



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

TESIS

**TARJETA DIGITALIZADORA DE 4 CANALES Y 16 BITS
PARA SEÑALES BIOMÉDICAS**

QUE PARA OBTENER EL TÍTULO DE
INGENIERO ELÉCTRICO ELECTRÓNICO

PRESENTA:

FEDERICO MEJÍA CRUZ

DIRECTOR DE TESIS:

M.I. SERGIO QUINTANA THIERRY

MÉXICO, D.F., CIUDAD UNIVERSITARIA, 2015.





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Contenido

INTRODUCCIÓN	1
Capítulo I: MARCO TEÓRICO	3
Capítulo II: ADQUISICIÓN DE DATOS	8
Capítulo III: CONVERTIDOR ANALÓGICO-DIGITAL	12
III.I: Tipo de arquitectura.....	12
III.I.I: Tipo Flash	12
III.I.II: Tipo Sigma-Delta	13
III.I.III: Tipo Aproximaciones Sucesivas.....	14
III.II: Velocidad de muestreo	16
III.III: Voltaje de entrada a escala completa	18
III.IV: Número de bits	18
III.V: Precisión.....	19
III.VI: El convertidor analógico-digital ADS8361.....	21
Capítulo IV: SEÑAL DE RELOJ	26
Capítulo V: PUERTO DE COMUNICACIÓN A LA PC	29
V.I: Características de los puertos de comunicación.....	29
V.II: El puerto de comunicación USB.....	30
V.III: El módulo DLP-USB245M.....	31
Capítulo VI: EL MICROCONTROLADOR DSPIC	36
VI.I: Dispositivo programable	36
VI.II: Lenguaje de programación	38
VI.II.I: Lenguaje C.....	38
VI.II.II: Lenguaje ensamblador.....	39
VI.III: Programa principal.....	40
VI.IV: Ambiente de programación y programador para el dispositivo	43
Capítulo VII: SOFTWARE DE PROCESAMIENTO Y VISUALIZACIÓN DE DATOS.....	46
VII.I: Software o etapa de procesamiento	46
VII.II: Software o etapa de visualización	47
VII.III: Opciones de software de programación.....	50

VII.IV: Interfaz de usuario y algoritmo de programación en LabVIEW.....	52
Capítulo VIII. LA TARJETA DIGITALIZADORA	56
CONCLUSIONES.....	60
BIBLIOGRAFÍA	61
MESOGRAFÍA.....	62
APÉNDICE.....	64

INTRODUCCIÓN

La medicina es la ciencia encargada del estudio de la vida del ser humano, tanto en la salud, las enfermedades y en la muerte; dando origen al conjunto de conocimientos y técnicas creadas para la preservación y recuperación de la salud.

Así, cuando se presenta algún tipo de enfermedad o dolencia, se llevan a cabo una serie de procedimientos: con base en los síntomas ocasionados por una enfermedad, llamada sintomatología y los signos vitales medibles, la semiología, se realizan diferentes estudios y pruebas médicas para determinar el padecimiento, en otras palabras, el diagnóstico, y el origen del mismo, su etiología, analizando el desarrollo de la enfermedad sobre el individuo desde el inicio o su patogenia y las consecuencias o secuelas que se presenten al final, el pronóstico. Durante todo el proceso se llevan a la práctica todos los conocimientos y técnicas que tiene a su disposición y que podrían marcar la diferencia.

Con el transcurso de los años, la medicina se ha valido de recurrir de diferentes métodos para su ampliación y perfeccionamiento que en un principio no eran los mejores ni los más apropiados, pero con el avance de la tecnología en la época reciente, se cuenta con material y equipo médico de alta calidad y gran exactitud que la auxilia en el estudio del cuerpo humano.

Como resultado del avance de la medicina, se ha logrado caracterizar casi en su totalidad el cuerpo humano, por ejemplo, el bombeo de la sangre, la contracción y relajación de los músculos en el movimiento, el proceso de respiración de los pulmones, la actividad cerebral y la comunicación nerviosa, la posición y distribución de los huesos, entre otras más.

En cada situación hospitalaria, se encuentra un instrumento médico especializado que se apoya de algún principio físico o químico para medir de manera exacta la señal biomédica. Como por ejemplo, el nivel de oxigenación en la sangre se mide a través de un oxímetro de pulso, el cual emite un haz de luz y con base en la absorción de la misma, se puede medir la concentración presente en la hemoglobina oxigenada; los rayos X se utilizan para realizar una imagen del interior del cuerpo, se expone a una emisión directa que se recoge en una placa o un detector especial conectado a una computadora en la que se puede apreciar y distinguir las estructuras internas de acuerdo con su densidad.

Para medir la actividad del corazón se usa un electrocardiógrafo que registra la señal bioeléctrica que genera por medio de unos electrodos colocados en ciertos puntos en la piel. Para el resto de los músculos, se usa un electromiógrafo que

mide la señal bioeléctrica. Este tipo de señales, llamadas biomédicas, poseen características muy peculiares, son limitadas en ancho de banda y en magnitud con valores reducidos, en comparación con otro tipo como lo son las de radiofrecuencia, microondas o rayos ultravioleta.

Todas estas señales podrían ser capturadas y digitalizadas por un dispositivo único diseñado con especificaciones bien definidas para satisfacer las necesidades del médico, evitando la necesidad de contar con múltiples instrumentos, sin dejar de lado la calidad requerida.

El diseñar un equipo propio y a la medida traerá consigo ventajas con respecto a los que se encuentran en el mercado; estos últimos son fabricadas con altas especificaciones y una gran gama de funciones conformando un sistema denso y costoso de adquirir que durante la práctica se vuelve algo innecesario, aplicado al caso de manipular señales biomédicas.

Si se plantea el desarrollo de una tarjeta de modo que sea enfocada a este tipo de señales y dedicada solo en la digitalización, se tendrá un ahorro de recursos que traerá consigo bajo costo, sencillez y sea más asequible.

Por lo tanto, nuestro objetivo, para poder satisfacer este problema, será:

Diseñar y construir una tarjeta digitalizadora para señales biomédicas que tenga como características:

- ✓ *16 bits de resolución:*
- ✓ *4 canales analógicos de entrada:*
- ✓ *Comunicación USB a la PC:*
- ✓ *Software de visualización:*

En los capítulos que se presentan a continuación, se establecerán las bases para cumplir con cada una de estas características y los elementos que usarán para la creación de la tarjeta.

Capítulo I: MARCO TEÓRICO

Antes de comenzar a diseñar la tarjeta, primero debemos de conocer las características de las señales biomédicas, en cuanto a rango de voltaje y ancho de banda, ya que a partir de éstas se podrán tener los criterios para elegir los componentes y la lógica más adecuada.

Las señales biomédicas son aquellas que son generadas por algún evento biológico en el cuerpo humano y, de acuerdo con su naturaleza, se pueden clasificar en:

- *Señales Biomecánicas*: son generadas por alguna función mecánica como la locomoción, el flujo y presión de la sangre, entre otras.
- *Señales Bioacústicas*: son producidas por algún evento mecánico, lo que resulta en ruido acústico, como por ejemplo el latido del corazón, la contracción de los músculos, el flujo del torrente sanguíneo, entre otros.
- *Señales de Bioimpedancia*: se obtienen de los tejidos de los órganos al momento de ser estimulados por una corriente eléctrica, dando información sobre las características y composición de los órganos.
- *Señales Bioquímicas*: son el resultado de mediciones químicas de los tejidos a través de muestras analizadas, como por ejemplo la concentración de oxígeno y bióxido de carbono en la sangre.
- *Señales Biomagnéticas*: debido a la presencia de pequeñas corrientes eléctricas, se pueden encontrar campos magnéticos de muy baja intensidad que aportan información sobre el comportamiento de los tejidos excitables.
- *Señales Bioeléctricas*: son propias del sistema biológico y su fuente es el potencial transmembrana que origina una diferencia de potencial bajo ciertas condiciones que es medible a nivel celular.

Las señales bioeléctricas son de las más importantes por el hecho de que todo sistema biológico posee células excitables, las que generan un campo eléctrico que se propaga con facilidad por todo el medio biológico asiendo más fácil su captura, a través de la superficie del sistema sin recurrir a la invasión.

De este grupo de señales se encuentran dos tipos que analizar: las que son producidas por los músculos y por el corazón, la electromiografía y la electrocardiografía, respectivamente.

La electromiografía es el registro de la actividad eléctrica de los músculos del cuerpo humano que dan origen al movimiento y posición del mismo. Se produce a nivel celular en los músculos gracias a un potencial de acción originado por las

diferentes concentraciones de distintos iones dando como resultado una función contráctil.

El músculo presenta dos tipos de unidades básicas: la unidad anatómica y la unidad funcional (figura 1). La unidad anatómica está comprendida por la fibra muscular que al ser estimulada se contrae desarrollando fuerza, así un músculo está formado por haces paralelos de fibras musculares que en conjunto pueden dar movimiento a una zona específica del cuerpo.

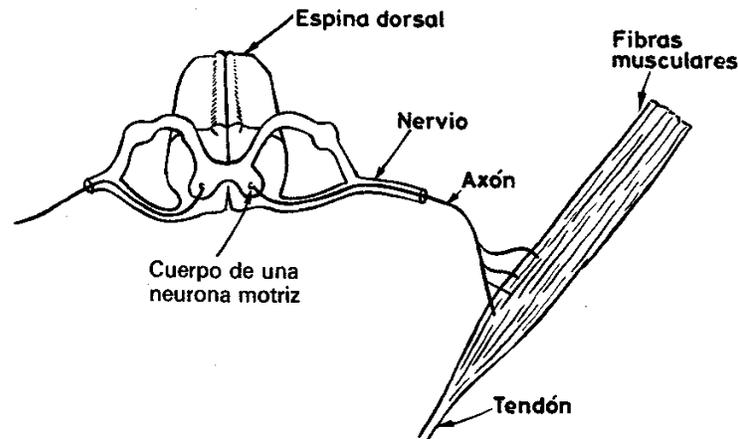


Figura 1: Estructura de la unidad anatómica y la unidad funcional.

La unidad funcional o motora la componen el grupo formado por la neurona motriz localizada en la espina dorsal, el nervio, el axón y las fibras musculares que éste inerva. Así, para la realización de un movimiento simple es necesario la intervención de una o muchas unidades motoras que trabajan en conjunto para poder llevar a cabo la acción indicada. Además, un músculo puede tener más de un axón unido a él, dependerá de la posición y función del mismo.

En el cerebro se crea el impulso eléctrico, éste se desplaza por medio de la espina dorsal hasta las unidades motoras necesarias que estén inervadas al músculo dando como resultado la contracción o relajación, a veces es imprescindible varios impulsos a la vez para llevar a cabo esta tarea.

Por medio de una electromiografía se registran estos impulsos eléctricos, del cual se puede deducir la activación de las unidades motoras, las características de estas activaciones y la relación que existe entre cada una de las unidades.

Las características de los impulsos son muy variables, dependiendo de la función del músculo a realizar, pero se pueden encontrar entre rangos de valores establecidos. La amplitud de estas señales van de $500[\mu V_{pp}]$ hasta los $10[mV_{pp}]$ con un ancho de banda que comprende frecuencias que pueden llegar a los $3[kHz]$ aunque aquellas que predominan se encuentran hasta los $500[Hz]$ (figura 2), donde las que poseen mayor energía se limitan de los $30[Hz]$ a los $150[Hz]$.

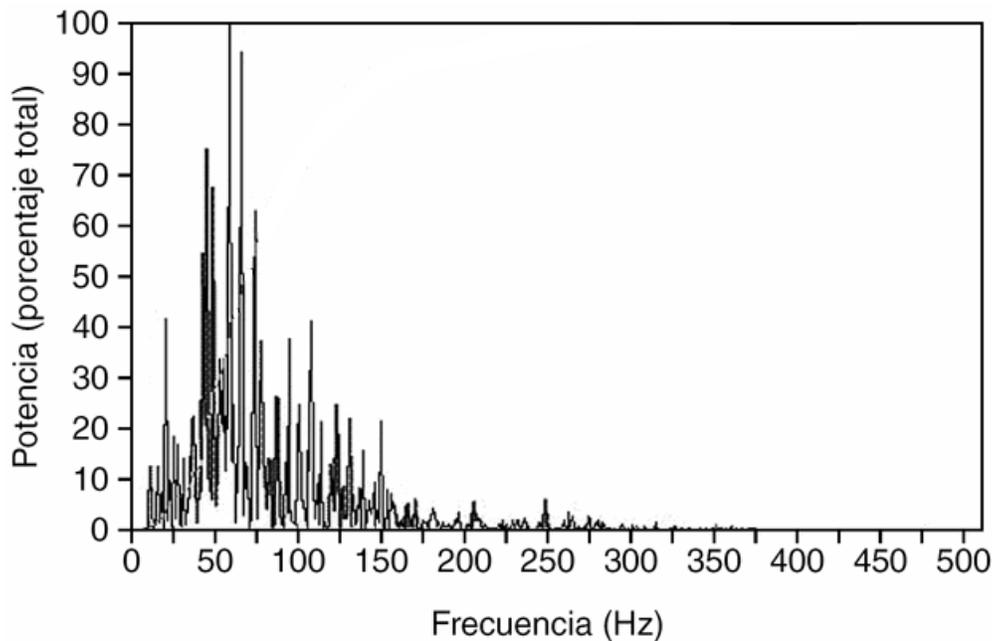


Figura 2: Espectro de frecuencias de las señales electromiográficas.

Así, la electromiografía representa la suma de los potenciales de acción de cada una de las unidades motoras, creados por los impulsos eléctricos continuos e irregulares de las unidades activas en el músculo, lo que ocasiona una forma de onda irregular.

Un caso particular en la electromiografía lo representa el corazón, el cual es un músculo hueco que se encarga del bombeo de la sangre, principal motor del sistema circulatorio del cuerpo humano. Éste posee una forma de onda particular que es estudiada por una rama específica de la electromiografía, denominada electrocardiografía.

La electrocardiografía es un registro de la actividad eléctrica del corazón sobre la superficie de la piel, la cual se recoge por medio de una gráfica, llamada electrocardiograma, que refleja las variaciones del potencial eléctrico generadas por las células cardiacas durante el proceso de bombeo de la sangre.

Para poder obtener un electrocardiograma es necesario colocar unos electrodos en puntos específicos del cuerpo humano, entonces se registra la diferencia de potencial entre dos puntos, llamado derivación, ya sea entre dos electrodos (derivación bipolar) o entre un punto virtual y un electrodo (derivación monopolar). Estos puntos suelen ser el brazo derecho, el brazo izquierdo, la pierna izquierda y como punto virtual el centro del triángulo imaginario creado por los puntos anteriores, donde el potencial eléctrico es cercano a cero.

Todas las derivaciones presentan gráficas distintas bien estudiadas, cada una de ellas representan el mismo estímulo eléctrico aunque visto desde diferentes puntos de referencia. De aquí se desprende la señal cardiaca característica (figura 3) que ejemplifica el funcionamiento del corazón, tomada de la derivación aVF (mide el potencial del pie izquierdo respecto al de la unión de los brazos).

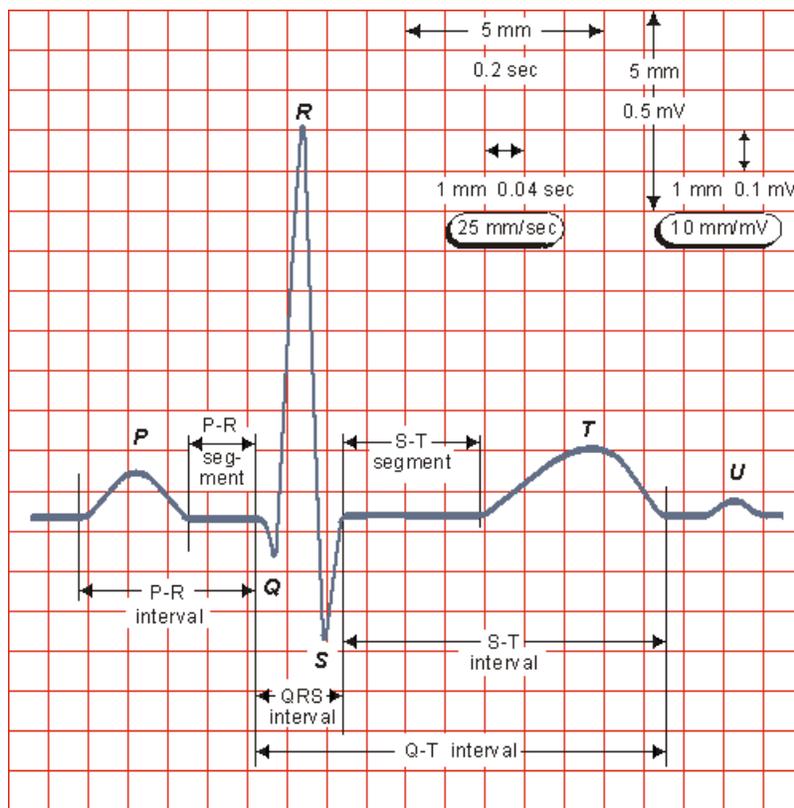


Figura 3: Señal cardiaca característica de una persona sana.

Esta señal tiene las siguientes propiedades:

- *Onda P*: indica el inicio de la forma de onda de la señal, corresponde a la despolarización de las aurículas. Su duración es menor de $100[ms]$ y su voltaje no excede los $0.25[mV]$.
- *Onda Q*: la primera deflexión negativa resultante de la despolarización ventricular. Su duración está entre los $10[ms]$ y $20[ms]$.
- *Onda R*: la primera deflexión positiva resultante de la despolarización ventricular.
- *Onda S*: la segunda deflexión negativa proveniente de la despolarización ventricular.
- *Complejo QRS*: es el rasgo más característico de la señal cardiaca y la forman la onda Q, R y S. Representa el tiempo total de la despolarización ventricular, tiene un voltaje no mayor a $1[mV]$ y un lapso de $60[ms]$ a $100[ms]$.
- *Intervalo P-R*: figura el retraso fisiológico que sufre el estímulo que viene de las aurículas a su paso por el nodo auriculoventricular. Va desde el inicio de la onda P hasta la de la onda R, tiene una duración de $120[ms]$ a $200[ms]$.
- *Onda T*: la deflexión positiva ocasionada por la repolarización ventricular. Tiene un voltaje de entre $0.15[mV]$ a $0.20[mV]$ y un tiempo menor de $200[ms]$.
- *Intervalo Q-T*: comprende desde el inicio de la onda Q hasta el fin de la onda T, es la suma de la despolarización y repolarización ventricular. Tiene una duración de $320[ms]$ a $400[ms]$.
- *Onda U*: es apenas perceptible, con escaso voltaje y muy corta duración. Se cree que es originada por la repolarización de los músculos papilares.

Como se puede apreciar, la señal cardiaca contiene elementos de baja frecuencia, como la onda T, y de alta frecuencia, como el complejo QRS, a pesar de que la componente fundamental es de $1[Hz]$, así el ancho de banda va de $0.5[Hz]$ hasta $150[Hz]$ en el caso cuando el corazón tiene una palpitación muy apresurada.

Capítulo II: ADQUISICIÓN DE DATOS

Como se ha establecido en nuestro objetivo, la tarea principal de nuestra tarjeta es la digitalización de las señales analógicas de entrada, que es una de las etapas de la adquisición de datos.

La adquisición de datos es un proceso que consiste en una representación equivalente, por la cual una variable física es representada en una señal digital, con el fin de que puedan ser manipuladas por sistemas electrónicos para su análisis. Son convertidas en señales eléctricas como voltaje o corriente, para luego digitalizarlas y obtener información binaria que es menos vulnerable al ruido y a otras interferencias.

Este proceso consta de tres partes principales: sensor o transductor, dispositivo de acondicionamiento y una PC (figura 4).

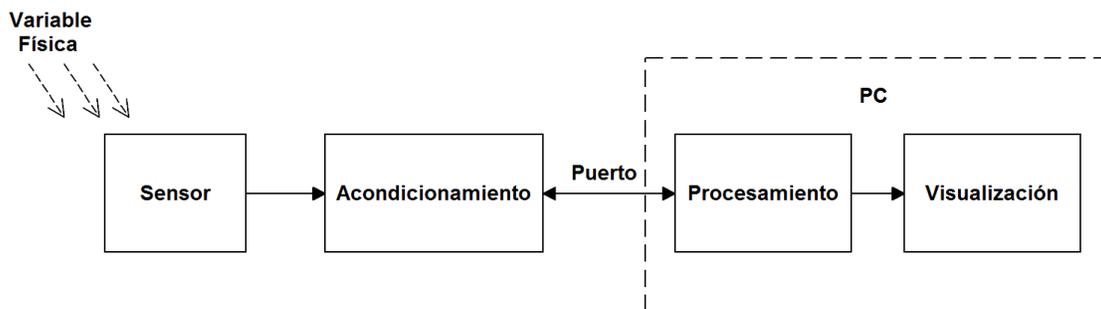


Figura 4: Estructura básica de un sistema de adquisición de datos.

El sensor o transductor trabaja transformando una variable física en una señal eléctrica que, dependiendo del tipo, puede ser voltaje, corriente, impedancia o similar. En muchos casos, esta señal eléctrica sensada no está en condiciones de ser medida de manera correcta, a lo que habrá de ser acondicionada por medio de un circuito de amplificación, filtrado, atenuación o aislamiento, según el tipo de transductor y de la señal requerida.

El dispositivo de acondicionamiento se encarga de convertir estas señales eléctricas en señales digitales por medio de un convertidor analógico-digital, el cual se encarga de proporcionar una representación digital de una señal analógica

en un tiempo determinado. Pero como las señales de este tipo cambian continuamente en el tiempo, la conversión se deberá repetir para que la representación sea lo más fidedigna posible.

Entre el dispositivo de acondicionamiento y la PC se tiene un puerto de comunicación, que tiene como función la conexión entre ellos para el envío de instrucciones y datos. Existen diferentes tipos de puertos de comunicación, de acuerdo a con capacidades y limitaciones, y cada uno se adapta mejor que otro según la aplicación.

Con base en el esquema para la adquisición de datos, la PC tiene como propósito controlar el dispositivo de acondicionamiento y procesar, almacenar y visualizar la información recopilada, además, el usuario puede interactuar con el sistema gracias al uso de una interfaz creada con un software programable dando la libertad de poder seleccionar diferentes opciones de acuerdo al requerimiento que tenga.

Hoy en día, los dispositivos de acondicionamiento han encapsulado muchas y diversas funciones en un solo módulo gracias a una gran y densa circuitería interna que ofrece unas excelentes características aunque esto se ve reflejado en los precios de los mismos.

Existen muchas empresas que se encargan de la fabricación de estos dispositivos, como Analog Devices, Advantech Corporation, Freescale Semiconductor, Xilinx, Altera Corporation, Microchip Technology, entre otros, donde el mayor productor es National Instruments, que acapara más de la mitad del mercado con un muy extenso catálogo que van de dispositivos pequeños con características muy limitadas hasta complejos con características muy elevadas.

Estos sistemas en el mercado, se clasifican de acuerdo al tipo de puerto de salida a la PC, número de canales analógicos y/o digitales de entrada y/o salida, número de bits de resolución, número máximo de muestras por segundo, tipo de medida, rango de voltaje de entrada, número de circuitos auxiliares como contadores, temporizadores, tipo de aislamiento, disparo, entre otros; entre más elementos cuente el dispositivo o tarjeta, su precio se elevará considerablemente.

Por ejemplo, existe el producto NI USB-6000, un dispositivo de adquisición de señales multifunción de bajo costo para medidas básicas de calidad. El producto NI PXIe-4499, un analizador de señal dinámica de ganancia flexible, ambos de National Instruments y con características muy diferentes (tabla 1).

Tabla 1: Comparación entre los productos NI USB-6000 y NI PXIe-4499.

<i>Producto</i>	<i>NI USB-6000</i>	<i>NI PXIe-4499</i>
<i>Tipo de puerto</i>	USB	PXI Express
<i>Número de canales</i>	8	16
<i>Número de bits</i>	12	24
<i>Velocidad de muestreo</i>	10kSPS (compartido)	204.8kSPS (por canal)
<i>Voltaje de entrada</i>	-10 a 10 [V]	-10 a 10 [V]
<i>Precio</i>	\$1799	\$153,700

Como puede apreciarse, el producto NI PXIe-4499 posee unas características muy superiores a las de NI USB-6000 junto a su precio, pero tal vez excesivas para varias aplicaciones las cuales no aprovecharían al máximo sus capacidades al grado de considerarse un lujo innecesario.

Para nuestro caso en el que nuestra meta es muestrear señales de electromiografía y electrocardiografía y tener una buena representación en la pantalla de una PC, no sería óptimo utilizar una de estas tarjetas ya que, además de tener un precio elevado, no concuerdan con las especificaciones planteadas (tabla 2).

Por lo tanto, contar con un dispositivo o tarjeta que se ajuste a las demandas de la electromiografía y electrocardiografía que, por sus características, no requieran excesivos recursos de hardware permitirá que tengan una circuitería adecuada y a la medida, haciendo hincapié solo en la digitalización o conversión de las señales sin dejar de lado la calidad y la capacidad para obtener un producto de buenas características a un precio mucho menor respecto a las que se pueden conseguir en el mercado.

El producto final contará con la tarjeta digitalizadora que contemple un bus de comunicación y un software con una interfaz de usuario para la PC. Las especificaciones de la tarjeta (tabla 2) son obtenidas a partir de las características de las señales biomédicas y algunas establecidas en nuestro objetivo. Como referencia, éstas son superiores a las encontradas en dispositivos de bajo costo, aunque no contará con algún tipo de acondicionamiento para dar la posibilidad de conectar diferentes tipos de señales biomédicas, esto será tarea de cada canal.

Tabla 2: Especificaciones generales de la tarjeta digitalizadora

<i>Tipo de puerto</i>	USB
<i>Número de canales</i>	4
<i>Número de bits</i>	16
<i>Frecuencia máxima de entrada</i>	3[kHz] (por canal)
<i>Voltaje de entrada</i>	-2.5 a 2.5 [V]

La construcción de la tarjeta se iniciará con la elección del convertidor analógico-digital que se implementará, a partir de él se irán integrando los demás bloques que conforman a un sistema de adquisición respetando las especificaciones preestablecidas y agregando elementos extras para el perfecto acoplamiento entre ellos.

Capítulo III: CONVERTIDOR ANALÓGICO-DIGITAL

El convertidor analógico-digital (ADC) es la parte fundamental de la tarjeta al realizar la labor de la digitalización, dependiendo de sus características se deberán implementar, junto a él, ciertos circuitos integrados para su correcto funcionamiento como por ejemplo una señal de reloj a determinada frecuencia, etc.

Actualmente, los convertidores ADC vienen integrados en un solo chip y disponibles en una infinidad de variantes en sus características para así ajustarlo a nuestra aplicación. Para poder elegir el más adecuado, necesitamos tomar en cuenta 5 especificaciones principales:

1. *Tipo de arquitectura*
2. *Velocidad de muestreo*
3. *Voltaje de entrada a escala completa*
4. *Número de bits*
5. *Precisión*

III.1: Tipo de arquitectura.

Existen varios tipos de convertidores analógico-digital, pero en el mercado están presentes tres de ellos: Flash, Sigma-delta y por Aproximaciones Sucesivas. Cada una de ellas tiene sus ventajas y desventajas que se pueden equilibrar dependiendo de la aplicación donde se use.

III.1.1: Tipo Flash

El ADC tipo flash tiene como principal ventaja la velocidad de conversión que se limita al tiempo de respuesta de sus elementos pero es el más costoso de todos. Está formado por una serie de divisores de voltaje conectados a comparadores y un codificador (figura 5); al aplicar un voltaje de entrada en los comparadores, éstos activan el codificador el cual genera un código binario que representa el resultado de la conversión.

El número de comparadores está dado por $2^n - 1$, donde n es el número de bits del ADC, así para $n = 8$ se necesitan 255 comparadores para hacer la conversión lo que resulta impráctico y muy costoso, solo cuando el número de bits es bajo se vuelve rentable.

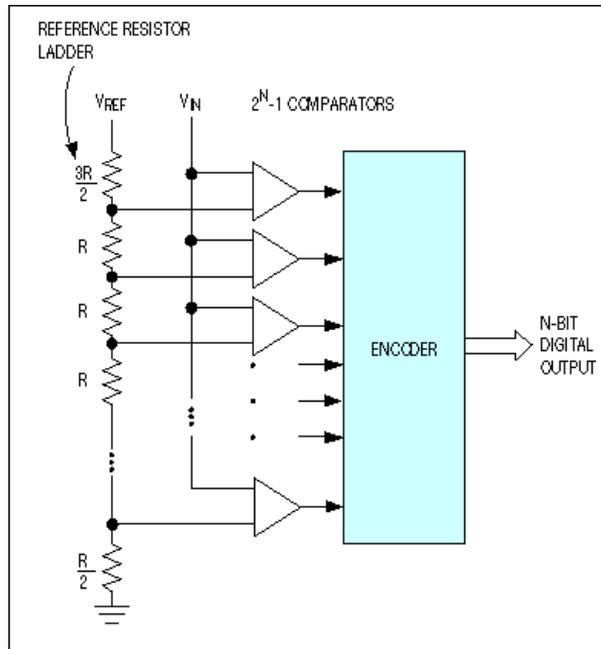


Figura 5: Estructura general de un convertidor analógico-digital tipo flash.

III.1.11: Tipo Sigma-Delta

El ADC tipo sigma-delta tiene una estructura sencilla, consiste en un modulador sigma-delta seguido de un filtro digital-decimador. Este modulador cuenta con un integrador y un comparador con un lazo de realimentación en el que se conecta a un convertidor digital-analógico de 1 bit, además posee una señal de reloj que sincroniza de manera correcta al modulador y al filtro (figura 6).

En el modulador analógico, la señal de entrada se suma con la salida del comparador, que es un flujo de datos o una onda de pulsos modulados a una gran velocidad y de un solo bit que pasa por el DAC de 1 bit. Luego son integradas para después ser comparada con respecto a un voltaje de referencia, dando como resultado ceros o unos.

Este tren de pulsos pasa por un filtro seguido de un decimador, el cual convierte la cadena de ceros y unos en una señal PCM multibit, lo cual se traduce como la resolución o número de bits, que es la señal promediada.

Una de las mayores desventajas que posee este convertidor es que el modulador $\Sigma\Delta$ tiene un alto nivel de ruido, lo cual puede ser compensado incluyendo varios niveles de integración (más integradores conectados uno después de otro) o sobremuestrear la señal, además de que el filtro digital quitará el resto del ruido.

Una de las características por la que sobresale de los otros ADC es por la alta resolución (número de bits) con las que puede representar la señal de entrada, si aumenta la velocidad de conversión la resolución se reducirá significativamente y con el sobremuestreo para aminorar el ruido, el circuito se vuelve cada vez más complejo.

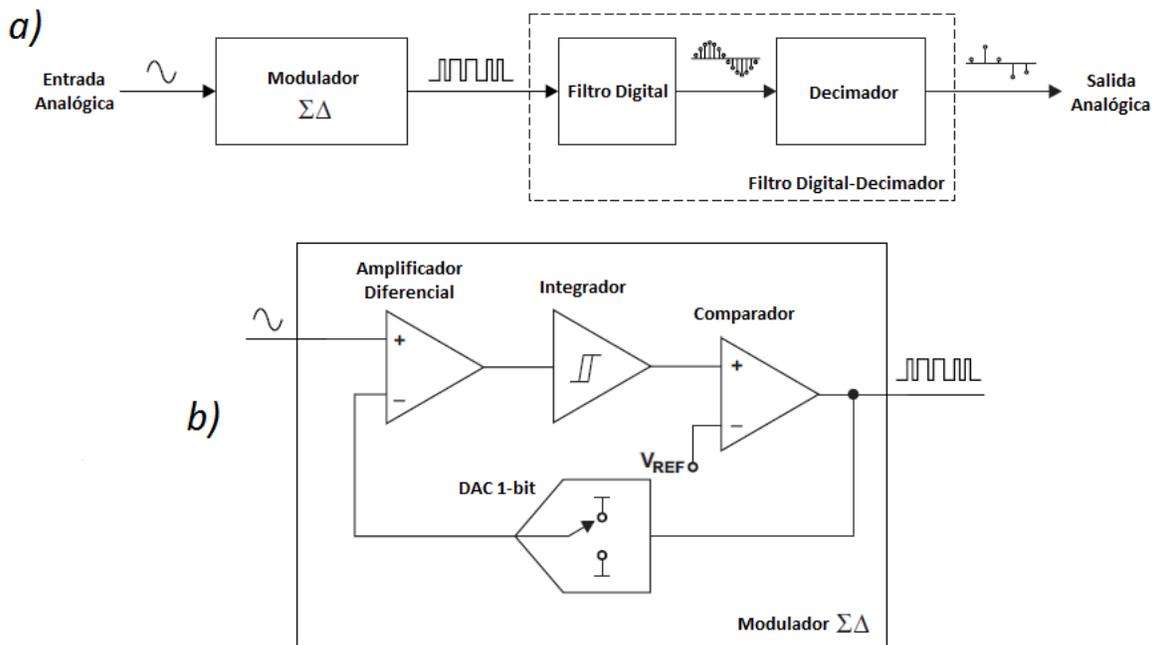


Figura 6: a) Estructura general de un convertidor analógico-digital tipo sigma-delta y b) bloques internos del modulador $\Sigma\Delta$.

III.I.III: Tipo Aproximaciones Sucesivas

El ADC por aproximaciones sucesivas funciona a partir de un registro de aproximaciones sucesivas (SAR), donde se realizan comparaciones sucesivas de manera ascendente o descendente hasta obtener la combinación que iguala el voltaje de entrada con la salida del DAC. Este convertidor está compuesto por un registro de aproximaciones sucesivas, un reloj, un comparador, un circuito Muestreo & Retención y un DAC (figura 7).

El voltaje de entrada llega al circuito Muestreo & Retención donde se mantendrá durante toda la conversión. Antes de comenzar, el registro contiene el valor más bajo posible (000...0), al recibir la instrucción de comenzar su valor cambia y en el

bit más significativo (MSB) almacena un 1 que equivale a la mitad del voltaje de referencia de la conversión.

Este nuevo valor del registro es enviado al DAC donde se convierte en una señal analógica y es mandada al comparador, donde se indicará si es mayor o menor al voltaje de entrada. Si es mayor, se escribe un 1 en el segundo bit del registro, que pasa de 100...0 a 110...0 pero si es menor, el MSB pasa a 0 y el segundo bit pasa a 1, es decir, de 100...0 a 010...0.

Nuevamente el valor del registro pasa al DAC y su salida al comparador donde se verifica si es mayor o menor a la señal de entrada modificando los siguientes bits del registro y así sucesivamente. El número de repeticiones de este proceso es igual al número de bits en su salida digital, todas marcadas por una señal de reloj.

Este tipo de convertidor posee la gran ventaja de tener un equilibrio entre velocidad de conversión y resolución en su salida digital, entre mayor resolución mayor exactitud, aunque su costo se puede elevar.

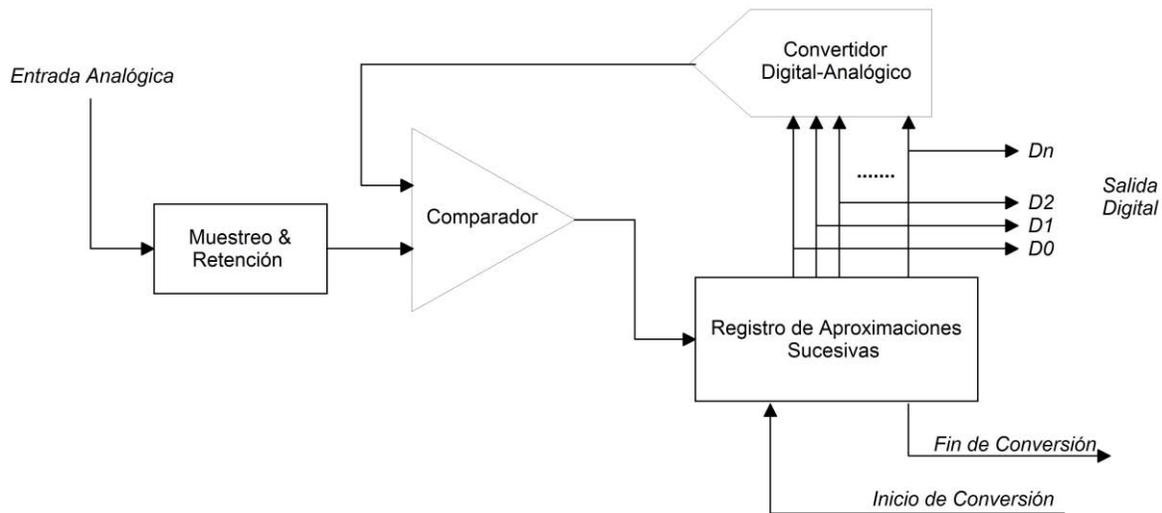


Figura 7: Estructura general de un convertidor analógico-digital por aproximaciones sucesivas.

Así, el tipo de arquitectura sentará la relación entre la velocidad de respuesta y el máximo número práctico posible (figura 8) y que dependiendo de la aplicación, resultará mejor uno que otro.

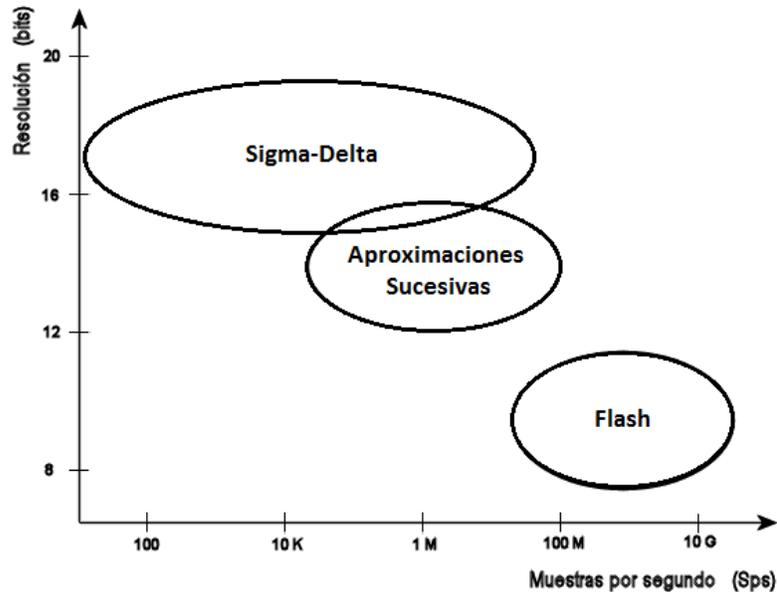


Figura 8: Gráfica de comparación entre los tipos de convertidores.

III.II: Velocidad de muestreo

Una de las especificaciones más importantes a considerar al momento de elegir un convertidor analógico-digital es la velocidad de muestreo, que es la velocidad a la que se puede tomar muestras de la señal de entrada, la cual se mide en muestras por segundo [SPS].

El teorema de Nyquist-Shannon establece:

Sea una señal de banda limitada ($f_{MÁX}$), entonces la señal estará determinada por sus muestras si se cumple que

$$f_s \geq 2f_{MÁX} \quad (\text{Oppenheim, 2009, p.158})$$

Donde f_s es la velocidad de muestreo. De manera práctica y por las limitaciones de los circuitos reales, se requiere una velocidad de muestreo mayor a la establecida por Nyquist. Una buena reconstrucción (figura 9) se logra muestreado mínimo a diez veces mayor rápido que la señal $f_s \geq 10f_{MÁX}$, aunque se pueden obtener buenos resultados desde $f_s \geq 5f_{MÁX}$.

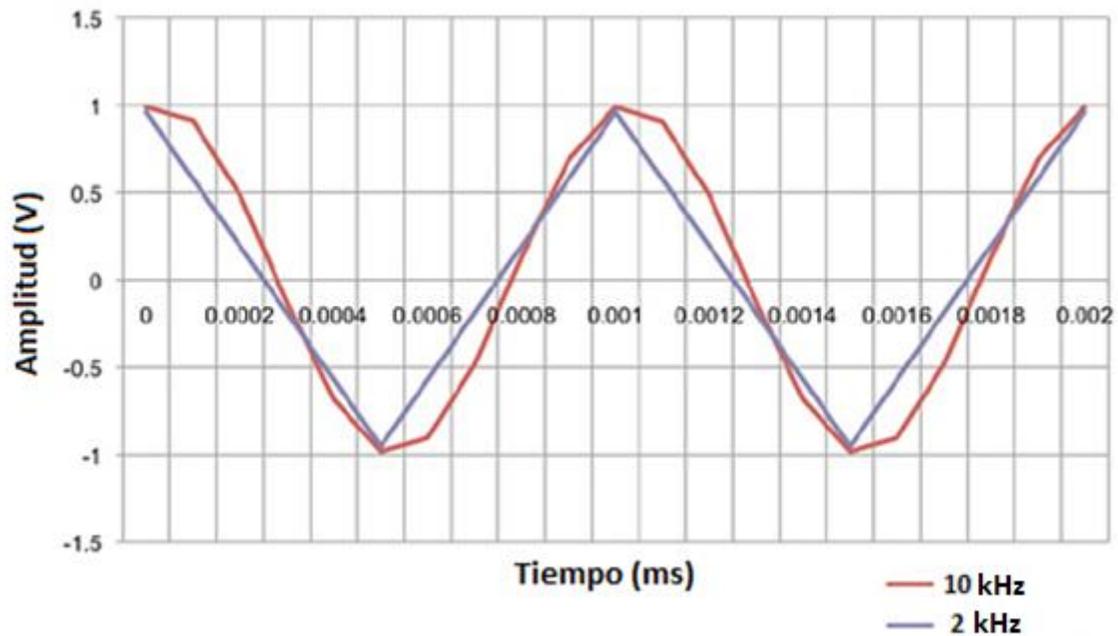


Figura 9: Reconstrucción de una señal senoidal de 1kHz con dos diferentes velocidades de muestreo.

Muestrear una señal por debajo de la frecuencia recomendada por Nyquist, es decir, $f_s \geq 2f_{MAX}$ puede dar origen al fenómeno llamado aliasing, que es cuando se obtienen muestras de tal forma que al reconstruirla se obtiene la misma señal pero a una frecuencia menor a la original, o simplemente se obtienen formas de onda incorrectas con frecuencias variadas (figura 10).

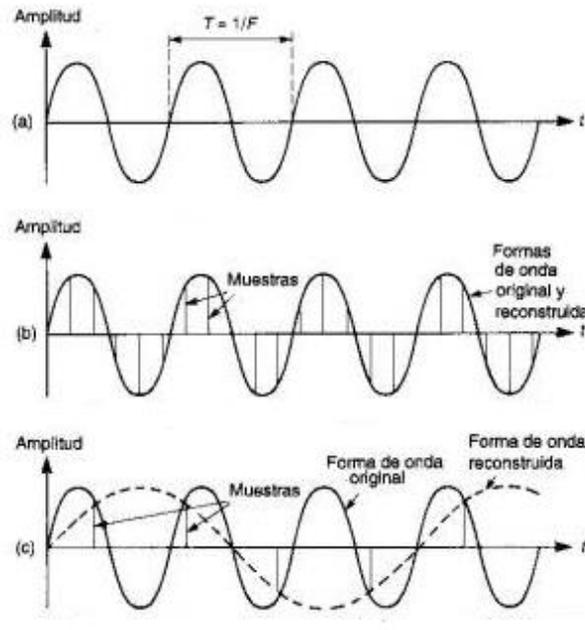


Figura 10: a) una señal senoidal muestreada con b) una frecuencia mayor a la de Nyquist y a c) una frecuencia menor a la de Nyquist (aliasing).

III.III: Voltaje de entrada a escala completa

Las señales de salida que generan algunos sensores son muy difíciles de medir, ya sea por ser muy pequeñas o por ser muy grandes. Para esto se requiere que estas señales se acondicionen, con etapas de amplificación, filtrado y/o linealización, atenuación y/o aislamiento.

Este acondicionamiento va enfocado para que la señal se ajuste al voltaje de entrada a escala completa y así aprovechar al máximo el convertidor. Los típicos voltajes presentes en los convertidores analógico-digital son de $\pm 2.5[V]$, $\pm 5[V]$ y $\pm 10[V]$, aunque pueden variar.

III.IV: Número de bits

El número de bits (resolución) indica la cantidad de niveles binarios con los que se puede representar a la señal de entrada. Así, se tiene que:

$$\text{Número de niveles binarios} = 2^n$$

Donde n es el número de bits. Se aprecia que entre más grande sea, la señal de entrada será representada con mayor exactitud, aunque puede requerir un circuito

más complejo o un mayor tiempo de conversión. Cuando $n = 3$ se tienen 8 niveles binarios y para $n = 16$ son 65536 niveles binarios (figura 11).

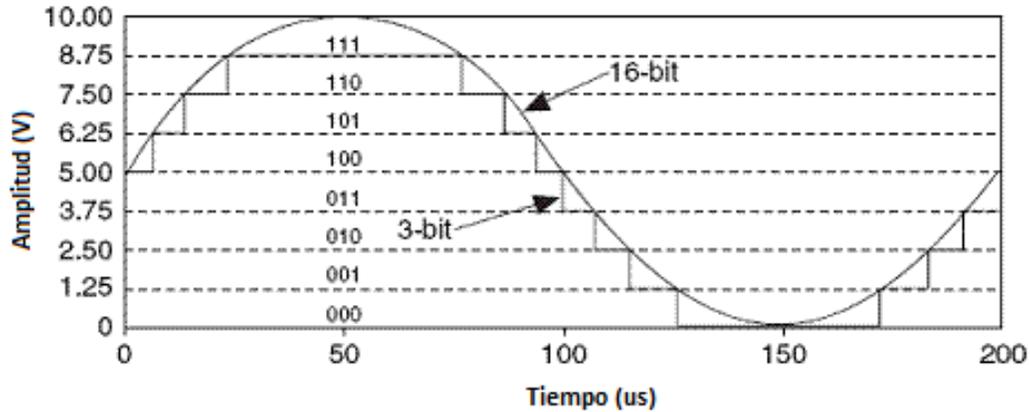


Figura 11: Representación de una señal senoidal con dos resoluciones diferentes, a 3 bits y a 16 bits.

Existe una relación entre el número de bits y el voltaje de entrada a escala completa, el cambio detectable más pequeño de la señal de entrada, que es la razón de cambio del voltaje de entrada que se requiere para cambiar en un 1 LSB (bit menos significativo) en la salida digital. Se define como:

$$\text{Voltaje mínimo para cambiar en 1LSB} = \frac{V_{FS}}{2^n}$$

Donde V_{FS} es el voltaje de entrada a escala completa y n es el número de bits. Entre menor sea este valor, se puede hacer una reconstrucción más fiel a la original.

III.V: Precisión

La precisión en un convertidor analógico-digital recae en su comportamiento al momento de la conversión, en como su relación entrada-salida se acerca a lo ideal (figura 12), aunque con las limitaciones de los circuitos reales esto no es posible.

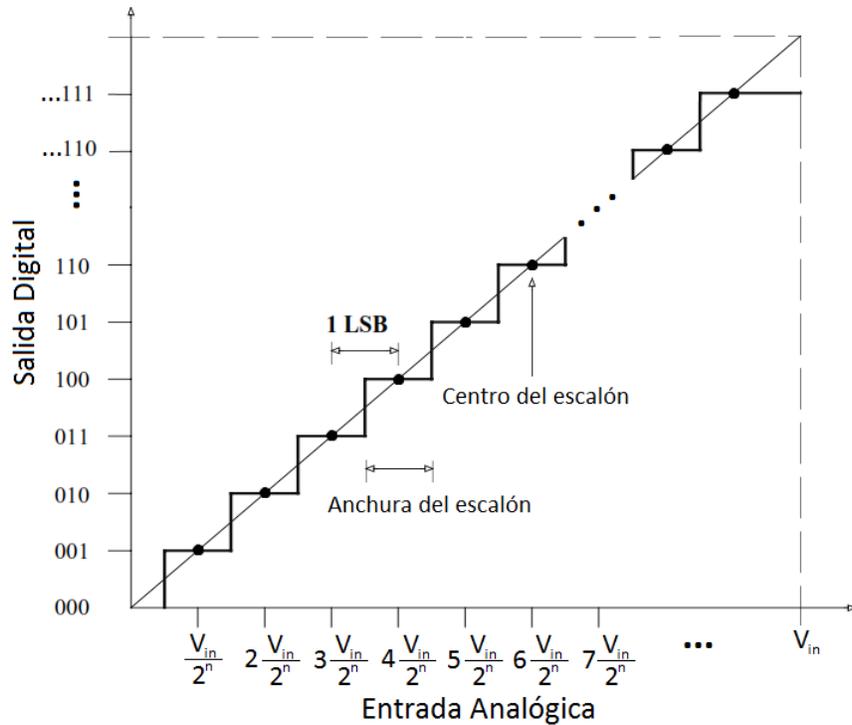


Figura 12: Relación entrada-salida en un convertidor analógico-digital ideal

La conversión ideal es una función escalonada, donde cada escalón mide $1LSB$ de ancho por cuyo centro pasa una línea imaginaria con una pendiente de 45° que inicia en el origen, además su altura también mide $1LSB$.

Al momento de la conversión, se pueden presentar errores que hacen que la función escalonada no tenga el comportamiento deseado que, dependiendo de su origen, se puede corregir. Algunos de estos errores son:

Error de offset: es la diferencia entre el punto ideal de origen de la función y el punto real de origen.

Error de fondo de escala: es la diferencia entre el punto ideal a fondo de escala de la función y el punto real a fondo de escala. Este error provoca que la línea imaginaria tenga una inclinación o pendiente diferente a la de 45° .

Error diferencial no lineal (DNL): es la diferencia entre el valor de la anchura de un escalón ideal y la anchura de un escalón real. El valor ideal es de $\pm\frac{1}{2}LSB$ y está determinado por el caso máximo encontrado. Para garantizar que no haya pérdida de información, este error debe ser menor a $1LSB$.

Error integral no lineal (INL): es la diferencia o desviación que existe entre la línea imaginaria ideal y la línea imaginaria real, al comparar los centros de los escalones. Hay dos formas de medirlo, una es al considerar los errores de offset y de fondo de escala en la línea imaginaria ideal (Beststraight-line INL) y la otra es con la línea imaginaria que parte del origen y termina en el voltaje de fondo de escala, es decir, sin considerar los errores de offset y de fondo de escala.

Otras especificaciones a considerar son:

THD (Distorsión Armónica Total): es la relación que existe entre la componente fundamental de la señal (primera armónica) y el contenido armónico (señales múltiplos de la original) a la salida que es introducido por el ADC. Este parámetro es expresado en unidades porcentuales (%) o en decibeles (dB) y es un muy buen indicador de la distorsión en la señal de salida.

SNR (Relación Señal-Ruido): es la relación existente entre la potencia de la señal de entrada y la potencia de ruido que intenta corromperla. Se mide en decibeles (dB) y entre mayor sea, el ruido no podrá alterar la señal de entrada.

SINAD (Relación Señal-Ruido-Distorsión): es la relación existente, en potencia, entre la señal de entrada y las señales indeseadas (perturbaciones), que en este caso son la distorsión y el ruido. Sus unidades son los decibeles (dB) y entre más grande sea su valor, la calidad de la señal de entrada no se verá afectada.

SFDR (Rango Dinámico Libre de Espurios): es la relación que existe entre la amplitud de la componente fundamental de la señal y la amplitud de la componente armónica más prominente. Se expresa en decibeles (dB) y entre mayor sea su valor, la potencia de la componente de la fundamental no se verá tan afectada por la potencia de las armónicas.

III.VI: El convertidor analógico-digital ADS8361

El ADS8361 es un convertidor analógico-digital fabricado por Texas Instruments Incorporated (AP-01), de tecnología CMOS, que tiene como características:

Tabla 3: Características del convertidor analógico-digital ADS8361

<i>Tipo de Arquitectura</i>	Aproximaciones Sucesivas
<i>Velocidad de Muestreo Máxima</i>	500kSPS (máx.)
<i>Frecuencia de Reloj de Operación</i>	100[kHz] a 10[MHz]
<i>Voltaje de Entrada a Escala Completa</i>	-2.5 a 2.5 [V]
<i>Número de Bits</i>	16
<i>Número de Canales</i>	4
<i>Tipo de Entrada</i>	Diferencial y Terminal Única
<i>Error de Offset</i>	±0.5[mV]
<i>Error Diferencial no Lineal (DNL)</i>	+1.5LSB
<i>Error Integral no Lineal (INL)</i>	±3LSB
<i>Distorsión Armónica Total (THD)</i>	-94[dB]
<i>Relación Señal-Ruido (SNR)</i>	83[dB]
<i>Relación Señal-Ruido & Distorsión (SINAD)</i>	83[dB]
<i>Rango Dinámico Libre de Espurios (SFDR)</i>	94[dB]
<i>Formato de Salida</i>	Serial
<i>Voltaje de Polarización</i>	5[V]

Este convertidor cuenta con las características que se han establecido para la tarjeta digitalizadora, como el manejo de 4 canales, 16 bits de resolución y la velocidad de muestreo capaz de soportar señales de 3[kHz], siendo capaz de muestrear como máximo:

$$\frac{500[kSPS]}{4canales} = 125[kSPS] \text{ muestreando a } f_s = 10f_{in}$$

Se tiene que cada canal puede registrar señales de 12.5[kHz] y si $f_s = 5f_{in}$ puede admitir señales de 25[kHz].

Una vez que elegimos el convertidor, podemos calcular el valor del voltaje más pequeño detectable de la señal de entrada, teniendo que el voltaje de entrada a escala completa es de 5[V] y el número de bits es de 16, entonces:

$$\text{Voltaje mínimo para cambiar en 1LSB} = \frac{V_{FS}}{2^n} = \frac{5}{2^{16}} = 76.3[\mu V]$$

Así, la variación del voltaje de la señal más pequeña que puede ser detectada por el convertidor es de $76[\mu V]$ y la variación más pequeña en la entrada es de $500[\mu V]$, por lo que no habrá pérdida de información.

El funcionamiento del convertidor gira entorno a la operación de dos señales de control, la señal del reloj de entrada (*Señal de reloj*) y la señal de inicio de conversión (*Señal "Inicio de conv."*), donde existe la posibilidad de configurar la salida de los datos (*Señal "Datos en serie"*) de 4 modos distintos a través de tres pines de control (pin 15: *M0* pin 14: *M1* y pin 16: *A0*):

- Modo I: con $M0=0$ y $M1=0$, la conversión de los canales 1 y 2 se transmitirán a través de "SERIAL DATA A" (pin 23) y la de los canales 3 y 4 se hará a través de "SERIAL DATA B" (pin 22), con $A0=0$ se convertirán los canales 1 y 3, y con $A0=1$ los canales 2 y 4.
- Modo II: con $M0=0$ y $M1=1$, la conversión de todos los canales se harán de forma secuencial a través de "SERIAL DATA A", con $A0=0$ se convertirán los canales 1 y 3, y con $A0=1$ los canales 2 y 4.
- Modo III: con $M0=1$ y $M1=0$, la conversión de los canales 1 y 2 se transmitirán, de forma secuencial, a través de "SERIAL DATA A" (pin 23) y la de los canales 3 y 4 se hará, de forma secuencial, a través de "SERIAL DATA B" (pin 22) sin importar el valor de *A0*.
- Modo IV: con $M0=1$ y $M1=1$, la conversión de todos los canales se harán de forma secuencial a través de "SERIAL DATA A" sin importar el valor de *A0*.

El convertidor estará configurado en el Modo IV, así solo se maneja una línea de salida y los datos de los canales se tendrán uno tras otro y en orden ascendente.

Los datos de salida son de tipo serial con el siguiente formato:

Tabla 4: Formato de salida de los datos en serie

Ciclo de reloj	1	2	3	4	5	6	7	8	9	10
Datos serial	MSB_{ChX}	LSB_{ChX}	$D15$	$D14$	$D13$	$D12$	$D11$	$D10$	$D09$	$D08$

Ciclo de reloj	11	12	13	14	15	16	17	18	19	20
Datos serial	$D07$	$D06$	$D05$	$D04$	$D03$	$D02$	$D01$	$D00$	0	0

Por cada 20 ciclos de reloj se tiene una conversión completa, los dos primeros ciclos corresponde al número de canal, que está dado en binario donde el ciclo 1 es el bit más significativo (MSB_{ChX}) y el ciclo 2 el bit menos significativo (LSB_{ChX}). Desde el ciclo 3 hasta el ciclo 18 corresponde a los 16 bits de la conversión, empezando desde el bit más significativo ($D15$) hasta el menos significativo ($D00$). Los últimos ciclos son tomados por el convertidor para prepararse para la siguiente conversión y éstos siempre serán ceros.

La señal de inicio de conversión le indica al convertidor en qué momento empezar su operación y que, desde el ciclo de reloj en el cual está presente, comienzan los 20 ciclos que dura la conversión, esta debe durar al menos 2 ciclos y como máximo 18 ciclos, para nuestro caso, es de 7 ciclos.

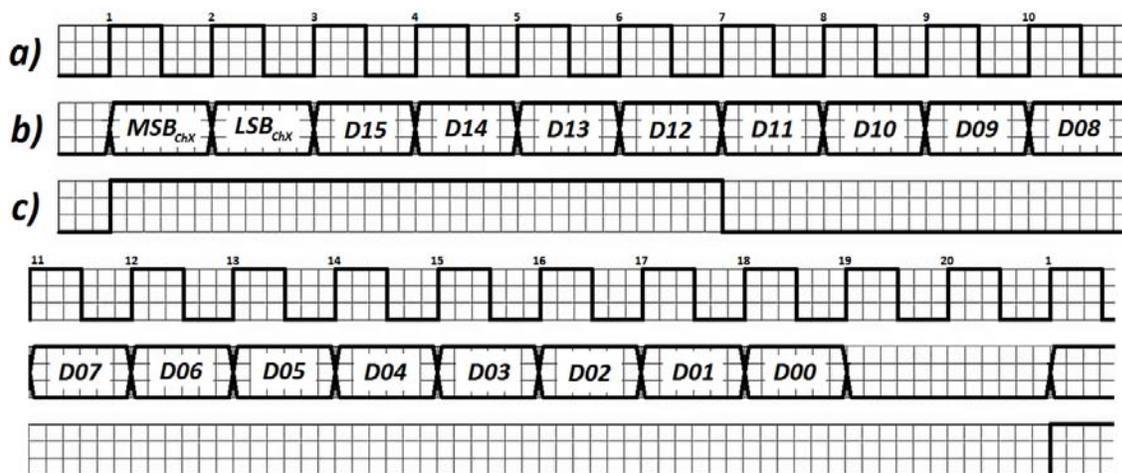


Figura 13: Diagrama de tiempo con a) la señal de reloj, b) la salida serial de los datos y c) la señal de inicio de conversión.

Así, el diagrama de tiempos del convertidor (figura 13) muestra la relación entre las señales de entrada y de salida.

El circuito del convertidor (figura 14) indica que los canales estarán conectados en entrada de terminal única, donde el negativo de cada señal tendrá un voltaje común V_{COM} que, por especificaciones del fabricante, valdrá $2.5[V]$ y sirve como punto medio en la escala de conversión. Además, se necesita un voltaje de referencia V_{REF} de $2.5[V]$ suministrado por el mismo convertidor (pin 11 conectado a pin 10); el cual indica los límites en la amplitud de la señal de entrada, la cual va de $V_{COM} + V_{REF}$ a $V_{COM} - V_{REF}$. Así, las entradas de los canales pueden variar de $-2.5[V]$ a $2.5[V]$ con un voltaje común, o de offset, de $2.5[V]$.

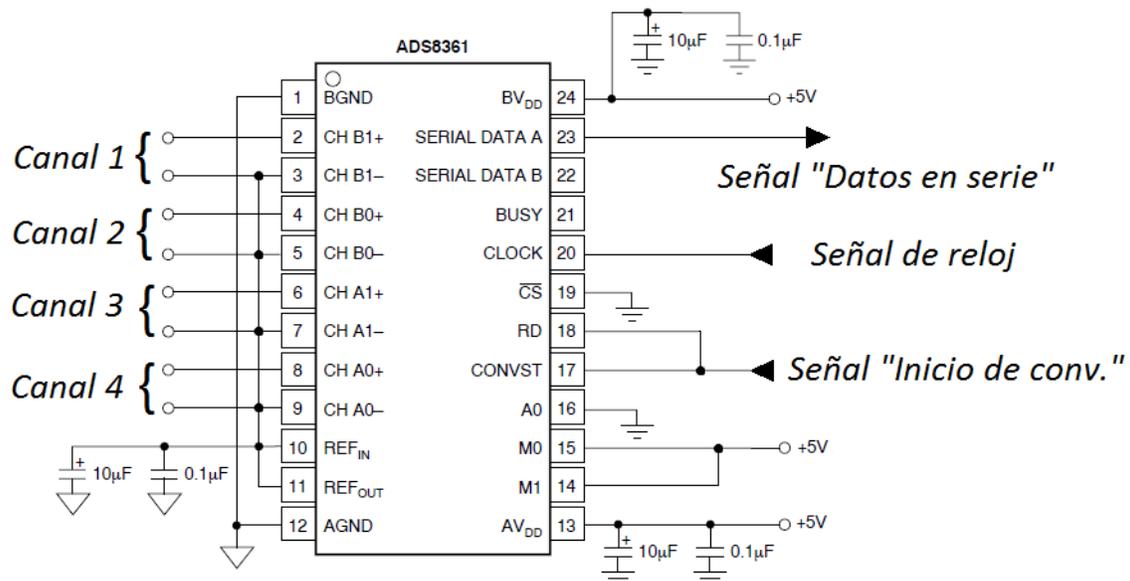


Figura 14: Diagrama de conexiones del convertidor ADS8361.

Capítulo IV: SEÑAL DE RELOJ

Una vez elegido el convertidor ADC, podemos comenzar por establecer los demás bloques que conforman la tarjeta, empezando por la señal de reloj que determina la velocidad de operación.

La ejecución de la tarjeta digitalizadora es regida por la sincronización a través de una señal de reloj, la cual debe ser proporcionada por un bloque específico construido por un temporizador que sea capaz de suministrarla con la frecuencia necesaria.

La conversión de un canal tarda 20 ciclos de reloj, donde se tiene que $f_{ChX} = \frac{f_{CLK}}{20}$, pero el convertidor está operando en el Modo IV, lo que pasa a ser $f_{ChX} = \frac{f_{CLK}}{20(4)} = \frac{f_{CLK}}{80}$ y recordando que si se muestrea a $f_s = 10f_{in}$ al final resulta:

$$f_{ChX} = \frac{f_{CLK}}{800}$$

La cual es la frecuencia máxima que puede admitir cada canal dependiendo del valor del reloj de entrada. Si $f_s = 5f_{in}$ entonces $f_{ChX} = \frac{f_{CLK}}{400}$.

Si la frecuencia máxima de una señal de entrada permitida por cada canal es de $3[kHz]$ y su relación con el reloj de entrada, para nuestro caso, es de $f_{ChX} = \frac{f_{CLK}}{400}$ entonces:

$$f_{CLK} = 400f_{ChX} \text{ donde } f_{ChX} = 3(10^3) \text{ por lo que se tiene}$$

$$f_{CLK} = 1.2[MHz]$$

Si se mantiene fija la frecuencia en este valor, las señales alrededor de lo máximo permitido por la tarjeta estarán muestreadas alrededor de $f_s = 5f_{in}$ pero las que poseen una frecuencia de unos cuantos hertz serán muy sobremuestreadas. Si cada canal tiene una $f_s = 15[kHz]$ y se introduce una señal, por ejemplo, de

$$f_{in} = 10[Hz] \text{ entonces la relación que se tiene es de } \frac{f_s}{f_{in}} = \frac{15 \times 10^3}{10} = 1500, \text{ es decir,}$$

se muestrea a una frecuencia mayor a 1500 veces, donde lo práctico es de 5 a 10 veces.

Este sobremuestreo excesivo se va reduciendo cuando aumenta la frecuencia de la señal de entrada aunque se siguen teniendo demasiadas muestras de sobra, las cuales tienen la peculiaridad de que muchas de ellas representan el mismo punto; así se puede hacer una discriminación y solo conservar las necesarias para poder hacer una buena reconstrucción.

Para poder hacer una discriminación más rápida en las señales de unos cuantos hertz, será necesario bajar la frecuencia de la señal de reloj, la cual está limitada por el convertidor analógico-digital que solo funciona a partir de una frecuencia mínima de $f_{CLK} = 100[kHz]$. Tomando en cuenta la relación $f_{chX} = \frac{f_{CLK}}{800}$ se tiene que cada canal puede admitir una frecuencia de $f_{chX} = 125[Hz]$.

La tarjeta digitalizadora operará con dos distintas frecuencias, una de $f_{CLK} = 100[kHz]$ para señales que van de $0[Hz]$ a $125[Hz]$ y otra de $f_{CLK} = 1.2[MHz]$ para cubrir el intervalo de $125[Hz]$ a $3[kHz]$, dando lugar a dos modos de operación.

Para tener una completa compatibilidad en tecnología de los chips, el temporizador debe ser de tipo CMOS y además, poder ofrecer frecuencias en el orden de los megahertz.

El chip TLC556 es un doble temporizador de tecnología CMOS que puede alcanzar una frecuencia máxima de $f_{CLK} = 2.1[MHz]$, puede ser polarizado en un rango de voltaje de $2[V]$ a $18[V]$, posee la capacidad de operar en modo astable, tiene un consumo de energía de $2[mW]$ cuando es polarizado con $V_{DD} = 5[V]$ y se configura de igual manera que el temporizador NE556. Uno de los temporizadores estará operando a una frecuencia de $f_{CLK1} = 100[kHz]$, el otro a una de $f_{CLK2} = 1.2[MHz]$ y a través de un multiplexor de 2 a 1 se seleccionará la frecuencia que tendrá el sistema (figura 15); así se tienen los dos modos de operación de la tarjeta digitalizadora.

Gracias a un interruptor tipo ON-OFF de un polo - un tiro junto a un transistor 2N3904 configurado como una compuerta NOT, se puede seleccionar la frecuencia de operación habilitando uno u otro temporizador.

Para el caso del multiplexor, el chip 74HC157 con tecnología CMOS (AP-02), posee 4 multiplexores 2 a 1 junto a una línea común de selección donde sus salidas son no invertidas y son controlados por un bit de encendido. Así, se introducen las dos señales de reloj y con el control de selección se elegirá la frecuencia a la cual correrá el sistema.

El multiplexor es habilitado por una señal de encendido (*Señal "ON/OFF"*) proveniente de un interruptor tipo ON-OFF de un polo - un tiro, que hará la función de encender y apagar la operación de la tarjeta.

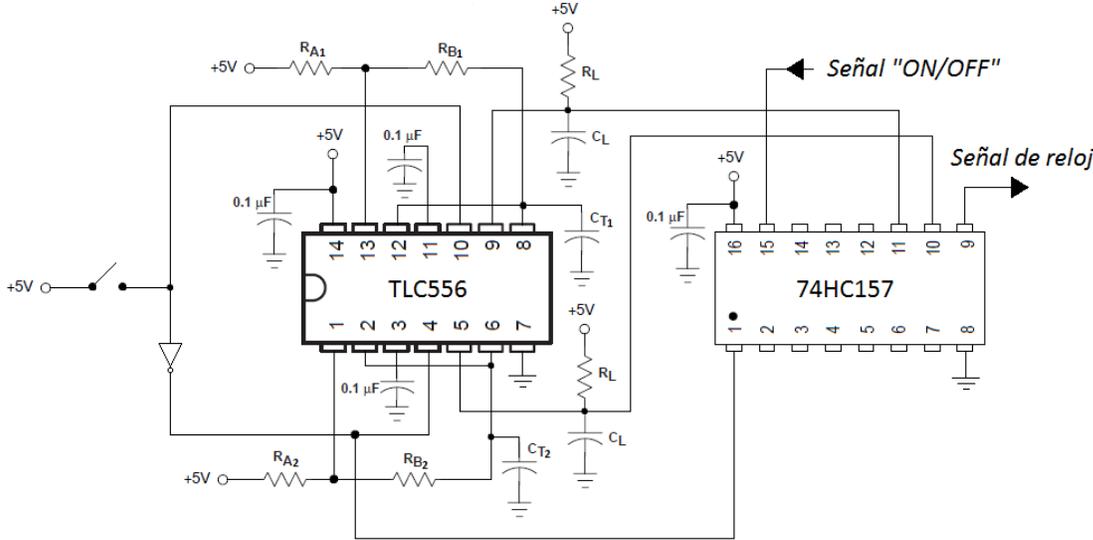


Figura 15: Diagrama de conexiones de la señal de reloj.

Capítulo V: PUERTO DE COMUNICACIÓN A LA PC

Hasta el momento se ha diseñado la parte de la digitalización, dando como resultado una cadena de bits representativos de las señales de entrada; ahora es momento de enviar esta información a la PC para que se puedan procesar, con la ayuda de un puerto de comunicación.

Existe una gran variedad de protocolos de comunicación, cada uno de ellos con sus ventajas y desventajas en el marco del rendimiento, ancho de banda, protocolo de comunicación, latencia, portabilidad, distancia óptima a la PC sin pérdida de datos, entre otros; y será uno mejor que otro dependiendo la aplicación a montar.

V.I: Características de los puertos de comunicación

Los puertos de comunicación más usados en la actualidad son:

- *PCI, PCI Express*: es un puerto estándar que usan los dispositivos periféricos para conectarse directamente sobre la tarjeta madre de la PC, así ocupa un espacio en el interior de la misma que, por lo común, suele ser un poco voluminoso dado al tamaño del conector del hardware. Tiene un ancho de banda hasta de $250[MB/s]$ y se polariza con $5[V]$ los cuales se obtienen desde la tarjeta madre.
- *PXI*: es un puerto que tiene un chasis robusto el cual funciona con una variante del puerto PCI, donde se conectan múltiples periféricos que trabajan simultáneamente a través de una sincronización especializada y un software muy eficaz. Tiene un ancho de banda que puede alcanzar los $4[GB/s]$ y este chasis puede ser controlado con una PC independiente o una integrada al mismo. Tiene un alto costo por lo que solo se utiliza en pruebas industriales, militares, de manufactura y monitoreo de máquinas.
- *Ethernet*: es un puerto muy usado en las redes de PC de área local. Se compone de una tarjeta PCI donde se conecta un cable Ethernet el cual puede medir hasta $100[m]$ para evitar pérdida de datos o hasta más con la ayuda de unos repetidores de señal, esto permite que exista una comunicación a distancia entre el punto de la medición y la PC. Tiene un ancho de banda de $125[MB/s]$, dependiendo de tipo de protocolo y de cable; además de que es de bajo costo comparado con otros.

- *Inalámbrico*: este tipo de comunicación ofrece flexibilidad y portabilidad al no contar con cables y sin importar la construcción del lugar, lo que da como resultado un bajo costo por el mínimo uso de hardware y tiempo de instalación aunque posee la mayor latencia entre todos los tipos de comunicación. Según sea el protocolo usado, el ancho de banda es de $6.75[MB/s]$, por lo que no es recomendado para aplicaciones de alta velocidad.
- *USB*: este puerto originalmente se empezó a utilizar para conectar dispositivos periféricos a la PC; actualmente, gracias a su bajo costo y su facilidad para usar, se puede encontrar en casi cualquier tipo de dispositivo electrónico. Sin embargo, es inherentemente latente y no determinista. Su ancho de banda es de $60[MB/s]$ dependiendo del protocolo, tiene la característica de la detección automática de los periféricos, la autoconfiguración y no se necesita apagar la PC para adicionar o remover un dispositivo (Plug & Play).

La estandarización del uso de la comunicación USB en los dispositivos actuales se ha originado gracias a sus ventajas, optar por este tipo de puerto hará que la tarjeta pueda ser usada sin problema alguno en cualquier PC.

V.II: El puerto de comunicación USB

Para implementar, de manera sencilla, la comunicación USB necesitamos un dispositivo que opere con este tipo de puerto, ya que los problemas que se presentan al momento de configurarlo son el protocolo de transferencia y la velocidad de transmisión hacia la PC.

Existen cuatro tipos de protocolo o modo de transferencia:

- *Transferencia de Control*: se usa normalmente para hacer operaciones de control o de estado de los periféricos por parte del software a usar, que por lo común son peticiones o respuestas de comunicación.
- *Transferencia Asíncrona*: es de tipo constante, es la sincronización a una señal de reloj la cual está dada por el emisor y el receptor, donde los caracteres están separados por intervalos aleatorios.
- *Transferencia Isocrónica*: es continua, los caracteres están separados por intervalos preestablecidos y se constituye en un sincronismo similar a la transferencia asíncrona.

- *Transferencia Sincrónica*: es de tipo constante, aquí se tiene que la señal de reloj va implícita en la transmisión, dando como resultado menos errores y una mayor distancia de alcance.
- *Transferencia de Volumen*: no es continúa, se usa al momento de transferir una gran cantidad de información en forma de paquetes de gran tamaño, de manera rápida usando todo el ancho de banda disponible.
- *Transferencia por Interrupción*: no es periódica, se utiliza cuando se desea tener un flujo de datos con muy poca información de manera rápida, que va solamente desde el dispositivo USB a la PC.

La eficacia del modo de transferencia ya ligada con la máxima velocidad de transmisión que puede soportar el canal del puerto. Se cuenta con 4 diferentes versiones:

- *USB 1.0 (Baja Velocidad)*: con una velocidad de transmisión de $1.5[Mbit / s]$ o $188[kB / s]$ se usa principalmente en dispositivos de interfaz humana, como teclados, ratones, entre otros.
- *USB 1.1 (Velocidad Completa)*: tiene una máxima velocidad de transmisión de $12[Mbit / s]$ o $1.5[MB / s]$,
- *USB 2.0 (Alta Velocidad)*: alcanza una velocidad de transferencia de $480[Mbit / s]$ o $60[MB / s]$, esta versión es la que se utiliza en la mayoría de los dispositivos electrónicos en el día de hoy.
- *USB 3.0 (Súper Alta Velocidad)*: llega hasta los $4.8[Gbit / s]$ o $600[MB / s]$, es el más reciente de todos, se ha modificado la construcción del puerto para poder alcanzar esta velocidad, aunque es totalmente compatible con las versiones anteriores.

V.III: El módulo DLP-USB245M

Este módulo es un adaptador a USB (AP-03), fabricado por DLP Design, que no requiere tener un amplio conocimiento de este tipo de puerto para poder manejarlo, muchas de las funciones de configuración y comunicación hacia la PC se realizan de manera automática dependiendo del tipo de información a enviar o recibir, lo cual hace más fácil la implementación y uso del USB.

Una de las ventajas que tiene este módulo es la posibilidad de ser reconocido por la PC como un puerto COM virtual, para ser tratado como un puerto serie, o como

un dispositivo USB. En ambos casos, existen drivers, proporcionados por el fabricante, que facilitan su manipulación a través de un software de programación.

Posee las siguientes características:

Tabla 5: Características generales del módulo DLP-USB245M

<i>Voltaje de Polarización</i>	4.4[V] a 5.25[V]
<i>Modo de Transferencia</i>	Isocrónica y de Volumen
<i>Velocidad de Transmisión</i>	USB 2.0
<i>Interfaz de Entrada de Datos</i>	Paralelo
<i>Tipo de Conector USB</i>	Tipo B

La forma en que se introducen los datos al módulo es por medio de una entrada de tipo paralelo, por medio de 8 pines que forman un byte pero el convertidor tiene una salida tipo serie. Para que haya una compatibilidad entre ambos, será necesario que exista un bloque que pase de serie a paralelo, para este caso se trata de un registro de corrimiento serie-paralelo de 8 bits.

Para saber cómo los unirá, necesitamos establecer el protocolo para enviar los datos, es decir, en qué forma se mandarán los 20 bits del convertidor a la PC.

El envío de una conversión consta de 4 bytes, el bit 7 (MSB) de cada byte es usado como bit de reconocimiento, los bits 6 y 5 no son usados por lo que siempre serán "0", y del bit 4 al 0 estarán presentes los datos de la conversión. Los bytes se presentan a continuación:

- *Byte 1*: En su bit más significativo se coloca un "1", el cual indica que es el primero de la serie (en los demás bytes es un "0"), los bits 4 y 3 están asociados con el ciclo de reloj 19 y 20 de la conversión respectivamente, por lo que a partir del bit 2 inicia un nuevo dato, así el bit 2 y 1 contienen el número de canal y el bit menos significativo posee el bit 15 de la resolución del convertidor.

Tabla 6: Formato del Byte 1 del envío de una conversión.

Bit	7	6	5	4	3	2	1	0
Dato	1	0	0	0	0	MSB_{ChX}	LSB_{ChX}	D15

- **Byte 2:** En el bit 7 se coloca un “0” y del bit 4 al 0 se asocian con los bits 14 al 10 de la resolución, respectivamente.

Tabla 7: Formato del Byte 2 del envío de una conversión.

Bit	7	6	5	4	3	2	1	0
Dato	0	0	0	D14	D13	D12	D11	D10

- **Byte 3:** Su bit 7 tiene un “0” y los bits 9 al 5 de la resolución están presentes del bit 4 al 0, respectivamente.

Tabla 8: Formato del Byte 3 del envío de una conversión.

Bit	7	6	5	4	3	2	1	0
Dato	0	0	0	D09	D08	D07	D06	D05

- **Byte 4:** Su bit más significativo es un “0”, los bits 4 al 0 tienen los bits 4 al 0 de la resolución del convertidor.

Tabla 9: Formato del Byte 4 del envío de una conversión.

Bit	7	6	5	4	3	2	1	0
Dato	0	0	0	D04	D03	D02	D01	D00

El registro de corrimiento a usar es el 74HC164 (figura 16), de tecnología CMOS y de tipo serie a paralelo de 8 bits, el cual va a estar controlado por la señal de reloj de la tarjeta que, para fines prácticos, estará desfasada 180° para evitar una pérdida de datos.

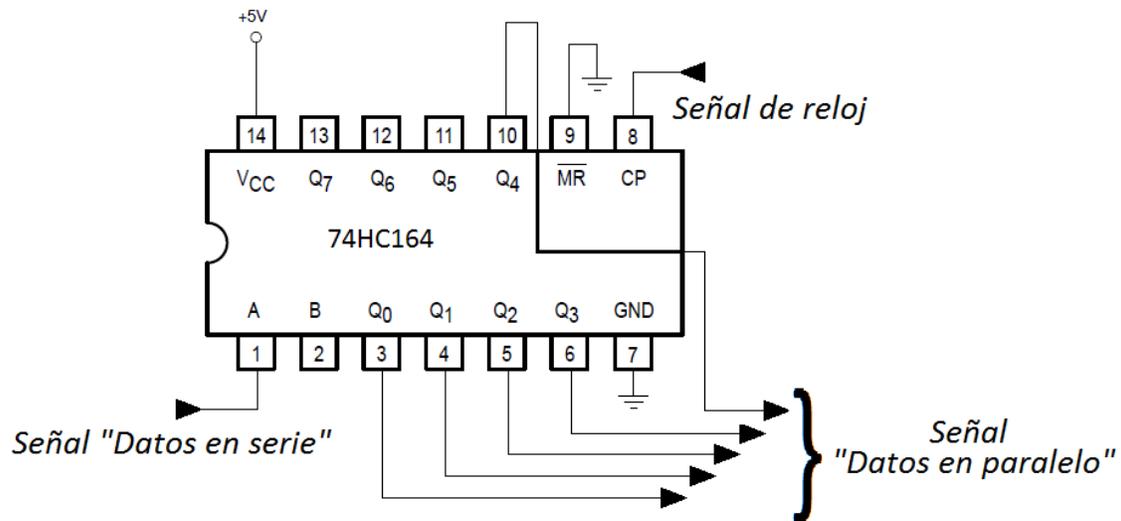


Figura 16: Diagrama de conexiones del registro de corrimiento.

Para poder realizar el bit de reconocimiento, nos auxiliamos de la señal de inicio de conversión (*Señal "Inicio de conv."*) la cual está configurada de tal forma que dura lo suficiente para estar presente solo al momento del envío del primer byte. La transferencia de datos está marcada por una señal (*Señal "Enviar datos"*) que indica que cada 5 ciclos de reloj de la tarjeta se envía un byte.

El módulo tiene cuatro tipos diferentes de configuraciones de operación:

- *Configuración 1:* El módulo es alimentado con $5[V]$ los cuales son suministrados por el mismo puerto, sin requerir una fuente externa. Debido a las limitaciones del USB, la corriente máxima que se puede obtener es de $100[mA]$ por dispositivo y $500[mA]$ por cada puerto.
- *Configuración 2:* El módulo es polarizado con $5[V]$ a través de una fuente externa al puerto, con lo que se puede exigir una mayor corriente sin afectar al USB.
- *Configuración 3:* El módulo puede trabajar con un voltaje de $3.3[V]$ y se maneja de igual manera que la configuración 2.
- *Configuración 4:* A través de un pin que tiene el módulo, se controla su polarización a través de un MOSFET tipo Canal P. Se usa principalmente cuando se requiere una corriente mayor a $100[mA]$ y es compatible con los voltajes $5[V]$ y $3.3[V]$.

El voltaje de polarización de la tarjeta es 5[V], que proviene de una fuente externa, a los cuales trabaja el convertidor analógico-digital y los otros circuitos, entonces para que el módulo trabaje a la misma fuente se seleccionará la configuración 2 (figura 17), sin recurrir a la alimentación disponible del USB.

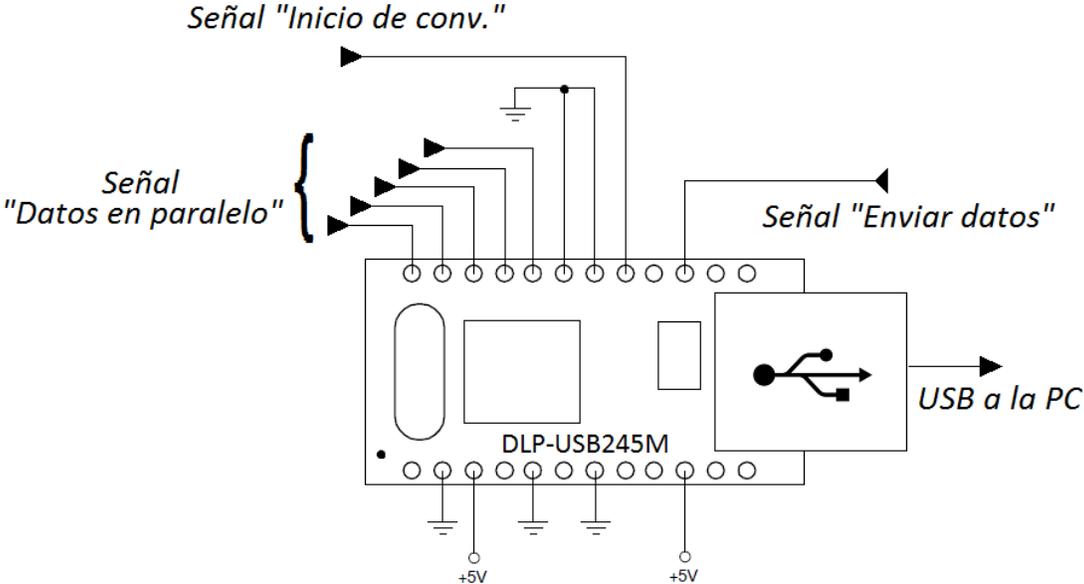


Figura 17: Diagrama de conexiones del módulo USB.

Capítulo VI: EL MICROCONTROLADOR DSPIC

Con el convertidor ADC digitalizando las señales analógicas y el módulo USB recibiendo y enviando los bits de información, se requiere que exista una correcta sincronización entre estos dos bloques y para llevarlo a cabo será necesario un dispositivo que proporcione las señales con las que trabajan en el momento indicado.

La implementación consta de cuatro elementos:

- 1) El dispositivo programable
- 2) El lenguaje de programación
- 3) El ambiente de programación
- 4) El programador para el dispositivo

VI.1: Dispositivo programable

El dispositivo programable debe ser lo suficiente rápido y flexible en su programación, deberá leer la señal de reloj, llevar el conteo de los ciclos y mandar las señales correspondientes a cada circuito.

Existen una gran variedad de dispositivos programables, según su arquitectura, que podrían realizar esta tarea pero existe un fabricante en particular que además de producir los dispositivos programables, también hacen el resto de los elementos necesarios para la implementación, lo cual asegura una máxima compatibilidad entre ellos para evitar algún error.

El fabricante MicroChip Technology Inc. produce una gama de dispositivos programables llamada PIC, de tecnología CMOS, que ofrece facilidad con la programación y diferentes frecuencias de trabajo. Su arquitectura tiende a ser minimalista al contar con un set de instrucciones reducido donde el tiempo de ejecución, por lo común, es de cuatro ciclos de reloj, un área de datos separada de la de código, posee una memoria RAM de limitada capacidad direccionada a la memoria FLASH que contiene el programa, opera con un solo registro de trabajo o acumulador a donde se transfiere la información por medio de las instrucciones donde se selecciona de forma implícita, entre otras.

Una de las consideraciones a tomar en cuenta al seleccionar el PIC es que opere a una frecuencia suficiente como para leer la señal de reloj, ejecute las instrucciones del programa fuente y mandar las señales necesarias a cada

dispositivo. Existe una clasificación de acuerdo con las características y por la frecuencia de trabajo:

- *Gama baja*: llega a las 2MIPS (millones de instrucciones por segundo), son pequeños al contar solo con 6 pines, su hardware es limitado por lo que los programas que permiten están condicionados. Aquí están los PIC10F.
- *Gama media*: alcanza las 5MIPS, es la más extensa de todas, a partir de aquí se establecen los componentes o elementos con los que contará un PIC tradicional al contar con varios periféricos aunque en pequeña cantidad, sus pines van desde 8 hasta 64 según el modelo y la programación no está condicionada por los recursos del PIC. Aquí se encuentran los PIC12F y los PIC16F.
- *Gama alta*: pueden operar hasta las 12MIPS, sus recursos son más robustos donde la característica principal es su expansibilidad o capacidad de ampliación por medio de sus pines a los cuales se pueden conectar distintos puertos o interfaces de comunicación, comprende de entre 18 hasta 100 pines. Esta gama la integra los PIC18F.
- *Gama dsPIC*: este tipo de PIC está especialmente diseñado para comportarse como un controlador digital de señales, capaz de manejar digitalmente las señales analógicas. Son aún más robustos que los PIC18F agregando el hardware necesario para manipular las señales, pueden alcanzar desde 16MIPS hasta 70MIPS, con la particularidad de que las operaciones de suma, resta, multiplicación y acumulación se realizan de manera rápida y eficiente. Pueden contar de 14 hasta 144 pines, por lo que son algo grandes y densos, aquí se encuentran los PIC24F, PIC24E, PIC24H, dsPIC30F, dsPIC33EP y dsPIC33F.

Usaremos un dsPIC dada su alta velocidad de operación, así será posible leer la señal de reloj, que en nuestro caso alcanza los $1.2[MHz]$, una frecuencia un poco alta aún para la gama alta. Al ser un PIC de este tipo, cuenta con elementos que, para nuestra aplicación, no necesitamos usar por lo que al elegir un dsPIC básico tendremos la mínima cantidad de elementos y conservaremos la velocidad de operación que caracteriza esta gama.

El dsPIC30F2011 es un dsPIC básico (AP-04) que posee las siguientes características:

Tabla 10: Características generales del dsPIC30F2011

<i>Voltaje de Polarización</i>	2.2[V] a 5.5[V]
<i>Velocidad de Operación</i>	30MIPS
<i>Tipo de Encapsulado</i>	PDIP, SOIC, QFN
<i>Número de Pines</i>	18
<i>Rango de Temperatura</i>	-40°C a 125°C
<i>Compiladores Compatibles</i>	C30, Ensamblador

Ocuparemos el dsPIC con encapsulado PDIP, que cual trabajará con un voltaje de 5[V] como el resto de los circuitos.

VI.II: Lenguaje de programación

Dentro del mundo de los microcontroladores dsPIC, existen dos principales lenguajes admitidos, con ventajas y desventajas las cuales son variables respecto al tipo de aplicación:

1. *Lenguaje C*
2. *Lenguaje Ensamblador*

VI.II.I: Lenguaje C

Este lenguaje posee las siguientes características:

- Es un lenguaje de alto nivel, lo que ayuda a facilitar el aprendizaje y desarrollo de diversos programas.
- Se pueden construir algoritmos matemáticos de manera rápida y sencilla.
- Posee la capacidad de incluir lenguaje ensamblador en los programas, que beneficia a la optimización de recursos, sobre todo en las gamas altas.
- Tiene la capacidad de crear rutinas preestablecidas o macros, que ayuda a la simplificación de los programas.

Al ser un lenguaje de alto nivel, no se tiene una buena optimización sobre los recursos del PIC, por lo que al compilar los programas llegan a resultar un poco extensos y pesados. Además, no existe un control sobre los tiempos de ejecución y los registros que se lleguen a usar.

En el caso del lenguaje C para dsPIC, podemos encontrar dos distintos compiladores, cada uno desarrollado por un fabricante diferente: C30 y CCS.

CCS está hecho por CCS, Inc. que ofrece la posibilidad de elegir la versión del compilador de acuerdo con la gama de PIC que implemente la aplicación: PCW IDE compatible con PIC10F a PIC16F, PCWH IDE con PIC10F a PIC18F y PCWHD IDE con PIC10F a dsPIC. CCS no es gratuito por lo que hay que pagar una licencia para poder usarlo.

C30 está desarrollado por Microchip Technology Inc., que solo es compatible con la gama de dsPIC. Está altamente optimizado para la generación de programas o código fuente (en gran parte por ser del mismo fabricante que los PIC) al tomar en cuenta los elementos que componen a cada dsPIC en particular. C30 tampoco es gratuito por lo que su uso está permitido por medio de una licencia.

En ambos casos, cuentan con un ensamblador, un enlazador y librerías de los dsPIC, Además existe una versión de prueba que está limitada en el tamaño de los programas o en el uso del hardware del dsPIC.

VI.II.II: Lenguaje ensamblador

Tiene las siguientes cualidades:

- Es el lenguaje de bajo nivel que usa un PIC, sin importar la gama en que se esté usando.
- Se pueden manejar los recursos del PIC para lograr una buena optimización en el uso.
- Se tiene un control sobre los tiempos de ejecución y manejo de los registros bit a bit.
- Puede manejar interrupciones simultáneas.
- Pueden crearse rutinas preestablecidas o macros, para la creación de programas reducidos.
- La compilación de los códigos fuente se hacen más rápido y eficiente, a diferencia del lenguaje C donde primero los programas se pasan a su equivalente a ensamblador para luego compilarlo.

Uno de los contratiempos que se presentan al usar el lenguaje ensamblador es la complejidad del formato del programa y de las instrucciones, que puede llegar a ser muy confusa al inicio del desarrollo de un programa fuente nuevo y pueden provocar que sean muy extensos.

El lenguaje ensamblador usado en los dsPIC es el ASM30, proporcionado por Microchip Technology Inc., el cual posee la gran ventaja de que es totalmente gratuito, sin ningún tipo de restricción en el desarrollo de los programas en cuanto al uso de hardware o en el tamaño de los mismos. El propio fabricante proporciona documentación como ejemplos y el set de instrucciones detalladas de uso, aunque

eso no quita la complejidad del lenguaje, que con la experiencia va disminuyéndose.

El lenguaje que se usará para la programación del dsPIC30F2011 será ASM30, en gran medida por la garantía de ser gratuito, el programa principal no será muy extenso o complicado, por lo que esta cualidad es muy determinante a la hora de compararlo con los productos del lenguaje C, que todos son de paga.

VI.III: Programa principal

El programa principal consiste en una serie de decisiones que se irán tomando de acuerdo con las señales que se necesitan para el funcionamiento de la tarjeta, éstas se proporcionarán en el momento indicado tomando como referencia el ritmo de los ciclos de reloj presentes en una conversión (figura 18).

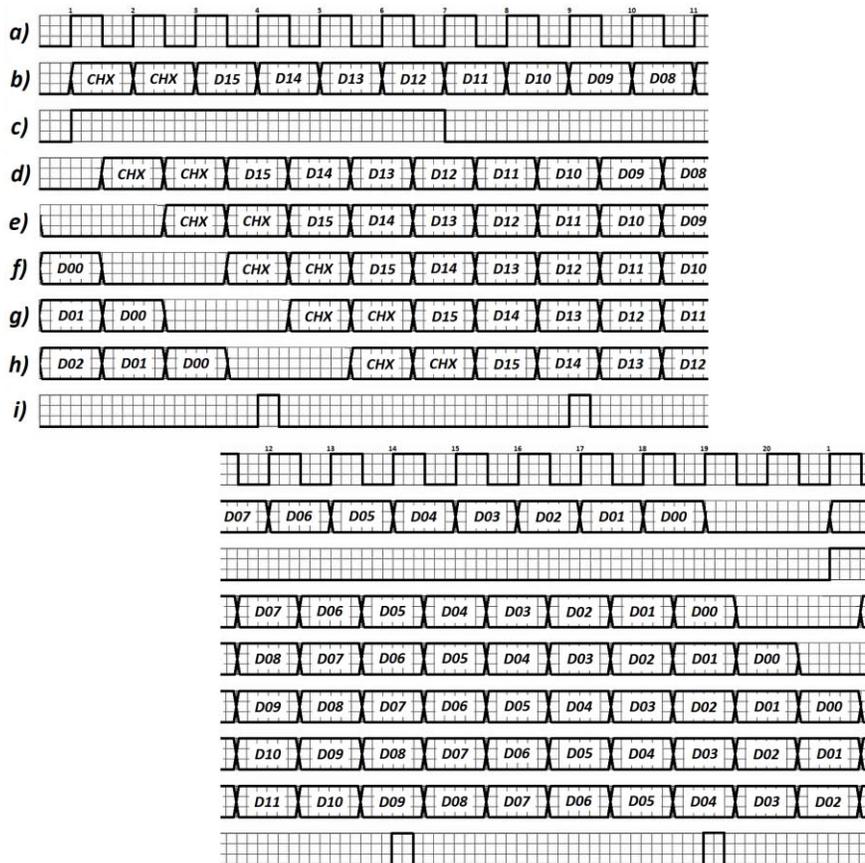


Figura 18: Diagrama de tiempos de las señales de la tarjeta al momento de una conversión, donde a) señal de reloj, b) salida serial del convertidor, c) señal de inicio de conversión, d) e) f) g) h) salida paralela del registro de corrimiento e i) señal indicadora de envío de datos.

Como se aprecia, los instantes importantes son en el ciclo 1 y 7, donde la señal de inicio de conversión actúa sobre el convertidor y el módulo USB. En los ciclos 4, 9, 14 y 19 se envían los bytes con la información de la conversión.

El programa principal llevará un conteo sobre el número de ciclos de reloj transcurridos y en los ciclos 1, 4, 7, 9, 14 y 19 tendrá que realizar el proceso correspondiente para que al momento del ciclo 20 se reinicie el conteo (figura 19).

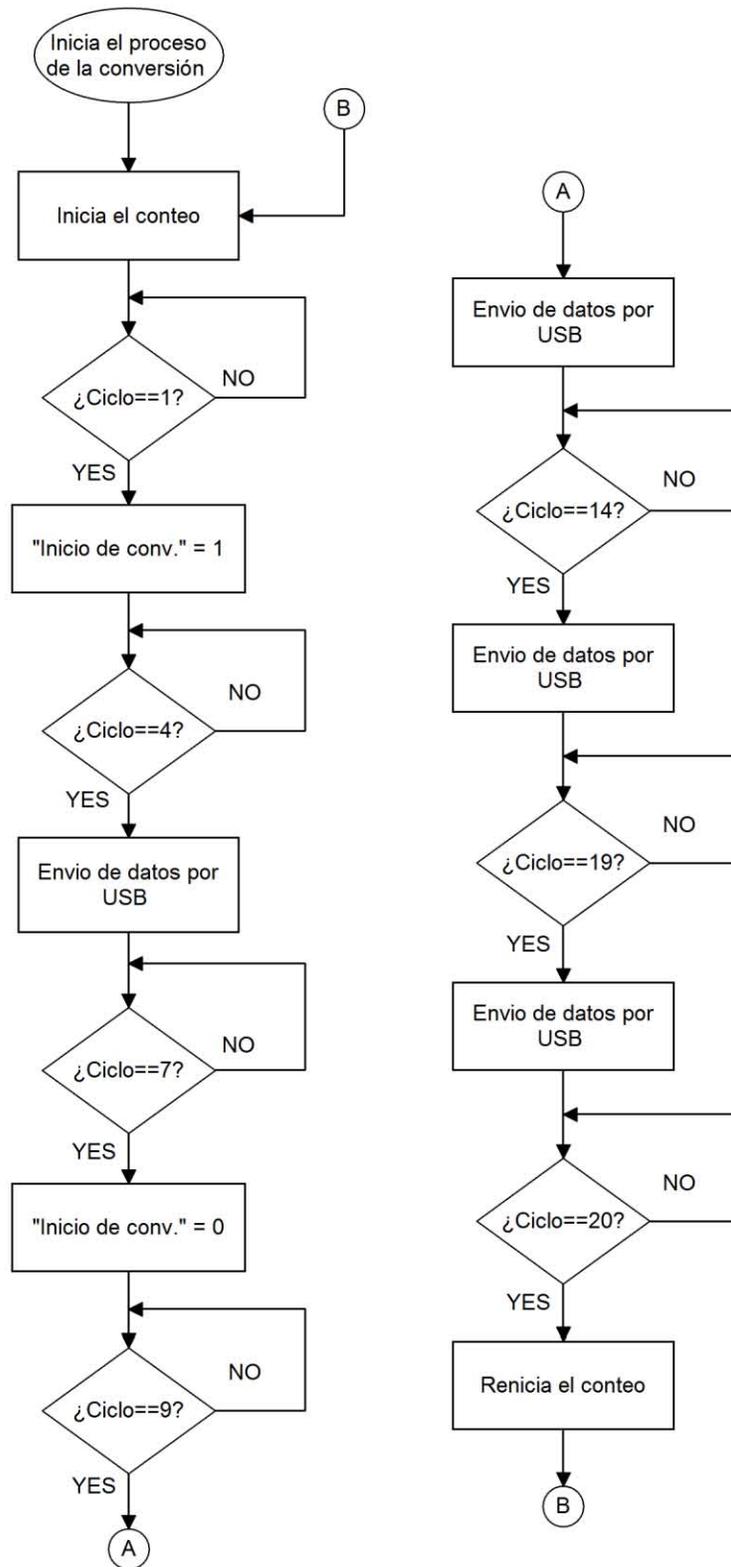


Figura 19: Diagrama de flujo del programa principal.

Al trabajar a una alta frecuencia, se tiene un menor tiempo para la ejecución del programa lo que puede provocar un retraso en las señales si éste llega a ser muy poco extenso. Para evitarlo, será necesario dividir y crear subprogramas que proporcionen cada una de las señales que se necesitan; la señal que va conectada al convertidor (*Señal "Inicio de conv."*) es provista por un dsPIC y la del módulo USB (*Señal "Enviar datos"*) es compuesta de la suma de las señales dadas por dos dsPIC (figura 20) y la sincronización entre éstos es llevada a cabo por un interruptor ON-OFF (*Señal "ON/OFF"*).

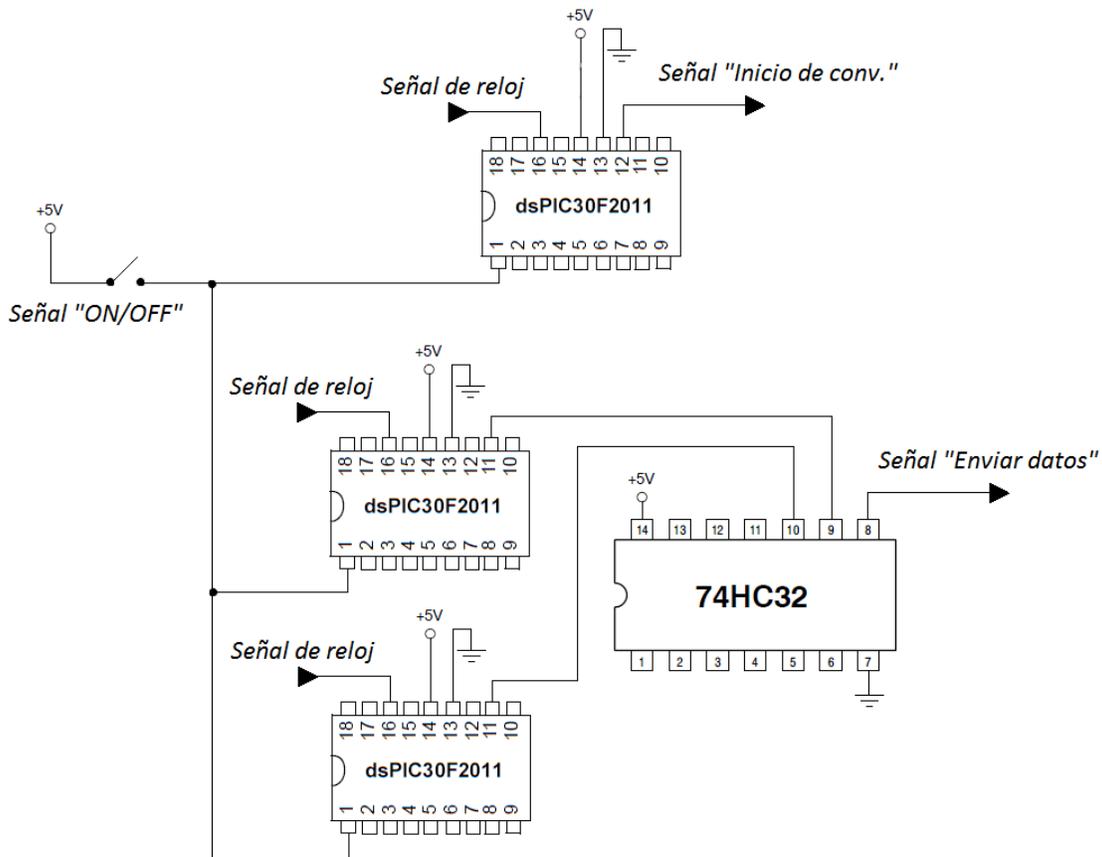


Figura 20: Diagrama de conexiones del conjunto de dsPIC30F2011.

VI.IV: Ambiente de programación y programador para el dispositivo

El software que proporciona Microchip Technology Inc. para la manipulación de cualquier gama de los microcontroladores PIC es MPLAB, un entorno de desarrollo integrado (IDE) que consiste en un editor de código, un depurador y un constructor de interfaz gráfica donde se pueden crear y optimizar los programas

fuerza, además permite la grabación de los diferentes tipos de PIC con la ayuda de un programador conectado a la PC.

MPLAB es de carácter gratuito, es compatible con los sistemas operativos Windows®, Mac OS®, Linux® y trae incluido el lenguaje ensamblador para los PIC de gama baja, media, alta y dsPIC con los archivos necesarios para su correcta implementación (figura 21). Usaremos MPLAB en su versión 8.92.

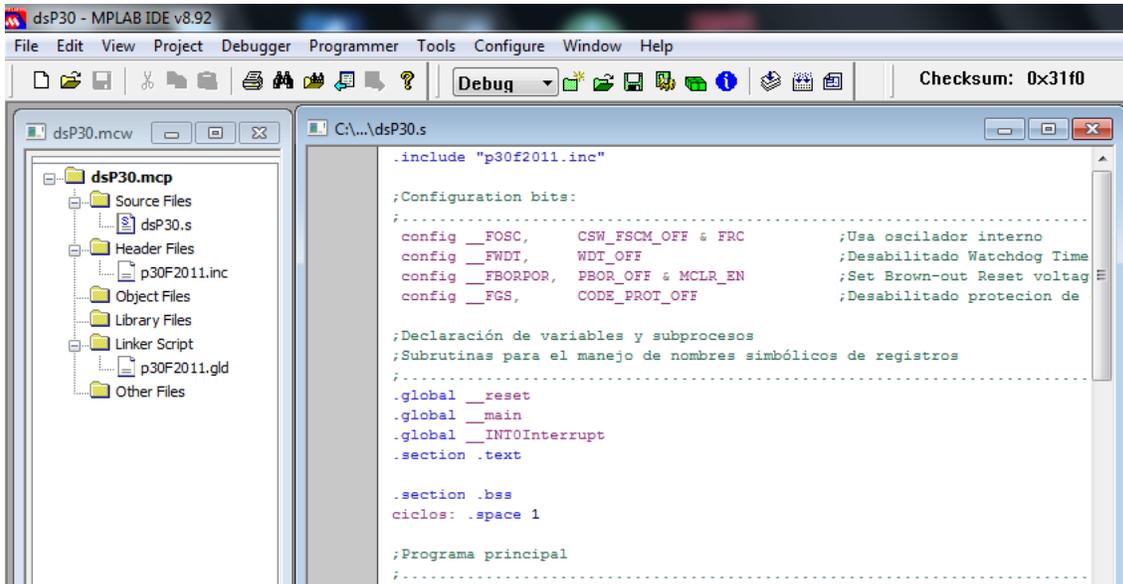


Figura 21: Ambiente de programación MPLAB

Para la creación de un nuevo programa fuente, es necesario indicarle al software el PIC (dsPIC30F2011 en nuestro caso) que usaremos, el lenguaje con el que realizaremos el código y el lugar en la PC donde se guardará. Una vez escrito el programa fuente, procedemos a vincular los archivos de memoria del PIC seleccionado, verificamos que no tenga error alguno de sintaxis y compilamos o depuramos.

En el proceso de la compilación se crea un archivo de extensión .hex que contiene la transliteración del programa principal a código hexadecimal, con el cual será programado el dsPIC.

Para la programación del dsPIC, existe una herramienta fabricada por Microchip Technology Inc., llamada MPLAB ICD 2 compatible con las gamas baja, media, alta y la mayoría de los dsPIC. Posee la facultad de programar y depurar en circuito, aunque no simultáneamente, se conecta a la PC a través de un puerto

USB 2.0 o un puerto serial RS-232y al PIC con un conector RJ-12. Tiene dos formas de alimentación, a través del mismo puerto o por medio de un regulador externo (figura 22).



Figura 22: Estructura general para el uso del MPLAB ICD 2.

Se controla por medio de funciones integradas a MPLAB, en donde los programas pueden ser cargados, ejecutados y modificados en tiempo real sin tener que desconectar o eliminar la alimentación. La programación y depuración se lleva a cabo a través de dos pines que posee todo tipo de PIC en donde va conectado el ICD 2, así se pueden realizar estas acciones sin necesidad de mover el dispositivo programable.

Capítulo VII: SOFTWARE DE PROCESAMIENTO Y VISUALIZACIÓN DE DATOS

El convertidor analógico-digital toma muestras de las señales de entrada en un intervalo de tiempo preestablecido, las cuales son proporcionadas en formato serie. Con la ayuda de un registro de corrimiento, éstas pasarán a paralelo para que el módulo USB pueda leerlas de manera correcta y así, enviarlas por un puerto USB a la PC. Este proceso está sincronizado con una señal de reloj y controlado por un dsPIC.

Las muestras son enviadas de acuerdo a un protocolo, garantizando que la información no sufra tipo alguno de alteración por algún fallo en la comunicación, con lo que se evita la pérdida de datos.

La PC toma una función muy importante, mediante el uso de un software de programación se reciben los datos, se leen y procesan de acuerdo con las peticiones del usuario y finalmente se visualiza la información deseada.

Las características que debe reunir el software de programación son: ser capaz de comportarse como un software de procesamiento y como un software de visualización.

VII.1: Software o etapa de procesamiento

Esta etapa se encarga de recibir, leer y convertir las muestras en información que puedan ser mostradas en la pantalla de la PC (figura 23).



Figura 23: Diagrama de bloques del software o etapa de procesamiento.

Una vez que el módulo USB ha enviado las muestras conforme al protocolo, son recibidas por la PC de forma continua para ser almacenadas en un buffer que se tiene en la memoria. Este paso es realizado de forma automática, si se excede el tamaño disponible de este buffer se sobrescribirá la información más antigua por la nueva, ocasionando pérdida de datos.

Entonces, para evitar esa situación, cada cierto tiempo establecido se extraen o recuperan todos los bytes almacenados en el buffer traspasándolos a un registro temporal en la memoria, dejándolo vacío para el siguiente llenado con nueva información. Al momento de este paso, también se obtiene la cantidad de bytes el cual servirá para determinar el número de veces que se repetirá el algoritmo que se encarga de discriminar y convertir las muestras en datos útiles.

Este algoritmo empieza a leer todo el registro temporal, comenzando con las muestras más antiguas, en busca de un byte que contenga en su bit más significativo con un "1" lógico. Esto indica que, conforme al protocolo diseñado, es el inicio de una secuencia de 4 bytes que almacena una conversión, junto con los tres bytes que le preceden, además de indicar a que canal pertenece.

De la secuencia de 4 bytes se extraen los bits con la información, al ser números en binario se procede a transformarlos a números decimales mediante un algoritmo de conversión de base 2 a base 10.

Con la información ahora en decimal, se agrupan en registros independientes conforme al canal al que pertenecen y por orden de antigüedad. El tamaño que poseen estos registros es lo suficiente como para albergar poco más de 4000 puntos sin importar a qué velocidad esté trabajando la tarjeta.

VII.II: Software o etapa de visualización

Esta etapa tiene como funciones la de ajustar datos y crear las señales a desplegar en la pantalla de la PC acorde con las peticiones dadas por el usuario (figura 24).

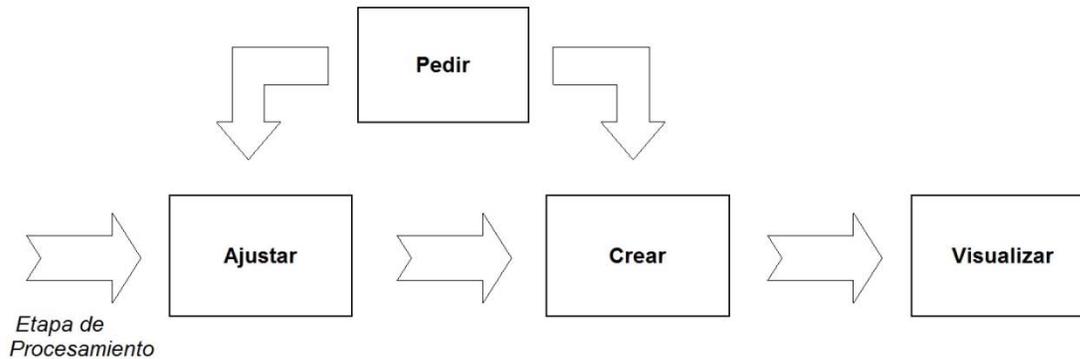


Figura 24: Diagrama de bloques del software o etapa de visualización

El ajuste de datos se regula por medio de la velocidad de muestreo, la cual es una petición del usuario. La tarjeta siempre trabajará con dos velocidades de operación: $f_{CLK1} = 100[kHz]$ y $f_{CLK2} = 1.2[MHz]$, para el caso de $f_{CLK1} = 100[kHz]$ las muestras que son enviadas a la PC no serán ajustadas por lo que solamente se transformarán a su equivalente en decimal y almacenadas directamente a los registros de los canales que correspondan.

Cuando la frecuencia de trabajo es $f_{CLK2} = 1.2[MHz]$ el ajuste se realizará mediante una discriminación, al ser constante la velocidad los datos se irán eliminando en múltiplos de números naturales; esto provoca una aparente reducción de la frecuencia de muestreo, la cual beneficiará al momento de tener señales a distinta frecuencia.

El tamaño de la reducción estará dada por el usuario a través de un control, para cada uno de los canales, independientes uno de otro. Así, el control tendrá disponibles varios valores de frecuencia de muestreo, en donde la opción más baja será igual a la velocidad de operación de la tarjeta $f_s = f_{CLK1} = 100[kHz]$ (figura 25).

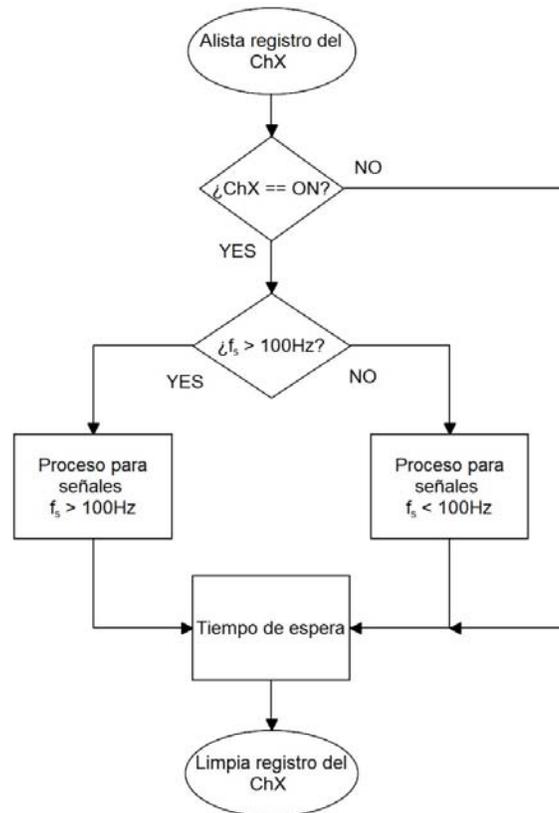


Figura 25: Diagrama de flujo para el proceso de las señales de acuerdo con la velocidad de muestreo.

Ya con los registros previamente ajustados, solo es cuestión de crear las señales que se mostrarán en la pantalla y para hacer esto, es necesario transformar estos registros a un nuevo tipo de dato, está conformado por dos arreglos de igual tamaño, uno está destinado para almacenar los datos de los registros que serán tratados como los puntos en el eje Y (ordenadas) y el otro es para guardar los puntos del eje X (abscisas), que marcan la separación entre las muestras, la cual es constante y está directamente relacionado con la velocidad de muestreo seleccionado.

Por medio de la ayuda de dos controles, podemos alterar las señales antes de ser visualizadas, escalando con respecto al voltaje (eje Y) y al tiempo (eje X). Estos controles están presentes en todos los canales, independientes entre ellos y son proporcionados por el usuario a través de la interfaz en la pantalla (figura 26).

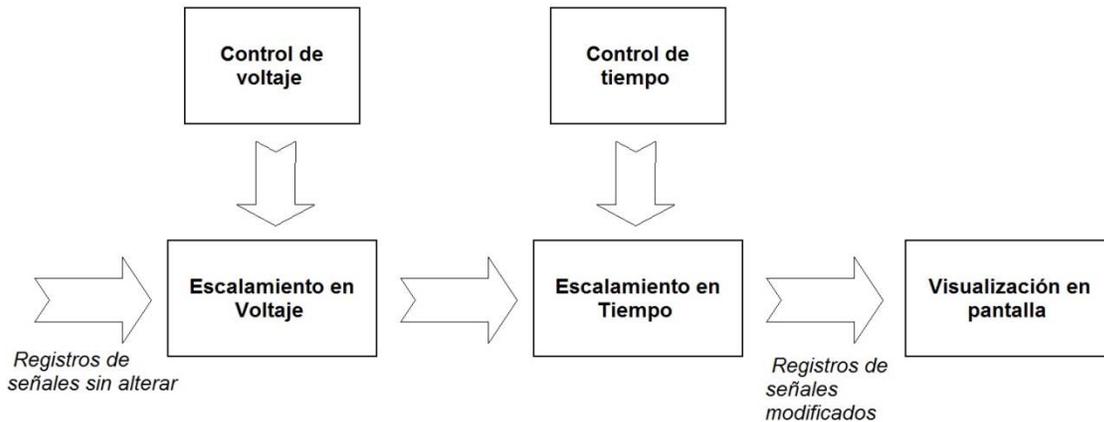


Figura 26: Diagrama de bloques para la creación de las señales de acuerdo con el usuario.

El voltaje se puede escalar de entre $2[mV]$ a $5[V]$ y el tiempo de $100[\mu s]$ a $2[s]$

VII.III: Opciones de software de programación

Para la creación de la interfaz de usuario, debemos recurrir a un software de programación con las características antes mencionadas. Existen varias opciones en el mercado, cada una diferenciada una de la otra por el tipo de lenguaje de programación que usa; las más utilizadas son:

- *Visual Basic*: su programación está enfocado por eventos en donde el flujo de los programas va encaminado por el usuario o por los mismos eventos que los componen. Comprende un entorno de desarrollo integrado con el cual se pueden crear interfaces de una manera sencilla, es un lenguaje fácil de aprender aunque requiere un gran conocimiento en el manejo de las librerías DLL, solo es compatible con Windows y los ejecutables creados llegan a ser lentos cuando se tratan de grandes aplicaciones.
- *Visual Basic .NET*: es la evolución de Visual Basic pero ahora adoptando la programación orientada a objetos con la ayuda de la framework .NET de Windows, lo que obliga a tener grandes conocimientos de ella. No hay mucha diferencia en la sintaxis entre Visual Basic y visual Basic .NET aunque varía en la metodología de programación y bibliotecas. Es algo robusto y multiplataforma, para la ejecución de los programas es obligatorio tener instalada la framework .NET. Consta de un entorno de desarrollo

integrado para el que se necesita tener una licencia pagada, aunque existe una versión gratuita muy limitada.

- *MATLAB*: es un software de cálculo numérico, de la empresa The Mathworks, que consta de un entorno de desarrollo integrado junto a un lenguaje exclusivo llamado lenguaje M, que posee una sintaxis muy parecida al lenguaje C pero con la variedad de que se enfoca en el uso de matrices. Se usa principalmente para procesamiento de señales y comunicaciones, procesamiento de imagen y vídeo, sistemas de control, entre otros. La gran limitante es que se necesita una licencia para su uso sin contar con una versión gratuita, además de ser un software de entorno muy cerrado.
- *LABVIEW*: es un software de diseño de sistemas, desarrollado por National Instruments Corporation, que está conformado por un entorno de desarrollo integrado donde se usa un lenguaje especial llamado lenguaje G, que se caracteriza por ser totalmente gráfico, donde las funciones están encapsuladas en bloques y el flujo del programa se va determinado en la forma en cómo se conectan éstos. Tiene compatibilidad con distintos tipos de software y hardware, la posibilidad de usar todas las funciones del software requiere una licencia, aunque existe una versión gratuita que limita el tamaño de los programas.

El lenguaje G posee la gran ventaja de ser fácil de usar, con o sin experiencia en la programación se pueden crear programas de una manera rápida y sencilla, aunque con la práctica se logra una gran optimización de los mismos. Además, al ser un lenguaje gráfico, las funciones y operaciones están encapsuladas en bloques por lo que para realizar un programa, éste se debe dibujar y no escribir, a manera de diagrama de bloques o como si fuera un circuito eléctrico, facilitando la lectura y comprensión, sin tener que aprender la sintaxis de numerosas funciones que, a menudo, resulta un poco complicado.

Por lo tanto, tomaremos como software de programación a LabVIEW 2011 Service Pack 1, en su versión gratuita que, para nuestros fines, será lo suficiente para la creación de la interfaz de usuario.

VII.IV: Interfaz de usuario y algoritmo de programación en LabVIEW

LabVIEW trabaja, en su entorno de desarrollo integrado, a través del manejo de dos ventanas que a su vez forman un instrumento virtual VI, denominado así al programa ejecutable resultante. Estas ventanas son:

- *Panel Frontal*: esta ventana es dedicada a la interfaz de usuario, en donde el usuario puede ver y controlar el flujo de la información cuando el programa se esté ejecutando. Esto se hace definiendo los controles (o entrada de datos, como botones, perillas, entre otros) y los indicadores (o salida de información, como una gráfica, un cuadro de texto, entre otros).
- *Diagrama de bloques*: aquí se introduce el programa descrito en lenguaje G, donde las funciones en forma de bloques se interconectan a partir de una nomenclatura establecida para cada caso.

Así, la interfaz de usuario (figura 27) se divide en tres partes:

- A) Aquí se encuentra el botón “STOP”, que sirve para detener el programa sin tener que usar el botón que tiene incluido LabVIEW, para evitar un posible conflicto en el puerto USB. Además de contar con un cuadro de texto que indica si la conexión con la tarjeta es exitosa, desplegando el modelo del módulo USB.
- B) En este lugar se encuentra el área de la pantalla donde se visualizan las señales introducidas a la tarjeta, se divide en cuadrantes para una mejor apreciación, en donde el eje “X” representa el tiempo y el eje “Y” es el voltaje. Cada gráfica tiene la opción de habilitar o deshabilitar el canal, independientemente uno de otro, además de contar con un color en particular para evitar cualquier confusión.
- C) Aquí se localiza una pestaña de opciones para cada canal, en donde tenemos un control para variar la escala de voltaje, para la posición vertical y para la escala de tiempo. También cuenta uno para la velocidad de muestreo junto al de trigger, donde se puede elegir el flanco y variar su valor.

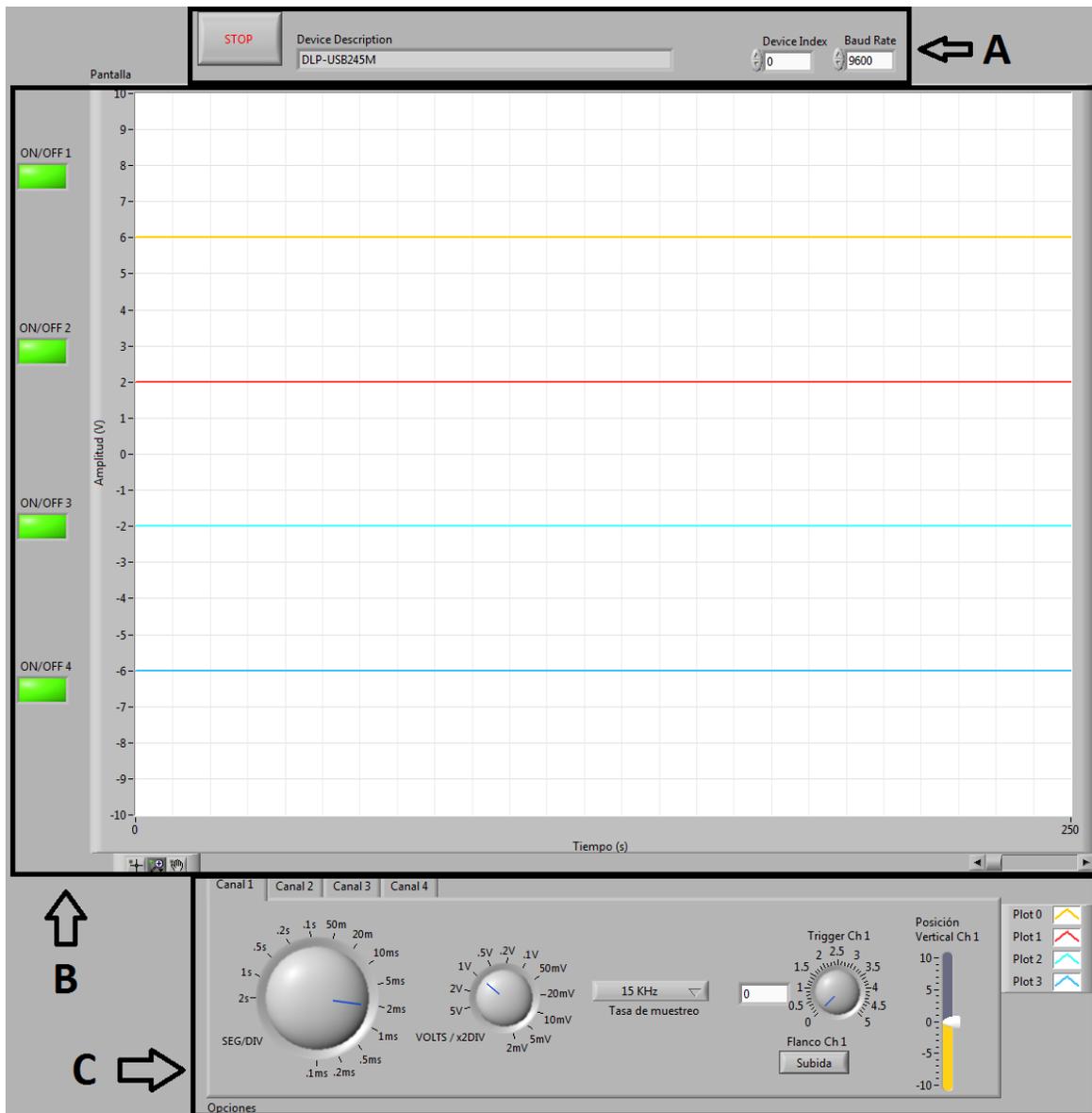


Figura 27: Interfaz de usuario creada en LabVIEW.

Para la programación, ya hemos establecido los bloques de la estructura principal, ahora en LabVIEW (figura 28) éstos están compuestos por pequeñas secciones de código que forman procesos preestablecidos con el fin de minimizar el programa lo más posible, ya que en algunos casos hay operaciones que se repiten en diversos lugares durante la ejecución lo que puede hacer el programa demasiado grande y algo complicado de leer y entender.

Así, el programa arranca estableciendo la conexión por el puerto USB a la tarjeta, si es exitosa se iniciará el envío de la información de manera continua e

ininterrumpida, por lo que el software se repite cada cierto tiempo de forma definida.

Una vez que se recopila la información del buffer USB se toma la decisión de la manera en cómo se tratará, con base en qué canales están habilitados y qué velocidad de muestreo tiene asignado. Recordemos que se han clasificado las señales de acuerdo con la frecuencia que poseen, con lo que la tarjeta cuenta con dos modos de operación para evitar el exceso de muestras según sea el caso, pero esta elección no es vista por la interfaz de usuario.

Entonces, para saber qué camino elegir el software irá revisando los canales habilitados de manera ascendente a la par de la velocidad de muestreo, cuando encuentre un primer canal encendido tomará la decisión, si la velocidad de muestreo es de $f_s = 100[Hz]$ la ejecución se tornará por el camino de las señales con $f_s < 100[Hz]$, en cualquier otro caso el flujo será por el trayecto de las señales con $f_s > 100[Hz]$, sin tomar en cuenta los otros canales habilitados. Cabe aclarar que no se pueden mezclar los dos tipos de señales, los cuatro canales deben tener la misma naturaleza, en caso de no cumplirse existirá un error en las gráficas de salida.

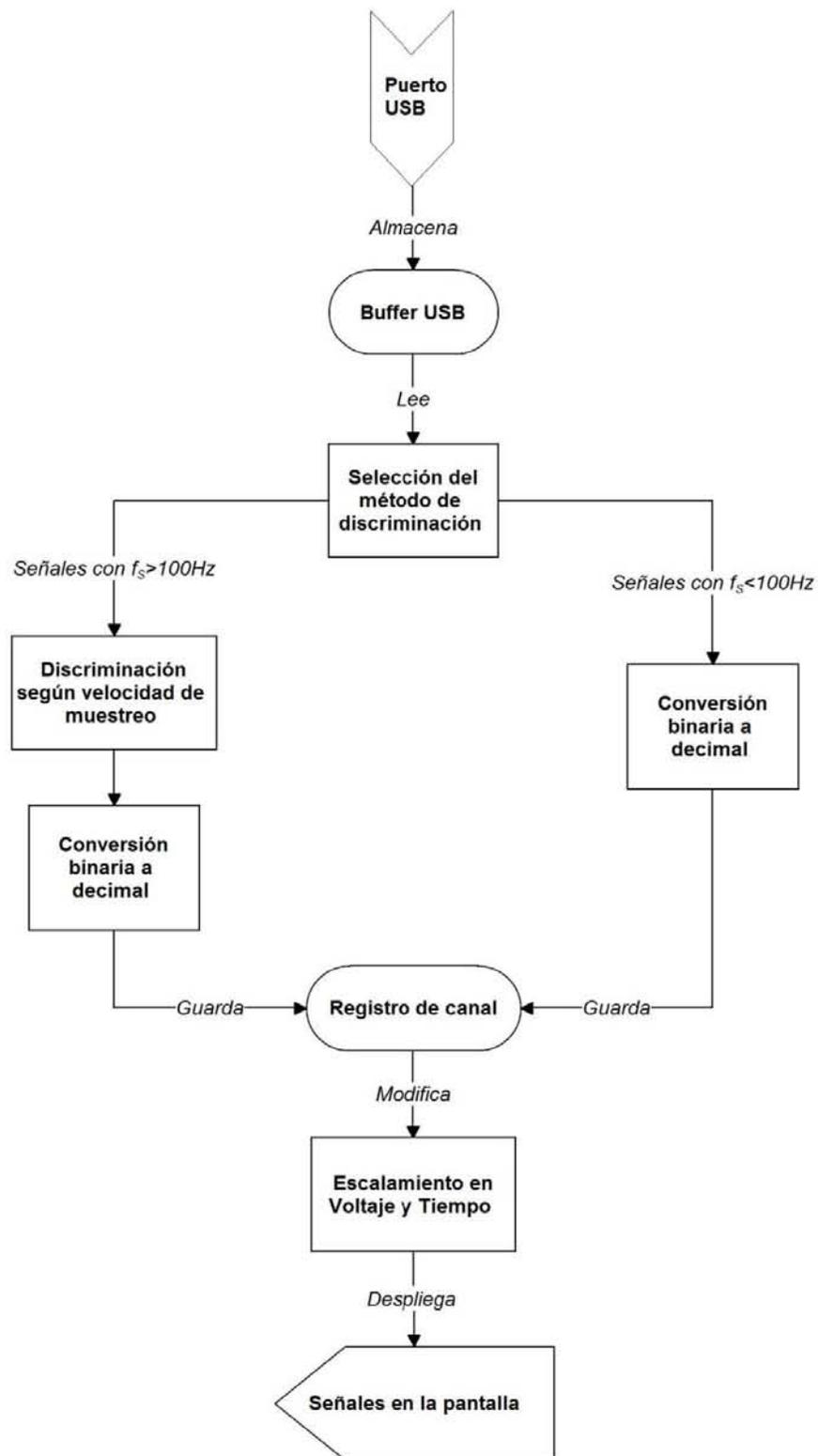


Figura 28: Estructura principal del programa de la interfaz de usuario en LabVIEW.

Capítulo VIII. LA TARJETA DIGITALIZADORA

La tarjeta digitalizadora queda, finalmente, constituida por un convertidor analógico-digital, un registro de corrimiento y un módulo USB (figura 29) por los cuales pasa el flujo de información principal, que contiene las muestras obtenidas de los canales de entrada. Estos son controlados por un conjunto de dsPIC que les proporcionan las señales necesarias para su funcionamiento; todo esto sincronizado a través de una señal de reloj que provee dos frecuencias distintas de operación, para adaptarse mejor a las señales de entrada.

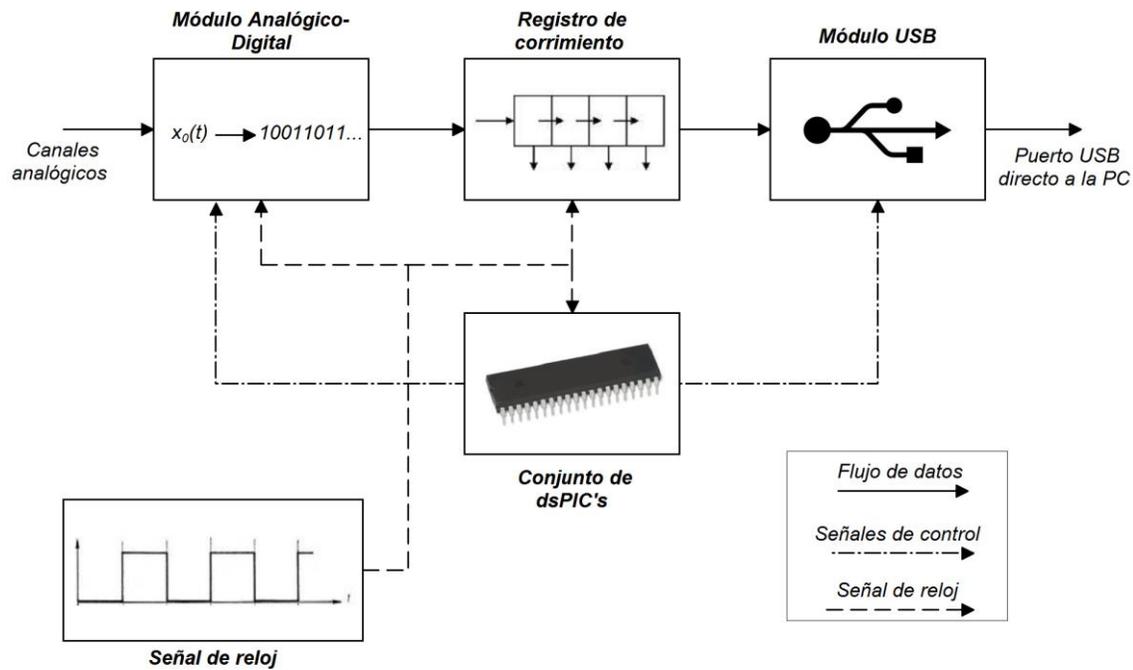


Figura 29: Estructura por bloques de la tarjeta digitalizadora.

A continuación se muestra la visualización de dos señales (figura 30, 31 y 32) con características diferentes introducidas en los cuatro canales de entrada, en el modo de operación donde $f_{CLK} = 1.2[MHz]$.



Figura 30: Visualización de una señal cardiaca de $V_p = 2[V]$ y $f_{in} = 100[Hz]$.

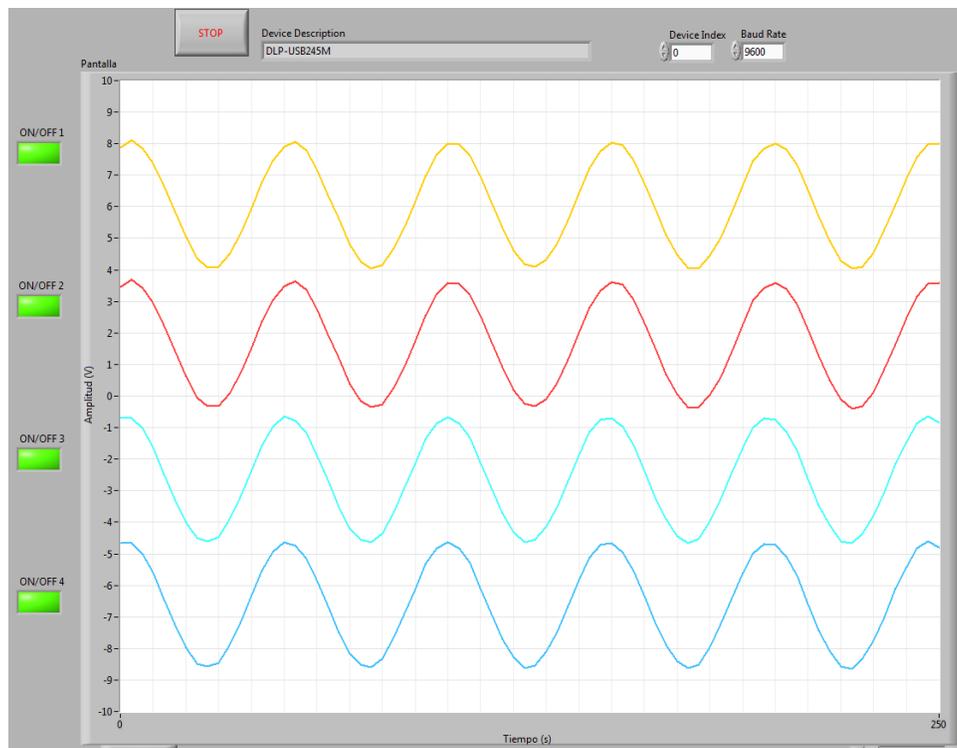


Figura 31: Visualización de una señal senoidal de $V_p = 2[V]$ y $f_{in} = 1[kHz]$.

Ahora se muestra la visualización de una señal cardiaca introducida en dos canales de entrada, en el modo de operación donde $f_{CLK} = 100[kHz]$.



Figura 32: Visualización de una señal cardiaca de $V_p = 2[V]$ y $f_{in} = 2[Hz]$.

Como se puede apreciar, las gráficas resultantes presentan una muy buena aproximación a las señales de entrada sin tener algún tipo de ruido en ellas, y esto es debido a las especificaciones de la tarjeta y al condicionamiento que éstas tengan para poder aprovechar al máximo las capacidades de la misma.

La PC donde se esté ejecutando el software de la tarjeta deberá contar con un buen hardware, entre más canales se usen y las señales de entrada tengan frecuencias más altas acercándose al límite permitido, se necesitará de una mayor velocidad de procesamiento para cumplir su tarea sin retrasos, de lo contrario la visualización empezará a sufrir un retardo por la cantidad de datos esperando en el buffer del puerto.

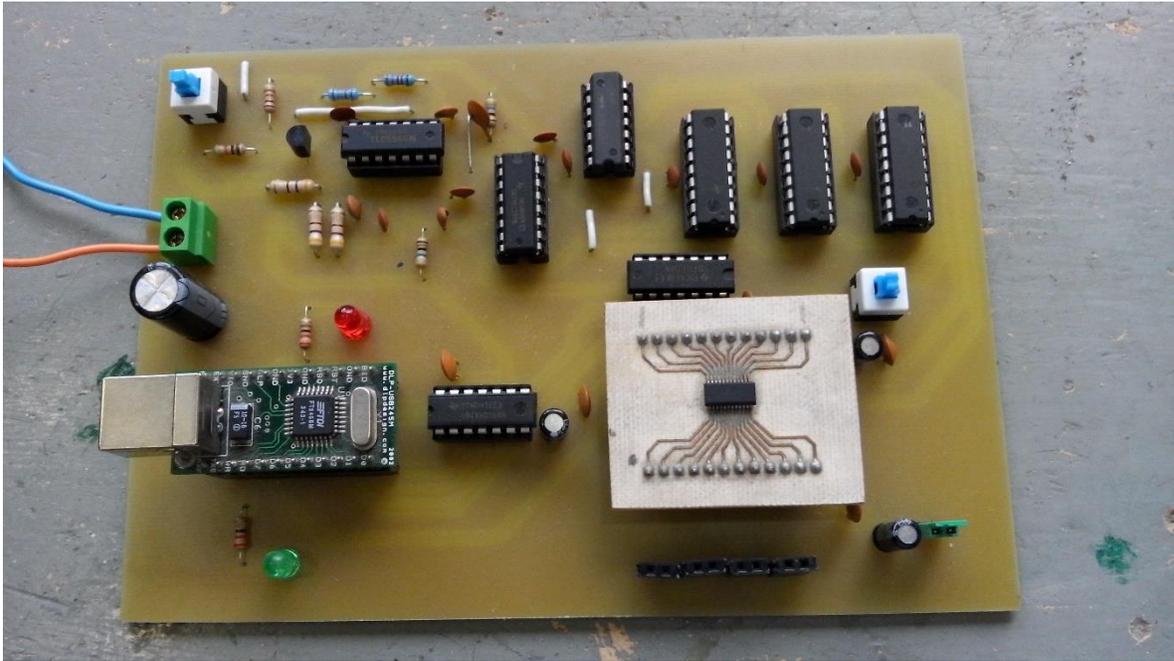


Figura 33: Circuito impreso de la tarjeta digitalizadora.

CONCLUSIONES

- Se ha diseñado e implementado una tarjeta digitalizadora enfocada para señales biomédicas, que posee especificaciones para hacer una buena representación sin el temor de perder datos, como lo muestran los resultados obtenidos.
- El sistema puede digitalizar señales que no sean de origen biomédico, siempre y cuando cumplan con los criterios de voltaje de entrada y ancho de banda establecidos para la tarjeta.
- Durante la operación, puede presentarse un error cuando se visualizan las señales de entrada en la pantalla de PC, debido a que los dsPIC no llegan a sincronizarse correctamente. Este error ocurre por el hecho de que cada uno de ellos maneja su propio reloj y al momento de capturar la señal de reloj del sistema, se origina la falla que alcanza un ciclo de reloj de desfase, suficiente para que los datos sean alterados al momento del envío y la PC los visualice de forma incorrecta.
- El envío de datos puede verse afectado por el hardware que constituye la PC donde se ejecute el software de procesamiento y visualización, si ésta tiene características limitadas los datos se perderán por el exceso de información que está recibiendo, sobrescribiendo bits antiguos por nuevos. Para evitar esto, la PC deberá ser lo bastante rápida, tratando de tener las mejoras características posibles.
- La máxima velocidad de procesamiento de la tarjeta está restringida por las características de cada uno de los circuitos integrados que la conforman, al requerirse que operen conjuntamente a una alta frecuencia salen a la luz las limitaciones en su funcionamiento afectando la sincronización, algo que no sucede en bajas frecuencias.
- Se pueden mejorar, como trabajo a futuro, varias características a fin de aumentar el rendimiento de la tarjeta, como la resolución o número de bits, por bloques o circuitos integrados que sean más rápidos o estén diseñados con otro tipo de arquitectura. Estas mejoras ocasionarían cambios desde pequeños hasta los que obliguen a cambiar la lógica de operación, además de su costo.

BIBLIOGRAFÍA

- Bronzino J., (2000). *Biomedical Engineering Handbook*. Estados Unidos: CRC Press LLC.
- Enderle J., Blanchard S., & Bronzino J., (2005). *Introduction to Biomedical Engineering* (2ª Ed.). Reino Unido: Elsevier Academic Press Inc.
- Ochoa Parra, F. (s.f.), *Procesamiento de Señales Biomédicas*. [En línea]. Ecuador, disponible en: <http://es.slideshare.net/felo8/procesamiento-de-seales-biomdicas> [Accesado el día 23 de marzo de 2014]
- Wakerly, J. (2000), *Digital Design: Principles & Practices* (3ª Ed.). Estados Unidos: Prentice Hall.
- Uyemura, J. (2000), *Diseño de Sistemas Digitales: Un enfoque integrado*. México: Thomson.
- Coughlin, R., Driscoll, F. (1993), *Amplificadores Operacionales y Circuitos Integrados Lineales* (4ª ed.). México: Prentice Hall.
- Boylestad, R., Nashelsky, L. (2009). *Electrónica: Teoría de Circuitos y Dispositivos Electrónicos* (10ª ed.). México: Prentice Hall.
- Paredes, A. (s.f.). *Diseño e Implementación de un Oxímetro de Pulso con Microcontrolador PIC18F4550*. Tesis de licenciatura. México, Facultad de Ingeniería, Universidad Nacional Autónoma de México.
- Oppenheim A. & Schafer R., (2009). *Tratamiento de señales en tiempo discreto* (3ª Ed.). España: Pearson S.A.

MESOGRAFÍA

- Platas, M. (s.f.). *Introducción a la Ingeniería Biomédica*. [En línea]. México, disponible en <http://gama.fime.uanl.mx/~mplatas/temas/intro.pdf> [Accesado el día 3 de marzo de 2013]
- Carrillo, C. (2014). *Electromiografía*. [En línea]. México, disponible en <https://prezi.com/d4bkapztvpa0/electromiografia/> [Accesado el día 8 de abril de 2013]
- Álvarez Fiallo, R., Santos Anzorandia, C., & Medina Herrera, E. (2006). *Desarrollo histórico y fundamentos teóricos de la electromiografía como medio diagnóstico*. [En línea]. Cuba, disponible en http://www.bvs.sld.cu/revistas/mil/vol35_4_06/mil08406.htm [Accesado el día 17 de octubre de 2013]
- Pellegrino, F. (s.f.), *Evaluación de la Función Neuromuscular: Electroneuromiografía*. [En línea]. Argentina, disponible en <http://neurolatinvet.com/publi/neuromus.htm> [Accesado el día 5 de mayo de 2014]
- Texas Instrument (2007), *ADS8361 Datasheet*. [En línea]. Estados Unidos, disponible en <http://www.ti.com/lit/ds/sbas230e/sbas230e.pdf> [Accesado el día 9 de febrero de 2014]
- Microchip Technology (2010), *dsPIC30f2011 Datasheet*. [En línea]. Estados Unidos, disponible en <http://ww1.microchip.com/downloads/en/DeviceDoc/70139G.pdf> [Accesado el día 15 de febrero de 2014]
- DLP Design (2007). *DLP-USB245M Datasheet*. [En línea]. Estados Unidos, disponible en <http://www.dlpdesign.com/usb/dlp-usb245mv15.pdf> [Accesado el día 22 de marzo de 2014]
- Texas Instrument (1997). *TLC556 Datasheet*. [En línea]. Estados Unidos, disponible en <http://www.ti.com/lit/ds/symlink/tlc556.pdf> [Accesado el día 4 de junio de 2014]
- Texas Instrument (1997). *SN74HC164N Datasheet*. [En línea]. Estados Unidos, disponible en http://pdf.datasheetcatalog.com/datasheets/120/491561_DS.pdf [Accesado el día 19 de junio de 2014]
- Texas Instrument (2003). *SN74HC157N Datasheet*. [En línea]. Estados Unidos, disponible en <http://www.ti.com/lit/ds/symlink/cd74hct157.pdf> [Accesado el día 18 de julio de 2014]
- Philips Semiconductors (1990). *74HC32 Datasheet*. [En línea]. Estados Unidos, disponible en

http://pdf.datasheetcatalog.com/datasheet/philips/74HC_HCT32_CNV_2.pdf
[Accesado el día 2 de agosto de 2014]

- Salinas, P. (2008). *Manual de Uso del Programa de Diseño de Circuitos y Simulación: Proteus Layout Editor*. [En línea]. Colombia, disponible en <http://www.tecnica1lomas.com.ar/tutoriales/manual-proteus.pdf> [Accesado el día 12 de septiembre de 2014]
- Cubillo, J. (s.f.). *Manual de Proteus Ares*. [En línea]. Ecuador, disponible en <http://web.fpsantacatalina.com/documentos/departamentos/electronica/manualares/MANUALARES.pdf> [Accesado el día 18 de septiembre de 2014]
- Torres, A. (2012). *Introducción al Procesamiento Digital de Señales con dsPIC y el Compilador C30*. [En línea]. México, disponible en https://www.editorialdigitaltec.com/materialadicional/P107_Antonio_IntroduccionProcesamientoDigital.cap1.pdf [Accesado el día 3 de octubre de 2014]
- Microchip Technology (2005). *Getting Started with dsPIC30F Digital Signal Controllers*. [En línea]. Estados Unidos, disponible en <http://ww1.microchip.com/downloads/en/DeviceDoc/70151a.pdf> [Accesado el día 2 de diciembre de 2014]
- Microchip Technology (2013). *Assembler/Linker/Librarian User's Guide*. [En línea]. Estados Unidos, disponible en <http://ww1.microchip.com/downloads/en/DeviceDoc/33014L.pdf> [Accesado el día 8 de noviembre de 2014]
- Microchip Technology (2013). *MPLAB® IDE User's Guide*. [En línea]. Estados Unidos, disponible en: http://ww1.microchip.com/downloads/en/DeviceDoc/MPLAB_User_Guide_5_1519c.pdf [Accesado el día 15 de noviembre de 2014]
- National Instruments Corporation (2014). *Aprenda LabVIEW*. [En línea]. Estados Unidos, disponible en <http://www.ni.com/academic/students/learn-labview/esa/> [Accesado el día 20 de octubre de 2014]
- Hans-Petter, H (2014). *An Introduction to LabVIEW*. [En línea]. Noruega, disponible en <http://home.hit.no/~hansha/documents/labview/training/Introduction%20to%20LabVIEW/Introduction%20to%20LabVIEW.pdf> [Accesado el día 23 de octubre de 2014]

APÉNDICE

- AP-01.



ADS8361

SBAS230E – AUGUST 2002 – REVISED AUGUST 2007

Dual, 500kSPS, 16-Bit, 2 + 2 Channel, Simultaneous Sampling ANALOG-TO-DIGITAL CONVERTER

FEATURES

- 2 SIMULTANEOUS 16-BIT DACs
- 4 FULLY DIFFERENTIAL INPUT CHANNELS
- 2 μ s THROUGHPUT PER CHANNEL
- 4 μ s TOTAL THROUGHPUT FOR FOUR CHANNELS
- LOW POWER: 150mW
- INTERNAL REFERENCE
- FLEXIBLE SERIAL INTERFACE
- 16-BIT UPGRADE TO THE 12-BIT ADS7861
- PIN COMPATIBLE WITH THE ADS7861
- OPERATING TEMPERATURE RANGE: –40°C to +125°C

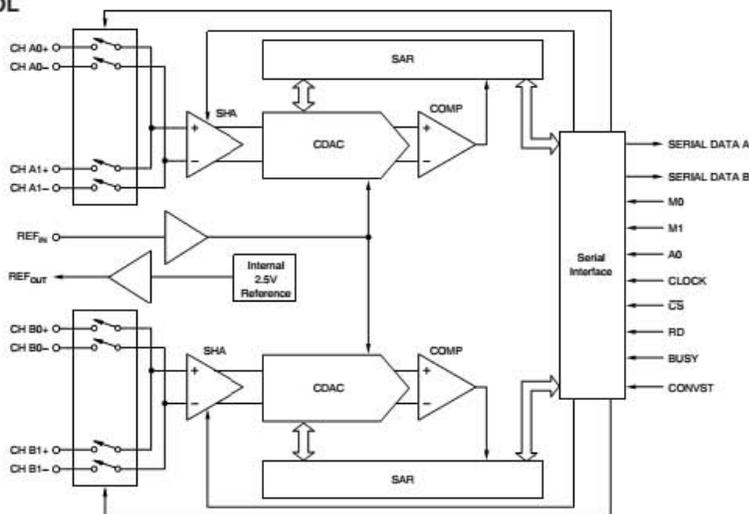
DESCRIPTION

The ADS8361 is a dual, 16-bit, 500kSPS, Analog-to-Digital (A/D) converter with four fully differential input channels grouped into two pairs for high-speed, simultaneous signal acquisition. Inputs to the sample-and-hold amplifiers are fully differential and are maintained differentially to the input of the A/D converter. This provides excellent common-mode rejection of 80dB at 50kHz, which is important in high-noise environments.

The ADS8361 offers a high-speed, dual serial interface and control inputs to minimize software overhead. The output data for each channel is available as a 16-bit word. The ADS8361 is offered in SSOP-24 and QFN-32 (5x5) packages and is fully specified over the –40°C to +125°C operating range.

APPLICATIONS

- MOTOR CONTROL
- MULTI-AXIS POSITIONING SYSTEMS
- 3-PHASE POWER CONTROL



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

All trademarks are the property of their respective owners.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.



Copyright © 2002-2007, Texas Instruments Incorporated

▪ AP-02.

TLC556, TLC556Y
DUAL LinCMOS™ TIMERS

SLFS047B – FEBRUARY 1984 – REVISED SEPTEMBER 1997

- Very Low Power Consumption . . . 2 mW Typ at $V_{DD} = 5\text{ V}$
- Capable of Operation in Astable Mode
- CMOS Output Capable of Swinging Rail to Rail
- High Output-Current Capability
Sink 100 mA Typ
Source 10 mA Typ
- Output Fully Compatible With CMOS, TTL, and MOS
- Low Supply Current Reduces Spikes During Output Transitions
- Single-Supply Operation From 2 V to 15 V
- Functionally interchangeable With the NE556; Has Same Pinout

description

The TLC556 series are monolithic timing circuits fabricated using the TI LinCMOS™ process, which provides full compatibility with CMOS, TTL, and MOS logic and operates at frequencies up to 2 MHz. Accurate time delays and oscillations are possible with smaller, less-expensive timing capacitors than the NE556 because of the high input impedance. Power consumption is low across the full range of power supply voltages.

Like the NE556, the TLC556 has a trigger level approximately one-third of the supply voltage and a threshold level approximately two-thirds of the supply voltage. These levels can be altered by use of the control voltage terminal. When the trigger input falls below the trigger level, the flip-flop is set and the output goes high. If the trigger input is above the threshold level and the threshold input is above the threshold level, the flip-flop is reset and the output is low. The reset input can override all other inputs and can be used to initiate a new timing cycle. If the reset input is low, the flip-flop is reset and the output is low. Whenever the output is low, a low-impedance path is provided between the discharge terminal and ground.

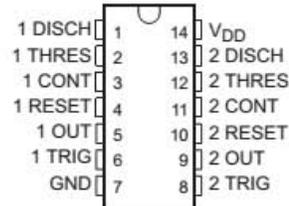
While the CMOS output is capable of sinking over 100 mA and sourcing over 10 mA, the TLC556 exhibits greatly reduced supply-current spikes during output transitions. This minimizes the need for the large decoupling capacitors required by the NE556.

These devices have internal electrostatic-discharge (ESD) protection circuits that prevent catastrophic failures at voltages up to 2000 V as tested under MIL-STD-883C, Method 3015. However, care should be exercised in handling these devices, as exposure to ESD may result in degradation of the device parametric performance.

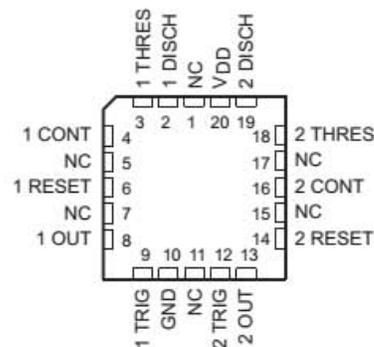
All unused inputs should be tied to an appropriate logic level to prevent false triggering.

The TLC556C is characterized for operation from 0°C to 70°C. The TLC556I is characterized for operation from –40°C to 85°C. The TLC556M is characterized for operation over the full military temperature range of –55°C to 125°C.

D, J, OR N PACKAGE
(TOP VIEW)



FK PACKAGE
(TOP VIEW)



NC—No internal connection

LinCMOS is a trademark of Texas Instruments Incorporated.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 1997, Texas Instruments Incorporated



- AP-03.



DLP-USB245M-G USB to FIFO Parallel Interface Module

The DLP-USB245M-G is the 2nd generation, lead free version of DLP Design's USB adapter. This device adds extra functionality to its DLP-USB1 predecessor with a reduced component count and a new low price.

DLP-USB245M FEATURES

- Send / Receive Data over USB at up to 1 M Bytes / sec
- 384 byte FIFO Transmit buffer / 128 byte FIFO receive buffer for high data throughput
- Simple interface to CPU or MCU bus
- No in-depth knowledge of USB required as all USB Protocol is handled automatically within the module
- FTDI's Virtual COM port drivers eliminate the need for USB driver development in most cases.
- Integrated 3.3v Regulator – No External Regulator Required
- Integrated Power-On-Reset circuit
- Integrated 6MHz – 48Mhz clock multiplier PLL
- USB Bulk or Isocronous data transfer modes
- 4.4v - 5.25v Single Supply taken directly from the USB port
- UHCI / OHCI / EHCI host controller compatible
- USB 1.1 and USB 2.0 compatible
- USB VID, PID, Serial Number and Product Description
- Strings stored in on-board EEPROM.
- EEPROM programmable on-board via USB
- VIRTUAL COM PORT (VCP) DRIVERS for
 - Windows 98 and Windows 98 SE
 - Windows 2000 / ME / XP
 - Windows CE **
 - MAC OS-8 and OS-9
 - MAC OS-X
 - Linux 2.40 and greater
- D2XX (USB Direct Drivers + DLL S/W Interface)
 - Windows 98 and Windows 98 SE
 - Windows 2000 / ME / XP

[** = In planning or under development]

APPLICATION AREAS

- Rapid Prototype Development
- USB ISDN and ADSL Modems
- High Speed USB PDA Interface
- USB Interface for Digital Cameras
- USB Interface for MP3 players
- High Speed USB Instrumentation
- USB Smart Card Readers
- Set Top Box (S.T.B.) PC-USB interface
- USB Hardware Modems
- USB Wireless Modems
- USB Bar Code Readers

- AP-04.



MICROCHIP

dsPIC30F2011/2012/3012/3013

High-Performance, 16-bit Digital Signal Controllers

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the "dsPIC30F Family Reference Manual" (DS70046). For more information on the device instruction set and programming, refer to the "16-bit MCU and DSC Programmer's Reference Manual" (DS70157).

High-Performance Modified RISC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set architecture
- Flexible addressing modes
- 83 base instructions
- 24-bit wide instructions, 16-bit wide data path
- Up to 24 Kbytes on-chip Flash program space
- Up to 2 Kbytes of on-chip data RAM
- Up to 1 Kbytes of nonvolatile data EEPROM
- 16 x 16-bit working register array
- Up to 30 MIPS operation:
 - DC to 40 MHz external clock input
 - 4 MHz - 10 MHz oscillator input with PLL active (4x, 8x, 16x)
- Up to 21 interrupt sources:
 - 8 user-selectable priority levels
 - 3 external interrupt sources
 - 4 processor trap sources

DSP Features:

- Dual data fetch
- Modulo and Bit-Reversed modes
- Two 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single-cycle hardware fractional/integer multiplier
- All DSP instructions are single cycle
 - Multiply-Accumulate (MAC) operation
- Single-cycle ± 16 shift

Peripheral Features:

- High-current sink/source I/O pins: 25 mA/25 mA
- Three 16-bit timers/counters; optionally pair up 16-bit timers into 32-bit timer modules
- 16-bit Capture input functions
- 16-bit Compare/PWM output functions
- 3-wire SPI modules (supports four Frame modes)
- I²C™ module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- Up to two addressable UART modules with FIFO buffers

Analog Features:

- 12-bit Analog-to-Digital Converter (ADC) with:
 - 200 ksp/s conversion rate
 - Up to 10 input channels
 - Conversion available during Sleep and Idle
- Programmable Low-Voltage Detection (PLVD)
- Programmable Brown-out Reset

Special Microcontroller Features:

- Enhanced Flash program memory:
 - 10,000 erase/write cycle (min.) for industrial temperature range, 100K (typical)
- Data EEPROM memory:
 - 100,000 erase/write cycle (min.) for industrial temperature range, 1M (typical)
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Flexible Watchdog Timer (WDT) with on-chip low-power RC oscillator for reliable operation
- Fail-Safe Clock Monitor operation:
 - Detects clock failure and switches to on-chip low-power RC oscillator
- Programmable code protection
- In-Circuit Serial Programming™ (ICSP™)
- Selectable Power Management modes:
 - Sleep, Idle and Alternate Clock modes

CMOS Technology:

- Low-power, high-speed Flash technology
- Wide operating voltage range (2.5V to 5.5V)
- Industrial and Extended temperature ranges
- Low-power consumption