidos armónicos.....	17
CAPITULO 2	19
AFINADORES MUSICALES	19
2.1 Características y funciones de los afinadores.....	19
2.2 Afinadores por Software.....	22
2.3 Afinadores Portátiles	23
CAPITULO 3	25
CARACTERÍSTICAS DEL CONTROLADOR DIGITAL DE SEÑALES dsPIC30F3014	25
3.1 Características principales	27
3.1.1 Características del DSP.....	27
3.1.2 Características de los periféricos.....	29
3.1.3 Características Analógicas.....	29
3.1.4 Características Especiales del Microcontrolador.....	29
3.1.5 Tecnología CMOS.....	30
3.2 Descripción de Módulos.....	30
3.3 Instrucciones y librerías para el procesamiento digital de señales.....	35

CAPITULO 4	38
ALGORITMOS PARA PROCESAMIENTO DIGITAL DE SEÑALES A SER USADOS EN EL AFINADOR	38
4.1 Algoritmos existentes para el cálculo de la FFT	39
4.1.1 Transformada de Fourier	39
4.1.2 Transformada Discreta de Fourier	40
4.1.3. Aplicación de la Fast Fourier Transform (FFT)	42
4.1.4 Método de Interpolación Lineal para el Cálculo de la FFT	43
4.1.5 Remuestreo mediante Diezmado e Interpolación para Cálculo de la FFT	43
4.1.5.1 Reducción de la frecuencia de muestreo para Cálculo de la FFT	44
4.1.5.2 Incremento de la frecuencia de muestreo para el Cálculo de la FFT	44
4.2 Selección del algoritmo apropiado	45
4.3 Código del algoritmo seleccionado en Basic	46
CAPITULO 5	48
DISEÑO DEL PROTOTIPO	48
5.1 Hardware	48
Diagrama de bloques	48
Teclas de control	49
Entrada de audio	50
Conversor A/D	51
Cálculo de la FFT	52
Comparación de la frecuencia calculada con la frecuencia establecida para una nota musical según la Escala Temperada	54
LCD e interfaz gráfica	54
Alimentación	54
Diagrama esquemático del circuito completo para el afinador de instrumentos musicales con el dsPIC30F3014	¡Error! Marcador no definido.
5.2 Software	56
5.2.1 Consideraciones para manejo de MikroBasic para proyectos con DSPIC	56
5.2.2 Diagrama de flujo para el afinador de instrumentos musicales con el dsPIC30F3014	59
5.2.3 Código del programa, explicación	60
5.2.4 Explicación de los algoritmos más importantes	61
5.2.4.1. Algoritmo para funcionamiento de los botones de control	61
5.2.4.2. Algoritmo de conversión Análoga - Digital	62
5.2.4.3 Algoritmo para la presentación en LCD del valor de frecuencia fundamental de la señal de entrada	63
5.2.4.4 Algoritmo de comparación para la frecuencia	66
CAPITULO 6	68
PRUEBAS Y RESULTADOS	68
6.1. Pruebas Realizadas	68
CAPITULO 7	77
CONCLUSIONES Y RECOMENDACIONES	77

7.1. Conclusiones.....	77
7.2. Recomendaciones.....	79
REFERENCIAS BIBLIOGRAFICAS	80
ANEXOS	
DIAGRAMA DE FLUJO SELECCION INSTRUMENTO (BAJO)	83
DIAGRAMA DE FLUJO SELECCION INSTRUMENTO (GUITARRA).....	84
DIAGRAMA DE FLUJO SELECCION INSTRUMENTO (VIOLÍN).....	85
DIAGRAMA DE FLUJO SELECCION INSTRUMENTO (CHELO)	86
DIAGRAMA DE FLUJO SELECCION INSTRUMENTO	87
DIAGRAMA DE FLUJO SELECCION CUERDAS (GUITARRA).....	88
DIAGRAMA DE FLUJO SELECCION CUERDAS (VIOLÍN).....	89
DIAGRAMA DE FLUJO SELECCION CUERDAS (BAJO)	90
DIAGRAMA DE FLUJO SELECCION CUERDAS (CHELO)	91
DIAGRAMA DE FLUJO PARA COMPARACION DE FRECUENCIA	92
PROGRAMA COMPLETO PARA EL AFINADOR DIGITAL DE INSTRUMENTOS MUSICALES CON dsPIC30F3014	93
MANUAL DEL USUARIO	110

INDICE DE FIGURAS

CAPITULO 1

Figura: 1.1. Onda Sinusoidal	4
Figura: 1.2. La Serie Armónica	12
Figura: 1.3. Ejemplo de diferentes escalas	17
Figura: 1.4.....	18
Figura: 1.5.....	18

CAPITULO 2

Figura: 2.1. OOBOX de PIXSYS	22
Figura: 2.2. In - Tune de Joseph Heimiller.....	23
Figura: 2.3. Afinador Portátil Yamaha	24
Figura: 2.4. Afinador Portátil Seiko	24
Figura: 2.5. Afinador Portátil Korg	24

CAPITULO 3

Figura: 3.1. Diagrama General de pines del microcontrolador dsPIC30F3014	25
Figura: 3.2. Diagrama de Bloques del dsPIC30F3014	26
Figura: 3.3. Diagrama de Bloques del módulo DSP.....	28
Figura: 3.4. Diagrama de bloques de la estructura de un puerto compartido	31
Figura: 3.5. Diagrama de bloques del modulo A/D.....	34

CAPITULO 4

Figura: 4.1. Diagrama de bloques del diezrado por un factor M.....	44
Figura: 4.2. Diagrama de bloques de la interpolación por un factor L.....	45

CAPITULO 5

Figura: 5.1. Diagrama de bloques.....	48
Figura: 5.2. Teclas de control del Afinador.....	49
Figura: 5.3. Diagrama de la Entrada de Audio	50
Figura: 5.4. Diagrama para la conversión A/D.....	51
Figura: 5.5. Ejemplo de una señal de entrada (seno (x))	53
Figura: 5.6. Decimación en el tiempo	53
Figura: 5.7. Pantalla de inicio de mikroBasic compiler for dsPIC30/33 and PIC24.....	57
Figura: 5.8. Definición de los parámetros básicos para la programación del dsPIC30Fxxxx	58

ANEXOS

Figura: 1. Diagrama del Afinador digital de instrumentos musicales	110
--	-----

INDICE DE TABLAS

CAPITULO 1

Tabla 1.1. Notas Armoniosas	13
Tabla 1.2. Ejemplo de las notas siguientes a Do	13
Tabla 1.3. Himno a San Juan.....	14
Tabla 1.4. Notas Musicales	14
Tabla 1.5. Escala Musical.....	16

CAPITULO 3

Tabla 3.1. Mapa de Registros de los Puertos E/S.....	¡Error! Marcador no definido.
--	--------------------------------------

CAPITULO 5

Tabla 5.1. Correspondencia de frecuencias del instrumento original y el dsPIC.....	64
--	----

CAPITULO 6

Tabla 6.1. Prueba Tipo A.....	69
Tabla 6.2. Prueba Tipo A.....	70
Tabla 6.3. Valor experimental medido vs Valor deseado.....	70
Tabla 6.4. Prueba Tipo B (Guitarra Acústica).....	71
Tabla 6.5. Prueba Tipo D.....	71
Tabla 6.6. Prueba Tipo D.....	72
Tabla 6.7. Prueba Tipo D.....	72
Tabla 6.8. Prueba Tipo D.....	72
Tabla 6.9. Prueba Tipo D.....	73
Tabla 6.10. Prueba Tipo D.....	73
Tabla 6.11. Valor medido experimental vs Valor deseado.....	74
Tabla 6.12. Prueba Tipo C.....	74
Tabla 6.13. Prueba Tipo D.....	74
Tabla 6.14. Prueba Tipo D.....	75
Tabla 6.15. Prueba Tipo D.....	75
Tabla 6.16. Prueba Tipo D.....	75

GLOSARIO

TERMINO	SIGNIFICADO
ADC	Analog to Digital Convertion – Conversión Análoga Digital.
Afinar	Es la acción de poner en tono justo los instrumentos musicales de acuerdo a un diapasón o ajustarlos bien unos con otros.
Algoritmo	Es una lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema.
Altura	La altura del sonido depende de su frecuencia, es decir, del número de vibraciones por segundo del cuerpo vibrante
Altura	Se entiende por altura de un sonido su calidad de agudo ("alto") o grave ("bajo").
Amplitud	Es la distancia por encima y por debajo de la línea central de la onda de sonido.
Armónicos	Combinación de notas que se emiten simultáneamente.
Arquitectura Harvard	Tiene la unidad central de proceso (CPU) conectada a dos memorias (una con las instrucciones y otra con los datos) por medio de dos buses diferentes.
Bit	
CMOS	Complementary Metal Oxide Semiconductor - Metal Óxido Semiconductor Complementario, es una tecnología utilizada para crear circuitos integrados, los chips CMOS consumen menos potencia y funcionan con tensiones desde los 3 V hasta los 15 V .
CPU	Central Processor Unit – Unidad de procesamiento central
Cuarta	Otro intervalo entre notas que se obtiene con una cuerda de largura tres cuartos de la inicial.
DFT	Discrete Fourier Transform – Transformada Discreta de Fourier.
Diapasón	Es una pieza en forma de U de metal elástico. Cuando se le golpea haciéndolo vibrar, genera una onda sinusoidal casi inaudible. El diapason más utilizado es el llamado la 440 (que significa que genera una nota la ₄ de exactamente 440 Hz
DSC	Digital Signal Controller - Controlador digital de señales.
DSP	Digital Signal Processor - Procesador digital de señales.

dsPIC	Es un DSC y un DSP.
EEPROM	Electrically Erasable Programmable Read-Only Memory.
Escala cromática	12 notas en una octava, separadas por un semitono.
Escala diatónica	Compuesta por semitonos y tonos completos, tienes siete notas por octava.
Escala temperada	Consta también de doce notas, como la cromática, pero la relación de la frecuencia de una nota y la anterior es siempre igual a $\sqrt[12]{2}$.
FFT	Fast Fourier Transform – Transformada Rápida de Fourier
Frase	Una unidad aún más prolongada con una sensación de final más definitiva, que corresponde a grandes trazos a una oración del lenguaje hablado.
Frecuencia	Número de oscilaciones por segundo.
Im	Parte Imaginaria de un número complejo.
Intensidad	La cualidad que permite distinguir entre sonidos fuertes o débiles.
Intervalos musicales	Es la diferencia de tono entre dos notas.
Ksps	Kilo Samples Per Second – Kilo muestras por segundo.
LCD	Liquid Cristal Display – Pantalla de Cristal Líquido.
Loop	Lazo de programación.
MAC	Media Access Control address o dirección de control de acceso al medio
Motivo	La idea melódica más corta que conforma una unidad musical completa relativa.
Nyquist	Se refiere a la frecuencia de Nyquist, que es el valor de frecuencia al cual se muestra la señal de entrada para luego calcular su FFT.
Octava	Es la repetición de un sonido con una cuerda con la mitad de largura.
Pin	Bit es el acrónimo de Binary digit. (dígito binario). Un bit es un dígito del sistema de numeración binario.
PORTx	Puertos de entrada y salida del dsPIC, mayor información en Datasheet del dsPIC30F3014.
Progresión	Es el enlace de un acorde con otro que va generando el movimiento

Armónica	en la música.
PWM	Pulse Width Modulation- Modulación por ancho de Pulso.
Quinta	Es otro intervalo entre notas que se obtiene con una cuerda de largura dos tercios de la inicial.
Re	Parte Real de un número complejo.
RESET	del inglés reponer. Se conoce como reset a la puesta en condiciones iniciales de un sistema.
Resonancia	Cuando dos cuerpos tienen frecuencias iguales y uno de ellos está en vibración, el otro sin ser tocado.
Semitono	Es el intervalo entre dos notas consecutivas de la escala cromática.
Serie Armónica	Es la sucesión de armónicos que se producen al vibrar una cuerda o una columna de aire.
SET	Es un comando para mostrar y asignar valor a las variables de entorno en algunos sistemas operativos.
Síncopa	Un acento que opera en contra del compás establecido.
Sonido	Se produce por la vibración de los cuerpos, el cual se transmite al aire que lo rodea y, a través de éste, llega hasta nuestros oídos.
Sub Function	Son rutinas o subprogramas dentro un programa más grande que desarrolla una cierta tarea basada en un número de parámetros de entrada, una función si retorna un valor.
Sub Procedure	Son rutinas o subprogramas dentro un programa más grande que desarrolla una cierta tarea basada en un número de parámetros de entrada, un procedimiento no retorna ningún valor.
Timbre	Permite distinguir los sonidos producidos por los diferentes instrumentos.

CAPITULO 1

TEORÍA MUSICAL

Las artes se distinguieron entre sí por la diferente materia estructurada por el artista en la creación de su obra. El sonido es una de estas materias, es por eso que la teoría de este arte debe comenzar por el estudio del hecho sonoro y de las diversas formas de su producción.

Una composición, no es solo una suma de sonidos simples, simultáneos o sucesivos, sino que entre ellos se establece una conexión. Por lo que, el primer sonido resultante es el derivado de la relación entre dos sonidos que recibe el nombre de intervalo. A partir de ella se constituyen otras más complejas (escalas, modos, acordes, ritmos, temas, series, etc.) que pueden considerarse como los elementos básicos de la forma musical, y condicionan, por tanto, los principios estéticos de este arte.

1.1 Propiedades del Sonido

El término sonido tiene un doble sentido: el primero puede ser empleado en sentido subjetivo para designar la sensación que experimenta un observador cuando las terminaciones de su nervio auditivo reciben un estímulo, pero también puede ser empleado en sentido objetivo para describir las ondas producidas por compresión del aire que puede estimular el nervio auditivo de un observador [1].

Acústica es la parte de la física que estudia el sonido en toda su amplitud, ocupándose de su producción, propagación, registro y reproducción, de la naturaleza del proceso de audición, de los instrumentos y aparatos para la medida, etc. Como rama de

la física, la acústica culminó su desarrollo en el siglo XIX, gracias a los trabajos de Hermann Von Helmholtz y de Lord Rayleigh, y sus bases teóricas han permanecido prácticamente sin cambios desde finales de ese siglo. Sin embargo, desde el punto de vista técnico, a lo largo del siglo XX los progresos de la acústica han sido constantes, especialmente por lo que se refiere a sistemas para el registro y la reproducción del sonido.

El sonido se produce por la vibración de los cuerpos, el cual se transmite al aire que lo rodea y, a través de éste, llega hasta nuestros oídos.

Por medio de dos experimentos se podrá confirmar estas afirmaciones.

1. Se disponen dos panderetas, próximas una frente a otra. De una de ellas se suspende un pequeño péndulo. Al golpear la otra, el péndulo comienza a vibrar.
2. Dentro de una campana en la que se ha hecho el vacío, se coloca un despertador: cuando se dispare la alarma no se oirá ningún sonido.

El primer experimento demuestra que el sonido se produce por la vibración de un cuerpo, mientras que el segundo demuestra que para que el sonido se transmita debe existir un medio elástico a través del cual se puedan propagar las vibraciones que lo originaron. Ese medio elástico es normalmente el aire, pero puede ser cualquier otro gas, un líquido o un sólido.

Cuando una onda sonora llega al tímpano del oído, éste entra en vibración y su vibración se transmite a los huesecillos que se apoyan suavemente sobre él. Es una situación del todo similar a la del experimento con dos panderetas dispuestas una frente a otra que se mencionó.

Los instrumentos musicales ilustran perfectamente la variedad de cuerpos cuya vibración puede dar origen a un sonido. Esencialmente, en los instrumentos de viento, lo que vibra es la columna de aire contenida en el instrumento; en los instrumentos de

cuerda, lo que vibra son las cuerdas del instrumento; y en los instrumentos de percusión lo que vibra es un diafragma o bien un objeto metálico (unos platillos, por ejemplo).

La música se hace con sonidos. En el sonido distinguimos diversos elementos, como la intensidad o fuerza con que se produce; la altura que nos hace considerarlo como agudo, medio o grave; el timbre, que es aquella cualidad del sonido gracias a la cual sabemos que lo que escuchamos está producido por uno u otro instrumento o voz; y la duración que nos permite apreciar el tiempo que el sonido está en nuestro oído.

Existe una distinción entre sonido y ruido. El sonido está producido por vibraciones regulares y periódicas, y el ruido por vibraciones irregulares que dan esta sensación confusa, sin entonación determinada. Tradicionalmente la música se hacía con sonidos y no con ruidos, pero hoy en día esto no se puede afirmar. La música utiliza cualquier sonido o ruido, ya sea natural o artificial.

1.1.1 La altura o tono.

La altura del sonido depende de su frecuencia, es decir, del número de vibraciones por segundo del cuerpo vibrante [2]. Cuanto más alta sea la frecuencia más agudo será el sonido y cuanto más baja sea menor será su altura. Esta propiedad recibe el nombre de tono. La duración de las vibraciones de instrumentos de un mismo tipo es proporcional a sus dimensiones lineales.

La medida de un sonido se encuentra fácilmente por comparación con otro parecido del cual se pueda determinar la frecuencia de una forma simple. Los sonidos de mayor o menor frecuencia se denominan respectivamente, agudos o graves, ya que entre los tonos diferentes uno de ellos será siempre más agudo que el otro y a la inversa.

1.1.2 La intensidad.

Es la cualidad que permite distinguir entre sonidos fuertes o débiles. Además de la amplitud en la percepción de la intensidad, influye la distancia a que se encuentra situado el oyente y la capacidad auditiva de este [1].

1.1.3 Amplitud.

La primera propiedad que una onda de sonido ha de tener es la amplitud. La intensidad de un sonido corresponde a nuestra percepción del mismo como más o menos fuerte [3]. Cuando elevamos el volumen de la cadena de música o del televisor, lo que hacemos es aumentar la intensidad del sonido. La amplitud es la distancia por encima y por debajo de la línea central de la onda de sonido. En la Figura: 1.1 se puede ver la onda sinusoidal.

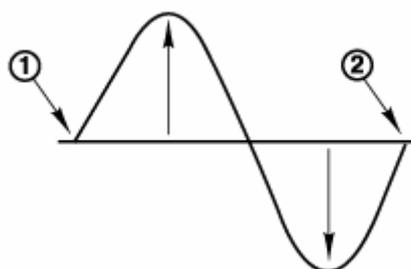


Figura: 1.1. Onda Sinusoidal

La línea central es la línea horizontal, llamada cero grados. La flecha vertical en la Figura: 1.1 denota la amplitud. La mayor distancia arriba y debajo de la línea central nos da el volumen del sonido (volumen es la palabra que se utiliza en los amplificadores de sonido) Si trabajáramos con estaciones o editores de audio digital, lo llamaríamos amplitud. Los displays de las estaciones de trabajo muestran el sonido grabado como una onda de sonido izquierda y derecha. Las ondas izquierda y derecha se presentan en dos cajas o huecos rectangulares uno al lado de otro. Cuando se reproduce el sonido, el display se moverá y veremos el promedio del volumen de una compleja forma de onda. Si los puntos de luz o la aguja llegan al final de la escala, entonces habrá distorsión. Así,

este display (en forma de aguja o de puntos luminosos) nos mostrará la amplitud de la onda y nos permitirá en todo momento saber cuando nos excedemos del volumen o cuando es inaudible.

1.1.4 Frecuencia.

Se mide en Hertz (el número de veces que vibra una onda sonora en un segundo) y nos permite saber a cuantos ciclos por segundo va una onda. Un ciclo es cuando la onda sube hasta un punto máximo de amplitud, baja hasta atravesar la línea central y llega hasta el punto de amplitud máximo negativo y vuelve a subir hasta alcanzar la línea central [3]. El *tono* o *altura* de un sonido depende de su frecuencia, es decir, del número de oscilaciones por segundo. El principio y el final de un ciclo se muestra por los números 1 y 2 de la Figura: 1.1

Esta medida, que puede tener cualquier longitud, se conoce como longitud de onda y el número de veces que pasa esto en un segundo, se conoce como frecuencia de la onda. Cuanto mayor sea la frecuencia, más agudo será el sonido. Cuantos más ciclos por segundo, más elevado será el tono. Así, la frecuencia hace el tono. La altura de un sonido corresponde a nuestra percepción del mismo como más grave o más agudo. Cada nota musical, tiene un valor en Hertz, nosotros vemos las frecuencias representadas en nuestras mesas de mezclas o grabación como un conjunto. Conociendo como ciertas frecuencias afectan el sonido de un instrumento, podremos fácilmente ecualizar ese instrumento y cambiar su “personalidad”. Esto nos ayudará a mejorar el sonido gracias a la ecualización. Por ejemplo, entre 20Hz y 100Hz nos proporciona un fondo o cuerpo, entre 100Hz y 200Hz, calor, entre 500Hz y 1500Hz definición, entre 1500Hz y 4KHz articulación, entre 4KHz y 10KHz brillo y entre 10KHz y 20KHz expansión.

1.1.5 El timbre o calidad.

Si el tono permite diferenciar unos sonidos de otros por su frecuencia, y la intensidad los sonidos fuertes de los débiles, el timbre completa las posibilidades de variedades del arte musical desde el punto de vista acústico, porque es la cualidad que

permite distinguir los sonidos producidos por los diferentes instrumentos [3]. Esta cualidad físicamente se llama forma de onda.

Podemos así distinguir si una nota ha sido tocada por una trompeta o por un violín. Esto se debe a que todo sonido musical es un sonido complejo que puede ser considerado como una superposición de sonidos simples. De esos sonidos simples, el **sonido fundamental** de frecuencia n es el de mayor intensidad y va acompañado de otros sonidos de intensidad menor y de frecuencia $2n$, $3n$, $4n$, etc. Los sonidos que acompañan al fundamental constituyen sus **armónicos** y de sus intensidades relativas depende el timbre. Sin embargo, muchos instrumentos, tales como el piano, el arpa, etc., no emiten un único sonido musical que se pueda considerar como una superposición de sonidos simples armónicos, sino que emiten un sonido constituido por superposición de sonidos parciales.

Cuando distinguimos entre un oboe y una guitarra es debido a que la cantidad, intensidad y calidad de los armónicos que vibran junto con la frecuencia principal son distintos en cada caso. Un aspecto importante dentro de la física sonora es la entonación. ¿Qué ocurre cuando durante una actuación observamos con inquietud que "algo va mal", que alguien está tocando más grave o más agudo que los demás? Está desafinando. Lo que ocurre realmente es que cuando dos notas tienen la misma frecuencia, por ejemplo 440 Hz sabemos que vibran a la misma altura, son *unísonos*. Pero si una de ellas vibra ligeramente por debajo, por ejemplo a 435 Hz., estará produciendo ondas más cortas que la otra, que chocarán unas contra otras produciendo una pulsación acústica cuyo ritmo será igual a la diferencia entre las dos frecuencias. En el caso que nos ocupa, esta pulsación sería de 5 Hz. Es interesante observar que más allá de unas 30 pulsaciones por segundo el efecto *disonante* disminuye.

1.1.6 Resonancia.

El principio de la resonancia es muy simple: cuando dos cuerpos tienen frecuencias iguales y uno de ellos está en vibración, el otro sin ser tocado vibra *por simpatía*[1]. Cuando cantamos no sólo vibran las cuerdas vocales si no que también se producen una vibración simpática en las cavidades de nuestra cabeza. De igual modo, los instrumentos

musicales experimentan el mismo fenómeno, el sonido del violín que oímos al frotar la cuerda con el arco está formado por las vibraciones que se producen en el cuerpo del instrumento.

1.2 Altura

Se la usa para determinar la percepción del tono (frecuencia) de un sonido. Se entiende por altura de un sonido su calidad de agudo ("alto") o grave ("bajo"). El que un sonido sea agudo o grave depende de su frecuencia medida en hertz. Cuanto más alta sea la frecuencia de la onda sonora, mayor será la altura del sonido (más agudo será). La relación entre la gravedad del sonido y la frecuencia de la onda sonora se puede experimentar de forma sencilla. Suponiendo que fijamos una pieza de metal alargada por uno de sus extremos, quedando el extremo libre en contacto con una rueda dentada de modo que al girar la rueda genere un rozamiento y por tanto un ruido de golpeteo cada vez que pasa un diente, entonces, si la rueda tiene 110 dientes y la hacemos girar una vez cada cuatro segundos oiremos un golpeteo muy rápido (22,5 golpes por segundo, o 22,5 hercios). Si aceleramos el giro de la rueda, el golpeteo gradualmente se convertirá en un sonido grave. Cuando estemos haciendo girar la rueda al doble de velocidad (una vuelta cada dos segundos, o 55 "golpeteos" por segundo), ya estaremos oyendo un LA_1 ¹ que por convención equivale a 55 Hz). Si hacemos girar la rueda al doble de velocidad (una vuelta por segundo) estaremos generando un LA_2 de 110 Hz, y si aumentamos la velocidad al doble, la frecuencia del sonido también crecerá al doble (un LA_3 de 220 Hz). Y así sucesivamente. Si hiciéramos girar la rueda a 10 vueltas por segundo escucharíamos un agudísimo y casi inaudible LA_9 de 14080 Hz. Más arriba de eso se oye un zumbido más o menos irritante, y encima de los 16.000 a 20.000 Hz (según el oído de cada persona) se deja de oír sonido, debido a que está generando un ultrasonido.

¹ LA . - Nota musical LA .

1.3 Armónicos

Armonía, en música es la combinación de notas que se emiten simultáneamente. El término armonía se emplea tanto en el sentido general de un conjunto de notas o sonidos que suenan al mismo tiempo, como en el de la sucesión de estos conjuntos de sonidos. La armonía sería el término contrapuesto al de melodía (en que los sonidos se emiten uno después de otro).

Cuando dos o más notas aparecen al mismo tiempo en cualquier composición musical se produce un tipo característico de armonía; en los acordes a la guitarra que acompañan una canción; en los bloques de sorprendentes acordes de la armónica que se tocan junto a la melodía en la música japonesa; y en los sonidos prolongados o insistentemente repetidos (llamados pedales) que sirven de base a géneros tan diversos como la música de gaitas de Escocia o la música clásica de la India.

1.3.1 Progresiones armónicas.

El enlace de un acorde con otro (llamado progresión armónica) va generando el movimiento en la música. Las progresiones armónicas incluyen la partida desde la tónica, los movimientos hacia la dominante, la resolución en la tónica o una resolución elusiva en otra armonía. Estas progresiones inciden en otros aspectos de la construcción musical como son los comienzos y finales de frases, y la estructuración de grandes secciones dentro de las composiciones. Los finales de frases y secciones se construyen mediante cadencias. Éstas pueden ser perfectas, en estado fundamental (que terminan en la tónica) o imperfectas, en cuyo caso van precedidas por la dominante. De hecho, en muchas épocas, estilos y géneros diferentes, la música tonal ha tendido a presentar frases o secciones emparejadas, la primera mitad con un final abierto sobre una cadencia imperfecta, y la segunda con un final cerrado sobre una cadencia perfecta.

Dentro de las frases, los puntos de cambio armónico suelen coincidir con los tiempos acentuados del compás. En otras palabras, la ubicación de los cambios armónicos es uno

de los elementos que hacen que el oyente escuche la alternancia regular de tiempos fuertes y débiles en el compás.

1.3.2 Armonía y Estructura.

Las armonías pueden aparecer con todas sus notas sonando juntas y sostenidas hasta la siguiente armonía. Ello también puede ocurrir con otras estructuras, en las cuales las notas se alternan o se repiten en diferentes patrones de acompañamiento que se escuchan como una unidad.

1.3.3 Ritmo musical.

Aspectos de la música que tratan sobre su movimiento en el tiempo y sobre la estructura de éste. Además de su significado general, el término ritmo suele hacer referencia a unos hechos temporales específicos como los patrones de longitudes de un cierto grupo de notas.[12]

1.3.4 Pulso y Compás.

Al igual que los ritmos en la naturaleza, como el movimiento de los planetas, la sucesión de las estaciones o el pulso del corazón, el ritmo musical suele organizarse en patrones de recurrencia regular. Dichos patrones controlan el movimiento de la música y ayudan al oído humano a comprender su estructura. La unidad rítmica básica por excelencia es el pulso, un patrón espaciado regularmente que se parece al ritmo de un reloj. En la mayor parte de la música de baile y en la popular, el pulso aparece de forma explícita, a menudo por medio del batir de los tambores o mediante un patrón de acompañamiento regular.[12] En músicas más complejas, el pulso sólo está implícito; es una especie de denominador común para las longitudes de las notas, que pueden ser más largas o cortas que el pulso mismo (sin embargo, cuando un oyente lleva el ritmo con el pie, el pulso vuelve a ser explícito). Para que el pulso pueda ser oído como denominador común, las longitudes de las notas individuales generalmente serán múltiplos o

subdivisiones exactas (como la mitad o el doble de la duración del pulso). El **tempo** de la música determina la velocidad del pulso.

Así como los pulsos regulan las duraciones de algunos tipos de realizaciones musicales cortas como una nota o un par de notas, éstas están a su vez reguladas por unidades recurrentes más largas llamadas compases. Los compases se forman acentuando el primer pulso o tiempo de una serie de dos o más, de modo que se agrupen en un patrón: por ejemplo; UNO² dos, UNO dos, o bien UNO dos tres, UNO dos tres. El término compás o metro puede referirse, en primer lugar, al proceso general de acentuación regular, y en segundo, al tipo de agrupación métrica particular usada en una obra determinada. [12]

En la notación musical, el compás se indica por medio de una armadura de compás. Al establecerse las armaduras de compás, se consideró la figura de la **redonda**³ como el valor de nota fundamental y por ello se expresaba la longitud del compás en relación con ella, y se le otorgó el valor 1. El número de abajo en una armadura de compás representa un valor de nota expresado como una fracción de una redonda. El número de arriba muestra cuántas unidades de dicho valor de nota hay en cada compás. Por ejemplo, uno de los compases o armaduras más comunes es el de $4/4$, que efectivamente significa 'cuatro cuartos': la unidad del compás es una **negra** (una cuarta parte de la redonda) y hay cuatro negras como éstas por compás. Los compases como los de $2/4$ y $3/4$ se llaman compases compuestos, dado que cada agrupación rítmica dentro del compás está compuesta por un subgrupo de valores rítmicos más pequeños. $2/4$ representa a dos grupos de tres notas cada uno, y $3/4$ a tres grupos de tres notas, y así con todos.

La música organizada métricamente está muy estructurada y tiende a ser regular. Sin embargo, una vez establecido el compás, no necesita que haya una adherencia rígida todo el tiempo; la mente del oyente retendrá el patrón incluso cuando la música lo contradiga temporalmente. Por ejemplo, puede acentuarse un tiempo generalmente débil para producir una **síncopa** (un acento que opera en contra del compás establecido). En

² UNO.- inicio del tiempo de una serie para dar un ejemplo.

³ Redonda.- representación de la nota fundamental.

el caso contrario, un tiempo fuerte puede ocasionalmente suprimirse completamente. De hecho, en las músicas de cierta complejidad rítmica siempre existe un grado de tensión entre el compás como sistema abstracto de regulación, por una parte, y, por otra, el flujo rítmico de las longitudes de las notas -un flujo que a veces sirve de apoyo al compás y que no sucede en otros tiempos. Más aún, el pulso no necesita mantenerse con absoluta rigidez, puede tocarse rubato⁴, es decir, con variaciones tan delicadas que no destruyan el valor básico.

1.3.5 Unidades de tiempo más largas.

Así como los pulsos se agrupan en compases, los compases se agrupan a su vez en unidades mayores. Dichos agrupamientos producen los segmentos de tiempo más extensos que determinan la forma de la música. Un **motivo** (la idea melódica más corta que conforma una unidad musical completa relativa) puede tener más de un compás. Pueden repetirse y variarse uno o dos motivos para formar una **frase** (una unidad aún más prolongada con una sensación de final más definitiva, que corresponde a grandes trazos a una oración del lenguaje hablado). Las frases se combinan en **secciones**, y las secciones en **composiciones** completas. La forma musical queda determinada por las relaciones entre estas distintas unidades de tiempo y también por las relaciones de dichas unidades respecto al todo, así como respecto a los esquemas armónicos a gran escala.

1.3.6 La Serie Armónica.

La serie armónica es la sucesión de armónicos que se producen al vibrar una cuerda o una columna de aire [4]. Cuando reproducimos el sonido **DO**⁵, en realidad, están sonando todos los armónicos que se puede observar en la Figura: 1.2

⁴**Rubato** - Modos de ejecutar un pasaje musical con cierta libertad en el tiempo de los compases.

⁵**DO**.- Nota musical

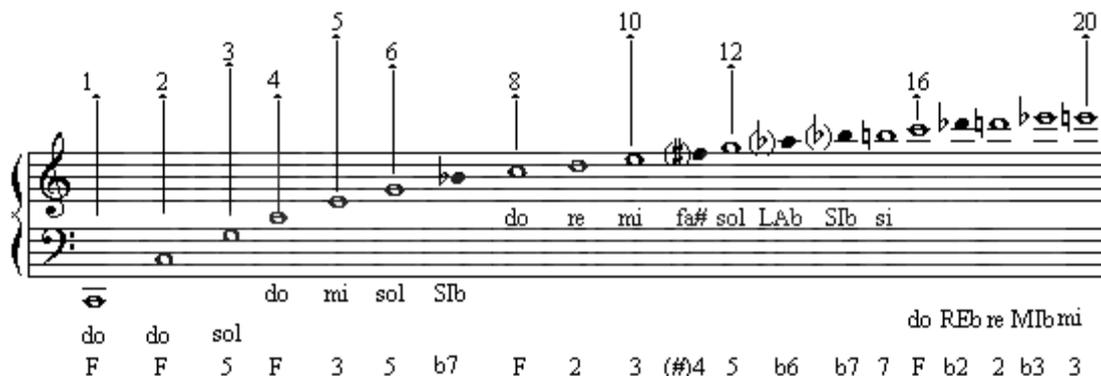


Figura: 1.2. La Serie Armónica

1.4 Escalas

La escala actual (escala occidental) es el resultado de un largo proceso de aprendizaje de las notas. Los pitagóricos construyeron un aparato llamado monocordio que se componía de una tabla, una cuerda tensa y una tabla más pequeña que se iba moviendo por la grande[4].

Los pitagóricos observaron que haciendo más o menos larga la cuerda (moviendo la tabla móvil) se producían sonidos diferentes. Entre estos sonidos escogieron algunos que eran armoniosos con el sonido original (cuerda entera).

Los más importantes, por su simplicidad y su importancia a la hora de construir la escala musical, son:

- * *La octava*, cuando la cuerda medía un medio del total, el sonido se repetía, pero más agudo. La octava es lo que correspondería a un salto de ocho teclas blancas del piano; o mejor dicho, una octava es la repetición de un sonido con una cuerda con la mitad de largura, por tanto, otra nota armoniosa. Su frecuencia es doble.

- * *La quinta* es otro intervalo entre notas que se obtiene con una cuerda de largura dos tercios de la inicial. Su frecuencia es de tres medios del sonido inicial. Corresponde a un salto de cinco teclas blancas en un piano.
- * *La cuarta* es, como las anteriores, otro intervalo entre notas que se obtiene con una cuerda de largura tres cuartos de la inicial. Su frecuencia es cuatro tercios de la nota inicial.

Así, a partir de un sonido original obtenemos diferentes notas armoniosas. Así tenemos la Tabla 1.1

Tabla 1.1. Notas Armoniosas

Nota	Frecuencia	Long. cuerda
Original	F	L
Octava	2f	1/2·L
Quinta	3/2·f	2/3·L
Cuarta	4/3·f	3/4·L

En donde F es frecuencia y L longitud.

Si suponemos que la nota inicial es el **DO**, entonces la octava, quinta y cuarta son las notas de la Tabla 1.2

Tabla 1.2. Ejemplo de las notas siguientes a Do

Nota base	Cuarta	Quinta	Octava
Do	Fa	Sol	Do (1 octava más alta)

Que corresponden a la cuarta, quinta y octava notas respectivamente de la escala diatónica. Todas estas relaciones entre las notas se denominan intervalos. A partir de esto, el nombre de las notas musicales tiene su origen en un himno a San Juan, en concreto, en la primera sílaba de cada verso. El himno es el que se muestra en la Tabla 1.3

Tabla 1.3. Himno a San Juan

Ut queant laxis
Resonare libris
Mira gestorum
Famuli tuorum
Solve polluti
Labii reatum
Sancte Joannes

Posteriormente la nota Ut, se transformó en nuestro DO. Actualmente también se utilizan las letras **A, B, C, D, E, F, G** para designar las notas musicales. La letra **A** corresponde al **LA**, y la letra **G** a la nota **SOL**. A partir de aquí, los símbolos se repiten y corresponden a las mismas notas una octava más alta. Estas son las 7 notas de la escala diatónica. De cualquier forma, en una octava se utilizan 12 notas (las de la escala cromática). Las 5 notas restantes se simbolizan añadiendo a la derecha el carácter # (sostenido) o **b** (bemol). En la Tabla 1.4 podemos ver la asignación de cada letra a la nota correspondiente.

Tabla 1.4. Notas Musicales

Letra	Nota Musical
A	LA
B	SI
C	DO
D	RE
E	MI
F	FA
G	SOL

1.4.1 Los intervalos musicales.

Un intervalo es la diferencia de tono entre dos notas. Mientras el nombre de la nota expresa su tono o frecuencia de vibración, el intervalo indica la relación entre una nota y otra.

El intervalo más pequeño que se utiliza en la música occidental es el *semitono*, y es el intervalo entre dos notas consecutivas de la escala cromática. La siguiente medida de un intervalo es un **tono**, que equivale a dos semitonos en la escala cromática.

Los intervalos más largos: **segunda, tercera, cuarta, quinta, sexta, séptima y octava** reciben sus nombres de la escala diatónica. Este nombre indica la relación existente entre una nota y la primera de la escala diatónica. Así, el intervalo do-sol se llama quinta ya que comprende cinco notas de la escala diatónica.

Los intervalos llamados perfectos son la quinta, la cuarta y la octava. A partir de ellos se pueden obtener el resto de las notas de la escala musical.

1.4.2 Las escalas musicales.

El sonido, para convertirse en materia artística, debe partir de un orden. El sonido, entonces, debe ser ordenado, y en Occidente, se la llama escala musical. Esta secuencia ha sido fruto de un largo proceso. Desde la elección de un sonido base, a partir del cual construir el resto, a la determinación del intervalo que hay entre una nota y la siguiente. Así, una escala es una serie de notas ordenadas de forma ascendente o descendente, a la primera de las notas se la llama **tónica**. A continuación se presenta un resumen de las escalas más conocidas.

1.4.2.1 La escala diatónica. Desde la Edad Media las escalas que se han utilizado son las escalas diatónicas, que se pueden simbolizar con las teclas blancas del piano. Estas escalas tienen dos intervalos diferentes: el semitono (en las teclas blancas, MI-FA y SI-DO) y tonos completos (entre las otras parejas de notas adyacentes). Tienen

siete notas por octava (la octava nota de esta serie es simplemente la repetición de la primera, pero situada una octava más arriba).

1.4.2.2 La escala cromática. A finales del siglo XIX, y dado el hecho del uso cada vez más frecuente de los sostenidos y los bemoles, la música occidental comenzó a basarse no en la escala diatónica, sino en la cromática: 12 notas en una octava, separadas por un semitono: DO, DO#, RE, RE #, MI, FA, FA #, SOL, SOL #, LA, LA #, SI (y DO).

1.4.2.3 La escala temperada. Los problemas de afinación en instrumentos con intervalos fijos (piano, guitarra), hizo construir una escala en la que el intervalo entre dos notas consecutivas fuese siempre el mismo. Esta es la escala temperada, que consta también de doce notas, como la cromática, pero la relación de la frecuencia de una nota y la anterior es siempre igual a $\sqrt[12]{2}$.

En 1939 se fijó la frecuencia de una nota de referencia, a partir de la cual se pueda deducir todas las otras. La nota y frecuencia escogidas fueron el LA₄⁶ a 440 Hz. A esta nota se le llama tono de referencia o tono de cámara.

A partir de esta se pueden deducir todas las demás. Las frecuencias de las notas que van del DO₄ al DO₅ serán las de la Tabla 1.5

Tabla 1.5. Escala Musical

Nombre Nota	Do	Do#	Re	Re#	Mi	Fa	Fa#	Sol	Sol#	La	Si b	Si	Do
Hz	261	277	294	311	330	349	370	392	415	440	466	494	523

1.4.2.4 Otras escalas. Se ha hablado de la escala diatónica y la temperada, pero en el mundo no occidental hay otras. Como ejemplos se puede ver tres escalas diferentes. En el gráfico se representa el intervalo entre una nota y la siguiente, ver la Figura: 1.3

⁶ LA₄- Nota musical LA con frecuencia de 440Hz.

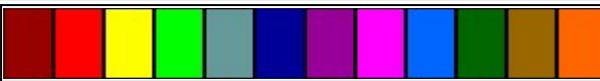
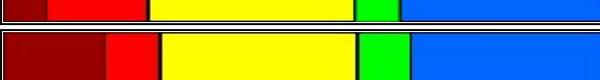
Nombre	Mapa de los intervalos
Escala temperada	
Escala Diatónica	
Shree – India	
Sorog – Bali	
Hirajoshi - Japón	

Figura: 1.3. Ejemplo de diferentes escalas

Podemos ver que los únicos intervalos comunes entre todas las escalas son la octava y la quinta.

1.4.3 Sonidos armónicos.

La primera nota diferente de la fundamental que encontramos en orden de aparición (tercer armónico) forma un intervalo de quinta justa con aquélla. Por ejemplo, siendo la fundamental el LA, su quinta es el MI, pero en realidad se trata del MI de la octava siguiente de la fundamental ($440 \times 3 = 1320$ Hz). La octava siguiente (2640 Hz) también será un MI, que será el sexto armónico del LA de 440 Hz. El quinto armónico está separado de la fundamental por dos octavas y un intervalo más llamado de tercera mayor. En el caso de nuestro ejemplo, sería la nota DO#, cuya frecuencia sería de $440 \times 5 = 2200$ Hz (el décimo armónico será el DO# de 4400 Hz) [5]. Ver Figura: 1.4

El séptimo armónico no coincide con ninguna nota de la afinación occidental, siendo de frecuencia algo inferior, en este caso, al SOL. Tampoco afina el armónico undécimo RE# bajo), aunque sí el noveno; en este caso, SI ($440 \times 9 = 3960$ Hz). Ver Figura: 1.5

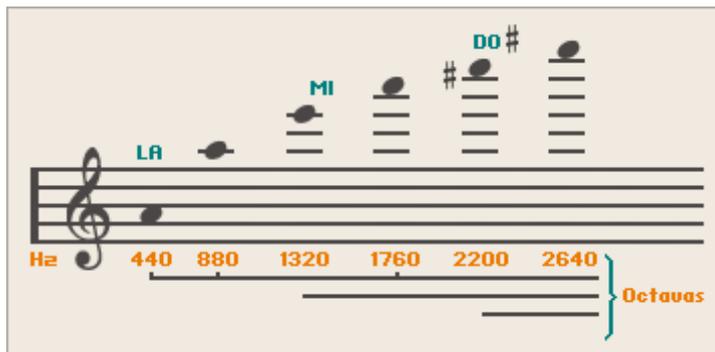


Figura: 1.4



Figura: 1.5

CAPITULO 2

AFINADORES MUSICALES

La afinación es la acción de poner en tono justo los instrumentos musicales de acuerdo a un **diapasón**⁷ o ajustarlos bien unos con otros. También se le llama afinación al canto o ejecución de un instrumento entonando con perfección los sonidos. Es por ello que se ha utilizado varios sistemas de afinación musical para determinar la frecuencia relativa de las notas de una escala musical, a lo largo de la historia se han usado varios estándares de afinación en un intento de fijar la altura absoluta de la escala. En 1955, la Organización Internacional de Estandarización fijó la frecuencia del **LA**⁸ en 440 Hz. Pero en el pasado se utilizaron distintos sistemas de afinación.

2.1 Características y funciones de los afinadores

Durante el siglo XV Michael Praetorius rechazó varios estándares de afinación altos (de más de 480 Hz debido a que provocaba que se rompieran las cuerdas más agudas (más delgadas) del violín, que en esa época se hacían con intestinos de animales.

La necesidad de estandarizar los niveles de afinación (al menos dentro de una misma ciudad o un mismo país) surgió cuando se empezó a popularizar la ejecución de música combinada de órgano con ensambles instrumentales. Una manera en que se empezó a controlar la afinación era con el uso de diapasones de horquilla, aunque tampoco aquí se ponían de acuerdo. A lo largo de la primera mitad del siglo XIX, hubo una tendencia a subir la afinación de la orquesta. Esto probablemente se haya debido a que las orquestas

⁷ **Diapasón**.- es una pieza en forma de U de metal elástico. Cuando se le golpea haciéndolo vibrar, genera una onda sinusoidal casi inaudible El diapasón mas utilizado es el llamado *la* 440.

⁸ **LA**.- Nota musical

competían unas con otras, tratando de llenar las salas de concierto cada vez más grandes con un sonido más brillante que el de sus competidoras. Fueron ayudadas en sus esfuerzos por la durabilidad mejorada de la cuerda MI de los violines (la más aguda de las cuatro cuerdas). En el pasado las cuerdas de tripa de animales no aguantaban tanta tensión. Pero las nuevas cuerdas de acero podían aguantar más tensión sin romperse.

En 1939 el LA encima del DO central se afino a 440 Hz. El estándar fue aceptado por la Organización Internacional de Estandarización en 1955 (y fue reafirmado por ellos en 1975) como **ISO 16**⁹ [6]. La diferencia entre esta afinación y el "diapasón normal" se debió a la confusión acerca de cuál era la temperatura a la que se debía medir el estándar francés. El estándar inicial era la 439 Hz, pero fue reemplazado por el LA 440 Hz después de registrarse quejas acerca de la dificultad de reproducir los 439 Hz en laboratorio debido a que 439 es un número primo. A pesar de esta confusión, el LA 440 Hz ahora se utiliza prácticamente en todo el mundo, por lo menos en teoría. En la práctica, las orquestas afinan con el LA que genera el oboísta principal, en vez de hacerlo con algún dispositivo electrónico (lo cual sería más confiable), y el oboísta mismo no utiliza tal dispositivo para afinar su instrumento en primer lugar, así que todavía puede haber una ligera diferencia en la afinación exacta utilizada. Los instrumentos solistas como el piano (con quien afina la orquesta cuando tienen que tocar juntos) a veces tampoco están afinados con el LA 440 Hz. De todos modos se cree que desde mediados del siglo veinte ha existido una ligerísima tendencia a subir la afinación estándar, aunque ha sido casi imperceptible.

Al menos en las orquestas de cámara y sinfónicas formadas por alumnos de los conservatorios de música europeos actuales se toma como referencia un LA₄ de 442 Hz producido por un dispositivo electrónico. En los estudios de teoría se sigue hablando de 440 Hz para el LA₄ pero la práctica instrumental se considera ajena a esta sujeción. Aunque sigue siendo el oboe el instrumento encargado de dar la referencia al resto de la orquesta, el oboísta afina su instrumento *in situ*¹⁰ por un afinador digital. Esto es así incluso entre los grupos de música antigua, que suelen afinar a 415 Hz (un semitono temperado por debajo de 440 Hz) la música renacentista y a 435 Hz la música barroca.

⁹ **ISO 16**.- según la norma ISO 16 el "la" 440 Hz es la afinación estándar desde (1975) hasta la actualidad.

¹⁰ **IN SITU**.- en su lugar, en su sitio

La guitarra y otros instrumentos de cuerda con trastes en el mástil, afinan sus cuerdas por cuartas o quintas perfectas en el caso de las cuerdas al aire. En música Pop, si se utiliza un afinador electrónico, éste seguirá el sistema temperado para la afinación de las cuerdas. Los instrumentos de cuerda frotada sin trastes en el mástil, como el violín, gozan de libre albedrío en cuanto a la altura de los sonidos de las cuerdas pisadas¹¹, aunque en la práctica los ejecutantes aplican una técnica fija para la producción de las notas. La consigna en este caso es hacer los tonos grandes y los semitonos diatónicos pequeños, como en el sistema de Pitágoras, y en cambio hacer las terceras armónicas pequeñas como en el sistema justo.

Los instrumentos de viento-metal con llaves, válvulas o pistones se afinan por el sistema temperado en lo que respecta a estos mecanismos, y según la serie armónica en lo que respecta a los sonidos conseguidos como armónicos de una nota base correspondiente a una posición dada de los mismos. También siguen estrictamente la serie armónica los instrumentos sin ningún mecanismo que altere la longitud real del tubo, como las trompas o trompetas naturales. La serie armónica presenta una gran variedad de intervalos entre sus notas: quintas y cuartas justas, tonos grandes y pequeños, terceras justas, varios tipos de semitonos e incluso notas "prohibidas" como los múltiplos de 7, 11 ó 13.

Los instrumentos de viento-madera tienen sus llaves o agujeros dispuestos según el sistema temperado, pero algunas notas pueden hacerse como armónicos de otras, lo que da lugar a intervalos justos que son algo diferentes. La voz humana goza de libertad total en todos los sentidos, pero los coros entonan las escalas mayores según un sistema semejante al de Pitágoras y los acordes mayores con terceras pequeñas como en el sistema justo, cuando el coro está adiestrado para la correcta entonación de la música antigua. En la música moderna (sobre todo cuando está acompañada de piano) se tiende más hacia el sistema temperado.

¹¹ **Pisada**.- que esta sujeta o apretada contra el mástil del instrumento de cuerda.

2.2 Afinadores por Software

Existen hoy en día en el mercado una gran cantidad de afinadores por software, es decir que se los puede instalar en una computadora personal o en computadores portátiles, existen de varios fabricantes y la mayoría cuentan con características parecidas como:

- Rápida y aproximada afinación de instrumentos musicales de cuerda entre ellos guitarra, bajo, violín, banjo, mandolín, cello y guitarra de 12 cuerdas.
- Uso del micrófono como entrada, o la conexión directa de la guitarra eléctrica
- Indicador gráfico para afinación, por medio de barras.

Los requerimientos mínimos de este tipo de afinadores son:

- Un computador con entrada de micrófono, y aproximadamente 0,63Mbytes de espacio libre.
- Sistema operativo Win95/Win98/WinMe/Windows2000/WindowsXP.

A continuación se presenta unos ejemplos de los afinadores por software existentes en la Figura: 2.1 y en la Figura: 2.2 se puede apreciar la presentación de dichos afinadores:

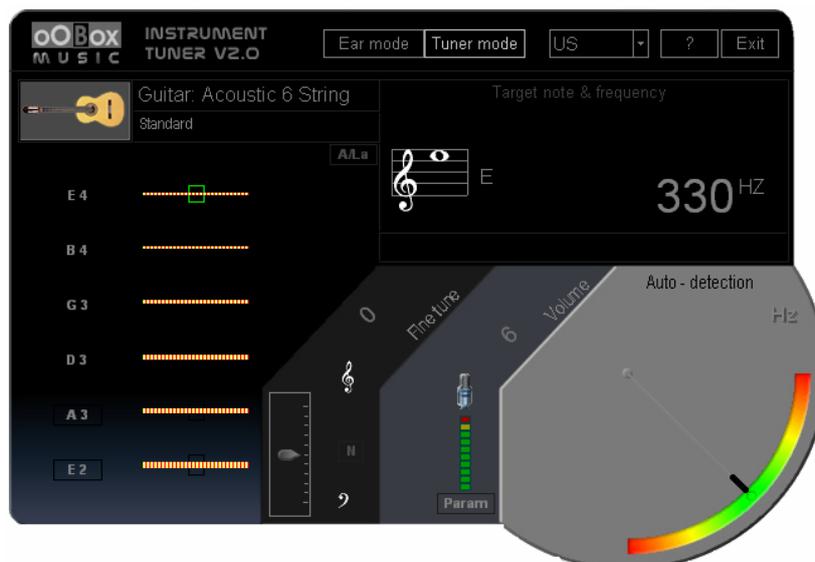


Figura: 2.1. OOBX de PIXSYS

Características:

- Afinación de guitarra acústica, banjo y bajo.
- Entrada por medio del micrófono.
- Visualización grafica.
- Emisión de sonido de cada nota del instrumento de cuerda.

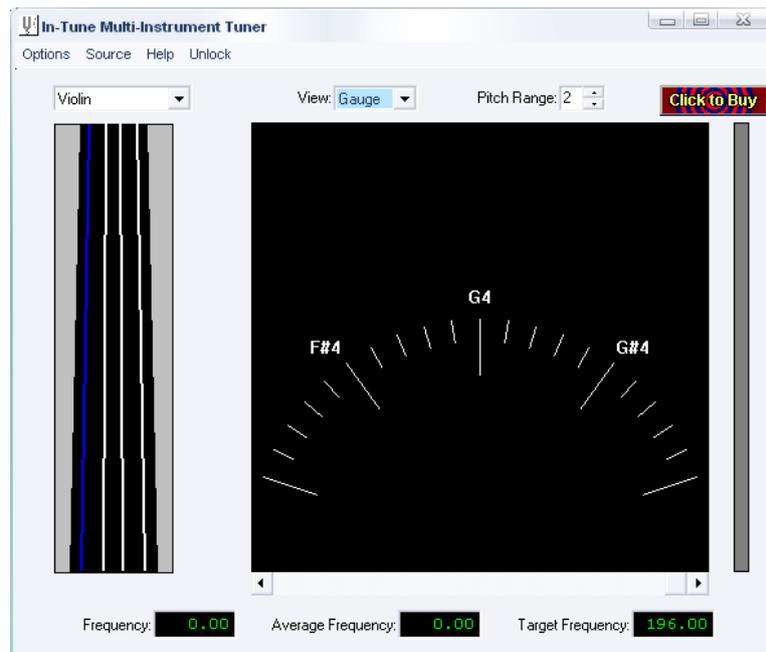


Figura: 2.2. In - Tune de Joseph Heimiller

Características:

- Afinación para guitarra, violín, banjo, bajo, mandolin, viola, cello y doble bajo.
- Visualización por medio de pluma o grafico de barras.
- Entrada por medio del micrófono.

2.3 Afinadores Portátiles

Los afinadores portátiles son mucho más comunes, por la facilidad de transportarlos a cualquier lugar. Se los encuentra en diversas marcas, pero con las mismas características en la mayoría de los casos, aunque existen pequeñas variaciones dependiendo de la marca y la necesidad del consumidor, así como en la variedad de precios. A continuación se presenta las funciones básicas de la mayoría de estos

afinadores y la Figura: 2.3, la Figura: 2.4 y la Figura: 2.5 muestran algunos de los mismos:

- Pantalla LCD
- Bocina interna que emite sonido o led's de referencia para afinar
- Auto power off
- Jack de entrada y salida para ingreso de la señal de audio
- Botones controladores

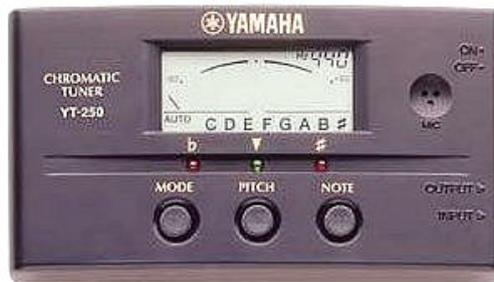


Figura: 2.3. Afinador Portátil Yamaha



Figura: 2.4. Afinador Portátil Seiko



Figura: 2.5. Afinador Portátil Korg

CAPITULO 3

CARACTERISTICAS DEL CONTROLADOR DIGITAL DE SEÑALES dsPIC30F3014

El dsPIC30F3014 no es solo un controlador digital de señales (DSC), sino que también es un dispositivo que tiene la funcionalidad de un procesador digital de señales (DSP), junto con el alto rendimiento de la arquitectura de un microcontrolador de 16-bits. En la Figura: 3.1 se muestra el diagrama general de pines y el diagrama de bloques correspondiente en la Figura: 3.2 [7].

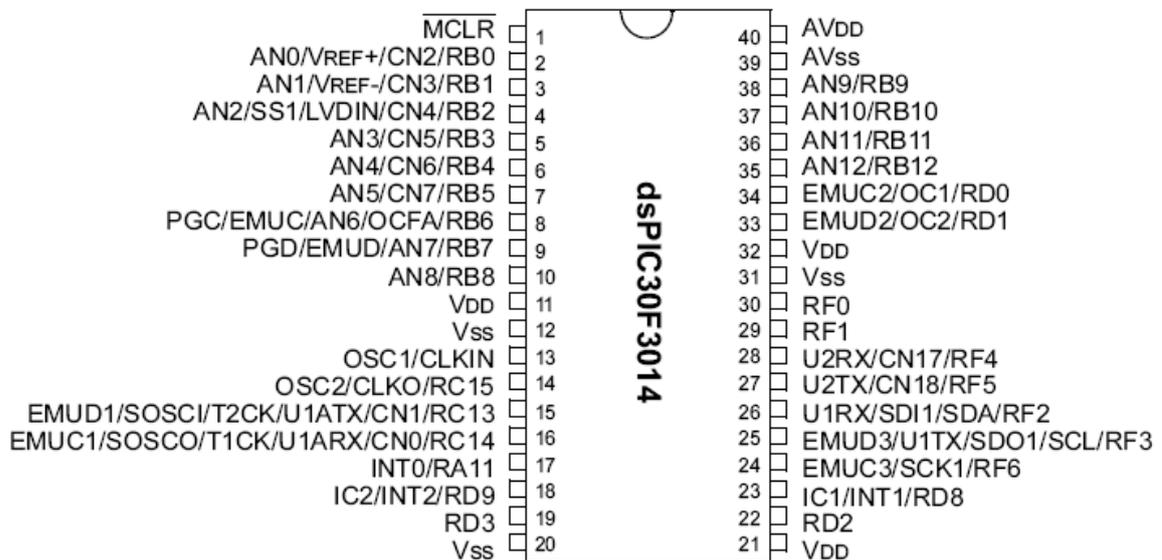


Figura: 3.1. Diagrama General de pines del microcontrolador dsPIC30F3014

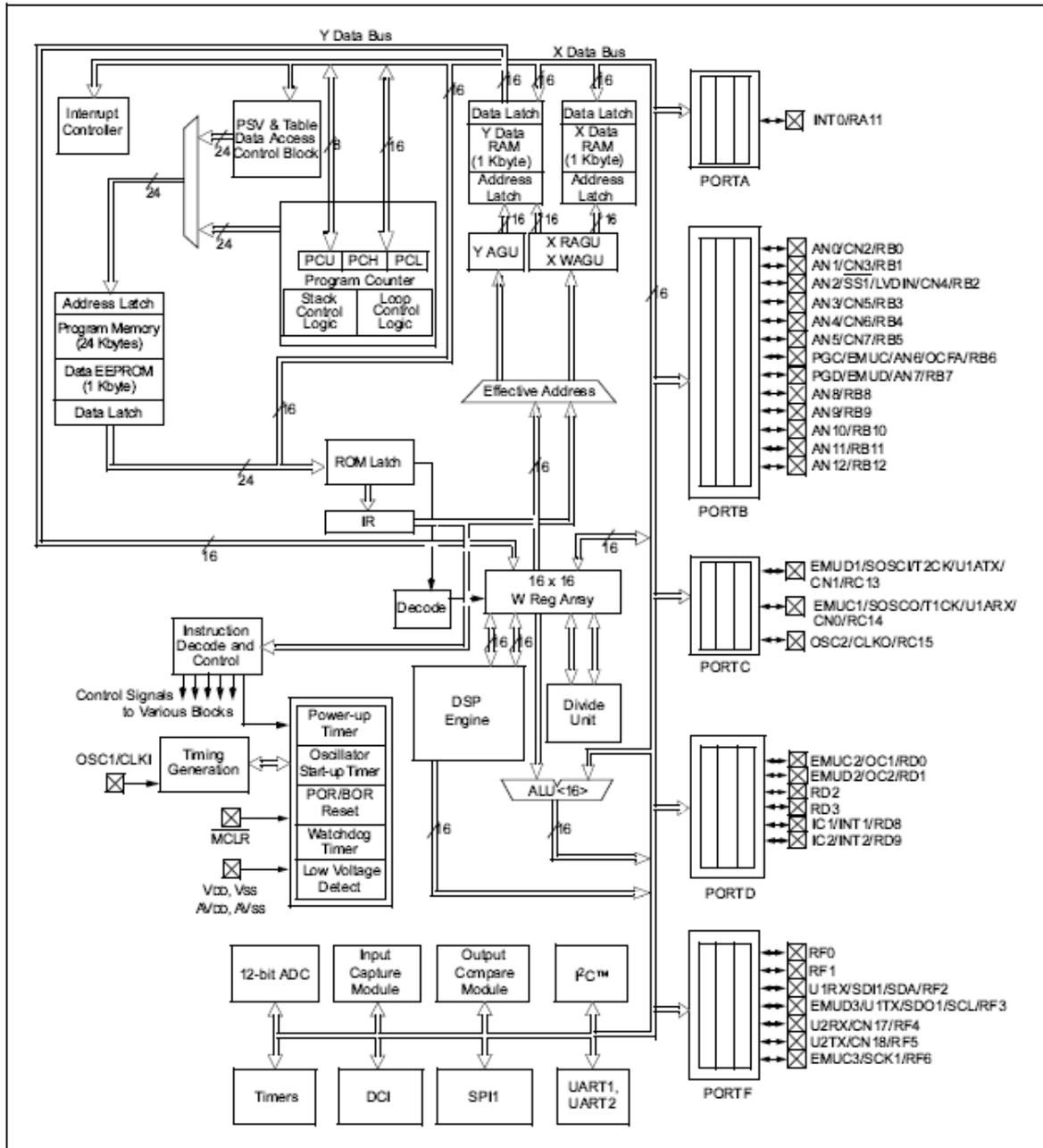


Figura: 3.2. Diagrama de Bloques del dsPIC30F3014

3.1 Características principales

Cuenta con un *CPU* (Central Processor Unit) de alto desempeño basado en una arquitectura Harvard¹², el dsPIC30F3014 esta provisto con modos de direccionamiento flexibles y una base de 84 instrucciones, con un espacio de programación de 48Kbytes y memoria RAM de datos de 2Kbytes, junto con 1Kbyte de datos *EEPROM* (Electrically Erasable Programmable Read-Only Memory) [7].

3.1.1 Características del DSP.

PROCESAMIENTO DIGITAL DE SEÑAL (*DSP*) es una operación o transformación de una señal en un hardware digital a través de un software específico. Una de las más importantes características de un DSP es su capacidad de realizar operaciones de multiplicación y acumulación (**MACs**) en sólo un ciclo de reloj. Las características y cualidades en la arquitectura de los DSP's más comunes son: una unidad funcional rápida que puede multiplicar y acumular en un ciclo de instrucción, varias unidades funcionales que realizan operaciones en paralelo, incluyendo accesos a memoria y cálculo de direcciones. Las unidades poseen típicamente una unidad principal (*ALU*) junto con dos o más unidades de generación de direcciones. Soporte para tipos especiales de direccionamiento, especialmente módulo y *bit-reverse* (bits invertidos), requerido en el cálculo de la FFT, el direccionamiento módulo es muy eficiente para la implementación de buffers circulares y por último, soporte para manejo de *loop* (lazo de programación) con bajo costo en tiempo y manejo rápido de interrupciones, especialmente aquellas que se deben a los puertos seriales [7].

En dsPIC30F3014 cuenta con las siguientes características para el módulo *DSP*, un multiplicador de 17 bit x 17 bit, un *barrel shifter*¹³ y un sumador / restador de 40 bits. El *DSP* tiene la capacidad de desarrollar operaciones acumulador – a – acumulador que no requieren datos adicionales, el DSP tiene una arquitectura de flujo de un solo ciclo de instrucción, por lo que durante el flujo de operación del DSP no es posible la operación de las instrucciones del microcontrolador, sin embargo algunos recursos del

¹² **Arquitectura Harvard.**- La arquitectura Harvard tiene la unidad central de proceso (CPU) conectada a dos memorias (una con las instrucciones y otra con los datos) por medio de dos buses diferentes.

¹³ **Barrel Shifter.**- es un circuito digital que puede cambiar de puesto una palabra de datos por un número especificado de bits.

3.1.2 Características de los periféricos.

Los pines de entrada/salida tiene una capacidad de corriente de 25mA, cuenta con cinco contadores o timers de 16 bits, con la opción de unirlos en módulos de timers de 32 bits, 4 funciones de captura de entrada de 16 bits y 4 funciones de salida de comparación o PWM (Pulse Width Modulation) de 16 bits, una interfaz convertora de datos que soporta protocolos de audio comunes incluyendo I2S y AC'97. Además esta equipado con un módulo SPI (Serial Peripheral Interface) de 3 cables y otro para I2C (Inter Integrated Circuit) que soporta el modo Multi-Master/Slave y direccionamiento de 7 bits/10 bits y por último dos módulos UART (Universal Asynchronous Receiver Transmitter) con buffers FIFO (First In First Out) y un módulo CAN (Controller Area Network) bus compatible con el estándar CAN 2.0B. 14 [7].

3.1.3 Características Analógicas.

Provisto de un convertor Análogo/Digital de 12 bits con un rango de conversión de 100 Ksps (Kilo Samples Per Second), 13 canales de entrada y conversión durante el modo Sleep¹⁵ e Idle¹⁶. Detección programable de bajo voltaje y por último programación de Brown-out Detection¹⁷ y generación de Reset.

3.1.4 Características Especiales del Microcontrolador.

Programación de memoria Flash mejorada con 100.000 ciclos de borrado/escritura para un rango de temperatura industrial 100°K (típicamente). Auto-programación bajo software de control. Adicionalmente tiene operaciones a prueba de fallas de reloj las cuales detectan fallas del reloj y cambios en el oscilador RC del integrado cuando esta con baja alimentación. La protección de código de programación junto con la programación serial del circuito (ICSP) y los modos de manejo de alimentación como Sleep, Idle y modos de alternar el reloj, hacen de este integrado uno de los más completos en su clase.

¹⁴ Para mayor información favor referirse al Datasheet del dsPIC30F3014 que esta disponible en línea en www.microchip.com

¹⁵ **Sleep**.- modo dormido del dsPIC30F3014

¹⁶ **Idle**.- En informática, un proceso, un procesador o un circuito electrónico es descrito como "idle" cuando está siendo usado por ningún programa, aplicación o mensaje.

¹⁷ **Brown-out Detection**.- Detención de baja tensión

3.1.5 Tecnología CMOS.

A todas las características anteriormente citadas se suma la tecnología CMOS¹⁸ de bajo consumo, un amplio rango de voltaje de operación que va de 2.5V a 5.5V, rangos de temperatura de funcionamiento extendidos.

3.2 Descripción de Módulos

El dsPIC30F3014 esta compuesto de los siguientes módulos:

3.2.1 Puertos de entrada/salida.

Todos los pines del microcontrolador (excepto V_{DD} , V_{SS} , MCLR, y OSC1/CLK) son compartidos entre los periféricos y el puerto paralelo de entrada/salida, cada uno de los puertos de entrada tiene la característica *Schmitt Trigger*¹⁹ para mejorarla inmunidad al ruido. Cuando un periférico se habilita y si esta manejando activamente un pin, el uso del pin como salida de propósito general es deshabilitado, pero si el periférico se habilita pero no esta manejando un pin activamente, el pin puede ser usado como puerto. Todos los pines de tiene tres registros directamente asociados con la operación del puerto. El registro de dirección de datos (TRISx) determina cuando un pin es entrada o cuando es salida. Si la dirección de los datos es un “1”, el pin es de entrada. Todos los pines del puerto están definidos como entrada después de un RESET. Lo que hace que se siga la siguiente secuencia, leer del *latch*²⁰ (LATx), escribir en el latch (LATx), leer del puerto (PORTx), leer y escribir en los pines del puerto y escribir en el match (LATx) [7].

¹⁸ **CMOS** (del inglés *Complementary Metal Oxide Semiconductor*, "Metal Óxido Semiconductor Complementario") es una tecnología utilizada para crear circuitos integrados, los chips CMOS consumen menos potencia Funcionan con tensiones desde los 3 V hasta los 15 V

¹⁹ **Schmitt trigger** cambia su estado de salida cuando la tensión en su entrada sobrepasa un determinado nivel; la salida no vuelve a cambiar cuando la entrada baja de ese voltaje, sino que el nivel de tensión para el cambio es otro distinto, más bajo que el primero. El trigger Schmitt usa la histéresis para prevenir el ruido que podría solaparse a la señal original y que causaría falsos cambios de estado si los niveles de referencia y entrada son parecidos.

²⁰ **Latch** es un circuito electrónico usado para almacenar información en sistemas lógicos asíncronos. Un latch puede almacenar un bit de información.

Cualquier bit en los puertos de entrada/salida, sus datos asociados y los registros que no sean válidos (leídos como cero) para un elemento en particular serán deshabilitados, es decir que los correspondientes registros LATx y TRISx y sus pines serán leídos como cero. El registro (TRISA) controla la dirección de los pines RA<7:0>, así como también los pines de INTx y V_{REF}, el registro LATA provee de datos a las salida y a sus funciones de lectura/escritura.

En la Figura: 3.4 se muestra los puertos que estas compartidos con otros periféricos y asociados a otras celdas que estén conectadas, y en la **¡Error! No se encuentra el origen de la referencia.** muestra los formatos de los registros compartidos por los puertos, desde el PORTB hasta PORTG.

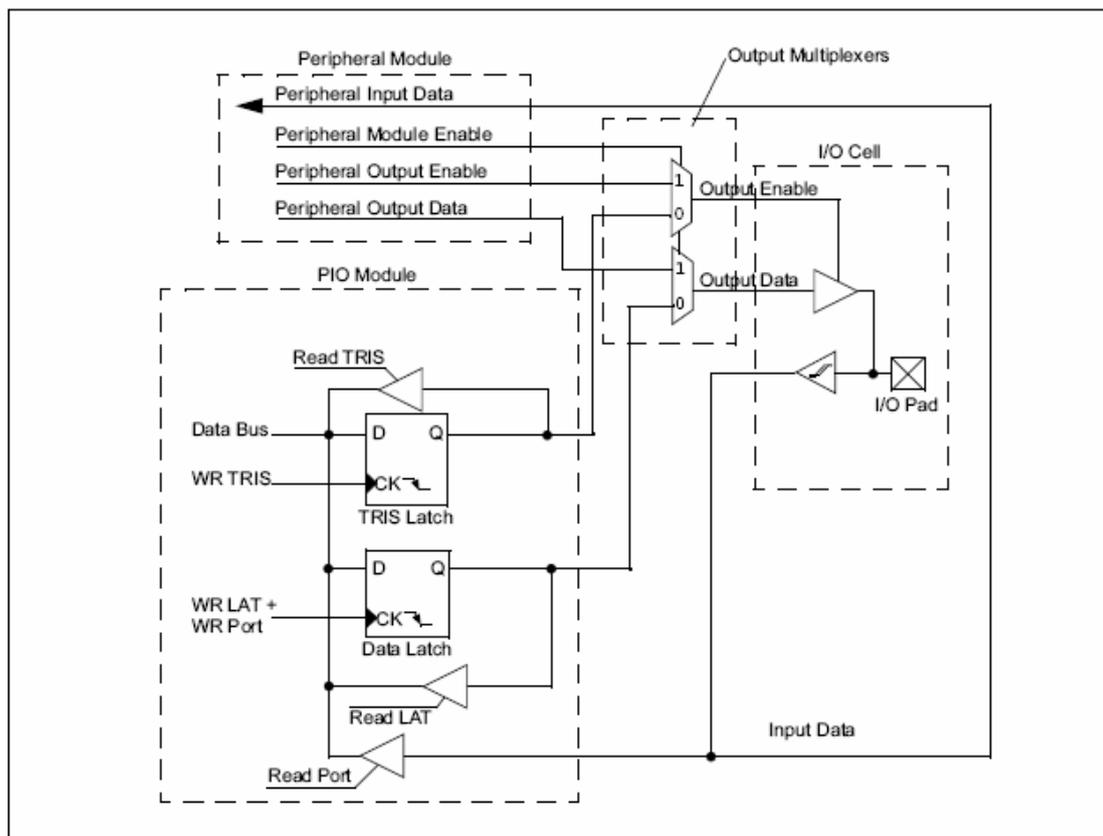


Figura: 3.4. Diagrama de bloques de la estructura de un puerto compartido

Tabla 3.1. Mapa de Registros de los Puertos
E/S

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
TRISA	02C0	—	—	—	—	TRISA11	—	—	—	—	—	—	—	—	—	—	—	0000 1000 0000 0000
PORTA	02C2	—	—	—	—	RA11	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
LATA	02C4	—	—	—	—	LATA11	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
TRISB	02C6	—	—	—	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	0001 1111 1111 1111
PORTB	02C8	—	—	—	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	0000 0000 0000 0000
LATB	02CB	—	—	—	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000 0000 0000 0000
TRISC	02CC	TRISC15	TRISC14	TRISC13	—	—	—	—	—	—	—	—	—	—	—	—	—	1110 0000 0000 0000
PORTC	02CE	RC15	RC14	RC13	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
LATC	02D0	LATC15	LATC14	LATC13	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
TRISD	02D2	—	—	—	—	—	—	TRISD9	TRISD8	—	—	—	—	TRISD3	TRISD2	TRISD1	TRISD0	0000 0011 0000 1111
PORTD	02D4	—	—	—	—	—	—	RD9	RD8	—	—	—	—	RD3	RD2	RD1	RD0	0000 0000 0000 0000
LATD	02D6	—	—	—	—	—	—	LATD9	LATD8	—	—	—	—	LATD3	LATD2	LATD1	LATD0	0000 0000 0000 0000
TRISF	02DE	—	—	—	—	—	—	—	—	—	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	0000 0000 0111 1111
PORTF	02E0	—	—	—	—	—	—	—	—	—	RF6	RF5	RF4	RF3	RF2	RF1	RF0	0000 0000 0000 0000
LATF	02E2	—	—	—	—	—	—	—	—	—	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	0000 0000 0000 0000

- **Configuración de los puertos analógicos.** El uso de los registros ADPCFG y TRIS controlan la operación de los pines del puerto análogo/digital, los pines del puerto que se desea que sean entradas analógicas deben tener su correspondiente bit TRIS seteando (1) (entrada). Si el bit TRIS esta *clear* (0) será una salida, el nivel de salida digital (V_{OH} o V_{OL}) deberá ser convertido. Cuando se lee el registro del puerto, todos los pines están configurados como canales de entrada analógica y se los leerá como *clear* (0 o nivel bajo). Los pines configurados como entrada digital no convertirán una entrada analógica,
- **Tiempo de lectura/escritura del puerto entrada/salida.** Se requiere un ciclo de instrucción entre el cambio de dirección del puerto o de la operación de escritura y lectura para el mismo puerto.

3.2.2 Módulo de notificación de cambios en una entrada.

Este módulo provee al DSPIC30F la habilidad de generar una petición de interrupción al procesador, en respuesta a un cambio de estado en los pines de entrada seleccionados, este módulo es capaz de detectar el cambio en una entrada incluso en el modo *Sleep*, cuando el reloj está deshabilitado. Existen 24 señales externas (CN0 hasta CN23) que se pueden seleccionar (habilitar) para generar una petición de interrupción o un cambio de estado. [7]

3.2.3 Conversor análogo – digital de 12-bits.

Este conversor análogo – digital permite realizar una conversión de una señal de entrada analógica a un número digital de 12 bits. Este modulo se basa en la arquitectura de un registro de aproximaciones sucesivas que provee un rango máximo de muestreo de 100 ksp/s. El modulo A/D tiene 16 entradas análogas que son multiplexadas en amplificador de muestra y retención. La salida del amplificador es la entrada al conversor que es el que genera el resultado. Cuenta con la opción de tener un voltaje de referencia que puede ser tanto vía software como una señal de voltaje de entrada en el pin (V_{REF+}/V_{REF-}).

El convertor A/D tiene la característica de ser capaz de operara mientras el dispositivo se encuentra en modo *Sleep* con la selección de un oscilador RC. El modulo A/D tiene seis registros de 16-bits distribuidos de la siguiente manera, los *ADCON1*, *ADCON2* y *ADCON3* controlan la operación del convertor, el registro *ADCHS* selecciona el canal de entrada que va a ser convertido, el *ADPCFG* configura los pines del puerto como entradas/salidas analógicas o digitales y por ultimo el *ADCSSL* selecciona las entradas para el escaneo [7]. En la Figura: 3.5 se muestra el diagrama de bloques del modulo

NOTA. Los bits *SSRC*<2:0>, *ASAM*, *SMPI*<3:0>, *BUFM* y *ALTS* del registro *ADCON1* así como los registros *ADCON3* y *ADCSSL* no se deben escribir mientras *ADON* = 1, ya que podría causar resultados indeterminados.

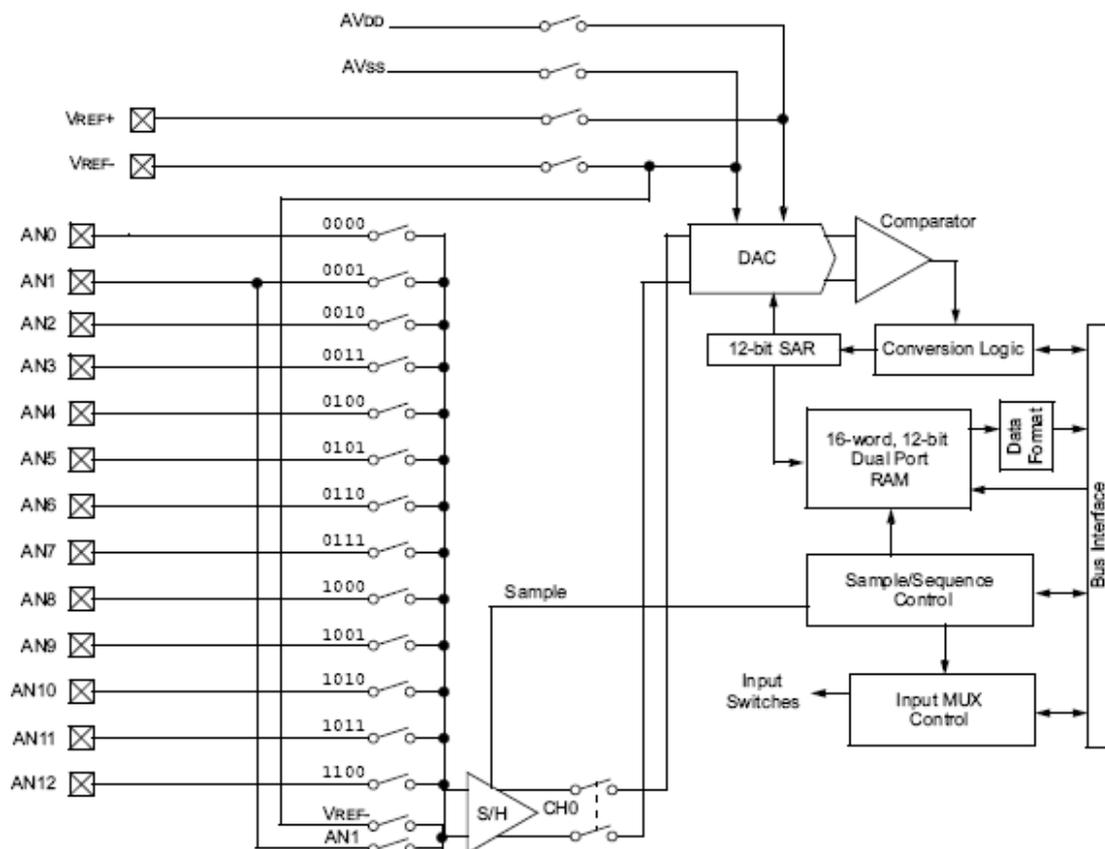


Figura: 3.5. Diagrama de bloques del modulo A/D

3.3 Instrucciones y librerías para el procesamiento digital de señales

El set de instrucciones del dsPIC30F reúne muchas mejoras en cuanto al set de instrucciones del PICmicro, pero manteniendo una migración fácil de las instrucciones del PICmicro hacia el dsPIC30F, la mayoría de las instrucciones son palabras de 24 bits por lo que requieren de un solo espacio de memoria, cada una de estas palabras esta dividida en un código de 8-bits que especifica el tipo de instrucción, y uno o mas operandos que luego indicaran la operación de la instrucción. El set de instrucciones esta agrupado en cinco categorías básicas: operaciones orientadas a palabras o bytes, operaciones orientadas a bits, operaciones literales, operaciones de DSP y operaciones de control. El software utilizado para la programación del dsPIC es Mikrobasic for dsPIC [8] que cuenta con una serie de librerías desarrolladas para el funcionamiento del DSC, basados en este software tenemos la librería del DSP que cuenta con varias rutinas las que serán utilizadas en el prototipo son:

- FFT

El subprocedimiento para la FFT que esta definido por MikroBasic recibe los siguientes parámetros de la siguiente forma:

```
sub procedure FFT(dim log2N as word, dim TwiddleFactorsAddress as LongInt, dim byref Samples as array [1024] of word). [8]
```

Esta función aplica la transformada de Fourier (FFT) de las muestras de entrada, estas muestras deben estar ubicadas en el espacio de memoria Y^{21} , donde:

- N – es el tamaño del buffer de entrada (debe ser potencia de 2).
- TwiddleFactorsAddress es la dirección de un arreglo constante que contiene los *complex twiddle factors*²². Este arreglo debe estar en la memoria del programa.
- Samples – es el arreglo de las muestras de entrada.

Una vez realizada la conversión del las muestras el resultado es colocado en la variable Samples.

²¹ **Espacio de memoria Y.** - empieza en la dirección 000C del mapa de memoria del dsPIC30F3014, para mayor información refiérase al datasheet (ANEXO 1)

²² **Complex Twiddle Factors.** - para la FFT se refiere a coeficientes constantes que se multiplican por los datos de la FFT durante su ejecución.

La operación que realiza esta función esta basada en la siguiente fórmula:

$$F(k) = \frac{1}{N} \sum_{n=0}^{N-1} (f(n)WN(kn))$$

$$WN(kn) = e^{\left[\frac{-j2\pi kn}{N} \right]}$$

donde

$$\begin{aligned} n &\{0, 1, \dots, N-1\} \\ k &\{0, 1, \dots, N-1\} \\ \text{con } N &= 2^m \text{ con } m \text{ elementos de } Z \\ &WN - \text{TwiddleFactors} \end{aligned}$$

La amplitud de la FFT se calcula como:

$$F[k] = \sqrt{(\text{Re}[k]^2 + \text{Im}[k]^2)}$$

El arreglo complejo de la muestras de la FFT es colocado en el parámetro Samples, la muestras de entrada son organizadas de la manera Re, Im, Re, Im,... (donde Im es siempre cero (0)). Las muestras de salida son colocadas de la misma manera pero la parte Im es diferente de cero (0), las muestras de salida son simétricas lo que quiere decir que la primera mitad de muestras de salidas son idénticas a las muestras de salida de la segunda mitad. Los datos de entrada están en un vector complejo en el que la magnitud de la parte real como de la imaginaria de cada elemento es menor que 0.5. Si es igual o mayor a este valor el resultado podría producir saturación. Note que los valores de salida están escalados por un factor de $1/N$, donde N es el tamaño de la FFT. La entrada esta en orden natural, mientras que la salida esta en *bit reverse ordering*.

- BitReverseComplex

El subprocedimiento para BitReverseComplex que esta definido por MikroBasic recibe los siguientes parámetros de la siguiente forma:

sub procedure BitReverseComplex(dim log2N as word, dim byref ReIm as array [1024] of word). [8]

Esta función realiza la reorganización compleja (en el lugar) de los bits (*Bit Reverse*), donde:

- N – es el tamaño del buffer (debe ser potencia de 2)
- ReIm – son las muestras de salida (de la FFT)

La operación que realiza es:

$$f(k) = \frac{1}{N} \sum_{n=0}^{N-1} F(n)WN(kn)$$

$$WK(kn) = e^{\left[\frac{j2\pi kn}{N} \right]}$$

donde,

F_n – es el arreglo complejo de las muestras de entrada

$n \{0, 1, \dots, N-1\}$

$k \{0, 1, \dots, N-1\}$

con $N = 2^m$ con m elementos de Z

WN – TwiddleFactors

Las muestras de entrada debe estar en el espacio de memoria Y

CAPITULO 4

ALGORITMOS PARA PROCESAMIENTO DIGITAL DE SEÑALES A SER USADOS EN EL AFINADOR

El Procesamiento Digital de Señal es una operación o transformación de una señal en un hardware digital según reglas bien definidas las cuales son introducidas al hardware a través de un software específico que puede o no manejar lenguajes tanto de alto como de bajo nivel. Procesamiento Digital de Señal se refiere al procesamiento digital de señales tales como sonido, radio y microondas usando técnicas matemáticas para realizar transformaciones o extraer información. En la práctica, las características que hacen a los *DSP's* (*Digital Signal Processing*) tan buenos en el manejo de señales los hacen adecuados para muchos otros propósitos, tales como procesamiento de gráficos de alta calidad y simulaciones en ingeniería. Eventualmente cuándo el *DSP* ha terminado su trabajo, los datos digitales pueden convertirse otra vez a señales analógicas, con calidad mejorada. [9]

Una forma de clasificar los *DSP's* y sus aplicaciones es a través de su rango dinámico. El rango dinámico es un conjunto de números, de pequeños a grandes, que pueden ser procesados en el curso de una aplicación. La capacidad del procesador es una función de su rango de datos (el número de bits manipulados) y el tipo de aritmética que posee (punto fijo o flotante). Cada tipo de procesador es ideal para un rango particular de aplicaciones. *DSP's* de 16 bits son ideales para sistemas de voz

Una de las más importantes características de un *DSP* es su capacidad de realizar operaciones de multiplicación y acumulación (**MACs**) en sólo un ciclo de reloj.

4.1 Algoritmos existentes para el cálculo de la FFT

A continuación presentamos algunos de los algoritmos existentes para el cálculo de la Transformada de Fourier.

4.1.1 Transformada de Fourier

La transformada de Fourier es una operación matemática que transforma una señal de dominio en el tiempo a dominio en la frecuencia y viceversa. Una *DFT* (Transformada de Fourier Discreta - por sus siglas en inglés) es el nombre dado a la transformada de Fourier cuando se aplica a una señal digital (discreta) en vez de una analógica (continua). Una *FFT* (Transformada Rápida de Fourier) es una versión más rápida de la *DFT* que puede ser aplicada cuando el número de muestras de la señal es una potencia de dos. Un cálculo de *FFT* toma aproximadamente $N * \log_2(N)$ operaciones, mientras que *DFT* toma aproximadamente N^2 operaciones, por lo que la *FFT* es significativamente más rápida [10].

Si la señal no es periódica, el período P aumenta hasta el infinito. Entonces, el espacio entre componentes armónicas $\frac{1}{P} = \frac{\omega}{2\pi}$ decrece en $\frac{d\omega}{2\pi}$ hasta llegar a cero. Esto conduce a un cambio de frecuencias discretas $n\omega$ a la variable continua ω . Por lo tanto:

$$c(\omega) = \frac{d\omega}{2\pi} \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (4-1)$$

donde, $c(\omega)$ es el calculo de FFT y $f(t)$ es la señal que va a ser transformada.

Si normalizamos la ecuación anterior dividiendo por $\frac{d\omega}{2\pi}$, nos queda

$$\frac{c(\omega)}{d\omega/2\pi} = F(j\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (4-2)$$

donde $F(j\omega)$ se define como la Integral de Fourier o Transformada de Fourier [10].

4.1.2 Transformada Discreta de Fourier

En el dominio digital, las señales carecen de continuidad dentro de un intervalo determinado produciendo saltos discretos en tiempo y en amplitud. Si la señal $f(t)$ la representamos ahora en el dominio digital como $f[n]$, y deseamos descomponerla en su Serie de Fourier, debemos buscar una relación entre ambos espacios.

La Transformada Discreta de Fourier (*DFT*) de una señal muestreada $f[n]$ de longitud, es proporcional a los coeficientes de la Serie de Fourier de la misma señal en dominio analógico. Más precisamente, la DFT de N muestras de un período completo es igual a N veces los coeficientes de su Serie de Fourier en el espacio continuo. Para prevenir aliasing ²³ se ha de asegurar que el ancho de banda de la señal sea al menos la mitad de la frecuencia de muestreo (Nyquist). Si $f(t)$ es de banda limitada, puede ser muestreada en intervalos de T segundos sin aliasing. Una manera de muestrear la señal dentro de una expresión integral es mediante la multiplicación de la misma por un tren de impulsos en tiempo continuo [10],

$$\Psi_T(t) \stackrel{\Delta}{=} T \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (4-3)$$

donde, $\Psi_T(t)$ es la señal muestreada, T es el período de muestreo y $\delta(t - nT)$ representa el tren impulsos. Debido a que deseamos encontrar la Serie de Fourier de la señal periódica muestreada $f(nT)$, reemplazamos $f(t)$ en

$$f_s(t) \stackrel{\Delta}{=} f(t) \cdot \Psi_T(t) \quad (4-4)$$

$f_s(t)$, es la señal periódica muestreada.

Y por examinación de las propiedades de la función delta, la Serie de Fourier de $f_s(t)$ es,

$$F_s(w_k) = \frac{1}{P} \int_0^P f_s(t) e^{-jw_k t} dt = \frac{1}{P} \sum_{n=0}^{\lfloor P/T \rfloor - 1} f(nT) e^{-jw_k nT} \quad (4-5)$$

²³ **Aliasing**- es el efecto que causa que señales continuas distintas se tornen indistinguibles cuando se les muestrea digitalmente. Cuando esto sucede, la señal original no puede ser reconstruida de forma unívoca a partir de la señal digital.

Si el período de muestreo T es escogido de manera que divida al periodo P de la señal en un número entero, entonces el número de muestras dentro de la integral es el número entero $N = P/T$, con lo que obtenemos

$$F_s(\omega_k) = \frac{T}{P} \sum_{n=0}^{N-1} f(nT) e^{-j\omega_k nT} \quad (4-6)$$

$$F_s(\omega_k) \stackrel{\Delta}{=} \frac{1}{N} DFT_{N,k}(f_p), \quad (4-7)$$

$$k = 0, \pm 1, \pm 2, \dots$$

donde $f_p \stackrel{\Delta}{=} [f(0), f(T), \dots, f((N-1)T)]$, f_p es la señal muestreada, *DFT (Discret Fourier Transform)* Por lo tanto $F_s(\omega_k) = F(\omega_k)$ para todo valor de k donde la señal periódica de banda limitada tiene un armónico diferente de cero. Cuando N es impar, $F(\omega_k)$ puede ser diferente de cero para $k \in [-(N-1)/2, (N-1)/2]$, mientras que para N impar, el máximo rango para armónicos diferente de cero es para $k \in [-(N/2-1), N/2-1]$.

Calcular el espectro de Fourier, continuo en frecuencia y periódico con 2π , a partir de una señal $f(n)$ arbitraria y discreta en el tiempo, requiere una cantidad de tiempo de computación infinito debido a que:

1. El sumatorio opera sobre una secuencia de infinita longitud,
2. La frecuencia es continua sobre el intervalo $[-2\pi, 2\pi]$, lo cual conduciría a un infinito número de funciones básicas del tipo $e^{j\omega_k n}$

En la práctica, consideramos una cantidad finita de muestras debido al tiempo finito de medida y límite de almacenamiento. Las N muestras pueden ser escogidas mediante el inventanado de la señal $f[n]$ con una función rectangular $w_r[n]$,

$$\hat{f}(n) = f(n) \cdot w_r(n) = \begin{cases} 0 & n < 0 \\ f(n) & 0 \leq n \leq N-1 \\ 0 & n > N-1 \end{cases} \quad (4-8)$$

donde $\hat{f}(n)$ es la señal muestreada.

Esto nos soluciona el sumatorio dentro de un rango infinito pero no resuelve el problema de las frecuencias continuas. De todas formas, debido a que la nueva señal consiste en N muestras, simplemente limitamos el análisis a N frecuencias uniformemente espaciadas dentro del espectro.

Nota: La resolución frecuencial depende de la frecuencia de muestreo y del número de muestras utilizadas para el cálculo de la DFT. El resultado puede ser interpretado como una serie de recipientes a frecuencias específicas repartidas sobre un rango de 0 a $fs/2$ (Hz) (fs es la frecuencia de muestreo) espaciadas en fs/N (Hz). Si las N muestras contienen uno o varios períodos completos, la componente frecuencial entrará dentro de un recipiente frecuencial. Si las N muestras no contienen uno o varios períodos completos, algunos componentes frecuenciales serán repartidos en recipientes adyacentes. Por lo tanto fs/N determina la resolución del dominio frecuencial [10].

4.1.3. Aplicación de la Fast Fourier Transform (FFT)

Para realizar el cálculo de la *DFT* se requiere una gran cantidad de operaciones. Por ejemplo, para una *DFT* de N puntos habrá N multiplicaciones complejas y N sumas complejas. Luego habrá N componentes armónicas para ser evaluadas. Es decir, el cálculo de una *DFT* de N puntos requiere N^2 multiplicaciones complejas y $N(N - 1)$ sumas complejas[8]. Si $N = 1024$, se requiere una cantidad aproximada de dos millones de operaciones complejas [10].

Diferentes técnicas son utilizadas para aprovechar las propiedades de simetría de la *DFT* y disminuir el número de operaciones necesarias para su cálculo. El desarrollo de la *Fast Fourier Transform (FFT)* para el cálculo de la *DFT* de señales discretas en el tiempo y de duración finita, utiliza técnicas computacionales para el procesamiento y el análisis de señales.

4.1.4 Método de Interpolación Lineal para el Cálculo de la FFT

La Interpolación Lineal funciona dibujando una línea recta entre dos muestras vecinas y devolviendo el punto apropiado a lo largo de esa línea. Específicamente, determina η como un número entre 0 y 1 que representa por cuanto queremos interpolar una señal y entre un instante n y un instante $n+1$ [10]. Entonces podemos definir el valor linealmente interpolado de la siguiente manera

$$\hat{y}(n + \eta) = (1 - \eta).y(n) + \eta.y(n + 1) \quad (4-9)$$

Para $\eta = 0$, obtenemos exactamente $y(\eta) = y(n)$, y de la misma forma para $\eta = 1$, obtenemos $y(\eta+1) = y(n+1)$. Entre ambos casos el error de interpolación $|y(n + \eta) - y(n + \eta)|$ es diferente de cero, excepto cuando la señal $y(t)$ es una función lineal entre $y(n)$ y $y(n + 1)$. La expresión anterior la podemos rescribir escogiendo η como factor común. De esta manera nos queda un solo término multiplicativo,

$$\hat{y}(n + \eta) = y(n) + \eta.[y(n + 1) - y(n)] \quad (4-10)$$

De esta manera, la complejidad computacional de la interpolación lineal queda reducida en dos sumas y una multiplicación por cada muestra de salida. En nuestro caso el objetivo es adecuar, una vez muestreada la señal $y(t)$, a una ventana de duración fija. Por lo tanto trataremos de contraer o expandir dicha señal en el tiempo discreto. Para ello es necesario calcular la relación de frecuencia real/deseada. La parte entera de este resultado determinará el valor de n y mediante la parte fraccional obtendremos el valor de η .

4.1.5 Remuestreo mediante Diezmado e Interpolación para Cálculo de la FFT

El Procesado Multirate es básicamente una eficiente técnica para cambiar la frecuencia de muestreo de una señal digitalmente. Los procesos de diezmado e interpolación son las operaciones fundamentales del procesado Multirate. Estas operaciones permiten reducir o incrementar la frecuencia de muestreo sin errores significativos. [10]

4.1.5.1 Reducción de la frecuencia de muestreo para Cálculo de la FFT

El diagrama de bloques que representa el proceso de diezmar una señal $f[n]$ por un factor entero M , [10] aparece en la Figura: 4.1. Consiste en un filtro digital antialiasing, $h(k)$, y un compresor de la frecuencia de muestreo de f_s a f_s/M . La relación entrada-salida de este proceso es

$$y(m) = w(mM) = \sum_{k=-\infty}^{\infty} h(k).f(mM - k) \quad (4-11)$$

donde $y(m)$ es la relación de entrada/salida del proceso, $h(k)$ es el filtro digital y $f(mM-k)$ es la señal de acuerdo a la frecuencia de muestreo, donde:

$$w(n) = \sum_{k=-\infty}^{\infty} h(k).f(n - k) \quad (4-12)$$

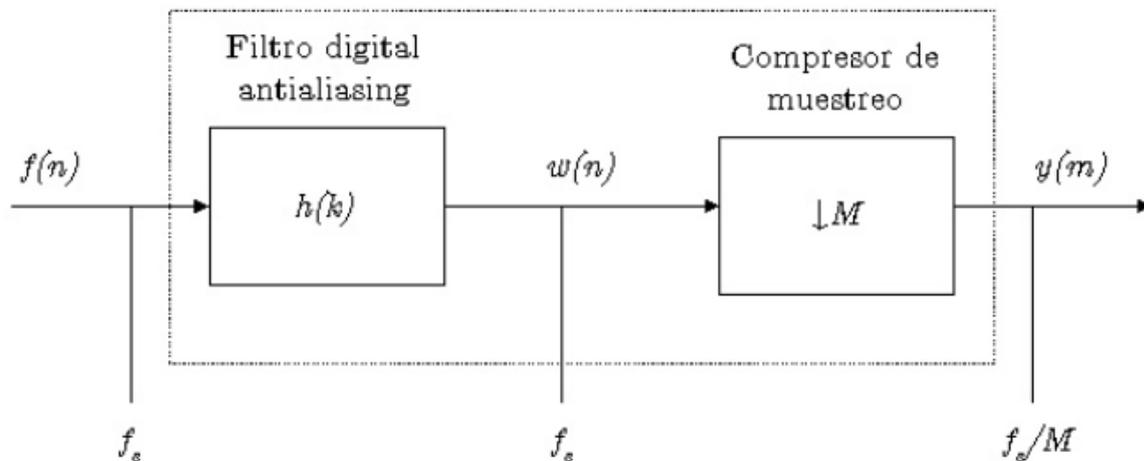


Figura: 4.1. Diagrama de bloques del diezmo por un factor M

4.1.5.2 Incremento de la frecuencia de muestreo para el Cálculo de la FFT

Dada una señal $f[n]$, y una frecuencia de muestreo f_s , el proceso de interpolación, descrito en la Figura: 4.2, incrementa la frecuencia de muestreo a Lf_s . [10] Esto significa que es un proceso de expansión y la relación entrada-salida es

$$y(m) = \sum_{k=-\infty}^{\infty} h(k) \cdot w(m-k)$$

donde

$$w(m) = \begin{cases} f(m/L), & m = 0, \pm L, \pm 2L, \dots \\ 0 & \text{otherwise} \end{cases}$$

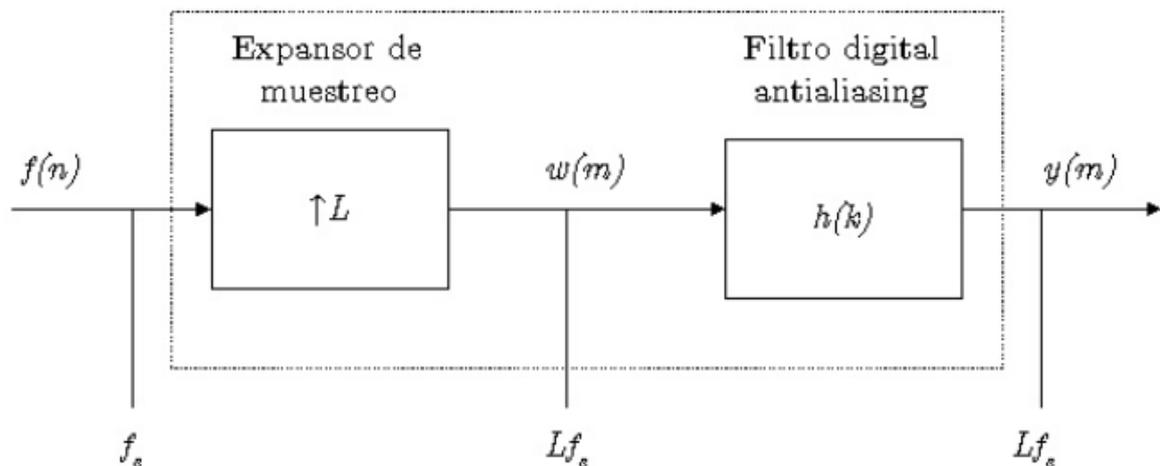


Figura: 4.2. Diagrama de bloques de la interpolación por un factor L

4.2 Selección del algoritmo apropiado

La transformada rápida de Fourier es un algoritmo rápido para la evaluación numérica de integrales de Fourier, su importancia radica en la rapidez de cálculo conseguida, pero mucho más en otro tipo de aplicaciones: ecualización y filtrado en equipos de audio/vídeo en tiempo real, comunicaciones, etc.

La diferencia de velocidad de cálculo entre la tradicional transformada discreta y la FFT aumenta según aumenta el número de muestras a analizar, ya que mientras una aumenta el número de operaciones necesarias para la resolución de forma exponencial, la otra lo hace de forma prácticamente lineal.

4.3 Código del algoritmo seleccionado en Basic

Para el cálculo de la FFT para la medición de la frecuencia el dsPIC incluye ya una función para realizar este cálculo, la cual es transparente, a continuación se presenta el código que incluye el ingreso de las muestras análogas las cuales se convierten a muestras digitales, estas pasan a la FFT y su respuesta se almacena de la variable Samples, el siguiente paso es ordenar dichas muestras para luego continuar con el cálculo de la frecuencia máxima y su escritura en el LCD.

```

SampleInput

FFT(7, @TwiddleCoeff_128, Samples)

BitReverseComplex(7, Samples)

WriteData

```

Para la escritura de la frecuencia máxima se realiza el cálculo de esta elevando la parte real al cuadrado y la parte imaginaria al cuadrado luego de cual se suman los dos resultados y se saca la raíz cuadrada, esta respuesta es la frecuencia máxima de la señal ingresada, la cual se deberá multiplicar por un factor de escalamiento, que viene dado

por: $r = \frac{fs}{N}$ [10], donde fs es la frecuencia de muestreo y N es el número de muestras.

```

'Escritura de el valor maximo de frecuencia
sub procedure WriteData
dim Re, Im,
  j, k, max as word
  Rer, Imr, tmpR as float

  j = 2
  k = 0
  max = 0
  freq = 0
  while k <= 31
    Re = Samples[j]
    inc(j)

    Im = Samples[j]
    inc(j)

    Rer = Fract2Float(Re)

```

```
    Imr = Fract2Float(Im)

    tmpR = Rer * Rer
    Rer = tmpR
    tmpR = Imr * Imr
    Imr = tmpR
    tmpR = sqrt(Rer + Imr)
    Rer = tmpR*256.
    Re = Rer

    if Re > max then
        max = Re
        freq = k
    end if

    inc(k)

wend

if ((valins = 1) and (valcue = 3)) or ((valins = 1) and (valcue = 7)) or((valins = 2) and
(valcue = 6)) or((valins = 2) and (valcue = 3))or((valins = 4) and (valcue = 6)) then
    freq = freq * 37
    Lcd_Cmd(LCD_CLEAR)
    escribir(1,1,"frecuencia:")
    WordToStr(freq, txt)
    escribir(2,1,txt)
    Delay_ms(1000)
else
    freq = freq * 7
    Lcd_Cmd(LCD_CLEAR)
    escribir(1,1,"frecuencia:")
    WordToStr(freq, txt)
    escribir(2,1,txt)
    Delay_ms(1000)
end if

end sub
```

CAPITULO 5

DISEÑO DEL PROTOTIPO

El diseño del prototipo esta basado en la flexibilidad y comodidad que debe brindar el afinador de instrumentos musicales al usuario, siendo portátil y de fácil manejo, cuenta con un botón de encendido y apagado, dos botones de control uno de los cuales realiza la selección de la opciones que posee el afinador y el otro ayuda a la búsqueda de dichas opciones, tiene un micrófono por el cual se ingresa la señal de audio al microcontrolador y un LCD que permite visualizar el menú del afinador de instrumentos haciéndolo más amigable.

5.1 Hardware

Diagrama de bloques

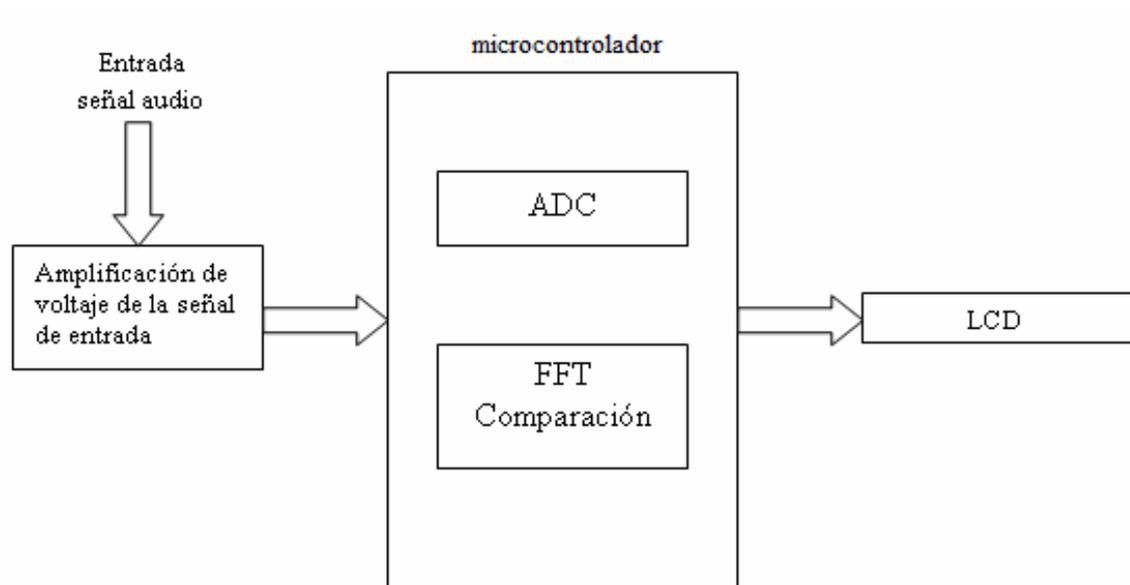


Figura: 5.1. Diagrama de bloques

El primer bloque es el de ingreso de la señal y amplificación de la misma, es necesario realizar una amplificación de la señal de audio dado que el micrófono que se está utilizando tiene un voltaje de salida máximo de 100mV, este valor de voltaje no es suficiente para que el microcontrolador pueda adquirir los datos analógicos para realizar la conversión analógica – digital, en esta etapa se usa el amplificador operacional LM386.

Una vez que se realizó el ingreso de la señal de audio el siguiente paso es tomar las muestras analógicas necesarias para el obtener el espectro de frecuencias por medio de la FFT, para obtener un valor más cercano a la realidad fue necesario hacer una discriminación de los valores de voltaje que ingresan al microcontrolador, por lo que solo se está tomando valores entre 3.5V y 4.5V. Una vez obtenidos los valores de voltaje se procede a digitalizar la señal, la cual pasará luego a la etapa en la se que obtendrá el espectro de frecuencias del que se sacará el valor con la mayor energía siendo este el resultado de frecuencia de la señal ingresada. Existe ya en el dsPIC una tabla de valores para las frecuencias de cada una de las cuerdas que pueden ser afinadas, con la se comparará el valor de frecuencia obtenido en la etapa anterior, en la etapa de comparación existen tres posibilidades de comparación, la primera que el valor de frecuencia obtenido sea mayor que el valor asignado en la tabla (debe aflojar la cuerda), la segunda que el valor de frecuencia obtenido sea menor que el valor asignado en la tabla (debe ajustar la cuerda) y la tercera que el valor de frecuencia obtenido sea igual que el valor asignado en la tabla (afinación correcta).

Teclas de control

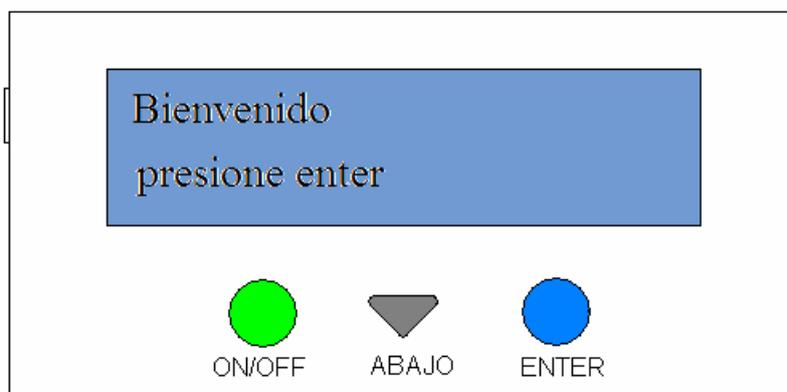


Figura: 5.2. Teclas de control del Afinador

El afinador de instrumentos musicales cuenta con tres teclas para su control, una tecla de encendido y apagado del afinador, una tecla de *enter* para el acceso a las diferentes opciones del afinador, y una tecla de manejo (*abajo*) dentro del menú para escoger las opciones que se encuentran dentro del afinador.

Encienda el prototipo (presione el botón de encendido), el momento de encender el afinador se presenta en el LCD un mensaje de bienvenida, el cual permanecerá en la pantalla hasta que se presione la tecla *enter* para continuar. Una vez que ha presionado la tecla *enter*, en el LCD aparecerá el primer instrumento que puede seleccionar (guitarra), si desea afinar una guitarra presione *enter*, caso contrario presione la tecla de *abajo*, cada vez que presione esta tecla aparecerá en el LCD un instrumento diferente (violín, bajo o chelo) para seleccionar cualquiera de estos instrumentos solo debe ubicarse en el que desea y presionar *enter*. Seleccionado el instrumento que desea afinar el siguiente paso es escoger la cuerda que va a ser afinada, de la misma forma que hizo la selección del instrumento, ahora deberá seleccionar la cuerda correspondiente para ello se presentará otro menú en el LCD, para cada instrumento existe un menú diferente de cuerdas, al igual que en la selección anterior deberá buscar la nota deseada por medio de la tecla *abajo* y presionar *enter* para afirmar la opción.

Con esto se concluye el proceso de selección, a continuación se procederá a afinar el instrumento, se le pedirá que toque la cuerda y en unos instantes se le indicará que acción debe tomar, ya sea esto ajustar la cuerda, aflojar la cuerda o no realizar ninguna acción (cuando el mensaje es Afinación correcta). Si el mensaje fue de ajustar o aflojar deberá repetir el proceso y tocar la cuerda nuevamente con el cambio correspondiente hasta que se presente en el LCD el mensaje Afinación correcta.

Entrada de audio



Figura: 5.3. Diagrama de la Entrada de Audio

En la Figura: 5.3 se muestra el ingreso de la señal de audio por medio de un micrófono y luego se envía hacia el dsPIC, este proceso se realiza por medio de un amplificador de voltaje, debido a que la señal de entrada del micrófono tiene una señal máxima de 100mV, por lo que es necesario realizar una amplificación para poder ingresar al dsPIC y realizar la conversión A/D, para realizar esta etapa es necesario un voltaje de entrada comprendido entre 1V y 5V (para mayor información refiérase al datasheet [7]). Para esto se utiliza el amplificador operacional LM386 que es alimentado con 5V.

Conversor A/D

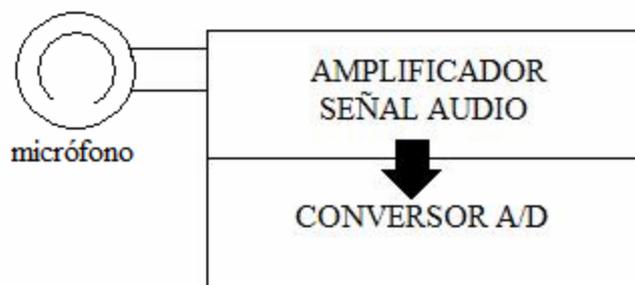


Figura: 5.4. Diagrama para la conversión A/D

El conversor A/D integrado en el dsPIC30F3014 es un conversor de 12 bits que permite la conversión de una señal de entrada analógica a un número digital de 12 bits de los cuales 2 con el signo. Este modulo esta basado en una arquitectura para un registro de aproximación sucesiva (SAR de sus siglas en ingles *Successive Approximation Register*) y provee un rango de muestreo máximo de 100 ksps. Este modulo tiene 16 entradas analógicas las cuales están multiplexadas en un amplificador de muestra y retención (Sample and Hold). La salida de dicho amplificador es la entrada un conversor A/D el cual nos da el valor numérico de la señal de voltaje. Este conversor tiene la característica de poder operar mientras el dispositivo se encuentra en modo Sleep (dormido) con un oscilador RC.

Cálculo de la FFT

Luego de haber seleccionado el instrumento y la nota deseada, el siguiente paso es realizar el cálculo de la frecuencia de la señal ingresada, para realizar este proceso se hace uso de la función FFT que tiene integrada el dsPIC esta función esta definida como sub procedure **FFT(dim log2N as Word, dim TwiddleFactorsAddress as Longint, dim byref Samples as array[1024] of Word)** [8], esta función nos dará el cálculo de la frecuencia fundamental de la onda de entrada, el objetivo de este proceso es convertir una señal en el dominio del tiempo al dominio de la frecuencia para esto es necesario una secuencia digital de entrada de N puntos muestreada a una frecuencia “fs”Hz y su salida es una secuencia digital de N puntos con componentes de frecuencia en “fs/2”Hz en la Figura: 5.5. se muestra un ejemplo de lo que obtiene luego de este proceso, la señal ingresada es un seno (x) a 800Hz con una frecuencia de muestreo fs de 8000Hz.

$$x[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}}$$

La FFT reduce los requerimientos computacionales ya que usando el algoritmo de decimación en el tiempo para la FFT en donde una secuencia de datos de N puntos es dividida en dos secuencias de datos N/2 de manera que se separan en muestras de datos pares (n = 0, 2, 4, ...) y en muestras impares (n = 1, 3, 5, ...), cada secuencia es repetidamente una decimación en el tiempo, obteniendo una secuencia de N/2 de 2 puntos cada una, en la Figura: 5.6 se muestra un ejemplo de la decimación en el tiempo.[11] Es necesario para este cálculo el uso de una factores establecidos en la memoria del dsPIC (Twiddle Factors), estos coeficientes se definen como $e^{\frac{-j2\pi k}{N}}$ y se conocen como Twiddle Factors, para la operación se utiliza N/2 twiddles de (k = {0, 1, ..., (N/2)-1}) de donde se obtiene una muestra en términos de senos y cosenos (proveniente de la Ecuación de Euler). [11] Una vez realizado este cálculo se procede a comparar dicha frecuencia con la frecuencia de la nota seleccionada que esta almacenada en memoria, de acuerdo al resultado de esta comparación, se indicará al usuario la acción que debe tomar, sea esta aflojar o ajustar la cuerda de su instrumento.

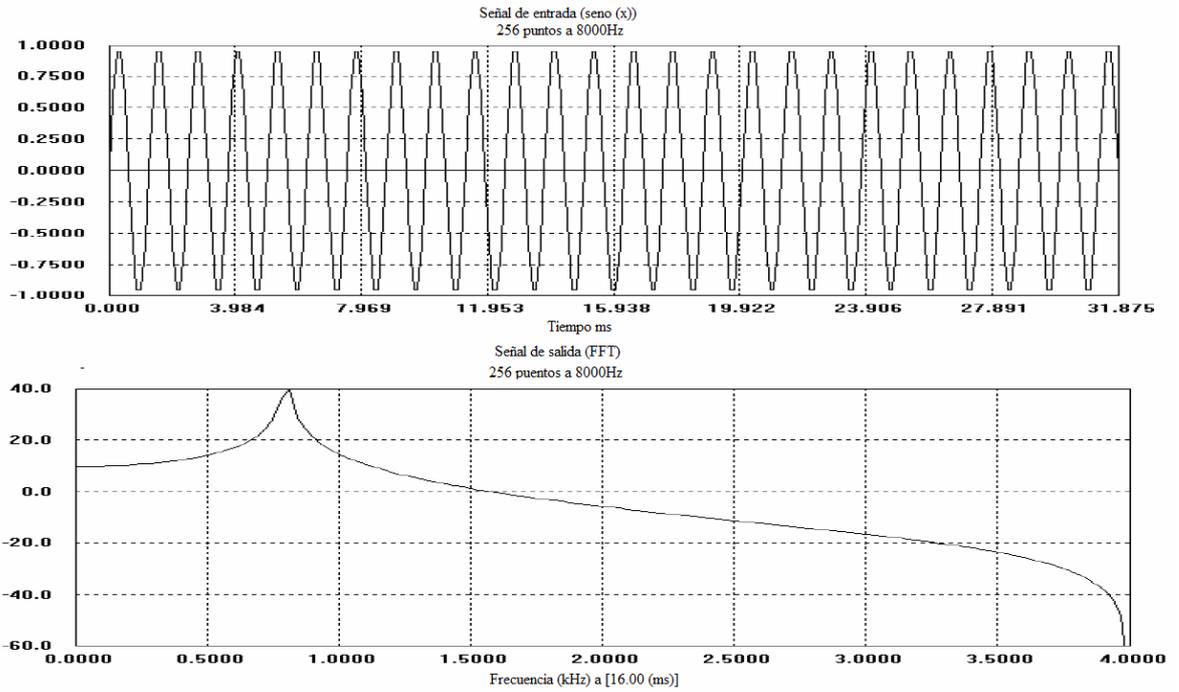


Figura: 5.5. Ejemplo de una señal de entrada (seno (x))

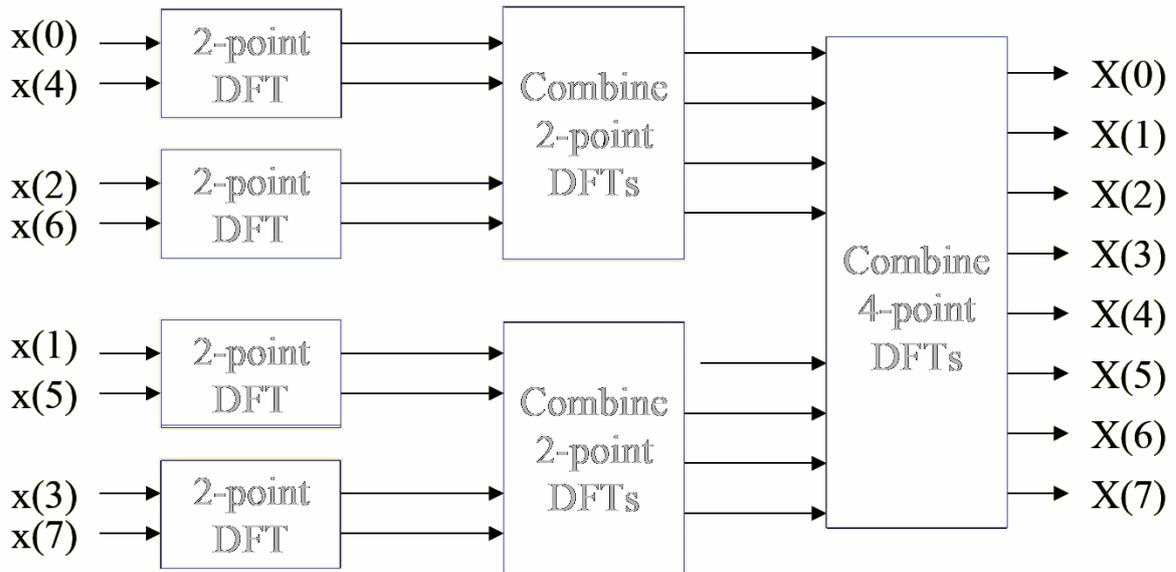


Figura: 5.6. Decimación en el tiempo

Comparación de la frecuencia calculada con la frecuencia establecida para una nota musical según la Escala Temperada

El proceso de comparación es el que nos dice que acción debemos tomar para afinar la cuerda del instrumento que hemos seleccionado, este proceso cuenta con tres sentencias de comparación, la primera en la que se define si la frecuencia ingresada es mayor que la frecuencia de comparación, la segunda en la que se define si la frecuencia ingresada es menor que la frecuencia de comparación y la tercera en la que se define si la frecuencia ingresada es igual que la frecuencia de comparación, de acuerdo al resultado obtenido se toma la acción necesaria.

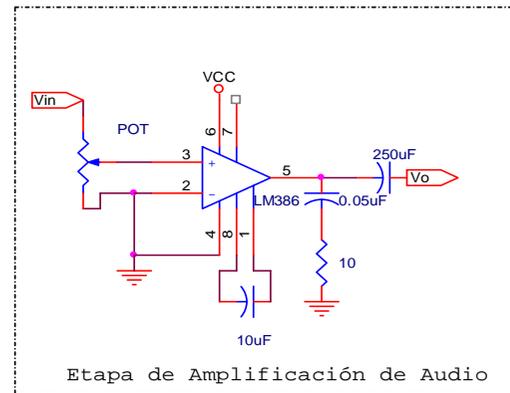
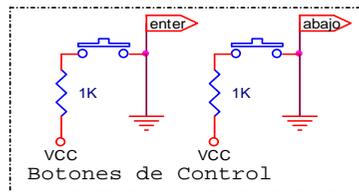
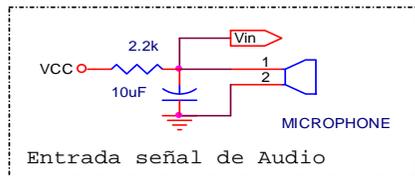
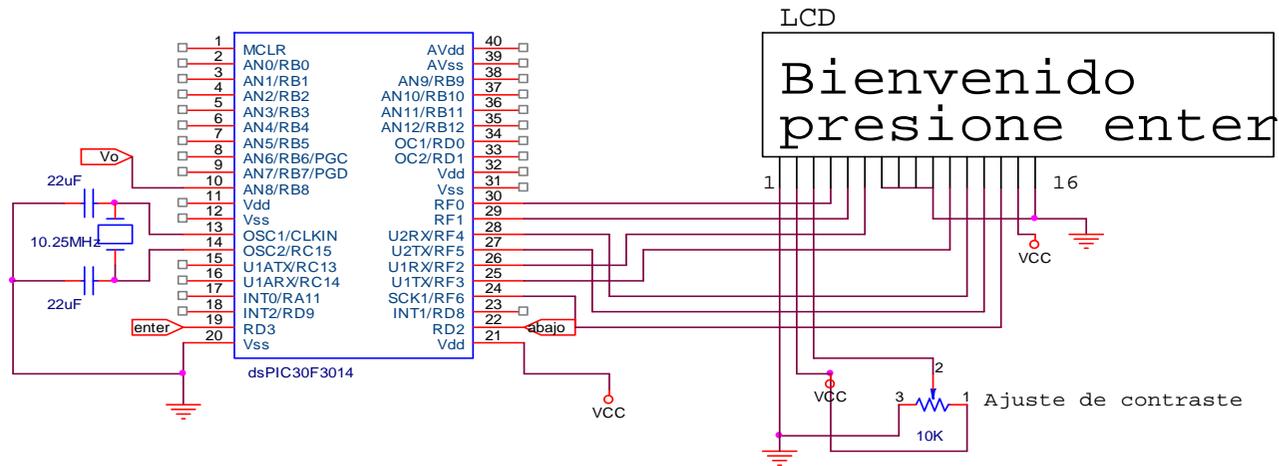
LCD e interfaz gráfica

El afinador de instrumentos musicales esta provisto de un LCD para la interacción con le usuario, brindándole la oportunidad de seleccionar la tarea que desea realizar, así como también visualizar la respuesta que le proporciona el dispositivo. Una vez que se ha escogido una cuerda en especial para afinar, se deberá ingresar la señal de audio de la cuerda por medio del micrófono. Dicha señal será procesada, pasando primero por un conversor análogo – digital, luego se realiza la transformada de fourier de la señal para determinar la frecuencia fundamental de onda, la cual se compara con una base de datos establecida bajo las frecuencias de la escala temperada [6], luego de lo cual se observara en el LCD un mensaje y el valor de frecuencia que tiene la señal ingresada para realizar su ajuste.

Alimentación

Para el funcionamiento del dispositivo solo es necesario de una fuente de 5Vdc. Por lo que se decidió alimentar el afinador con una batería de 9V y mediante un divisor de voltaje para obtener la alimentación necesaria para el mismo.

Diagrama esquemático del circuito completo para el afinador de instrumentos musicales con el dsPIC30F3014



5.2 Software

En los siguientes puntos se resume las características del software utilizado en el desarrollo del proyecto. Se decidió usar este software ya que es de fácil uso, amigable para el usuario, además que cuenta con herramientas para simulación y de esta manera poder realizar pruebas que ayudan en la programación y verificación del procedimiento que realiza el dsPIC, y sobre todo es útil para el dsPIC30F3014. Este software tiene la ventaja de que incluye ya varias funciones desarrolladas, con las cuales se puede hacer tanto programas básicos como programas mas complejos, además que brinda una ayuda muy completa que es de fácil comprensión. Por ser un entorno que utiliza lenguaje Basic es de fácil programación Es por estas razones que se escogió este software como medio de programación.

5.2.1 Consideraciones para manejo de MikroBasic para proyectos con DSPIC

El software utilizado es mikroBasic, mikroElektronika Basic compiler for Microchip dsPIC 30/33 and PIC 24 microcontrollers Version: 4.0.0.0 © 2002-2007 mikroElektronika. All rights reserved. MikroBasic [12] es una poderosa herramienta creada para los microprocesadores dsPIC30/33 y PIC24, ha sido diseñada para proveer al usuario la solución más fácil posible para el desarrollo de aplicaciones para sistemas embebidos, sin comprometer su desempeño o control. Este IDE altamente avanzado cuenta con librería para hardware, una documentación comprensiva y varios ejemplos listos para ser usados, lo que permite una programación rápida de los microcontroladores.

La creación de aplicaciones en MikroBasic es muy fácil e intuitiva. El Project Wizard le permite establecer las características del programa como el nombre, el tipo de integrado, y los bits de control. Este programa le permite distribuir sus proyectos en tantas carpetas como usted desee. El proyecto contiene la siguiente información:

- Nombre del proyecto y una descripción opcional
- El tipo de microcontrolador
- Palabra de configuración y el reloj necesario para el funcionamiento del integrado
- Lista de los archivos fuente del proyecto junto con su raíz

A continuación presentamos los pasos a seguir para la creación de un nuevo proyecto:

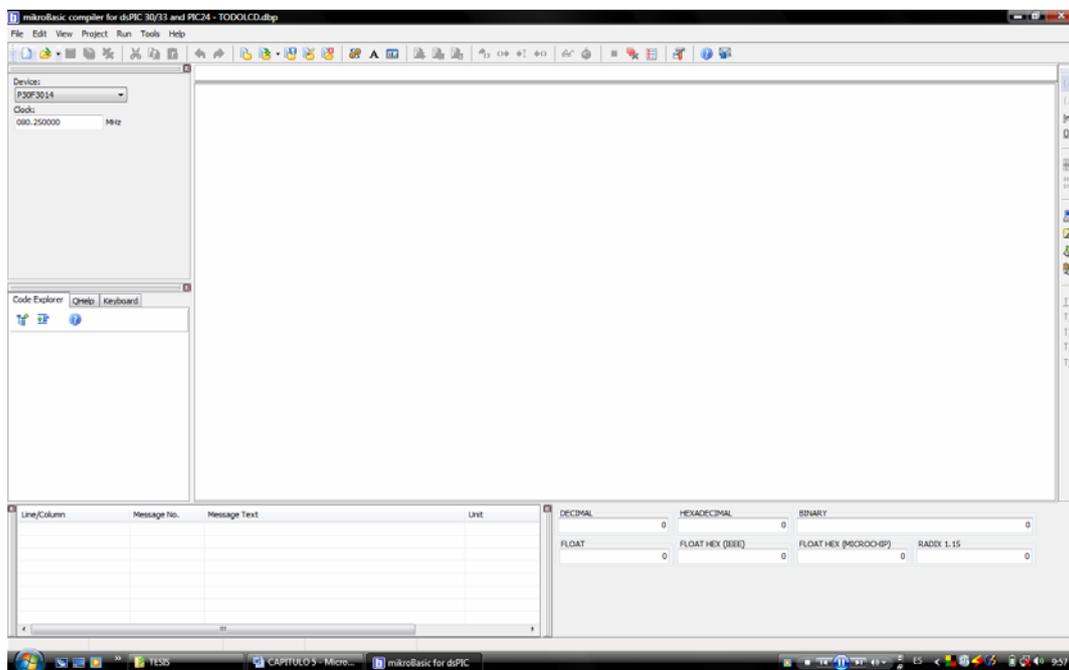


Figura: 5.7. Pantalla de inicio de mikroBASIC compiler for dsPIC30/33 and PIC24

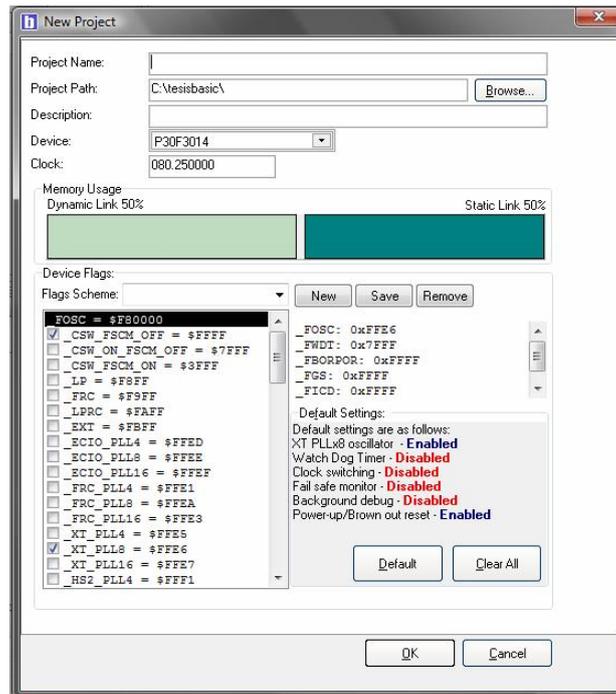
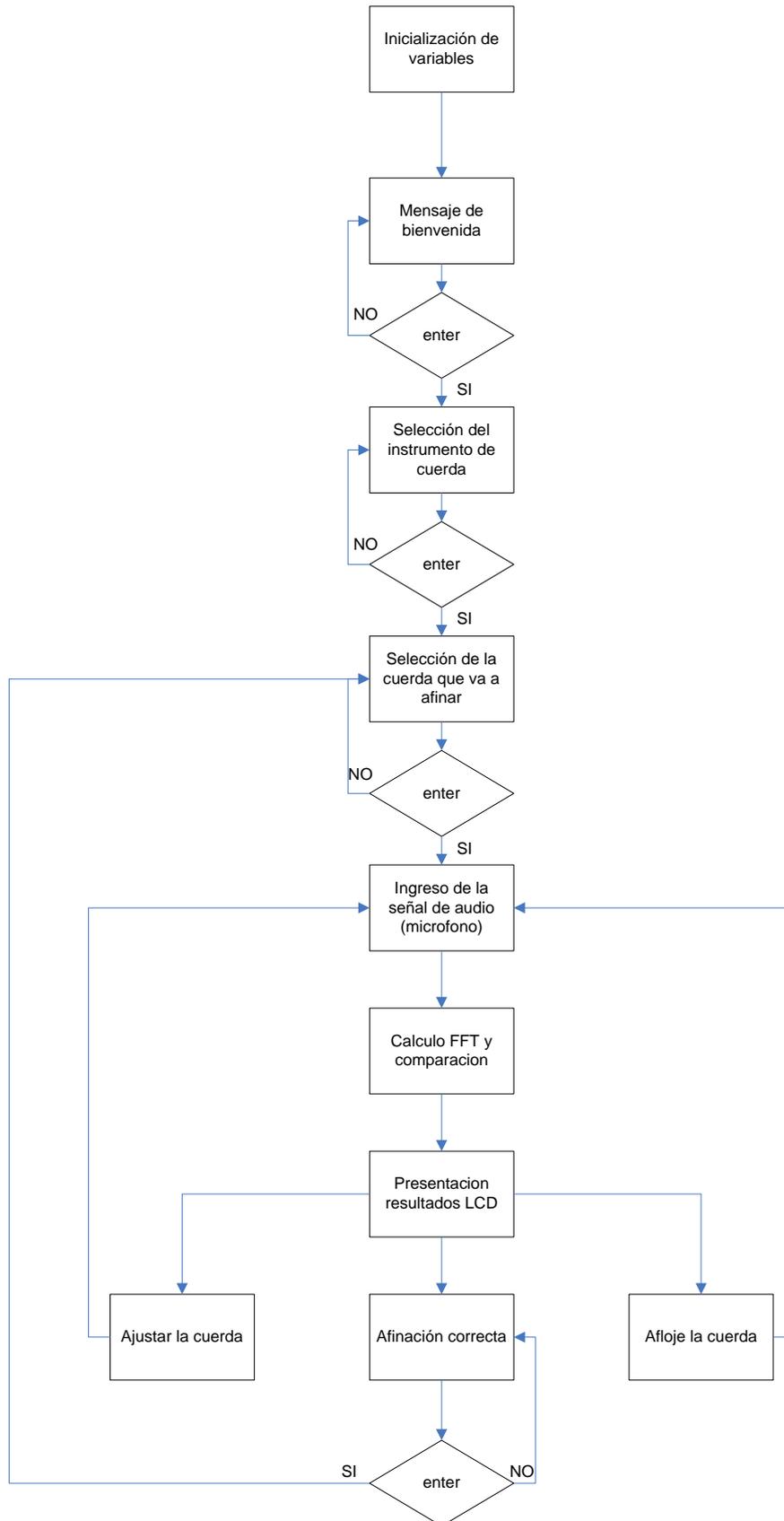


Figura: 5.8. Definición de los parámetros básicos para la programación del dsPIC30Fxxxx

La Figura: 5.8 muestra la ventana que se activa al hacer click en el botón New Project , en esta ventana se debe llenar parámetros como: el nombre del proyecto, la carpeta en la que se va a guardar el proyecto, el tipo de integrado que se va a programar, el reloj con el que va a trabajar el microcontrolador y por último se setea los bits de control haciendo click en el botón default , para empezar a programar solo se debe hacer click en el botón , con esto se completa el proceso para iniciar un nuevo proyecto.

5.2.2 Diagrama de flujo para el afinador de instrumentos musicales con el dsPIC30F3014



De forma general programa esta organizado de la siguiente manera:

Primero es necesario una inicialización de todas las variables necesarias y puertos que se va a ocupar tanto para el ingreso como para la salida de señales, el siguiente paso es presentar en el LCD un mensaje de bienvenida, el cual permanecerá hasta que se presione la tecla *enter*, una vez que se ha presionada esta tecla se presenta en el LCD el primer menú que nos permitirá escoger el instrumento que deseamos afinar, entre las opciones tenemos: guitarra, violín, bajo y chelo, la selección de cualquiera de estos instrumentos se la hará por medio de las teclas de control (*enter* y *abajo*), luego se continúa con la selección de la cuerda que desea afinar, existe un menú diferente para cada instrumento que permite afinar el prototipo, la selección de la cuerda, al igual que la selección del instrumento, se la realiza por medio de las teclas de control (*enter* y *abajo*). Seleccionados el instrumento y la cuerda el siguiente paso es ingresar la señal de audio por medio del micrófono, una vez ingresada la señal el microcontrolador se encarga de obtener el valor de la frecuencia el que se comparará con la base de datos que se encuentra grabada en el microcontrolador para saber a que valor de frecuencia se debe llegar para tener una correcta afinación, dependiendo del valor de frecuencia ingresado y la comparación realizada, se dirá al usuario la acción que debe tomar sea esto ajustar o aflojar la cuerda. Una vez alcanzado el valor de frecuencia necesario el usuario deberá presionar la tecla *enter* para continuar con la afinación de otra cuerda del instrumento elegido o caso contrario solo debe apagar el afinar para terminar con la afinación.

Nota.- para un diagrama de flujo mas detallado sobre cada rutina favor refiérase al Anexo 1.

5.2.3 Código del programa, explicación

La programación del prototipo esta diseñada de la siguiente manera: al encender el dispositivo  ON/OFF, se presenta un mensaje de bienvenida y junto a este el mensaje “presione enter”, seguido esto es necesario presionar la tecla *enter*  para continuar con la selección del instrumento que desea afinar, la selección del instrumento se la

realiza con la tecla *abajo*  y para ingresar a las opciones del instrumento es necesario presionar la tecla *enter*. Una vez realizada la selección se procede a escoger la cuerda a ser afinada de la misma manera que se selecciono el instrumento.

Seleccionadas las dos operaciones, se presenta un mensaje (“toque la cuerda”) que indica que el afinador esta listo para ser usado, en este momento se activa el conversor A/D del dsPIC para tomar las muestras necesarias para obtener la frecuencia fundamental de la señal de audio ingresada. Calculada la frecuencia fundamental el siguiente paso es realizar una comparación con una pequeña base de datos que contiene las frecuencias con las que se debe comparara y así establecer la frecuencia necesaria de la cuerda seleccionada, con dicho valor se compara la frecuencia de la señal y a continuación se indica en el LCD el mensaje correspondiente, sea este aflojar o ajustar la cuerda cuando se presenta uno de estos dos mensajes se procede a repetir la función de afinación donde se ingresa la señal de audio, esto se lo realizará las veces necesarias hasta conseguir la frecuencia deseada, hecho esto se presentará un mensaje de Afinación correcta, con lo que se concluye el proceso de afinación de la cuerda y se puede continuar con la siguiente cuerda si desea, caso contrario solo debe apagar el afinador.

Nota: Para ver el programa completo por favor refierase al Anexo 2.

5.2.4 Explicación de los algoritmos más importantes

Entre los algoritmos mas importantes se encuentran:

5.2.4.1. Algoritmo para funcionamiento de los botones de control

En este algoritmo se realiza la lectura del pin 2 del Puerto D, que es el pin que por donde entra la señal del botón *enter*, gracias a las librerías con las que cuenta el MikroBasic, es posible utilizar la librería Util [13] donde se encuentra esta función, lo que se debe hacer es definir el puerto a utilizar, el pin de lectura, el tiempo en milisegundos de lectura del puerto y el estado de activación sea 1 o 0. Esta algoritmo fue creada para el funcionamiento del botón *enter* y devuelve el valor entero de 1 si es que el botón fue presionado.

```

sub function leerpe as integer
  dim aux as integer
  aux = 0
  while aux = 0
    if Button(PORTD, 2, 50, 1) = 1 then
      aux = 1
      result = aux
    end if
  wend
end sub

```

5.2.4.2. Algoritmo de conversión Análoga - Digital

Este algoritmo fue creada para realizar el ingreso de la señal analógica en el dsPIC y digitalizarla para poder realizar el cálculo de la FFT, esta algoritmo devuelve un valor fraccional con signo que es almacenado en la variable Samples[i], la misma que luego almacenará el resultado de la FFT. El primer paso es la activación del módulo ADC del dsPIC para lo cual es necesario activar la lectura del Puerto B como análogo, en este caso la entrada análoga se la realiza por el Pin 8 del Puerto B, una vez activo el puerto se procede a tomar las muestras necesarias, en este caso se hace un muestro de 256 muestras, las cuales se convierten a datos digitales para luego ser procesadas por la FFT.

```

sub function ReadAdc as word

```

```

  ADCON1.1 = 1
  while ADCON1.0=0
    nop
  wend
  result = ADCBUFO

```

```

end sub

```

```

'Funcion de llenado de la variable Samples[]

```

```

sub procedure SampleInput

```

```

  dim i as integer

```

```

  i =0

```

```

  if ((valins = 1) and (valcue = 3)) or ((valins = 1) and (valcue = 7)) or((valins = 2)
and (valcue = 6)) or((valins = 2) and (valcue = 3)) or((valins = 4) and (valcue = 6))
then

```

```

  while i <= 255

```

```

    valor = ReadAdc
    if (valor >= 20608) and (valor <= 32752) then
        Samples[i] = valor 'muestras del ADC
        inc(i)
        Samples[i] = 0
        inc(i)
    ' Delay_us(238)
    end if
wend
else
    while i <= 255
        valor = ReadAdc
        if (valor >= 20608) and (valor <= 32752) then
            Samples[i] = valor 'muestras del ADC
            inc(i)
            Samples[i] = 0
            inc(i)
            Delay_us(238)
        end if
    wend
end if
end sub

```

5.2.4.3 Algoritmo para la presentación en LCD del valor de frecuencia fundamental de la señal de entrada

Primero se inicializa en cero (0) el valor de la frecuencia (freq) para realizar una nueva lectura, se toma la parte real y la imaginaria (que es igual a 0) de la variable *Samples* a continuación se realiza la conversión de fracción a flotante acorde con la IEEE, se calcula la amplitud de la señal de la FFT en ese instante para luego establecer la amplitud máxima de la señal, la cual es el centro de la frecuencia de la señal ingresada. De acuerdo con la teoría de la FFT para poder sacar el valor de la frecuencia el resultado de la transformación se debe multiplicar por un factor que esta dado por

$$f = \frac{Fnyquist}{\#muestras} = \frac{2000}{256} = 7.85 [10], \text{ pero para determinar el factor que más se acerque a}$$

la realidad se realizo el calculo de este en forma experimental por lo que se llega a un resultado de 7Hz, para frecuencias menores a 220Hz y para frecuencias mayores se debe utilizar el factor de 37Hz. Esto quiere decir que la resolución del dsPIC para esta aplicación es de 7Hz y 37Hz, debido a esto es necesario hacer una aproximación de todos los valores de frecuencia de las notas de cada cuerda para cada uno de los

instrumentos, en la Tabla 5.1 se presenta la correspondencia de cada nota. Para poder visualizar la frecuencia analizada es necesario realizar la conversión del valor de frecuencia de una variable *WORD* a una variable *STRING* para poder imprimir en el LCD.

Tabla 5.1. Correspondencia de frecuencias del instrumento original y el dsPIC

Nota	Valor Original (Hz)	Valor Afinador (Hz)	Diferencia (Hz)
Guitarra			
MI	82	84	2
LA	110	112	2
SOL	147	147	-
RE	196	196	-
SI	247	259	12
MI (segunda)	330	333	3
Violín			
SOL	98	98	-
RE	147	147	-
LA	220	222	2
MI	330	333	3
Bajo			
RE	73	77	4
MI	41	42	1
SOL	98	98	-
LA	55	56	1
Chelo			
DO	147	147	-
RE	73	77	4
SOL	49	49	-
LA	220	222	2

```

sub procedure WriteData
dim Re, Im,
    j, k, max as word
    Rer, Imr, tmpR as float

j = 2
k = 0
max = 0
freq = 0
while k <= 31
    Re = Samples[j]
    inc(j)

    Im = Samples[j]
    inc(j)

    Rer = Fract2Float(Re)
    Imr = Fract2Float(Im)

    tmpR = Rer * Rer
    Rer = tmpR
    tmpR = Imr * Imr
    Imr = tmpR
    tmpR = sqrt(Rer + Imr)
    Rer = tmpR*256.
    Re = Rer

    if Re > max then
        max = Re
        freq = k
    end if

    inc(k)

wend

if ((valins = 1) and (valcue = 3)) or ((valins = 1) and (valcue = 7)) or((valins = 2)
and (valcue = 6)) or((valins = 2) and (valcue = 3))or((valins = 4) and (valcue = 6))
then
    freq = freq * 37
    Lcd_Cmd(LCD_CLEAR)
    escribir(1,1,"frecuencia:")
    WordToStr(freq, txt)
    escribir(2,1,txt)
    Delay_ms(1000)
else
    freq = freq * 7
    Lcd_Cmd(LCD_CLEAR)
    escribir(1,1,"frecuencia:")
    WordToStr(freq, txt)

```

```

        escribir(2,1,txt)
        Delay_ms(1000)
    end if

end sub

```

5.2.4.4 Algoritmo de comparación para la frecuencia

La última algoritmo es para la comparación de la frecuencia analizada de la señal de entrada con una base de datos guardada en memoria para cada cuerda del instrumento seleccionado, entra a sentencias condicionales donde se determina si la frecuencia calculada es la indicada para la cuerda seleccionada, de acuerdo al resultado obtenido se presenta en el LCD el mensaje correspondiente, sea este “Ajuste cuerda”, cuando la frecuencia obtenida es menor a la deseada, “Afloje cuerda”, cuando la frecuencia obtenida es mayor a la deseada o “Afinación correcta” cuando la frecuencia obtenida es la deseada.

```

'Funcion de comparación
sub procedure comparacion (dim a, b as integer)
    dim ins as integer
    dim cue as integer
    ins = a
    cue = b
    frecuencia = afinar(a, b)
    InitAdc

comparauno:

    Delay_ms (500)

    SampleInput

    FFT(7, @TwiddleCoeff_128, Samples)

    BitReverseComplex(7, Samples)

    WriteData

    if freq < frecuencia then
        Lcd_Cmd (LCD_CLEAR)
        escribir(1, 1, "Ajuste cuerda")
    end if
end sub

```

```
    goto compara
end if

if freq > frecuencia then
  Lcd_Cmd (LCD_CLEAR)
  escribir(1, 1, "Afloje cuerda")
  goto compara
end if

if freq = frecuencia then
  Lcd_Cmd (LCD_CLEAR)
  escribir(1, 1, "Afinacion")
  escribir(2, 1, "correcta")
  Delay_ms(2000)
  bandera = 1
  goto compara
end if

compara:
  while bandera = 1
    if Button(PORTD, 2, 50, 1) = 1 then
      bandera = 2
      exit
    end if
  wend
  goto comparauno
end sub
```

CAPITULO 6

PRUEBAS Y RESULTADOS

Una vez que se ha completado el proceso de diseño e implementación del prototipo, se realizaron las pruebas necesarias a fin de obtener los resultados que definan el funcionamiento del prototipo de manera que se cumplan los objetivos planteados.

6.1. Pruebas Realizadas

Para verificar el funcionamiento del prototipo se realizó dos tipos de pruebas, la prueba Tipo A: esta prueba se realizó con generador de señales, un frecuencímetro y afinador de instrumentos para verificar el valor de frecuencia que es capaz de leer el prototipo, el procedimiento para realizar esta prueba fue ingresar un señal senoidal de 4V en la entrada análogo del prototipo y así poder leer el valor de frecuencia en el LCD, al mismo tiempo que con el frecuencímetro se verifica el valor de frecuencia enviado por el generador. La prueba Tipo B: esta prueba se realizó con un instrumento musical de cuerda (guitarra) y una afinador por software, el Instrument Tuner V2.0 de OOBOX Music Copyright (c) 2002-2005 PYXSYS [14], lo que se hizo en esta prueba fue afinar cada una de las cuerdas de la guitarra con el prototipo de afinador digital de instrumentos musicales y luego comparar el valor de frecuencia que se obtuvo con el afinador por software, de esta manera que tan acertado es el prototipo con respecto a un afinador comercial. La prueba Tipo C esta prueba se realizó con un instrumento musical de cuerda (violín) y una afinador por software, el In-Tune Multi-Instrument Tuner V1.93 Copyright (c) 2001-2006 Joseph Heimiller[15], lo que se hizo en esta prueba fue afinar cada una de las cuerdas del violín con el prototipo de afinador digital de

instrumentos musicales y luego comparar el valor de frecuencia que se obtuvo con el afinador por software, de esta manera que tan acertado es el prototipo con respecto a un afinador comercial. Y por último la prueba Tipo D que se realizó la comparación de cada cuerda 10 veces, es decir que se afino primero la cuerda con el afinador digital, luego este valor se leyó con el afinador comercial tanto para la guitarra como para el violín esto se repitió 10 veces para poder determinar cuan acertado es el afinador digital de instrumentos con el dsPIC30F3014.

El la Tabla 6.1 se muestra la prueba Tipo A con un factor de multiplicación de 7, lo que quiere decir que el valor obtenido del cálculo de la FFT es multiplicado por 7 para tener el valor real de frecuencia, siendo la señal ingresada una señal senoidal muestreada a 4000Hz.

Tabla 6.1. Prueba Tipo A

Señal Generada (Hz)	Señal Medida dsPIC (Hz)	% Error
36	35	2.78
53	56	5.66
77	84	9.09
118	119	0.85
156	147	5.77
172	175	1.74
192	189	1.56
217	217	0.00

El resultado que obtenemos de esta prueba es que para valores de frecuencia menores a 220Hz es necesario realizar un muestreo a 4000Hz, ya que podemos tener valores de frecuencia con mayor precisión a los ingresados por el generador de ondas, que si se realizara el muestreo con otro valor.

Tabla 6.2 La Tabla 6.2 se muestra la prueba Tipo A con un factor de multiplicación de 37, lo que quiere decir que el valor obtenido del cálculo de la FFT es multiplicado por 37 para tener el valor real de frecuencia, siendo la señal ingresada una señal senoidal muestreada a 10000Hz.

Tabla 6.2. Prueba Tipo A

Señal Generada (Hz)	Señal Medida dsPIC (Hz)	% Error
158	148	6.33
185	185	0.00
217	222	2.30
256	259	1.17
323	333	3.10
370	370	0.00
400	407	1.75
482	481	0.21

El resultado que obtenemos de esta prueba es que para valores de frecuencia mayores a 220Hz es necesario realizar un muestreo a 10000Hz, ya que podemos tener valores de frecuencia con mayor precisión a los ingresados por el generador de ondas, que si se realizara el muestreo con otro valor.

Las siguientes pruebas realizadas fueron las Tipo B y C, de las cuales podemos concluir si el afinador digital es bueno para afinar diferentes instrumentos. En la Tabla 6.3 podemos ver una comparación de la frecuencia definida para una guitarra acústica por la escala temperada y el valor que podemos obtener con el afinador digital de instrumentos.

Tabla 6.3. Valor experimental medido vs Valor deseado

Nota	Frecuencia Original (Hz)	dsPIC (Hz)	% Error
MI	330	333	0.90
LA	247	259	4.85
SOL	196	196	0
RE	147	147	0
SI	110	112	1.81
MI	82	84	2.43

Esta comparación nos da una idea de cuan cercano a la realidad puede afinar el prototipo diseñado, por las limitaciones que tiene el dsPIC podemos ver que para ciertas

cuerdas de la guitarra existe una mínima variación del valor de frecuencia que se alcanza.

En la Tabla 6.4 tenemos los resultados de la prueba Tipo B, como se puede apreciar en los resultados obtenidos existe cierta variación de lo que afina con el prototipo y lo que se midió en el afinador por software.

Tabla 6.4. Prueba Tipo B (Guitarra Acústica)

Nota	dsPIC (Hz)	Afinador por Software (Hz)	% Error
MI	333	332	0.20
LA	259	245	5.40
SOL	196	195	0.51
RE	147	147	0
SI	112	108	5.57
MI	84	83	0

Las siguientes tablas muestran los resultados de la prueba Tipo D, que nos permite determinar si el prototipo diseñado es lo suficientemente bueno para afinar un instrumento, que instrumentos son los que mejor se puede afinar y que cuerdas quedan mejor afinadas.

Tabla 6.5. Prueba Tipo D

	Nota	dsPIC (Hz)	Afinador por Software (Hz)	% Error
1	MI	333	330	0.90
2	MI	333	330	0.90
3	MI	333	329	1.20
4	MI	333	328	1.50
5	MI	333	331	0.60
6	MI	333	332	0.20
7	MI	333	330	0.90
8	MI	333	328	1.50
9	MI	333	330	0.90
10	MI	333	331	0.60
Total				0.93

Tabla 6.6. Prueba Tipo D

	Nota	dsPIC (Hz)	Afinador por Software (Hz)	% Error
1	LA	112	110	1.79
2	LA	112	111	0.89
3	LA	112	110	1.79
4	LA	112	110	1.79
5	LA	112	109	2.68
6	LA	112	111	0.89
7	LA	112	112	0.00
8	LA	112	112	0.00
9	LA	112	113	0.89
10	LA	112	110	1.79
Total				1.07

Tabla 6.7. Prueba Tipo D

	Nota	dsPIC (Hz)	Afinador por Software (Hz)	% Error
1	SOL	196	196	0.00
2	SOL	196	196	0.00
3	SOL	196	197	0.51
4	SOL	196	198	1.02
5	SOL	196	197	0.51
6	SOL	196	197	0.51
7	SOL	196	198	1.02
8	SOL	196	198	1.02
9	SOL	196	199	1.53
10	SOL	196	196	0.00
Total				0.61

Tabla 6.8. Prueba Tipo D

	Nota	dsPIC (Hz)	Afinador por Software (Hz)	% Error
1	RE	147	147	0.00
2	RE	147	147	0.00
3	RE	147	148	0.68
4	RE	147	147	0.00
5	RE	147	145	1.36
6	RE	147	145	1.36
7	RE	147	145	1.36
8	RE	147	147	0.00
9	RE	147	147	0.00
10	RE	147	148	0.68
Total				0.27

Tabla 6.9. Prueba Tipo D

	Nota	dsPIC (Hz)	Afinador por Software (Hz)	% Error
1	SI	259	258	0.39
2	SI	259	259	0.00
3	SI	259	258	0.39
4	SI	259	257	0.77
5	SI	259	255	1.54
6	SI	259	257	0.77
7	SI	259	255	1.54
8	SI	259	258	0.39
9	SI	259	259	0.00
10	SI	259	258	0.39
Total				0.62

Tabla 6.10. Prueba Tipo D

	Nota	dsPIC (Hz)	Afinador por Software (Hz)	% Error
1	MI	84	83	1.19
2	MI	84	83	1.19
3	MI	84	85	1.19
4	MI	84	85	1.19
5	MI	84	83	1.19
6	MI	84	82	2.38
7	MI	84	85	1.19
8	MI	84	83	1.19
9	MI	84	83	1.19
10	MI	84	84	0.00
Total				0.48

De los resultados de las pruebas anteriores, podemos decir que para la nota LA se comete el mayor error con un 1.07% es decir que podemos obtener una afinación con 3Hz mas de lo deseado en el peor caso.

En la Tabla 6.11 podemos ver una comparación de la frecuencia definida para un violín por la escala temperada y el valor que podemos obtener con el afinador digital de instrumentos.

Tabla 6.11. Valor medido experimental vs Valor deseado

Nota	Frecuencia Original (Hz)	dsPIC (Hz)	% Error
SOL	98	98	0
RE	147	147	0
LA	220	222	0.90
MI	330	333	0.90

En la Tabla 6.12 tenemos los resultados de la prueba Tipo C, como se puede apreciar en los resultados obtenidos existe cierta variación de lo que afina con el prototipo y lo que se midió en el afinador por software.

Tabla 6.12. Prueba Tipo C

Nota	dsPIC (Hz)	Afinador por Software (Hz)	% Error
SOL	98	96	2.04
RE	147	148	0.68
LA	222	220	0.90
MI	333	329	1.20

Las siguientes tablas muestran los resultados de la prueba Tipo D, que nos permite determinar si el prototipo diseñado es lo suficientemente bueno para afinar un instrumento, que instrumentos son los que mejor se puede afinar y que cuerdas quedan mejor afinadas.

Tabla 6.13. Prueba Tipo D

	Nota	dsPIC (Hz)	Afinador por Software (Hz)	% Error
1	SOL	98	98	0.00
2	SOL	98	97	1.02
3	SOL	98	100	2.04
4	SOL	98	99	1.02
5	SOL	98	96	2.04
6	SOL	98	98	0.00
7	SOL	98	97	1.02
8	SOL	98	98	0.00
9	SOL	98	99	1.02
10	SOL	98	98	0.00

Total	0.81
-------	------

Tabla 6.14. Prueba Tipo D

	Nota	dsPIC (Hz)	Afinador por Software (Hz)	% Error
1	RE	147	146	0.68
2	RE	147	145	1.36
3	RE	147	148	0.68
4	RE	147	147	0.00
5	RE	147	147	0.00
6	RE	147	146	0.68
7	RE	147	145	1.36
8	RE	147	145	1.36
9	RE	147	146	0.68
10	RE	147	148	0.68
Total				0.48

Tabla 6.15. Prueba Tipo D

	Nota	dsPIC (Hz)	Afinador por Software (Hz)	% Error
1	LA	222	219	1.35
2	LA	222	220	0.90
3	LA	222	220	0.90
4	LA	222	219	1.35
5	LA	222	221	0.45
6	LA	222	222	0.00
7	LA	222	220	0.90
8	LA	222	220	0.90
9	LA	222	221	0.45
10	LA	222	219	1.35
Total				0.86

Tabla 6.16. Prueba Tipo D

	Nota	dsPIC (Hz)	Afinador por Software (Hz)	% Error
1	MI	333	330	0.90
2	MI	333	330	0.90
3	MI	333	329	1.20
4	MI	333	331	0.60
5	MI	333	328	1.50
6	MI	333	329	1.20
7	MI	333	331	0.60
8	MI	333	328	1.50
9	MI	333	330	0.90
10	MI	333	331	0.60
Total				0.99

De los resultados de las pruebas anteriores, podemos decir que para la nota MI se comete el mayor error con un 0.99% es decir que podemos obtener una afinación con 3Hz menos de lo deseado en el peor caso de acuerdo con la tabla establecida en el programa.

Después de haber realizado estas pruebas podemos decir que el afinador digital de instrumentos con el dsPIC30F3014 cumple con el objetivo planteado, que es afinar cada cuerda de un instrumento musical, pero debemos tomar en cuenta que para los diferentes instrumentos que podemos afinar existen ciertas variantes, es decir que dependiendo del instrumento y de la cuerda que se afina se tendrá un mayor error en la afinación, esto lo podemos ver claramente en el caso de afinar una guitarra acústica, en la afinación de la cuerda SI, en esta cuerda se comete el mayor error, ya que la diferencia de frecuencias entre la escala temperada y lo que obtenemos en el dsPIC es de 12Hz, aunque esta valor es el de mayor diferencia podemos decir que no afecta en nada a la afinación, ya que 10Hz o 12Hz es casi imperceptible al oído humano, por lo que podemos concluir que la afinación es adecuada.

CAPITULO 7

CONCLUSIONES Y RECOMENDACIONES

A continuación se presenta las conclusiones y recomendaciones a las que se ha llegado luego de realizar el diseño y las pruebas del prototipo de afinador de instrumentos musicales con el dsPIC30F3014.

7.1. Conclusiones

- Durante el proceso diseño surgieron ciertos inconvenientes tanto con el ingreso de la señal de audio que se ingresa como con la frecuencia de muestreo y la presentación de la frecuencia en el LCD. Debido a que la señal de audio es muy baja en voltaje fue necesario diseñar un amplificador de voltaje, para el caso de la frecuencia de muestreo fue necesario hacer dos rutinas diferentes para la conversión A/D. Por último para la presentación de la frecuencia al igual que la conversión A/D fue necesario programar dos rutinas diferentes para cada frecuencia de muestreo establecida.
- Generalmente para lograr la afinación de un instrumento de cuerda se lo hace solo con el oído y la capacidad que la persona tenga para diferenciar entre una y otra nota, por lo que un músico experimentado lo realiza sin ningún problema, por otro lado para alguien que esta empezando en la música es más complicado afinar el instrumento por lo que un afinador digital es muy útil en este caso, por lo que este prototipo es de fácil manejo para el usuario al igual que para un músico experimentado.

- El algoritmo escogido para transformar la señal del dominio en el tiempo a dominio en la frecuencia es una FFT que se aplica cuando el número de muestras de la señal es una potencia de dos, para nuestro caso tenemos 256 muestras que es igual a 2^7 , lo cual es suficiente para lograr el objetivo del trabajo.
- Para el ingreso de la señal al dsPIC es necesario primero realizar una etapa de amplificación de la señal del micrófono para que esta al menos alcance un voltaje entre 1V y 5V.
- Durante el desarrollo del programa se presentaron algunos inconvenientes en la medición de la frecuencia ingresada, los cuales se pudieron superar con el uso de dos diferentes rutinas tanto para la adquisición de datos como para la escritura del resultado de la FFT. Una rutina que opera cuando la frecuencia a medir es menor de 220 Hz y otra que opera para medir entre 221 Hz y 700 Hz.
- Para asegurarnos de que la señal que ingresa al dsPIC es correcta fue necesario incluir en el programa una sentencia de comparación para verificar que los valores de voltaje deseados se encuentren entre 3.5 y 5 voltios. Para de esta manera rechazar frecuencias no deseadas.
- El prototipo está diseñado para realizar la afinación de cuatro instrumentos de cuerda que son: la guitarra, el violín, el bajo y el chelo, para cada uno de estos existe una tabla de asignación de frecuencia con la cual se debe comparar la frecuencia de la señal de entrada.
- Dependiendo tanto del instrumento como de la cuerda seleccionada, el porcentaje de error varía, esto lo podemos observar en la cuerda SI de la guitarra donde tenemos una diferencia de 12Hz más de lo que debería tener de acuerdo a la escala temperada.

7.2. Recomendaciones

- Para mejorar la calidad de la señal de entrada sería recomendable diseñar un circuito con el que se pueda obtener una señal de audio más pura, es decir libre de ruido por lo que hacer uso de un micrófono adherible para colocarlo en la caja del instrumento sería una buena opción.
- Debido a la poca capacidad de memoria del dsPIC30F3014 solo se puede realizar una FFT de máximo 256 puntos, esto quiere decir que la resolución que se puede obtener con este integrado sobre todo para señal con alta frecuencia es muy baja, por lo que se recomienda utilizar un integrado con mayor capacidad como por ejemplo los de la familia dsPIC30F60xx o dsPIC33Fxxxx de microchip, ya que con mayor espacio de memoria disponible es posible tomar un número mayor de muestra de la señal análoga lo que nos da una mejor resolución en el resultado del cálculo de la frecuencia.
- El momento de la afinación se recomienda realizar el procedimiento en un ambiente libre de ruido para evitar cualquier lectura errónea por parte del afinador.
- El dsPIC30F3014 es de fácil programación, y puede ser usado en muchas aplicaciones, pero se recomienda que para el cálculo de la FFT de una señal, utilizar un integrado con mayor capacidad de memoria, ya que mejora muchísimo la resolución en cuanto a la lectura de la frecuencia de una señal analógica.
- Antes de realizar cualquier proyecto con el dsPIC30F3014 es mejor que se investigue a fondo las capacidades del integrado y si cumple con los requerimientos que se necesite, es también importante que consiga un simulador para el integrado, ya que por ser este relativamente nuevo aun no existe un simulador con todas las funciones básicas.

REFERENCIAS BIBLIOGRAFICAS

- [1] http://www.xtec.es/centres/a8019411/caixa/musica_es.htm#so, sonido, publicado en 2002, consultado en septiembre 2006
- [2] <http://www.fortunecity.com/tinpan/lennon/193/elsonido.htm#Altura>, altura del sonido, publicado en 2003, consultado en septiembre2006
- [3] <http://www.video-computer.com/Propiedades%20del%20sonido.htm>, propiedades del sonido, publicado en 2001, consultado en septiembre 2006
- [4] <http://www.xtec.es/centres/a8019411/caixa/escalas.htm#origen>, escalas musicales, publicado en 2002, consultado en septiembre 2006
- [5] <http://www.FichasTeoríadelamúsicaSolfeoSonidosarmónicosHagaseLaMúsica.com.htm>, sonidos armónicos, publicado en 2001, consultado en 2006
- [6] <http://www.wikipedia.com>, altura y escalas, publicado en 2000, consultado en 2006
- [7] <http://www.microchip.com>, Datasheet del microcontrolador dsPIC30F3014, publicado en 2004, consultado en septiembre 2006
- [8] <http://www.mikroe.com>, mikoBasic, publicado en 2004, consultado en abril 2007
- [9] <http://www.wikipedia.com> “Introductory Digital Signal Processing with Computer Applications” Autores: Lynn and Fuerst
- [10] Medida de armónicos en Clase I para redes eléctricas de frecuencia variable. pdf publicado en 2004, Gaston Schelotto Dowek, consultado en agosto 2007

- [11] <http://www.microchip.com>, Advanced Signal Processing with the dsPIC30F 2004 Microchip Technology Incorporated, publicado en 2004, consultado en Julio 2007
- [12] <http://library.thinkquest.org/C0120343/Espanol/Paginaprincipal.htm>, pulso y compas, publicado en 2002, consultado en septiembre 2006
- [13] <http://www.mikroe.com>, MikroBasic for dsPIC30F/33 and PIC24 User's Manual, mikroelektronika 2003 – 2006, consultado en abril 2007
- [14] <http://www.oobox.com>, OOBOX Music Software, publicado en octubre 2005, consultado en septiembre 2006
- [15] <http://www.musicmasterworks.com>, In- Tune Software, publicado en 2005, consultado en septiembre de 2006

ANEXOS

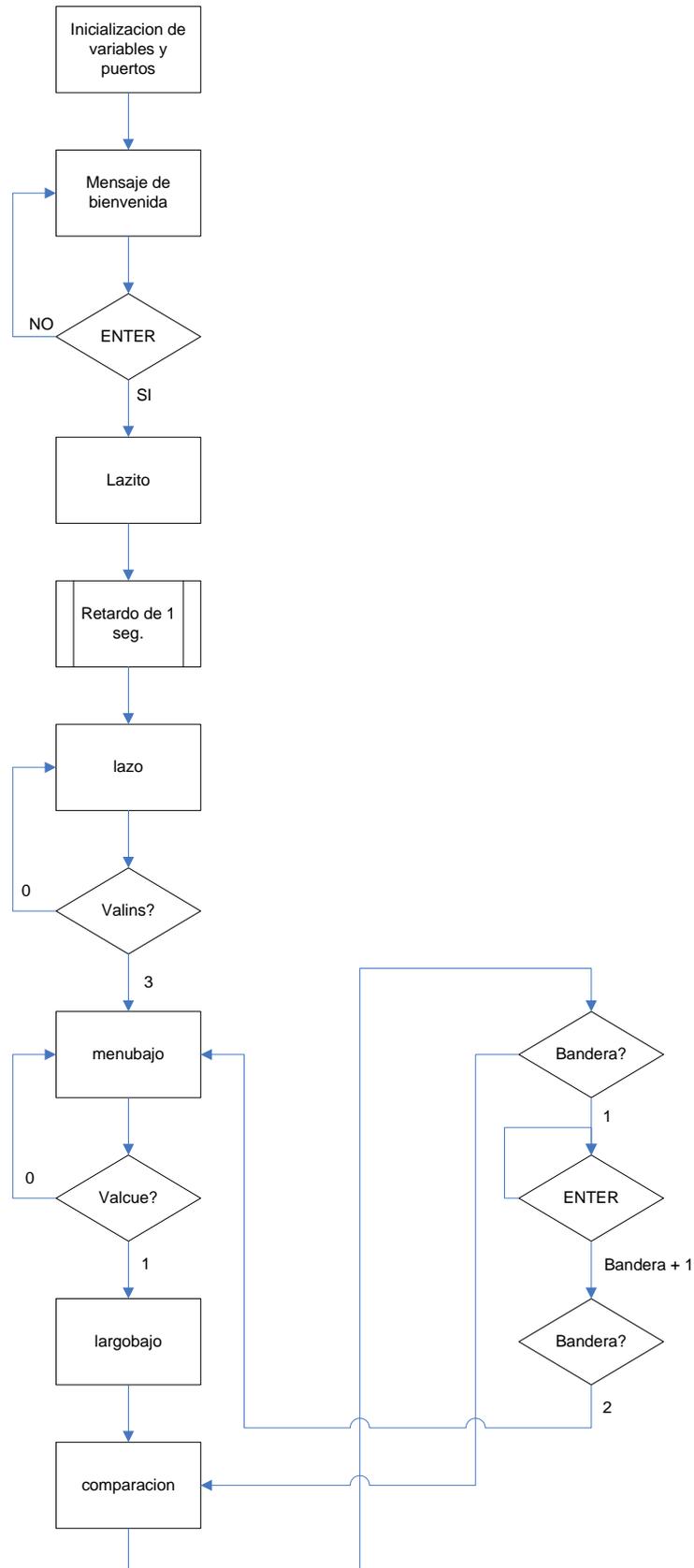
DIAGRAMA DE FLUJO SELECCION INSTRUMENTO (BAJO)

DIAGRAMA DE FLUJO SELECCION INSTRUMENTO (GUITARRA)

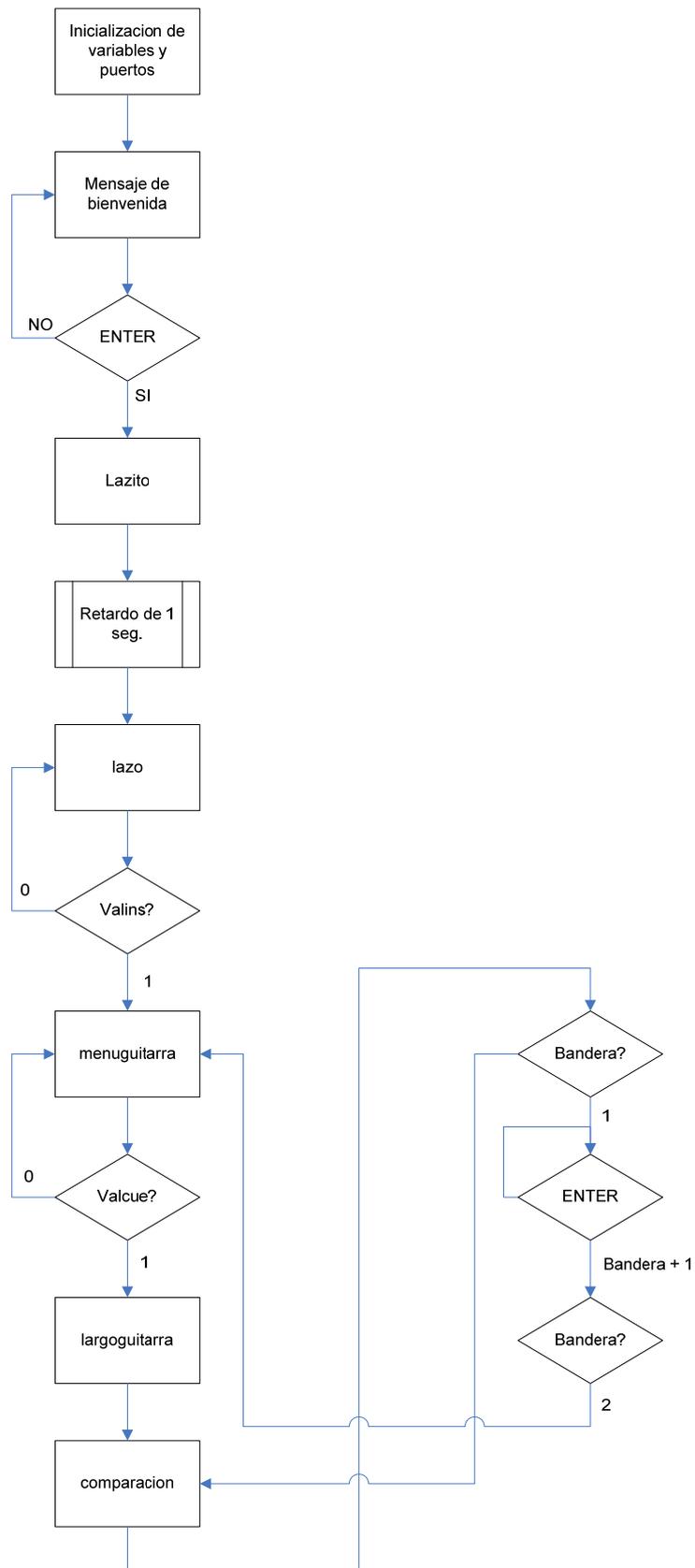


DIAGRAMA DE FLUJO SELECCION INSTRUMENTO (VIOLÍN)

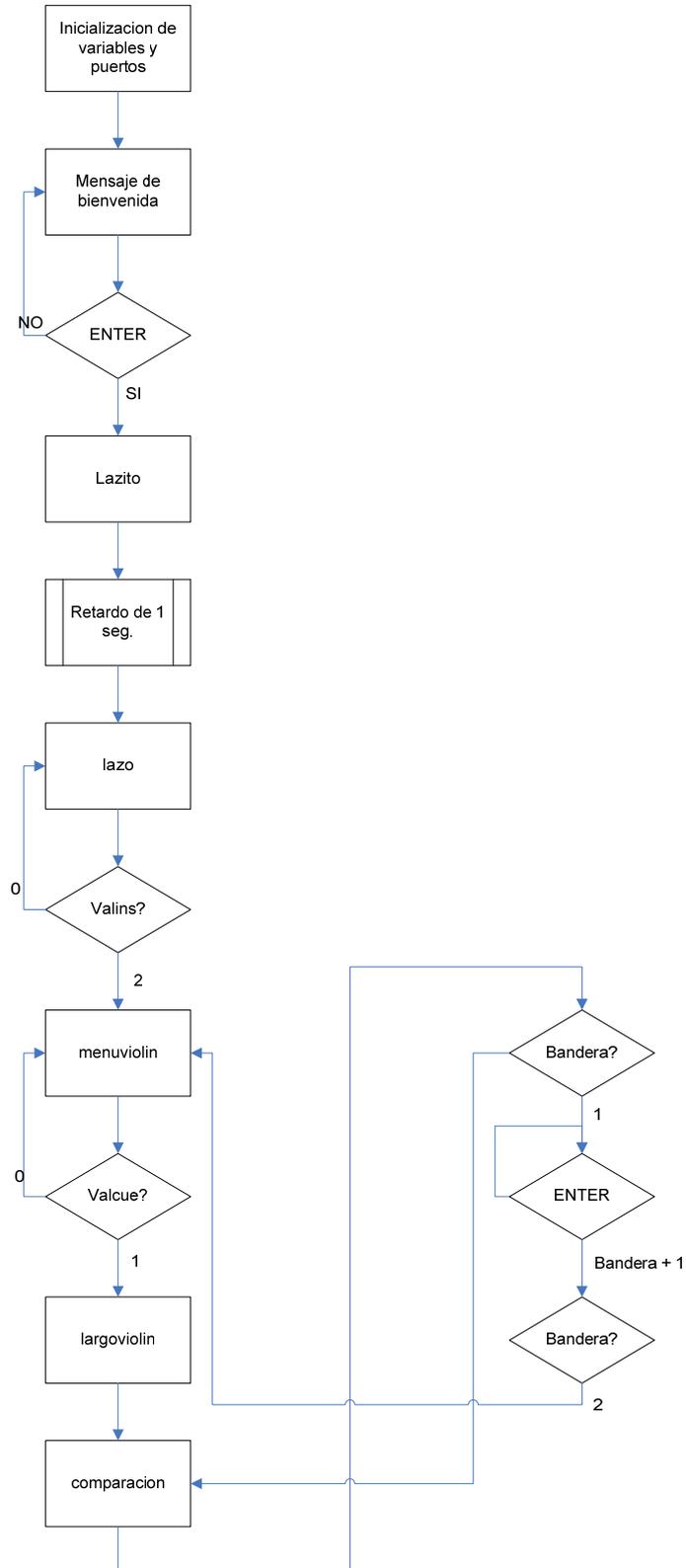


DIAGRAMA DE FLUJO SELECCION INSTRUMENTO (CHELO)

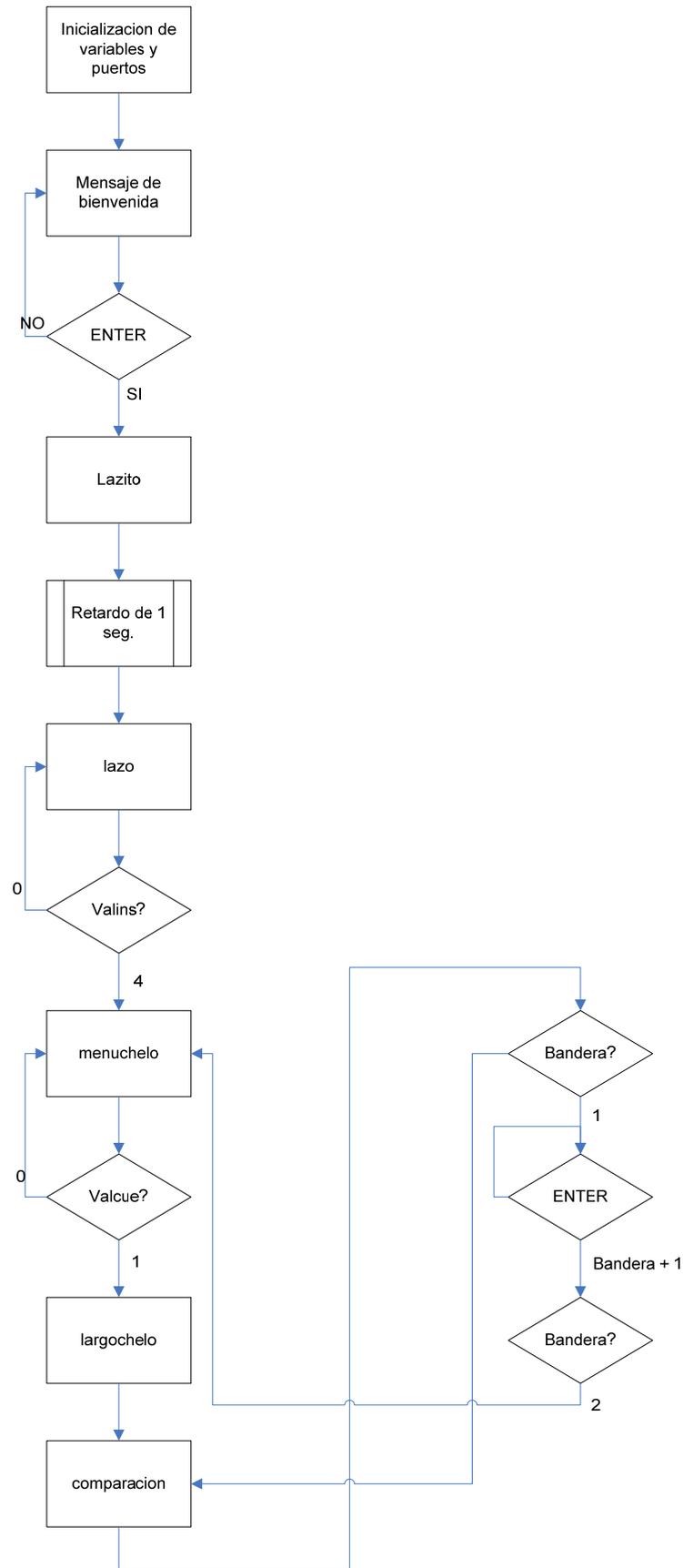


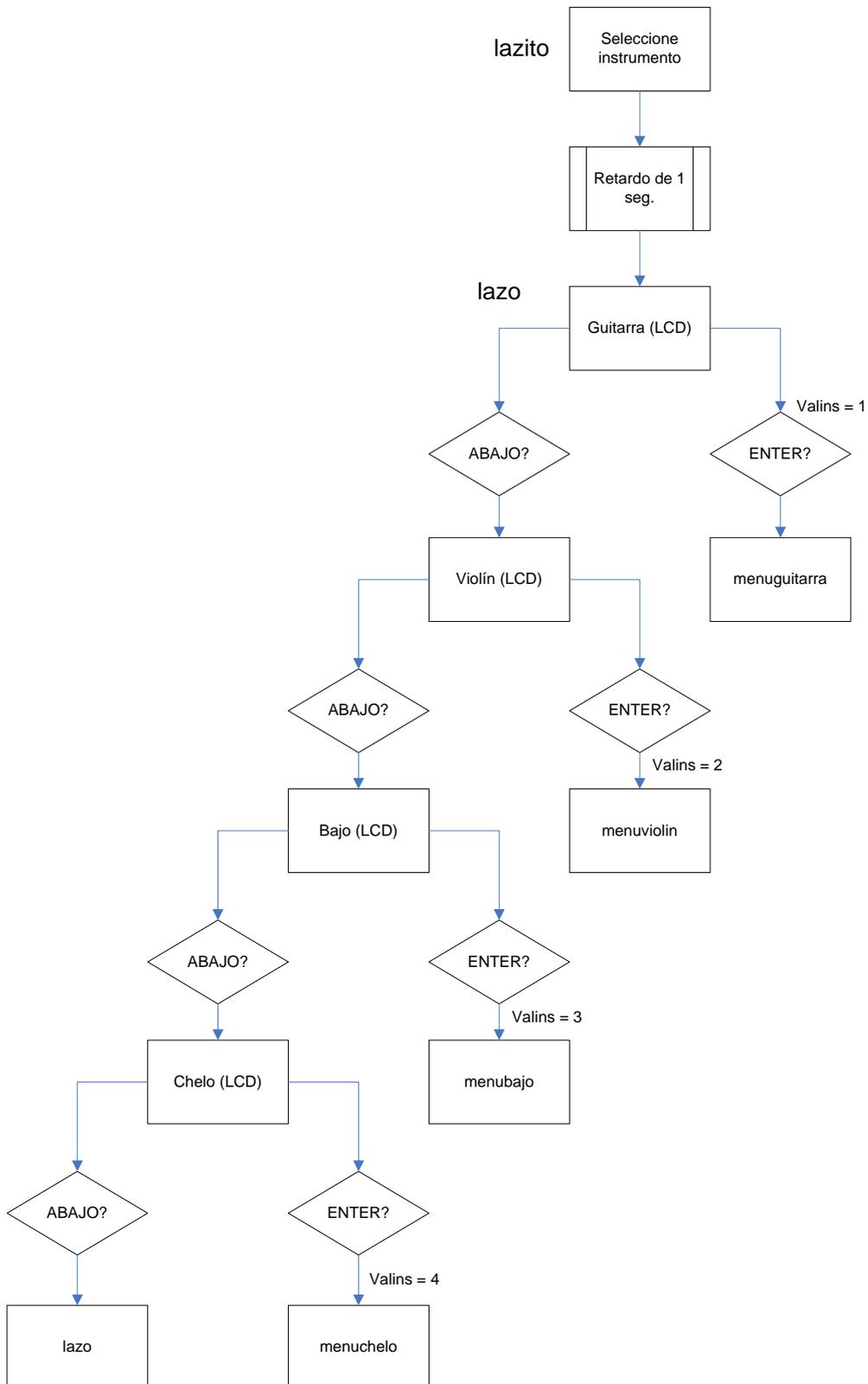
DIAGRAMA DE FLUJO SELECCION INSTRUMENTO

DIAGRAMA DE FLUJO SELECCION CUERDAS (GUITARRA)

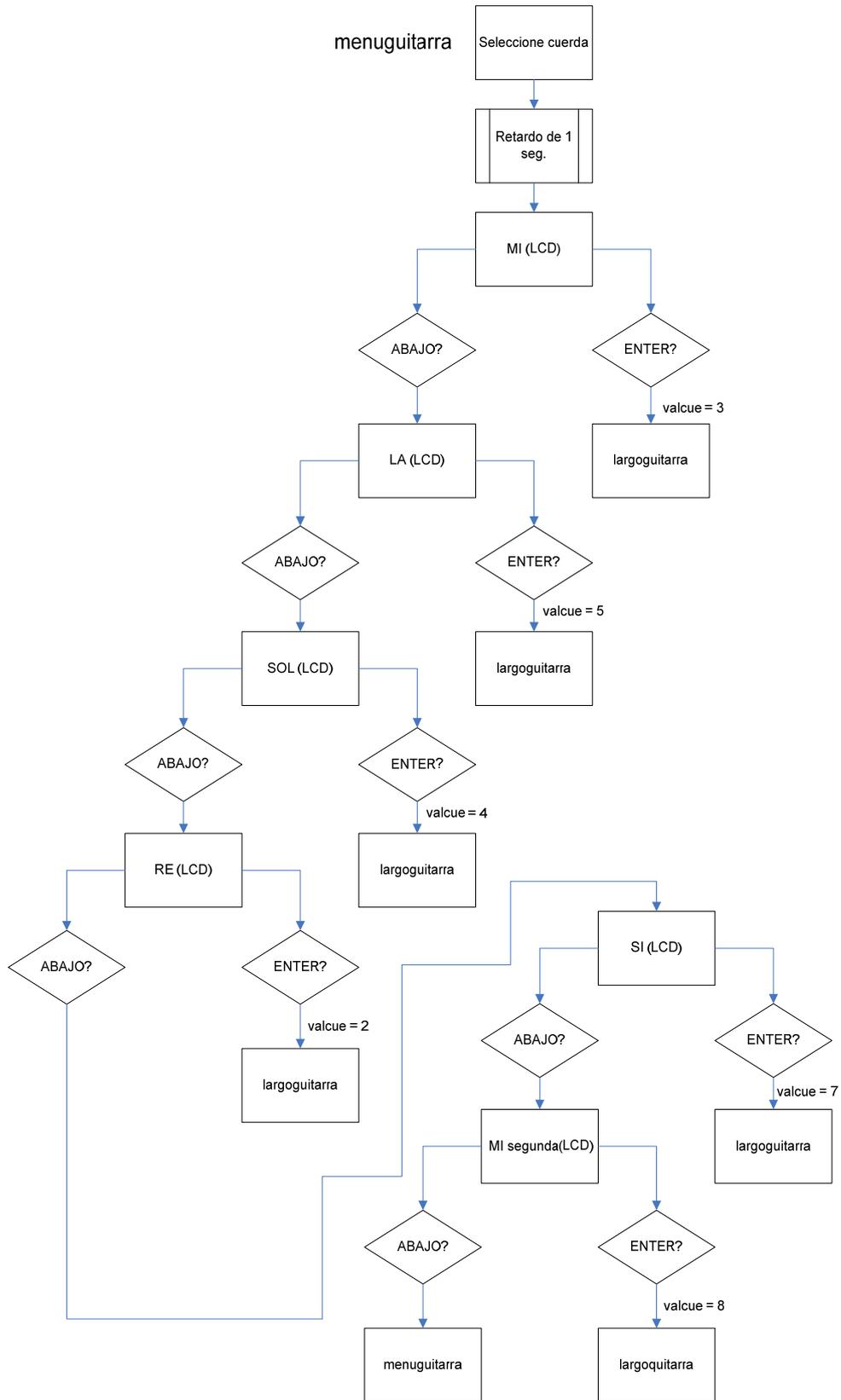


DIAGRAMA DE FLUJO SELECCION CUERDAS (VIOLÍN)

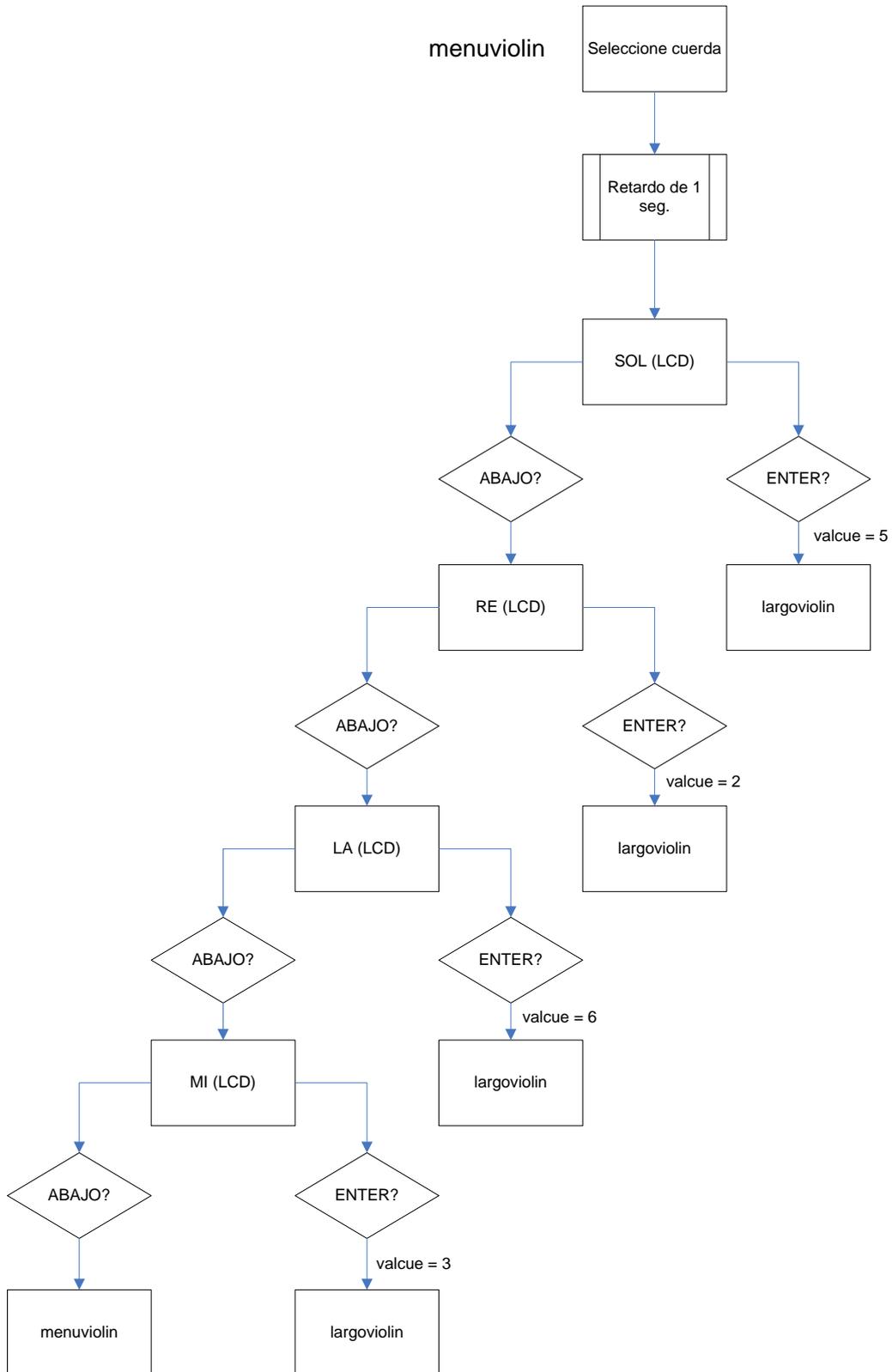


DIAGRAMA DE FLUJO SELECCION CUERDAS (BAJO)

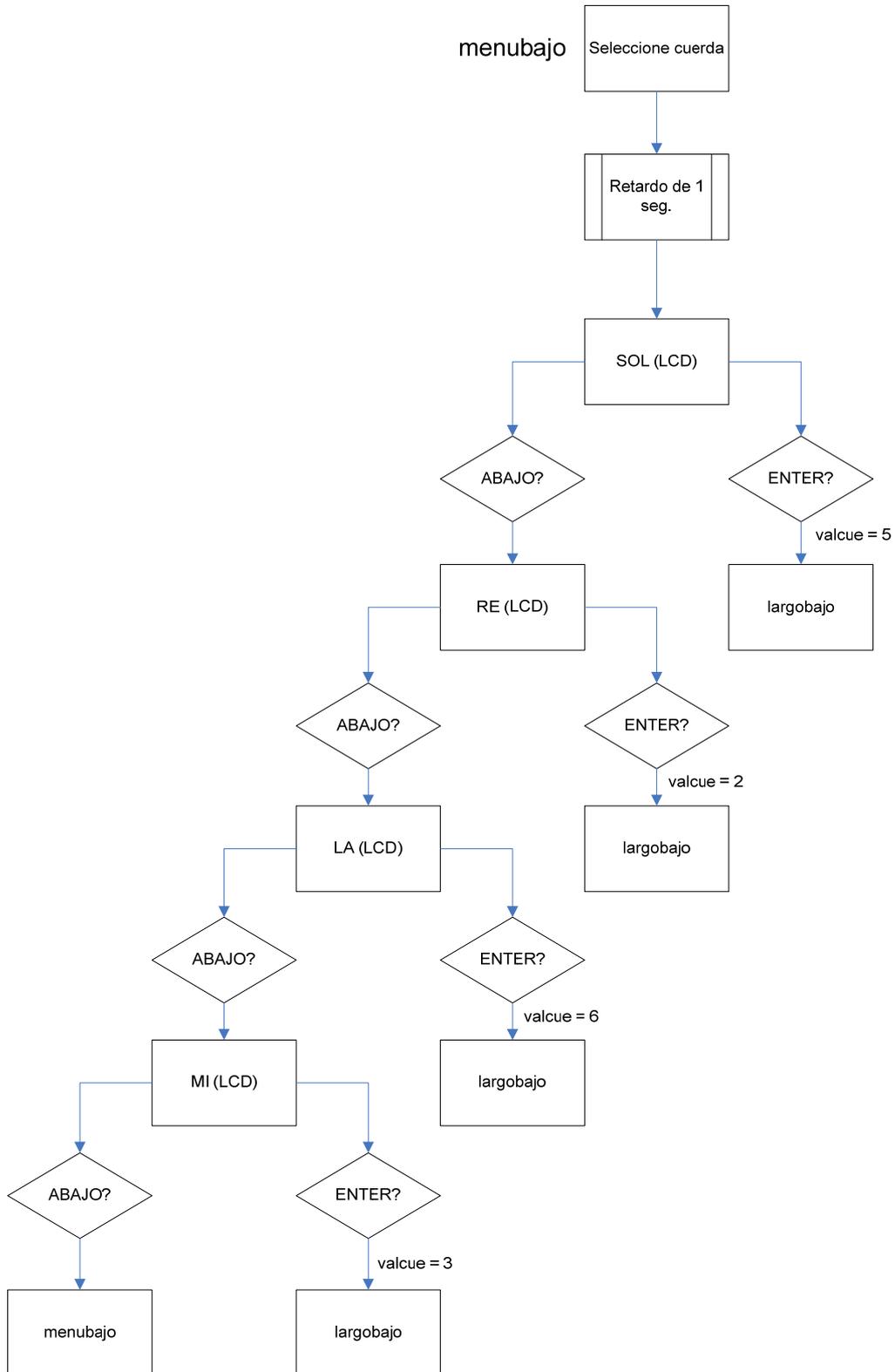
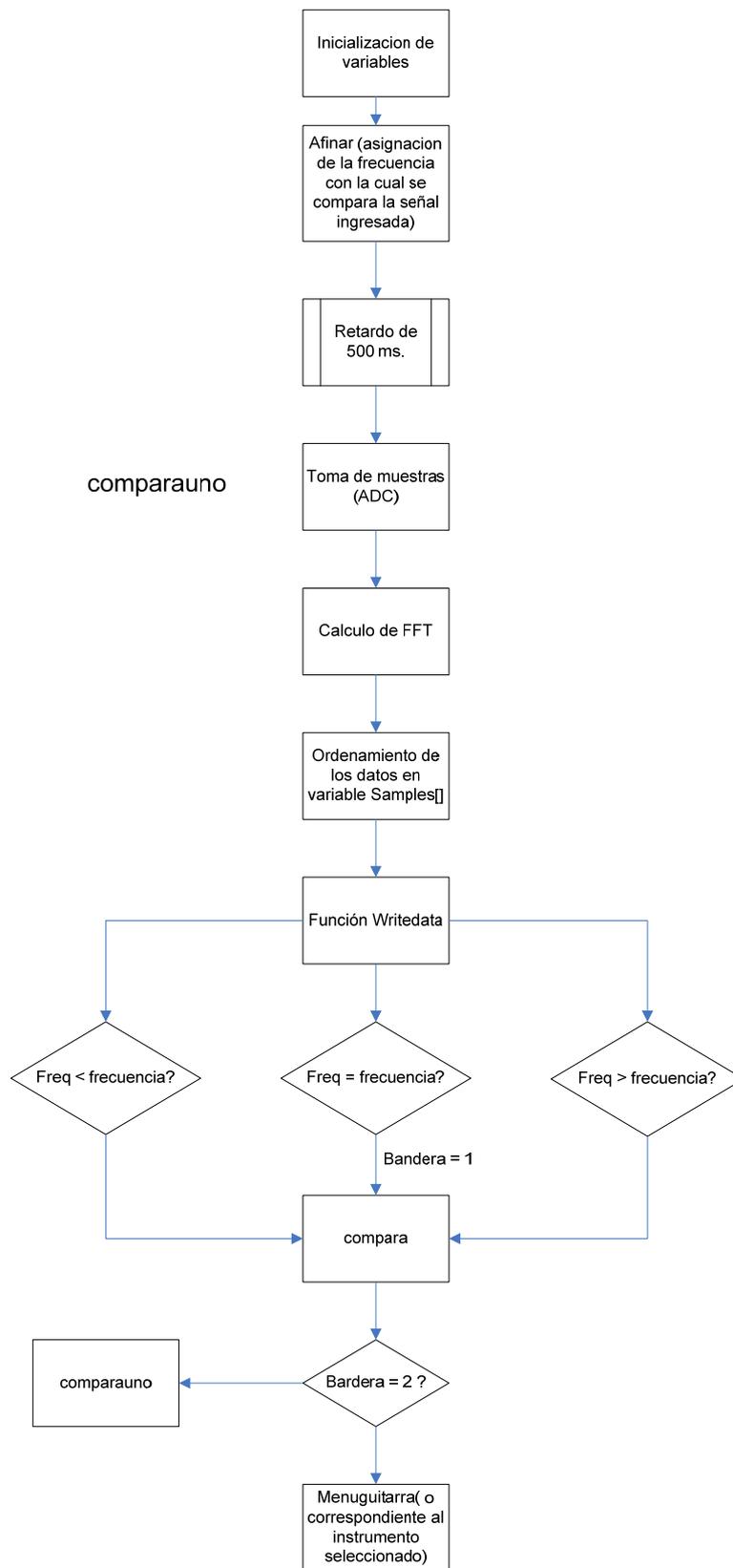


DIAGRAMA DE FLUJO PARA COMPARACION DE FRECUENCIA

PROGRAMA COMPLETO PARA EL AFINADOR DIGITAL DE INSTRUMENTOS MUSICALES CON dsPIC30F3014

```

program TODOLCD

'Variables globales
dim cuerda as integer
dim i as integer
dim j as integer
dim cont as integer
dim instru as integer
dim aux3 as integer
dim aux4 as integer
dim menu as integer
dim valins as integer
dim valcue as integer
dim bandera as integer
dim frecuencia as integer
dim valor as word
dim Samples as word[256] absolute $0C00 'Espacio de memoria Y, usado para calculo
                                     'FFT

    freq  as word
    txt   as string[5]

'Escritura en el LCD
sub procedure escribir (dim x, y as integer, dim byref texto as string[20])
    Lcd_Cmd (LCD_FIRST_ROW)
    Lcd_Cmd (LCD_CURSOR_OFF)
    Lcd_Cmd (LCD_SECOND_ROW)
    Lcd_Out (x, y, texto)
end sub

'Lectura del puerto 2D (boton de control)
sub function leerpe as integer
    dim aux as integer
    aux = 0
    while aux = 0
        if Button(PORTD, 2, 50, 1) = 1 then
            aux = 1
            result = aux
        end if
    wend
end sub

'Funcion para asignación de frecuencia de la cuerda deseada
sub function afinar (dim uno, dos as integer) as integer

```

```
uno = valins  
dos = valcue
```

```
'Cuerdas para Guitarra
```

```
  if ((uno = 1) and (dos = 3)) then  
    result = 333  
  end if  
  if ((uno = 1) and (dos = 6)) then  
    result = 112  
  end if  
  if ((uno = 1) and (dos = 5)) then  
    result = 196  
  end if  
  if ((uno = 1) and (dos = 2)) then  
    result = 147  
  end if  
  if ((uno = 1) and (dos = 7)) then  
    result = 259  
  end if  
  if ((uno = 1) and (dos = 8)) then  
    result = 84  
  end if
```

```
'Cuerdas para Violín
```

```
  if ((uno = 2) and (dos = 2)) then  
    result = 147  
  end if  
  if ((uno = 2) and (dos = 3)) then  
    result = 333  
  end if  
  if ((uno = 2) and (dos = 5)) then  
    result = 98  
  end if  
  if ((uno = 2) and (dos = 6)) then  
    result = 222  
  end if
```

```
'Cuerdas para Bajo
```

```
  if ((uno = 3) and (dos = 2)) then  
    result = 77  
  end if  
  if ((uno = 3) and (dos = 3)) then  
    result = 42  
  end if  
  if ((uno = 3) and (dos = 5)) then  
    result = 98  
  end if  
  if ((uno = 3) and (dos = 6)) then  
    result = 56
```

```

    end if

'Cuerdas para Chelo
  if ((uno = 4) and (dos = 1)) then
    result = 147
  end if
  if ((uno = 4) and (dos = 2)) then
    result = 77
  end if
  if ((uno = 4) and (dos = 5)) then
    result = 49
  end if
  if ((uno = 4) and (dos = 6)) then
    result = 222
  end if
end sub

'Inicialización ADC
sub procedure InitAdc

  ADPCFG = 0x00FF
  ADCHS = 8
  ADCSSL = 0
  ADCON3 = $1F3F ' sample time = 31 Tad.
  ADCON2 = 0
  ADCON1 = $83E0

  TRISB.8 = 1
end sub

'Funcion para conversión a 1.15 punto flotante
sub function Fract2Float(dim input_ as integer) as float
  if (input_ < 0) then
    input_ = - input_
  end if
  result = input_/32768.
end sub

'Escritura de el valor maximo de frecuencia
sub procedure WriteData
dim Re, Im,
  j, k, max as word
  Rer, Imr, tmpR as float

  j = 2
  k = 0
  max = 0
  freq = 0
  while k <= 31
    Re = Samples[j]

```

```

inc(j)

Im = Samples[j]
inc(j)

Rer = Fract2Float(Re)
Imr = Fract2Float(Im)

tmpR = Rer * Rer
Rer = tmpR
tmpR = Imr * Imr
Imr = tmpR
tmpR = sqrt(Rer + Imr)
Rer = tmpR*256.
Re = Rer

if Re > max then
    max = Re
    freq = k
end if

inc(k)

wend

if ((valins = 1) and (valcue = 3)) or ((valins = 1) and (valcue = 7)) or ((valins = 2) and (valcue
= 6)) or ((valins = 2) and (valcue = 3)) or ((valins = 4) and (valcue = 6)) then
    freq = freq * 37
    Lcd_Cmd(LCD_CLEAR)
    escribir(1,1,"frecuencia:")
    WordToStr(freq, txt)
    escribir(2,1,txt)
    Delay_ms(1000)
else
    freq = freq * 7
    Lcd_Cmd(LCD_CLEAR)
    escribir(1,1,"frecuencia:")
    WordToStr(freq, txt)
    escribir(2,1,txt)
    Delay_ms(1000)
end if

end sub

'Conversion ADC
sub function ReadAdc as word

ADCON1.1 = 1
while ADCON1.0=0
    nop

```

```

wend
result = ADCBUFO

end sub

'Funcion de llenado de la variable Samples[]
sub procedure SampleInput

dim i as integer
i = 0
if ((valins = 1) and (valcue = 3)) or ((valins = 1) and (valcue = 7)) or ((valins = 2) and (valcue = 6)) or ((valins = 2) and (valcue = 3)) or ((valins = 4) and (valcue = 6)) then
while i <= 255
valor = ReadAdc
if (valor >= 20608) and (valor <= 32752) then
Samples[i] = valor 'muestras del ADC
inc(i)
Samples[i] = 0
inc(i)
' Delay_us(238)
end if
wend
else
while i <= 255
valor = ReadAdc
if (valor >= 20608) and (valor <= 32752) then
Samples[i] = valor 'muestras del ADC
inc(i)
Samples[i] = 0
inc(i)
Delay_us(238)
end if
wend
end if
end sub

'Funcion de comparación
sub procedure comparacion (dim a, b as integer)
dim ins as integer
dim cue as integer
InitAdc
ins = a
cue = b
frecuencia = afinar(a, b)

comparauno:
Delay_ms (1000)
Lcd_Cmd(LCD_CLEAR)
escribir(1,1,"ya")
SampleInput

```

```
    FFT(7, @TwiddleCoeff_128, Samples)

    BitReverseComplex(7, Samples)

    WriteData

if freq < frecuencia then
    Lcd_Cmd (LCD_CLEAR)
    escribir(1, 1, "Ajuste cuerda")
    goto compara
    Delay_ms(500)
end if

if freq > frecuencia then
    Lcd_Cmd (LCD_CLEAR)
    escribir(1, 1, "Afloje cuerda")
    Delay_ms(500)
    goto compara
end if

if freq = frecuencia then
    Lcd_Cmd (LCD_CLEAR)
    escribir(1, 1, "Afinacion")
    escribir(2, 1, "correcta")
    Delay_ms(2000)
    bandera = 1
    Delay_ms(500)
    goto compara
end if

compara:
    while bandera = 1
        if Button(PORTD, 2, 50, 1) = 1 then
            bandera = 2
            exit
        end if
    wend
    goto comparauno
end sub

'Programa principal
main:
    cont = 0
    menu = 0
    cuerda = 0
    instru = 0
    valins = 0
```

```
valcue = 0
bandera = 0
frecuencia = 0
TRISF = 0
TRISD = $000F
```

```
Lcd_Init (PORTF, 6, 5, 4, 3, PORTF, 0, 1, 2)
```

```
escribir (1, 1, "Bienvenido")
escribir (2, 1, "presione enter")
```

```
cont = cont + leerpe
```

lazo:

```
Lcd_Cmd (LCD_CLEAR)
```

```
escribir (1, 1, "seleccione")
escribir (2, 1, "instrumento")
Delay_ms(1000)
aux3=0
```

lazo:

```
Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "Guitarra")
while (aux3=0)
  if PORTD.2=1 then
    aux3=1
  end if
  if PORTD.3=1 then
    aux3=1
  end if
wend
aux3=0
if PORTD.2 = 1 then
  valins = 1
  goto menuguitarra
end if
while ((PORTD.2=1) or (PORTD.3=1))
  NOP
wend
Delay_ms(500)
```

```
Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "Violin")
while (aux3=0)
  if PORTD.2=1 then
    aux3=1
  end if
  if PORTD.3=1 then
    aux3=1
  end if
```

```
wend
aux3=0
if PORTD.2 = 1 then
  valins = 2
  goto menuviolin
end if
while ((PORTD.2=1) or (PORTD.3=1))
  NOP
wend
Delay_ms(500)
```

```
Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "Bajo")
while (aux3=0)
  if PORTD.2=1 then
    aux3=1
  end if
  if PORTD.3=1 then
    aux3=1
  end if
wend
aux3=0
if PORTD.2 = 1 then
  valins = 3
  goto menubajo
end if
while ((PORTD.2=1) or (PORTD.3=1))
  NOP
wend
Delay_ms(500)
```

```
Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "Chelo")
while (aux3=0)
  if PORTD.2=1 then
    aux3=1
  end if
  if PORTD.3=1 then
    aux3=1
  end if
wend
aux3=0
if PORTD.2 = 1 then
  valins = 4
  goto menuchelo
end if
while ((PORTD.2=1) or (PORTD.3=1))
  NOP
wend
Delay_ms(500)
```

goto lazo

menuguitarra:

```
delay_ms(500)
if valins <> 0 then
  Lcd_Cmd (LCD_CLEAR)
  escribir (1, 3, "escoja cuerda")
  Delay_ms(1000)
  aux4 = 0

  Lcd_Cmd (LCD_CLEAR)
  escribir (1, 2, "MI")
  while (aux4=0)
    if PORTD.2=1 then
      aux4=1
    end if
    if PORTD.3=1 then
      aux4=1
    end if
  wend
  aux4=0
  if PORTD.2 = 1 then
    valcue = 3
    goto largoguitarra
  end if
  while ((PORTD.2=1) or (PORTD.3=1))
    NOP
  wend
  Delay_ms(500)

  Lcd_Cmd (LCD_CLEAR)
  escribir (1, 2, "LA")
  while (aux4=0)
    if PORTD.2=1 then
      aux4=1
    end if
    if PORTD.3=1 then
      aux4=1
    end if
  wend
  aux4=0
  if PORTD.2 = 1 then
    valcue = 6
    goto largoguitarra
  end if
  while ((PORTD.2=1) or (PORTD.3=1))
    NOP
  wend
```

```
Delay_ms(500)
```

```
Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "SOL")
while (aux4=0)
  if PORTD.2=1 then
    aux4=1
  end if
  if PORTD.3=1 then
    aux4=1
  end if
wend
aux4=0
if PORTD.2 = 1 then
  valcue = 5
  goto largoguitarra
end if
while ((PORTD.2=1) or (PORTD.3=1))
  NOP
wend
Delay_ms(500)
```

```
Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "RE")
while (aux4=0)
  if PORTD.2=1 then
    aux4=1
  end if
  if PORTD.3=1 then
    aux4=1
  end if
wend
aux4=0
if PORTD.2 = 1 then
  valcue = 2
  goto largoguitarra
end if
while ((PORTD.2=1) or (PORTD.3=1))
  NOP
wend
Delay_ms(500)
```

```
Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "SI")
while (aux4=0)
  if PORTD.2=1 then
    aux4=1
  end if
  if PORTD.3=1 then
    aux4=1
  end if
end if
```

```
        end if
    wend
    aux4=0
    if PORTD.2 = 1 then
        valcue = 7
        goto largoguitarra
    end if
    while ((PORTD.2=1) or (PORTD.3=1))
        NOP
    wend
    Delay_ms(500)

    Lcd_Cmd (LCD_CLEAR)
    escribir (1, 2, "MI segunda")
    while (aux4=0)
        if PORTD.2=1 then
            aux4=1
        end if
        if PORTD.3=1 then
            aux4=1
        end if
    wend
    aux4=0
    if PORTD.2 = 1 then
        valcue = 8
        goto largoguitarra
    end if
    while ((PORTD.2=1) or (PORTD.3=1))
        NOP
    wend
    Delay_ms(500)

    goto menuguitarra
end if
```

```
largoguitarra:
    Lcd_Cmd (LCD_CLEAR)
    escribir (1, 1, "toque la cuerda")
    Delay_ms (1000)
    comparacion (valins, valcue)
    if bandera = 2 then
        goto menuguitarra
    end if
```

menuviolin:

```
    delay_ms(500)
    if valins <> 0 then
        Lcd_Cmd (LCD_CLEAR)
        escribir (1, 3, "escoja cuerda")
```

```
Delay_ms(1000)
aux4 = 0
```

```
Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "RE")
while (aux4=0)
  if PORTD.2=1 then
    aux4=1
  end if
  if PORTD.3=1 then
    aux4=1
  end if
wend
aux4=0
if PORTD.2 = 1 then
  valcue = 2
  goto largoviolin
end if
while ((PORTD.2=1) or (PORTD.3=1))
  NOP
wend
Delay_ms(500)
```

```
Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "MI")
while (aux4=0)
  if PORTD.2=1 then
    aux4=1
  end if
  if PORTD.3=1 then
    aux4=1
  end if
wend
aux4=0
if PORTD.2 = 1 then
  valcue = 3
  goto largoviolin
end if
while ((PORTD.2=1) or (PORTD.3=1))
  NOP
wend
Delay_ms(500)
```

```
Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "SOL")
while (aux4=0)
  if PORTD.2=1 then
    aux4=1
  end if
```

```
        if PORTD.3=1 then
            aux4=1
        end if
    wend
    aux4=0
    if PORTD.2 = 1 then
        valcue = 5
        goto largoviolin
    end if
    while ((PORTD.2=1) or (PORTD.3=1))
        NOP
    wend
    Delay_ms(500)

    Lcd_Cmd (LCD_CLEAR)
    escribir (1, 2, "LA")
    while (aux4=0)
        if PORTD.2=1 then
            aux4=1
        end if
        if PORTD.3=1 then
            aux4=1
        end if
    wend
    aux4=0
    if PORTD.2 = 1 then
        valcue = 6
        goto largoviolin
    end if
    while ((PORTD.2=1) or (PORTD.3=1))
        NOP
    wend
    Delay_ms(500)

    goto menuviolin
end if

largoviolin:
Lcd_Cmd (LCD_CLEAR)
escribir (1, 1, "toque la cuerda")
Delay_ms (1000)
comparacion (valins, valcue)
if bandera = 2 then
    goto menuviolin
end if

menubajo:

delay_ms(500)
if valins <> 0 then
```

```
Lcd_Cmd (LCD_CLEAR)
escribir (1, 3, "escoja cuerda")
Delay_ms(1000)
aux4 = 0
```

```
Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "RE")
while (aux4=0)
  if PORTD.2=1 then
    aux4=1
  end if
  if PORTD.3=1 then
    aux4=1
  end if
wend
aux4=0
if PORTD.2 = 1 then
  valcue = 2
  goto largobajo
end if
while ((PORTD.2=1) or (PORTD.3=1))
  NOP
wend
Delay_ms(500)
```

```
Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "MI")
while (aux4=0)
  if PORTD.2=1 then
    aux4=1
  end if
  if PORTD.3=1 then
    aux4=1
  end if
wend
aux4=0
if PORTD.2 = 1 then
  valcue = 3
  goto largobajo
end if
while ((PORTD.2=1) or (PORTD.3=1))
  NOP
wend
Delay_ms(500)
```

```
Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "SOL")
while (aux4=0)
  if PORTD.2=1 then
```

```
        aux4=1
    end if
    if PORTD.3=1 then
        aux4=1
    end if
wend
aux4=0
if PORTD.2 = 1 then
    valcue = 5
    goto largobajo
end if
while ((PORTD.2=1) or (PORTD.3=1))
    NOP
wend
Delay_ms(500)

Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "LA")
while (aux4=0)
    if PORTD.2=1 then
        aux4=1
    end if
    if PORTD.3=1 then
        aux4=1
    end if
wend
aux4=0
if PORTD.2 = 1 then
    valcue = 6
    goto largobajo
end if
while ((PORTD.2=1) or (PORTD.3=1))
    NOP
wend
Delay_ms(500)

    goto menubajo
end if

largobajo:
Lcd_Cmd (LCD_CLEAR)
escribir (1, 1, "toque la cuerda")
Delay_ms (1000)
comparacion (valins, valcue)
if bandera = 2 then
    goto menubajo
end if

menuchelo:
```

```
delay_ms(500)
if valins <> 0 then
  Lcd_Cmd (LCD_CLEAR)
  escribir (1, 3, "escoja cuerda")
  Delay_ms(1000)
  aux4 = 0

  Lcd_Cmd (LCD_CLEAR)
  escribir (1, 2, "DO")
  while (aux4=0)
    if PORTD.2=1 then
      aux4=1
    end if
    if PORTD.3=1 then
      aux4=1
    end if
  wend
  aux4=0
  if PORTD.2 = 1 then
    valcue = 1
    goto largochelo
  end if
  while ((PORTD.2=1) or (PORTD.3=1))
    NOP
  wend
  Delay_ms(500)

  Lcd_Cmd (LCD_CLEAR)
  escribir (1, 2, "RE")
  while (aux4=0)
    if PORTD.2=1 then
      aux4=1
    end if
    if PORTD.3=1 then
      aux4=1
    end if
  wend
  aux4=0
  if PORTD.2 = 1 then
    valcue = 2
    goto largochelo
  end if
  while ((PORTD.2=1) or (PORTD.3=1))
    NOP
  wend
  Delay_ms(500)

  Lcd_Cmd (LCD_CLEAR)
  escribir (1, 2, "SOL")
```

```
while (aux4=0)
  if PORTD.2=1 then
    aux4=1
  end if
  if PORTD.3=1 then
    aux4=1
  end if
wend
aux4=0
if PORTD.2 = 1 then
  valcue = 5
  goto largochelo
end if
while ((PORTD.2=1) or (PORTD.3=1))
  NOP
wend
Delay_ms(500)

Lcd_Cmd (LCD_CLEAR)
escribir (1, 2, "LA")
while (aux4=0)
  if PORTD.2=1 then
    aux4=1
  end if
  if PORTD.3=1 then
    aux4=1
  end if
wend
aux4=0
if PORTD.2 = 1 then
  valcue = 6
  goto largochelo
end if
while ((PORTD.2=1) or (PORTD.3=1))
  NOP
wend
Delay_ms(500)

  goto menuchelo
end if
largochelo:
Lcd_Cmd (LCD_CLEAR)
escribir (1, 1, "toque la cuerda")
Delay_ms (1000)
comparacion (valins, valcue)
if bandera = 2 then
  goto menuchelo
end if
end.
```

MANUAL DEL USUARIO

El presente documento esta orientado a guiar al usuario del Afinador digital de instrumentos musicales con dsPIC30F3014 durante el proceso de afinación de un instrumento para su mejor utilización.

Características Básicas del Afinador.

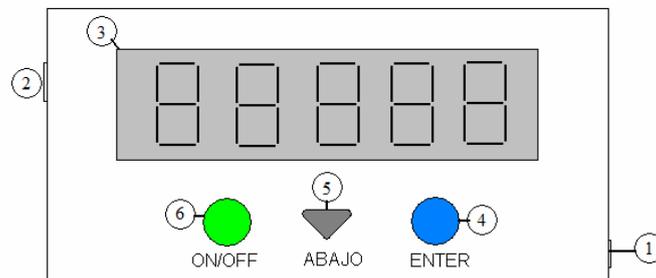


Figura: 9. Diagrama del Afinador digital de instrumentos musicales

- 1 : Alimentación : 5 Voltios
- 2 : Entrada de audio para ingreso de la señal de entrada (micrófono)
- 3: LCD para visualización tanto del menú de control como la afinación
- 4 y 5: Teclas de control, *enter* para el acceso a las diferentes opciones y selección para escoger las opciones deseadas.
- 6: Un botón para encendido y apagado del afinador

Con este afinador usted puede afinar cuatro instrumentos diferentes que son:

- * Guitarra
- * Violín
- * Bajo
- * Chelo

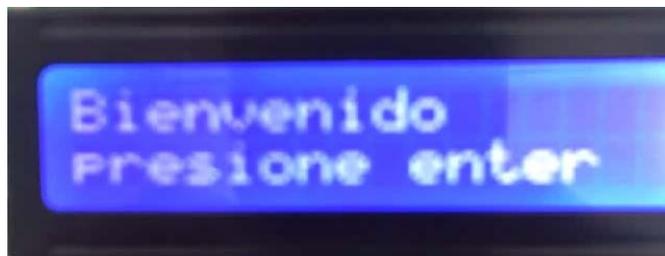
Antes de empezar con la afinación verifique que las posiciones de los botones de control se encuentre como le indica la Figura: 9, también verifique que se encuentre conectada la entrada la audio y la fuente de alimentación. Para afinar un instrumento debe seguir 5 pasos que están detallados a continuación:

PASO 1

Ubique cada una de las características mencionadas arriba antes de encender el prototipo.

PASO 2

Encienda el prototipo (presione el botón de encendido), el momento de encender el afinador se presenta en el LCD un mensaje de bienvenida, el cual permanecerá en la pantalla hasta que se presione la tecla *enter* para continuar.



PASO 3

Una vez que ha presionado la tecla *enter*, en el LCD aparecerá el primer instrumento que puede seleccionar (guitarra), si desea afinar una guitarra presione *enter*, caso contrario presione la tecla de *abajo*, cada vez que presione esta tecla aparecerá en el LCD un instrumento diferente (violín, bajo o chelo) para seleccionar cualquiera de estos instrumentos solo debe ubicarse en el que desea y presionar *enter*. Así usted habrá escogido el instrumento de su elección.



En la siguiente ilustración esta un ejemplo de la selección del instrumento, solo debe presionar la tecla *abajo* una vez y aparecerá la palabra violín.



PASO 4

Seleccionado el instrumento que desea afinar el siguiente paso es escoger la cuerda que va a ser afinada, de la misma forma que hizo la selección del instrumento, ahora deberá seleccionar la cuerda correspondiente para ello se presentará otro menú en el LCD, dependiendo del instrumento seleccionado aparece en pantalla el nuevo menú de selección de la cuerda, al igual que en la selección anterior deberá buscar la nota deseada por medio de la tecla *selección* y presionar *enter* para afirmar la opción. Con

esto se concluye el proceso de selección, a continuación se procederá a afinar el instrumento.



PASO 5

Luego aparecerá en pantalla el mensaje (toque la cuerda) en este momento usted debe esperar a que aparezca la palabra “ya” para pulsar la cuerda y en unos instantes se le indicará que acción debe tomar la afinación, ya sea esto ajustar la cuerda (cuando vemos el mensaje ajustar) o aflojar la cuerda (cuando vemos el mensaje aflojar) o no realizar ninguna acción (cuando el mensaje es Afinación correcta). Si el mensaje fue de ajustar o aflojar deberá repetir el proceso y tocar la cuerda nuevamente con el cambio correspondiente hasta que se presente en el LCD el mensaje Afinación correcta.



Si el mensaje es Afinación Correcta debe presionar *enter* y concluirá el proceso de afinación de su instrumento. Para realizar la afinación completa de un instrumento por favor repita el proceso desde PASO 4.

Gracias por preferirnos, para mayor información contacte a su fabricante en el e-mail pool_guita@yahoo.com estaremos prestos a ayudarle.

Fecha de Entrega :

Ing. Víctor Proaño

**COORDINADOR DE CARRERA EN INGENIERIA ELECTRONICA EN
AUTOMATIZACION Y CONTROL**

Srta. Helena Dolores Ibarra Ruiz

AUTORA