

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DISEÑO Y CONSTRUCCIÓN DE UNA INTERFAZ PARA UN SISTEMA DE INSTRUMENTACIÓN POR COMPUTADORA

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO ELÉCTRICO-ELECTRÓNICO

PRESENTA:

VANIA ROJAS DÍAZ

DIRECTOR DE TESIS:

M. en I. MIGUEL ÁNGEL BAÑUELOS SAUCEDO

LABORATORIO DE ELECTRÓNICA

CENTRO DE CIENCIAS APLICADAS Y DESARROLLO TECNOLÓGICO



CD. UNIVERSITARIA, MÉXICO D.F. 2009

Universidad Nacional Autónoma de México
UNAM
Biblioteca Central
Dirección General de Bibliotecas de la UNAM

UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

AGRADECIMIENTOS

A mi director de tesis Miguel Ángel Bañuelos Saucedo por la dedicación y paciencia que tuvo para que este trabajo saliera adelante y también por compartirme su pensamiento en todas esas charlas eternas que, en mis últimos días de formación como universitaria, ampliaron mi panorama y me hicieron ver la vida desde una perspectiva distinta.

A mis padres, Alejandro y Ma. Elena, por dejarme ser, por permitirme desde pequeña tomar mis propias decisiones (inclusive las más descabelladas), por nunca reprimirme, por creer siempre en mi, por estar siempre presentes y darme soporte y alivio en los momentos más obscuros de mi vida. Lo que representa este trabajo no es un logro mio, considérenlo un logro de nuestra familia. Los amo.

A mi hermano Alex, por enseñarme el verdadero significado de lo que es perseguir un sueño. Te agradezco por todas las veces que, en silencio, te has preocupado por mí y también cuando me lo has hecho saber a través de tus consejos. Te admiro y te quiero muchísimo.

A mi abuela Olimpia, por haber ejercido tanta influencia positiva en mí, por darme mis primeras lecciones de feminismo y forjarme como una mujer fuerte, libre y sin limitaciones…gracias por tu ejemplo!. A mi abuela Conchita por darme el cariño y la ternura que solo una abuela puede darle a una nieta, por mostrarme la nobleza y la bondad en su más pura expresión. Las sigo extrañando mucho.

A mis padrinos Aracely Gutiérrez y José Suárez por darme una gran lección de fortaleza y superación. En verdad son un ejemplo de lucha. A mi tío César por ser el primero en hacerme voltear a ver las estrellas. A mis tios Antonio Díaz y Adolfo Díaz por dejarme ver que las personas justas, honestas y buenas existen.

A Hector Alaniz por todo el apoyo que me diste durante este periodo de mi vida. Por nunca darme la espalda, escucharme y siempre estar al pie del cañón. Yo no sé que fuerza hizo que nos conociéramos, pero creeme que agradeceré eternamente que haya ocurrido. Eres mi ángel.

A Sergio Quintana Thierry y a José Castillo Hernández así como a todos los compañeros del Laboratorio de Electrónica por el interés y el apoyo mostrado hacia esta tesis. Además, por dejarme tan buenos recuerdos de la convivencia que tuve con ustedes en el CCADET y por permitirme trabajar con el gran equipo que han formado.

A mis amigos de la Facultad de Ingeniería: Balam Zaldivar,Emilio García Morla , Adrián Reséndiz, Job Jiménez, Ángel Álvarez, Esteban Badillo, Ivonne Olavarrieta, Héctor Yuen, Fernando Castillo, Lisandra Jarquín, José Antonio González, Jessica Robles, Ana Laura Ortíz,

Edgar Cruz, Lourdes Martínez por el grupo tan diverso que llegamos a formar, tantas comidas compartidas, tanta risa, tantos anécdotas... Mis recuerdos en la facultad estarían incompletos sin ustedes.

A Blanca Pérez Arriaga y a Marcela Valdés Mosqueda, mis hermanitas. A Lucelly Montaño, Claudia González, Juan Antonio Jaramillo, Javier Ávila, Ramsés López, Benjamín López, Argel Ibarra, Ángel Reyes, Elban López, por que a pesar de tener tantos años de conocernos, cada vez que nos volvemos a ver tenemos la capacidad de seguirnos riendo juntos como cuando teníamos trece años.

ÍNDICE

INTRODUCCIÓN.....1

OBJETIVO.....2

1. ANTECEDENTES.....3

1.1. Situación encontrada en el sistema de instrumentación.....3

1.2. Definición del problema y análisis de necesidades.....4

1.3. Alcance del proyecto.....5

1.4. Campos de aplicación del sistema de instrumentación.....6

2. DETERMINACIÓN DE LOS REQUERIMENTOS DEL SISTEMA.....7

2.1. Acondicionamiento de una señal y multiplexado.....7
2.1.1. Multiplexado y convertidores A/D.....7
2.1.2. Amplificación de la señal y conversión A/D.....7
2.1.3. Entradas de un solo terminal o entradas diferenciales.....8

2.2. El amplificador no inversor.....9
2.2.1. Conversión analógica-digital y etapas de la conversión.....10
2.2.2. Principales tipos de convertidores analógico-digitales.....13
2.2.3. Familia de Microcontroladores PIC.....16
2.2.4. Comunicación USB.....19

3. DISEÑO Y CONSTRUCCIÓN DE LA INTERFAZ.....21

3.1. Estructura general de la interfaz.....21

3.2. Los módulos.....22

3.3. La adquisición de variables provenientes de los módulos.....24
3.3.1. Acondicionamiento de la señal.....24
3.3.2. Conversión de la señal.....25

3.4. La transmisión y recepción de datos en el microcontrolador.....28

3.5. Transmisión de datos a la computadora.....30
3.5.1. El módulo de comunicación DLPUSB245M-G.....31

3.6. Pruebas.....32
3.6.1. Pruebas de depuración.....32
3.6.2. Acoplamiento con el software.....33

4. DISEÑO DEL FIRMWARE.....36

4.1. Desarrollo del firmware.....36
4.1.1. CONTROL DE CONVERTIDORES A/D.....42
4.1.2. CONTROL DEL ENVIO DE DATOS 47
4.1.2.1. PROTOCOLO N.....47

4.2. Protocolo de comunicación entre el CAD y el microcontrolador.....51

4.3. Protocolo de comunicación entre el microcontrolador y el módulo USB.....57
4.3.1. Envío de datos a la computadora personal.....62

CONCLUSIONES.....64

BIBLIOGRAFÍA.....65

INTRODUCCIÓN

La medición ha sido una actividad sustancial para las sociedades que han habitado el planeta. Si profundizáramos en la historia, nos percataríamos de que el hombre ha pasado mucho tiempo creando técnicas para poder medir y registrar lo que ocurre a su alrededor.

La medición y el registro de las variables medidas también es fundamental para el desarrollo de la ciencia. Tradicionalmente este proceso se llegó a desarrollar de manera manual, lo cual es una labor tediosa y que regularmente trae desventajas, como puede ser el perder la atención del objetivo principal de un experimento o consumir tiempo extra.

En la actualidad, no es extraño encontrar instrumentos de medición que ayudan con esta tarea, los cuales, ponen énfasis en lograr una mayor exactitud y precisión al reproducir los resultados que obtienen. Este es el campo de estudio del cual se encarga la instrumentación electrónica, la cual, actualmente se ha visto beneficiada debido a la generalización del uso de sistemas digitales tanto en la vida cotidiana como en el ámbito científico.

La instrumentación electrónica se aplica en el sensado y procesamiento de las señales provenientes de variables físicas y químicas, a partir de las cuales se realiza el monitoreo, control y análisis de procesos, empleando los dispositivos apropiados.

Con la expansión en el uso de las computadoras personales para apoyar los ámbitos académicos, los registros de magnitudes medidas de un fenómeno en estudio pueden ser capturados en una computadora para posteriormente ser analizados dentro del mismo ambiente.

Con la intención de satisfacer lo expuesto, en el Centro de Ciencias Aplicadas y Desarrollo Tecnológico de la UNAM (CCADET), se planteó un sistema de instrumentación por computadora cuya principal misión es la de ser una herramienta para el registro y análisis de datos en tareas de apoyo a la enseñanza o bien, en pruebas sencillas de laboratorio.

La motivación que se tuvo para emprender el proyecto surge del compromiso que tiene la Universidad con la mejora de la educación. En la actualidad, se observa un progresivo aumento en el número de niños y jóvenes que crecen usando la computadora, por tanto, considerando que en el país existe la imperiosa necesidad de aumentar el número de individuos relacionados con los ámbitos científicos, es conveniente incitarlos a que, desde temprana edad, entiendan la ciencia de manera amena y aprovechando los recursos con los que están familiarizados.

En Estados Unidos, la Evaluación Nacional del Progreso Educacional (Nacional Assessement of Educational Progress) realizada por el Centro Nacional de

Estadísticas de la Educación (Nacional Center for Education Statistics), llega a la conclusión de que, los alumnos que utilizan este tipo de herramientas, tienen resultados más altos cuando se les evalúa en tópicos como son: la capacidad de razonamiento de los procesos de ciencia, la integración de lo comprendido a una cadena más larga de conocimiento y, como es de esperarse. la habilidad al usar los instrumentos.

Existen algunos instrumentos didácticos desarrollados fuera del país que cubren las expectativas de este tipo de sistemas de instrumentación, sin embargo, los costos que implica la importación de estos y el poder adquisitivo que presenta la moneda mexicana provocan que la compra de los instrumentos por este medio no resulte una alternativa viable. Por esto, resulta conveniente la implementación de estas herramientas en el país utilizando dispositivos que cumplan con las características requeridas, cuyo costo sea accesible y que su disponibilidad sea alta en el mercado nacional. Con esto se logrará aminorar su costo de producción y lograr un mantenimiento realizable de manera sencilla sin tener por esto que sacrificar calidad en sus funciones.

OBJETIVO

Diseñar y construir una interfaz con el propósito de mejorar un sistema de instrumentación didáctica por computadora utilizado para el registro y el análisis de variables físicas y químicas.

1. ANTECEDENTES

1.1. Situación encontrada en el sistema de instrumentación

Como se mencionó en párrafos anteriores, inicialmente, en el CCADET surgió la iniciativa de crear un sistema encargado de realizar la medición y análisis de variables físicas y químicas con el propósito de comercializarlo en centros de enseñanza tanto internos como externos a la UNAM. En el proyecto inicial se elaboró un producto que cubría las expectativas para aquel entonces, no obstante, en el presente, algunos de los dispositivos que conformaron el sistema necesitan ser mejorados de manera sustancial.

Cabe mencionar que el trabajo propuesto se considera como parte de un proyecto conjunto del Laboratorio de Electrónica del CCADET en donde, de manera paralela, se estará desarrollando una nueva versión de software específico para el sistema. El proyecto pretende restituir el sistema de instrumentación para que vuelva a ser una alternativa económica y eficiente en el campo de la enseñanza en laboratorios de ciencia y que en cierto momento logre ser promovido para su fabricación comercial.

Para lograr este propósito, se comenzó identificando las partes que integran al sistema que a continuación, de manera general, se menciona.

El sistema de instrumentación cuenta con un conjunto de módulos intercambiables. Cada módulo permite medir una variable diferente: Voltaje, Corriente, Resistencia, capacitancia, Inductancia, Frecuencia, Intensidad luminosa, Intensidad sonora, pH, Temperatura y Presión. Cada uno de estos módulos está compuesto por los sensores respectivos a cada variable y a su vez se comunican a la computadora por medio de una interfaz tipo serie RS-232.

El punto medular que atañe a este trabajo es la interfaz, de tal forma que, durante este desarrollo, prestaremos mayor atención en ella.

Para fines de esta tesis, entenderemos como interfaz el equipo que recibe señales analógicas de diferentes módulos simultáneamente, las convierte a formato digital y las envía a una computadora personal para su posterior análisis. Además, la interfaz estará encargada de proporcionar los voltajes de alimentación necesarios para que los módulos operen.

Al observar el diseño de la interfaz antigua se notó que el dispositivo que controlaba toda la interfaz era un microcontrolador Microchip de la familia de los PIC´s. En el capítulo 2, se hará una descripción de este tipo de dispositivos.

La interfaz anterior es conectada con los módulos por medio de conectores DB-25. Existen cuatro canales para conectar los módulos intercambiables. Los módulos proporcionan una señal analógica de entre 0 y 2 volts de corriente directa, la cual se acopla al microcontrolador por medio de un circuito buffer. La conversión de datos se hace mediante un convertidor analógico digital de 8 bits con el que cuenta el microcontrolador. Los datos son enviados a la PC a través

de un puerto serial en formato RS-232.

1.2. Definición del problema y análisis de necesidades

La primera actividad dentro de la tesis fue el clasificar las partes del sistema en tres grupos: las que tuvieran que ser renovadas totalmente, las que podían ser rescatadas íntegras para la nueva interfaz y las que, mediante alguna modificación, pudieran también seguir siendo usadas.

Al tener una visión de la interfaz que se estuvo manejando anteriormente, se prosiguió a concretar los puntos débiles que tenia.

El primer punto que se consideró en el sistema fue la resolución. El sistema antiguo está diseñado para manejar una resolución de 8 bits, en su época fue realizado de esta manera por el alto costo que implicaba la compra de convertidores de mayor resolución. Esto ha traído como consecuencia que en el software, los datos que son desplegados en pantalla no sean suficientemente precisos limitando así el funcionamiento del sistema entero y por consecuencia limitando su campo de aplicación.

En el caso particular de la nueva interfaz, se propuso cambiar la resolución de la conversión aumentándola en 4 bits más uno de signo, con lo que la resolución total será de 13 bits.

Por otro lado, para poder aprovechar las partes útiles del sistema, los módulos de entrada no van a tener modificaciones, por esto, se diseñarán los circuitos necesarios que cumplan con la función de acondicionar la señal proveniente de los módulos para que la interfaz pueda adaptarse a ellos y así funcionar de manera correcta.

Conjuntamente, en la tarjeta se implementarán los circuitos necesarios para proporcionar los voltajes a los módulos.

Otro punto en el que se tomará particular atención es el mantenimiento. Debido a que con el paso del tiempo, los dispositivos electrónicos van adquiriendo características superiores, este sistema necesita un mantenimiento periódico por lo cual, es conveniente crear los recursos necesarios para evitar conflictos en esos momentos. Actualmente, la programación del sistema está hecha en lenguaje ensamblador, lo que dificulta la tarea señalada. Por tal motivo, se desarrollará una nueva versión de firmware en el microcontrolador de la interfaz cuidando que esta cumpla con lo dispuesto y que pueda acoplarse de manera eficiente con el software. Las rutinas de programación de este nuevo firmware serán efectuadas en un lenguaje de programación de alto nivel que permita, mediante modificaciones sencillas, expandir las características de la interfaz. Con lo anterior, se podrá solucionar el problema anteriormente expuesto, obteniendo así versatilidad en el equipo.

Al realizar el sistema en conjunto, se concentrarán los esfuerzos con el fin de incrementar la velocidad de transmisión de datos. El sistema anterior funciona con una velocidad de 100 datos/s. Se planea incrementar dicha velocidad en

por lo menos diez veces la velocidad de la versión anterior. Con esto, se logrará que el ambiente gráfico sea más dinámico y que los procesos que se requieran efectuar para los datos, se realicen de manera más eficiente.

1.3. Alcance del proyecto

Considerando esto, el presente proyecto cumplirá con el objetivo plasmado al hacer el diseño e implementación de una interfaz que servirá de puente entre los módulos que contendrán la información de las variables sensadas y la computadora que las recibirá. El diagrama general del sistema se muestra en la figura 1.1.

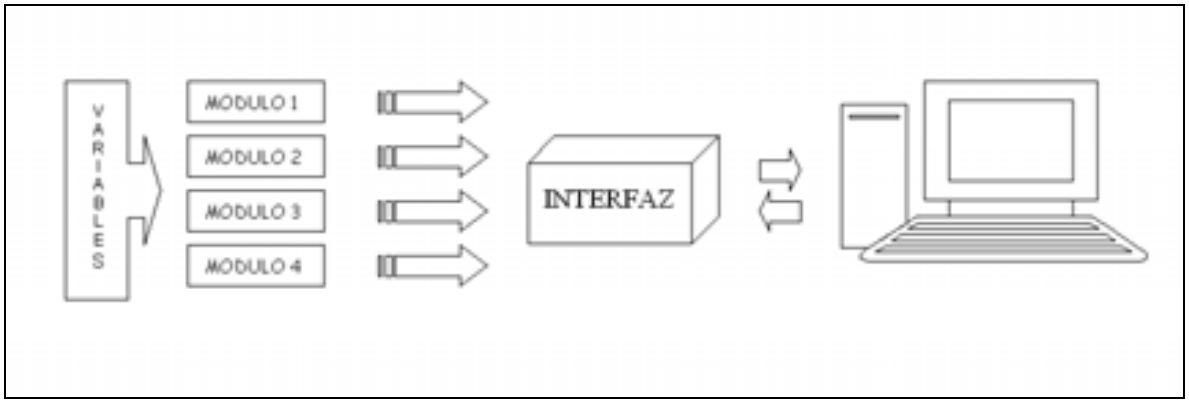


Figura 1.1: Diagrama general del sistema de instrumentación.

Para que la interfaz funcione de manera óptima, tendrá que cumplir con características concretas, de tal forma que sea adaptable con los módulos ya creados anteriormente y a su vez, consiga comunicarse de manera eficiente con el software que será creado. Por otro lado, también se empeñarán los esfuerzos en obtener un sistema flexible de tal suerte que, si éste necesita expandirse, se logre de manera sencilla y rápida sin que se tenga que hacer una reestructuración total de los dispositivos que lo conforman. Las funciones principales de la interfaz se pueden resumir en la figura 1.2:

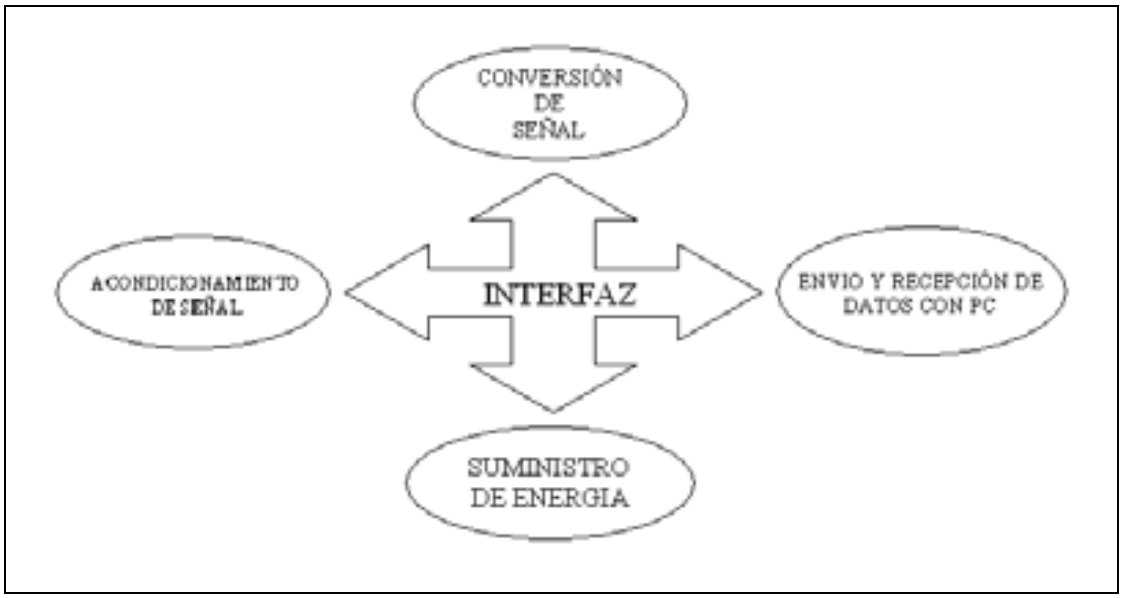


Figura 1.2: Funciones de la interfaz.

Pasos a seguir en el desarrollo de esta tesis:

- Selección de herramientas (microcontrolador, tipo de comunicación a implementar, programador, equipo, etc.)
- Selección de lenguaje de programación (compilador, entorno de programación del microcontrolador, etc.)
- Desarrollo de rutinas de programación para control de la interfaz (Adquisición, procesamiento y envío de datos).
- Construcción de la interfaz
- Pruebas de funcionamiento con datos reales
- Evaluación de resultados

1.4. Campos de aplicación del sistema de instrumentación

El sistema de adquisición de datos está planteado para ser utilizado en centros de enseñanza así como en experimentos sencillos de laboratorio. Debido a la diversidad de módulos que presenta, puede ser usado en experimentos para medir variables físicas o químicas.

2. DETERMINACIÓN DE LOS REQUERIMENTOS DEL SISTEMA

En el siguiente capítulo se definirán conceptos teóricos básicos que van a permitir comprender de manera más sencilla como fue diseñada la interfaz. Cabe mencionar que el orden en el que se llevará este capítulo, estará relacionado con el orden que tiene cada una de las etapas incluidas en la interfaz, iniciando con la entrada de señales a través de los transductores y finalizando con los datos que son enviados a la computadora para ser manipulados por el software.

2.1. Acondicionamiento de una señal y multiplexado

Como quedó definido en el capítulo anterior, una de las actividades de las que se realizarán en el proyecto será la de convertir señales analógicas en digitales utilizando convertidores A/D. En muchas aplicaciones, incluidas la presente, dichos convertidores necesitan dispositivos adicionales para poder recibir de forma correcta la señal analógica, regularmente, se incluye circuitería adicional para acondicionar dicha señal evitando así problemas en el momento de adquirirla. La multiplexión y la amplificación de una señal son ejemplos de lo descrito. Cada aplicación requerirá soluciones específicas que podrán ser implementadas en gran variedad de configuraciones físicas. A continuación se muestran algunas definiciones que servirán para conocer aspectos básicos a considerar sobre los convertidores A/D y los procesos involucrados en el acondicionamiento de una señal. Al tener conocimiento de esto, se estará preparado para afrontar la etapa de entrada de la interfaz dedicada a resolver este propósito.

2.1.1. Multiplexado y convertidores A/D

En muchas aplicaciones industriales y de laboratorio se deben convertir diversas señales analógicas a formato digital. Si la velocidad no es un factor limitante, se puede diseñar un mecanismo de conmutación llamado multiplexor cuyo fin será el hacer que se comparta el convertidor A/D entre múltiples canales de entrada. Esto es realizado debido al costo relativamente alto de los convertidores. Los multiplexores también permiten que se comparta, entre múltiples canales, la amplificación y el resto de circuitería de acondicionamiento de la señal. La conmutación se controla mediante software o hardware auxiliar. Es importante saber que una de las desventajas a considerar al usar un multiplexor es que reduce la frecuencia con la que se adquieren los datos y que, dicha velocidad, también se verá afectada al tener en cuenta el criterio de Nyquist. No obstante, dependiendo de la aplicación en específico y las características de un sistema dado, se obtendrá un equilibrio entre velocidad y costo con el que habrá la posibilidad de realizar un sistema que satisfaga las necesidades del dispositivo.

2.1.2. Amplificación de la señal y conversión A/D

Muchos convertidores A/D trabajan mejor con señales en el rango de 1 a 10 V, debido a esto, las señales de bajo voltaje se deben amplificar antes de realizar

una conversión A/D ya sea individualmente o multiplexando un circuito compartido. Del mismo modo, se deben atenuar las señales de alto voltaje.

Los amplificadores también pueden aumentar la resolución de los convertidores A/D con señales de bajo nivel. Por ejemplo, un convertidor A/D de 12 bits con una ganancia de 4 puede digitalizar una señal con la misma resolución que un convertidor de 14 bits con una ganancia de 1. Esto, debido a que la cantidad de valores a ser representados va a aumentar gracias a la ganancia o de manera numérica implicaría que el número de valores a representar es: $4x2^{12}=2^{14}$. Sin embargo, es importante tener en cuenta que los amplificadores de ganancia fija, los cuales multiplican todas las señales de manera proporcional, aumentan la sensibilidad a señales de bajo voltaje pero no aumentan el rango dinámico del convertidor, en otras palabras, no aumentan la relación existente entre el mayor y menor número que pueden ser representados por el convertidor.

Por otro lado, existen los amplificadores de ganancia programable (PGA) los cuales se pueden configurar para aumentar automáticamente la ganancia a medida que disminuye el nivel de la señal aumentando de manera efectiva el rango dinámico del sistema. Un PGA con tres niveles de ganancia establece tres órdenes de magnitud y puede hacer que un convertidor de 12 bits se comporte casi como un convertidor de 18 bits. Sin embargo, esta funcionalidad disminuye la frecuencia de muestreo.

Desde el punto de vista del sistema, el rendimiento del amplificador debe estar a la par con el del convertidor A/D en sí; la precisión de la ganancia se debe especificar como un pequeño porcentaje de la ganancia total. El ruido del amplificador y el error del desplazamiento también deben ser pequeños.

2.1.3. Entradas de un solo terminal o entradas diferenciales

Otra cosa a tener en cuenta, a la hora de especificar el hardware de adquisición de datos analógicos, es el tipo de entradas que se utilizarán; ya sea de un solo terminal o de entrada diferenciales. Por un lado, las entradas de un solo terminal son menos costosas pero pueden ser problemáticas si existen diferencias de potencial en las tierras.

En una configuración de un solo terminal, la tensión de alimentación de la señal y la entrada del amplificador está tomada respecto de la tierra. Esto es adecuado para señales de alto nivel donde la diferencia de potencial entre las tierras es relativamente baja. Sin embargo, una diferencia de potencial en las tierras crea un flujo de corriente a través de la tierra que genera errores.

Por el contrario, las entradas diferenciales conectan las entradas positiva y negativa del amplificador a ambos terminales del origen de la señal. Cualquier voltaje inducido de bucle de tierra aparece en ambas terminales y es rechazado como un error de modo común. El inconveniente de estas conexiones diferenciales es que son dos veces más costosas que las entradas de un solo

terminal. Por ejemplo, una tarjeta de entrada analógica de ocho canales sólo manejará cuatro entradas diferenciales mientras que si se configurara como de un solo terminal, sería de ocho entradas.

2.2. El amplificador no inversor

A continuación se menciona una breve descripción de la configuración llamada “no inversor” que es una configuración usual en las aplicaciones con amplificadores operacionales. Esta configuración fue útil al desarrollar la interfaz para poder acondicionar las señales entregadas a ella por los transductores.

En el amplificador no inversor, la señal de entrada se aplica en la terminal no inversora del amplificador, la ganancia de lazo cerrado viene dada por

$$A_V = 1 + \frac{R_2}{R_1}$$

En el amplificador no inversor, la salida está en fase con la entrada.

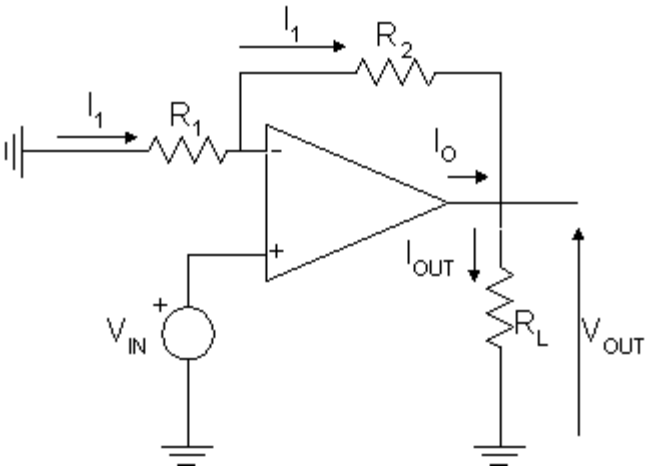


Fig 2.1: Amplificador no inversor

Considerando las ecuaciones del amplificador operacional con realimentación negativa tenemos que:
$I_+ = I_- = 0$
$e_+ = e_-$

Y analizando el circuito, por leyes de Kirchoff tenemos lo siguiente:

$e_+ = V_{IN}$
$e_- = -R_1 I_1 \Rightarrow I_1 = -\frac{e_-}{R_1} = -\frac{V_{IN}}{R_1}$

$V_{OUT} = - I_1 (R_1 + R_2) = (1+\frac{R_2}{R_1})V_{IN}$ ==> Ganancia $A_V = 1+\frac{R_2}{R_1}$

$$I_{OUT} = \frac{V_{OUT}}{R_L} = \frac{1}{R_L}(1+\frac{R_2}{R_1})V_{IN}$$

$$I_0 = I_{OUT} - I_1 = \frac{1}{R_L}(1+\frac{R_2}{R_1})V_{IN} + \frac{V_{IN}}{R_1}$$

De este circuito se obtienen como características más relevantes:

- Ganancia del circuito independiente de la del operacional.
- Ganancia fijada por el cociente de dos resistencias.
- Resistencia de entrada infinita.
- La entrada y la salida son del mismo signo (no inversor)

2.2.1. Conversión analógica-digital y etapas de la conversión

La mayoría de los sensores para medir temperatura, presión y otras variables continuas proporcionan una salida que varía continuamente para representar la magnitud de la variable en cuestión. Para hacer que esta señal pueda ser interpretada por un microprocesador, debe ser convertida de un valor contínuo, que varía ligeramente respecto al tiempo, a un número digital y discreto.

Los sistemas de adquisición de datos, dividen el tiempo en intervalos discretos. En general, no hay información sobre el comportamiento del proceso en el intervalo de tiempo que hay entre las muestras recogidas. Por lo que, se debe tener un cuidado especial para asegurarse de que no se pierda ningún dato significativo, y que la interpolación entre los puntos recogidos sea una suposición válida.

El proceso de conversión analógico-digital (A/D) tiene tres etapas principales: una de muestreo en el tiempo, otra de cuantización y finalmente una de codificación.

La primera etapa, el muestreo, consiste en tomar muestras periódicas de la amplitud de la señal analógica que se va a digitalizar. El intervalo entre muestras que se van a tomar debe ser constante y el ritmo de este muestreo, llamado frecuencia o tasa de muestreo, determina el número de muestras que se toma en un intervalo de tiempo.

Durante un proceso típico de adquisición de datos en donde interviene más de un canal, cada uno de estos, se leen secuencialmente. A esto se le llama muestreo estándar o distribuido. A una lectura de todos los canales se le llama un escaneo. Debido a que cada canal se adquiere y convierte en un intervalo de tiempo diferente, se crea una asimetría en el tiempo de muestreo entre las distintas muestras.

Si es importante la sincronización de las entradas, algunas tarjetas de adquisición ofrecen una operación en modo “burst” o circuitería simultánea de muestreo y retención. En el modo “burst”, se adquiere cada canal a la máxima velocidad de la tarjeta, luego se espera un intervalo de tiempo definido por el usuario antes de muestrear otra vez.

Los verdaderos sistemas de muestreo y retención simultánea pueden muestrear todos los canales dejando unos pocos nanosegundos entre muestreos, eliminando las discontinuidades de fase y tiempo en casi todos los procesos (excepto en los más rápidos).

En el proceso de conversión digital de una señal no sólo es necesario muestrear la señal analógica sino también cuantizarla. Básicamente, la cuantificación lo que hace es convertir una sucesión de muestras de amplitud continua en una sucesión de valores discretos preestablecidos según el código utilizado.

Durante el proceso de cuatización se mide el nivel de tensión de cada una de las muestras, obtenidas en el proceso de muestreo, y se les atribuye a 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.

Se le llama cuantización uniforme a aquella en la que todos los niveles son iguales. La mayoría usan un número de niveles que es una potencia de 2. Si $L=2^B$, cada uno de los niveles es codificado a un número binario de B bits.

Los errores asignados con la cuantización tienen que ver con la imprecisión introducida por la conversión de un voltaje analógico a un número digital.

Cuando se ha terminado la cuantización, la señal analógica se convierte en una señal digital, ya que los valores que están preestablecidos son finitos. No obstante, todavía no se traduce al sistema binario. La señal ha quedado representada por valores finitos.

La codificación consiste en la traducción de los valores de tensión eléctrica analógicos al sistema binario, mediante códigos preestablecidos. La señal analógica va a quedar transformada en un tren de impulsos digital (sucesión de ceros y unos).

Un dispositivo encargado de realizar el proceso de digitalización es el convertidor analógico-digital (CAD). Se puede colocar un convertidor analógico-digital en gran variedad de configuraciones. Además del convertidor, el circuito podrá estar

acompañado de circuitos de muestreo y retención, un amplificador, un multiplexor, circuitos de sincronización y elementos acondicionadores de la señal dependiendo la aplicación en la que se vaya a utilizar.

A la hora de elegir que tipo de convertidor A/D se va a utilizar en una aplicación, el rendimiento debe ser suficiente para cumplir los requerimientos de los transductores analógicos de entrada a utilizar. También se debe tener en cuenta la precisión, la máxima amplitud y el rango de la señal.

Una característica importante en el rendimiento de un convertidor A/D es su resolución, expresada en bits. Un convertidor A/D divide el intervalo de entrada en 2^n partes, donde n es el número de bits. Dicho de otra forma, la *resolución* indica el número de niveles utilizados para representar el intervalo de entrada analógico y determina la sensibilidad del convertidor a un cambio en la entrada analógica. Se puede amplificar la señal de entrada para aumentar la sensibilidad aparente, en el caso de que el intervalo máximo esperado de la señal sea menor que el intervalo de entrada del convertidor A/D. Como los convertidores A/D de mayor resolución son más caros, es especialmente importante no comprar más resolución de la necesaria. *Por ejemplo, si se tiene unos transductores de temperatura con el 1 % de precisión (1 de 100), un convertidor A/D de 16 bits (1 de 65.536) ofrece más resolución de la necesaria.*

Por lo tanto, como el número de bits determina el número de partes en las que se divide el intervalo de entrada, para aproximar un voltaje de entrada analógica podemos poner como ejemplo que si tenemos una resolución de 13 bits, para una señal de entrada de -4.096 V a 4.096 V, significa que el rango se divide en 2^{13} =8192 partes de 0.001 V cada una.

La precisión absoluta de la conversión A/D es una función de la estabilidad del voltaje de referencia (el voltaje conocido con respecto al cual se compara el voltaje desconocido) y del rendimiento del comparador. En conjunto, de poco sirve conocer la precisión del convertidor A/D en sí. Es más significativa la precisión del sistema, junto con el multiplexor, el amplificador y el resto de cirtuitería asociados.

Otro parámetro de rendimiento del convertidor A/D a considerar es la velocidad. En conjunto, la velocidad del sistema depende de los tiempos de conversión, de adquisición, de transferencia y del número de canales a los que da servicio el sistema:

• *Tiempo de adquisición*: Es el tiempo que necesita el circuito analógico para adquirir la señal. También llamado tiempo de apertura, es el tiempo durante el cual el convertidor debe observar el voltaje analógico para poder completar una conversión.

• *Tiempo de conversión*: Es el tiempo necesario para generar el valor digital que se corresponda con el valor analógico.

• *Tiempo de transferencia*: Es el tiempo necesario para enviar el valor digital a la memoria del microcontrolador.

Entonces, la productividad será igual al número de canales a los que se da servicio dividido por el tiempo necesario para realizar estas tres funciones.

Otro punto a considerar en estas aplicaciones es el teorema de Nyquist, el cual define la relación necesaria entre la frecuencia más alta contenida en una señal y la frecuencia de muestreo mínima requerida. Nyquist afirmó que la frecuencia de muestreo debe ser mayor al doble de la componente de mayor frecuencia contenida dentro de la señal de entrada. *Así que, para muestrear una onda senoidal de 1 Hz, la frecuencia de muestreo debe ser mayor de 2 Hz. (Pero una frecuencia de 8 a 16 Hz sería mejor para resolver la verdadera forma de la onda.)*

Entre las consecuencias principales de ignorar el criterio de Nyquist están la pérdida de la información de alta frecuencia y la introducción de aliasing. Si la frecuencia de muestreo no es lo suficientemente elevada, pueden aparecer frecuencias inexistentes.

2.2.2. Principales tipos de convertidores analógico-digitales

Los convertidores analógico-digital pueden ser clasificados según diversos criterios como son la resolución o el método con el que el convertidor consigue realizar la conversión.

Actualmente, en cuanto a la resolución de un CAD, las tareas industriales y de laboratorio típicamente requieren de 12 a 16 bits, aunque 12 es el valor más común. En general, si se aumenta la resolución aumentan los costes y disminuye la velocidad de conversión.

Por otro lado, considerando la clasificación según el método para realizar la conversión, hay cuatro tipos principales de convertidores A/D que se utilizan en aplicaciones industriales y de laboratorio: de aproximaciones sucesivas, flash/paralelo, de integración y rampa/contador. Algunos están optimizados para ser rápidos, otros para ser económicos y otros para obtener un compromiso entre prioridades de competencia. A continuación se mencionan los principales

- A*proximaciones sucesivas*:

Es el diseño más común utilizado en aplicaciones industriales y de laboratorio en general. Este diseño ofrece un compromiso efectivo entre resolución, velocidad y costo. En este tipo de diseño se utilizan un convertidor digital-analógico (D/A) interno y un sólo comparador (un circuito que decide cuál de los dos voltajes es mayor) para aproximar el voltaje desconocido. Esto lo hace cambiando los bits del convertidor D/A hasta que los voltajes coincidan dentro del bit menos significativo.

La velocidad de muestreo bruta para los convertidores de aproximación sucesiva está en el intervalo de 50 kHz a 1 MHz. Para conseguir velocidades de muestreo mayores se utiliza una técnica redundante, que genera una conversión aproximada inicial, seguida de un paso para corregir el bit menos significativo después de permitir un tiempo de establecimiento suficiente. Por lo tanto, la conversión se completa más rápido a expensas de añadir hardware adicional. La redundancia es útil cuando se deseen conseguir grandes velocidades y a la vez grandes resoluciones.

Los convertidores de aproximaciones sucesivas contienen un valor fijo en su tiempo de conversión que no depende del valor de la entrada analógica, la disposición básica es semejante a la del CAD de rampa digital, sin embargo, el convertidor de aproximaciones sucesivas no utiliza ningún contador para dar la entrada en el bloque del CAD, pero en cambio usa un registro con lógica de control que modifica el contenido del registro bit a bit hasta que los datos del registro son el equivalente digital de la entrada analógica.

El tiempo de conversión de los convertidores de aproximaciones sucesivas de "n" bits requieren "n" ciclos de reloj para realizar su conversión sin importar la magnitud del voltaje que esta presente en su entrada, esto se debe a que los circuitos de control tienen que ensayar un 1 lógico en cada posición del bit para ver si se necesita o no, es por esto que los convertidores de aproximaciones sucesivas tienen tiempos de conversión muy rápidos, su uso en aplicaciones de sistemas con adquisición de datos permitirán que se adquieran más valores de datos en un intervalo de tiempo dado. Esto puede ser muy importante cuando los datos analógicos cambian su valor rápidamente.

- *Flash/paralelo*:

Cuando se requiere una operación de gran velocidad, se utiliza una conversión A/D en paralelo o de tipo flash. Este diseño utiliza múltiples comparadores dispuestos en paralelo para procesar muestras a más de 100 MHz con una resolución que va desde 8 hasta 12 bits. La comparación se realiza con una cadena de comparadores, con sus apropiadas referencias, operando en paralelo.

El inconveniente de este diseño es el gran número de comparadores relativamente caros que hacen falta. Por ejemplo, un convertidor de 12 bits necesita 4095 comparadores.

Integración: Este tipo de convertidor A/D integra un voltaje de entrada desconocido durante un periodo de tiempo específico, integrándolo luego hasta cero. Este tiempo se compara con la cantidad de tiempo que ha sido necesaria para realizar una integración similar con un voltaje de referencia conocido. Los tiempos relativos necesarios y el voltaje de referencia conocido permiten obtener el voltaje de entrada desconocido. Hay disponibles en el mercado convertidores con una

resolución que va desde 12 hasta 18 bits, y con unas frecuencias de muestreo netas de 10 a 500 kHz.

Debido a que este tipo de diseño calcula de forma efectiva el promedio del voltaje de entrada con respecto al tiempo, también suaviza el ruido de la señal. Además, si se elige un periodo de integración que sea múltiplo de la frecuencia de la línea alterna, se consigue un excelente rechazo al ruido de modo común. Los convertidores por integración, más precisos y más lineales que los de aproximaciones sucesivas, son una buena elección para las señales de bajo voltaje.

- *Rampa/contador*:

De diseño similar al de aproximaciones sucesivas, los convertidores A/D de tipo rampa o contador utilizan un circuito comparador y un convertidor D/A. Este diseño progresivamente incrementa un contador digital, y con cada nueva cuenta genera el correspondiente voltaje analógico y lo compara con el voltaje de entrada desconocido. Cuando coinciden, el contador contiene el valor digital equivalente al de la señal desconocida.

Una variación del método del contador es el método de la rampa, que sustituye un amplificador operacional u otro circuito de rampa analógico por el convertidor D/A. Esta técnica es un poco más rápida.

- Delta-Sigma:

Se trata de un convertidor de un solo bit que muestrea a gran velocidad. La secuencia binaria es filtrada digitalmente para obtener el valor medio con mayor precisión. A mayor precisión, mayor tiempo de conversión.

La operación básica de un convertidor Sigma-Delta es intercambiar tasa de muestreo por resolución. Las señales son muestreadas a una tasa mucho mayor que la de Nyquist, pero con un solo bit de resolución en amplitud.

La señal análoga de entrada y el flujo de bits (bitstream), cuya densidad de un bit es una representación de la magnitud de la señal análoga, son sumadas. Luego son integradas y entran a un comparador el cual tiene como salida 0 o 1 dependiendo sí la salida del integrador es mayor o menor que el voltaje de referencia del comparador. Por ejemplo, suponer una entrada análoga positiva y el promedio del "bitstream" alto comparado con la entrada análoga, entonces la salida del sumador será negativa. Esto en el tiempo dará como resultado una salida del integrador la cual hará que el comparador de un cero como resultado. Es claro que el promedio del "bitstream" debe seguir de cerca de la señal análoga de entrada. El comparador es simplemente un amplificador con muy alta ganancia

el cual tendrá salidas de 1 o 0 dependiendo de la diferencia entre la entrada y su voltaje de referencia.

El precio que tiene que pagarse por la alta resolución de la tecnología Sigma – Delta siempre ha sido velocidad; el hardware tiene que operar a una tasa de sobremuestreo, mucho más alta que el ancho de banda de la señal, demandando circuitos digitales complejos.

Se considera que tiene características satisfactorias de ruido y alta resolución además de excelente linealidad y un error muy pequeño. Son utilizados para el audio digital y en la implementación de receptores de radiofrecuencia integrados como dispositivos móviles de comunicación (telefonía móvil, GPS).

2.2.3. Familia de Microcontroladores PIC

Se conoce como PIC a una familia de microcontroladores fabricados por Microchip Technology Inc basados en memoria EPROM/FLASH. El nombre de PIC viene de su nombre completo en inglés: Peripheral Interface Controler (Controlador de Interfaz Periférico).

La historia del PIC comienza cuando en 1965 la Microelectronics Division de GI es creada. Entre sus objetivos se encontraban el generar las primeras arquitecturas viables de memoria EPROM y EEPROM y comenzar a desarrollar una variedad de funciones digitales y analógicas de sus familias AY3-xxxx y AY5-xxxx. A principio de los años 70, también se dio a la tarea de crear un microprocesador de 16 bits, denominado CP1600. Este fue un microprocesador razonable, pero no particularmente bueno manejando puertos de E/S. Para algunas aplicaciones muy específicas GI diseñó un Controlador de Interface Periférico (PIC) en 1975. Fue diseñado para ser muy rápido, además de ser un controlador de E/S para una máquina de 16 bits pero sin necesitar una gran cantidad de funcionalidades, por lo que su lista de instrucciones fue pequeño.

No es de extrañar que la estructura diseñada en 1975 es, sustancialmente, la arquitectura del actual PIC16C5x. Además, la versión de 1975 fue fabricada con tecnología NMOS y sólo estaba disponible en versiones de ROM de máscara, pero seguía siendo un buen microcontrolador pequeño. El mercado, no obstante, no pensó así y el PIC quedó reducido a ser empleado por grandes fabricantes únicamente.

Durante los 80, GI renovó su apariencia y se reestructuró. GI Microelectronics Division cambió a GI Microelectronics Inc., la cual fue finalmente vendida en 1985 a Venture Capital Investors. La gente de Ventura realizó una profunda revisión de los productos en la compañía, desechando la mayoría de los componentes AY3, AY5 y otra serie de cosas, dejando sólo el negocio de los PIC y de las memorias EEPROM y EPROM. Se tomó la decisión de comenzar una nueva compañía,

denominada Arizona Microchip Technology, tomando como elemento diferenciador sus controladores integrados.

Como parte de esta estrategia, la familia NMOS PIC165x fue rediseñada para emplear algo que la misma compañía fabricaba bastante bien, memoria EPROM. De esta forma nació el concepto de basarse en tecnología CMOS, OTP y memoria de programación EPROM, naciendo la familia PIC16C5x.

Actualmente Microchip ha realizado un gran número de mejoras a la arquitectura original del PIC, adaptándola a las actuales tecnologías y al bajo costo de los semiconductores.

En el mercado actual, existen diversas familias de PIC, las cuales se amplían constantemente, sin embargo, las más básicas son:

- PIC16C5x: instrucciones de 12 bit, 33 instrucciones, 2 niveles de acumulador, sin interrupciones. En algunos casos la memoria es del tipo ROM, definida en fábrica.
- PIC16Cxx: instrucciones de 14 bit, 35 instrucciones, 8 niveles de acumulador. .
- PIC17Cxx: instrucciones de 16 bit, 55 instrucciones, 16 niveles de acumulador. A menos que se indique, la memoria es del tipo EPROM.
- Adicionalmente existen otras familias derivadas, como los PIC16Fxx, PIC18FXX, PIC24FXX que emplean memoria del tipo FLASH.

La aceptación que tienen los PIC en la actualidad se debe a diversas causas. Son dispositivos económicos y sencillos de manejar debido a que su juego de instrucciones es reducido (35 instrucciones en la gama media). Poseen un buen desempeño en la velocidad de su funcionamiento considerando su consumo, tamaño y precio.

Otras característica importante para preferirlos es que tanto la información para su uso como las herramientas de desarrollo son económicas y fáciles de conseguir. Existe una gran variedad de herramientas de hardware y software que permiten programar, simular en software, emular en tiempo real, grabar, depurar, borrar y comprobar el comportamiento de los PIC.

Por otro lado, la gran variedad de modelos de PIC permite elegir el que mejor responde a los requerimientos de la aplicación con la ventaja de que cuando se aprende a manejar uno de ellos, conociendo su arquitectura y su repertorio de instrucciones, es muy fácil emplear otro modelo pudiendo así emigrar de un modelo a otro con mínima dificultad.

Características relevantes

Arquitectura

La arquitectura del procesador sigue el modelo Harvard modificado.A diferencia de la arquitectura Harvard tradicional, en esta arquitectura, la CPU se conecta de forma dependiente y con los mismos buses con la memoria de instrucciones y con la de datos pero el formato de la instrucción es más largo logrando realizar instrucciones en menos ciclos de reloj

Segmentación

Se aplica la técnica de segmentación "pipe-line" en la ejecución de las instrucciones.
La segmentación permite al procesador realizar al mismo tiempo la ejecución de una instrucción y la búsqueda del código de la siguiente. De esta forma se puede ejecutar cada instrucción en un ciclo (un ciclo de instrucción equivale a cuatro ciclos de reloj).
Las instrucciones de salto ocupan dos ciclos al no conocer la dirección de la siguiente instrucción hasta que no se haya completado la de bifurcación.

Formato de las instrucciones

El formato de todas las instrucciones es de la misma longitud.
Todas las instrucciones de los microcontroladores de la gama baja tienen una longitud de 12 bits. Las de la gama media tienen 14 bits y más las de la gama alta. Esta característica es muy ventajosa en la optimización de la memoria de instrucciones y facilita enormemente la construcción de ensambladores y compiladores.

Juego de instrucciones

Procesador RISC (Computador de Juego de Instrucciones Reducido).
Los modelos de la gama baja disponen de un repertorio de 33 instrucciones, 35 los de la gama media y casi 60 los de la alta.

Todas las instrucciones son ortogonales

Cualquier instrucción puede manejar cualquier elemento de la arquitectura como fuente o como destino.

Arquitectura basada en un "banco de registros"

Esto significa que todos los objetos del sistema (puertas de E/S, temporizadores, posiciones de memoria, etc.) están implementados físicamente como registros.
Diversidad de modelos de microcontroladores con prestaciones y recursos diferentes

La arquitectura Harvard modificado y la técnica de segmentación son los principales recursos en los que se apoya el elevado rendimiento que caracteriza estos dispositivos programables, mejorando dos características esenciales: velocidad de ejecución y eficiencia en la compactación del código.

2.2.4.Comunicación USB

En el desarrollo de la interfaz, se planteó que la comunicación entre la computadora y el microcontrolador se hiciera a través de un módulo con comunicación USB. A continuación se mostrará un panorama general de este tipo de comunicación, que aunque no va a ser un tópico a profundizar en el desarrollo de la tesis, se considera conveniente mencionar para que exista un entendimiento de las ventajas que se obtuvieron al utilizar dicho módulo.

Podemos comenzar mencionando que el USB es un bus punto a punto: dado que el lugar de partida es el *host (PC o hub*), el destino es un periférico u otro *hub.* No hay más que un único *host* (PC) en una arquitectura USB.

Los PC estándar tienen dos tomas USB, lo que implica que, para permitir más de dos periféricos simultáneamente, es necesario un h*ub.* Algunos periféricos incluyen un hub integrado, por ejemplo, el teclado USB, al que se le puede conectar un mouse USB.

Los periféricos comparten la banda de paso del USB. El protocolo se basa en el llamado *paso de testigo (token).* La computadora proporciona el testigo al periférico seleccionado y seguidamente, éste le devuelve el testigo en su respuesta. Este bus permite la conexión y la desconexión en cualquier momento sin necesidad de apagar el equipo.

A continuación se describen las principales características de este protocolo:

Características eléctricas

A nivel eléctrico, el cable USB transfiere la señal y la alimentación sobre 4 hilos.
A nivel de alimentación, el cable proporciona la tensión nominal de 5 V. Es necesario definir correctamente el diámetro del hilo con el fin de que no se produzca una caída de tensión demasiado importante en el cable. Una resistencia de terminación instalada en la línea de datos permite detectar el puerto y conocer su configuración.

Existen tres velocidades de transmisión de datos: la máxima, denominada “high speed” que equivale a 480 Mb/s., la intermedia, designada como “full speed”, que corresponde a 12 Mb/s. y la mas baja, llamada “low speed”, cuya velocidad de transferencia de datos es de 1.5 Mb/s.

A nivel de señal, se trata de un par trenzado con una impedancia característica de 90 Ω La velocidad puede ser tanto de 12 Mbits/s como de 1,5 Mbits/s. La sensibilidad del receptor puede ser de, al menos, 200mV y debe poder admitir un buen factor de rechazo de tensión en modo común. El reloj se transmite con el reloj se transmite junto con los datos, la codificación es de tipo NRZI, existiendo un dispositivo que genera un bit de relleno (bit stuffing) que garantiza que la

frecuencia de reloj permanezca constante. Cada paquete va precedido por un campo de sincronismo.

Consumo

Cada sección puede proporcionar una determinada potencia máxima siendo el PC el encargado de suministrar la energía. Además, el periférico puede estar autoalimentado *(self powered).* La computadora gestiona el consumo, teniendo capacidad de poner en reposo (*suspend)* o en marcha a un periférico USB. En reposo, este reduce su consumo (si puede),quedándose la parte USB funcional.
Esta gestión está orientada especialmente a los equipos portátiles.

Principales características del bus:

• Banda de paso, disponibilidad desde algunos kilobits a varios megabits;
• Transferencia isócrona y asíncrona en el mismo bus;
• Varios tipos de periféricos en el mismo bus;
• Posibilidad de conectar hasta 127 perifericos;
• Tiempo de respuesta garantizado (para audio y vídeo)
• Flexibilidad a nivel de banda de paso;
• Fiabilidad, control de errores;
• Perfectamente integrado en el PC, *plug and play* (conectar y usar)
• Coste reducido en la versión de baja velocidad (1,5 Mbits/s);
• Posible expansión del bus.

3. DISEÑO Y CONSTRUCCIÓN DE LA INTERFAZ

El presente capitulo está enfocado en explicar todos las fases que fueron consideradas para poder llegar a construir físicamente la interfaz. Se comienza presentando el como fue plasmada en su aspecto general y posteriormente, se adentra en los aspectos particulares de cada una de las etapas de diseño.

3.1. Estructura general de la interfaz

Considerando un panorama general de la interfaz física, podemos dividirlo en tres partes, como lo muestra la figura 3.1.

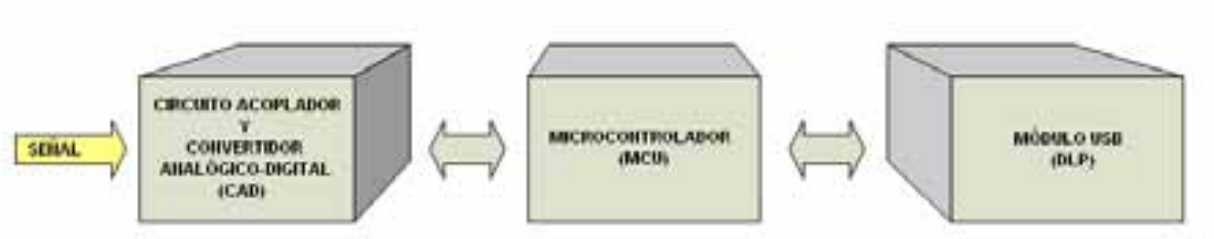


Figura 3.1: Dispositivos principales de la interfaz

La primera corresponde a la relacionada con las señales que entran a la interfaz provenientes de cada uno de los módulos encargados de sensar las variables en estudio. En esta sección se incluye lo relacionado al acondicionamiento de la señal para que esta pueda introducirse a las entradas del convertidor analógico-digital (CAD). También incluye la comunicación que tenga el CAD con el microcontrolador para que el primero le mande la información de la señal después de convertirla a formato digital.

La segunda se enfoca a todo lo referente al microcontrolador, es la parte primordial de la interfaz debido a que aquí se lleva a cabo el seguimiento de los procesos de control como el tiempo entre muestras sensadas, los protocolos de envío y adquisición de muestras, etc. Es conveniente mencionar que, mientras la interfaz este funcionando correctamente, la etapa del microcontrolador va a mantener comunicación constante con los dispositivos de las otras etapas debido a que será una especie de “intermediario” del envío y recepción de la información.

Finalmente, la tercera parte está constituida por lo referente a la comunicación con la computadora por medio del dispositivo USB. Este segmento de la interfaz además, se comunicará con el segmento del microcontrolador debido a que éste último le envía los datos al dispositivo USB con el formato correspondiente.

En la figura 3.2 podemos observar un diagrama detallado, el cual muestra la representación de las partes más importantes de la interfaz.

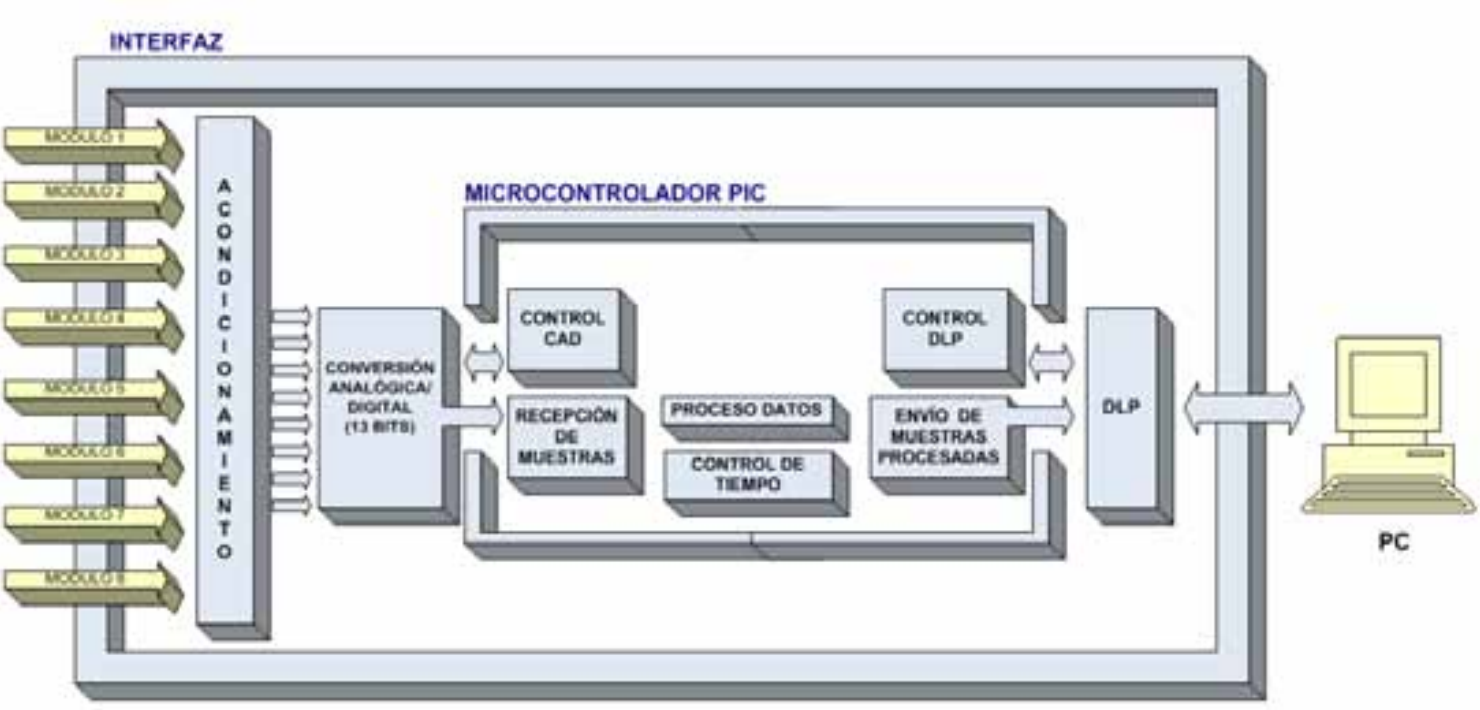


Figura 3.2: Diagrama detallado de la interfaz.

3.2. Los módulos

El sistema de adquisición de datos fue diseñado para poder conectar hasta un total de ocho módulos de manera simultánea. Cada módulo permite medir una variable diferente: Corriente, Capacitancia, Intensidad sonora, Intensidad luminosa, Frecuencia, Inductancia, Resistencia, Temperatura, Voltaje, pH, y Presión. A continuación se comentan las características y especificaciones principales de cada uno de los módulos.

AMPERÍMETRO

El amperímetro permite medir señales de corriente directa o DC (corriente que se mantiene constante durante el tiempo) y de corriente alterna o AC (corriente que sufre cambios en su amplitud conforme transcurre el tiempo). En lo particular el amperímetro está diseñado para medir corrientes alternas con una forma de onda senoidal. En conjunto con la interfaz para computadora y el software de control, el amperímetro permite tomar lecturas automatizadas de corriente, tanto de AC como de DC, con tres escalas de medida: 2mA, 200 mA y 2 A.

CAPACITÓMETRO

El capacitómetro es un instrumento de medición que permite medir capacitancia (medida de la cantidad de carga requerida para crear una diferencia de potencial dado entre dos conductores). En el caso del capacitómetro del sistema, las escalas que maneja son de 0.2uF, 2uF y 20 mF. Con una tolerancia de +/- 3%. Se alimenta cun + 5 V, +12 V y -12 V, los cuales son proporcionados por la interfaz.

DECIBELÍMETRO

Es un instrumento de medición diseñado para medir la intensidad sonora, para verificar cualquier tipo de ambiente acústico. Tiene un intervalo de medición de sonido que va desde 60 dB hasta 100 dB en una sola escala de medición. Para su funcionamiento requiere de un voltaje de alimentación de +5 V proporcionado por la interfaz.

FOTÓMETRO

Es un equipo electrónico encargado de medir la intensidad luminosa para verificar cualquier tipo de ambiente. Tiene un intervalo de luminosidad que va desde 0 luxes hasta 10000 luxes en una sola escala. Puede medir tanto luz natural como artificial. Incluye un sensor óptico.

FRECUENCÍMETRO

Es un instrumento de medición encargado de medir frecuencia de señales periódicas, tanto analógicas como digitales. Tiene un intervalo de medición que va desde 1 Hz hasta 100 kHz, en tres escalas: 1kHz, 10 kHz y 100 kHz. También la amplitud de la señal de entrada puede estar en un intervalo que va desde 1 V hasta 120 V.

INDUCTÓMETRO

Es un instrumento de medición que se utiliza para medir inductancia (la capacidad de un material o dispositivo eléctrico para resistirse al cambio de corriente a través del material). Las escalas que maneja son de 2mHF, 20 mH y 200 mHF con una tolerancia de +·- 3%. Se alimenta gracias a fuentes de + 5 V, +12 V y -12 V, las cuales son proporcionadas por la interfaz.

ÓHMETRO

El óhmetro es un instrumento de medición encargado de medir la resistencia eléctrica (capacidad de diversos materiales de oponerse al flujo de la corriente eléctrica sobre ellos). El ohmetro permite tomar lecturas automatizadas de valores de resistencia con tres escalas de medida: 2kΩ, 200 kΩ y 2 MΩ.

TERMÓMETRO

El termómetro del sistema LABSIS tiene un intervalo de operación de 0 a 100 ºC. Incluye una punta sensora. Provee de una exactitud de +/- ¼ ºC y tiene una impedancia de salida de 0.1 Ω para 1 mA de carga.

VOLTÍMETRO

El voltímetro permite tomar lecturas automatizadas de voltaje con tres escalas de medida 2 V, 20 V, 200 V. Permite medir señales de corriente directa y de corriente alterna senoidal. Cuenta con protecciones de sobrevoltaje de 3.3 a 250 V. El instrumento cuenta con sus propias puntas de prueba.

Al comenzar con el diseño de la interfaz, se sabía que los módulos enviaban a su salida una señal analógica entre 0 y 2 V de corriente directa. En la interfaz

desarrollada, se consideró que la entrada a esta fuera de 0 a 4.096 V. Este parámetro fue considerado de tal manera debido a que, beneficiaría la conversión analógica-digital, por el tipo de convertidor que se estaba considerando. Por lo tanto, para acoplar la señal de entrada de la interfaz con la señal de salida de los módulos se tuvo que implementar un circuito amplificador. La explicación detallada del acoplamiento de la señal será desarrollada más adelante dentro de este mismo capítulo.

3.3. La adquisición de variables provenientes de los módulos.

3.3.1. Acondicionamiento de la señal

Como se mencionó en el punto anterior, se utilizó un circuito amplificador para acoplar la señal. Después de considerar las características que necesitaba el circuito, se tomó la decisión de escoger el amplificador LMC660. Este circuito es fabricado por la empresa National y consiste en un circuito integrado con cuatro amplificadores independientes dentro del mismo encapsulado. Lo anterior constituyó una ventaja debido a que en la interfaz entrarían simultáneamente cuatro señales que necesitarían ser amplificadas con lo que un solo dispositivo se amplificarían todas las señales. Entre las características que encontramos en el amplificador se encuentra que tiene un rango de salida “rail to rail”, tiene baja distorsión, puede manejar una carga de 600 Ω.

La configuración que se utilizó para cada amplificador fue la de no inversor, la figura 3.3 muestra las conexiones que fueron realizadas en cada uno de los circuitos amplificadores.

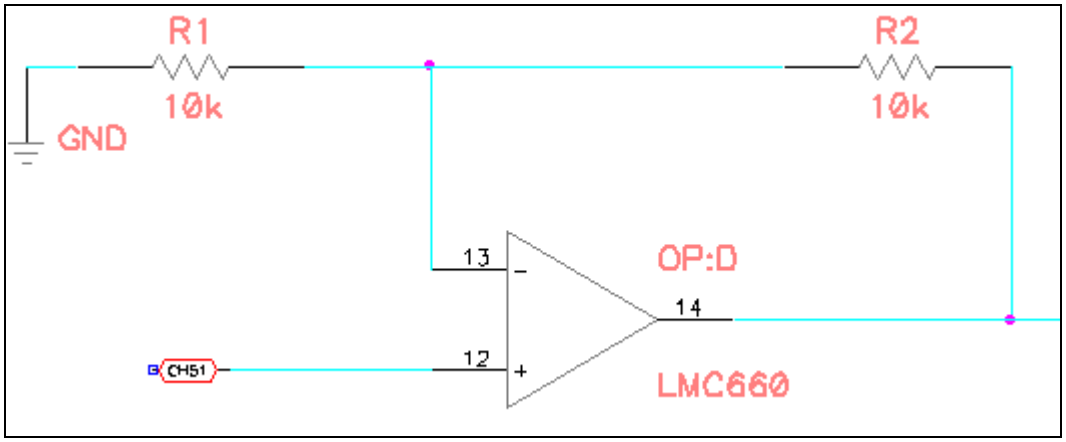


Figura 3.3: Amplificador no inversor

Con esto, cada vez que cualquier módulo conectado a la interfaz midiera una variable, el módulo en respuesta enviaría la señal al amplificador y esta, al ser amplificada por el circuito estaría lista para ser procesada por el convertidor A/D.

3.3.2 Conversión de la señal

Los microcontroladores trabajan con señales digitales, debido a esto, en el proyecto fue necesario convertir las señales analógicas enviadas por los módulos a bits digitales que correspondieran a un dato entendible por el microcontrolador. Esto se logró gracias a un convertidor analógico-digital (CAD). Se puede realizar una investigación tan profunda como el proyecto lo requiera debido a que en el mercado existe una amplia gama de modelos de convertidor disponibles. Estos modelos varían en muchas de sus características, desde como se encargan de hacer la conversión, cuantos canales tienen, hasta el precio y disponibilidad que tienen en el país.

En el caso particular del CAD utilizado en la interfaz, los objetivos de búsqueda estuvieron orientados a conseguir un dispositivo que tuviera una resolución más alta con respecto al sistema anterior. Esto representaba un punto esencial a mejorar debido a que con a una mayor resolución se obtendrían valores más cercanos a los reales, aumentando así la credibilidad del sistema e incluso las áreas de aplicación.

Otro punto que se consideró en la búsqueda del CAD fue su velocidad. Se necesitaba que el funcionamiento del convertidor afectara en la menor medida posible a la velocidad del sistema en general para poder llegar así a las metas previstas.

Al comenzar a averiguar sobre los distintos tipos de convertidores que los fabricantes ofrecían, se observó que, en la mayoría de los casos, las características mencionadas anteriormente estaban relacionadas proporcionalmente con el precio que tenían, por tal motivo, con el fin de seguir con la filosofía de lograr un sistema de costo accesible, se buscó lograr un equilibrio entre la eficiencia y el costo que representaba la adquisición del convertidor.

Con base en esto, se hizo una selección de los modelos más apegados a lo que el proyecto necesitaba, entre ellos tenemos los destacados en la figura 3.4.

Modelo		**MCP3208**	**MCP3304**	**ADC12038**	**ADC12048**	**ADC12138**
Fabricante		Microchip	Microchip	National	National	National
Resolución		12 bits	12 bits màs uno de signo	12 bits más uno de signo	12 bits más uno de signo	12 bits más uno de signo
Frecuencia de reloj		20 MHz	20 MHz	5 MHz	5 MHz	5 MHz
Velocidad		100 ksamples/seg	100 ksamples/seg	----------	216 ksamples/s, min	----------
Tiempo de conversión		----------	----------	14 us, max	3.6 us, max	14 us, max
No. de canales		8	8	8	8	8
Canales_diferenciales		SI	SI	SI	SI	SI
Error de linealidad		+/- 1 LSB, max	+/- 1 LSB, max	+/- 1 LSB,	+/- 1 LSB,	+/- 2 LSB,
Alimentación		5 V	5 V	5 V	5 V	5 V
Consumo	**Modo**	320 uA	450 uA	33 mW	34 mW	33 mW
	Stand by	500 nA	50 nA	100 uW	75 uW	100 uW
COSTO (dólares)		$2.48	$2.49	3.65	5.75	3.7

Figura 3.4: Tabla comparativa de convertidores analógicos-digitales

Después de analizar las características de cada uno de ellos, se llegó a la conclusión de que el dispositivo que convenía más al proyecto era el convertidor MCP3304. En lo general, se observó que existía un equilibrio entre la eficiencia que proporcionaba éste y su costo. Con esto, se cumpln con muchas características que permitirían que el proyecto funcionara de manera óptima, sin tener que involucrar un gasto excesivo. A continuación se explican las más significativas.

Una característica básica para la elección del convertidor fue la resolución que manejaba, con 12 bits, se lograría aumentar considerablemente la resolución con respecto al sistema anterior y a su vez, se lograrían satisfacer las necesidades del sistema sin elevar los costos. Una característica importante del convertidor fue que, al contar con varios pines de entrada, permitiría desarrollar el proyecto sin tener que utilizar más de dos encapsulados. Además, al polarizarse con 5 V cumplía con las

características adecuadas para operar, sin tener que invertir en una fuente adicional exclusiva para él. Para los alcances del proyecto, se consideró satisfactoria la velocidad entregada por el convertidor (100 ksps).

Otra característica favorable fue que la recepción de señales analógicas podía efectuarse de dos formas. Por un lado, las entradas del convertidor se podían usar en modo “Single” con lo cual, se contaba con ocho pines correspondientes a ocho canales de entrada. Por otro lado, los mismos pines de entrada se podían utilizar en modo “Diferencial”, usando así dos pines por cada canal para contar finalmente con cuatro canales por cada circuito integrado.

Como se mencionará más adelante, al igual que el convertidor, el modelo de microcontrolador que utilizamos pertenece a la compañía Microchip. Esto resultó una ventaja adicional ya que se logró mayor compatibilidad en la comunicación de ambos dispositivos, por un lado, facilitando la programación y evitando problemas con el envío y recepción de datos.

Ya elegido el convertidor, se procedió a ser adaptado al sistema. Se estudió el protocolo de envío y recepción de datos que manejaba. Según las hojas de especificaciones notamos que el convertidor contaba con cuatro pines por los cuales controlaba el envío y recepción de datos:

- CS: pin encargado de activar el CAD cuando se le enviaba una señal en alto.

- SCLK: pin que funciona como reloj para controlar los procesos del CAD:

- DIN: pin encargado de recibir, en forma serial, un byte que en cada bit contiene información correspondiente a diversos parámetros útiles para realizar la conversión correctamente (canal que a sensar, single o diferencial, etc.). La información de este byte es preestablecida en un código obtenido de la hoja de especificaciones.

- DOUT: pin encargado de enviar el resultado de la conversión en forma serial.

Se procedió a hacer la conexión del amplificador al convertidor A/D como se observa en la figura 3.5.

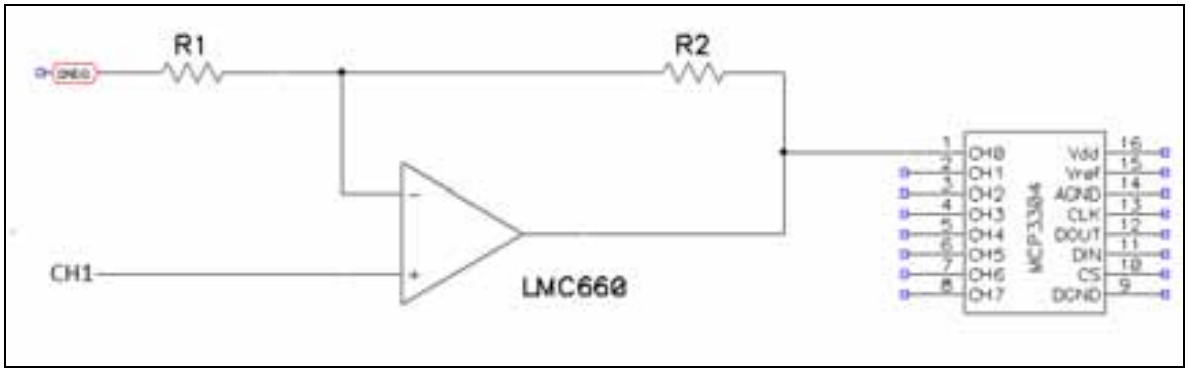


Figura 3.5: Conexión entre amplificadores y CAD.

En el capítulo encargado del firmware de la interfaz se extenderán estos conceptos y se mencionará como fueron aplicados en el proyecto.

3.4. La transmisión y recepción de datos en el microcontrolador.

Como se ha venido mencionando en capítulos anteriores, la parte de control de la interfaz fue delegada a un microcontrolador de la familia de los PIC. Dentro de la variedad de modelos existentes, se decidió buscar un modelo que contara con memoria flash. Debido a esto, se encontró la subfamilia de los PIC16F8XX. Se hizo una comparación con la familia anterior con la que se venía haciendo la interfaz, se encontró que cumplía con características similares y algunas adicionales que nos servirían al rediseñar la interfaz, en la figura 3.6 se muestran características de ambas familias:

Característica	**FAMILIA PIC16C7X**	**FAMILIA PIC16F87X**
PINES	28/40	28/40
TIMERS	3	3
INTERRUPCIONES	11 ó 12	13 ó 14
COMUNICACIÓN	PSP, USART, SSP (SPI, I^2C Slave)	PSP, USART, SSP (SPI, I^2C Slave)
FRECUENCIA	20 MHz	20 MHz
VOLTAJE	2.5 V-5.5 V	2.5 V-5.5 V
CONVERTIDOR A/D	8 bits	10 bits
CCP	2	2
MEMORIA DE PROGRAMA	4K, 8K EPROM	4K, 8K FLASH
RAM	192,368 bytes	192, 368 bytes
EEPROM	Ninguna	128, 256 bytes
OTROS	----------	In-Circuit Debugger, Low Voltaje Programming

Figura 3.6: Familias de microcontroladores PIC16C7X y PIC6F87X

Al estudiar la familia de microcontroladores PIC16F87X, se comenzaron a definir características adicionales que nos serían útiles en nuestro diseño. Después de hacer una búsqueda paramétrica dentro de la página web del fabricante, se agruparon los dispositivos de esa familia que contaran con memoria Flash y que a su vez, tuvieran varios puertos de entrada-salida ya que serían convenientes para desarrollar el sistema. Los resultados se encuentran en la figura 3.7

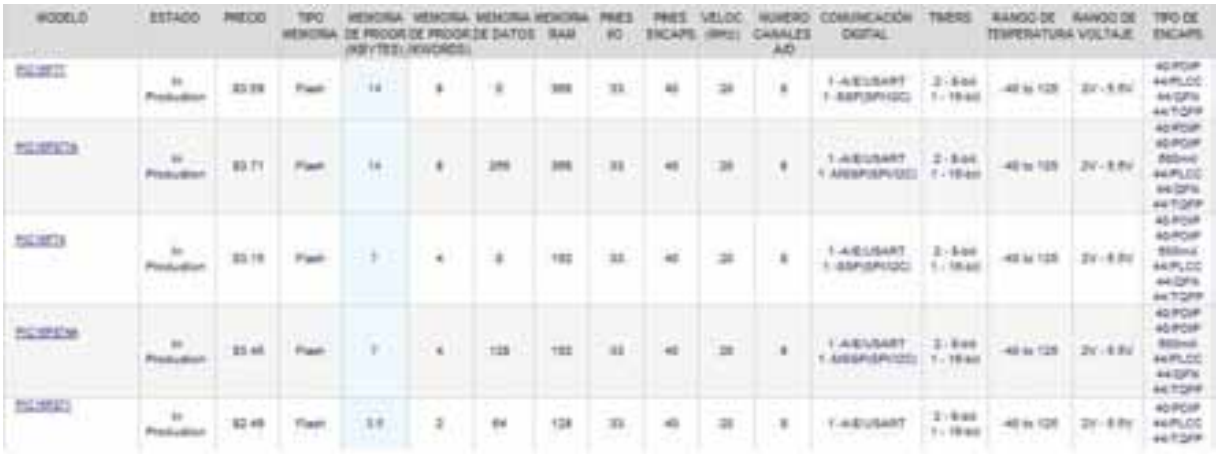


Figura 3.7: Microcontroladores de familia PIC16F87X

Finalmente se decidió utilizar el PIC16F877A. Este microcontrolador dispone de memoria Flash y está fabricado con tecnología CMOS. Su consumo es reducido, a 4MHz utiliza típicamente 2 mA y unos 40 uA funcionando en modo Sleep. El encapsulado es de 40 pines. Por medio de un cristal de cuarzo y dos capacitores puede hacerse funcionar a 20 MHz.

Al igual que los demás dispositivos de la gama, la arquitectura es tipo Harvard Modificado, es decir, dispone de dos memorias independientes: una que contiene solo instrucciones y otra solo con datos. Ambas poseen sus respectivos sistemas de buses y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ellas.

La memoria de programa del PIC (Program Memory) tiene la característica de tener una longitud de palabra de 14 bits. En cada palabra, los bits más altos representan el tipo de instrucción en código máquina (“opcode”) y en los bits bajos puede guardar el resultado de una operación si así le es indicado. Esta es una ventaja con respecto a otros microcontroladores que tienen solamente 8 bits en su longitud de palabra para las instrucciones debido a que se logra mayor velocidad al tener que efectuar menor cantidad de lecturas.

Otra característica es su juego de instrucciones reducido (35 instrucciones) RISC, que en su mayoría se ejecutan en un solo ciclo de reloj excepto las instrucciones de salto que necesitan dos ciclos.

Poseen dos bloques de memoria separados, la memoria de programa y los bancos de registros.

Los bancos de registros RAM son de 8 bits (byte) a excepción del contador del programa que es de 13 bits. Las direcciones bajas contienen los registros especiales y los registros de recursos, encontrándose a continuación los registros de propósito general. El PIC16F877 posee 68 registros que van de la dirección $00 a la $4F, aunque no todas estas se usan.

Cuenta con dos timers de 8 bits y un timer de 16 bits, los cuales fueron útiles en el momento de implementar las interrupciones dentro del firmware. La explicación de este punto se extiende en el capítulo siguiente.

Mencionar por último que los cinco puertos de entrada y salida (E/S) están dispuestos de la siguiente manera: el puerto A con 5 líneas de la RA0 a la RA4, tanto el puerto B, como el C y el D disponen de 8 lìneas de entrada/salida que van de RX0 a RX7. Finalmente el puerto E dispone de 3 lineas que van desde RE0 hasta RE2.

3.5. Transmisión de datos a la computadora

Al estar los datos ya procesados por el microcontrolador, se necesitaba un medio que los enviara a la computadora personal, en el sistema anterior, dicha tarea se hacía por medio del puerto serie. A pesar de que el protocolo del puerto serie sigue siendo utilizado en algunas aplicaciones, en la nuestra representaba una desventaja debido a la baja velocidad que típicamente maneja con respecto a otros protocolos.

Actualmente podemos ver que el protocolo USB (Universal Serial Bus) se está expandiendo en muchas aplicaciones tecnológicas. Por tal razón, se buscó una forma sencilla de que la interfaz contara con comunicación con la computadora a través del puerto USB. Se encontró que la empresa estadounidense DLP Design, fabricaba módulos encargados de comunicar por medio del puerto USB las diversas aplicaciones que un diseñador quisiese crear. Entre las ventajas que encontramos es que era posible que nosotros combinaramos los dispositivos que ya teníamos seleccionados con dichos módulos. Entre la gama de productos que ofrece la compañía consideramos que existían algunos que eran económicos y nos permitirían desarrollar nuestra apliación. Los modelos y características de los más allegados a nuestra necesidad se encuentran en la figura 3.8.

Modelo	**DLP-USB245M-G**	**DLPUSB232M-G**	**DLP245PB-G**
Descripción	Módulo Puerto paralelo-USB	Módulo Puerto Serial-USB	Módulo USB-FIFO con PIC16F877A integrado
Velocidad de envío/recepción de datos	Arriba de 1 Mbyte/seg	1 Mbaud	Arriba de 1 Mbyte/seg
Tamaño de buffer de envío	384 bytes	384 bytes	384 bytes
Tamaño de buffer de recepción	128 bytes	128 bytes	128 bytes
Alimentación	4.4-5.25 V	4.4-5.25 V	4.4-5.25 V
Drivers para PC	Virtual COM Port	Virtual COM Port	Virtual COM Port
Precio (dólares)	20	20	40

Figura 3.8: Tabla comparativa de módulos USB

Considerando lo anterior, se decidió utilizar el módulo DLPUSB245M-G

3.5.1. El módulo de comunicación DLPUSB245M-G

El módulo que utilizamos es diseñado por la empresa DLP Design. Este módulo representó diversas ventajas que fueron útiles para nuestro diseño. Además de un precio bastante accesible (alrededor de veinte dólares), cuenta con una gran versatilidad en su diseño, lo que permitió adaptarlo de manera perfecta a las necesidades de nuestro circuito. Se trata de un adaptador con un puerto paralelo de entrada-salida y un conector USB con dispositivos tales que, logra el envío de información a una computadora por medio de cualquier puerto USB que esta maneje.

Es conveniente mencionar que al hacer la investigación de las opciones disponibles en el mercado con respecto a la comunicación USB, se encontraron algunas alternativas que eran viables en el proyecto.

Por un lado, el fabricante DLP Design además del módulo que se encontró, también ofrecía un módulo que incluía un PIC16F877 adaptado a la misma tarjeta con lo cual se hubiera simplificado el desarrollo de la interfaz debido a que la comunicación entre el módulo DLP y el PIC no se hubiera tenido que programar. A pesar de esta ventaja, la razón por la que se escogió el módulo que se encuentra la interfaz fue que se planea que la siguiente versión de la interfaz tenga comunicación USB directamente a través del PIC. Por lo tanto, al tener desarrollada la programación de la comunicación USB, será más sencillo realizar la siguiente versión del firmware.

Proveé de un método para transferir datos desde/hacia un host por arriba de 1 megabyte sobre segundo. Para enviar datos de un periférico (en nuestro caso el microcontrolador) se debe de escribir un byte en los pines correspondientes al puerto paralelo y después controlarlo por medio de pines de control desde el microcontrolador. En el caso de la recepción de datos desde el host hacia el periférico, el módulo enviará una señal que indicará que existe un dato en el buffer

con lo que podremos hacer las acciones necesarias en el microcontrolador para recibir el dato. El módulo tiene un buffer de transmisión de 384 bytes mientras que el de recepción es de 128 bytes.

En cuanto al protocolo USB, el módulo se comunica con la computadora por medio de puertos “virtuales” COM y sus respectivos controladores proporcionados por el fabricante. Estos controladores hacen que el periférico luzca como un puerto COM estándar para la computadora. Debido a esto, no es necesario que el diseñador agregue un controlador extra en su diseño añadiendo así un punto a nuestro favor.

Una característica del dispositivo DLP que también fue importante para nuestro diseño es la referida a la compatibilidad de su puerto paralelo con el microcontrolador. Esto debido a que el PIC elegido contaba también con un puerto paralelo al cual, haciendo las adaptaciones correspondientes, lograríamos que se pudiese comunicar con el módulo. Con esto nos percatamos de que se obtendría un incremento considerable en la velocidad de transmisión de datos hacia la computadora ya que, de esta forma, se prescindiría del puerto serie.

3.6. Pruebas

3.6.1. Pruebas de depuración

El sistema fue probado para servir de manera óptima al conjuntarlo con el software. Debido a que este último fue diseñado paralelamente con la interfaz, en un principio se tuvo que resolver el problema usando un programa encargado de controlar puertos COM. El software elegido fue el llamado COM-PORT Tool Kit. Entre otras funciones, este programa permite controlar un puerto COM de la computadora previamente especificado por el usuario.

El programa se encarga de enviar o recibir los datos en tiempo real. Cuenta con una pantalla de salida que tiene como función imprimir los datos, tanto en código hexadecimal como en ASCII. A su vez, muestra el momento exacto en que los datos fueron transmitidos desde o hacia el puerto. A continuación, en la figura 3.9, se muestra una pantalla típica del programa.

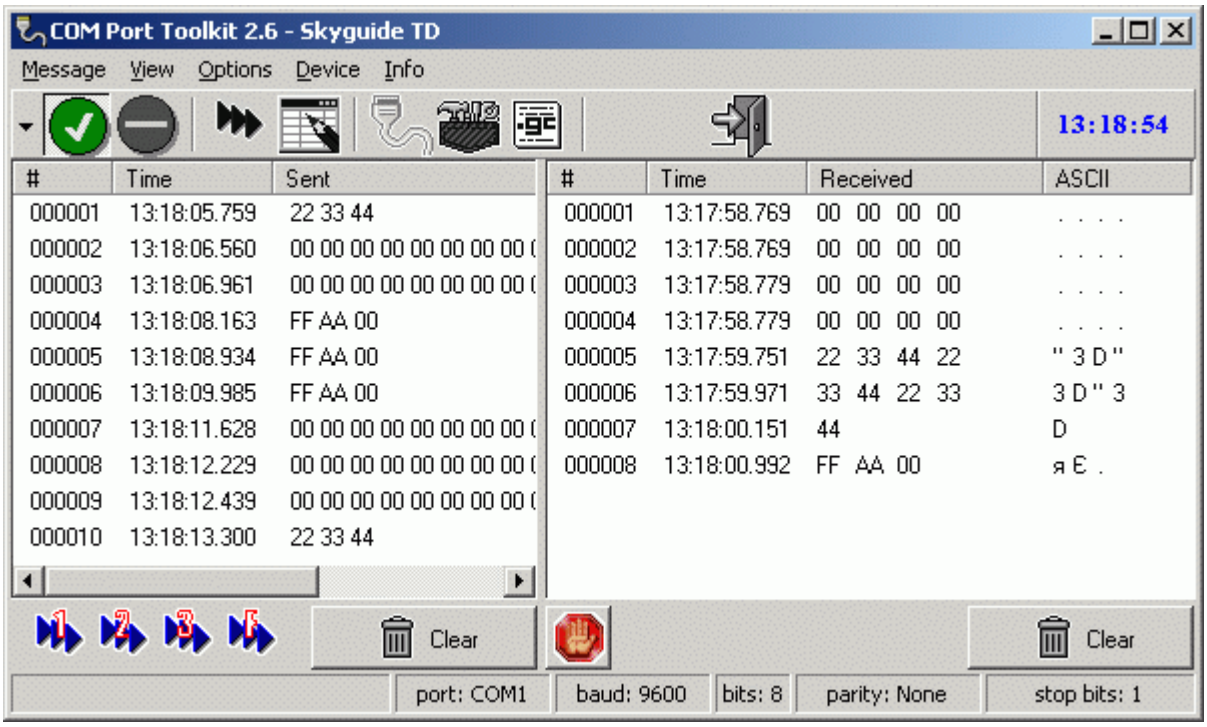


Figura 3.9: Pantalla típica del programa COM Port Toolkit

Cabe mencionar que el programa fue una gran ayuda debido a que en él se detectaron muchas de las fallas de transmisión de manera sencilla y confiable.

3.6.2. Acoplamiento con el software

Después de haber probado el programa con el COM-PORT Tool Kit ,el acoplamiento con el software desarrollado en el laboratorio fue hecho de manera eficaz debido a que existían pruebas previas en donde se simulaba la comunicación que se tendría con el software. Se ajustaron los parámetros necesarios y finalmente, en las figuras 3.10 y 3.11 se muestra el resultado de la unión entre el software y hardware.

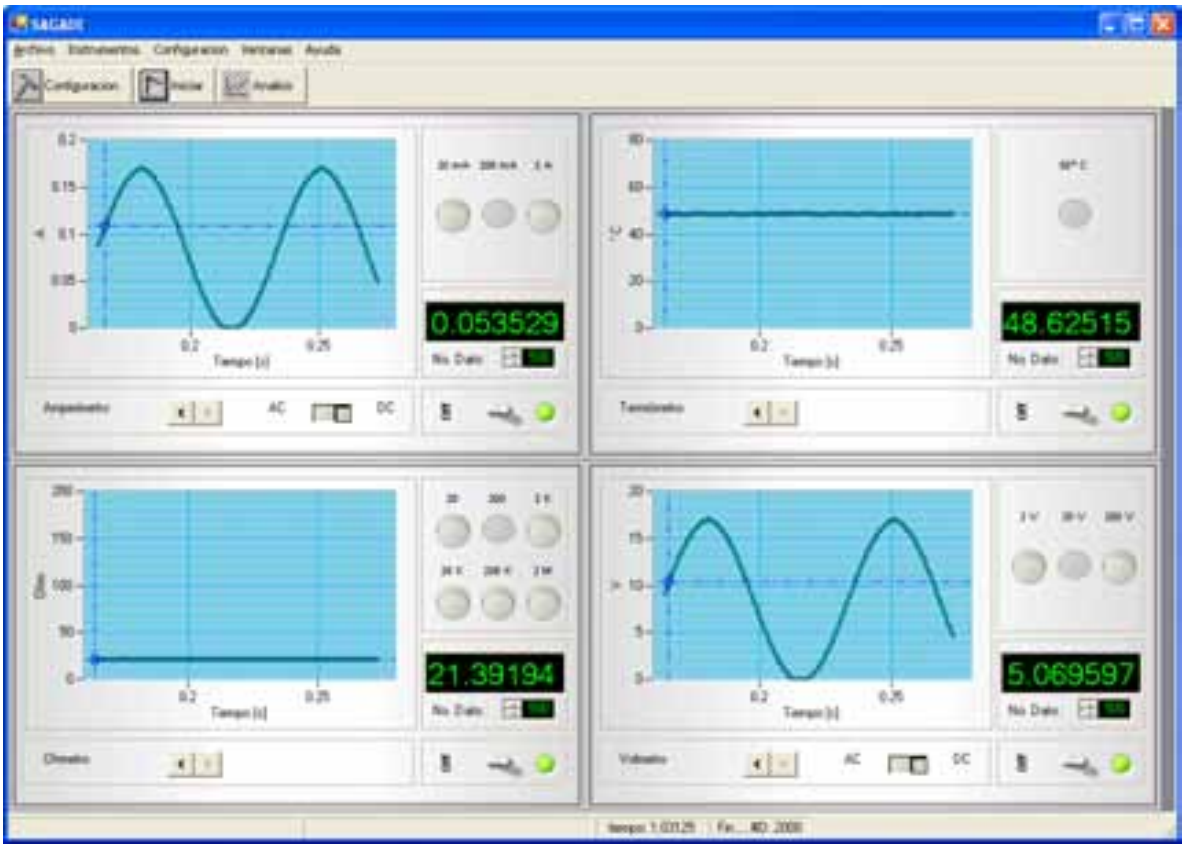


Figura 3.10: Ventana del software durante la adquisición de señal en cuatro canales.

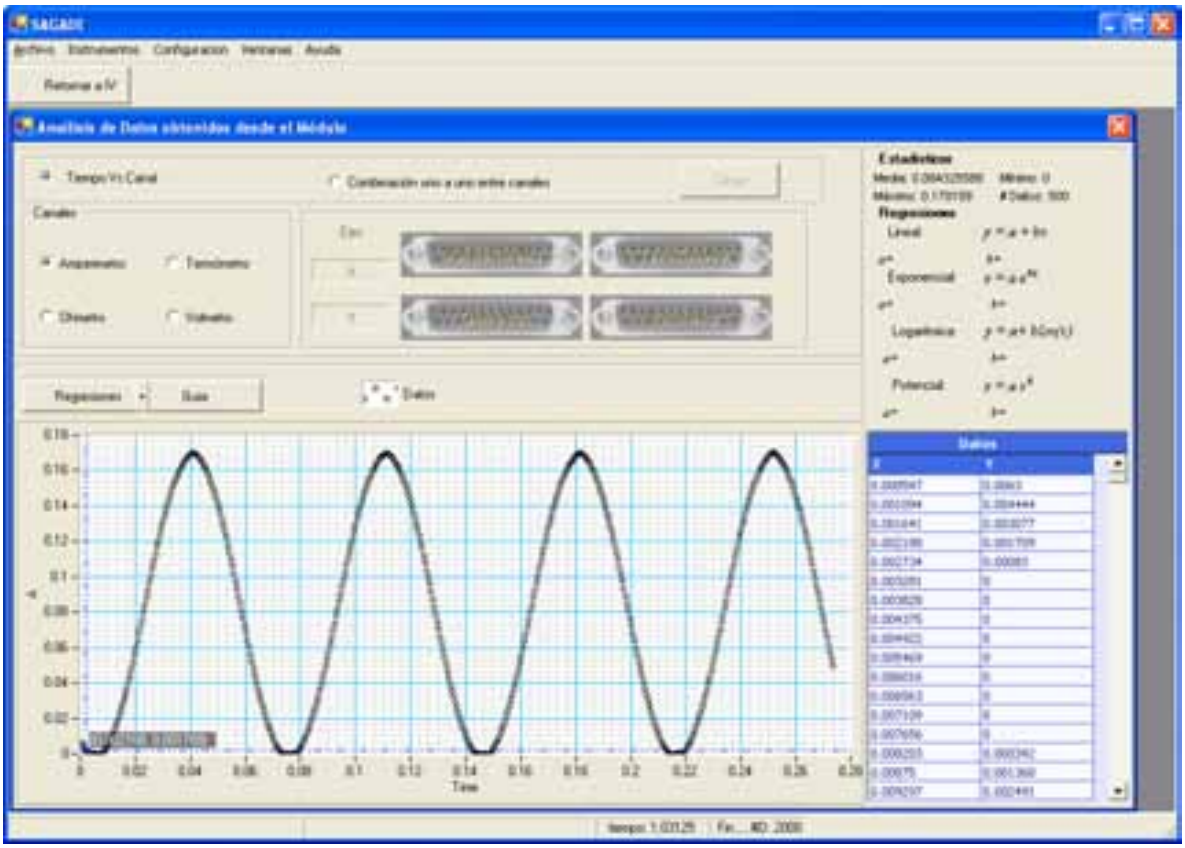


Figura 3.11: Ventana software durante la graficación a detalle, mostrando la tabla de datos y las opciones de procesamiento de datos experimentales.

4. DISEÑO DEL FIRMWARE

En este capítulo se presentarán los parámetros que fueron utilizados para lograr el desarrollo de firmware encargado de controlar la interfaz. Cabe mencionar que las explicaciones del funcionamiento de los dispositivos electrónicos involucrados ya han sido expuestas anteriormente. Esta sección está enfocada en explicar de manera detallada como fue realizada la programación del microcontrolador.

El programa fue desarrollado en lenguaje C y fue compilado, probado y simulado bajo el ambiente de MPLAB versión 7.8.

En primera instancia, el capítulo está dedicado a plantear la estructura general del programa. En general, explica las funciones encargadas de llevar el orden en los procesos de control de la interfaz. Así mismo se da un panorama general de la forma en que fueron planteados los protocolos de comunicación que se utilizaron en el sistema.

Terminado esto, en el siguiente fragmento del capítulo, se explica a detalle como fue realizada la comunicación del microcontrolador con el convertidor analógico-digital. El protocolo fue implementado principalmente mediante dos funciones que se desglosarán en dicha sección.

Finalmente, en el último segmento del capítulo se plasma la forma en la que se efectuó la comunicación entre el módulo USB y el microcontrolador y a su vez, como fueron enviados los datos a la computadora personal.

4.1 Desarrollo del firmware

Esta sección involucra las funciones del programa que van a estar dedicadas a los procesos de control que se harán en el microcontrolador, ya sea para dar formato a los datos que se enviarán a la PC, mandar llamar funciones, escoger protocolos a usar, así como hacer la cuenta de tiempo para enviar los datos en el momento adecuado.

Para comenzar a entender la filosofía que se siguió para realizar el desarrollo del programa se debe mencionar que, para que el sistema en conjunto tuviera dos formas distintas de funcionar, se decidió realizar dos tipos de protocolos de comunicación. El primero, denominado "protocolo P", que permite que el usuario introduzca el número de muestras que necesiten ser sensadas en cada canal y que el sistema las obtenga a su máxima capacidad, en cuanto a velocidad se refiere. El otro protocolo desarrollado fue el “protocolo N”, en el cual, el usuario podrá obtener muestras de cada canal sensado a un determinado tiempo entre cada una de ellas; este tiempo será solicitado por el mismo usuario.

En el diagrama siguiente, se observa de manera general la lógica que se siguió para hacer la programación en el “protocolo P”:

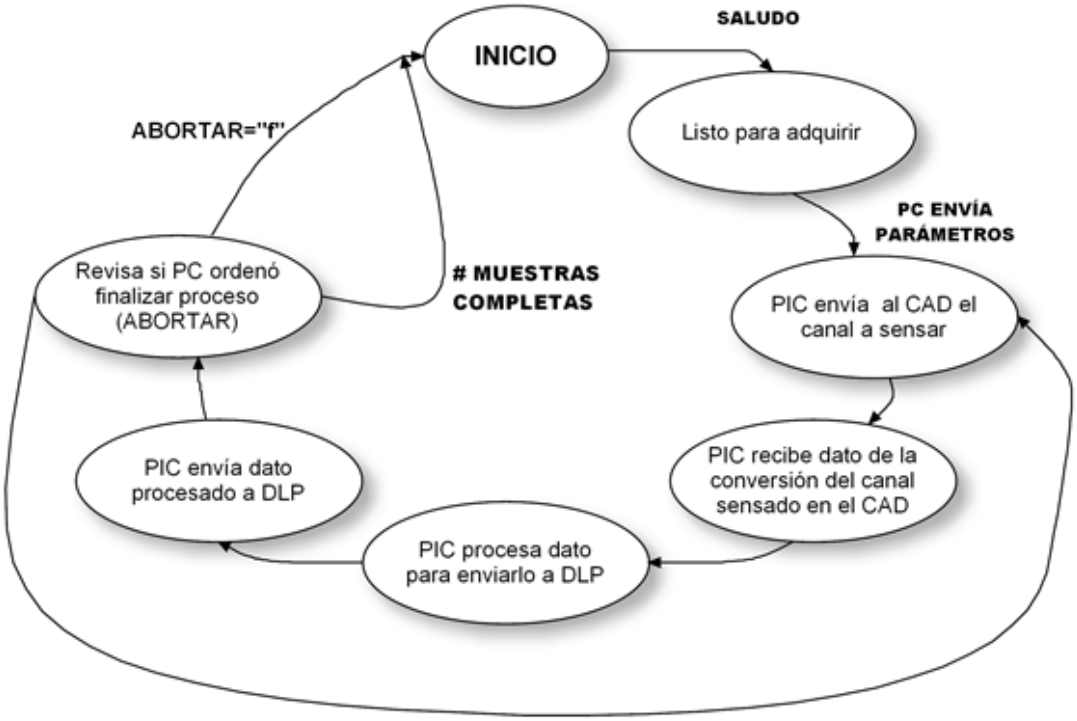


Figura 4.1: Funcionamiento del protocolo P

En el diagrama se puede observar que, para el protocolo P, el PIC estuvo encargado de controlar el funcionamiento de los dispositivos principales de la interfaz como lo son el convertidor analógico-digital y el dispositivo DLP. En lo general, el PIC va a estar encargado de seguir una rutina de tal forma que en el momento que la computadora, a través del DLP, le envíe los parámetros solicitados por el usuario, éste va a comenzar un proceso cíclico en el cual, los canales van a ser sensados el número de veces que el usuario haya solicitado hasta completar el número de muestras requerido. A partir de que el PIC envía los parámetros de la primera muestra al CAD, el sistema sensará repetidamente cada uno de los canales que le hayan sido requeridos, a menos que la computadora le envíe un carácter (“f”) al PIC por medio del DLP. Si este carácter no es enviado, la interfaz estará en un proceso que no se detendrá hasta finalizar el número de muestras que se le hubiese solicitado

Por otro lado, en la siguiente figura podemos ver el diagrama general del “protocolo N”. El diagrama es muy parecido al del “protocolo P” en cuanto a los procesos de envío y recepción de datos que se realizarán en el microcontrolador, sin embargo, el diagrama del protocolo N varía en los instantes en que se estarán enviando cada uno de dichos datos al DLP. El momento en el que el envío de las muestras (sensadas y después procesadas por el PIC) se generará a través de una interrupción de tiempo con la que cuenta el microcontrolador (TIMER 1) . Esta interrupción irá contando el tiempo hasta llegar al requerido por el usuario y en este momento, dará las órdenes para enviar los datos procesados previamente por el microcontrolador hacia el módulo DLP.

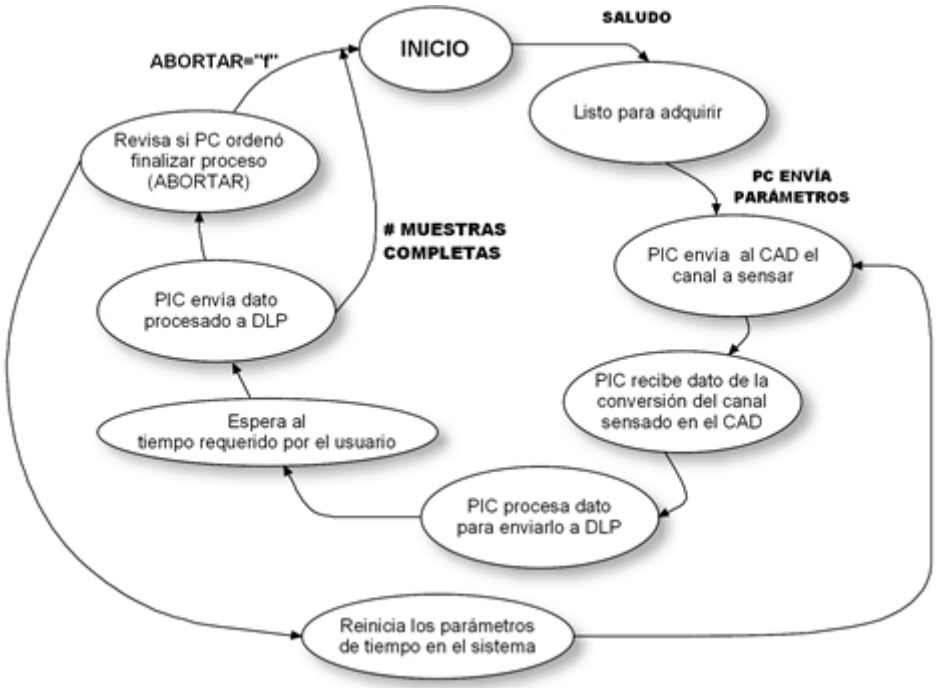


Figura 4.2: Funcionamiento del protocolo N

Considerando los diagramas anteriores, a continuación se desarrollará la explicación de las funciones principales que fueron creadas para que la interfaz trabajase de manera exitosa para ambos protocolos, las cuales, si son sumadas a las funciones que son usadas para la comunicación entre los dispositivos a través de protocolos, nos darán un panorama completo del firmware que se creó para la interfaz.

Primeramente se describe la función INICIO que, como es de esperarse, va a ser la primera que se ejecute cada vez que comience el programa al encender la interfaz.

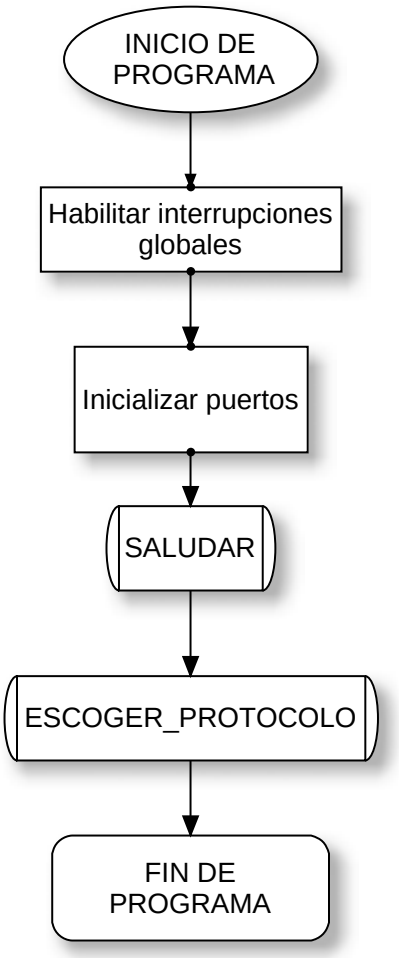


Figura 4.3: Función INICIO

Después de que la interfaz es encendida, el programa habilita las interrupciones e inicializa los puertos que se utilizarán durante el programa. Teniendo esto habilitado, se dirigirá a la función SALUDAR y al salir de esta última regresará para entrar a la función ESCOGER_PROTOCOLO. Cabe mencionar que a partir de la función ESCOGER_PROTOCOLO, se irán llamando a otras funciones que se encargarán del resto del proceso.

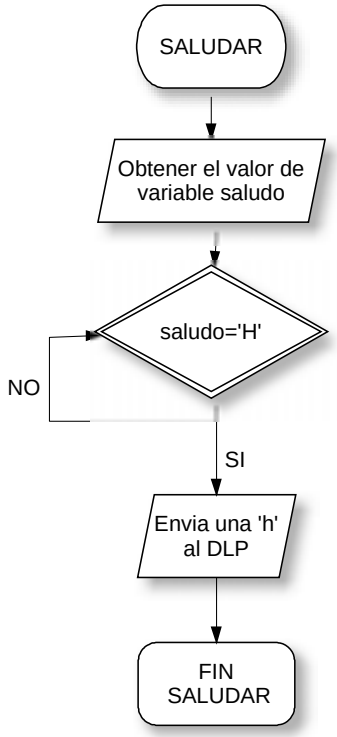


Figura 4.4: Función SALUDAR

La función SALUDAR se encarga de esperar un carácter de saludo entre el software y el hardware. Esta función es útil debido a que con ella es posible corroborar que existe la comunicación correcta entre la interfaz y la computadora a la que esté conectada.

En este caso, el microcontrolador, mediante un ciclo, va a esperar a que el software le envíe un carácter específico (“H”) y en el momento que esto ocurra, el ciclo finalizará dando lugar a que la interfaz en respuesta envíe otro (“h”) que será recibido por la computadora. Con esto, tanto la computadora como la interfaz interpretarán que las dos están listas para el envío y recepción de datos.

La función ESCOGER_CANAL tiene como objetivo llevar el orden en el que se irán haciendo cada una de las conversiones que el usuario haya solicitado.

La función consiste en un ciclo que se repetirá tantas veces como muestras sean requeridas. Su función será la de enviar cada uno de los canales que se van a sensar y llamará a la función llamada CONVERSIÓN que se encargará de hacer directamente la conversión para que finalmente pueda enviarse el resultado de la primera muestra correspondiente al primer canal solicitado al DLP. Terminado el envío de este primer dato se irá repitiendo dicho proceso para cada canal solicitado por el software de la computadora. El ciclo finalizará cuando todos los canales solicitados hayan sido sensados una vez y el número de muestras requeridas hayan sido enviadas hacia el módulo DLP para que a su vez, este se encargue de enviarlas al software de la computadora.

Un detalle importante a considerar es que cada vez que el ciclo se haya completado, la siguiente instrucción va a enviar al programa a otra función

denominada DLP_ENVIA_AL_PIC_ABORTAR, la cual se va a encargar de verificar si la PC ha mandado el caracter correspondiente con el cual se avisaría si el usuario mandó finalizar la actividad y de haber sido así, reiniciar el programa del PIC y a su vez deshabilitar la interrupción del timer llamada TIMER1_isr.

En caso de que no exista la reinicialización, el programa verificará en que protocolo se había comenzado el sensado, si encuentra que es en el protocolo N, verificará el numero de muestras por canal que hayan sido sensadas, si este no supera al numero de muestras que el usuario pidió, volverá a cargar los valores de los registros necesarios y el valor del timer para reestablecer la interrupción por tiempo y terminando esto, esperará a que dicha interrupción se efectúe para continuar procesando la siguiente muestra.

En caso de haberse comenzado con el protocolo P, simplemente el programa saldrá de la rutina y seguirá leyendo la siguiente instrucción. El diagrama de flujo mostrado a continuación, representa esta función.

Dentro de los parámetros que el DLP envía al PIC se encuentra el “j”, este va a indicar cuantos canales se van a sensar según lo que haya elegido el usuario.

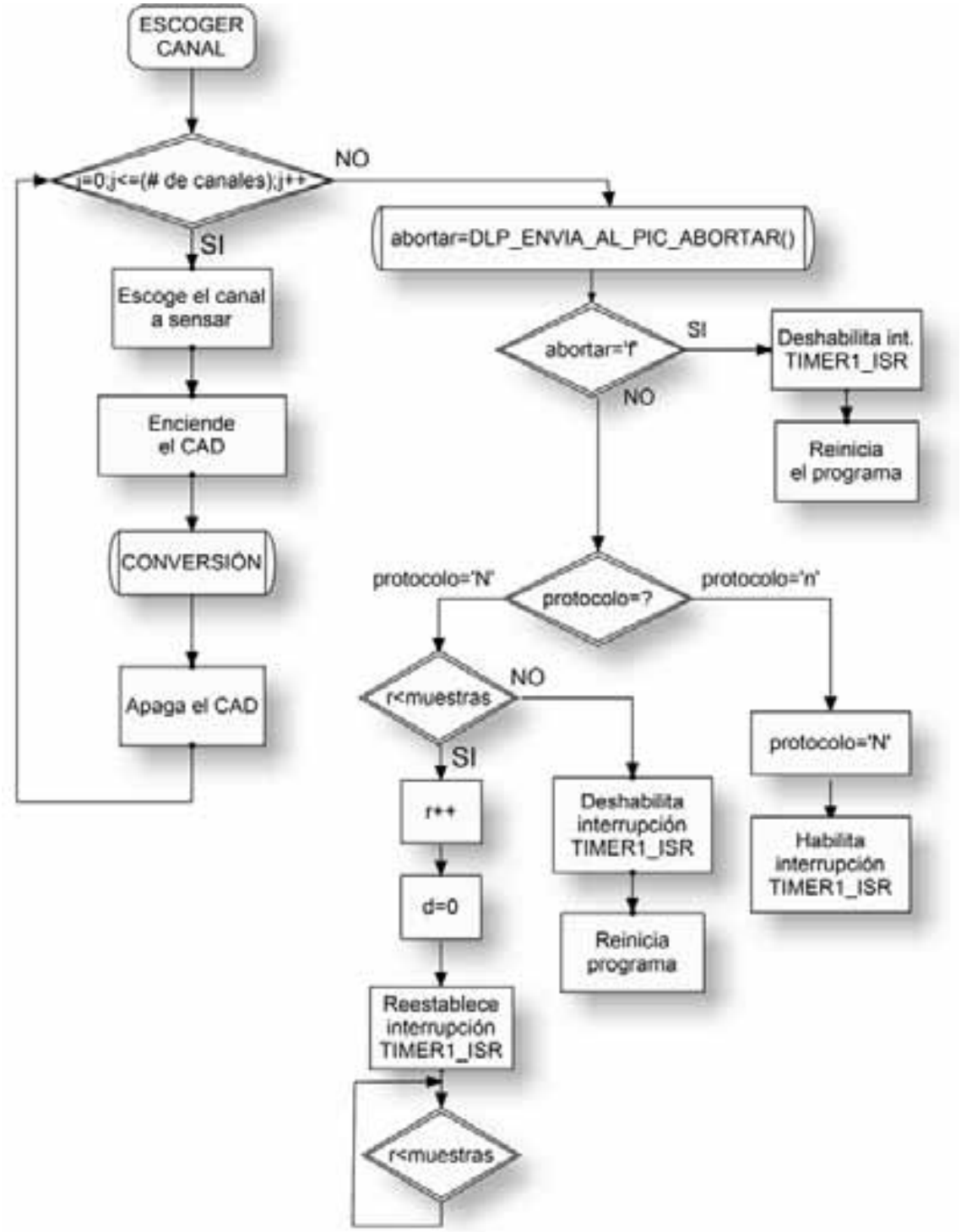


Figura 4.5: Función ESCOGER_CANAL

4.1.1.CONTROL DE CONVERTIDORES A/D

La siguiente función a la que se hará referencia es denominada CONVERSION, mostrada en la siguiente figura. Esta función es la parte fundamental para poder controlar al convertidor A/D. En este caso, se usaron los convertidores en modo “SINGLE” y no en modo “DIFERENCIAL”, con lo cual, según la estructura del dispositivo, se puede contar hasta con ocho

canales disponibles en él.

En un principio, la función CONVERSIÓN se va a encargar de recopilar datos que el convertidor necesita para iniciar la conversión, dichos datos son: modo en que va a trabajar el convertidor y el canal que vamos a utilizar. Teniendo estos datos guardados, la función se encargará de controlar cada uno de los procesos que el CAD necesita para realizar la conversión. A continuación se dedscribe cada uno de estos procesos mediante diagramas de flujo.

Para comprender mejor dichos diagramas, es conveniente mostrar la siguiente tabla que contiene los nombres asignados a cada pin de entrada/salida del convertidor A/D MCP3208 y que fueron utilizados para la programación:

PIN	**FUNCIÖN**
CH0-CH7	Canales de entrada al CAD (entradas)
CLK	Reloj serial (entrada)
D_{IN}	Entrada de datos serial
D_{OUT}	Salida de datos serial
CS_1	Pin de encendido del CAD 1
CS_2	Pin de encendido del CAD 2

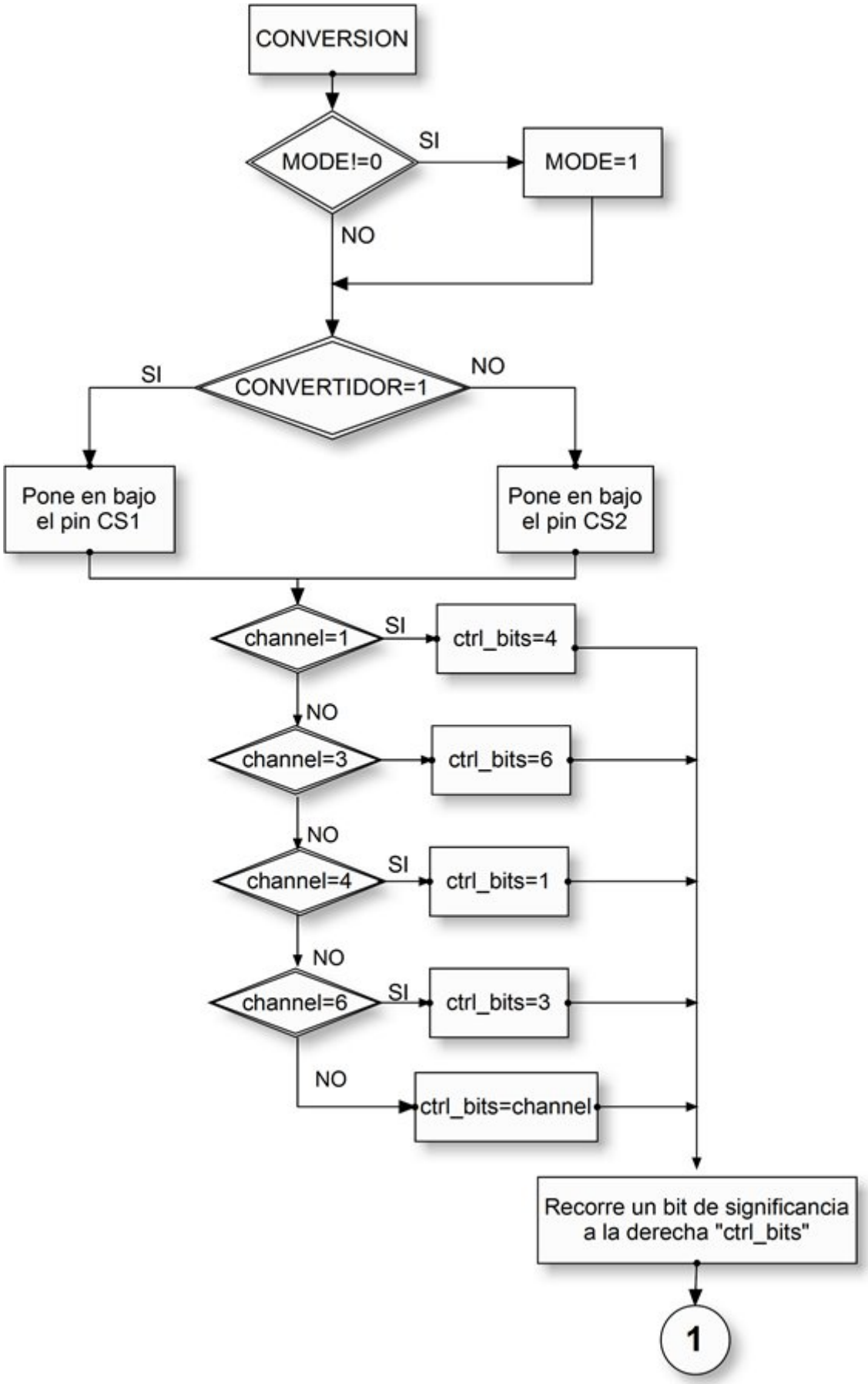


Figura 4.6: Función CONVERSIÓN

Cuando los datos anteriores hayan sido recibidos, el microcontrolador, por medio de la función CAD_RECIBE_DEL_PIC, se encargará de codificarlos en un "byte de control" al convertidor (ctrl_bits). La función CAD_RECIBE_DEL_PIC se explicará posteriormente. Para un mayor entendimiento, es importante mencionar que la variable llamada "ctrl_bits" está conformada por un pequeño código que fue creado por el fabricante del convertidor. Este código contiene la información del canal que se va a utilizar y el modo en el que se usará el convertidor. En la siguiente tabla se muestran los valores utilizados en este código para cuando es utilizado el modo "SINGLE"

.

ctrl_bits*				**Configuración de la entrada**	**Canal seleccionado**
Single/ $\overline{Diferencial}$ (M)	D2	D1	D0		
1	0	0	0	**Single**	**CH0**
1	0	0	1	**Single**	**CH1**
1	0	1	0	**Single**	**CH2**
1	0	1	1	**Single**	**CH3**
1	1	0	0	**Single**	**CH4**
1	1	0	1	**Single**	**CH5**
1	1	1	0	**Single**	**CH6**
1	1	1	1	**Single**	**CH7**

*La posición que tomarán los bits de la tabla dentro del byte "ctrl._bits" será la que se muestra a continuación:

ctrl_bits=SM$D_2D_1D_0$XXX

DONDE:
S=:start bit;
M=modo(single o diferencial);
$D_2D_1D_{0=}$valor para indicar el número de canal a utilizar;
XXX=don´t care bits

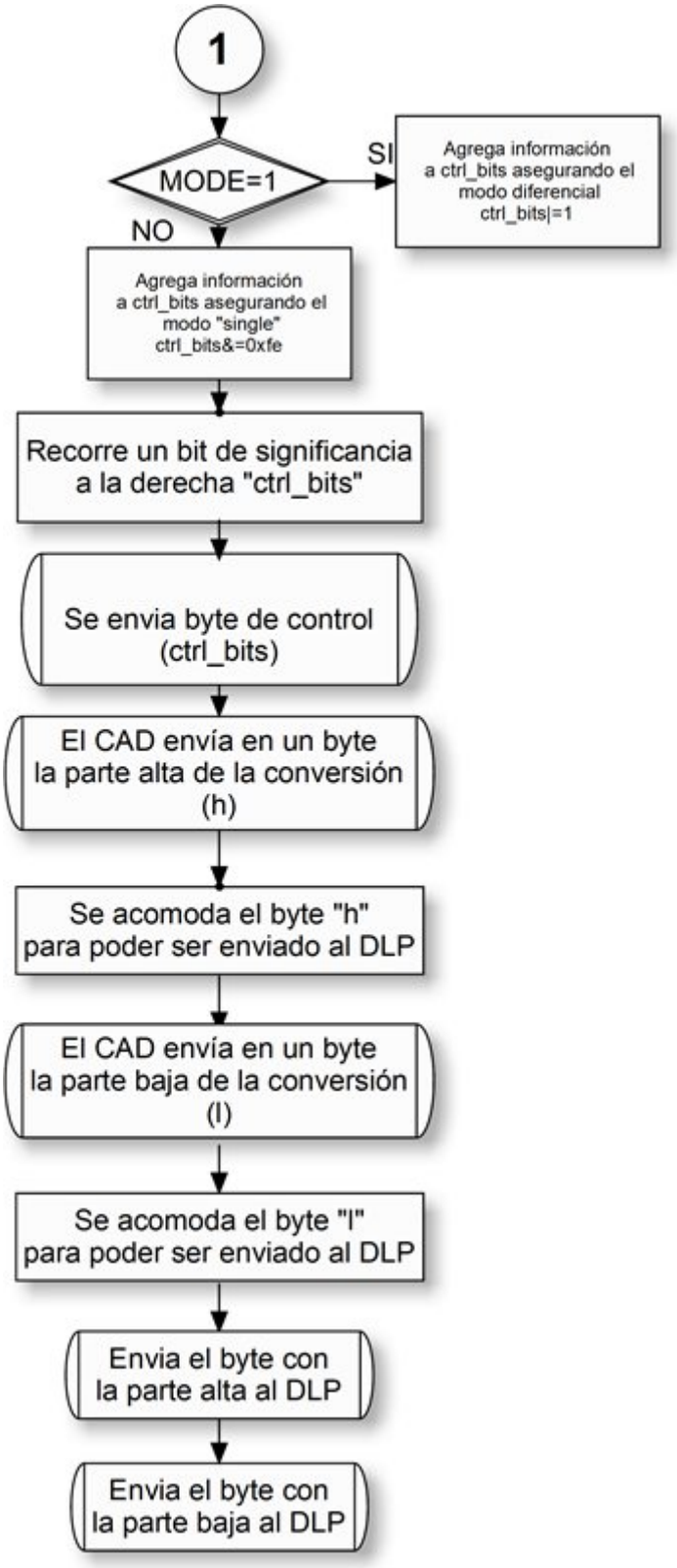


Figura 4.7: Cont. Función CONVERSIÓN

El convertidor iniciará la conversión sensando el canal correspondiente y mandando de forma serial por el pin DOUT la conversión efectuada. Esta conversión será enviada del CAD al PIC. Por medio de la función CAD_ENVIA_AL_PIC (explicada más adelante).

Al estar usando el CAD MCP3208 la información se mandó de forma serial de manera sincronizada con el reloj de pulsos de entrada con el que cuenta el CAD. El primer bit enviado por el CAD fue el de mayor significancia y fue recolectado a través del microcontrolador PIC. El CAD fue mandando cada uno de los bits de salida y el PIC los obtuvo de tal forma que para la parte alta de la conversión se obtuvo un primer paquete (h) que contenía 6 bits que representaron los bits de mayor significancia de la conversión y el segundo paquete contenía la parte baja (l) con los 7 bits de más baja significancia de la conversión enviada en sistema binario.

4.1.2. CONTROL DEL ENVIO DE DATOS

Como fue explicado anteriormente, al realizar el programa, se plantearon dos protocolos para el envío de datos entre la computadora (conectada al módulo DLP) y el PIC, la diferencia entre uno y otro va a depender del usuario. Si el usuario le pide al software de la computadora un tiempo determinado entre un dato y el subsecuente, el PIC va a trabajar con el protocolo “N”; en cambio, si el usuario solo le pide al software que el sistema le envíe un cierto número de muestras, sin importar el intervalo de tiempo entre una y otra, se utilizará el protocolo “P”

4.1.2.1. PROTOCOLO N

El protocolo “N” se respalda en dos funciones: PROTOCOLO_N y TIMER1_ISR.

Comenzaremos describiendo la función PROTOCOLO_N, la cual es la encargada de obtener el valor de la variable tiempo enviada por el software de usuario, así como de habilitar la interrupción por desbordamiento del TIMER 1 y de estar esperando a que se dé dicha interrupción cada vez que sea requerido. En la figura se muestra el diagrama con las tareas mencionadas.

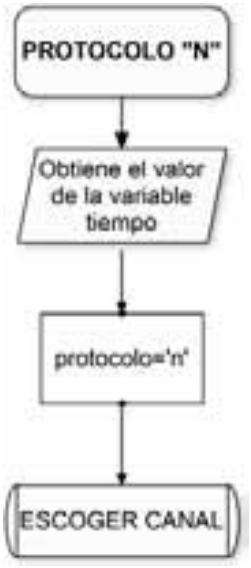


Figura 4.8: Función PROTOCOLO _N

Cabe mencionar que el valor de la variable “d”, no va a llegar a ser mayor que la variable “tiempo” debido a que es un contador que es inicializado en cero y se va a ir incrementando hasta llegar a ser igual que el valor de la variable “tiempo”, para finalmente, después de que se cumpla esta condición, la variable “d” vuelva a ser inicializada en cero. De tal suerte, el único caso en el que se saldrá de este ciclo será cuando la subrutina por interrupción del timer 1 sea llamada. Por tal motivo, la función protocolo “N” se respalda en la función TIMER1_ISR, esta última es llamada en el instante en que el sistema indica que hay una interrupción por desbordamiento del TIMER_1.

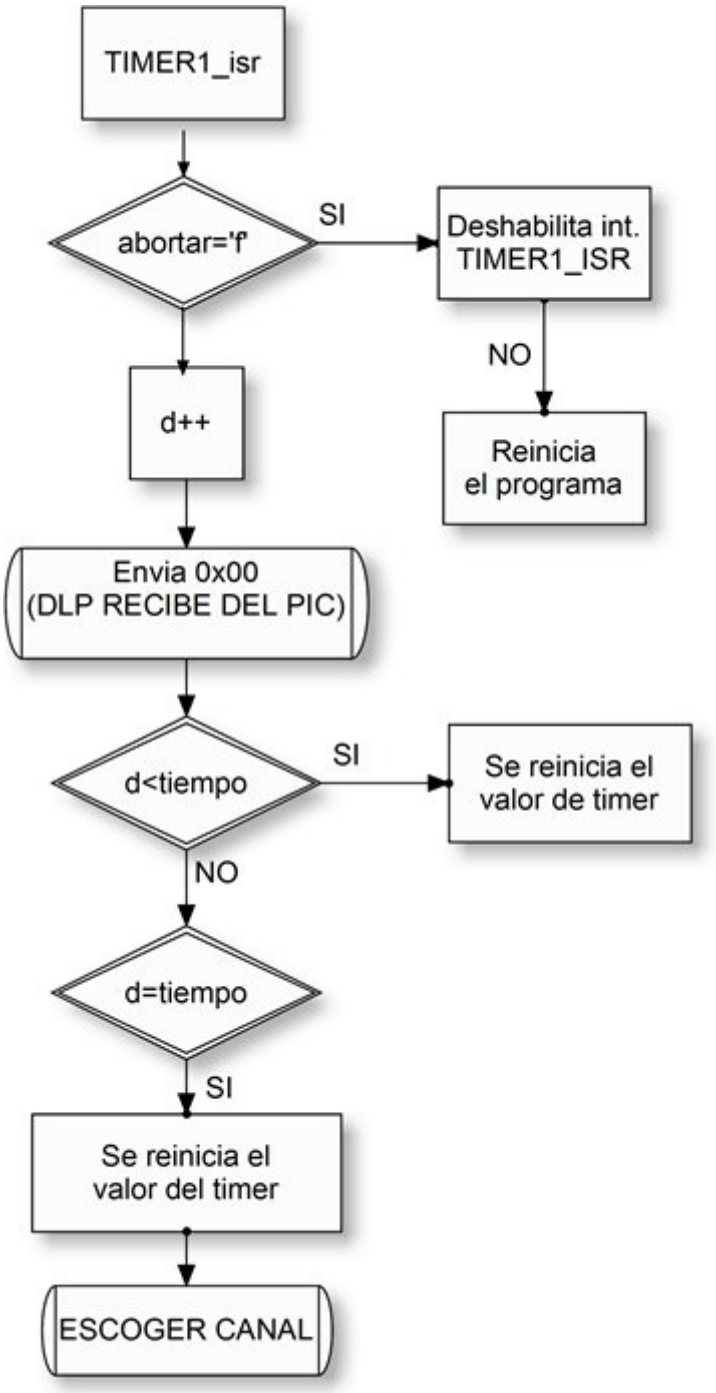


Figura 4.9: Función TIMER1_ISR

Para poder medir el tiempo, se tomó como base que el timer se desbordaría cada milisegundo debido a que es la cantidad mínima de tiempo que el usuario puede solicitarle al software para que el sistema haga el muestreo.

Considerando que el usuario puede pedir que cada dato sea muestreado en una cantidad de tiempo mayor a un milisegundo, hubo la necesidad de implementar un contador llamado “d” cuya misión fuese la de incrementarse cada vez que se llamara a la subrutina de interrupción TIMER_isr. En otras palabras, el contador se estará incrementando cada milisegundo. Con esto, podremos repetir la rutina de interrupción el número de veces que sea necesaria hasta llegar a la cuenta del tiempo pedida por el usuario. En el momento que se cumpla este tiempo, la función va a darle nuevamente el valor de desbordamiento al timer y se trasladará a la función ESCOGER_CANAL que es la encargada de controlar el muestreo, iniciando así el proceso de conversión en el momento que lo haya solicitado el usuario.

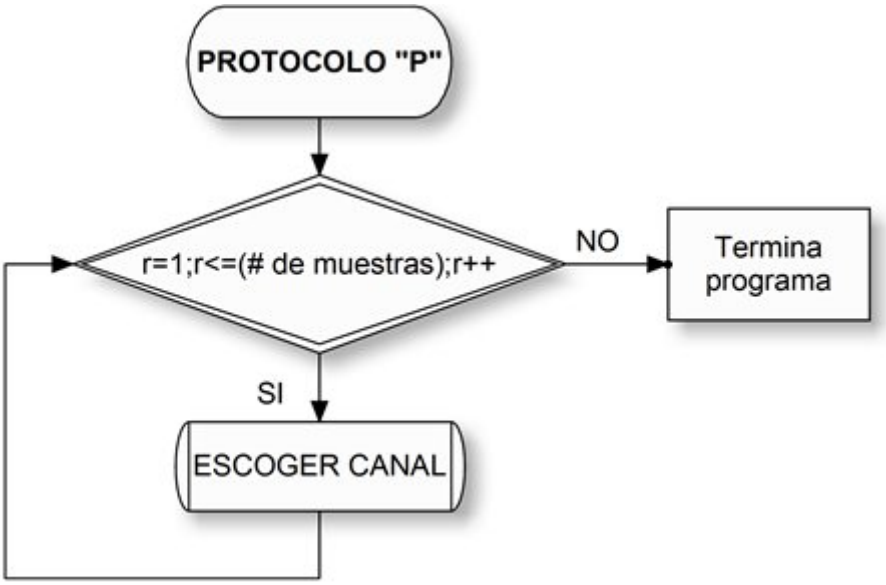


Figura 4.10: Función PROTOCOLO _P

La función protocolo P se va a encargar de coordinar el número de muestras que se van a enviar al módulo DLP. Esto lo hará por medio del contador denominado “r”, cada vez que pase por el ciclo, el contador será incrementado representando que se envió una muestra más al módulo DLP. Después de este incremento y comprobando que el contador “r” siga siendo menor o igual que el número de muestras solicitado. La subrutina llamará a la función ESCOGER_CANAL encargada de controlar la conversión. Cuando el contador llegue al número de muestras que el usuario solicitó, la función termina el programa dando por finalizado el proceso completo.

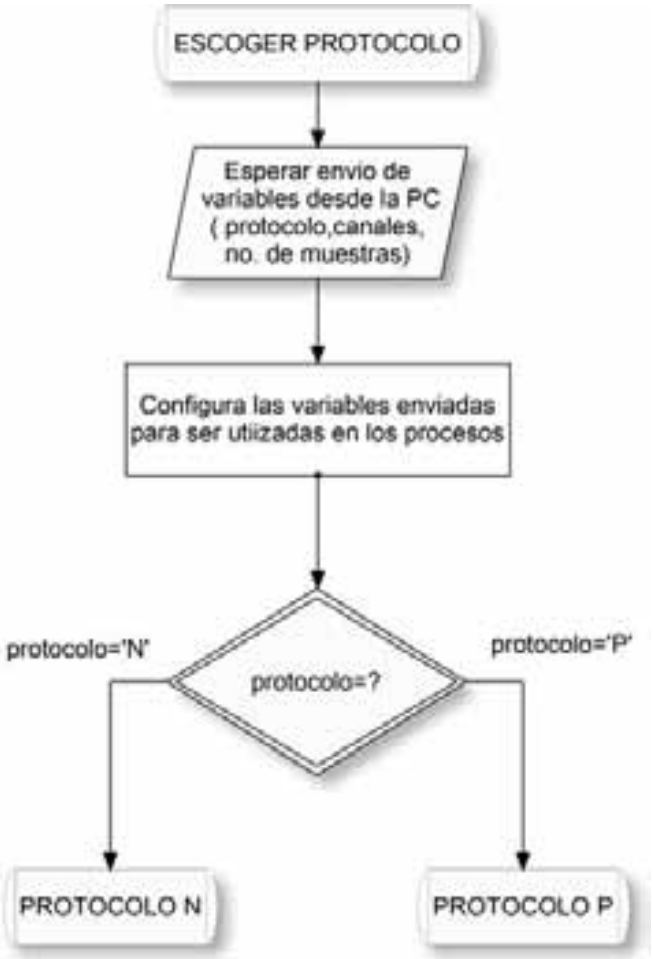


Figura 4.11: Función ESCOGER PROTOCOLO

En caso de que el contador no sea menor o igual al número de muestras solicitado, se considerará que el proceso terminó y que se ha finalizado con la tarea solicitada por el usuario.

La función ESCOGER_PROTOCOLO es la encargada de recibir y almacenar, en las localidades de memoria asignada, cada uno de los datos que le envíe el software de la computadora para indicar los parámetros que el usuario este pidiendo. Estos parámetros son: el protocolo que se va a utilizar, cuantos y en que orden los canales van a ser muestreados, el nombre de cada uno de los canales y por último el número de muestras solicitadas.

Se va a auxiliar de la función DLP_ENVIA_AL_PIC para recibir los datos que se le envían al PIC desde el DLP.

4.2. Protocolo de comunicación entre el CAD y el microcontrolador

Al estudiar la información que requería el convertidor MCP3304, se creó un protocolo de comunicación para él. Los pines del convertidor fueron conectados a los pines del puerto B del microcontrolador en forma serial para lograr el intercambio de información.

Las especificaciones del CAD indican que hay cuatro pines por los cuales es

controlado y que fueron utilizados en el diseño la interfaz.

- CS (activa el CAD al enviarle un “high”).
- SCLK (reloj para controlar los procesos del CAD).
- DIN (encargado de que se le envíe al CAD la información correspondiente del canal a sensar, modo a utilizar, etc.).
- DOUT(envía el resultado de la conversión en forma serial).

Una característica positiva que notamos en este convertidor es que después de realizar la conversión A/D, el resultado se puede enviar en bloques de ocho bits usando un protocolo serial. Debido a lo anterior, en el momento de llegar dichos datos al microcontrolador se facilita el manejo de los mismos dentro de las localidades de memoria. Gracias a esto, cada dato enviado se mandó en paquetes de dos bytes, los cuales por un lado contendrían un byte con la parte baja del dato digital y el otro con la parte alta, quedando además 3 bits libres que, como se mencionará más adelante, también fueron empleados.

En la siguiente figura, se muestra el orden en el que se requerían enviar los datos para que el convertidor MCP3304 pudiese realizar una conversión si se controlaba por un microcontrolador:

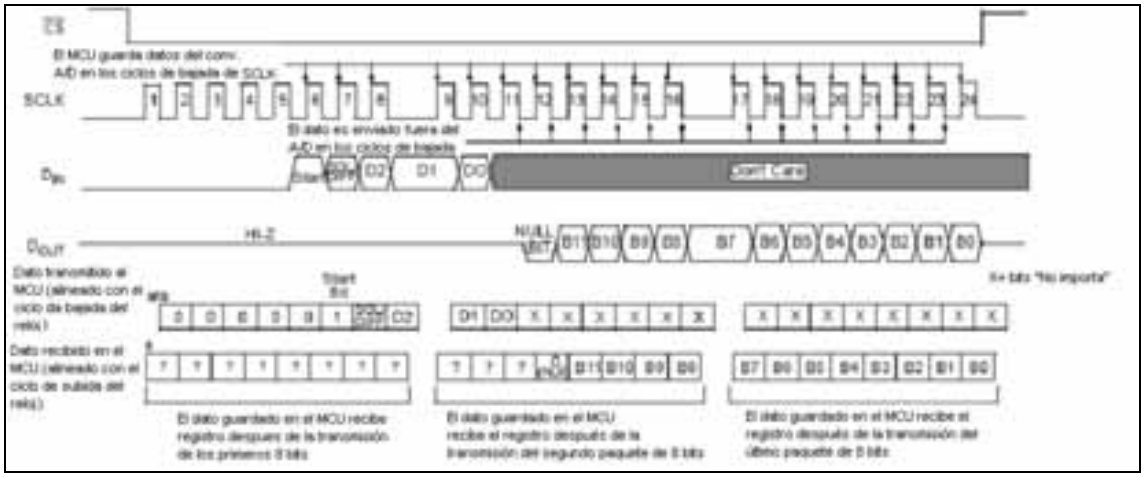


Figura 4.12: Comunicación entre el PIC y el MCP3304

Como se puede observar en la figura anterior, después de la conversión, los datos que manda el CAD al PIC son enviados en paquetes de 8 bits de forma serial. Considerando que el convertidor tiene una resolución de doce bits, para cada dato enviada, el convertidor se encarga de enviar dos bytes por cada dato de 12 bits. El primer byte enviado contendrá los cuatro bits con mayor significancia del dato (en las posiciones de menor significancia del byte enviado) y cuatro “bits basura” (en las posiciones de mayor significancia del byte enviado). El segundo byte enviado contendrá los 8 bits restantes del dato que, al unirlos con los otros 4 bits enviados, se logrará tener el dato completo.

Las siguientes funciones fueron las utilizadas en el programa para el envío y recepción de datos entre el CAD y el PIC:

La función CAD_ENVIA_AL_PIC, tiene como destino recibir en el PIC de

manera serial el resultado de la conversión analógica a digital proveniente de los convertidores A/D. Para lograrlo, cada bit de la conversión, es mandado por medio de los pines interconectados entre los convertidores y el PIC. Es una función sencilla, en donde el PIC se sincroniza con los pines de salida de los convertidores (DOUT) de tal forma que si por el pin DOUT del convertidor en uso se envía un “high”, el pin de entrada del PIC lo recibirá y lo interpretará como un “cero digital” y por el otro lado, si recibe un “high” del convertidor, lo interpretará como un “uno digital.”.

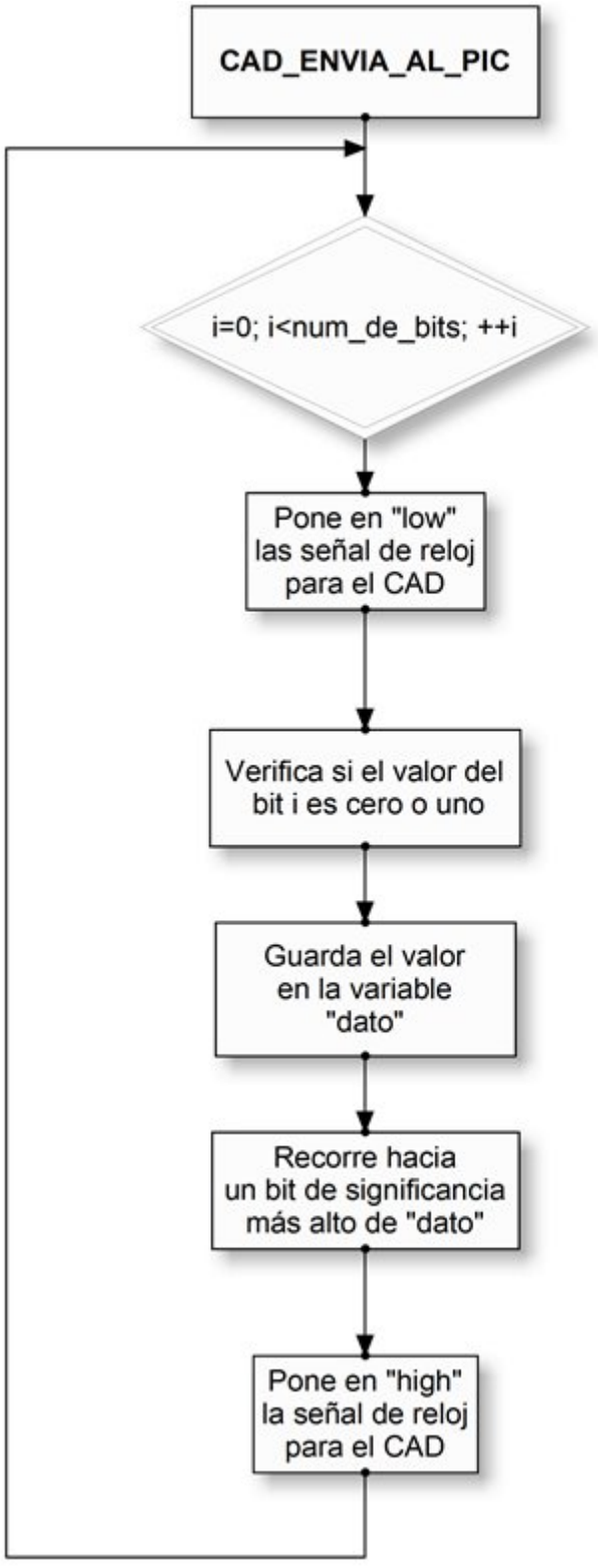


Figura 4.13: Función CAD ENVIA AL PIC

La función está desarrollada de tal manera que se necesita conocer el número de bits que serán enviados hacia el microcontrolador y con esto podrá saber cuantas veces va a tener que interpretar lo que le sea enviado a sus pines de salida.

Del protocolo que siguen los convertidores A/D modelo MCP3304 de Microchip se envía en primer lugar el MSB(More Significant Bit), y después de que avanza el ciclo de reloj por el pin CLK se envía por DOUT el bit con un orden de significancia menor, este proceso se repite controlado por el reloj hasta terminar con el ultimo bit de significancia, mientras que el controlador va guardando estos bits en una sola localidad de memoria la cual se va recorriendo un bit a la derecha cada vez que le llega un bit nuevo.

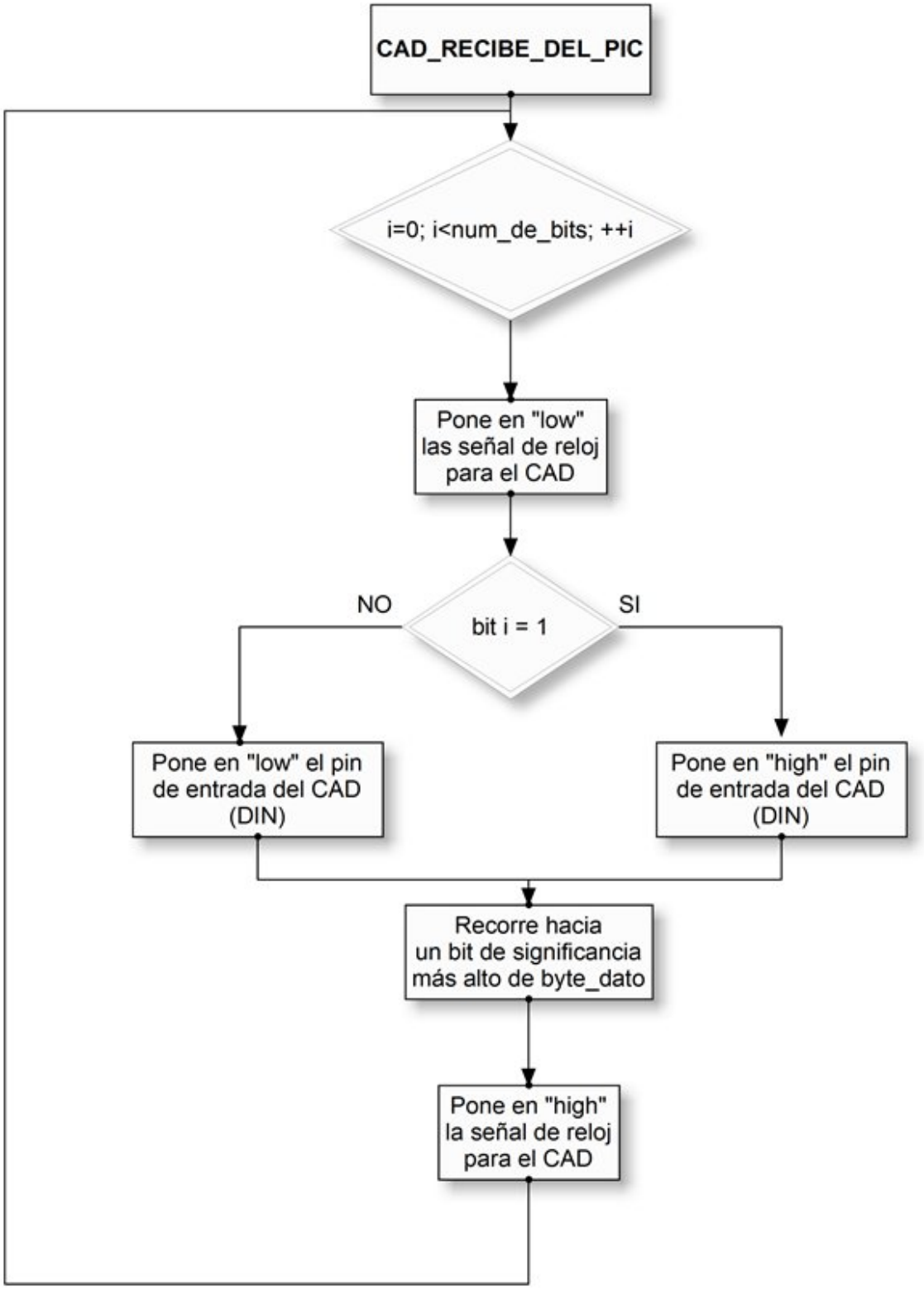


Figura 4.14: Función CAD RECIBE DEL PIC

Como su nombre lo indica, la función CAD_RECIBE_DEL_PIC va a tener como misión el encargarse de enviar desde el PIC los datos que necesite el convertidor para ser controlado, al igual que en la función CAD_ENVIA_AL_PIC, los datos serán enviados al convertidor de manera serial debido a que es la forma como trabaja el convertidor MCP3304.

El envío se sincroniza por medio de una señal de reloj que se manda por un pin del PIC que va hacia los pines de reloj de los convertidores.

Según el convertidor elegido, la señal de reloj irá de “high” a “low” para iniciar el envío del primer bit (el más significativo del byte). Cuando esto ocurra, el microcontrolador verificara si el bit a enviar es cero o uno y se mandará un “low” o un “high” al pin de salida del microcontrolador que esté conectado a la entrada DIN del convertidor dependiendo del resultado. Terminado esto, se avanza el reloj (high), se toma el bit anterior al más significativo y se repite el ciclo hasta que se llegue al bit menos significativo del dato a enviar.

4.3. Protocolo de comunicación entre el microcontrolador y el módulo USB

Para el envío y recepción de datos desde el microcontrolador PIC al puerto USB de la computadora, nos auxiliamos del módulo DLP-245USB. Los detalles de funcionamiento de dicho módulo fueron explicados en el capítulo de hardware. Lo que corresponde a este segmento es describir la sección del programa que se diseñó para el envío y recepción de datos por medio del módulo.

Esta parte del programa es la encargada de enviar la información necesaria para poder comunicarse con el software de la computadora mediante el módulo DLP-USB245. A su vez, este módulo, fue comunicado con el microcontrolador por medio de cuatro pines de control los cuales controlaron tanto el envío como la recepción de datos que se reciben o envían a través de 8 pines que contiene el módulo y que fueron conectados al puerto D del microcontrolador de forma paralela. Estos pines responden tanto a voltajes de salida como entrada altos y bajos. Según especificaciones del proveedor del módulo, se considera un voltaje de salida bajo o “low” el que va de 0.1 a 0.7 V de CD mientras que se considera un voltaje alto o “high” al que va desde 4.4 a 4.9 V de CD. En lo posterior, se hará referencia a estos voltajes como “low” o “high.

Los pines de control con los que cuenta el módulo son los siguientes:

PIN DE CONTROL		Función
RXF#	Salida	Cuando pasa de “high” a “low”, por lo menos un byte está presente en el buffer de 128 bytes FIFO’s y está listo para ser leído. Cuando está en alto, el buffer está vacío.
RD#	Entrada	Cuando recibe un “low”, RD# asume que los datos de las 8 líneas del módulo se

		pueden leer por el dispositivo con el que se comunica el DLP.
TXF#	Salida	Cuando está en “high”, el buffer está lleno u ocupado almacenando el último byte escrito. Cuando está en “low” indica que se pueden escribir datos en el buffer.
WR	Entrada	Cuando le es enviado un cambio de estado de “high” a “low”, WR lee las 8 líneas de datos y escribe en el buffer los datos para ser enviados.

Dentro de la función del programa involucrada a esta actividad, tendremos el ciclo en el cual podremos seguir el protocolo para enviar datos del DLP (provenientes a su vez del software de la computadora) hacia el microcontrolador. Estos datos son relacionados con instrucciones que el usuario debe de darle al programa con las preferencias que el mismo usuario decida. Para el control de esto, se usan los pines RXF# y RD# del módulo DLPUSB245. A continuación se muestra el protocolo utilizado provisto por el fabricante:

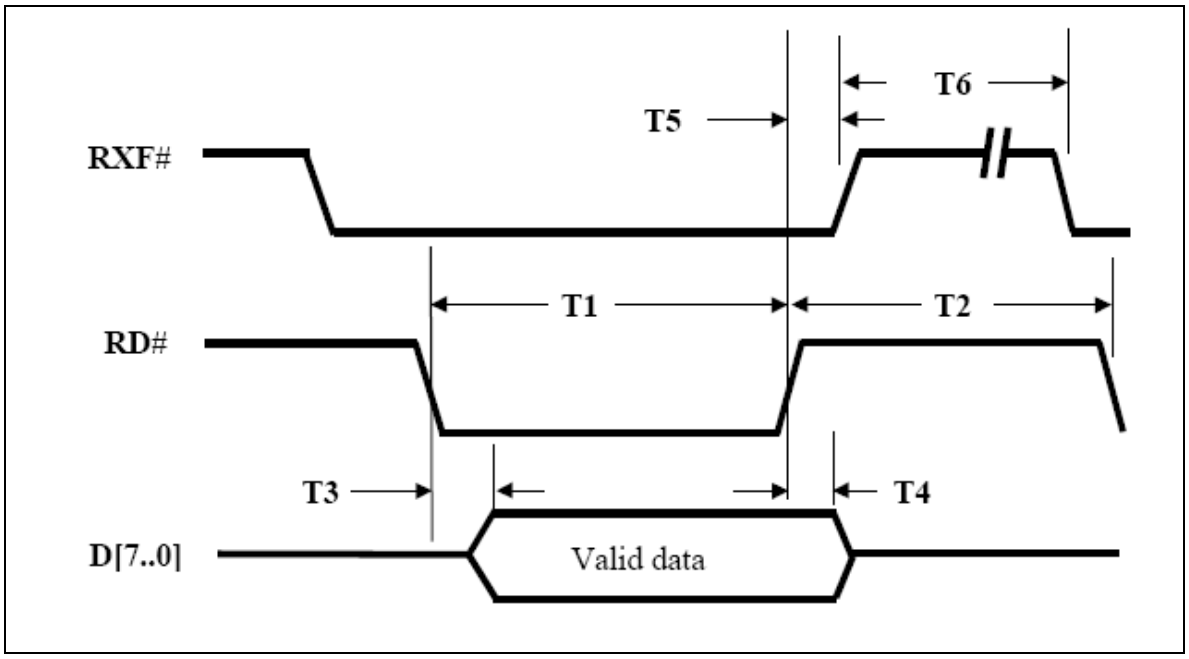


Figura 4.15: Protocolo para enviar datos del DLP al PIC

En este protocolo serán enviados desde el software, los siguientes datos:

- Número de canales a sensar,
- Canales sensados,
- Número de muestras y
- Tiempo entre muestras

En la interfaz, todos estos datos se enviaron a través de varios bytes, dentro de los cuales, se representaron los datos que el usuario le había asignado al software.

Para poder enviar esto, a partir del protocolo planteado, en el PIC se creó la rutina llamada DLP_ENVIA_AL_PIC que cumpliera los requisitos. En el siguiente párrafo, se describirá y mostrará el diagrama de flujo correspondiente.

En esta función, el puerto D del microcontrolador esta contemplado para ser una entrada en la que se puedan recibir datos que le sean enviados desde el DLP. Por medio de la función, se verificará el estado del pin de entrada del microcontrolador conectado al pin RXF del módulo. En caso de que dicho estado pase de “high” a “low”, el microcontrolador se encargará de mandar una señal de encendido hacia el DLP con lo que indicará que está listo y esperando para recibir un dato por el puerto D. En este momento, el DLP enviará el dato y cuando el PIC lo haya recibido satisfactoriamente y guardado en una localidad de memoria, el estado del pin del PIC conectado al pin RD del módulo cambiará a “low”. Con esto se finalizará el proceso.

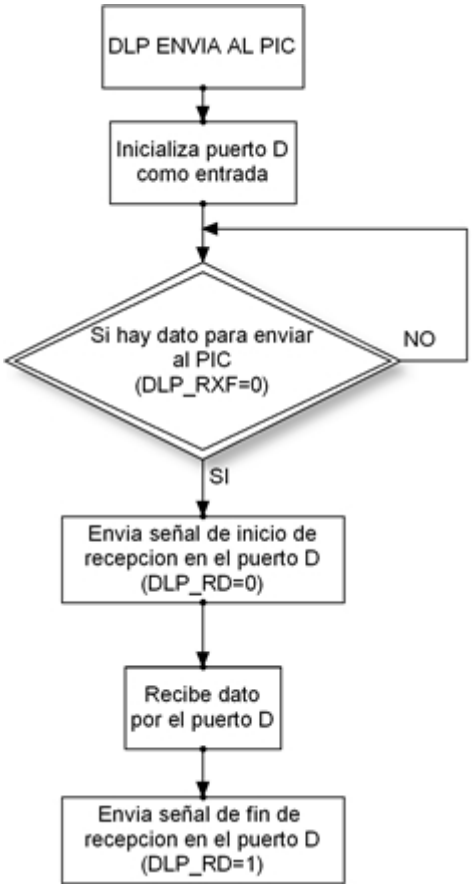


Figura 4.16: Función DLP ENVIA AL PIC

Además, se desarrolló una función llamada DLP_ENVIA_AL_PIC_ABORTAR cuya única diferencia con la función anterior es que si al ser llamada, no existe un estado "high" en RXF (no hay dato para enviar al PIC), en vez de esperar a que llegue dicho dato y almacenarlo, lo que hará será guardar un cero hexadecimal en la localidad donde sea recibido el dato. Cabe mencionar que el objetivo de enviar ceros cada vez que no existiesen datos válidos es debido a que fue un convenio realizado con el diseñador de software ya que le facilitaba la programación que el realizó, sin embargo para los fines del firmware no tiene ventajas visibles.

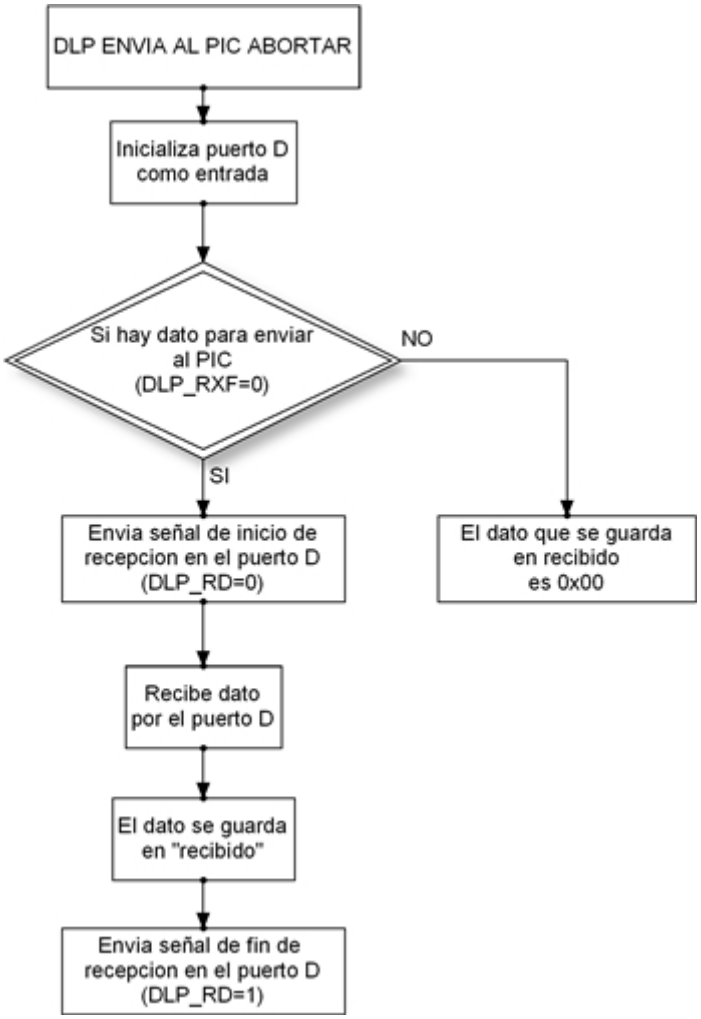


Figura 4.17: Función DLP ENVIA AL PIC ABORTAR

En el envío de información del PIC hacía el módulo DLP, nos encontramos en el caso de que los datos enviados serán los bytes que contendrán los datos digitales en un formato comprensible para el software de la computadora. Para este proceso, el protocolo proporcionado por el fabricante se muestra en el siguiente diagrama:

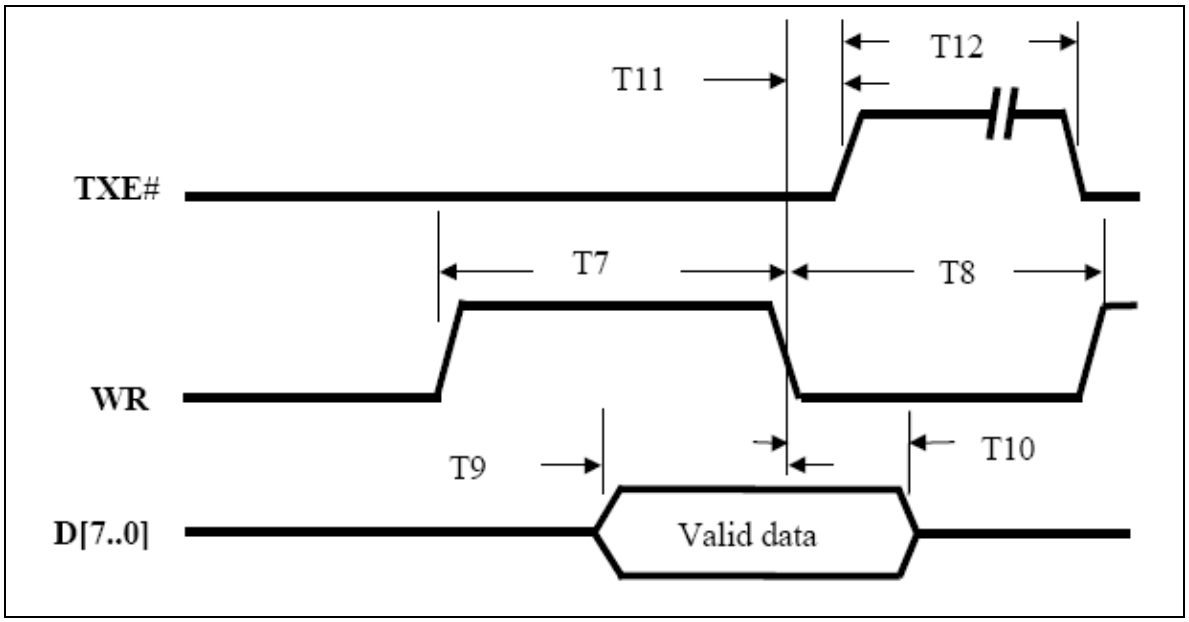


Figura 4.18: Envío de información del PIC al DLP

A continuación se describe la forma en la que se implementó la rutina para que el PIC pudiera enviar los datos de manera correcta.

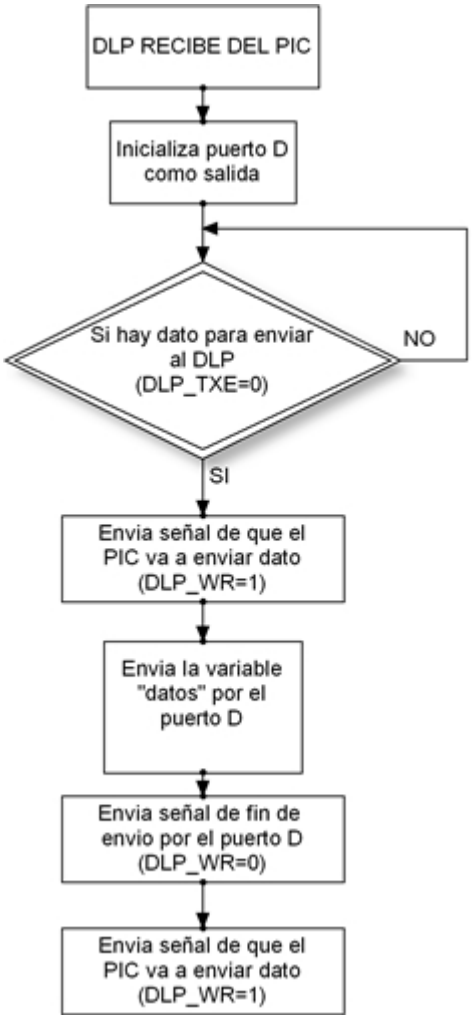


Figura 4.19: Función DLP RECIBE DEL PIC

En el caso de la función DLP_RECIBE_DEL_PIC, el puerto D del microcontrolador es configurado como salida paralela. Se verifica que el DLP esté disponible para enviarle un dato (el pin TXE del DLP conectado al microcontrolador debe de estar en “low”), de no ser así, el PIC espera hasta que se cumpla la condición, sin embargo, si el DLP está disponible en ese instante, el PIC mandará una señal para anunciar que el PIC va a enviar el dato al DLP (el pin del microcontrolador conectado al pin WR del DLP será puesto en high). Después de que el dato ha sido enviado, el PIC envía al puerto paralelo D el byte de dato que el programa solicitó enviar y al terminar envía un “low” hacia el pin WR del DLP.

Envío de datos a la computadora personal

Finalmente, cuando los datos han sido mandados al módulo DLP, estos, tendrán un formato preestablecido para que puedan ser interpretados por el software del sistema de adquisición de datos. Después de que el PIC procesa los datos, va a enviar el dato procesado en dos bytes. El primero va a ser la parte alta (H) y el segundo la parte baja (L). El formato que manejan es el siguiente:

La parte alta H = 0XDDDDDDD

La parte baja L = 0DDDDDDD

El bit de mayor significancia, tanto en la parte alta (H) como en la parte baja (L), va a tener forzosamente un cero debido a que fue convenido con el diseñador de software que se hizo en el laboratorio para indicar que esto era un dato válido. El siguiente bit en orden de significancia en el byte de la parte alta(H) va a ser visto como “don’t care” por el software. Los demás bits enviados, serán la información, considerando de mayor significancia el tercer bit en orden de significancia de la parte alta (H) y como bit de menor significancia el último bit de la parte baja (L). Finalmente, el software se encargará de juntar los datos como uno solo de 13 bits y realizará las acciones necesarias para poder desplegarlo en pantalla, así como hacerle los análisis que el usuario requiera.

Con lo anterior, se cumple con el objetivo de dar un panorama particular de cada una de las funciones que estuvieron involucradas en el desarrollo del firmware. La mayoría de estas funciones tienen una relación entre ellas. Las funciones cuyo objetivo es enviar y recibir datos funcionan como apoyo para las funciones encargadas de controlar los dispositivos utilizados (convertidores, módulo DLP y microcontrolador).

Es conveniente destacar que a pesar de que dichas funciones dependen unas de otras, cualquiera de ellas podría ser modificada fácilmente si fuera necesario crear una versión nueva del firmware. Esto debido a que cada función está claramente delimitada en su tarea, de tal forma que, si por algún motivo se necesitara cambiar una parte del hardware controlada por el PIC (supongamos cambiar el modelo del CAD), solo se tendrían que modificar las funciones

relacionadas con esa parte del hardware y las demás seguirían funcionando de manera correcta sin tener que efectuar el más mínimo cambio.

Al concluir esto, queda cumplido de manera satisfactoria uno de los objetivos principales que fueron solicitados en el planteamiento de la tesis. Con el nuevo sistema, al equipo se le puede dar, de manera sencilla, un mantenimiento periódico o incluso incluir nuevas características conforme el tiempo se lo llegue a exigir con el fin de que no quede relegado y conforme pase el tiempo siga siendo competitivo con respecto a sistemas similares ofrecidos en el mercado.

CONCLUSIONES

Se logró diseñar una interfaz de adquisición de datos que fue comunicada a una computadora por medio del protocolo USB. A pesar de que se encontró que existían diversas formas de construirla, al conjuntar las herramientas elegidas, todas coincidieron de tal forma que se logró el objetivo planteado, implementando así un dispositivo que cumplió con los parámetros establecidos en un principio.

Debido al continuo lanzamiento en el mercado de nuevos dispositivos electrónicos, nuestra interfaz siempre va a poder ser mejorada en muchas de sus características. En el momento de estar efectuando el diseño, esto fue considerado e incluso, se tomaron las medidas adecuadas para, en un futuro, poder hacer adecuaciones necesarias para seguir contando con una herramienta con características útiles y que sigan siendo competitivas con los dispositivos.

Entre las características más importantes que se lograron, tenemos que el equipo maneja una resolución de 13 bits (12 bits más uno de signo), a diferencia de los 8 bits que se manejaban en el sistema anterior. La velocidad que alcanza el equipo es de 1000 datos/seg, la cual es aumentada por poco más de 10 veces comparándola con el sistema anterior.

En cuanto al número de canales, se cuenta con 4 canales de entrada. Para futuras versiones, la interfaz cuenta con la infraestructura para poder funcionar con un total de hasta 16 canales.

A pesar de que se hizo un total rediseño de la interfaz, gracias al acoplamiento de dispositivos, se consiguieron rescatar las partes que seguían siendo útiles del sistema, como fueron los módulos y las fuentes de alimentación. Esto fue de ayuda para no tener que utilizar más recursos en dispositivos que seguían llevando a cabo su tarea de manera satisfactoria.

En cuanto al firmware de la interfaz, se cumplió con uno de los puntos medulares para poder realizar el mantenimiento del equipo. Las rutinas de programación se efectuaron en un lenguaje de alto nivel (lenguaje C), cuidando que fueran hechas de manera entendible a cualquier diseñador y procurando ser sencillas para poder ser modificadas en el futuro, ratificando así el empeño que se tuvo en lograr que el sistema fuera perfeccionado fácilmente en las siguientes versiones.

El costo aproximado para la implementación de esta interfaz es de 800 pesos, lo que podemos considerar adecuado para las funciones que logramos implementar.

BIBLIOGRAFÍA

- **Microchip PIC16F877A, Datasheet 2003**
- **National LMC660 CMOS Quad Operational Amplifier, Datasheet 2006**
- **Microchip MCP3304, Datasheet 2002**
- **DLP-USB245M User Manual, DLP Design 2002**
- **Manual de usuario del Compilador PCW de CSS**
  Canovas López, Antonio
- **AN232B-06 Debugging fT234BM/FT245BM Designs**
  Future Technology Devices International LTD 2004-2005
- **Programming and customizing PICmicro Microcontrollers**
  Myke Predko Mc Graw Hill 2° edition 2002
- **Manual de fabricación Interfaz LABSIS-TECNOS Versión 2.0**
  Bañuelos Saucedo, Miguel Ángel. México, 2004.
- **Manual de usuario Interfaz V4.A**
  Bañuelos Saucedo, Miguel Ángel. México, 2004.
- **Programación del microcontrolador PIC18F452**
  Bañuelos Saucedo, Miguel Ángel. México, 2004.
- **Conexión Serial con PIC's**
  Aguayo, Paul. Chile 2005
- **Design with PIC Microcontrollers**
  Peatman, John. Prentice Hall, 1998

SITIOS EN INTERNET

www.dlpdesign.com/usb

www.microchip.com

www.ccsinfo.com