



INSTITUTO POLITECNICO NACIONAL

**Escuela Superior
de Física y Matemáticas**



**Sistema digitalizador para resolver en el
dominio del tiempo pulsos transitorios del
orden de nanosegundos**

T E S I S

QUE PARA OBTENER EL TITULO
DE LICENCIADO EN FÍSICA Y MATEMÁTICAS

P R E S E N T A

Daniel Jiménez Olarte

**Director De Tesis
M. en C. Edgard Moreno García**

México, D.F.

Febrero 2008

Agradecimientos

Al profesor Edgard Moreno García por todos los conocimientos que me transmitió sobre electrónica y más importante, sobre la vida.

A los profesores Alfredo Godínez y Porfirio Reyes por el apoyo que me brindaron durante todo el tiempo que estuve en el laboratorio de electrónica.

A mi familia, a mi amor, a mis amigos, por apoyarme incondicionalmente

A Dios porque nunca me ha abandonado.

ÍNDICE

RESUMEN	1
----------------------	----------

CAPÍTULO 1

GENERALIDADES

1.1 Antecedentes	2
1.2 Objetivo	2
1.3 Justificación	2
1.4 Aportaciones	3
1.4 Contenido	3

CAPÍTULO 2

MEDICIÓN DE PULSOS

2.1 Aplicación en fluorometría resuelta en tiempo	4
2.2 Conversión analógica digital	6
2.3 Proceso de muestreo	9
2.4 Técnicas de muestreo	11
2.5 Propuesta de diseño para el sistema digitalizador	12
Referencias	14

CAPÍTULO 3

SISTEMA DIGITALIZADOR

3.1 Descripción del procesador digital dsPIC33FJ256GP710	16
3.2 Descripción del Convertidor A/D MAX12528 y kit	21
3.3 Base de Tiempos	24
3.4 Prototipo del digitalizador	26
Referencias	29

CAPÍTULO 4

PROGRAMAS

4.1	Herramientas de desarrollo.....	30
4.2	Programa en el dsPIC (ensamblador).....	33
4.4	Programa de control en LabVIEW.....	35
	Referencias.....	38

CAPÍTULO 5

MEDICIONES Y RESULTADOS

5.1	Mediciones y Resultados.....	39
	Referencias.....	43

CAPÍTULO 6

CONCLUSIONES Y TRABAJOS FUTUROS

6.1	Conclusiones.....	44
6.2	Trabajos futuros.....	45

APÉNDICES

- A. Diagrama esquemático y circuito impreso de la base de tiempos.
- B. Programa en Lenguaje Ensamblador grabado en el dsPIC33FJ256GP710.

RESUMEN

Se presenta la construcción de un sistema digitalizador capaz de resolver pulsos en el dominio del tiempo del orden de nanosegundos. Este sistema está compuesto por un convertidor de 750 MHz de ancho de banda, un controlador digital de señales dsPIC, una base de tiempos basada en un oscilador de 5MHz y tres líneas de retardo y una computadora. Las señales son adquiridas mediante la técnica de muestreo en tiempo equivalente, se aplica además un método de supresión de ruido para mejorar la calidad de la señal reconstruida. El sistema digitalizador puede medir señales analógicas de hasta $\pm 1.024V$ y es capaz de generar un barrido en tiempo de hasta 114.9ns con pasos de una resolución de 0.15ns. Los valores adquiridos por el digitalizador son transmitidos a la computadora por medio del puerto serie y son mostrados en una gráfica en tiempo real.

El dsPIC ejecuta el programa de adquisición de datos (que se encuentra residente en su memoria de programa tipo flash) bajo el control de un programa en la PC desarrollado en el lenguaje gráfico G de LabVIEW 7.0. El programa de control envía comandos al dsPIC y éste envía los datos codificados resultantes de las conversiones A/D. El programa en LabVIEW realiza un promedio de las mediciones para disminuir la relación señal a ruido y presenta en un indicador gráfico el trazo en el tiempo del pulso medido. Después de adquirir todas las mediciones las guarda en archivos de texto para su posible procesamiento y análisis posterior.

El digitalizador desarrollado resuelve pulsos con un tiempo de subida de 4ns. Lo anterior fue verificado experimentalmente al medir pulsos de prueba generados por un circuito monoestable. Los valores obtenidos concuerdan con los reportados en la hoja de datos del monoestable.

CAPÍTULO 1

GENERALIDADES

1.1 ANTECEDENTES

Debido al gran avance en los últimos años en el desarrollo de sistemas de transmisión de alta velocidad y en la generación de pulsos láser muy cortos, muchas aplicaciones requieren hoy día de detectores con alta resolución temporal, o equivalentemente con un ancho de banda muy grande. Las aplicaciones de las mediciones en el dominio del tiempo del orden de picosegundos continúan incrementándose debido a la expansión de métodos digitales y técnicas de pulsos en muchas áreas científicas. Un proceso automático para realizar mediciones con alta resolución temporal involucra el uso de un sistema digital programable o digitalizador. Este puede ser un instrumento específico para la aplicación, o bien, una computadora combinada con un procesador de señales digitales (DSP) o con un osciloscopio, por ejemplo. Particularmente estamos interesados en resolver en el dominio del tiempo pulsos de fluorescencia inducida por láser (LIF) del orden de nanosegundos. Actualmente la espectroscopia de fluorescencia esta siendo ampliamente utilizada para propósitos biomédicos y análisis clínicos entre otras aplicaciones y constituye un creciente y vasto campo de investigación. La espectrofluorometría in vivo esta emergiendo como una potente técnica en investigaciones biomédicas desde el estudio de estructuras celulares y de tejidos hasta funciones biológicas y la temprana detección de cáncer.

1.2 OBJETIVO

El objetivo que se planteo al inicio de este trabajo fue desarrollar un sistema digitalizador, con circuitos electrónicos, interfaces y programas, asociados a una PC; con la finalidad de medir pulsos de voltaje del orden de nanosegundos por medio de la técnica de muestreo en tiempo equivalente secuencial. El sistema a desarrollar debe presentar las mediciones adquiridas por el digitalizador en una grafica en tiempo real y los valores deben ser guardados en archivos para su futuro procesamiento.

1.3 JUSTIFICACIÓN

La relevancia del trabajo que se presenta consiste en crear la infraestructura tecnológica que mejore y facilite el trabajo de los investigadores en sus procesos experimentales. Dentro del IPN, el autoequipamiento tiene una importancia fundamental en sus programas institucionales, con el trabajo que se presenta se disminuyen costos en

equipamiento y se contribuye al desarrollo tecnológico que la institución en particular requiere. También se fundamentan las bases para que el trabajo desarrollado pueda aplicarse en procesos similares en nuestro país, en relación a este tipo de problemas.

1.4 APORTACIONES

Como producto del trabajo se presenta un sistema digitalizador capaz de medir señales de hasta $\pm 1.024\text{V}$ en un intervalo de 114.9ns con una resolución de 0.15ns . El sistema que se presenta tiene una interface a la PC de tipo serie y los datos son mostrados en una grafica y guardados en archivos de texto lo que permite que este sistema se pueda utilizar como un osciloscopio.

1.5 CONTENIDO

En este primer capítulo se dan los antecedentes, el objetivo, la justificación y las aportaciones del trabajo que se presenta. En el capítulo 2 se detalla el principio de operación del convertidor analógico digital y los tipos existentes, se describen además las principales técnicas de muestreo y se presenta la propuesta del sistema digitalizador. En el capítulo 3 se profundiza en el diseño del digitalizador y se describe el funcionamiento y recursos del controlador digital de señales dsPIC33FJ256GP710, el convertidor analógico digital MAX12528 y una base de tiempos desarrollada en el laboratorio de electrónica de la ESFM, que son los dispositivos que conforman al digitalizador.

En el capítulo 4 se presentan los algoritmos, las principales funciones y las subrutinas desarrollados para el dsPIC y el convertidor A/D para que en conjunto lleven a cabo el muestreo en tiempo equivalente secuencial. Finalmente en el capítulo 5 se presentan las mediciones obtenidas al medir pulsos transitorios de corta duración con el digitalizador desarrollado y se discuten los resultados. Se presentan además las conclusiones y las propuestas para trabajos futuros.

CAPÍTULO 2

MEDICIÓN DE PULSOS

En este capítulo se describe una aplicación de la medición de pulsos en el área de fluorimetría resuelta en tiempo. Se estudian los principios y características de convertidores analógico digital y los diferentes tipos de conversión. También se estudia el proceso de muestreo y las distintas técnicas de muestreo, haciendo hincapié en el muestreo en tiempo equivalente. Por último se propone el diseño de un sistema digitalizador de alta resolución y ancho de banda grande, conformado por un controlador digital de señales dsPIC, un convertidor analógico digital con ancho de banda grande y una base de tiempos.

2.1 APLICACIÓN EN FLUOROMETRÍA RESUELTA EN TIEMPO

Debido al gran avance en los últimos años en el desarrollo de sistemas de transmisión de alta velocidad y en la generación de pulsos láser muy cortos, muchas aplicaciones requieren hoy día de detectores con alta resolución temporal, o equivalentemente con un ancho de banda muy grande. Las aplicaciones de las mediciones en el dominio del tiempo del orden de picosegundos continúan incrementándose debido a la expansión de métodos digitales y técnicas de pulsos en muchas áreas científicas. En experimentos de química, física y biología se utilizan sistemas láser ultra rápidos de hasta femtosegundos como herramientas de diagnóstico en el estudio de procesos físicos fundamentales [2.1, 2.2]. Un proceso automático para realizar mediciones con alta resolución temporal involucra el uso de un sistema digital programable o digitalizador. Este puede ser un instrumento específico para la aplicación, o bien, una computadora combinada con un procesador de señales digitales (DSP) o con un osciloscopio, por ejemplo.

Particularmente estamos interesados en resolver en el dominio del tiempo pulsos de fluorescencia inducida por láser (LIF). El fenómeno de fluorescencia ocurre cuando un espécimen es excitado con luz; una parte de ésta luz de excitación es absorbida por el espécimen y rápidamente reemitida con diferentes propiedades. La fluorescencia emitida contiene información acerca del espécimen bajo estudio. La espectrofluorimetría está relacionada con la medición de espectros producidos por emisión [2.3]. Actualmente la espectroscopia de fluorescencia está siendo ampliamente utilizada para propósitos biomédicos y análisis clínicos entre otras aplicaciones y constituye un creciente y vasto campo de investigación. La espectrofluorimetría in vivo está emergiendo como una potente técnica en investigaciones biomédicas desde el estudio de estructuras celulares y de tejidos hasta funciones biológicas y la temprana detección de cáncer [2.4].

Las mediciones de fluorescencia pueden ser clasificadas en dos tipos: mediciones en estado estacionario y mediciones resueltas en tiempo. Las mediciones en estado estacionario se realizan con una iluminación constante y son las más comunes. La muestra es iluminada con un haz continuo de luz y la intensidad de la radiación fluorescente emitida es registrada en función de la longitud de onda (espectro de fluorescencia). Las mediciones resueltas en tiempo se utilizan para medir decaimientos de la intensidad fluorescente en función del tiempo. En este tipo de mediciones la muestra bajo estudio es expuesta a un pulso de luz, donde el ancho del pulso debe ser más corto que el tiempo de decaimiento de la fluorescencia emitida por la muestra. Estas mediciones dependientes del tiempo resuelven el decaimiento de la intensidad fluorescente en términos de su tiempo de vida, por lo tanto proporcionan información adicional acerca de la dinámica intra e intermolecular del fluoróforo (compuesto fluorescente de la muestra) bajo estudio [2.3, 2.5, 2.6]. El decaimiento de la intensidad se registra con un sistema de detección de alta velocidad que permite medir tiempos de vida del orden de nanosegundos.

La figura 2.1 muestra el ejemplo de un arreglo instrumental para resolver en el tiempo tiempos de vida de fluorescencia. Un pulso láser excita a la muestra a través de un sistema óptico dentro de una cámara que contiene también a la muestra. La luz fluorescente emitida por la muestra es colectada a la entrada del monocromador, el cuál selecciona una apropiada longitud de onda de la luz fluorescente y su señal óptica de salida es detectada con un tubo fotomultiplicador (PMT). La señal eléctrica producida por el PMT es amplificada y registrada con un sistema digitalizador de alta velocidad. Una fracción del pulso de excitación se detecta con un fotodiodo de avalancha (APD) y también se amplifica y se registra en el digitalizador.

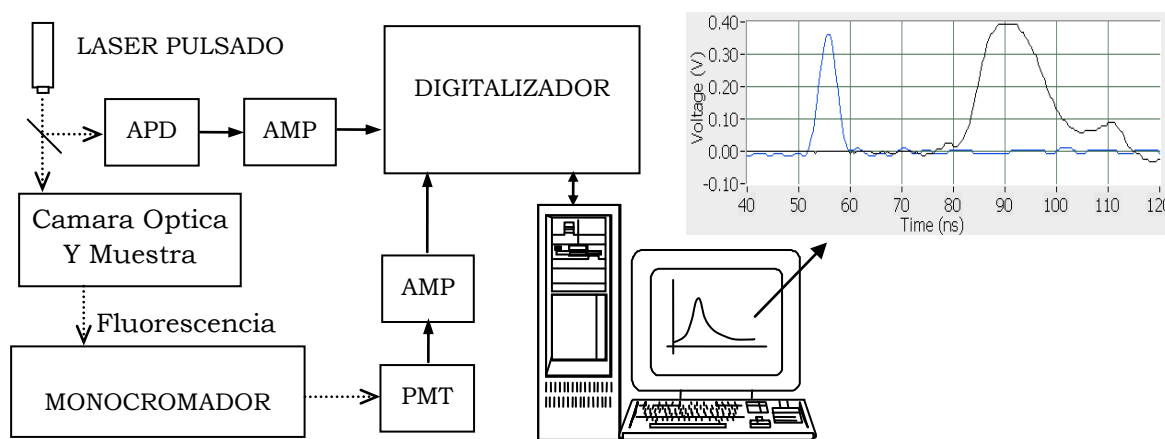


Figura 2.1. Ejemplo de un arreglo instrumental para resolver tiempos de vida de fluorescencia. El ejemplo presenta las mediciones de un pulso láser de excitación y del pulso de fluorescencia producido por papel blanco.

El programa que se ejecuta en la computadora controla la operación del monocromador y del digitalizador. La computadora envía comandos hacia el digitalizador y recibe datos para graficar las señales medidas. El sistema digitalizador de alta velocidad debe ser

capaz de resolver decaimientos de fluorescencia con tiempos de vida del orden de nanosegundos. En la figura 2.1 se muestra el ejemplo de mediciones de un pulso láser de excitación y del pulso de fluorescencia producido por papel blanco [2.7]. En este trabajo se presenta el desarrollo de un sistema digitalizador para medir pulsos con anchos del orden de nanosegundos.

2.2 CONVERSIÓN ANALÓGICA DIGITAL

Los procesos que se presentan en la naturaleza tienen cambios continuos en el tiempo los cuales podemos medir a través de sensores. La salida de estos es una señal analógica la cual podemos definir como una forma de onda que toma un continuo de valores en cualquier tiempo dentro de un intervalo de tiempos. El procesamiento de la información que contienen estas señales analógicas no resulta sencillo cuando se realiza por medio de circuitos electrónicos de tipo analógico (lineal). Almacenar, manipular, comparar, calcular, ó recuperar información, etc., son acciones de procesamiento que son más fáciles de realizar por medio de circuitos electrónicos digitales, los cuáles operan con patrones binarios. Cuando se utilizan sistemas digitales programables como microcontroladores ó computadoras personales (PCs), el procesamiento de la información se facilita aún más. En la actualidad el gran avance en las tecnologías de fabricación de circuitos integrados digitales ha permitido que sus costos se reduzcan, de tal manera que la información puede ser procesada en forma digital por medio de microcontroladores y computadoras personales a un relativo bajo costo [2.8].

Para procesar la información de origen analógico con técnicas digitales, como es nuestro caso para la medición de pulsos analógicos de voltaje, es necesario convertir el formato entre ambos dominios. Los Convertidores Analógico/Digital (CAD) posibilitan que la información del mundo analógico sea reconocida por las computadoras y los Convertidores Digital/Analógico (CDA) permiten que las computadoras generen señales que sean reconocidas por los procesos analógicos. El propósito de un convertidor A/D (CAD) es producir una palabra ó código digital de salida, que represente la magnitud de un voltaje analógico de entrada. Las principales características de un convertidor A/D se describen a continuación [2.8]:

- *RESOLUCIÓN*. Se define como el cambio del voltaje de entrada necesario para cambiar la salida del CAD entre dos códigos adyacentes, es decir, para cambiar la salida en 1 LSB (bit menos significativo). La resolución esta determinada por el número n de bits que forman la palabra binaria de salida, esto es: $Res=1/2^n$, es decir una parte en 2^n y también se puede representar en % a máxima escala ó bien, en términos de LSB.
- *EXACTITUD* (Accuracy). Es la diferencia del voltaje de entrada que debe idealmente producir un código de salida y el voltaje de entrada real que produce ese código. Normalmente la exactitud ó error se especifica en términos de LSBs, pero también puede expresarse como un porcentaje a máxima escala, ó bien en términos de voltaje.

- *TIEMPO DE CONVERSIÓN* es el tiempo que tarda el convertidor en producir un código binario de salida válido para un voltaje de entrada aplicado. También se puede especificar como el número de conversiones realizadas en un segundo.
- *POSIBLES CÓDIGOS DE SALIDA*. Binario Unipolar y Bipolar, BCD multiplexado, Decimal en 7 segmentos y Gray, entre otros.
- *MÉTODOS DE CONVERSIÓN*. Existen muy diversos métodos para realizar la conversión A/D, pero los 3 más comunes son los siguientes: Doble Rampa (integración), Aproximaciones Sucesivas y Paralelo ó Flash.

En las siguientes líneas se describen brevemente éstos tres métodos de conversión [2.9].

a. *Doble rampa*. Este tipo de convertidores comparan el tiempo necesario para cargar el condensador C (figura 2.2), mediante la corriente suministrada por el voltaje a medir (V_{in}), con el tiempo necesario para descargarlo hasta el voltaje de referencia (V_{ref}) mediante una corriente conocida. La lógica de control conmuta la entrada del integrador entre la tensión a medir y la fuente de referencia. Este convertidor es utilizado como el principal sistema de un voltímetro digital, su tiempo de conversión es grande de aproximadamente 300ms (3 conversiones por segundo). Requieren de un voltaje de referencia muy estable y de una señal de reloj. La frecuencia de oscilación recomendada por los fabricantes es de 3 lecturas/segundo para que se tenga alto rechazo a ruidos de 60Hz. Debido al uso de este tipo de convertidores la resolución frecuentemente se expresa en términos del número de dígitos de lectura y es ampliamente utilizado en instrumentos con lectura digital de 3 ½ dígitos.

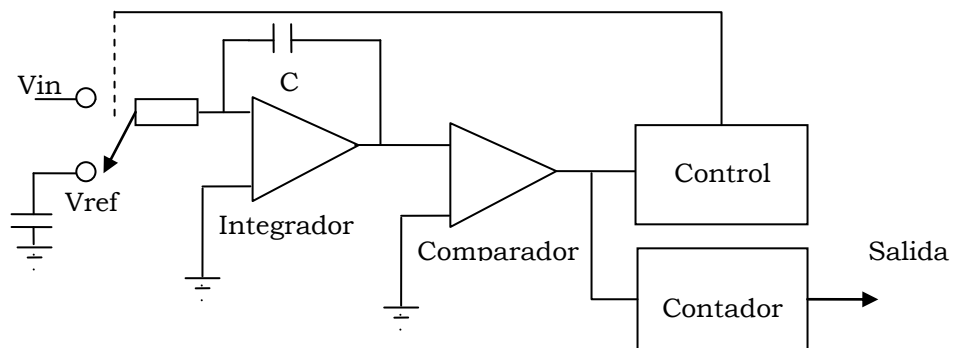


Figura 2.2. Convertidor analógico digital de doble rampa.

b. *Aproximaciones sucesivas*. El proceso de conversión para este tipo de convertidores se basa en la realización de comparaciones sucesivas de manera descendente o ascendente, hasta que se encuentra la combinación que iguala la tensión entregada por un convertidor D/A y la de entrada (figura 2.3). Es el más común en convertidores integrados cuando la exactitud requerida no es

determinante, ya que su diseño supone un equilibrio entre velocidad y complejidad. Su salida es un código binario.

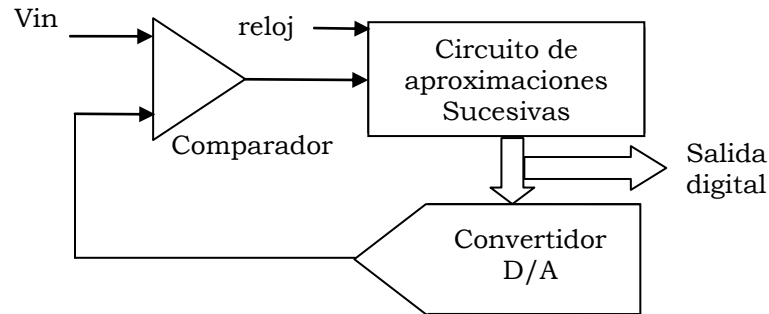


Figura 2.3. Diagrama de bloques de un convertidor analógico digital de aproximaciones sucesivas.

c. *Paralelo o flash.* Este convertidor es muy rápido, ya que la conversión se realiza de forma simultánea y casi instantánea. La figura 2.4 muestra el esquema interno de un CAD formado con comparadores en paralelo de 2 bits. Un divisor de voltaje coloca distintos voltajes de referencia en la entrada de cada uno de los comparadores, si la entrada de voltaje (V_{in}) en un comparador es más grande que su voltaje de referencia, la salida del comparador será un uno lógico, por lo tanto las salidas de los comparadores dan una representación digital del nivel de voltaje en la entrada. Sus elementos esenciales son la cadena de comparadores analógicos de alta velocidad y un codificador de prioridad.

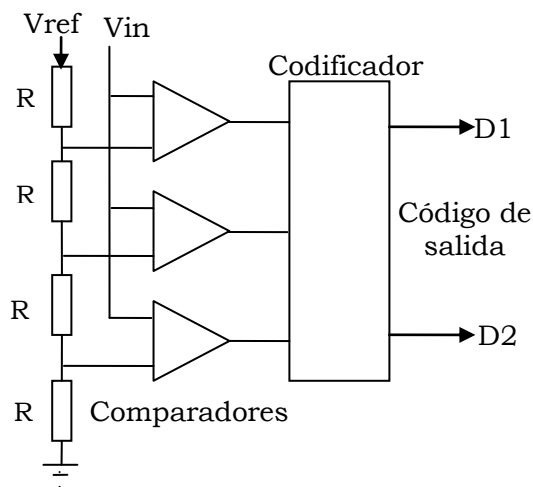


Figura 2.4. Convertidor analógico digital de 2 bits de resolución de tipo flash. El código de salida de los comparadores no es código binario estándar pero puede ser convertido con muy poca dificultad. Aquí V_{ref} representa la escala total de convertidor.

El tiempo de conversión esta determinado por la velocidad de los comparadores y el codificador. La principal desventaja de este tipo de convertidores es el numero de comparadores necesarios para producir un resultado con una resolución razonable, para producir una conversión de N bits de resolución se necesitan $(2^N - 1)$ comparadores.

2.3 PROCESO DE MUESTREO

De una manera más formal la operación de un convertidor A/D se puede representar como el diagrama de bloques de la figura 2.5 [2.10]. El primer componente es el muestreador el cual toma muestras periódicas de la amplitud de la señal analógica, la salida del muestreador es una señal discreta en el tiempo pero es continua en su amplitud ya que los valores muestreados pueden tener el mismo intervalo continuo de valores de la señal de entrada $x(t)$. Un concepto importante es la frecuencia de muestreo (f_m), que se define como el número de muestras por unidad de tiempo que se toman de una señal continua para producir una señal discreta, durante el proceso necesario para convertirla de analógica en digital [2.10].

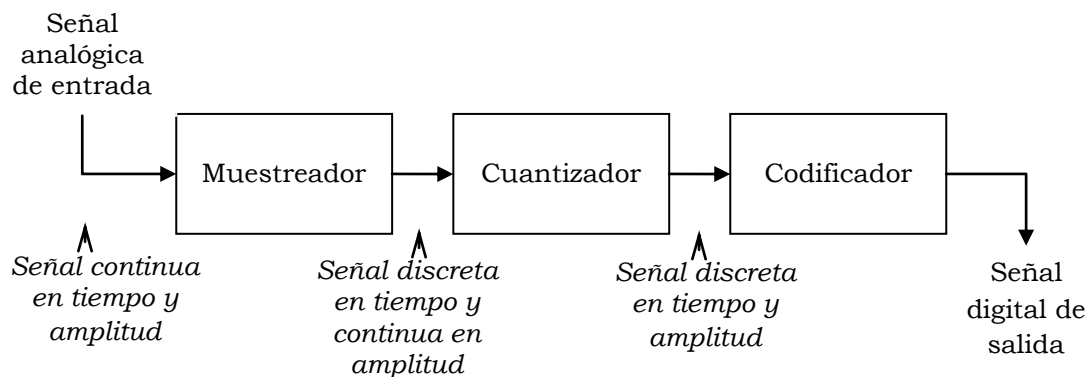


Figura 2.5. Convertidor analógico digital.

El segundo componente del convertidor A/D es un cuantizador, el cual convierte una sucesión de muestras de amplitud continua en una sucesión de valores discretos preestablecidos de acuerdo a un código. Durante el proceso de cuantificación se mide el nivel de tensión de cada una de las muestras obtenidas en el proceso de muestreo y se les atribuye un valor finito (discreto) de amplitud, seleccionado por aproximación dentro de un margen de niveles previamente fijado. Los valores preestablecidos para ajustar la cuantificación se eligen en función de la propia resolución que utilice el código empleado durante la codificación. Si el nivel obtenido no coincide exactamente con ninguno, se toma como valor el inferior más próximo. En este momento, la señal analógica se convierte en una señal digital, ya que los valores que están preestablecidos, son finitos. La codificación consiste en la traducción de los valores de tensión eléctrica analógicos que ya han sido cuantificados (ponderados) al sistema binario, mediante códigos preestablecidos.

Para muestrear una señal continua en el tiempo $x(t)$ se necesita representar $x(t)$ en un numero discreto de puntos, $t=nT$, donde T es el periodo del muestreo, es decir el tiempo entre la captura de cada muestra y n es un entero de la posición en tiempo de cada muestra. Este proceso se ilustra en la figura 2.6 en donde se muestra un conjunto de muestras de una señal continua en el tiempo y un interruptor muestreador, el cual puede ser visto como un modelo inicial de un dispositivo de muestreo. Para extraer las muestras de $x(t)$, el interruptor se cierra en un tiempo muy corto cada T segundos. Esto permite a las muestras tener un valor de $x(t)$ cada vez que el interruptor es cerrado y un valor de cero cuando éste está abierto.

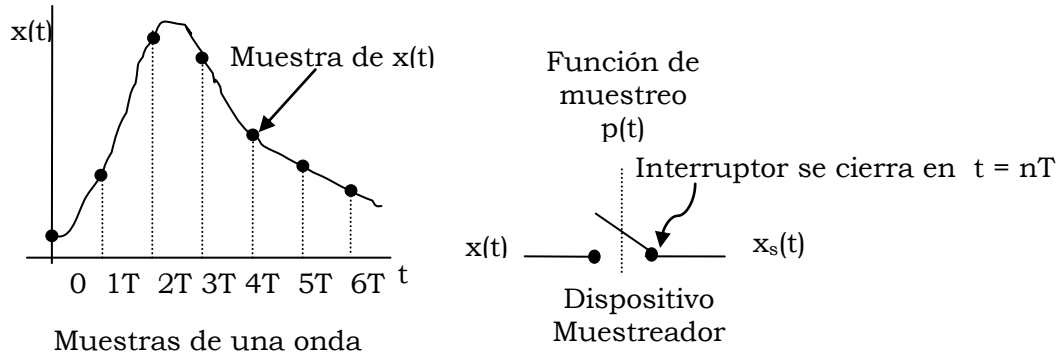


Figura 2.6. Proceso de muestreo.

El conjunto de muestras tomadas de $x(t)$, denotado por $x_s(t)$, puede ser escrito como:

$$x_s(t) = x(t)p(t) \quad (2.1)$$

donde $p(t)$ es la función de muestreo y modela la acción del interruptor muestreador. La función $p(t)$ es un tren de pulsos periódicos de periodo T . Transformando la función $x_s(t)$ al dominio de la frecuencia es posible determinar el teorema de Nyquist el cual establece que para poder replicar con exactitud la forma de una onda es necesario que la frecuencia de muestreo (f_m) sea superior al doble de la máxima frecuencia a muestrear (f_{max}) [2.10, 2.13, 2.14, 2.15]. Dado que el valor cuantizado es el único valor conservado después de que las muestras son cuantizadas se induce un error en el proceso de cuantización que no puede ser removido por procesos adicionales, el error cuadrático medio está dado por [2.10]:

$$E = \frac{S}{12} = \frac{D^2}{12} 2^{-2n} \quad (2.2)$$

donde n es la longitud de palabra cuantizada y D es el intervalo dinámico de la señal de entrada $x(t)$ y está definido como $D = \max[x(t)] - \min[x(t)]$. Se puede observar que el error cuadrático medio decrece exponencialmente con la resolución n del convertidor. Una medida cuantitativa de la calidad de la señal cuantizada es la relación señal a ruido (SNR) a la salida del cuantizador, la cual está definida como la relación de la potencia de la señal (P_s) y la potencia del ruido que la corrompe y puede ser expresado como [2.10]:

$$SNR = \frac{12P_s}{D^2} 2^{2n} \quad (2.3)$$

En esta ecuación se observa que la relación señal a ruido se incrementa con el incremento de n y con el decremento del intervalo dinámico D de la señal de entrada.

2.4 TÉCNICAS DE MUESTREO.

Aunque hay una gran variedad de técnicas para el muestreo de señales en el dominio del tiempo, hay dos métodos básicos, muestreo en tiempo real y muestreo en tiempo equivalente, este último se divide a su vez en aleatorio y secuencial. Cada método tiene distintas ventajas dependiendo del tipo de mediciones que se desean hacer. En el muestreo en tiempo real se requiere de solo una ocurrencia de la señal para adquirirla y el digitalizador opera a la máxima velocidad para adquirir la mayor cantidad de puntos posibles en el barrido. En el muestreo en tiempo equivalente se requiere de múltiples ocurrencias de la señal a medir para adquirir la forma de la señal y en cada repetición se pueden adquirir una o más muestras de la señal. El muestreo en tiempo equivalente toma ventaja de muchos de los eventos de la naturaleza y en los hechos por el hombre son repetitivos [2.10, 2.11].

En el muestreo en tiempo equivalente secuencial el digitalizador adquiere una muestra de la amplitud en cada periodo de la señal e incluye una señal de disparo o sincronía tal y como se muestra en la figura 2.7. Cuando se detecta el primer pulso de sincronía se toma la primera muestra después de un tiempo inicial T_1 . En el siguiente periodo una segunda muestra se toma después de un corto pero bien definido retardo de Δt , desde la posición donde se tomó la primera muestra. Cuando ocurre el siguiente pulso de sincronía, un pequeño tiempo Δt es agregado al retardo anterior y el digitalizador toma otra muestra. Este proceso es repetido muchas veces de tal manera que después de n periodos se han tomado n muestras que permiten reconstruir la forma de la señal.

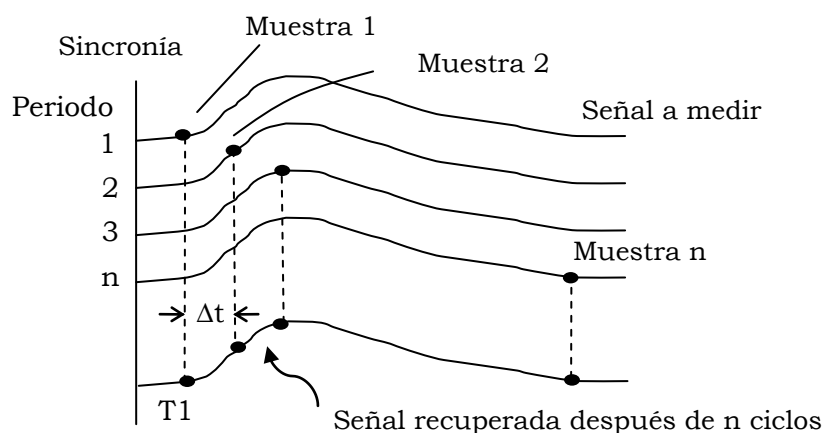


Figura 2.7. Proceso de muestro en tiempo equivalente secuencial.

Si el corrimiento Δt del punto de muestreo es suficientemente pequeño, entonces el ancho de banda de la señal está determinado solamente por el ancho de muestreo en cada punto. Debido a que el tiempo de procesamiento individual de cada muestra no afecta la resolución en tiempo, el método de muestreo proporciona un ancho de banda muy alto y una razón de muestreo también muy alta.

Para mejorar la relación señal a ruido el método de muestreo secuencial en tiempo equivalente se combina con un método de promediación [2.10, 2.12]. Por ejemplo en la figura 2.7 se mide un punto por cada ciclo y la señal completa se reconstruye con n ciclos, este procedimiento se repite m veces y las m formas de ondas obtenidas son promediadas. Este procedimiento mejora la relación señal a ruido en un factor de $n^{-1/2}$. El método Boxcar utiliza el mismo principio que el método de muestreo equivalente secuencial y solo se diferencia en la estrategia para promediar la señal. Mientras que en el primer método se procede inmediatamente a la siguiente señal y se promedian las curvas completas, en el método Boxcar se obtiene un promedio de n mediciones en n ciclos para un sólo punto de la señal y posteriormente se incrementa Δt .

2.5 PROPUESTA DE DISEÑO DEL SISTEMA DIGITALIZADOR

Como se menciona en la sección 2.1, el digitalizador desarrollado será utilizado en la medición de pulsos de fluorescencia del orden de nanosegundos. Dos importantes especificaciones a considerar en el comportamiento del digitalizador son el ancho de banda en su entrada analógica $BW(\text{dig})$ y su equivalente tiempo de subida $tr(\text{dig})$. El ancho de banda describe la habilidad de la entrada analógica del digitalizador para capturar una señal desde el mundo exterior al CAD con una mínima pérdida de amplitud y mínimo incremento del tiempo de subida. El ancho de banda se especifica como la frecuencia a la cual una señal sinusoidal es atenuada por un sistema al 70.7% de su amplitud original. El tiempo de subida está definido como el tiempo que tarda una señal en cambiar del 10% al 90% de la máxima amplitud. El ancho de banda y el tiempo de subida están inversamente relacionados por la ecuación 2.4, la cual está basada en un modelo unipolar de respuesta R-C de entrada limitada [2.13, 2.14, 2.15, 2.16, 2.17].

$$BW(\text{sig}) = \frac{0.35}{tr(\text{sig})} \quad (2.4)$$

Los conceptos de ancho de banda y su equivalente tiempo de subida se pueden aplicar tanto a señales como a dispositivos o sistemas completos. El ancho de banda del digitalizador $BW(\text{dig})$ introduce una pérdida de la amplitud e incremento del tiempo de subida. En la práctica se recomienda que el ancho de banda del digitalizador sea de 3 a 5 veces el ancho de banda de la señal $BW(\text{sig})$ a muestrear para reducir el error en la amplitud de 3% a 1% [2.17]. De acuerdo con 2.4 si el tiempo de subida de la señal es de 1 ns, el ancho de banda de la señal será entonces de 350MHz. El tiempo de subida medido puede ser estimado de la ecuación 2.5 donde $tr(\text{dig})$ es el tiempo de subida del digitalizador y $tr(\text{sig})$ es el tiempo de subida de la señal [2.13, 2.14, 2.15].

$$trm^2 = tr^2(dig) + tr^2(sig) \quad (2.5)$$

Elegimos un controlador digital de señales (dsPIC33FJ256GP710 de Microchip Technology Inc.) como el sistema digital programable para desarrollar la técnica de muestreo en tiempo equivalente. Entre la gran cantidad de recursos que el dsPIC33F ofrece, se encuentra un convertidor A/D de 12 bits, pero el ancho de banda de su entrada analógica es de solo 250KHz [2.18]. Este es un valor pequeño para nuestros propósitos, así entonces se recurrió a un convertidor externo tipo flash con ancho de banda grande (MAX12528 de Maxim/Dallas). La figura 2.8 muestra un diagrama esquemático del digitalizador con sus 3 bloques principales. El MAX12528 realiza conversiones continuamente con las transiciones positivas en su entrada de reloj CKAD y tiene un tiempo de latencia de 8 pulsos de CKAD para actualizar la salida digital (DATOS) de cada conversión [2.19].

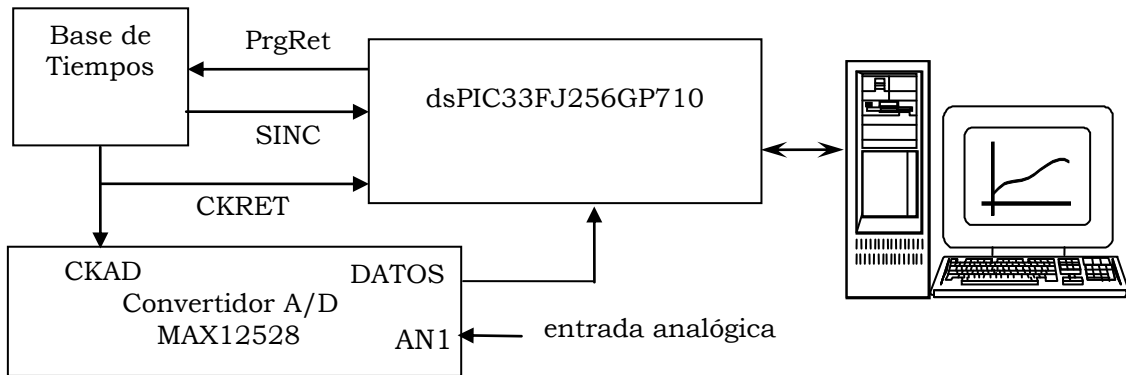


Figura 2.8. Diagrama esquemático del sistema digitalizador desarrollado.

Para aplicar la técnica de muestreo en tiempo equivalente la base de tiempos debe generar un retardo Δt a partir de la señal de un oscilador para producir la señal de reloj retardada CKRET. Este retardo Δt debe programarse desde el dsPIC (líneas de comando PrgRet). La base de tiempos debe generar un pulso de sincronía (SINC) periódico para iniciar el proceso de adquisición de un solo punto. Con cada transición positiva del reloj retardado CKRET el convertidor A/D toma una muestra del valor presente en su entrada analógica. El retardo se produce siempre respecto al pulso de sincronía SINC al inicio del proceso periódico de conversión y el pulso a medir también debe estar sincronizado con la señal SINC. Después de que el dsPIC detecta la señal SINC nos interesa leer el dato resultante de la conversión que se realiza con la siguiente subida de CKRET. Por lo tanto el dsPIC debe contar 8 pulsos de CKRET después de detectar SINC y luego leer el dato. Este dato se puede almacenar en la memoria del dsPIC, o bien, puede ser enviado a la PC. Al final del proceso se debe incrementar el retardo en un Δt y el dsPIC debe mantenerse en espera de la ocurrencia del siguiente pulso SINC.

El algoritmo para realizar un barrido en tiempo, aplicando la técnica de muestreo en tiempo equivalente secuencial puede ser descrito como se muestra en las siguientes líneas:

Repite desde $\Delta t=0$, hasta $\Delta t=\text{valor máximo}$
 espera ocurrencia del pulso SINC
 cuenta 8 pulsos de la señal CKRET
 lee dato de salida del convertidor A/D
 envía dato a PC o guárdalo en memoria del dsPIC
 actualiza Δt y programa su valor en la base de tiempos
 Fin

REFERENCIAS

- [2.1] K.D.Li Dessau, E.L., "Insights into High-Speed Detectors and High-Frequency Techniques", Application Note 1, <http://www.newfocus.com>
- [2.2] A. Davison, K.D. Li Dessau, "Faster Measurements for Today's Laboratory", Photonics Spectra, Sep 1998, pp 110-114.
- [2.3] Lakowicz Joseph R., "Principles of Fluorescence Spectroscopy", Klumer Academic/Plenum Publishers, 2nd Edition USA, 1999.
- [2.4] Paras N. Prasad, "Introduction to Biophotonics", Jhon Wiley&Sons, Inc., 2003.
- [2.5] Suzanne J. Lassiter, Wieslaw J. Stryjewski, Yun Wang, and Steven A. Soper, "Fluorescence Lifetime Methods", 14 Spectroscopy 17(6) June 2002, www.spectroscopyonline.com.
- [2.6] ORTEC, Application Note AN50 "Instrumentation for Fluorescence Lifetime
- [2.7] F.J. Bautista Diaz, "Sistema para la medición de Fluorescencia Inducida por Laser (LIF) Resuelta en el Tiempo", Tesis de grado de maestría, Departamento de Ing, Electrónica, SEPI-ESIME, IPN, México, Septiembre 2004.
- [2.8] E. Moreno, "Fundamentos de Instrumentación con PC y LabVIEW" Notas de Curso, Lab. de instrumentación electrónica, ESFM_IPN, Noviembre 2003.
- [2.9] Douglas V, Hall, "microprocessors and interfacing, programming and hardware", McGraw-Hill Inc., Singapore, 1986.
- [2.10] Rodger E. Ziemer, William H. Tranter and D. Ronald Fannin, "Signals and Systems: continuous and discrete, 4th Edition, Pearson Education Prentice Hall Inc., Upper Saddle River, NJ USA, 1998.
- [2.11] E. Moreno-Garcia, J.M. de la Rosa-Vazquez y O. Alonzo-Lárraga, "An Aproach to the Equivalent-Time Sampling Technique for pulse Transient Measurements.", IEEE Computer Society.
- [2.12] "PCS-150 / PCS-200 High Speed Boxcar Modules", <http://www.becker-hickl.de>, PCSAPP.DOC, Becker & Hickl GmbH, Kolonnenstr. 29, 10829 Berlin, (2002).
- [2.13] National Instruments, Application Note "Bandwidth, Sample Rate and Nyquist Theorem", <http://zone.ni.com>.
- [2.14] Tektronix Inc, Technique Primer 47W-7209, "Sampling Oscilloscope Techniques, 10/1989.
- [2.15] Agilent Technologies, "The Fundamentals of Signal Analysis", Application Note 243, <http://www.agilent.com/find/assist>
- [2.16] Tekronix Inc., "XYZs of Oscilloscopes", 2001 <http://www.tektronix.com>.

- [2.17] J. Ardizzoni. "High-Speed Time-Domain Measurements – Practical Tips for Improvement". Analog Dialogue 41-03, March 2007, www.analog.com
- [2.18] Microchip, "dsPIC33F Product Overview", DS70158C, Microchip Technology Inc., 2005.
- [2.19] Maxim/Dallas semiconductor, "MAX12528 Dual, 80Msps, 12-Bit, IF/Baseband ADC", Data Sheet, www.maxim-ic.com, Dicembre 2005.

CAPÍTULO 3

SISTEMA DIGITALIZADOR

En este capítulo se describen el funcionamiento y los principales recursos de los dispositivos que en conjunto forman el digitalizador propuesto en el capítulo anterior. Estos dispositivos son el controlador dsPIC33FJ256GP710, el convertidor analógico digital MAX12528 y una base de tiempos desarrollada en el laboratorio de electrónica de la ESFM cuyo funcionamiento se basa en el uso de tres líneas de retardo DS1020-15. Se presenta la interconexión de estos dispositivos y se describe la manera en que se lleva a cabo la técnica de muestreo en tiempo equivalente.

3.1 DESCRIPCIÓN DEL PROCESADOR DSPIC33FJ256GP710

Como se menciona en la propuesta de diseño del sistema digitalizador en la sección 2.5, en este trabajo se utiliza un miembro de la familia de controladores de señales digitales dsPIC33F fabricados por Microchip Technology. Estos dispositivos combinan la arquitectura y la programación de los microcontroladores PIC de 16 bits con los recursos de hardware y software necesarios para soportar las prestaciones fundamentales de los procesadores digitales de señales (DSP). La diferencia particular entre los miembros de la familia es el tamaño de su memoria FLASH para almacenar programas (no-volátil) y el tamaño de su memoria SRAM (Static RAM) para guardar datos temporales [3.1].

Estos controladores trabajan a 40 MIPS (Mega Instrucciones Por Segundo) y la arquitectura de su CPU es del tipo Harvard mejorada. Como soporte central de información la CPU contiene un banco de 16 registros de 16 bits cada uno; con un bus de datos de 16 líneas y otro bus de instrucciones de 24 líneas, dos acumuladores de 40 bits y un robusto registro de desplazamiento. Para potenciar la velocidad de las operaciones aritméticas complejas se tiene un procesador que contiene un multiplicador de hardware de 17x17 bits. Se tiene un conjunto de hasta 84 instrucciones, la mayoría de 24 bits de longitud y ejecutables en un ciclo de instrucción.

En particular se escogió trabajar con el dsPIC33FJ256GP710 para desarrollar el sistema digital programable, el cual contiene 256Kbytes de memoria de programa tipo FLASH y 30 Kbytes de 16 bits de memoria de datos SRAM. Esta memoria se divide en dos espacios X e Y, que pueden ser accedidos simultáneamente en las operaciones matemáticas del DSP. Existe una gran variedad de periféricos y dispositivos complementarios que se integran en los diversos modelos de la familia dsPIC33F,

algunos de los recursos que se implementan en el dsPIC33FJ256GP710 son los siguientes [3.1]:

- 85 líneas de entrada/salida programables que pueden realizar funciones alternas.
- 9 temporizadores de 16 bits, se pueden agrupar en 4 temporizadores de 32 bits
- 8 entradas de captura
- 8 salidas de comparación
- 2 convertidores A/D que pueden trabajar como convertidores de 10 bits de alta velocidad o de 12 bits de alta resolución.
- 2 módulos UART (Universal Asynchronous Receiver Transmitter)
- 2 SPI (Serial Peripheral Interface)

En la figura 3.1 se presenta un diagrama esquemático de la arquitectura interna del dispositivo DSPIC33FJ256GP710 y en las siguientes líneas se describen los recursos que se utilizan en este trabajo.

LÍNEAS DIGITALES DE ENTRADA/ SALIDA (E/S).

El dsPIC cuenta con 85 líneas de E/S que se pueden programar para realizar otras funciones alternas que permiten el manejo de los periféricos. Todas las líneas pueden suministrar o absorber hasta 25mA. Las líneas que se programan para realizar funciones alternas dejan de ser líneas de entrada/salidas digitales cuando están funcionando dichos periféricos. Las líneas de E/S se agrupan para conformar los puertos A, B y D de 16 bits, los puertos C y E de 8 bits y los puertos F y G de 11 bits. Cada puerto tiene asociados 3 registros:

- TRISX: Registro de dirección de datos. Cada uno de sus bits determina la dirección de la línea E/S asociada
- PORTX: Registro del puerto de E/S. Una lectura de PORTX recoge el valor lógico de las líneas E/S, mientras que una escritura los escribe.
- LATX: Registro tipo LATCH de E/S.

TEMPORIZADORES

El dsPIC tiene 9 temporizadores de 16 bits llamados TIMER1, TIMER2,..., TIMER9 ocho de los cuales pueden concatenarse por parejas formando temporizadores de 32 bits. El

TIMER1 funciona como temporizador tipo A, los TIMER 2 y 4 funcionan como temporizadores tipo B y los restantes como tipo C. Los de tipo B se pueden concatenar con los de tipo C para configurar temporizadores de 32 bits. Los temporizadores utilizan un divisor de frecuencias previo que puede dividir la frecuencia de los impulsos aplicados por 1, 8, 64 o 256. Para controlar el comportamiento de los temporizadores, cada uno dispone de 3 registros de 16 bits cada uno, estos son:

- TMRX: Guarda el valor que establece el tiempo a controlar.
- PRX: Es el registro de periodo asociado al temporizador.
- TXCON: Registro de control.

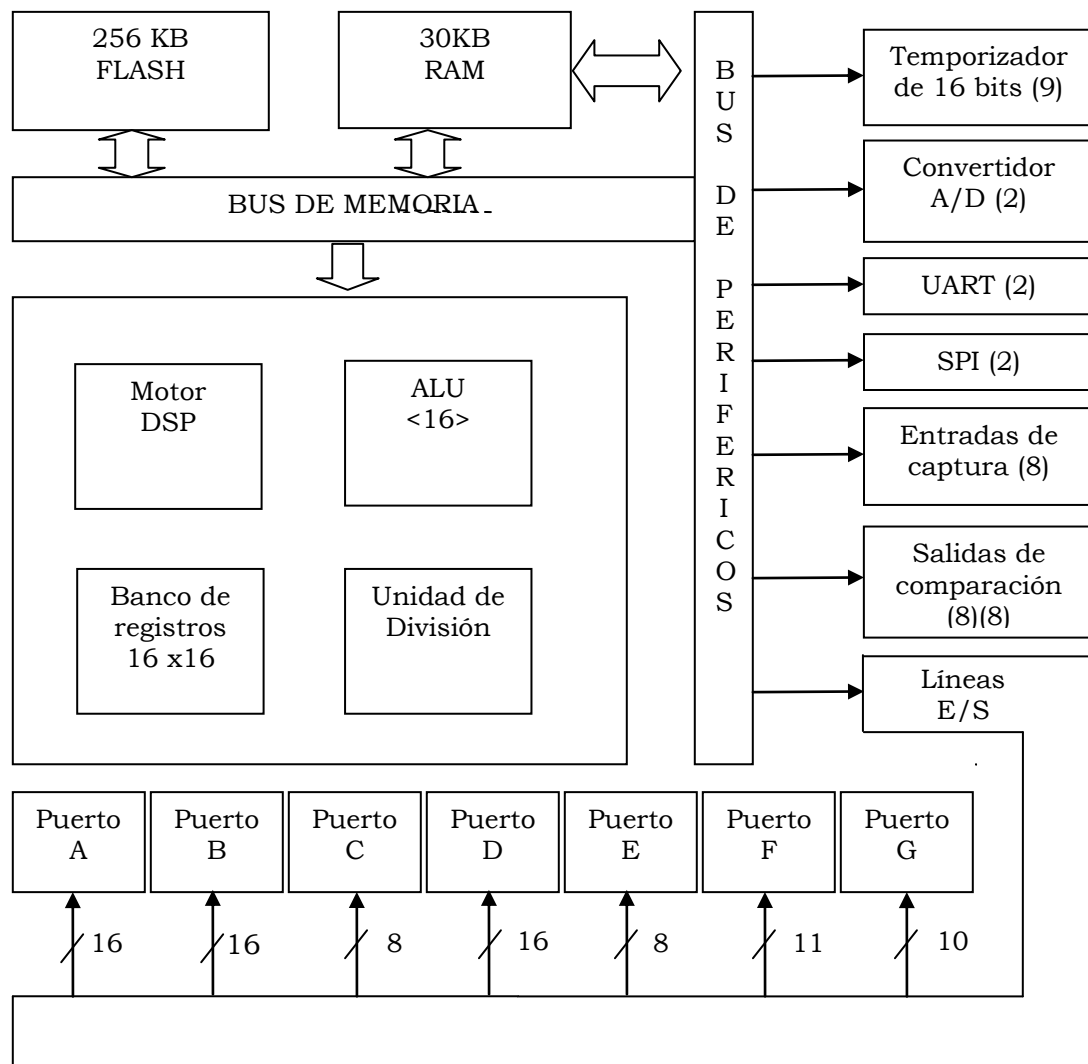


Figura 3.1. Diagrama correspondiente a la arquitectura interna del dispositivo DSPIC33FJ256GP710.

Los temporizadores pueden funcionar en 4 modos:

- 1.- TEMPORIZADOR SÍNCRONO. En este modo pueden trabajar 5 temporizadores y emplean una señal de reloj interno de sistema ($F_{osc}/4$). El valor del temporizador se incrementa con cada impulso de entrada.
- 2.- CONTADOR SÍNCRONO. En este modo funciona todos los temporizadores y los impulsos de reloj pueden provenir de una señal externa introducida por la línea TxCK que se sincroniza con la interna.
- 3.- CONTADOR ASÍNCRONO. En este modo sólo funciona el temporizador A y el incremento del valor de la temporización se produce de forma asíncrona con respecto al reloj interno del sistema.
- 4.- CONTADOR CON DISPARO POR ACUMULACIÓN DE TIEMPO. El registro contador se incrementa en función de la duración que tienen los impulsos aplicados por la línea TxCK.

CONVERTIDORES A/D.

Los convertidores A/D integrados en el dsPIC pueden ser de 10 bits de resolución de alta velocidad, o bien, de 12 bits de alta resolución. El convertidor A/D de 10 bits tiene las siguientes características:

- Velocidad máxima de conversión de 1Msps con una impedancia máxima de 500 Ohm, trabajando normalmente a 500 ksps a 5V y 100 ksps a 2.7V.
- Ancho de banda analógico de 250KHz
- Tiempo de muestreo de 154 ns.
- Capacidad de hasta 4 canales de entrada con muestreo simultaneo.
- Exploración automática de canal.
- Buffer de 16 palabras para el almacenamiento de los resultados de la conversión.

El convertidor de 12 bits sólo dispone de un canal CH0 de amplificador unipolar de captura y mantenimiento. La velocidad máxima de conversión es de 200 ksps, siendo la normal de 100 ksps a 5V y otra de 50 ksps a 2.7V.

MÓDULO UART

El dsPIC dispone de dos módulos UART. El UART es un elemento destinado a soportar la comunicación serie asíncrona y funciona de forma bidireccional adaptándose al trabajo de mucho periféricos. La comunicación se realiza con dos líneas, una para la transmisión UTX y otra para la recepción URX, entrando y saliendo los bits a una frecuencia controlada internamente. Funciona con datos de 8 o 9 bit, 1 o 2 bits de paro y un bit de paridad par o impar. El UART dispone de un generador de baudios con un predivisor de frecuencia que se encarga de generar la frecuencia de trabajo del módulo. El modulo UART esta compuesto por los siguientes registros:

- UXM0DE: Registro de modo.
- UXSTA: Registro de control y estado.
- UXRREG: Registro de recepci3n.
- UXTXREG: Registro de transmisi3n.
- UBRG: Registro de generador de baudios.

El comportamiento del UART se gobierna desde los registros UXSTA, UXRREG y UXTXREG. Pueden generar una interrupci3n tanto el bloque transmisor como el receptor.

TARJETA DE DESARROLLO EXPLORER 16

La tarjeta de desarrollo "Explorer 16" permite la evaluaci3n de las prestaciones de los microcontroladores de 16 bits PIC24 y de los controladores de 24 bits de la familia dsPIC33 de forma eficiente [3.2]. La tarjeta est1 dise1ada para poder trabajar con el depurador ICD2 (ver secci3n 4.1), las facilidades de depuraci3n y la emulaci3n a tiempo real ayudan a acelerar la evaluaci3n de los recursos de estos dos microcontroladores. En la figura 3.2 se esquematiza los principales recursos de la tarjeta y la manera en que se comunica con los perif3ricos.

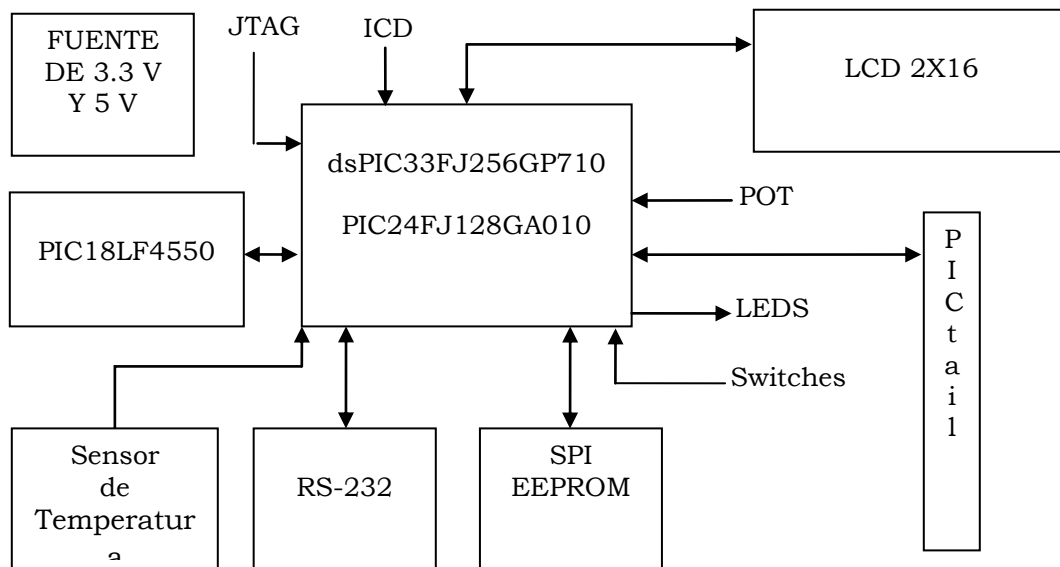


Figura 3.2. Diagrama de bloques de la tarjeta de desarrollo Explorer 16.

La tarjeta incluye un conector de 100 pines de donde se pueden conectar el emulador o bien, uno de los dos módulos que contiene el procesador con el que se quiere trabajar, uno con el PIC24FJ128GA010 y otro con el dsPIC33FJ256GP710. Además incluye también un conmutador para seleccionar el modo de trabajo con emulador o con la pieza física. Dado que varios de los pines del dsPIC realizan funciones alternas para atender recursos propios de la tarjeta, no todos los pines de las líneas de E/S están disponibles, por ejemplo el puerto D solo tiene 13 líneas disponibles, el A 4 líneas y el C solo 3. La tarjeta proporciona las siguientes funciones:

- Conexión para comunicación serie tipo RS-232
- Sensor de temperatura TC1047A de alta precisión
- Conectividad USB mediante un PIC18F4550
- Conector estándar de 6 líneas para el ICD2 (ver sección 4.1)
- Pantalla de cristal líquido de 2x16 caracteres (se puede añadir una pantalla gráfica)
- Pulsadores de reset y de entrada para el usuario
- Potenciómetro divisor de voltaje para entrada analógica
- 8 LEDS indicadores
- Memoria EEPROM
- Cristal de precisión de 8MHz y cristal para realización de un reloj en tiempo real de 32.768KHz
- Area de expansión para prototipos

3.2 DESCRIPCIÓN DEL CONVERTIDOR ANALÓGICO DIGITAL MAX12528

Como se menciona en la propuesta de diseño del sistema digitalizador en la sección 2.5, en este trabajo se utiliza un convertidor digital analógico de alta velocidad externo al que contiene el DSP. En la construcción del digitalizador se utiliza el convertidor analógico digital (CAD) MAX12528 (Maxim/Dallas) el cual es un circuito integrado que contiene dos convertidores independientes que pueden operar en forma concurrente con la misma señal de reloj. Estos convertidores son de 12 bits, con una razón de muestreo de 80Msps, un ancho de banda en su entrada analógica de 750MHz, pequeño tamaño y bajo consumo de energía (726mW). En la figura 3.3 se muestra el diagrama de bloques del convertidor A que se encuentra dentro del circuito MAX12528 [3.3].

La entrada analógica es completamente diferencial (INAP INAN) con un circuito de captura y retención y puede trabajar con una referencia de voltaje externa o una interna de 2.048V. El convertidor soporta una entrada de reloj diferencial (CLKP CLKN) o simple con dos modos de división de frecuencia ($f_{in}/2$ y $f_{in}/4$) y ecualizador de ciclo de aprovechamiento (DCE) para compensar las variaciones del ciclo de aprovechamiento. El CAD opera con una fuente de voltaje de 3.3V y una de 1.7 a 3.6V para las salidas digitales. El circuito proporciona una versión simple del reloj de entrada en la línea de salida DAV (Data Valid) para sincroniza la lectura de la salida digital.

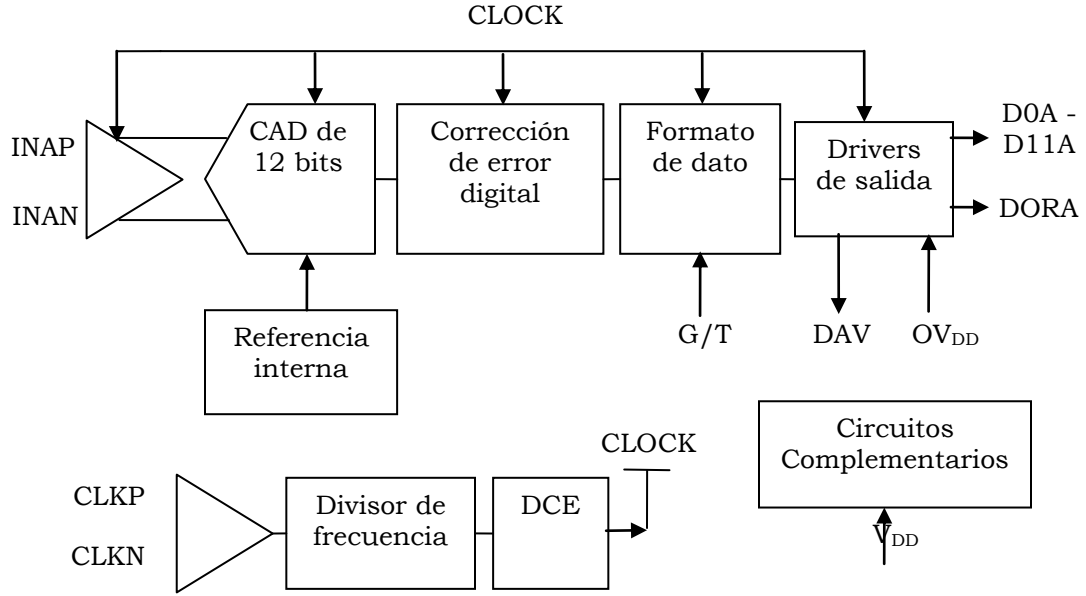


Figura 3.3. Diagrama funcional del convertidor analógico digital A del MAX12528.

Los datos de las salidas digitales cambian en la bajada de la señal DAV y el dato es válido en la subida de ésta. También proporciona una salida indicadora de sobreflujo en la medición (DORA). El convertidor posee dos puertos digitales de salida tipo CMOS de 12 bits cada uno con formato de salida complemento a dos o código Gray dependiendo del valor de la entrada lógica G/T. Las siguientes ecuaciones definen la relación entre la salida digital y la entrada analógica:

Código Gray (G/T=1):
$$V_{IN_P} - V_{IN_N} = \frac{\frac{4}{3}V_{REF}(CÓDIGQ_{10} - 2048)}{4096}$$

Complemento a dos (G/T=0):
$$V_{IN_P} - V_{IN_N} = \frac{\frac{4}{3}V_{REF}(CÓDIGQ_{10})}{4096}$$

Donde CODIGO₁₀ es el equivalente decimal del código de la salida digital.

El MAX12528 usa una arquitectura de 10 etapas totalmente diferenciales (figura 3.4) para permitir altas velocidades de conversión con poco consumo de energía. Las muestras que son tomadas en la entrada analógica se mueven progresivamente a través de las etapas en cascada cada medio ciclo de reloj, de tal manera que la información digital de la conversión se encuentra disponible después de 8 ciclos de reloj. El convertidor A/D de cada etapa es del tipo Flash.

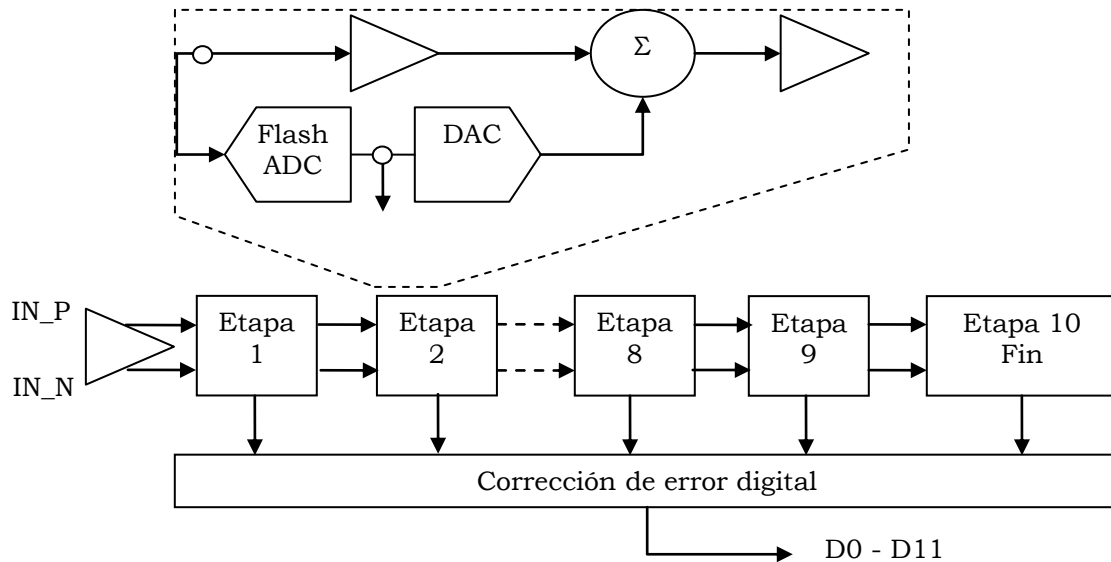


Figura 3.4. Arquitectura de un convertidor A/D del circuito MAX12528

REQUERIMIENTOS EN TIEMPO DEL SISTEMA

La figura 3.5 muestra la relación temporal entre el reloj, las entradas analógicas, la señal DAV (Data-Valid Output) y el dato de salida resultante. La entrada analógica es muestreada durante la bajada (subida) de CLKP (CLKN) y el dato resultante aparece en las salidas digitales después de 8 ciclos de reloj. La señal DAV es una versión de salida simple de la entrada de reloj y está compensada para corregir cualquier variación del ciclo de aprovechamiento en la entrada de reloj. Los datos provenientes del convertidor son cargados en la salida digital (D0A/B-D11A/B) en la caída de DAV y son validos 7ns antes hasta 7ns después de la subida de DAV, por lo que generalmente se ocupa este indicador para sincronizar la lectura de los datos por un circuito externo.

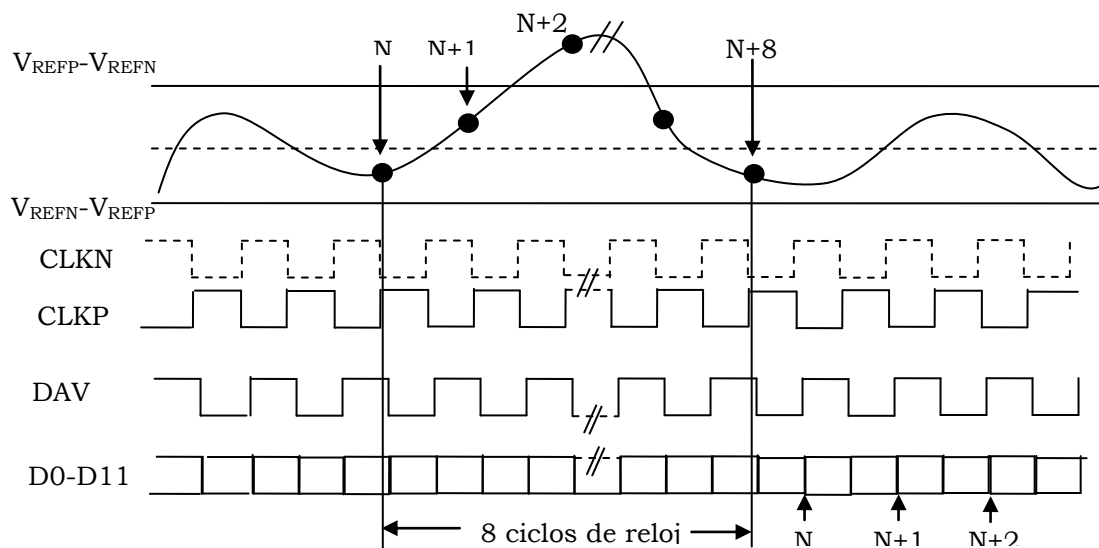


Figura 3.5. Requerimientos en tiempo del sistema.

TARJETA DE EVALUACIÓN DEL MAX12528.

El convertidor analógico digital antes descrito puede ser evaluado mediante la tarjeta de evaluación MAX12528 Kit, la cual contiene todos los componentes necesarios para evaluar el funcionamiento del convertidor A/D [3.4]. Las entradas analógicas y la entrada de reloj tienen un acoplamiento inductivo con un transformador de pulsos y un circuito que permite que la entrada analógica admita señales de tipo simple para luego transformarlas en señales de tipo diferencial, que son las requeridas para que los convertidores del MAX12528 pueda operar adecuadamente. La tarjeta incluye controladores de salida que almacenan temporalmente las señales digitales de salida. Como voltaje de referencia se ocupa la referencia interna del CAD de 2.048 V por lo que los voltajes de las señales a medir por la tarjeta deben estar en un intervalo de $\pm 1.0235V$. La tarjeta de evaluación requiere 2 fuentes de voltaje separadas para los circuitos digitales (OVDD=2.0V) y analógicos (VDD=3.3V) del convertidor, además de una fuente de 2V (VLOGIC) para los buffers de salida de la tarjeta.

3.3 BASE DE TIEMPOS

Como elementos principales de la base de tiempos se utiliza un oscilador de 5.068 MHz (SG-51P) con una estabilidad en frecuencia de ± 50 ppm y tres líneas programables de 8 bits (3xDS1020-A15). El circuito integrado DS1020 es una línea de retardo programable de 8 bits que reproduce el estado lógico de la entrada en la salida después de un retardo programado por el usuario [3.5]. Los valores de retardo pueden variarse en 256 pasos iguales y son programados ya sea por las 3 líneas de su puerto serie o por su puerto paralelo de 8 bits. El modelo más rápido (DS1020-15) ofrece un retardo máximo de 38.25ns con un incremento en el retardo de 0.15ns, mientras que el modelo más lento (DS1020-200) tiene un retardo máximo de 510ns con un incremento en el retardo de 2ns. Todos los modelos tienen un retardo inherente de 10ns (paso cero). Después de que se ha programado la línea de retardo el estado de la entrada lógica es reproducido en la salida sin inversión lógica. Dado que el DS1020 tiene un diseño tipo CMOS, los pines de entrada no usados (P0-P7) se deben conectar a un nivel lógico bien definido. En la figura 3.6 se muestra el bloque funcional del DS1020 y la descripción de sus terminales.

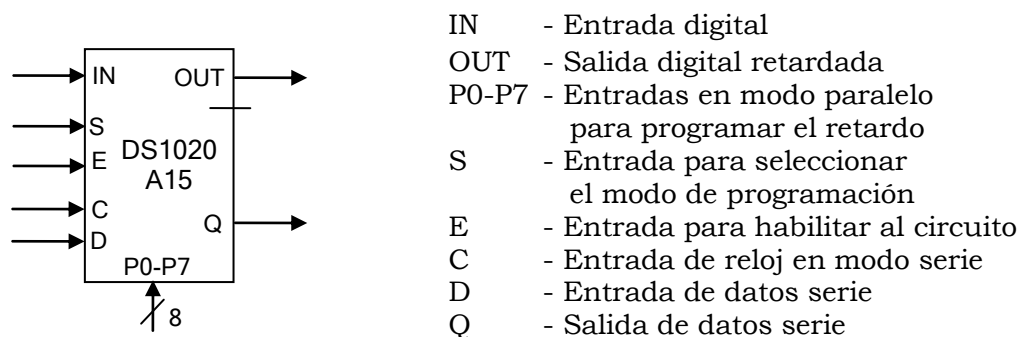


Figura 3.6 Diagrama funcional del DS1020 y descripción de terminales.

En este trabajo programamos a las tres líneas de retardo en modo serie ($S=0$). En este modo la salida del DS1020 reproducirá el estado lógico de la entrada después de un retardo determinado por un valor de 8 bits programado por el puerto serie D. La secuencia temporal para programar en modo serie se muestra en la figura 3.7.

El dato es cargado desde el bit de mayor peso (MSB) al bit de menor peso (LSB) durante la subida del reloj serie (C). El pin habilitar (E) debe estar en 1 lógico para cargar un dato en el puerto D. La transferencia de datos concluye y el nuevo valor de retardo se activa cuando el pin habilitar (E) regresa a 0 lógico.

Las tres líneas de retardo DS1020-15 se conectan en cascada de tal manera que se puede producir un retardo máximo de 114.75ns con una resolución de 0.15ns. Estos circuitos se utilizan para programar un retardo de la señal del oscilador. La señal de reloj retardada (CKRET en la figura 3.8) se aplica a los dos convertidores del MAX12528, los cuales toman una muestra de la señal presente en su entrada analógica con la transición positiva de CKRET. Los valores en el retardo para cada DS-1020 se programan con 5 líneas del puerto A del dsPIC.

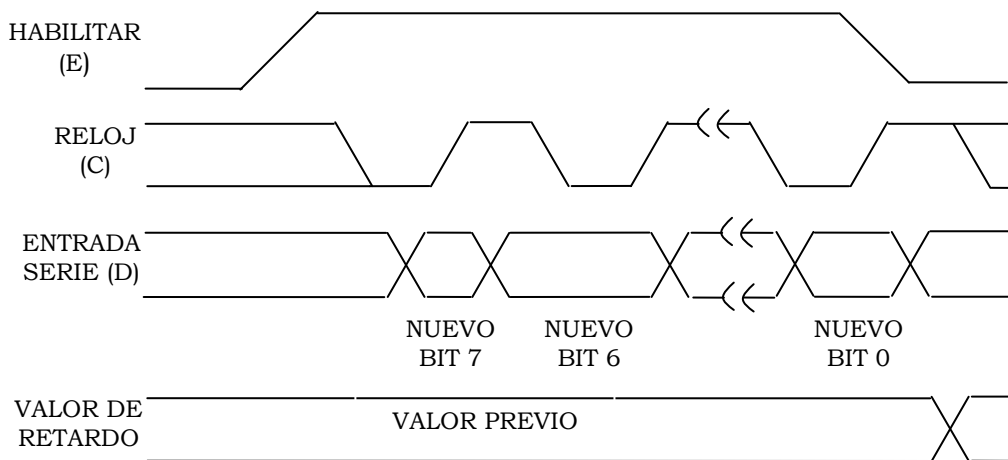


Figura 3.7. Programación en serie de la línea de retardo DS1020, ($S=0$). El ancho del pulso habilitar y de reloj debe ser de al menos 50ns.

La base de tiempos incluye un circuito para generar un pulso de sincronía (SINC en la figura 3.8) con una razón de repetición variable. Este pulso de sincronía es leído por el DSP e inicia el proceso de digitalización para un punto. La señal a medir debe estar sincronizada con el pulso SINC (ver figura 2.8) y la señal de reloj retardada CKRET realiza los corrimientos Δt para aplicar la técnica de muestreo en tiempo equivalente descrita en la sección 2.4.

Adicionalmente al pulso SINC, el cuál es generado con dos circuitos monoestables TTL (74LS123), en la base de tiempos se tiene un circuito monoestable programable de alta velocidad (DS1040Z-A15) para producir pulsos de prueba con duración de hasta 15ns. El ancho del pulso puede ser programado desde 5 hasta 15ns en cuatro pasos de 2.5ns

cada uno, con tiempos de subida y de bajada de 4ns. En el apéndice A se presenta el diagrama esquemático de la base de tiempos.

3.4 PROTOTIPO DEL DIGITALIZADOR

En la figura 3.8 se presenta la interconexión entre los cuatro principales bloques del digitalizador desarrollado.

En las siguientes líneas se describen los recursos del dsPIC que se utilizan en el sistema digitalizador y se indican los registros de programación asociados a esos recursos.

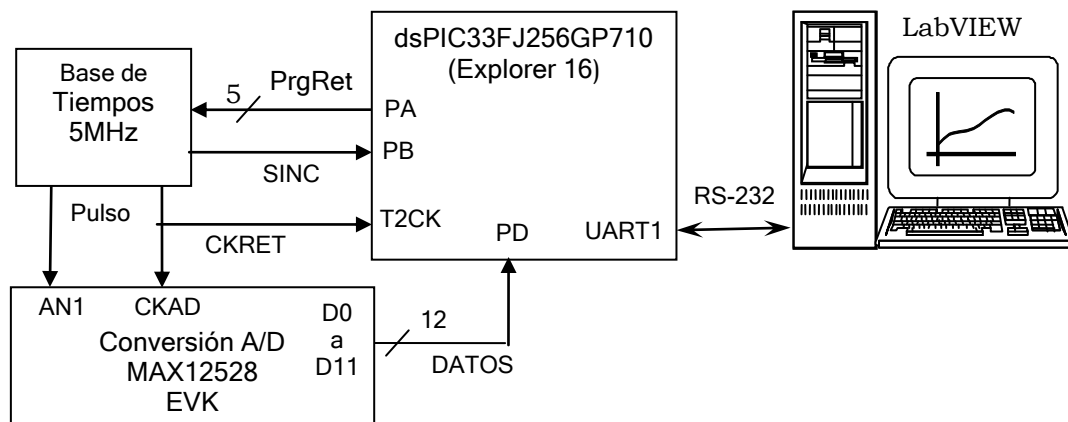


Figura 3.8. Diagrama a bloques del sistema digitalizador desarrollado. Se muestran los recursos utilizados en el dsPIC para enlazar los distintos dispositivos necesarios para llevar a cabo la técnica de muestreo equivalente.

Puerto A (PA). 5 líneas de salida (PrgRet) para controlar la programación del retardo Δt de la señal de reloj principal generada por el oscilador. La línea de reloj retardada (CKRET) aplica la técnica de muestreo en tiempo equivalente secuencial. Registros asociados: TRISA para programar la dirección de las 5 líneas y PORTA para enviar los códigos de programación hacia las líneas de retardo DS-1020 en la base de tiempos.

Puerto B (PB). Una línea de entrada para leer el pulso de sincronía (SINC) generado en la base de tiempos con una razón de repetición variable. Este pulso inicia la captura de un sólo punto en cada repetición. Registros asociados: TRISB para programar la dirección de la línea y PORTB para leer el estado de la línea SINC.

Temporizador 2 (T2CK). Se utiliza como contador de eventos externos. Cuenta hasta 8 pulsos del reloj retardado CKRET para sincronizar la lectura de datos en el puerto D desde la salida de la tarjeta de evaluación del MAX12528. Registros asociados: T2CON para programar en modo síncrono al temporizador 2, PR2 registro para

guardar el valor de un conteo preestablecido y TMR2 registro del contenido del contador.

Puerto D (PD). 12 líneas de entrada para leer los datos de la salida digital del MAX12528. Registros asociados: TRISD para programar la dirección de las 12 líneas y PORTD para leer el valor presente en las 12 líneas.

Transmisor Receptor Asíncrono Universal (UART1). Interfase tipo serie para intercambiar datos y comandos con la computadora vía un protocolo RS-232. Registros asociados: UXMODE para programar el modo de operación, UXSTA registro de control y estado, UXRREG registro de recepción, UXTXREG registro de transmisión y UBRG registro de generador de baudios.

Los pines RA9, RA14 y RA15 del puerto A se utilizan como líneas “ENABLE” para cada uno de los retardos DS1020. Los tres retardos comparten la entrada de reloj (C) así como la línea de programación (D), los pines RA11 y RA10 están asignados a realizar estas funciones respectivamente. Para programar uno de los retardos solo basta poner la línea enable de ese retardo en un nivel alto manteniendo las otras dos en un nivel bajo y después programar las líneas D y C como se ha descrito en la sección 3.2. El pin RB2 del puerto B tiene la función de ser la entrada del pulso de sincronía (SINC). Cuando el programa que se ejecuta en el dsPIC detecta la transición positiva de SINC, se inicia el conteo de 8 pulsos del reloj retardado (CKRET) y luego se lee el puerto D. El registro PR2 se programa con un valor de 8 y el registro de conteo TMR2 se reinicia cada vez que se lee un dato enviado por el MAX12528.

Dado que el puerto D del dsPIC es el que tiene más líneas disponibles (13) en la tarjeta Explorer16, es este el que se utiliza para la lectura de los datos de la salida digital de la tarjeta de evaluación MAX12528. Los pines RD6, RD7 Y RD13 están asociados a interruptores en la tarjeta Explorer16, sin embargo por rapidez se lee el puerto D completo (16 bits), por lo tanto se debe crear una rutina para descartar estos tres bits antes de procesar el dato. La comunicación entre el dsPIC y la PC se lleva a cabo mediante el modulo UART1 y está configurado para trabajar con datos de 8 bits, sin paridad y con un bit de paro. Dentro del programa que se ejecuta en el dsPIC se trabaja con los registros U2T1REG Y U2R1REG para transmitir o recibir bytes respectivamente.

PROCESO DE DIGITALIZACIÓN

Para aplicar la técnica de muestreo en tiempo equivalente secuencial se retarda la señal de reloj CK de 5.068 MHz de la base de tiempos para producir la señal de reloj retardada CKRET a la de entrada de reloj del MAX12528. Con cada transición positiva de ésta señal retardada el MAX12528 toma una muestra del valor presente en su entrada analógica. El retardo se produce siempre respecto al pulso de sincronía SINC al inicio del proceso periódico de conversión y el pulso a medir también debe estar sincronizado con la señal SINC. En la figura 3.9 se ilustra la relación temporal de las señales. Después del pulso de sincronía y de tomar una muestra de la señal analógica con la

subida de CKRET se cuentan 8 subidas más para leer el dato. Después de leer el dato, éste se envía a la computadora, se incrementa el valor de las líneas de retardo DS1020 para incrementar Δt en 0.15ns y se espera por el siguiente pulso de sincronía.

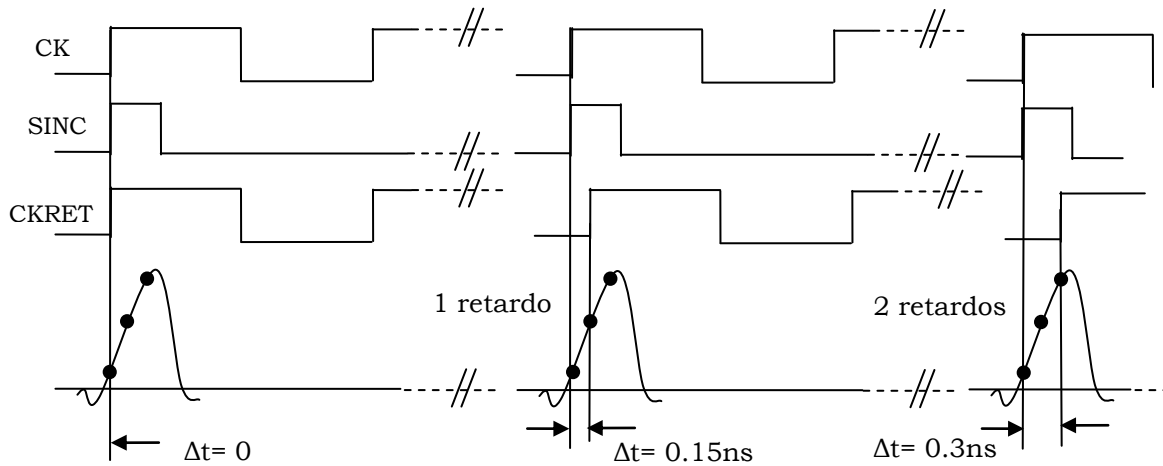


Figura 3.9. Señales del sistema digitalizar que muestrean en tiempo equivalente la señal analógica a medir.

Como ya se menciona en la sección 3.3 el circuito DS1020 puede producir un retardo desde 10 ns (retardo para código cero) hasta 48.25ns con incrementos de 0.15ns. Las tres líneas de retardo conectadas en serie pueden generar un retardo desde 30ns hasta 144.75ns. En el programa que ejecuta el dsPIC los retardos se programan de tal forma que en cada proceso periódico de conversión se incrementa 0.15ns el retardo de CKRET. La señal retardada (CKRET) también se utiliza para sincronizar en el programa del dsPIC la lectura de las líneas de DATOS, por medio del temporizador 2 (TMR2) del dsPIC como se ha explicado anteriormente. El dsPIC ejecuta un programa que aplica la técnica de muestreo en tiempo equivalente, recibe comandos de la computadora y envía datos hacia ella. En la computadora se ejecuta un programa de control escrito en el lenguaje gráfico G en el ambiente de LabVIEW, el cual lee la información de las mediciones desde el dsPIC, las procesa y las presenta en una gráfica.

El ancho de banda del digitalizador desarrollado está dado por el ancho de banda de su entrada analógica, el cual corresponde al valor de 750 MHz especificado para el MAX12528 [3.3]. De acuerdo a la ecuación (2.4) el equivalente tiempo de subida del digitalizador es de 0.466 ns. Para una señal de entrada con tiempo de subida de 1ns, según (2.4) su equivalente ancho de banda es de 350 MHz. A esta frecuencia la entrada analógica del digitalizador introduce un error en amplitud cercano al 5% [3.6]. Con la ecuación (2.5) se puede determinar que el incremento en la medición del tiempo de subida es cercano a 0.1 ns (10%). Este valor es inferior a la resolución del sistema de 0.15ns. Aplicando la técnica de muestreo en tiempo equivalente secuencial la máxima razón de muestreo del digitalizador está dado por el retardo Δt entre las muestras y no por la razón de muestreo especificada para el convertidor A/D (80 MSPS), esto es: $f_m = 0.15^{-1}\text{ns} = 6.66 \text{ GSPS}$.

REFERENCIAS

- [3.1] Microchip, “dsPIC33F Product Overview”, DS70158C, Microchip Technology Inc., 2005.
- [3.2] Microchip, “Explorer 16 Development Board, User’s Guide”, DS51589A, Microchip Technology Inc., 2005.
- [3.3] Maxim/Dallas semiconductor, “MAX12528 Dual, 80Msps, 12-Bit, IF/Baseband ADC”, Data Sheet, www.maxim-ic.com, Diciembre 2005.
- [3.4] Maxim/Dallas semiconductor, “MAX12528 Evaluation Kit”, Data Sheet, www.maxim-ic.com, Diciembre 2005.
- [3.5] Maxim/Dallas semiconductor, “DS1020 Programmable 8-Bit Silicon Delay Line”, Data Sheet, www.maxim-ic.com
- [3.6] J. Ardizzoni. “High-Speed Time-Domain Measurements – Practical Tips for Improvement”. Analog Dialogue 41-03, March 2007, www.analog.com
- [3.7] Angulo Usategui, José Ma. “DsPIC diseño práctico de aplicaciones”, McGraw-Hill, España, 2006.

CAPÍTULO 4

PROGRAMAS

En este capítulo se describe el software de desarrollo MPLAB, necesario para programar al microcontrolador dsPIC. También se describe brevemente LabVIEW, el cual es un ambiente de programación del lenguaje gráfico G, que permite la comunicación entre la PC y la tarjeta de desarrollo EXPLORER16. Se presenta además el programa que desarrolla la técnica de muestreo en tiempo equivalente y establece comunicación con la PC para que las señales medidas sean graficadas en tiempo real. Este programa está almacenado en la memoria del dsPIC33FJ256GP710 y se ejecuta por petición del usuario desde un programa de control desarrollado en el lenguaje gráfico G de LabVIEW.

4.1 HERRAMIENTAS DE DESARROLLO.

Para el control y operación del sistema digitalizador se desarrollaron dos programas: uno en el lenguaje ensamblador propio del dsPIC y otro en el lenguaje gráfico G de LabVIEW. Para desarrollar estos programas se utilizaron herramientas de programación como MPLAB en el primer caso y el ambiente de LabVIEW 7.0.

MPLAB es un software que permite desarrollar aplicaciones para los microcontroladores PIC y dsPIC de Microchip. MPLAB es un paquete que coordina todas las herramientas desde una interfaz gráfica, una vez que el código es escrito, este puede ser convertido a instrucciones ejecutables y cargado a un microcontrolador para ver como trabaja. En este proceso múltiples herramientas son necesarias: un editor para escribir el código, un administrador del proyecto para organizar archivos y un compilador o ensamblador para convertir el código fuente a código de máquina. El administrador de proyecto de MPLAB organiza los archivos a editar y otros archivos asociados para que puedan ser enviados a las herramientas de lenguaje y ser ensamblados o compilados. El editor de textos que se utiliza para escribir el código reconoce las estructuras en el texto y usa un código de colores para identificar varios elementos, tales como instrucciones nemotécnicas, estructuras en lenguaje C y comentarios. Después de que el código ha sido escrito, el editor trabaja con las otras herramientas para mostrar la ejecución del código en el depurador, también pueden ser colocados puntos de paro en el editor y los valores de las variables pueden ser inspeccionadas con tan solo colocar el puntero del ratón sobre el nombre de la variable [4.1].

Para la compilación de los programas se utiliza el compilador MPLAB C30 de Microchip[4.2], el cual convierte el código fuente de lenguaje C a lenguaje ensamblador, que al ser ensamblado produce el código hexadecimal que es grabado en la memoria de

programa del dsPIC. El compilador además permite a los usuarios escribir programas en modo mixto C y ensamblador, los cuales convierte a un único archivo ejecutable. Las librerías estándar del compilador incluyen funciones para manipulación de cadenas, asignación dinámica de memorias, conversión de datos y funciones matemáticas. Además, el compilador incluye un potente programa interprete de líneas de comandos, permitiendo que los programas de aplicación se creen, se ensamblen y se conecten en un sólo paso. El proceso se ilustra en la figura 4.1.

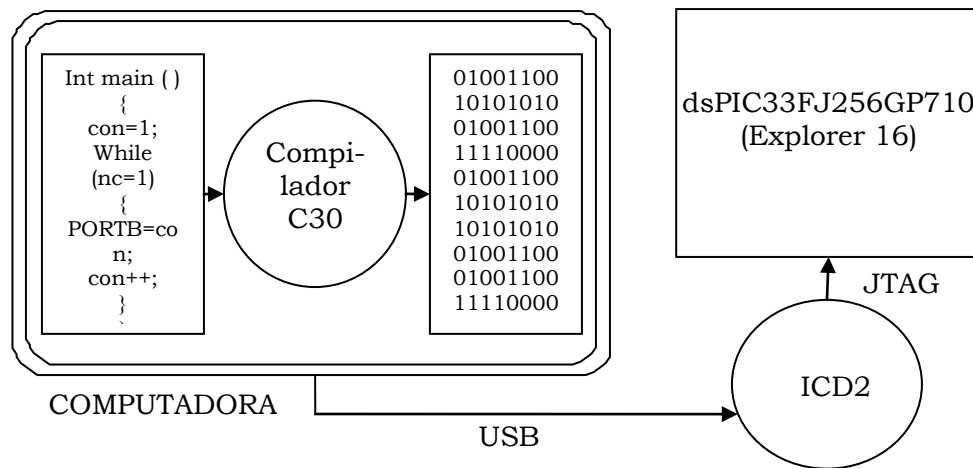


Figura 4.1. Esquema básico para programar un dsPIC33F

Existen varios programadores para los microcontroladores PIC y los dsPIC, entre ellos MPLAB PM3, PICKIT 2, PRO MATE II y MPLAB ICD2. La tarjeta de desarrollo EXPLORER 16 utiliza el programador MPLAB ICD2, el cual se compone de un programador serie (In-circuit serial programmer) y un depurador en circuito (In-circuit Debugger) [4.3]. MPLAB ICD2 soporta en tiempo real la visión de variables y registros, se pueden colocar puntos de ruptura en la memoria de programa además de que la memoria RAM de datos puede ser leída o escrita. El MPLAB ICD 2 también puede ser usado para programar o reprogramar el microcontrolador y realizar el borrado de la memoria de programa con verificación. El programador se conecta a la computadora por medio del puerto serie o por medio del puerto USB y la conexión hacia los microcontroladores.

Otro software que se utilizó es LabVIEW (Laboratory Virtual Instrument Engineering Workbench) el cual es un sistema de programación de propósito general, con extensas librerías de funciones y herramientas de desarrollo, similar a cualquier otro sistema moderno para programar en C, Basic, etc. LabVIEW además incluye librerías de funciones diseñadas específicamente para realizar: adquisición de datos, análisis de señales, control de instrumentos con interfaces paralelo, serie y GPIB, presentación de información y almacenamiento de datos. Como en otros sistemas de programación LabVIEW también incluye herramientas de desarrollo, tales como puntos de paro, ejecución de paso, y animación para observar gráficamente el flujo de datos. El

ambiente de desarrollo de LabVIEW permite crear programas en el lenguaje gráfico G, cuya principal diferencia con otros lenguajes tradicionales es que en estos la programación se basa en líneas de texto para crear el código fuente, mientras que en el G el código se genera en forma de diagramas a bloques, lo cual elimina los detalles de sintaxis. Los programas escritos en el lenguaje gráfico G de LabVIEW se llaman instrumentos virtuales porque su apariencia y operación pueden imitar aun instrumento real. Los subinstrumentos virtuales son similares a las subrutinas o funciones de otros lenguajes convencionales. Un Instrumento Virtual (VI) consiste de tres elementos fundamentales: Panel Frontal, Diagrama a Bloques, e Icono/Conector. En las siguientes líneas se describen estos tres elementos [4.4, 4.5]:

Panel Frontal.

Es una interfase interactiva con el usuario que simula el panel de un instrumento real. El Panel Frontal de un VI es una combinación de controles (entradas del usuario por medio del teclado y el ratón, en forma de perillas, botones, valores numéricos, etc), e indicadores (salidas del programa en forma de lecturas digitales, gráficos, tablas, etc). Los controles proporcionan datos al Diagrama a Bloques del VI, los indicadores muestran datos medidos ó generados por el Diagrama a Bloques.

Diagrama a Bloques.

Es un diagrama de iconos ó nodos interconectados por flujos de datos y representa el código fuente del VI en lenguaje gráfico G. El Diagrama a Bloques se construye alambando objetos que envían o reciben datos, realizan funciones específicas y controlan el flujo de ejecución. Los objetos del panel frontal tienen una correspondiente terminal en el diagrama a bloques, de tal manera que se pueden pasar datos entre el usuario y el programa. Los objetos del diagrama a bloques son fundamentalmente de tres tipos: nodos, terminales y alambres. Los nodos son elementos de ejecución del programa y son análogos a las sentencias, operadores, funciones, y subrutinas en otros lenguajes convencionales. Las terminales son los puntos de entrada y salida con el panel frontal, cada control en el panel tiene su correspondiente terminal de entrada en el diagrama y cada indicador tiene su terminal de salida en el diagrama. Los alambres son trayectorias de datos entre nodos y terminales. El flujo de datos de un nodo a otro establece el orden de ejecución de un programa escrito en G. Un nodo se ejecuta sólo cuando todas su entradas han arribado y proporciona datos a sus terminales de salida cuando termina su ejecución.

Icono/Conector.

Permiten que un VI sea llamado por otro VI de mayor jerarquía. Un VI que se encuentre dentro del diagrama de otro VI de mayor nivel es llamado subinstrumento virtual (subVI), y es un equivalente en G de las funciones o subrutinas de otros lenguajes. Los controles e indicadores de un subVI reciben y envían datos con el diagrama del VI de llamada. El icono es una representación pictórica de un VI dentro del diagrama de otro VI. El conector es un conjunto de

terminales que corresponden a los controles (entradas) e indicadores (salidas) del subVI. El conector es similar a la lista de parámetros en una función o subrutina, donde las terminales del conector actúan como parámetros. Cada terminal del conector corresponde a un particular control ó indicador del panel del VI. El conector recibe datos en sus terminales de entrada y pasa datos al código del subVI a través de sus controles, ó bien, recibe resultados en sus terminales de salida a través de los indicadores en el panel del subVI.

4.2 PROGRAMA EN EL DSPIC (ESAMBLADOR)

Este programa tiene como principal objetivo aplicar la técnica de muestreo equivalente generando un retardo Δt de la señal de reloj que controla el muestreo del convertidor A/D (CKRET en la figura 3.8). Las 3 líneas de retardo DS1020-A15 se programan de manera individual con el fin de producir un incremento en el retardo de 0.15ns cada vez que se programan. Dado que con cada transición positiva de la señal de reloj retardada se lleva a cabo una conversión en el MAX12528, las conversiones realizadas en cada reprogramación de las líneas de retardo corresponden a un punto en la señal que se encuentra 0.15 ns del punto anterior medido.

Después de que se han programado las líneas de retardo se sincroniza el sistema con el pulso SINC y se espera que transcurran 8 pulsos del reloj retardado para leer el dato. El dato se lee por el puerto D del dsPIC, se separan los dos bytes de información y se almacenan en arreglos. La reprogramación de los retardos se realiza nuevamente hasta que han completado un número específico de mediciones (nc) en el mismo punto (en el mismo Δt). El valor de nc lo selecciona el usuario desde el panel frontal de LabVIEW. Después de que se han capturado y guardado las nc mediciones son enviadas a LabVIEW para ser promediadas graficar el promedio del punto medido. La comunicación entre el dsPIC y LabVIEW se establece por medio del módulo UART del dsPIC.

Antes de que el programa ejecute la técnica de muestreo en tiempo equivalente se configuran los puertos a utilizar, el temporizador y el modulo UART. El puerto D es configurado como entrada puesto que por este puerto son capturados los datos enviados por el CAD. El puerto A es configurado como salida pues los pines RA10 (datos serie), RA11 (reloj serie), RA9, RA14 Y RA15 (enable's) sirven para programar los retardos en la base de tiempos. La línea RB2 del puerto B es programada como entrada pues en esta se verifica la llegada del pulso de sincronía. El contador del temporizador 2 del dsPIC detecta los pulsos por el pin RC1 por lo que este debe ser configurado como entrada. El modulo UART es configurado para que la comunicación serie se realice con 8 bits de datos, 1 bit de paro, sin paridad y con una velocidad de transmisión de 9600 baudios. Después de concluida la configuración del modulo UART1, se establece comunicación con la PC a través de LabVIEW para que sea enviado al dsPIC el valor de nc que como ya se ha mencionado anteriormente es el numero de veces que se mide en un mismo punto. Un arreglo de tamaño nc es generado para que sean guardados los datos leídos por el puerto D provenientes del CAD.

El algoritmo del programa que ejecuta el dsPIC se puede describir como se presenta en las siguientes líneas:

Programa MTE.c

nc = numero de conversiones en cada punto.

Configurara los recursos utilizados del dsPIC33FJ256GP710

Programa las tres líneas de retardos en cero.

Espera valor nc que es enviado desde LabVIEW.

Crea dos arreglos de tamaño nc.

Repite para i=0 hasta i=2

 k=1.

 Repite mientras k<256

 Repite para t=0 hasta t=nc

 Espera pulso de sincronía

 Espera a que temporizador 2 cuente 8 pulsos de CKRET

 Lee dato por el puerto D

 Separa dato en dos bytes

 Guarda bytes de dato leído en arreglos.

 Espera comando desde LabVIEW

 Envía arreglos de datos a labVIEW

 Si i=0

 “programaretardo” A con el valor k

 Si i=1

 “programaretardo” B con el valor k

 Si i=2

 “programaretardo” C con el valor k

 k= k+1

Fin.

En la rutina principal el temporizador 2 es inicializado en 0, después se espera por la subida del pulso de sincronía mediante la verificación continua del pin RB2. Inmediatamente después de que se ha detectado el pulso de sincronía se activa el temporizador 2 y cuando ha contado los 8 pulsos de latencia se captura el dato que envía CAD, el cual corresponde a la señal en un tiempo Δt después de la medición anterior. El dato es leído por el puerto D y es guardado en un arreglo. Como se ha mencionado anteriormente el puerto D solo tiene 13 pines disponibles, de los cuales 12 están conectados a las salidas del puerto A del CAD MAX12528, sin embargo cuando se lee el puerto D se lee el valor de las 16 líneas, dado que el modulo UART solo puede enviar bytes los valores guardados en el arreglo son divididos en 2 bytes para enviarlos a LabVIEW.

Después de que se han realizado las nc mediciones de un mismo punto se reprograman lo retardos, para esto se hace uso de la rutina “programaretardo” además de dos variables i y k, la variable k se incrementa desde 1 hasta 255 y sirve como parámetro

para la rutina, cuando esta llega a su valor máximo i se incrementa en una unidad, lo que significa que se comenzará a programarse el siguiente retardo, el retardo anterior queda además programado en su valor mas alto (255) y el la variable k se coloca nuevamente en 1. El proceso anterior se repite 3 veces lo que permite programar a los tres retardos de manera consecutiva. De esta manera se pueden obtener 766 valores distintos para programar a los retardos obteniendo así un intervalo en tiempo de 114.9 ns. Los retardos se programan por medio de la subrutina llamada “programaretardo”, la cual se describe a continuación.

k = valor de 8 bits para programar retardo

Poner a nivel alto la línea “habilitar”

Repetir para $m=0$ hasta $m=7$

Poner a nivel bajo reloj (RB3)

Seleccionar el $(8-m)$ bit del valor k

Colocar el valor del bit seleccionado en la salida “programa”

Poner a nivel alto reloj (RB3)

Poner a nivel bajo la línea “habilitar”

4.3 PROGRAMA DE CONTROL EN LABVIEW.

El panel de control del programa desarrollado en el lenguaje gráfico G de LabVIEW se muestra en la figura 4.2. Este programa inicia el programa que se ejecuta en el dsPIC al enviar por el puerto serie el numero de mediciones (nc) que se quieren realizar en cada punto. Posteriormente envía comandos al dsPIC para que éste envíe los datos codificados resultantes de las conversiones A/D. LabVIEW recibe dos bytes por cada conversión realizada pero solo doce de los 16 bits contienen la información enviada por el convertidor así que se desarrolló una rutina de selección en la que se descartan los dos últimos bits del primer byte (RD6 Y RD7) y el antepenúltimo y último bit del segundo byte recibido. Los bits resultantes forman un dato de 12 bits que esta codificado en complemento a dos por lo que se desarrolló otra rutina que lo convierte a un número decimal tomando (Sección 3.2): $V_{REF_P}=2.418V$ y $V_{REF_N}=.882V$. Así al recibir el valor 0X7FF corresponde a 1.0235V y 0X800 a -1.024V. Después de que se han recibido las nc mediciones del mismo punto en LabVIEW, éstas se promedian y se grafica el valor resultante, este proceso se repite 766 veces.

El algoritmo del programa de control desarrollado en LabVIEW se muestra a continuación:

Programa MTE.vi

DATOS=arreglo que contiene los 766 valores medidos (ya promediados)

Espera a que se pulse el botón inicio.

Limpia la grafica.

Configura el puerto serie.

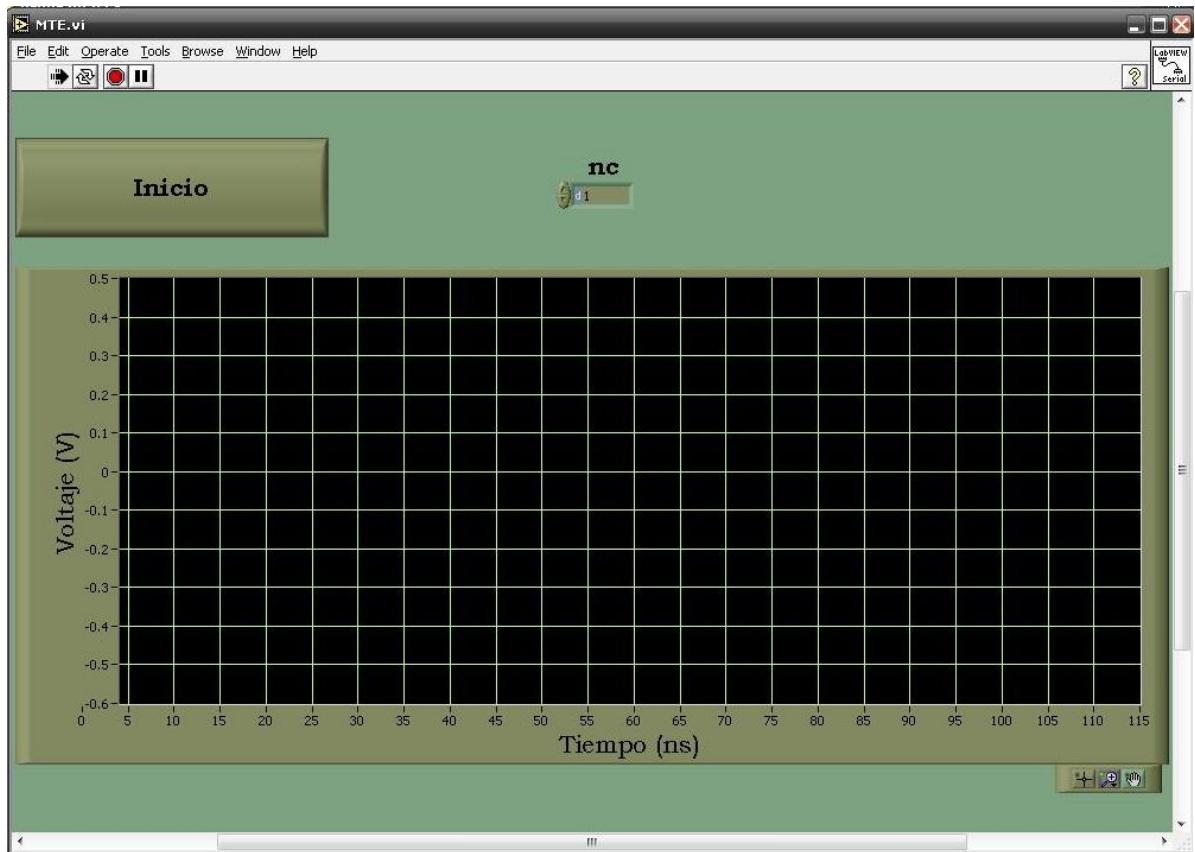


Figura 4.2. Panel de control del programa desarrollado en labVIEW, para comenzar la rutina de muestreo en tiempo equivalente es necesario especificar el número de conversiones por punto y después presionar la tecla inicio.

Envía nc al dsPIC.

Repite para $i=0$ hasta $i=765$

Envía comando a dsPIC para comenzar envío de bytes.

Lee $2 \cdot nc$ bytes provenientes del dsPIC.

Elimina bits innecesarios de cada byte.

Une los bytes resultantes correspondientes para recobrar los datos de 12 bits provenientes del CAD MAX12528.

Asigna un valor decimal a cada uno de los nc datos resultantes.

Promedia los nc datos.

Grafica el punto.

Guarda punto en la entrada i del arreglo DATOS.

Guarda arreglo datos en un archivo de texto.

Fin.

Como ejemplo de un diagrama a bloques en LabVIEW, en la figura 4.3 se muestra un fragmento del programa de control escrito en el lenguaje gráfico G, que corresponde a la secuencia 2(0..3). Dentro de esta secuencia se tiene una sentencia for que se repite 766

veces, una para cada incremento en Δt . Dentro del for se tiene la secuencia 2(0..2), la cual a su vez contiene una sentencia for que se repite el nc veces. El subinstrumento `eliminabit.vi` convierte los datos recibidos desde el dsPIC a un arreglo de elementos boléanos para después eliminar el 5 elemento de más peso y convertir el arreglo resultante a un numero entero sin signo. El subinstrumento `binariodec.vi` utiliza las ecuaciones de la sección 3.2 para asignar un valor decimal a los datos obtenidos.

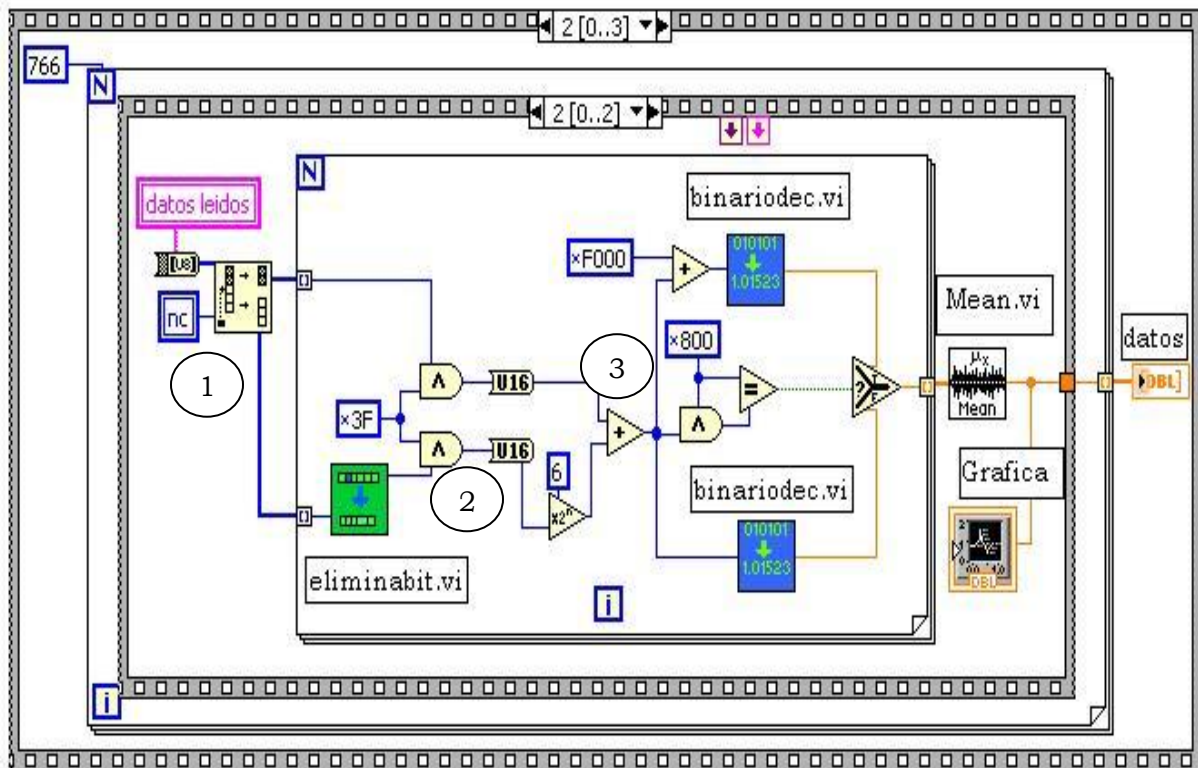


Figura 4.3. Diagrama a bloques de un fragmento del programa desarrollado en labVIEW. Sólo se muestra la parte en la que los datos enviados por el dsPIC son convertidos a un número decimal.

En la figura 4.4 se muestra el proceso de la reconstrucción del dato enviado por el convertidor analógico digital y la forma en que se encuentran los bits en cada proceso realizado en labVIEW (indicado en los círculos).

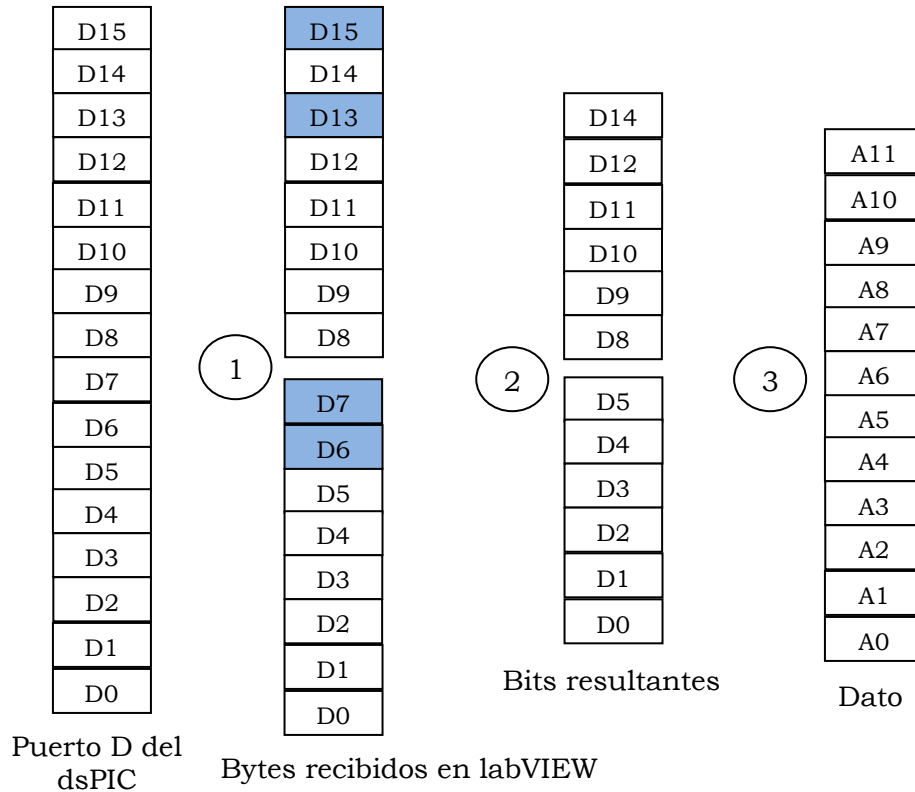


Figura 4.4. Proceso de reconstrucción del dato enviado por el convertidor MAX12528. Los números en los círculos corresponden a los procesos marcados en la figura 4.3

REFERENCIAS

- [4.1] Microchip, “MPLAB IDE, Quick start Guide”, Microchip Technology Inc., 2006.
- [4.2] Microchip, “MPLAB C30 C Compiler User’s Guide”, DS51284, Microchip Technology Inc., 2006.
- [4.3] Microchip, “MPLAB ICD 2 In-Circuit Debugger, User’s Guide”, Microchip Technology Inc., 2005.
- [4.4] National Instruments, “LabVIEW: Graphical Programming for instrumentation, Users Manual”, National instruments Corp., Edición 1996, Austin, TX, USA.
- [4.5] National Instruments, “G, Programming Reference Manual”, National instruments Corp., Edición 1996, Austin, TX, USA.

CAPÍTULO 5

MEDICIONES Y RESULTADOS

En esta sección se presentan los resultados obtenidos con el digitalizador al medir pulsos de prueba con tres distintos anchos de pulso: 5ns, 10ns y 15 ns. Se hace una comparación entre los pulsos medidos por el sistema y medidos con un osciloscopio de 300MHz de ancho de banda, también se muestra el efecto de las puntas utilizadas para medir.

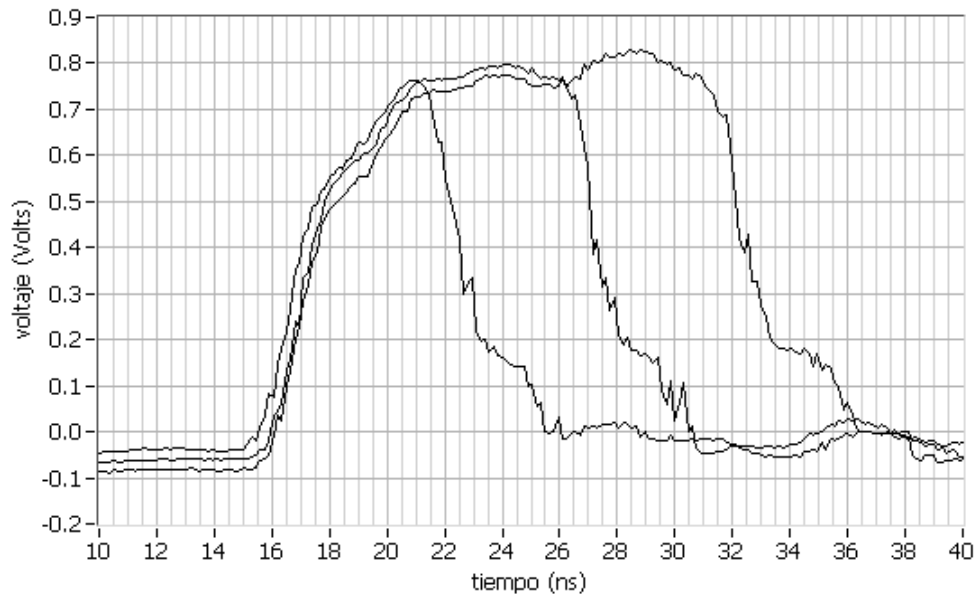
5.1 MEDICIONES Y RESULTADOS.

El sistema se probó exitosamente en la medición de pulsos de voltaje en el dominio del tiempo. Se usaron tres pulsos generados por un circuito monoestable programable DS1040Z-A15 con un FWHM (ancho a la mitad del máximo) de 5ns, 10ns y 15ns $\pm 5\%$ y 4ns de tiempo de subida [5.1]. El pulso de sincronía de la base de tiempos sincroniza al monoestable en una velocidad de repetición de 40KHz. La figura 5.1a muestra los pulsos de prueba medidos sin promediar. La figura 5.1b muestra los mismos tres pulsos después de promediar 10 mediciones por punto. Se observan trazos más suaves debido a la supresión de ruidos.

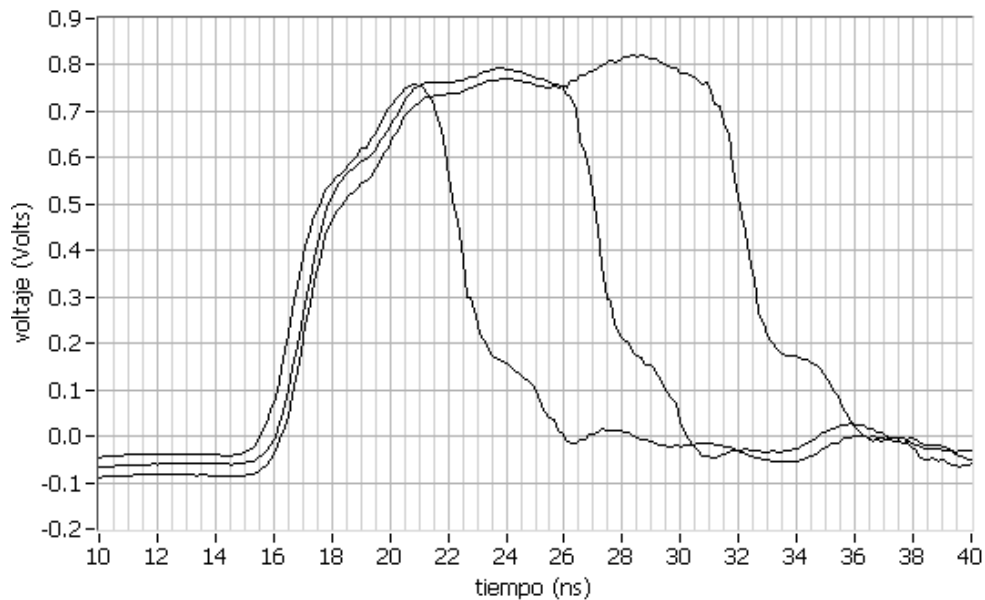
En la figura 5.1b se observa que el tiempo de subida de los tres pulsos es de aproximadamente 4ns (del 10% al 90% de la amplitud) y el FWHM de los valores medidos corresponde a los valores esperados. El tiempo de subida de la señal de entrada es $tr(sig)=4ns$ y el tiempo de subida del digitalizador puede ser calculado con la ecuación 2.4:

$$Tr(dig) = \frac{.35}{Bw(dig)} = \frac{.35}{750MHz} = .466ns$$

Con estos valores en la ecuación 2.5 el tiempo de subida medido para el digitalizador es de 4.02ns, esto significa que es 0.5% más grande que el tiempo de subida de la entrada. Este error de 0.02ns es menor que la resolución del sistema de 0.15ns.



a) mediciones sin aplicar la técnica de supresión de ruido



b) mediciones con un promedio de 10 muestras por punto.

Figura 5.1. Tres pulsos de prueba generados por el circuito monoestable programable DS1040Z-A15

La figura 5.2 muestra el pulso de prueba de 15ns medido con el sistema digitalizador desarrollado (línea continua), con una punta de conexión sin atenuador y un osciloscopio (línea cortada) y con una punta atenuada y un osciloscopio (línea punteada). Las mediciones realizadas con el digitalizador tienen un FWHM de 15ns (en el nivel de 0.4V) y cerca de 4ns de tiempo de subida. Los valores medidos corresponden

a los que están especificados en la hoja de datos del DS1040Z-A15 [5.1] además se observa un pulso bien definido. La atenuación y la distorsión producida por las puntas de prueba y el osciloscopio sobre la señal medida se observan claramente en la figura 5.2. El pulso medido con la punta no atenuada y el osciloscopio tiene un FWHM de sólo 12ns y un tiempo de subida más grande de 4.5ns.

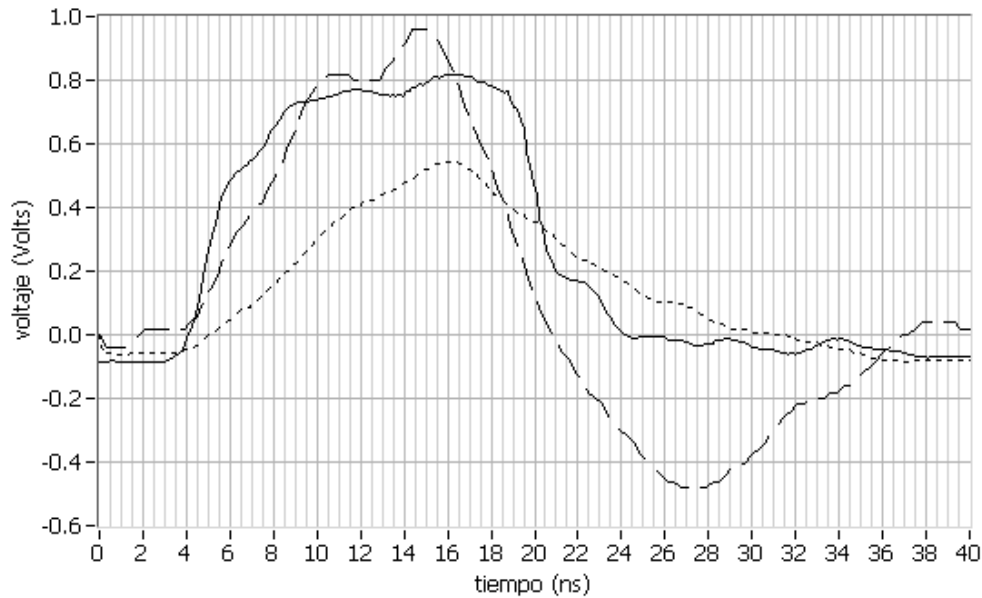


Figura 5.2. Pulso de prueba con un FWHM de 15ns y 4ns de tiempo de subida medido con el digitalizador (línea continua), con una punta de prueba no atenuada y un osciloscopio (línea cortada) y con una punta atenuada y un osciloscopio (línea punteada).

El osciloscopio usado es un osciloscopio digital Tektronix, modelo 2440, con una frecuencia máxima de muestreo de 500Ms/s y un ancho de banda de 300MHz. El tiempo de subida equivalente del osciloscopio puede ser calculado de la ecuación 2.4: $tr(osc)=1.66ns$, el cual es 2.5 veces más grande que el del digitalizador y se produce un aumento en el tiempo de subida en la entrada del osciloscopio, tal y como se puede observar en la figura 5.2. El pulso medido con la punta atenuada y el osciloscopio tiene un valor de FWHM de 12ns y un tiempo de subida de 8ns. La punta atenuada causa alrededor de 30% de atenuación de la amplitud de la señal, además de que el pulso no está bien definido. Esto es debido a que la punta usada tiene un ancho de banda de sólo 100MHz y un tiempo de subida equivalente de 3.4ns, estos valores no son suficientes para reproducir las componentes de alta frecuencia del pulso de entrada.

La figura 5.3 muestra el pulso de prueba medido de 10ns de FWHM. Se observa que el pulso medido con el digitalizador reproduce el valor de FWHM y el tiempo de subida del pulso de entrada. El pulso medido con el osciloscopio tiene un FWHM cercano al del pulso de entrada pero con un tiempo de subida de 4.5ns. El pulso medido con la punta atenuada es ahora atenuado alrededor del 50%. La figura 5.4 muestra las mediciones realizadas de un pulso de prueba de 5ns de FWHM con resultados similares a los mencionados anteriormente.

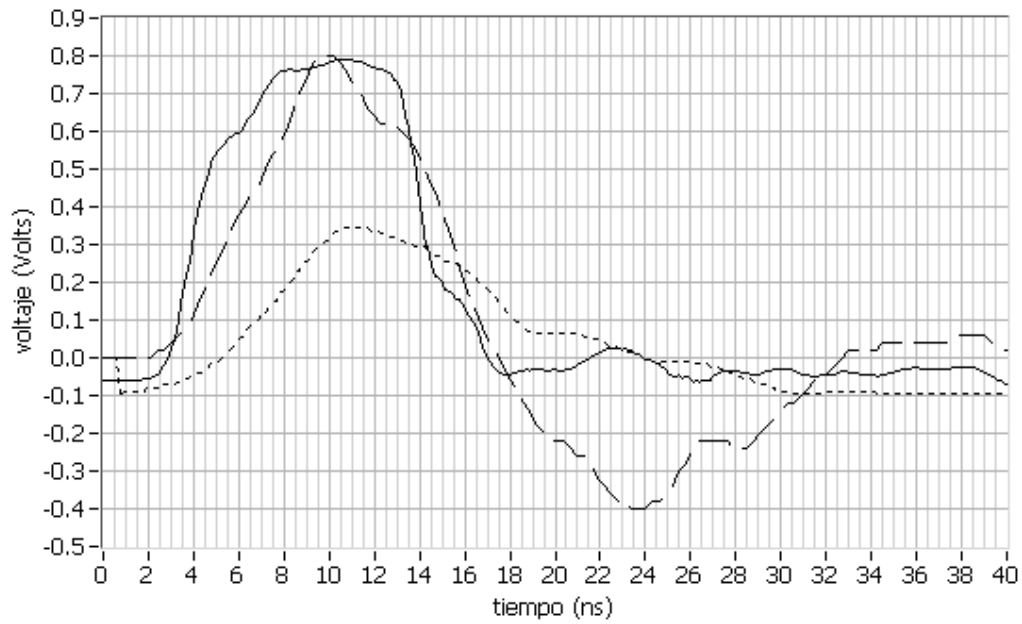


Figura 5.3. Pulso de prueba con un FWHM de 10ns y 4ns de tiempo de subida medido con el digitalizador (línea continua), con una punta de prueba no atenuada y un osciloscopio (línea cortada) y con una punta atenuada y un osciloscopio (línea punteada).

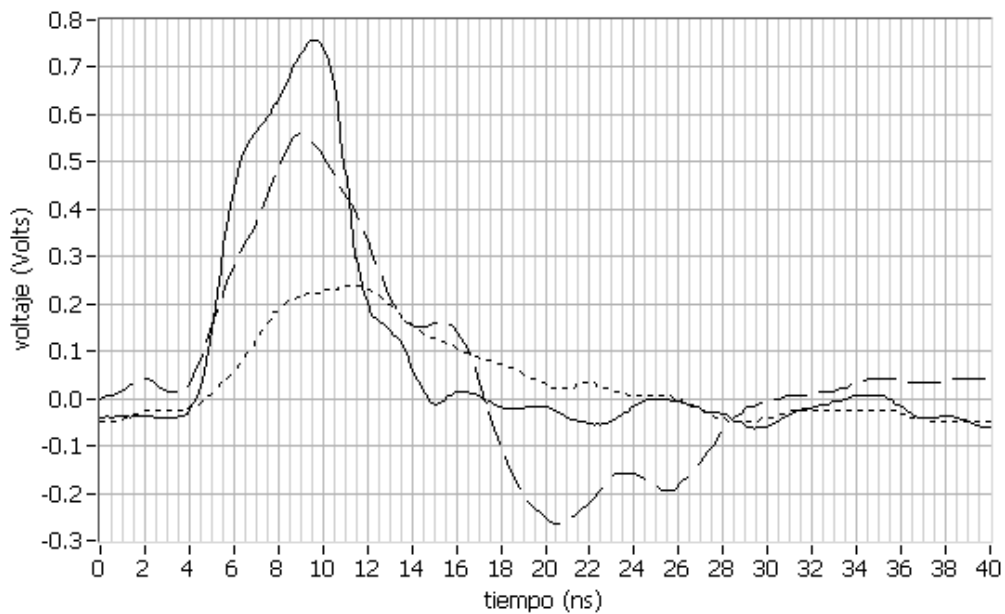


Figura 5.4. Pulso de prueba con un FWHM de 5ns y 4ns de tiempo de subida medido con el digitalizador (línea continua), con una punta de prueba no atenuada y un osciloscopio (línea cortada) y con una punta atenuada y un osciloscopio (línea punteada).

La figura 5.5 muestra la bajada del pulso de reloj de 5.068MHz sin promediar (línea continua) y con un promedio de 10 muestras por punto (línea cortada). Claramente se puede observar la reducción del ruido debido a la promediación de 10 valores en cada

punto. En la fotografía 5.1 se presenta el prototipo del sistema digitalizador desarrollado después de medir el pulso que se indica en el osciloscopio.

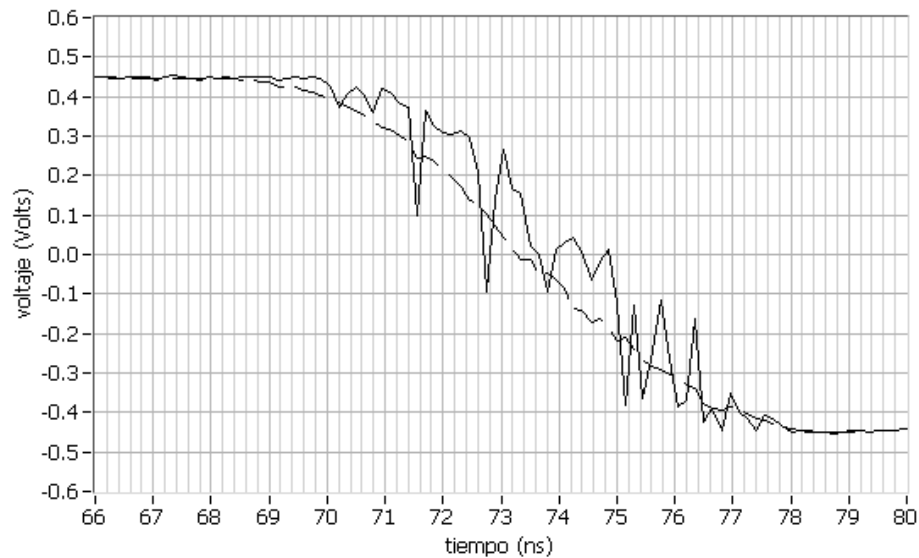
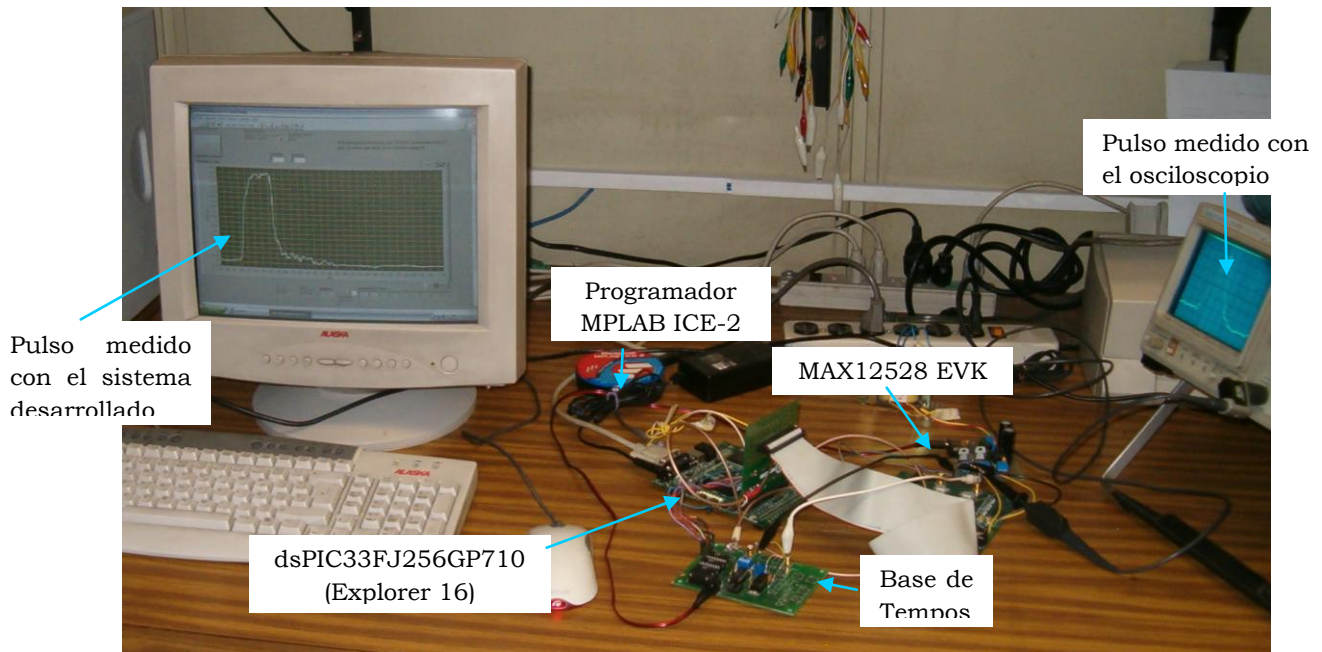


Figura 5.5. Bajada del pulso de reloj de 5.058MHz sin promediar (línea continua) y con un promedio de 10 muestras por punto (línea cortada)..



Fotografía 5.1. Sistema digitalizador desarrollado donde se observa el pulso medido

REFERENCIAS

- [5.1] Maxim/Dallas semiconductor, "DS1040 Programmable One-Shot Pulse Generator", Data Sheet.

CAPÍTULO 6

CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se presentan las conclusiones a la que se llegaron después de haber realizado mediciones con el sistema digitalizador. También se hace referencia a los posibles trabajos que pudieran derivarse del sistema desarrollado.

6.1 CONCLUSIONES.

En este trabajo se describió el desarrollo de digitalizador de alta velocidad. Todos los componentes fueron seleccionados de tal manera que la interface entre ellos y la computadora se realizara en forma práctica y sencilla. El prototipo es fiable y útil para resolver pulsos analógicos de hasta 1ns de tiempo de subida con una resolución de 150ps. El digitalizador desarrollado resuelve los 4ns de tiempo de subida de los pulsos de prueba generados en la base de tiempos con un error de 0.5%. Los pulsos de prueba fueron medidos además con un osciloscopio de 300MHz de ancho de banda, con una punta de prueba atenuada con un ancho de banda de 100MHz y con otra punta sin atenuador.

Las mediciones reportadas muestran claramente que el digitalizador desarrollado es capaz de resolver la subida de los pulsos mientras que las mediciones obtenidas con el osciloscopio junto con la punta no atenuada difieren de lo valor de tiempo de subida especificados en la hoja de datos del monoestable DS1040-A15. Esto es debido a que el osciloscopio con un ancho de banda de 300MHz no puede reproducir la componente más grande de frecuencia de los pulsos de prueba. Con la punta atenuada los puntos medidos están más distorsionados debido a su pequeño ancho de banda. El digitalizador desarrollado tiene un ancho de banda en su entrada analógica de 750MHz el cual produce un error en la amplitud alrededor del 9% para una señal con un ancho de banda de 350MHz o 1ns de tiempo de subida equivalente. La técnica de muestreo en tiempo equivalente es funcional y su algoritmo permite agregar un método de supresión de ruidos simple. La resolución y el ancho de banda del digitalizador desarrollado cumplen con los requerimientos necesarios para ser aplicado en mediciones de pulsos de fluorescencia.

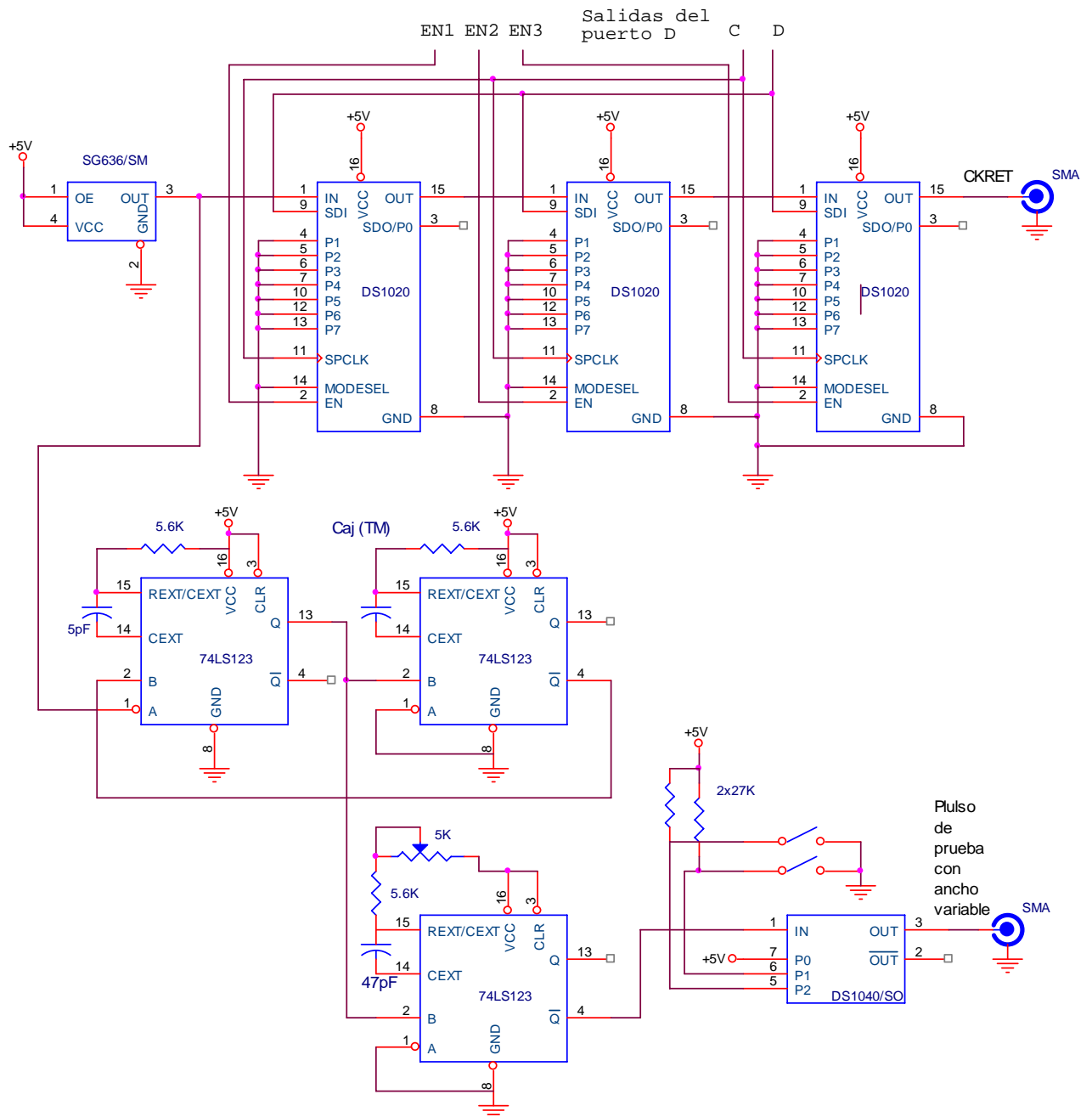
6.2 TRABAJOS FUTUROS

El sistema desarrollado permite la ampliación del rango temporal de medición al cambiar las líneas de retardo, por lo que se podrían medir pulsos de mayor duración, teniendo en cuenta que se pierde resolución.

El sistema digitalizador será utilizado para resolver en el dominio del tiempo pulsos de fluorescencia inducida por láser (LIF).

APÉNDICE A

DIAGRAMA ESQUEMÁTICO DE LA BASE DE TIEMPOS.



APÉNDICE B

PROGRAMA EN LENGUAJE ENSAMBLADOR GRABADO

EN EL DSPIC33FJ256GP710.

```
/* programa MTE.c*/

#include "p33FJ256GP710.h"

#define FCY 40000000
#define BAUDRATE 9600
#define BRGVAL ((FCY/BAUDRATE)/16)-1

_FOSCSEL(FNOSC_PRIPLL);
_FOSC(FCKSM_CSDCMD & OSCIOFNC_OFF & POSCMD_XT);
_FWDT(FWDTEN_OFF);

//esta función envía los bits para programar el retardo DS1020

void enviobits(int r)
{
    int auxiliar;
    int m;
    for (m=0; m<8; m++)
    {
        PORTBbits.RB3=0;           //se pone a 0 el reloj
        auxiliar=(r>>(7-m));       //se selecciona el bit a enviar
        auxiliar=(auxiliar)&0x0001; //se coloca en auxiliar el bit a enviar
        PORTAbits.RA10=auxiliar;    //se envía bit
        Nop();
        PORTBbits.RB3=1;           //se pone a 1 el reloj
    }                               //se acaba for de envío de bits
}                                 //se acaba la función

int main(void)
{
    // Se configura el oscilador para trabajar a 40MHz
    // Fosc= Fin*M/(N1*N2), Fcy=Fosc/2
    // Fosc= 8M*40(2*2)=80Mhz para 8MHz de entrada

    PLLFBD=38;                    // M=40
    CLKDIVbits.PLLPOST=0;         // N1=2
    CLKDIVbits.PLLPRE=0;         // N2=2
    RCONbits.SWDTEN=0;           // Deshabilita Watch Dog Timer
```

```

while(OSCCONbits.LOCK!=1){};    // espera a que se establezca el oscilador

    //Se configura el modulo UART2 para trabajar sin paridad
    //1 bit de paro, datos de 8 bits y baud rate de 9600.

U2MODE=0;
U2BRG = BRGVAL;                //Baud rate de 9600
U2STA=0;
U2MODEbits.UARTEN = 1;        //Habilita el modulo UART
U2STAbits.UTXEN = 1;          //Habilita transmisión UART Tx


    //Configuración del temporizador 2
    //cuenta los pulsos del reloj retardado

T2CON=0;
T2CONbits.TCS=1;              // Reloj externo
PR2 = 500;                    // Se carga el periodo a contar


    //Configuración de puertos

AD1PCFGH=0xFFFF;            //se habilita a los pines que tienen la subfunción
AD1PCFGL=0xFFFF;            //de entradas analógicas para trabajar como
AD2PCFGL=0xffff;            //entradas digitales

TRISA= 0x3900;                //RA0-RA7 LEDS; RA9 línea enable1; RA14 enable2;
                                //RA15 enable 3; RA10 envío de bits de prog
TRISB=0xFFF5;                //RB3 línea CLOCK y RB2 entrada de sincronía
TRISC=0xFFFF;                //RC1 entrada T2CK
TRISD=0xFFFF;                //puerto de lectura de datos de CAD


    //Variables necesarias.

int i, j, k, nc, t, lectura, M;

while(1)

{

    //Inicialización de variables

    i=k=nc=t=lectura=0;


    //Se programa los retardos en ceros

    PORTAbits.RA9=1;          //se pone a uno la línea "enable" 1
    enviobits(0);              //se envían los bits
    PORTAbits.RA9=0;          //se pone a cero la línea enable 1

    PORTAbits.RA14=1;         //se pone a uno la línea "enable" 2
    enviobits(0);              //se envían los bits

```

```

PORTAbits.RA14=0;          //se pone a cero la linea enable 2

PORTAbits.RA15=1;          //se pone a uno la linea "enable" 3
enviobits(0);              //se envian los bits
PORTAbits.RA15=0;          //se pone a cero la linea enable 3

    //Espera al menos 1/9600 antes de enviar el
    //primer dato.

    for(j = 0; j < 4160; j++)
    {
        Nop();
    }
    //Se espera recibo nc desde labVIEW

while (U2STAbits.URXDA == 0);
    nc = U2RXREG;
    int medicion[nc];
    int mediciona[nc];
    int auxiliar;
    int m;

    for (i=0; i<3; i++)
    {
        k=1;
        while (k<256)
        {

            //Comienza la captura de datos del convertidor

            for(t=0; t<nc; t++)
            {

                TMR2 = 0x00;          //Se limpia registro contador

                while (PORTBbits.RB2==1); //se espera el pulso de sincronía
                while (PORTBbits.RB2==0);

                T2CONbits.TON = 1;      // Se enciende el temporizador
                while (TMR2<6);         // Se espera a que el conteo llegue a 8

                //Se separa el dato leído en dos bytes para ser enviado

                medicion[t]=PORTD;       // se adquiere la medición del convertidor
                mediciona[t]=medicion[t]>>8; //se separa la palabra de 16
                medicion[t]=medicion[t]&0x00FF; //bits en dos bytes
                T2CONbits.TON = 0;       //Se apaga el temporizador 2
            }                           //Fin de for de nc

            while (U2STAbits.URXDA == 0); //Se espera comando desde LabVIEW
            M = U2RXREG;                 //para comenzar a enviar bytes

```

```

for (t=0; t<nc; t++)                                //Envío de datos
{
    while(U2STAbits.TRMT==0);
    U2TXREG = medicion[t];                            //Se envía la parte baja de la palabra
                                                    //de 16 bits
for (t=0; t<nc; t++)
{
    while(U2STAbits.TRMT==0);                        //Se envía la parte alta de la palabra
    U2TXREG = mediciona[t];                          //de la palabra de 16 bits leída.
}

// Se incrementan variables para programar los retardos

if(i==0)
{
    PORTAbits.RA9=1;    //Se pone a uno la línea "enable" 1
    enviobits(k);       //Se envían los bits
    PORTAbits.RA9=0;    //Se pone a cero la línea enable 1
}

if(i==1)
{
    PORTAbits.RA14=1;   //SE PONE A UNO LA LINEA "ENABLE" 2
    enviobits(k);       //SE ENVÍAN LOS BITS
    PORTAbits.RA14=0;   //SE PONE A CERO LA LINEA ENABLE 2
}

if(i==2)
{
    PORTAbits.RA15=1;   //Se pone a uno la línea "enable" 3
    enviobits(k);       //Se envían los bits
    PORTAbits.RA15=0;   //Se pone a cero la línea enable 3
}

k=k+1;

} // fin de while de k<255
} // fin de for de 3 ciclos
} // fin de while principal
} // fin de main

```