

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACION DE TECNÓLOGOS

CONSTRUCCIÓN DE UNA TARJETA ELECTRÓNICA DE ADQUISICIÓN DE DATOS CON COMUNICACIÓN POR PUERTO USB AL COMPUTADOR, PARA EL LABORATORIO DE MICROPROCESADORES DE LA ESFOT

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES

INTEGRANTES:

LIZARDO ALEXIS LUCERO MOLINA

alexlucero82@hotmail.com

BYRON ALEXIS PILAPAÑA ANAGUANO

aleksoboy@hotmail.com

DIRECTOR: ING. ALCIVAR COSTALES

eduardo.costales@epn.edu.ec

Quito, Mayo 2010

DECLARACIÓN

Nosotros LIZARDO ALEXIS LUCERO MOLINA y BYRON ALEXIS PILAPAÑA ANAGUANO, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Sr. Alexis Lucero

Sr. Alexis Pilapaña

CERTIFICACION

Certifico que el presente trabajo fue desarrollado por LIZARDO ALEXIS LUCERO MOLINA y BYRON ALEXIS PILAPAÑA ANAGUANO, bajo mi supervisión.

Ing. Alcivar Costales

DIRECTOR DE PROYECTO

INDICE

| | |
|---|-----------|
| CAPITULO 1 | 9 |
| 1 MARCO TEORICO | 9 |
| 1.1 ADQUISICIÓN DE DATOS | 9 |
| 1.1.1 CONCEPTO..... | 9 |
| 1.1.2 PROCESO DE ADQUISICION DE DATOS..... | 10 |
| 1.1.2.1 Definiciones..... | 10 |
| 1.1.2.2 Descripción de la adquisición de datos..... | 11 |
| 1.1.3 TIEMPO DE CONVERSION..... | 14 |
| 1.1.4 ACONDICIONAMIENTO DE LA SEÑAL..... | 14 |
| 1.1.5 EJEMPLOS DE SISTEMAS DE ADQUISICION Y CONTROL..... | 16 |
| 1.2 MICROCONTROLADORES | 17 |
| 1.2.1 ANTECEDENTES..... | 17 |
| 1.2.2 INTRODUCCION..... | 18 |
| 1.2.3 CARACTERISTICAS DE LOS MICROCONTROLADORES..... | 18 |
| 1.2.4 PROCESO DE DESARROLLO DE APLICACIONES BASADAS EN MICROCONTROLADORES..... | 20 |
| 1.2.4.1 Desarrollo del software..... | 20 |
| 1.2.4.2 Programación del microcontrolador..... | 22 |
| 1.2.4.3 Prueba y verificación..... | 23 |
| 1.2.5 DIFERENCIA ENTRE MICROPROCESADOR Y MICROCONTROLADOR..... | 23 |
| 1.2.5.1 Arquitectura Von Neumann..... | 23 |
| 1.2.5.2 Arquitectura Harvard..... | 24 |
| 1.2.6 PARTES DEL MICROCONTROLADOR..... | 25 |
| 1.2.6.1 Procesador..... | 26 |
| 1.2.6.2 Memoria de programa..... | 26 |
| 1.2.6.3 Memoria de datos..... | 27 |
| 1.2.6.4 Puertos E/S..... | 27 |
| 1.2.6.5 Recursos Auxiliares..... | 27 |
| 1.2.7 MICROCONTROLADORES PICS..... | 30 |
| 1.2.7.1 Introducción..... | 30 |
| 1.2.7.2 Ventajas de los PICs..... | 31 |
| 1.3 SENSORES | 31 |
| 1.3.1 CONCEPTO..... | 31 |
| 1.3.2 CARACTERISTICAS GENERALES DE LOS SENSORES..... | 33 |

| | | |
|------------------------|---|-----------|
| 1.3.2.1 | Características estáticas | 33 |
| 1.3.2.2 | Características dinámicas | 35 |
| 1.3.3 | TIPOS DE SENSORES | 35 |
| 1.3.4 | AREAS DE APLICACIÓN DE LOS SENSORES | 37 |
| 1.4 | COMUNICACIÓN USB..... | 37 |
| 1.4.1 | INTRODUCCION A USB..... | 37 |
| 1.4.2 | CARACTERISTICAS GENERALES DE USB..... | 38 |
| 1.4.3 | LIMITACIONES | 41 |
| 1.4.4 | CABLES Y CONECTORES USB | 41 |
| 1.5 | MAGNITUDES FISICAS MEDIDAS..... | 42 |
| 1.5.1 | PRESION | 42 |
| 1.5.1.1 | Definición | 42 |
| 1.5.1.2 | Presión absoluta y relativa | 42 |
| 1.5.1.3 | Unidades de medida y sus factores de conversión | 43 |
| 1.5.2 | TEMPERATURA..... | 43 |
| 1.5.2.1 | Definición | 43 |
| 1.5.2.2 | Unidades de temperatura | 43 |
| 1.5.2.3 | Conversión de unidades de temperatura | 44 |
| 1.5.3 | HUMEDAD..... | 45 |
| 1.5.3.1 | Definición | 45 |
| 1.5.3.2 | Humedad absoluta | 45 |
| 1.5.3.3 | Humedad relativa..... | 45 |
| CAPITULO 2..... | | 47 |
| 2 | DESARROLLO DEL HARWARE..... | 47 |
| 2.1 | CONSTRUCCIÓN DEL HARDWARE..... | 47 |
| 2.2 | REQUERIMIENTOS DEL HARDWARE | 48 |
| 2.3 | COMPONENTES DEL HARDWARE | 49 |
| 2.3.2 | SENSORES | 50 |
| 2.3.2.1 | Sensores de presión..... | 50 |
| 2.3.2.2 | Sensores de temperatura | 52 |
| 2.3.2.3 | Sensores de humedad | 53 |
| 2.3.3 | CIRCUITOS INTEGRADOS | 54 |
| 2.3.3.1 | Amplificador operacional AD620AN..... | 54 |
| 2.3.3.2 | Amplificador operacional LM358 | 55 |

| | | |
|--|--|-----------|
| 2.3.4 | MODULO GLCD (Graphic Liquid Cristal Display) | 56 |
| 2.3.4.1 | Distribución de pines | 56 |
| 2.3.5 | MICROCONTROLADOR PIC 18F4550..... | 57 |
| 2.3.5.1 | Características..... | 58 |
| 2.3.5.2 | Distribución de pines | 59 |
| 2.4 | ENSAMBLAJE DEL HARDWARE | 59 |
| 2.4.1 | DESARROLLO DEL CIRCUITO PROCESADOR DE SEÑAL..... | 60 |
| 2.4.2 | DESARROLLO DE LOS CIRCUITOS ADAPTADORES DE SEÑAL | 63 |
| 2.4.2.1 | Circuito adaptador del sensor de presión..... | 63 |
| 2.4.2.2 | Circuito adaptador del sensor de temperatura | 64 |
| 2.4.2.3 | Circuito adaptador del sensor de humedad | 65 |
| 2.5 | DISEÑO DE LOS CIRCUITOS IMPRESOS DEL SISTEMA | 67 |
| 2.5.1 | CIRCUITO IMPRESO DEL PROCESADOR DE SEÑAL | 67 |
| 2.5.2 | CIRCUITOS IMPRESOS DE LOS ADPTADORES DE SEÑAL..... | 70 |
| 2.5.2.1 | Circuito impreso del adaptador de señal del sensor de presión..... | 70 |
| 2.5.2.2 | Circuito impreso del adaptador de señal del sensor de temperatura..... | 71 |
| 2.5.2.3 | Circuito impreso del adaptador de señal del sensor de humedad | 72 |
| 2.6 | PRINCIPIO DE FUNCIONAMIENTO | 73 |
| CAPITULO 3 | | 76 |
| 3 DESARROLLO DEL SOFTWARE | | 76 |
| 3.1 | INTRODUCCIÓN..... | 76 |
| 3.2 | DESCRIPCION DE LAS HERRAMIENTAS COMPUTACIONALES | 77 |
| 3.2.1 | PROTON IDE..... | 78 |
| 3.2.2 | VISUAL BASIC | 79 |
| 3.2.3 | WINPIC..... | 80 |
| 3.3 | DESARROLLO DEL PROGRAMA PARA EL MICROCONTROLADOR PIC 18F4550..... | 81 |
| 3.3.1 | COMUNICACIÓN USB PIC – COMPUTADOR | 81 |
| 3.3.2 | DEFINICION DE PUERTOS..... | 86 |
| 3.3.3 | DEFINICIONES PARA EL GLCD | 89 |
| 3.3.4 | DEFINICIONES DE ALIAS | 90 |
| 3.3.5 | DEFINICIONES DEL CONVERTOR A/D | 91 |
| 3.3.6 | PRESENTACION DE IMÁGENES EN EL GLCD | 91 |

| | | |
|---------------------|--|------------|
| 3.3.7 | PRESENTACIÓN DE DATOS EN EL GLCD..... | 94 |
| 3.3.7.1 | Diseño del programa para el sensor de temperatura..... | 95 |
| 3.3.7.2 | Diseño del programa para el sensor de humedad..... | 96 |
| 3.3.7.3 | Desarrollo del programa para medición de presión..... | 97 |
| 3.4 | DESARROLLO DEL SOFTWARE PARA LA INTERFAZ GRÁFICA..... | 98 |
| 3.4.1 | PROGRAMA PARA LA INTERFAZ GRÁFICA..... | 98 |
| 3.4.1.1 | Código para comunicación Usb Pic - Pc..... | 98 |
| 3.4.1.2 | Código para lectura de datos desde el Pic..... | 99 |
| 3.4.2 | DISEÑO DE VISUALIZACIÓN DE LA INTERFAZ..... | 99 |
| 3.4.3 | GENERACIÓN DE ARCHIVO EJECUTABLE..... | 100 |
| 3.4.4 | PRESENTACION FINAL DE LA INTERFAZ GRÁFICA..... | 102 |
| 3.4.5 | DESCRIPCION BREVE DE LAS FUNCIONES DE LA INTERFAZ GRAFICA..... | 102 |
| CAPITULO 4 | | 103 |
| 4 | PRUEBAS Y RESULTADOS..... | 103 |
| 4.1 | PRUEBAS CON EL SENSOR DE PRESION..... | 103 |
| 4.2 | PRUEBAS CON EL SENSOR DE TEMPERATURA..... | 104 |
| 4.3 | PRUEBAS CON EL SENSOR DE HUMEDAD..... | 106 |
| 4.4 | PRUEBAS CON EL POTENCIOMETRO..... | 107 |
| 4.5 | COSTO DEL PROYECTO..... | 108 |
| CAPITULO 5 | | 111 |
| 5 | CONCLUSIONES Y RECOMENDACIONES..... | 111 |
| 5.1 | CONCLUSIONES:..... | 111 |
| 5.2 | RECOMENDACIONES..... | 113 |
| BIBLIOGRAFIA | | 115 |

ANEXOS

ANEXO A: PROGRAMAS DEL SISTEMA

ANEXO B: DATASHEETS

ANEXO C: MANUAL DE USUARIO

RESUMEN

En la actualidad el vertiginoso desarrollo de la electrónica y la microelectrónica han motivado que todas las esferas de la vida humana se estén automatizando, por ejemplo: la industria, el hogar, los comercios, la agricultura, la ganadería, el transporte, las comunicaciones, etc. En todo ese proceso de automatización el microprocesador y el microcontrolador juegan un papel de suma importancia. Ellos han permitido el desarrollo de sistemas inteligentes que resuelven los más diversos problemas, son los llamados Sistemas de Adquisición de Datos.

Ante estas exigencias, hemos implementado un sistema de adquisición de datos, que se compone de sensores de parámetros físicos, circuitos de acoplamiento de señal y circuito procesador de señales, el cual dispone de un elemento visualizador de resultados como es el GLCD, o también se pueden visualizar los datos en un computador con la ayuda de una interfaz grafica, y gracias a que este circuito dispone de un puerto para comunicación USB.

Los sensores son básicamente elementos que transforman un tipo de energía en otra, es así que cada uno de ellos dispone de su único circuito de acoplamiento de señal, el cual puede estar conformado por un amplificador operacional, resistencias, condensadores, etc., dependiendo de las características de cada sensor.

El circuito procesador de señal consta básicamente de un microcontrolador PIC 18F4550, el cual tomara las señales provenientes de cada circuito de acoplamiento y las procesara según el programa que tenga guardado en su memoria, enviara los datos hacia el GLCD para poder visualizarlos y también se encarga de establecer la comunicación USB con el computador.

En resumen de lo descrito se puede decir que, en conjunto los sensores, los circuitos de acoplamiento, el circuito procesador y la computadora forman parte de un sistema de adquisición de datos que replazan de una manera moderna y tecnológica a

métodos de medición individuales para cada parámetro físico, ya que en un solo sistema se puede chequear varias variables físicas.

El objetivo básico de este sistema es la integración de los diferentes recursos que lo integran: Transductores de diferentes tipos y naturaleza, amplificadores, conversores A/D y D/A, microcontrolador PIC 18F4550, utilizando de éste todas sus prestaciones tales como: interrupciones, temporizadores, comunicación USB, etc., que se encargue de una aplicación específica como es chequear variables, para una posterior utilización del mismo ya sea con fines docentes, científicos, de almacenamiento o control y utilización de la misma.

CAPITULO 1

1 MARCO TEORICO

1.1 ADQUISICIÓN DE DATOS

1.1.1 CONCEPTO

La adquisición de datos o adquisición de señales, consiste en la toma de muestras del mundo real (sistema analógico) para generar datos que puedan ser manipulados por un ordenador u otros sistemas electrónicos (sistema digital). Consiste, en tomar un conjunto de señales físicas, convertirlas en tensiones eléctricas y digitalizarlas manera que se puedan procesar en una computadora o un PAC¹. Se requiere una etapa de acondicionamiento, que adecua la señal a niveles compatibles con el elemento que hace la transformación a señal digital. El elemento que hace dicha transformación es el módulo de digitalización o tarjeta de Adquisición de Datos.

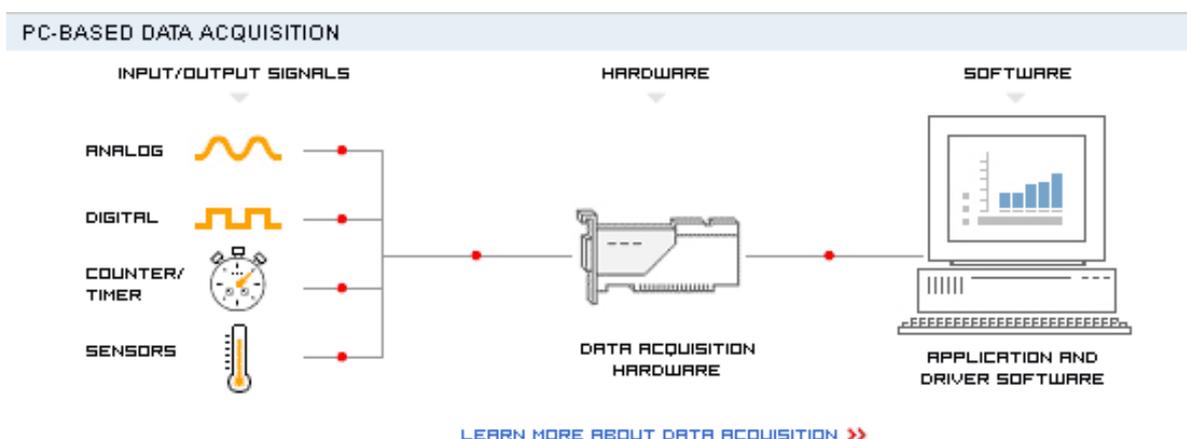


Figura 1.1: Esquema de adquisición de datos.²

El propósito de adquisición de datos es medir un fenómeno eléctrico y físico como voltaje, corriente, temperatura, presión o sonido. La adquisición de datos basada en

¹ PAC: Del inglés Programmable Automation Controller.

² <http://www.ni.com/dataacquisition/esa/whatis.htm>.

PC³ utiliza una combinación de hardware modular, software de aplicación y una PC para realizar medidas. Mientras cada sistema de adquisición de datos se define por sus requerimientos de aplicación, cada sistema comparte una meta en común de adquirir, analizar y presentar información. Los sistemas de adquisición de datos incorporan señales, sensores, actuadores, acondicionamiento de señales, dispositivos de adquisición de datos y software de aplicación.

Un sistema típico de adquisición utiliza sensores, transductores, amplificadores, convertidores analógico-digitales (A/D) y digitales-analógicos (D/A), para procesar información acerca de un sistema físico de forma digitalizada.

1.1.2 PROCESO DE ADQUISICION DE DATOS

1.1.2.1 Definiciones

Dato: Representación simbólica (numérica, alfabética...), atributo o característica de un valor. No tiene sentido en sí mismo, pero convenientemente tratado (procesado) se puede utilizar en la relación de cálculos o toma de decisiones.

Adquisición: Recogida de un conjunto de variables físicas, conversión en voltaje y digitalización de manera que se puedan procesar en un ordenador.

Sistema: Conjunto organizado de dispositivos que interactúan entre sí ofreciendo prestaciones más completas y de más alto nivel. Una vez que las señales eléctricas se transformaron en digitales, se envían a través del bus de datos a la memoria del PC. Una vez los datos están en memoria pueden procesarse con una aplicación adecuada, archivarlas en el disco duro, visualizarlas en la pantalla, etc.

Bit de resolución: Número de bits que el convertidor analógico a digital utiliza para representar una señal.

³ PC: Del inglés Personal Computer.

Rango: Valores máximo y mínimo entre los que el sensor, instrumento o dispositivo funcionan bajo unas especificaciones.

Teorema de Nyquist: Al muestrear una señal, la frecuencia de muestreo debe ser mayor que dos veces el ancho de banda de la señal de entrada, para poder reconstruir la señal original de forma exacta a partir de sus muestras. En caso contrario, aparecerá el fenómeno del aliasing que se produce al infra-muestrear. Si la señal sufre aliasing, es imposible recuperar el original. La velocidad de muestreo recomendada debe ser:

- 2 x frecuencia mayor (medida de frecuencia)
- 10 x frecuencia mayor (detalle de la forma de onda)

Sensor: es un dispositivo que convierte una propiedad física o fenómeno en una señal eléctrica correspondiente medible, tal como tensión, corriente, el cambio en los valores de resistencia o condensador, etc.

Transductor: es sinónimo de sensor en sistemas de DAQ⁴.

1.1.2.2 Descripción de la adquisición de datos

La adquisición de datos se inicia con el fenómeno físico o la propiedad física de un objeto (objeto de la investigación) que se desea medir. Un eficaz sistema de adquisición de datos puede medir todas estas diferentes propiedades o fenómenos.

La capacidad de un sistema de adquisición de datos para medir los distintos fenómenos depende de los transductores para convertir las señales de los fenómenos físicos mensurables en la adquisición de datos por hardware.

Hay transductores específicos para diferentes aplicaciones, como la medición de la temperatura, la presión, o flujo de fluidos. DAQ también despliega diversas técnicas

⁴ DAQ: Del inglés Data Acquisition

de acondicionamiento de Señales para modificar adecuadamente diferentes señales eléctricas en tensión, que luego pueden ser digitalizados.

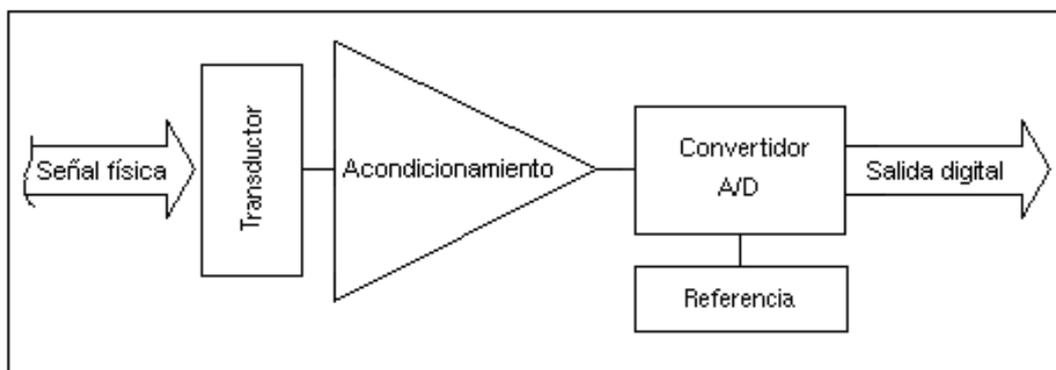


Figura 1.2: Esquema en bloques de un sistema DAQ.

Los componentes de los sistemas de adquisición de datos, poseen sensores adecuados que convierten cualquier parámetro de medición de una señal eléctrica, que se adquiere por el hardware de adquisición de datos. Los datos adquiridos se visualizan, analizan, y almacenan en un ordenador, ya sea utilizando el proveedor de software suministrado u otro software.

Los controles y visualizaciones se pueden desarrollar utilizando varios lenguajes de programación de propósito general como Visual BASIC, C++, Fortran, Java, Lisp, Pascal.

De la misma manera que se toma una señal eléctrica y se transforma en una digital para enviarla al ordenador, se puede también tomar una señal digital o binaria y convertirla en una eléctrica. En este caso el elemento que hace la transformación es una tarjeta o módulo de Adquisición de Datos de salida, o tarjeta de control. La señal dentro de la memoria del PC la genera un programa adecuado a las aplicaciones que quiere el usuario y, luego de procesarla, es recibida por mecanismos que ejecutan movimientos mecánicos, a través de servomecanismos, que también son del tipo transductores.

Las señales pueden ser digitales (también llamada señales de la lógica) o analógicas en función del transductor utilizado.

El acondicionamiento de señales suele ser necesario si la señal desde el transductor no es adecuado para el DAQ hardware que se utiliza. La señal puede ser amplificada o desamplificada, o puede requerir de filtrado, o un cierre patronal, en el amplificador se incluye para realizar demodulación. Varios otros ejemplos de acondicionamiento de señales podría ser el puente de conclusión, la prestación actual de tensión o excitación al sensor, el aislamiento, linealización, etc. Este pretratamiento de la señal normalmente lo realiza un pequeño módulo acoplado al transductor.

DAQ hardware: son por lo general las interfaces entre la señal y un PC. Podría ser en forma de módulos que pueden ser conectados a la computadora de los puertos (paralelo, serie, USB, etc.) o ranuras de las tarjetas conectadas a (PCI, ISA) en la placa madre. Por lo general, el espacio en la parte posterior de una tarjeta PCI es demasiado pequeño para todas las conexiones necesarias, de modo que una ruptura de caja externa es obligatoria. El cable entre este recuadro y el PC es cara debido a los numerosos cables y el blindaje necesario y porque es exótico. Las tarjetas DAQ a menudo contienen múltiples componentes (multiplexores, ADC, DAC, TTL-IO, temporizadores de alta velocidad, memoria RAM). Estos son accesibles a través de un bus por un micro controlador, que puede ejecutar pequeños programas. El controlador es más flexible que una unidad lógica dura cableada, pero más barato que una CPU de modo que es correcto para bloquear con simples bucles de preguntas.

Driver software: normalmente viene con el hardware DAQ o de otros proveedores, y permite que el sistema operativo pueda reconocer el hardware DAQ y dar así a los programas acceso a las señales de lectura por el hardware DAQ. Un buen conductor ofrece un alto y bajo nivel de acceso.

1.1.3 TIEMPO DE CONVERSION

Es el tiempo que tarda en realizar una medida el convertidor en concreto, y dependerá de la tecnología de medida empleada. Evidentemente nos da una cota máxima de la frecuencia de la señal a medir.

Este tiempo se mide como el transcurrido desde que el convertidor recibe una señal de inicio de "conversión", hasta que en la salida aparece un dato válido. Para que tengamos constancia de un dato válido tenemos dos caminos:

- Esperar el tiempo de conversión máximo que aparece en la hoja de características.
- Esperar a que el convertidor nos envíe una señal de fin de conversión.

Si no respetamos el tiempo de conversión, en la salida tendremos un valor, que dependiendo de la constitución del convertidor será:

- Un valor aleatorio, como consecuencia de la conversión en curso.
- El resultado de la última conversión.

1.1.4 ACONDICIONAMIENTO DE LA SEÑAL

Con más detalle, en una etapa de acondicionamiento podemos encontrar estas etapas, aunque no todas están siempre presentes:

- Amplificación
- Excitación
- Filtrado
- Multiplexado
- Aislamiento
- Linealización

Amplificación: Es el tipo más común de acondicionamiento. Para conseguir la mayor precisión posible la señal de entrada deber ser amplificada de modo que su máximo nivel coincida con la máxima tensión que el convertidor pueda leer.

Aislamiento: Otra aplicación habitual en el acondicionamiento de la señal es el aislamiento eléctrico entre el transductor y el ordenador, para proteger al mismo de transitorios de alta tensión que puedan dañarlo. Un motivo adicional para usar aislamiento es el garantizar que las lecturas del convertidor no son afectadas por diferencias en el potencial de masa o por tensiones en modo común.

Cuando el sistema de adquisición y la señal a medir están ambas referidas a masa pueden aparecer problemas si hay una diferencia de potencial entre ambas masas, apareciendo un "bucle de masa", que puede devolver resultados erróneos.

Multiplexado: El multiplexado es la conmutación de las entradas del convertidor, de modo que con un sólo convertidor podemos medir los datos de diferentes canales de entrada. Puesto que el mismo convertidor está midiendo diferentes canales, su frecuencia máxima de conversión será la original dividida por el número de canales muestreados. Se aconseja que los multiplexores se utilicen antes del conversor y después del acondicionamiento de la señal, ya que de esta manera no molestará a los aislantes que podamos tener.

Filtrado: El fin del filtro es eliminar las señales no deseadas de la señal que estamos observando. Por ejemplo, en las señales cuasi-continuas, (como la temperatura) se usa un filtro de ruido de unos 4 Hz, que eliminará interferencias, incluidos los 50/60 Hz de la red eléctrica.

Las señales alternas, tales como la vibración, necesitan un tipo distinto de filtro, conocido como filtro antialiasing, que es un filtro pasabajo pero con un corte muy brusco, que elimina totalmente las señales de mayor frecuencia que la máxima a medir, ya que se si no se eliminasen aparecerían superpuestas a la señal medida, con el consiguiente error.

Excitación: La etapa de acondicionamiento de señal a veces genera excitación para algunos transductores, como por ejemplos las galgas extensiométricas, termistores o RTD, que necesitan de la misma, bien por su constitución interna, (como el termistor, que es una resistencia variable con la temperatura) o bien por la configuración en que se conectan (como el caso de las galgas, que se suelen montar en un puente de Wheatstone).

Linealización: Muchos transductores, como los termopares, presentan una respuesta no lineal ante cambios lineales en los parámetros que están siendo medidos. Aunque la linealización puede realizarse mediante métodos numéricos en el sistema de adquisición de datos, suele ser una buena idea el hacer esta corrección mediante circuitería externa.

1.1.5 EJEMPLOS DE SISTEMAS DE ADQUISICION Y CONTROL

- DAQ para recoger datos (datalogger) medioambientales (energías renovables e ingeniería verde).
- DAQ para audio y vibraciones (mantenimiento, test).
- DAQ + control de movimiento (corte con láser).
- DAQ + control de movimiento+ visión artificial (robots modernos).

1.2 MICROCONTROLADORES

1.2.1 ANTECEDENTES

Inicialmente cuando no existían los microprocesadores las personas se ingeniaban en diseñar sus circuitos electrónicos y los resultados estaban expresados en diseños que implicaban muchos componentes electrónicos y cálculos matemáticos. Un circuito lógico básico requería de muchos elementos circuitos con muchos ajustes y fallos; pero en el año 1971 apareció el primer microprocesador el cual originó un cambio decisivo en las técnicas de diseño de la mayoría de los equipos. Con la aparición de este circuito integrado todo sería mucho más fácil de entender y los diseños electrónicos serían mucho más pequeños y simplificados. Entre los microprocesadores más conocidos tenemos el popular Z-80 y el 8085. Los diseñadores de equipos electrónicos ahora tenían equipos que podían realizar mayor cantidad de tareas en menos tiempo y su tamaño se redujo considerablemente; sin embargo, después de cierto tiempo aparece una nueva tecnología llamada microcontrolador que simplifica aun más el diseño electrónico.

Los microcontroladores surgen desde dos vías de desarrollo paralelas; una desde Intel y otra desde Texas Instruments. Los primeros microcontroladores son el 4004 y 4040 de Intel que dieron lugar al 8048, a su vez predecesor del 8051. Aún así el primer microcontrolador fue el TMS1000 de Texas Instruments. Éste integraba un reloj, procesador, ROM, RAM, y soportes de E/S en un solo chip. Los microcontroladores están conquistando el mundo. Están presentes en nuestro trabajo, en nuestra casa, en general. Se pueden encontrar controlando el funcionamiento de los ratones y teclados de los computadores, en los teléfonos, en los hornos microondas y los televisores de nuestro hogar. Pero el siglo XXI será testigo de la conquista masiva de estos diminutos computadores, que gobernarán la mayor parte de los aparatos que fabricaremos y usamos los humanos.⁵

⁵ <http://usuarios.lycos.es/sfriswolker/pic/uno.htm>

1.2.2 INTRODUCCION

Los microcontroladores son computadores digitales integrados en un chip que cuentan con un microprocesador o unidad de procesamiento central (CPU), una memoria para almacenar el programa, una memoria para almacenar datos y puertos de entrada salida. A diferencia de los microprocesadores de propósito general, como los que se usan en los computadores PC, los microcontroladores son unidades autosuficientes y más económicas.

El funcionamiento de los microcontroladores está determinado por el programa almacenado en su memoria. Este puede escribirse en distintos lenguajes de programación. Además, la mayoría de los microcontroladores actuales pueden reprogramarse repetidas veces.

Por las características mencionadas y su alta flexibilidad, los microcontroladores son ampliamente utilizados como el cerebro de una gran variedad de sistemas embebidos que controlan máquinas, componentes de sistemas complejos, como aplicaciones industriales de automatización y robótica, domótica, equipos médicos, sistemas aeroespaciales, e incluso dispositivos de la vida diaria como automóviles, hornos de microondas, teléfonos y televisores.

Frecuentemente se emplea la notación μC o las siglas MCU (por *microcontroller unit*) para referirse a los microcontroladores

1.2.3 CARACTERISTICAS DE LOS MICROCONTROLADORES

Las principales características de los μC son:

- Unidad de Procesamiento Central (CPU): Típicamente de 8 bits, pero también hay de 4, 32 y hasta 64 bits.

- Memoria de Programa: Es una memoria ROM, EPROM, EEPROM, o Flash que almacena el código del programa que típicamente puede ser de 1 kilobytes a varios megabytes según el microcontrolador.
- Memoria de Datos: Es una memoria RAM, que típicamente puede ser de 1, 2, 4, 8, 16, 32 kilobytes.
- Generador del Reloj: Usualmente un cristal de cuarzo de frecuencias que genera una señal oscilatoria de entre 1 a 40 MHz, o también resonadores o circuitos RC.
- Interfaz de Entrada/Salida: Puertos paralelos, seriales (UARTs, *Universal Asynchronous Receiver / Transmitter*), I2C (*Inter-Integrated Circuit*), Interfaces de Periféricos Seriales (SPIs, *Serial Peripheral Interfaces*), Red de Area de Controladores (CAN, *Controller Area Network*), USB (*Universal Serial Bus*).
- Otras opciones:
 - Conversores Análogo-Digitales (A/D, *analog-to-digital*) para convertir un nivel de voltaje en un cierto pin a un valor digital manipulable por el programa del microcontrolador.
 - Moduladores por Ancho de Pulso (PWM, *Pulse-Width Modulation*) para generar ondas cuadradas de frecuencia fija pero con ancho de pulso modificable.

La alta integración de subsistemas que componen un μC reduce el número de chips, la cantidad de pistas y espacio que se requeriría en un circuito impreso si se implementase un sistema equivalente usando chips separados.

Un aspecto fundamental de los microcontroladores es los puertos de entrada/salida. A través de los pines del chip asociados a las interfaces de entrada/salida el μC puede interactuar con otros circuitos externos enviándoles señales de comando o recibiendo estímulos correspondientes a variables externas. Por lo general varios pines de datos son bidireccionales, es decir pueden configurarse como entradas o salidas.

Cuando son entradas, pueden adquirir datos interpretando el valor de voltaje como un valor lógico 0 o 1, mientras que cuando son salidas pueden entregar una señal binaria de voltaje cuya magnitud dependerá del valor lógico 0 o 1. Monitoreando el valor de las entradas, el microcontrolador puede responder a eventos externos y

realizar una cierta acción, como variar las señales de salida de acuerdo al valor en las entradas. Para responder a eventos externos, los μ Cs cuentan con un recurso conocido como *interrupciones*. Las interrupciones son señales que se generan internamente en el microcontrolador que detienen la ejecución normal del programa para ejecutar alguna subrutina de respuesta al evento. Una vez ejecutada la subrutina de interrupción la ejecución del programa continúa en el punto en que se encontraba antes de generarse la interrupción. Un ejemplo típico es el de un botón pulsador conectado a un pin de entrada. Una vez pulsado, se genera una señal de interrupción que iniciará la ejecución de la subrutina de interrupción, que por ejemplo podría activar un pin de salida para encender un Led.

No todas las interrupciones necesariamente están asociadas al cambio del estado de los pines de entrada. También hay interrupciones que pueden estar asociadas al valor de una entrada AD, o al cumplimiento de un periodo de tiempo fijado por un *timer* o temporizador. Estas características dependerán del modelo de μ C empleado.

1.2.4 PROCESO DE DESARROLLO DE APLICACIONES BASADAS EN MICROCONTROLADORES

El proceso de desarrollo de una aplicación basada en microcontroladores se compone de las siguientes etapas principales:

- DESARROLLO DEL SOFTWARE
- PROGRAMACION DEL MICROCONTROLADOR
- PRUEBA Y VERIFICACION

1.2.4.1 Desarrollo del software

Esta etapa consiste en escribir y compilar/ensamblar el programa que determinará las acciones del μ C y su funcionamiento. Existen distintas maneras de desarrollar el programa, dependiendo del lenguaje inicial que se utiliza para escribir el programa. En la figura 1.3 se indica el proceso general para desarrollar y generar el código de máquina que es entendido por el microcontrolador.

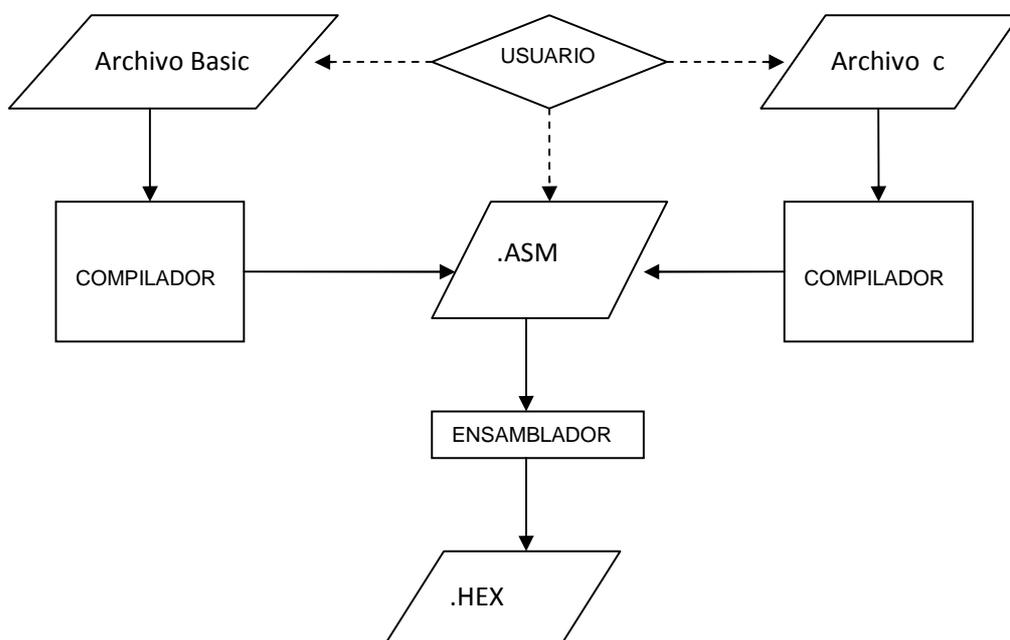


Figura 1.3: Proceso de desarrollo del código de máquina que es entendido por el microcontrolador

El método básico es escribir el programa en lenguaje Assembler en un archivo de texto con extensión `.asm` y luego utilizar un programa ensamblador para generar un archivo en lenguaje de máquina, también denominado código de máquina o código objeto, compuesto por instrucciones en código binario que son directamente entendidas por la CPU del microcontrolador. El ensamblador normalmente genera un archivo con extensión `.hex` (por hexadecimal), `.o` (por objeto), `.bin` (por binario), o `.coff` (common object file format) dependiendo del ensamblador. El lenguaje assembler se compone de instrucciones mnemónicas de bajo nivel, es decir que están ligadas a las características del microcontrolador y con un número mínimo o nulo de abstracciones. Al carecer de abstracciones, el lenguaje Assembler es más difícil de emplear, requiere experiencia y un mayor tiempo de desarrollo. La ventaja es que el código de máquina generado a partir de un programa escrito en lenguaje

de máquina es por lo general más eficiente, ya que el programa se desarrolla en un nivel cercano a las características del hardware.

Otra alternativa es emplear un lenguaje de alto nivel con una mayor cantidad de abstracciones, las cuales son más fáciles de usar y reducen los tiempos de desarrollo. Tal vez los lenguajes de alto nivel más comunes para la programación de controladores es el C y C++, pero también existen otros lenguajes variantes del BASIC y el Pascal. Una vez escrito el programa en el lenguaje de alto nivel, será necesario emplear un compilador para traducirlo, ya sea a lenguaje de Ensamblador o directamente a lenguaje de máquina. Es importante considerar que el código de Ensamblador generado por los compiladores tiende a ser más largo e ineficiente que aquel directamente desarrollado en lenguaje de Ensamblador. Esta desventaja puede ser crítica en ciertas aplicaciones que requieren programas compactos y de una alta velocidad de ejecución. Una vez que el compilador ha generado el código de Ensamblador (.asm), será necesario utilizar un ensamblador para generar el código binario de máquina.

1.2.4.2 Programación del microcontrolador

Este proceso corresponde a utilizar un programa en el PC que toma el código ensamblado (.hex, .o, .bin, .coff) para el microcontrolador específico, y lo envía mediante algún puerto (serial, paralelo, USB, etc.) a un dispositivo que lo escribe en la memoria del μC . Se acostumbra denominar *programador* tanto al software como al hardware involucrado para este propósito, lo cual puede prestarse a confusión. El software programador a veces recibe también el nombre de *downloader*, ya que su propósito es descargar o transferir desde el PC al μC el código ensamblado. En la *figura 1.4* se muestran las componentes involucradas en el proceso de programación del μC . Es importante mencionar que no deben confundirse los términos *desarrollo o programación del software* y *programación del μC* , el primero se refiere a escribir el programa, mientras que el segundo se refiere transferir el código de máquina a la memoria del μC .

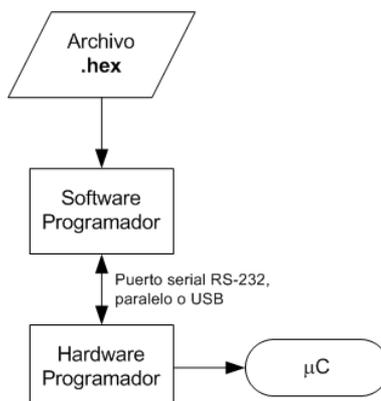


Figura 1.4: Proceso para programación del microcontrolador⁶

1.2.4.3 Prueba y verificación

Una vez programado el μC , se puede instalar en el circuito final para comprobar su adecuado funcionamiento. Existen herramientas de software que permiten simular el comportamiento de un μC , muy útiles cuando el programa alcanza cierta complejidad. Para resolver problemas en un circuito real, el instrumento más utilizado es el analizador lógico.⁷

1.2.5 DIFERENCIA ENTRE MICROPROCESADOR Y MICROCONTROLADOR

1.2.5.1 Arquitectura Von Neumann

La arquitectura tradicional de computadoras y microprocesadores se basa en el esquema propuesto por John Von Neumann, en el cual la unidad central de proceso, o CPU, está conectada a una memoria única que contiene las instrucciones del programa y los datos. El tamaño de la unidad de datos o instrucciones está fijado por el ancho del bus de la memoria, es decir que un **microprocesador** de 8 bits, que tiene además un bus de 8 bits que lo conecta con la memoria, **deberá manejar datos e instrucciones** de una o más unidades de 8 bits de longitud. Cuando deba acceder a una instrucción o datos de más de un byte de longitud, deberá realizar

⁶ http://www2.ing.puc.cl/~mtorrest/downloads/pic/tutorial_pic.pdf

⁷ http://www2.ing.puc.cl/~mtorrest/downloads/pic/tutorial_pic.pdf

más de un acceso a la memoria. Por otro lado **este bus único limita la velocidad de operación del microprocesador**, ya que no se puede buscar de memoria una nueva instrucción, antes de que finalicen las transferencias de datos que pudieran resultar de la instrucción anterior. Es decir que las dos principales limitaciones de esta arquitectura tradicional son:

a) que la longitud de las instrucciones esta limitada por la unidad de longitud de los datos,

Por lo tanto el microprocesador debe hacer varios accesos a memoria para buscar instrucciones complejas

b) que la velocidad de operación (o ancho de banda de operación) está limitada por el efecto de cuello de botella que significa un **bus único para datos e instrucciones**.

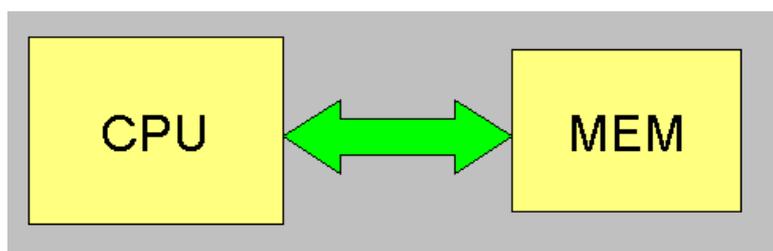


Figura 1.5: Arquitectura Von Neumann (Utilizada en microprocesadores)

1.2.5.2 Arquitectura Harvard

La arquitectura conocida como Harvard, consiste simplemente en un esquema en el que el CPU está conectado a **dos memorias por intermedio de dos buses separados**. Una de las memorias contiene solamente las instrucciones del programa, y es llamada **Memoria de Programa**. La otra memoria solo almacena los datos y es llamada **Memoria de Datos**. Ambos buses son totalmente independientes y pueden ser de distintos anchos. Para un procesador de Set de Instrucciones Reducido, o RISC⁸, el set de instrucciones y el bus de la memoria de programa pueden diseñarse de manera tal que todas las instrucciones tengan una

⁸ RISC: Reduced Instrucción Set Computer

sola posición de memoria de programa de longitud. Además, como los buses son independientes, el CPU puede estar accediendo a los datos para completar la ejecución de una instrucción, y al mismo tiempo estar leyendo la próxima instrucción a ejecutar. Se puede observar claramente que las principales ventajas de esta arquitectura son:

a) Que el tamaño de las instrucciones no está relacionado con el de los datos, y por lo tanto puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa, logrando así mayor velocidad y menor longitud de programa.

b) Que el tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad de operación.

Una pequeña desventaja de los procesadores con arquitectura Harvard, es que deben poseer instrucciones especiales para acceder a tablas de valores constantes que pueda ser necesario incluir en los programas, ya que estas tablas se encontrarán físicamente en la memoria de programa.

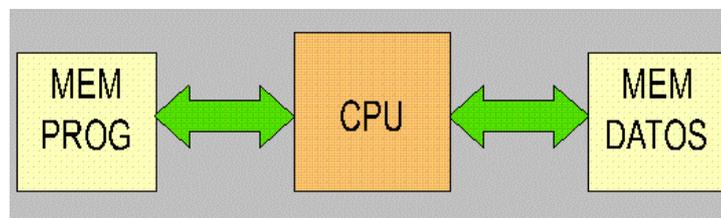


Figura 1.6: Arquitectura Harvard (Utilizada en los microcontroladores)

1.2.6 PARTES DEL MICROCONTROLADOR

- Procesador
- Memoria de datos
- Memoria de Programa
- Líneas de entrada / salida
- Recursos Auxiliares

1.2.6.1 Procesador

Es el cerebro del sistema que procesa todos los datos que viajan a lo largo del bus y determina sus principales características, tanto a nivel hardware como software. Se encarga de direccionar la memoria de instrucciones, recibir el código de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

Además existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales estas son:

- **CISC:** Un gran número de procesadores usados en los microcontroladores están basados en la filosofía CISC (Computadores de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución. Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros.
- **RISC:** Tanto la industria de los computadores comerciales como la de los microcontroladores están decantándose hacia la filosofía RISC.
En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo. La sencillez y rapidez de las instrucciones permiten optimizar el y el software del procesador.
- **SISC:** En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es “específico”, o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC (Computadores de Juego de Instrucciones Específico)

1.2.6.2 Memoria de programa

Es una memoria de almacenamiento no volátil (ROM, EPROM, etc.) en la que se almacena el programa que gobierna la aplicación a la que está destinado el

microcontrolador. En general tiene un tamaño muy reducido desde 512 a 32 K palabras en la familia de los PIC.

Además al tener una memoria específica de programa se puede adecuar el tamaño de las instrucciones y los buses al más adecuado para cada aplicación.

1.2.6.3 Memoria de datos

La memoria para almacenar datos debe ser de lectura y escritura, por lo que en general, en los microcontroladores se usa memoria SRAM, aunque algunos micros llevan memoria EEPROM para evitar la pérdida de los datos en caso de corte en el suministro de corriente. Los tamaños son mucho más reducidos que la memoria de programa

1.2.6.4 Puertos E/S

Estas líneas son los pines del microcontrolador que sirven para comunicarse con los periféricos conectados al sistema.

Pueden enviar o recibir datos digitales al o desde el periférico. Manejan la información en paralelo y se agrupan en conjuntos que reciben el nombre de **puertas o pórticos**. Los pines de las puertas pueden configurarse independientemente cada una de ellos como entrada o salida.

1.2.6.5 Recursos Auxiliares

Cada uno de los microcontroladores de un fabricante está orientado a una función concreta, por lo que algunos tienen una serie de complementos que aumentan su potencia y flexibilidad.

Los principales recursos auxiliares que incorporan los microcontroladores son:

- **Temporizadores o “Timers”**

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior.

Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso.

- **Perro guardián o Watchdog**

Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicializa el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día. El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema.

Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al Perro guardián antes de que provoque el reset. Si falla el programa o se bloquea, no se refrescará al Perro guardián y, al completar su temporización provocará el reseteo.

- **Protección ante fallo de alimentación o “Brown out”**

Se trata de un circuito que resetea al microcontrolador cuando el VDD es inferior a un voltaje mínimo, brown out. Mientras el voltaje de alimentación sea inferior al de brown out el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor.

- **Estado de reposo ó de bajo consumo**

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para ahorrar energía, factor clave en los aparatos portátiles, los microcontroladores disponen de una instrucción especial

(SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se congelan sus circuitos asociados, quedando sumido en un profundo sueño el microcontrolador. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

- **Conversor A/D (CAD)**

Los microcontroladores que incorporan un conversor A/D (Analógico/Digital) pueden procesar señales analógicas, tan abundantes en las aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde los pines del circuito integrado.

- **Conversor D/A (CDA)**

Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por una de los pines del integrado.

- **Comparador analógico**

Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las patitas de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra.

También hay modelos de microcontroladores con un módulo de tensión de referencia que proporciona diversas tensiones de referencia que se pueden aplicar en los comparadores.

- **Modulador de ancho pulsos o PWM**

Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de los pines del encapsulado.

- **Puertas de E/S digitales**

Las líneas digitales de los puertos del microcontrolador pueden configurarse como Entrada o como Salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración.

- **Puertas de comunicación**

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos.

Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

- UART, adaptador de comunicación serie asíncrona.
- USART, adaptador de comunicación serie síncrona y asíncrona
- Puerta paralela esclava para poder conectarse con los buses de otros microprocesadores.
- USB (Universal Serial Bus), que es un moderno bus serie para los PC.
- Bus I2C, que es un interfaz serie de dos hilos desarrollado por Philips.
- CAN (Controller Área Network), para permitir la adaptación con redes de conexionado multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles.

1.2.7 MICROCONTROLADORES PICS

1.2.7.1 Introducción

Los PIC⁹ son una familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc., originalmente desarrollado por la división de microelectrónica de General Instruments. El nombre actual no es un acrónimo. En realidad, el nombre completo es PICmicro, aunque generalmente se utiliza como Peripheral Interface Controller o Controlador de Interfaz Periférico. Los microcontroladores PIC fueron los

⁹ PIC.- peripheral interface controller

primeros microcontroladores RISC, es decir, microcontroladores con un juego de instrucciones reducido. El hecho de ser procesadores de tipo RISC generalmente implica simplicidad en los diseños, permitiendo más características a bajo costo. Los principales beneficios de esta simplicidad en el diseño son que los microcontroladores se implementan en chip muy pequeños, con pocos pines, y tienen un consumo de potencia muy bajo. Los microcontroladores PIC ganaron popularidad rápidamente. Su uso es cada vez mayor en gran número de diseños. Debido a su bajo costo, pequeño tamaño, y bajo consumo, estos microcontroladores pueden ahora ser usados en áreas en las que previamente no habrían sido apropiados, tal como circuitos lógicos.

1.2.7.2 Ventajas de los PICs

Los PICs, presentan grandes ventajas dentro del grupo de microcontroladores, estas son:

- Eficiencia de código.- permiten una gran compactación de los programas
- Rapidez de ejecución
- RISC.- juego de instrucciones reducido para un fácil aprendizaje
- Compatibilidad de pines y código entre dispositivos de la misma familia
- Gran variedad de versiones en distintos encapsulados
- Herramientas de desarrollo tanto en software como en Hardware a un bajo costo

1.3 SENSORES

1.3.1 CONCEPTO

Un sensor es un aparato capaz de transformar magnitudes que se quieren medir o controlar, sean estas físicas o químicas, llamadas variables de instrumentación, en magnitudes eléctricas de corriente o voltaje que puede ser manipulada (medida, amplificada, transmitida, etc.). Las variables de instrumentación dependen del tipo de sensor y pueden ser por ejemplo: temperatura, intensidad lumínica, distancia,

aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, pH, etc. Una magnitud eléctrica obtenida puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una tensión eléctrica (como en un termopar), una corriente eléctrica (como un fototransistor), etc.

Puede decirse también que es un dispositivo que aprovecha una de sus propiedades con el fin de adaptar la señal que mide para que la pueda interpretar otro dispositivo. Como por ejemplo el termómetro de mercurio que aprovecha la propiedad que posee el mercurio de dilatarse o contraerse por la acción de la temperatura. A este tipo se lo conoce como dispositivo de indicación directa.

Un sensor también puede decirse que es un dispositivo que convierte una forma de energía en otra.

Un sensor se diferencia de un transductor en que el sensor está siempre en contacto con la variable a medir o a controlar. Hay sensores que no solo sirven para medir la variable, sino también para convertirla mediante circuitos electrónicos en una señal estándar (4 a 20 mA, o 1 a 5VDC) para tener una relación lineal con los cambios de la variable sensada dentro de un rango (span), para fines de control de dicha variable en un proceso.

Los sensores pueden estar conectados a un computador para obtener ventajas como son el acceso a una base de datos, la toma de valores desde el sensor, etc.

Por lo general, la señal de salida de los sensores no es apta para su lectura directa y a veces tampoco para su procesado, por lo que se usa un circuito de acondicionamiento, como por ejemplo un puente de Wheatstone, amplificadores y filtros electrónicos que adaptan la señal a los niveles apropiados para el resto de la circuitería (posiblemente a un convertidor analógico a digital, un computador y un display) de modo que los valores detectados puedan ser leídos por un humano.

1.3.2 CARACTERÍSTICAS GENERALES DE LOS SENSORES

El transductor ideal sería aquel en que la relación entre la magnitud de entrada y la magnitud de salida fuese proporcional y de respuesta instantánea e idéntica para todos los elementos de un mismo tipo.

Sin embargo, la respuesta real de los transductores nunca es del todo lineal, tiene un rango limitado de validez, suele estar afectada por perturbaciones del entorno exterior y tiene un cierto retardo en la respuesta.

Las características de los transductores se pueden agrupar en dos grandes bloques:

- Características estáticas, que describen la actuación del sensor en régimen permanente o con cambios muy lentos de la variable a medir.
- Características dinámicas, que describen el comportamiento del sensor en régimen transitorio.

1.3.2.1 Características estáticas

Rango de medida: dominio en la magnitud medida en el que puede aplicarse el sensor. El conjunto de valores que puede tomar la señal de entrada comprendidos entre el máximo y el mínimo detectados por el sensor con una tolerancia de error aceptable.

Precisión: es el máximo error esperado en la medida, define la variación máxima entre la salida real obtenida y la salida teórica dada como patrón para el sensor.

Offset o desviación de cero: valor de la variable de salida cuando la variable de entrada es nula. Si el rango de medida no llega a valores nulos de la variable de entrada, habitualmente se establece otro punto de referencia para definir el offset.

Linealidad: un transductor es lineal si existe una constante de proporcionalidad única que relaciona los incrementos de la señal de salida con los respectivos incrementos de la señal de entrada en todo el rango de medida.

Sensibilidad: es la relación entre la variación de la magnitud de salida y la variación de la magnitud de entrada, indica la mayor o menor variación de la señal de salida por unidad de la magnitud de entrada. Cuanto mayor sea la variación de la señal de salida producida por una variación en la señal de entrada, el sensor es más sensible.

Resolución: Indica la mínima variación de la magnitud de entrada que puede producir una variación detectable en la magnitud de salida. Indica la capacidad del sensor para discernir entre valores muy próximos de la variable de entrada.

Derivas: son otras magnitudes, aparte de la medida como magnitud de entrada, que influyen en la variable de salida. Por ejemplo, pueden ser condiciones ambientales, como la humedad, la temperatura u otras como el envejecimiento (oxidación, desgaste, etc.) del sensor.

Repetitividad: Indica la máxima variación o error esperado entre valores de salida obtenidos al medir varias veces la misma entrada con el mismo sensor y en idénticas condiciones ambientales.

Ruido: cualquier perturbación aleatoria del propio sistema de medida que afecta la señal que se quiere medir.

Resolución y precisión: La resolución puede ser de menor valor que la precisión. Por ejemplo, si al medir una distancia la resolución es de 0,01 mm, pero la precisión es de 1 mm, entonces pueden apreciarse variaciones en la distancia medida de 0,01 mm, pero no puede asegurarse que haya un error de medición menor a 1 mm. En la mayoría de los casos este exceso de resolución conlleva a un exceso innecesario en el coste del sistema. No obstante, en estos sistemas, si el error en la medida sigue una distribución normal o similar, lo cual es frecuente en errores accidentales, es decir, no sistemáticos, la repetitividad podría ser de un valor inferior a la precisión.

Sin embargo, la precisión no puede ser de un valor inferior a la resolución, pues no puede asegurarse que el error en la medida sea menor a la mínima variación en la magnitud de entrada que puede observarse en la magnitud de salida.

1.3.2.2 Características dinámicas

Velocidad de respuesta: mide la capacidad del sensor para que la señal de salida siga sin retraso las variaciones de la señal de entrada. Puede ser un tiempo fijo o depender de cuánto varíe la magnitud a medir, depende de la capacidad del sistema para seguir las variaciones de la magnitud de entrada.

Respuesta en frecuencia: mide la capacidad del sensor para seguir las variaciones de la señal de entrada a medida que aumenta la frecuencia, generalmente los sensores convencionales presentan una respuesta del tipo pasabajos.

Estabilidad: indica la desviación en la salida del sensor con respecto al valor teórico dado, al variar parámetros exteriores distintos al que se quiere medir (condiciones ambientales, alimentación, etc.).

1.3.3 TIPOS DE SENSORES

En la siguiente tabla se indican algunos tipos y ejemplos de sensores electrónicos.

Tabla 1.1: Diversos tipos de sensores.

| Magnitud | Transductor | Característica |
|------------------------------|---|----------------|
| Posición lineal o angular | Potenciómetro | Analógica |
| | Encoder | Digital |
| Desplazamiento y deformación | Transformador diferencial de variación lineal | Analógica |
| | Galga extensiométrica | Analógica |
| | Magnetostrictivos | A/D |
| | Magnetorresistivos | Analógica |
| | LVDT | Analógica |

| | | |
|----------------------------------|-------------------------|-----------------------|
| Velocidad lineal y angular | Dinamo tacométrico | Analógica |
| | Encoder | Digital |
| | Detector inductivo | Digital |
| | Servo-inclinómetros | A/D |
| | RVDT | Analógica |
| | Giróscopo | |
| Aceleración | Acelerómetro | Analógica |
| | Servo-acelerómetros | |
| Fuerza y par (deformación) | Galga extensiométrica | Analógica |
| | Triaxiales | A/D |
| Presión | Membranas | Analógica |
| | Piezoeléctricos | Analógica |
| | Manómetros digitales | Digital |
| Caudal | Turbina | Analógica |
| | Magnético | Analógica |
| Temperatura | Termopar | Analógica |
| | RTD | Analógica |
| | Termistor NTC | Analógica |
| | Termistor PTC | Analógica |
| | Bimetal | I/O |
| | | |
| Sensores de presencia | Inductivos | I/O |
| | Capacitivos | I/O |
| | Ópticos | I/O y Analógica |
| Sensores táctiles | Matriz de contactos | I/O |
| Magnitud | Transductor | Característica |
| Visión artificial | Cámaras de video | Procesamiento digital |
| | Cámaras CCD o CMOS | Procesamiento digital |
| Sensor de proximidad | Sensor final de carrera | |
| | Sensor capacitivo | |
| | Sensor inductivo | |
| | Sensor fotoeléctrico | |
| Sensor acústico (presión sonora) | Micrófono | |

| | | |
|------------------------------|----------------------|--|
| Sensor de acidez | ISFET ¹⁰ | |
| Sensores de luz | Fotodiodo | |
| | Fotorresistencia | |
| | Fototransistor | |
| | Célula fotoeléctrica | |
| Sensor captura de movimiento | Sensores inerciales | |

Algunas magnitudes pueden calcularse mediante la medición y cálculo de otras, por ejemplo, la aceleración de un móvil puede calcularse a partir de la integración numérica de su velocidad. La masa de un objeto puede conocerse mediante la fuerza gravitatoria que se ejerce sobre él en comparación con la fuerza gravitatoria ejercida sobre un objeto de masa conocida (patrón).

1.3.4 AREAS DE APLICACIÓN DE LOS SENSORES

- Industria automotriz
- Industria aeroespacial
- Medicina
- Industria de manufactura
- Robótica, etc.

1.4 COMUNICACIÓN USB

1.4.1 INTRODUCCION A USB

La tecnología USB¹¹ se basa en una arquitectura de tipo serial. Sin embargo, es una interfaz de entrada/salida mucho más rápida que los puertos seriales estándar. La arquitectura serial se utilizó para este tipo de puerto por dos razones principales:

¹⁰ ISFET: Ion Sensitive Field Effect Transistor (transistor de efecto campo sensible a iones).

¹¹ USB.- universal serial bus (Bus serial Universal)

- La arquitectura serial le brinda al usuario una velocidad de reloj mucho más alta que la interfaz paralela debido a que este tipo de interfaz no admite frecuencias demasiado altas.
- Los cables seriales resultan mucho más económicos que los cables paralelos.

1.4.2 CARACTERISTICAS GENERALES DE USB

- Los cables USB están diseñados para trabajar en 4 hilos, de los cuales dos son de alimentación y dos de datos (data+ y data -). Ver figura 1.7

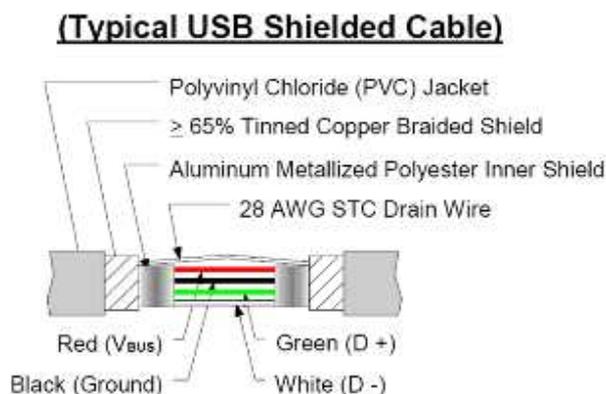


Figura 1.7: Diseño de cable usado por USB¹²

- La distancia máxima del cable que se puede conectar entre el PC¹³ y el periférico es de 5m, pero se puede incrementar la distancia utilizando un hub.
- Su conector es más pequeño en comparación con los conectores seriales o paralelo. (Fig. 1.8)

¹² <http://www.usb.org/developers/docs/> (Usb 2.0, Pag 87)

¹³ PC: Personal computer (computadora personal)

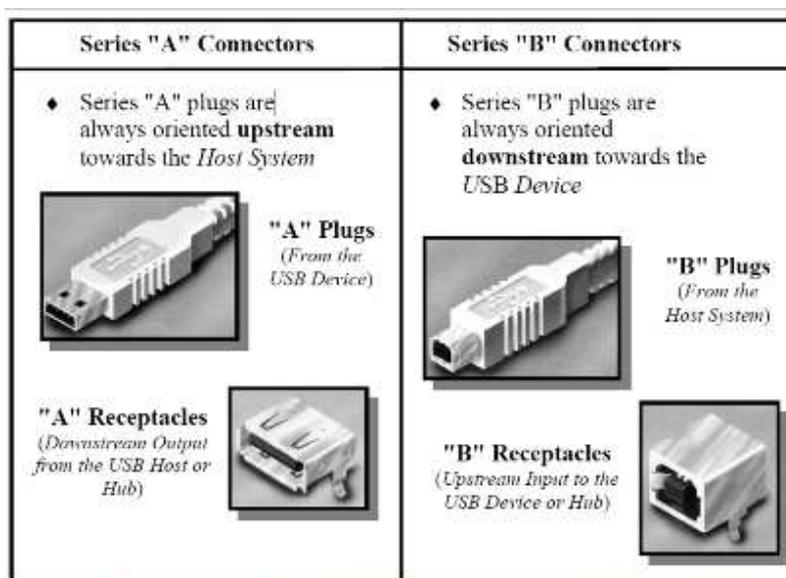


Figura 1.8: Tipos de conectores USB existentes ¹⁴

- El bus del USB puede soportar 3 niveles de transferencia de información HIGH SPEED a 480 Megabits/sec., FULL SPEED a 12 Megabits/sec. y LOW SPEED a 1.5 Megabits/sec. Los PC actuales soportan los tres tipos.
- Tanto los drivers USB como la recepción de los datos y cables garantizan una interfaz que elimina el ruido que pueda provocar error en los datos.

Las señales del USB se transmiten en un cable de par trenzado con impedancia característica de $90 \Omega \pm 15\%$, cuyos hilos se denominan D+ y D-. Estos, colectivamente, utilizan señalización diferencial en full dúplex para combatir los efectos del ruido electromagnético en enlaces largos. D+ y D- suelen operar en conjunto y no son conexiones simples. Los niveles de transmisión de la señal varían de 0 a 0.3 V para bajos (ceros) y de 2.8 a 3.6 V para altos (unos) en las versiones 1.0 y 1.1, y en ± 400 mV en alta velocidad (2.0). En las primeras versiones, los alambres de los cables no están conectados a masa, pero en el modo de alta velocidad se tiene una terminación de 45Ω a tierra o un diferencial de 90Ω para acoplar la

¹⁴ http://www.usb.org/developers/devclass_docs

impedancia del cable. Este puerto sólo admite la conexión de dispositivos de bajo consumo, es decir, que tengan un consumo máximo de 100 mA por cada puerto; sin embargo, en caso de que estuviese conectado un dispositivo que permite 4 puertos por cada salida USB (extensiones de máximo 4 puertos), entonces la energía del USB se asignará en unidades de 100 mA hasta un máximo de 500 mA por puerto.

Un dispositivo con una interfaz USB es probablemente igual o menos costoso que un dispositivo equivalente con una clásica interfaz o una de las más recientes tal como IEEE-1394¹⁵.

Posee un circuito de bajo consumo y un código que automáticamente puede poner el sistema en reposo logrando de esta forma ahorro de energía, y un tiempo de recarga de baterías mas largo.

USB originalmente utiliza una interfaz cableada, pero ahora existe dispositivos inalámbricos que usa el USB para comunicarse con los PCs.¹⁶

Para comunicación con los periféricos comunes tales como teclado, impresora, ratón, USB ha definido clases que especifican los requerimientos de los dispositivos y protocolos.

Un sistema operativo que soporta USB tiene 3 requerimientos mínimos que debe cumplir los cuales son:

- Detección del dispositivo USB en la conexión y desconexión.
- Al existir un nuevo dispositivo conectado, este debe ser capaz de descubrir como intercambiar información con este.
- Provee un mecanismo que habilita el software para comunicación entre el hardware del USB y la aplicación que quiere acceder a los periféricos USB.

Por otro lado los periféricos para manejar la comunicación incluyen en el hardware microcontroladores, los cuales son los encargados de manejar todos estos detalles.

¹⁵ IEEE: Instituto de Ingenieros eléctricos y electrónicos.

¹⁶ http://es.wikipedia.org/wiki/Universal_Serial_Bus

El periférico es responsable de responder las peticiones para enviar y recibir datos usados en la identificación y configuración del dispositivo y para leer y escribir otros datos en el bus.

1.4.3 LIMITACIONES

Como todo equipo o tecnología tienen notables ventajas, también tienen limitaciones que restringen al usuario de utilizar dicha tecnología, el USB no es una excepción.

A continuación se describe las limitaciones que tiene el Estándar USB para su uso:

- USB fue diseñado como un bus de expansión fijo con la expectativa que los periféricos estarían cerca, su cable puede llegar a ser hasta 5 metros. A diferencia de otras interfaces donde se permite mayor alcance, tales como el RS-232, RS-485, ETHERNET y IEEE-1394. Se puede incrementar la distancia a 30 metros colocando 5 hubs. Otra opción cuando se necesita transmitir a distancias grandes es cambiar de USB a RS-485.
- En la comunicación USB, solo se puede comunicar el periférico con el host, pero no host con host, tampoco lo hace entre periféricos como lo hacen otras interfaces como IEEE-1394.
- USB no puede transmitir información simultáneamente a diferentes dispositivos en el bus, los envía individualmente. ETHERNET o IEEE-1394 posee esta habilidad.

1.4.4 CABLES Y CONECTORES USB

En la figura 1.9 se puede apreciar los diferentes tipos de conectores existentes en el mercado, los cuales se usan para las diferentes aplicaciones de periféricos tales como mouses, teclados, HID, impresoras, celulares, etc.



Figura 1.9: Tipos de conectores USB existentes

1.5 MAGNITUDES FISICAS MEDIDAS

1.5.1 PRESION

1.5.1.1 Definición

En física y disciplinas afines, la presión es una magnitud física que mide la fuerza que actúa sobre una unidad de superficie, y sirve para caracterizar como se aplica una determinada fuerza resultante sobre una superficie.

Cuando sobre una superficie plana de área A se aplica una fuerza normal F de manera uniforme y perpendicularmente a la superficie, la presión P viene dada por:

$$P = \frac{F}{A}$$

1.5.1.2 Presión absoluta y relativa

En determinadas aplicaciones la presión se mide no como la presión absoluta sino como la presión por encima de la presión atmosférica, denominándose presión relativa, presión normal, presión de gauge o presión manométrica. Consecuentemente, la presión absoluta es la presión atmosférica más la presión manométrica (presión que se mide con el manómetro).

1.5.1.3 Unidades de medida y sus factores de conversión

En el Sistema Internacional de Unidades la presión se mide en una unidad derivada que se denomina pascal (Pa) que es equivalente a una fuerza total de un newton actuando uniformemente sobre un metro cuadrado de superficie. La presión atmosférica es de aproximadamente de 101.300 pascales (101,3 kPa), a nivel de mar.

Tabla 2.- Unidades de presión y sus factores de conversión

| | Pascal | bar | N/mm ² | kp/m ² | kp/cm ² | atm | Torr |
|-----------------------------------|--------------------|-----------------------|-----------------------|----------------------|------------------------|------------------------|--------|
| 1 Pa (N/m²) | 1 | 1×10 ⁻⁵ | 1×10 ⁻⁶ | 0,102 | 0,102×10 ⁻⁴ | 0,987×10 ⁻⁵ | 0,0075 |
| 1 bar (daN/cm²) | 100000 | 1 | 0,1 | 1020 | 1,02 | 0,987 | 750 |
| 1 N/mm² | 10×10 ⁵ | 10 | 1 | 1,02×10 ⁵ | 10,2 | 9,87 | 7500 |
| 1 kp/m² | 9,81 | 9,81×10 ⁻⁵ | 9,81×10 ⁻⁶ | 1 | 1×10 ⁻⁴ | 0,968×10 ⁻⁴ | 0,0736 |
| 1 kp/cm² | 98100 | 0,981 | 0,0981 | 10000 | 1 | 0,968 | 736 |
| 1 atm (760 Torr) | 101325 | 1,013 | 0,1013 | 10330 | 1,033 | 1 | 760 |
| 1 Torr (mmHg) | 133 | 0,00133 | 1,33×10 ⁻⁴ | 13,6 | 0,00132 | 0,00132 | 1 |

1.5.2 TEMPERATURA

1.5.2.1 Definición

La temperatura es una magnitud referida a las nociones comunes de caliente o frío. Es una medida del calor o energía térmica de las partículas en una sustancia. Como lo que medimos en su movimiento medio, la temperatura no depende del número de partículas en un objeto y por lo tanto no depende de su tamaño.

1.5.2.2 Unidades de temperatura

Las escalas de medición de la temperatura se dividen fundamentalmente en dos tipos, las relativas y las absolutas. Los valores que puede adoptar la temperatura en cualquier escala de medición, no tienen un nivel máximo, sino un nivel mínimo: el cero absoluto. Mientras que las escalas absolutas se basan en el cero absoluto, las relativas tienen otras formas de definirse.

Grados Celsius (°C): Para establecer una base de medida de la temperatura Anders Celsius utilizó (en 1742) los puntos de fusión y ebullición del agua. Se considera que una mezcla de hielo y agua que se encuentra en equilibrio con aire saturado a 1 atm está en el punto de fusión. Una mezcla de agua y vapor de agua (sin aire) en equilibrio a 1 atm de presión se considera que está en el punto de ebullición. Celsius dividió el intervalo de temperatura que existe entre éstos dos puntos en 100 partes iguales a las que llamó grados centígrados °C. Sin embargo en 1948 fueron renombrados grados Celsius en su honor; así mismo se comenzó a utilizar la letra mayúscula para denominarlos.

En 1954 la escala Celsius fue redefinida en la Décima Conferencia de Pesos y Medidas en términos de un sólo punto fijo y de la temperatura absoluta del cero absoluto. El punto escogido fue el punto triple del agua que es el estado en el que las tres fases del agua coexisten en equilibrio, al cual se le asignó un valor de 0,01 °C. La magnitud del nuevo grado Celsius se define a partir del cero absoluto como la fracción $1/273,16$ del intervalo de temperatura entre el punto triple del agua y el cero absoluto. Como en la nueva escala los puntos de fusión y ebullición del agua son 0,00 °C y 100,00 °C respectivamente, resulta idéntica a la escala de la definición anterior, con la ventaja de tener una definición termodinámica.

Grados Kelvin (K): El Kelvin es la unidad de medida del SI. La escala Kelvin absoluta es parte del cero absoluto y define la magnitud de sus unidades, de tal forma que el punto triple del agua es exactamente a 273,16 K.

1.5.2.3 Conversión de unidades de temperatura

Las siguientes fórmulas mostradas en la tabla 3, asocian con precisión las diferentes escalas de temperatura:

Tabla 3.- Unidades de temperatura y sus fórmulas de conversión.

| | Kelvin | Grado Celsius | Grado Fahrenheit | Grado Rankine |
|-------------------------|------------------------------|---------------------------------|--------------------------------|---------------------------------|
| Kelvin | $K = K$ | $K = C + 273,15$ | $K = (F + 459,67) \frac{5}{9}$ | $K = Ra \frac{5}{9}$ |
| Grado Celsius | $C = K - 273,15$ | $C = C$ | $C = (F - 32) \frac{5}{9}$ | $C = (Ra - 491,67) \frac{5}{9}$ |
| Grado Fahrenheit | $F = K \frac{9}{5} - 459,67$ | $F = C \frac{9}{5} + 32$ | $F = F$ | $F = Ra - 459,67$ |
| Grado Rankine | $Ra = K \frac{9}{5}$ | $Ra = (C + 273,15) \frac{9}{5}$ | $Ra = F + 459,67$ | $Ra = Ra$ |

1.5.3 HUMEDAD

1.5.3.1 Definición

Se denomina humedad ambiental a la cantidad de vapor de agua presente en el aire. Se puede expresar de forma absoluta mediante la humedad absoluta, o de forma relativa mediante la humedad relativa o grado de humedad.

1.5.3.2 Humedad absoluta

La humedad absoluta es la cantidad de vapor de agua presente en el aire. Se expresa en gramos de agua por unidad de volumen (g/m^3). A mayor temperatura, mayor es la cantidad de vapor de agua.

1.5.3.3 Humedad relativa

La humedad relativa es la humedad que contiene una masa de aire, en relación con la máxima humedad absoluta que podría admitir sin producirse condensación, conservando las mismas condiciones de temperatura y presión atmosférica, por

ejemplo una humedad relativa del 70% quiere decir que de la totalidad de vapor de agua (el 100%) que podría contener el aire a esta temperatura, solo tiene el 70%.

Esta es la forma más habitual de expresar la humedad ambiental. Se expresa en tanto por ciento. %

$$RH = \frac{P(H_2O)}{P^*(H_2O)} \times 100\%$$

Donde:

$P(H_2O)$: es la presión parcial de vapor de agua en la mezcla de aire;

$P^*(H_2O)$: es la presión de saturación de vapor de agua a la temperatura en la mezcla de aire; y

RH : es la humedad relativa de la mezcla de aire que se está considerando.

La importancia de esta manera de expresar la humedad ambiente estriba en que refleja muy adecuadamente la capacidad del aire de admitir más o menos vapor de agua, lo que, en términos de comodidad ambiental para las personas, expresa la capacidad de evaporar la transpiración, importante regulador de la temperatura del cuerpo humano.

CAPITULO 2

2 DESARROLLO DEL HARWARE

2.1 CONSTRUCCIÓN DEL HARDWARE

La presente investigación va enfocada a usuarios que deseen satisfacer la necesidad de adquirir datos de la medición de cualquier variable física, por ello se ha estudiado las características adecuadas para establecer un sistema de adquisición de datos capaz de solucionar este problema. A continuación se muestra en la figura 2.1, un diagrama de bloques del prototipo planteado para este proyecto.

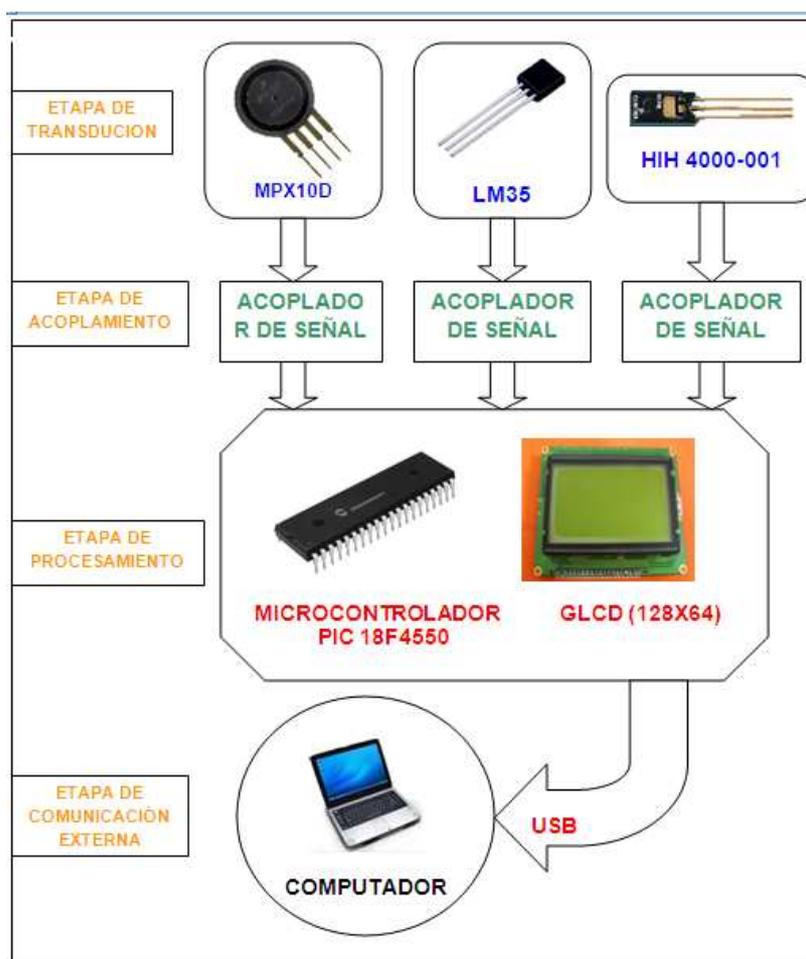


Figura 2.1: Diagrama en bloques del sistema.

Como puede verse consta de una etapa de adquisición de la señal eléctrica, una etapa de procesamiento y una etapa de presentación de datos con los dispositivos electrónicos principales mostrados para cada una de las etapas.

Los detalles que se han tomado en cuenta de la investigación para iniciar con este proyecto son:

- Bajos costos de construcción del equipo.
- Cómodo montaje del sistema de acceso.
- Fácil manejo del sistema de acceso por parte del usuario.
- Seguridad del área determinada, donde se instalará el prototipo.
- Cambio de un sistema mecánico para acceso a un área determinada por un sistema de acceso electrónico.

2.2 REQUERIMIENTOS DEL HARDWARE

Una vez planteadas las principales características del sistema de adquisición, se ha iniciado un estudio investigativo donde se ha logrado establecer que se necesitará dos circuitos y estos son:

- Circuito adaptador de señal: el mismo que consta de 3 circuitos que son:
 - Circuito adaptador de señal del sensor de presión.
 - Circuito adaptador de señal del sensor de temperatura.
 - Circuito adaptador de señal del sensor de humedad.
- Circuito procesador de señal.

El circuito adaptador tomara la señal proveniente de cada uno de los sensores y la adecuara a un rango de 0 a 5V variables, para que pueda ingresar a la etapa de procesamiento de la señal. Dependiendo de la señal a adaptar, el circuito estará construido de lo siguiente:

Para el caso del circuito adaptador de la señal de presión, estará construido con un circuito integrado AD6020, el cual es un amplificador operacional de instrumentación

y toma la señal proveniente del sensor de presión (MP10GS) y lo amplifica a un rango aceptable para el circuito de procesamiento de señal.

En el circuito adaptador de la señal de temperatura, estará construido con un circuito integrado LM358, el cual es un amplificador operacional que toma la señal proveniente del sensor de temperatura (LM35) y lo amplifica a un rango aceptable para el circuito de procesamiento de señal.

En el circuito adaptador de la señal de humedad, estará construido por una resistencia variable de instrumentación y un condensador de 10uF, los cuales en conjunto mantendrán la señal proveniente del sensor de humedad (HIH 4000) en niveles de voltaje aceptables para el circuito de procesamiento de señal.

El circuito procesador de señal es un circuito impreso donde se utiliza un microprocesador PIC18F4550 y un modulo GLCD, el primero toma las señales provenientes de los circuitos adaptadores para digitalizarlas y procesarlas de acuerdo al software que contenga y el segundo permite visualizar los datos en sistema decimal con las respectivas unidades de la variable física medida. A parte de lo anteriormente mencionado, el circuito procesador de señal cuenta con un puerto USB, el cual nos permite establecer comunicación a un computador y a través de este poder visualizar las variables físicas medidas con la ayuda de una interfaz grafica generada en Visual Basic.

2.3 COMPONENTES DEL HARDWARE

Para iniciar la construcción del sistema de adquisición de datos, se han tomado en cuenta los componentes necesarios que formarán parte de los circuitos adaptadores de señal y procesador de señal; se debe recordar a su vez, que el circuito acoplador d señal se componen de 3 partes.

A continuación se menciona el funcionamiento y las características principales de los dispositivos electrónicos del sistema de adquisición de datos que conformarán el proyecto.

2.3.2 SENSORES

2.3.2.1 Sensores de presión

Sensores de Presión Resistivos: Una presión sobre una membrana hace variar el valor de las resistencias montadas en puente de Wheatstone apareadas. Las Células de Carga y las Galgas Extensiométricas son elementos metálicos que cuando se someten a un esfuerzo sufren una deformación del material, y por lo tanto una variación de su resistencia interna.

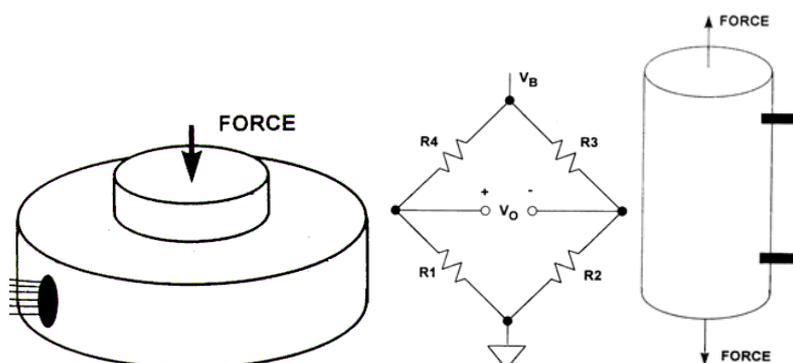


Figura 2.2: Mecanismo de sensor de presión resistivo.

Sensores de Presión Piezo-Cerámicos/Multicapa: La combinación de la tecnología piezo-cerámica y multicapa se utiliza para producir una señal eléctrica, cuando se aplica una fuerza mecánica en el sensor.

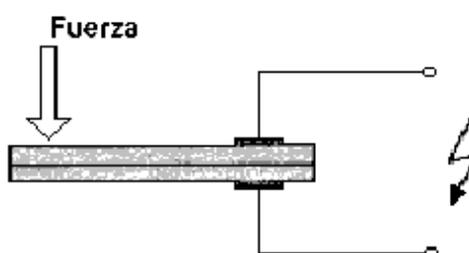


Figura 2.3: Esquema de funcionamiento de un sensor de presión piezo-cerámico.

Sensores de Presión con Semiconductores: Una variación de presión sobre una membrana, hace actuar un único elemento piezo-resistivo semiconductor. *Motorola* fabrica sensores de presión con su familia MPX.

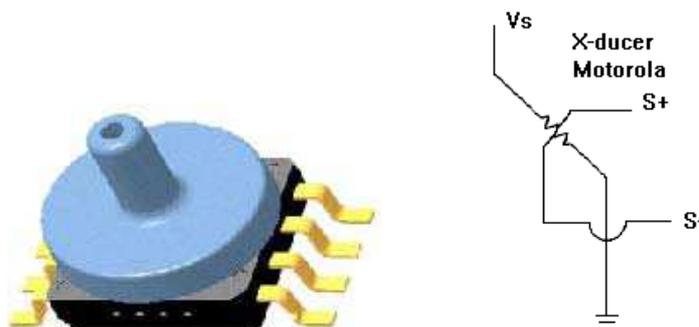


Figura 2.4: Sensor de presión con semiconductores.

Sensor de presión MPX10D

Motorola fabrica una serie de sensores de presión con semiconductores, cuyas prestaciones y su bajo precio son parte de la tecnología asociada. Un diseño único, hace que estos dispositivos sean altamente competitivos.

El funcionamiento de los sensores de presión de la serie MPX de *Motorola* está basado en el diseño patentado del calibrador de fuerza. A diferencia de los sensores de presión más convencionales que utilizan las cuatro resistencias exactamente apareadas en una configuración de puente de Wheatstone, la serie MPX utiliza solamente un único elemento piezo-resistivo implantado en un diafragma de silicio que sensa la tensión mecánica inducida en el diafragma por una presión externa. Al aplicar presión al diafragma, se produce un cambio de resistencia en la galga extensiométrico, que causa a su vez un cambio en el voltaje de salida en proporción directa a la presión aplicada. La salida es un voltaje analógico proporcional a la presión de entrada y al voltaje de alimentación radiométrico. La alta sensibilidad y una excelente repetibilidad a largo plazo hacen que sean las unidades más apropiadas para la mayoría de aplicaciones.

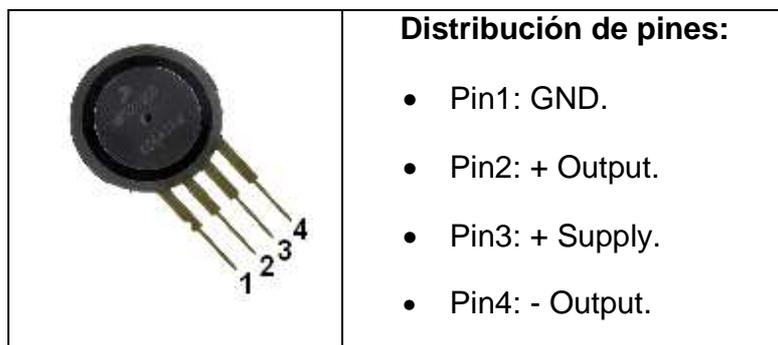


Figura 2.5: Distribución de pines del sensor de presión MPX10D.

2.3.2.2 Sensores de temperatura

Probablemente sea la temperatura el parámetro físico más común que se mide en una aplicación electrónica, incluso en muchos casos en que el parámetro de interés no es la temperatura, ésta se ha de medir para incluir indirectamente su efecto en la medida deseada.

La diversidad de sus aplicaciones ha condicionado igualmente una gran proliferación de dispositivos sensores y transductores, desde la sencilla unión bimetálica de los termostatos, hasta los dispositivos semiconductores más complejos.

Termopares: Los termopares utilizan la tensión generada en la unión de dos metales en contacto térmico, debido a sus distintos comportamientos eléctricos.

Resistivos: Lo constituyen las RTD (Resistance Temperature Detector) o PT100 basadas en la dependencia de la resistividad de un conductor con la temperatura, están caracterizadas por un coeficiente de resistividad positivo PTC (Positive Thermal Coefficient). También lo son las NTC (Negative Thermal Coefficient), que se llaman termistores y están caracterizadas por un coeficiente de temperatura negativo.

Semiconductores: Se basan en la variación de la conducción de una unión p-n polarizada directamente.

Sensor de temperatura LM35

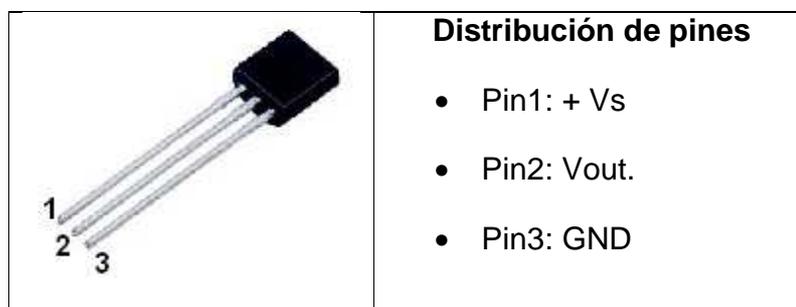


Figura 2.6: Distribución de pines del sensor de temperatura LM35.

2.3.2.3 Sensores de humedad

Sensores de Humedad Capacitivos: El sensor de los sensores lo forma un condensador de dos láminas de oro como placas y como dieléctrico una lámina no conductora que varía su constante dieléctrica, en función de la humedad relativa de la atmósfera ambiente. El valor de la capacidad se mide como humedad relativa.

Philips fabrica este tipo de sensores.

Sensores de Humedad Resistivos: Un electrodo polímero montado en tandem sensa la humedad en el material. Además un circuito acondicionador y linealizador dan una salida estándar.

Estos sensores se aplican en Alarmas en lavadoras, acondicionadores de aire, humidificadores, higrómetros, incubadoras, sistemas respiratorios en medicina, etc.

Sensor de humedad HIH 4000-001

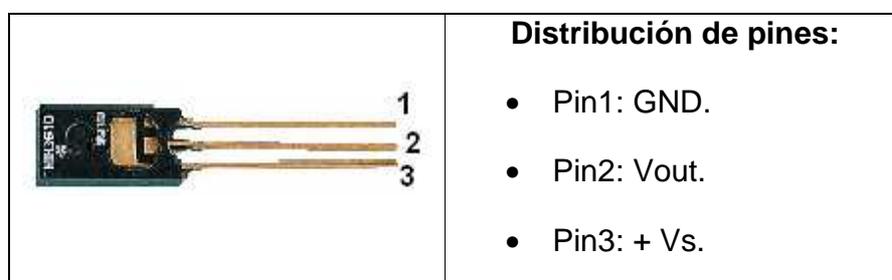


Figura 2.7: Distribución de pines del sensor de humedad HIH 4000-001.

2.3.3 CIRCUITOS INTEGRADOS

Un **circuito integrado** (CI) o **chip**, es una pastilla muy delgada en la que se encuentra una enorme cantidad (del orden de miles o millones) de dispositivos microelectrónicos interconectados, principalmente diodos y transistores, además de componentes pasivos como resistencias o condensadores. Su área es de tamaño reducido, del orden de un cm^2 o inferior. Algunos de los circuitos integrados más avanzados son los microprocesadores, que son usados en múltiples artefactos, desde computadoras hasta electrodomésticos, pasando por los teléfonos móviles. Otra familia importante de circuitos integrados la constituyen las memorias digitales.

El circuito integrado que nos será útil para adaptar la señal del sensor de presión a la tarjeta de procesamiento de señales de manera que ésta esté dentro del rango aceptable es: un amplificador operacional de instrumentación AD620AN.

2.3.3.1 Amplificador operacional AD620AN

Este circuito integrado es un amplificador operacional de instrumentación (precisión), entre sus características principales tenemos las siguientes:

- Ancho de banda: -3dB, 120KHZ
- CMRR: 73Db
- Desviación de ganancia ($\text{ppm}/^\circ\text{C}$): 50
- Desviación (mV) : 0.125
- Encapsulado: DIP8
- Error de ganancia (%) : 0.7
- Función: Baja potencia, regulación de ganancia con resistencia
- Gama de ganancia: 1 \rightarrow 10000
- Gama de suministro/división & única: $\pm 2,3 \rightarrow \pm 18 \text{ V}$
- Rango de temperaturas de funcionamiento: -40 - +85°C

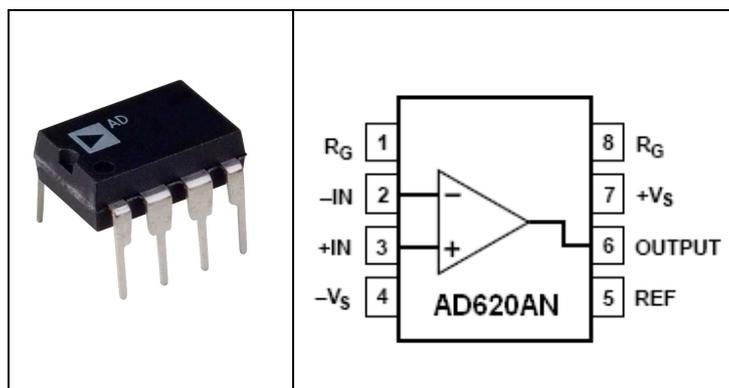


Figura 2.8: Distribución de pines del CI AD620AN.

2.3.3.2 Amplificador operacional LM358

El LM358 es un amplificador operacional con alimentación única de 5 Voltios.

La cápsula contiene dos circuitos idénticos independientes de alta ganancia, con frecuencia internamente compensada, fueron diseñados expresamente para funcionar con una sola fuente de energía con una amplia gama de voltajes.

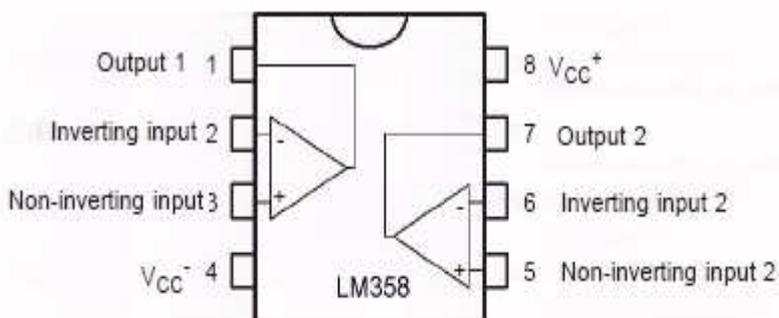


Figura 2.9: Distribución de pines del LM358.

Las áreas de aplicación del LM358 incluyen amplificadores para transductores, sistemas de ganancia en continua y todos los circuitos de op-amp convencionales que así pueden ser más fácilmente puestos en práctica en sistemas de fuente de energía única. Por ejemplo, estos circuitos pueden ser alimentados directamente de +5V estándar que se usa en sistemas lógicos y proporcionará fácilmente la electrónica de interfaz requerida sin una fuente de energía adicional.

2.3.4 MODULO GLCD (Graphic Liquid Cristal Display)

Dentro de las pantallas LCD hay un grupo que se denominan "gráficas". En lugar de representar caracteres, como las populares pantallas de dos filas de 16 caracteres, pueden, directamente, mostrar puntos sueltos.

Esto las hace especiales para la representación de gráficos, aunque por supuesto, con un programa adecuado, puede mostrar texto, incluso con diferentes fuentes y tamaños.

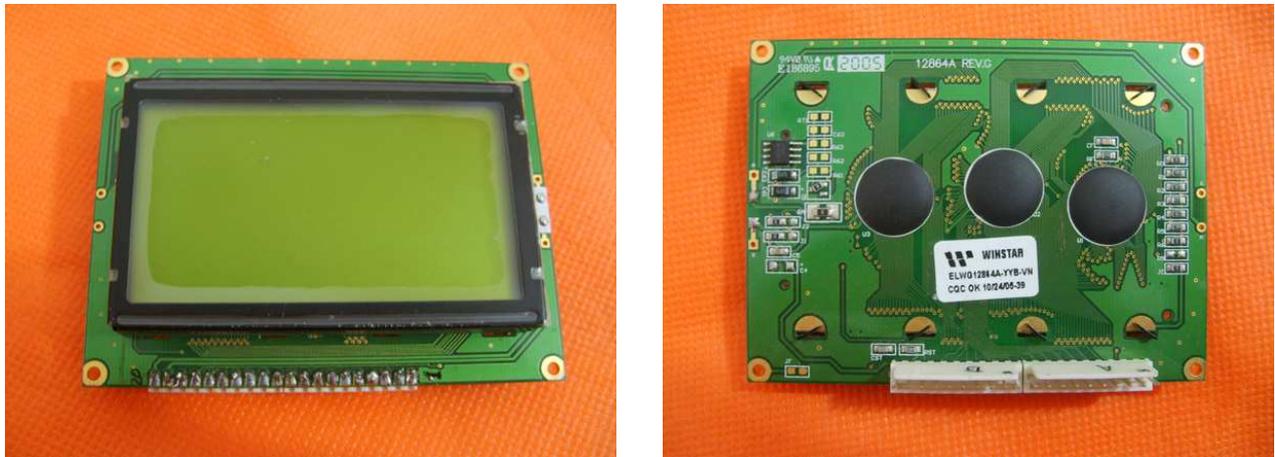


Figura 2.10: Display gráfico de cristal liquido 128x624 (GLCD).

2.3.4.1 Distribución de pines

La distribución de pines de nuestro GLCD es la que se muestra a continuación en la tabla 2.1.:

Tabla 2.1: Distribución de pines del GLCD

| Pin NO. | Symbol | Function |
|---------|------------------|----------------------------------|
| 1 | Vss | GND |
| 2 | Vdd | Power supply (+5V) |
| 3 | Vo | Contrast Adjustment |
| 4 | D/I | Data/Instruction |
| 5 | $\overline{R/W}$ | Data read/write |
| 6 | E | H→L Enable signal |
| 7 | DB0 | Data bus line |
| 8 | DB1 | Data bus line |
| 9 | DB2 | Data bus line |
| 10 | DB3 | Data bus line |
| 11 | DB4 | Data bus line |
| 12 | DB5 | Data bus line |
| 13 | DB6 | Data bus line |
| 14 | DB7 | Data bus line |
| 15 | CS1 | Chip select for IC1 |
| 16 | CS2 | Chip select for IC2 |
| 17 | \overline{RST} | Reset |
| 18 | Vee | Negative voltage output |
| 19 | A | Power supply for LED+(4.2V)RA=0Ω |
| 20 | K | Power supply for LED (0V) |

2.3.5 MICROCONTROLADOR PIC 18F4550

Un microcontrolador es un circuito integrado que en su interior contiene toda la arquitectura de un computador, este elemento debe ser programado para que realice una actividad determinada. Para este proyecto hemos utilizado el microcontrolador PIC 18F4550 para manejar el programa del circuito procesador de señal.

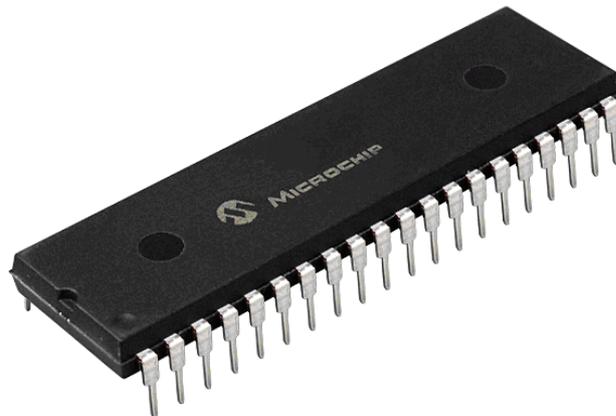


Figura 2.11: Circuito integrado PIC 18F4550.

2.3.5.1 Características

El PIC18F4550 es un microcontrolador de 40 pines, fácil de programar y que posee internamente características importantes como las siguientes:

- Interface USB 2.0 de alta velocidad 12Mbit/s.
- Tipo de memoria: Flash.
- Memoria de programa: 32 Kb.
- CPU Speed (MIPS): 12.
- Memoria Ram: 2048 bytes.
- EEPROM: 256 bytes.
- Voltaje de operación: 2 a 5.5 V.
- Memoria ROM: 16KB.
- Memoria RAM: 2Kb.
- Frecuencia: 48 MHz con cristal externo.
- Permite programación ICSP.
- Dispone de 5 puertos de E/S, que incluyen un total de 35 líneas digitales de E/S.
- Permite comunicación directa USB 2.0 por hardware.

2.3.5.2 Distribución de pines

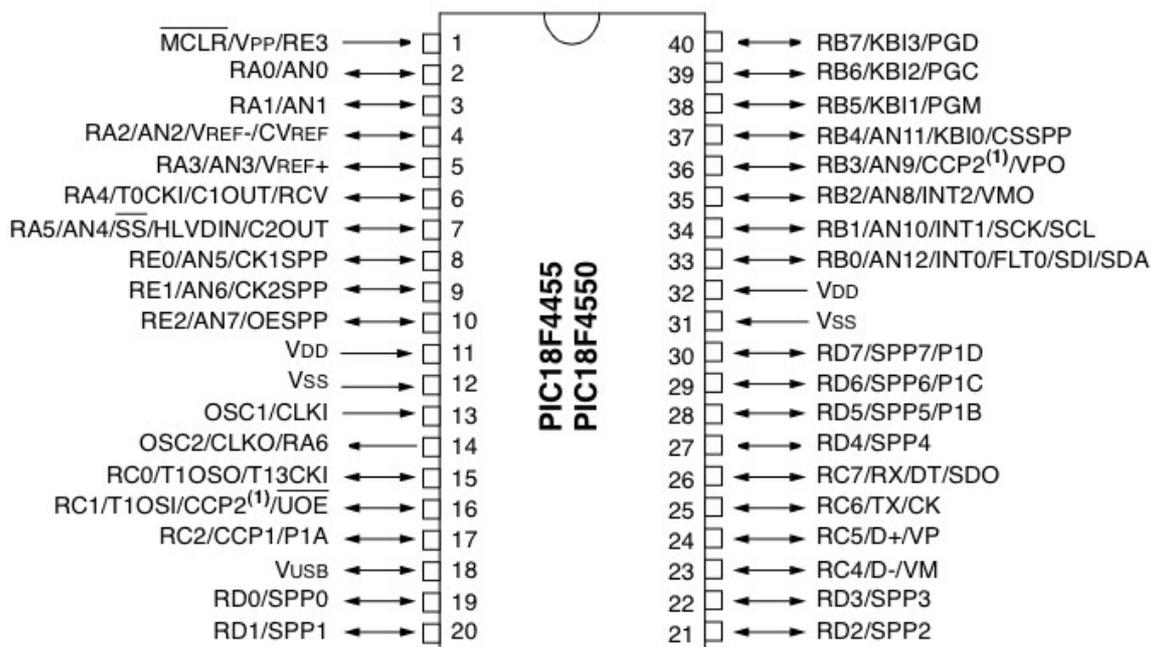


Figura 2.12: Distribución de pines de CI PIC 18F4550.

2.4 ENSAMBLAJE DEL HARDWARE

Para comenzar con la construcción del hardware, se debe considerar el tener listos el diseño del circuito a ensamblarse, los elementos necesarios a emplearse, el software que los manejan y todo lo indispensable para su armado, comprendiendo el funcionamiento de cada uno de ellos para poder lograr el objetivo planteado en el proyecto y así poder satisfacer las necesidades requeridas para el mismo.

Es por ello que para esta etapa del proyecto nos basaremos en la información recopilada anteriormente sobre los elementos electrónicos a utilizarse y en los anexos ubicados en las páginas posteriores.

2.4.1 DESARROLLO DEL CIRCUITO PROCESADOR DE SEÑAL

Los materiales que se han utilizado en el ensamblaje del circuito procesador de señal, se enlistan a continuación:

- Dos condensadores de 10uF.
- Un condensador de 100uF / 25V.
- Un condensador de 0.21uF.
- Dos condensadores de 22pF.
- Un condensador de 47uF / 25V.
- Dos diodos 1N4007.
- Un regulador de voltaje en CI 7805.
- Un jack de energía.
- Un header de 20 pines.
- Ocho conectores tipo bornera de 3 bornes.
- Cuatro Headers de 2 pines.
- Dos diodos LED.
- Un potenciómetro de precisión de 50K Ω .
- Tres resistencias de 1K Ω .
- Una resistencia de 100K Ω .
- Una resistencia de 10 Ω .
- Seis resistencias de 3.9K Ω .
- Seis Pulsadores N/A.
- Tres interruptores
- Un conector USB
- Un zócalo de 40 pines
- Un PIC 18F4550.
- Un cristal de 20MHz.
- Un display grafico de cristal líquido (GLCD).



Figura 2.13: Componentes electrónicos del circuito procesador de señal.

Como se definió anteriormente, este circuito es un procesador de señal, las cuales ingresan a al circuito a través de las borneras de manera analógica en un rango de 0 a 5 voltios, este valor variable depende del parámetro físico que el sensor este midiendo. Una vez en el circuito, las señales son digitalizadas gracias a los convertidores A/D que posee internamente el microprocesador, el cual también toma dichas señales digitales y las procesa según el programa que este guardado en su memoria, para posteriormente enviarlas hacia el GLCD y poder visualizar los datos de la medición. Dichos datos también son posibles de visualizarlos en un computador con la ayuda de una interfase grafica generada en Visual Basic, ya que el circuito tiene la capacidad de transmitir datos vía USB a un computador

El siguiente diagrama lógico, nos indica la distribución de los elementos pasivos y activos dentro del circuito. A continuación se muestra las conexiones realizadas para este circuito.

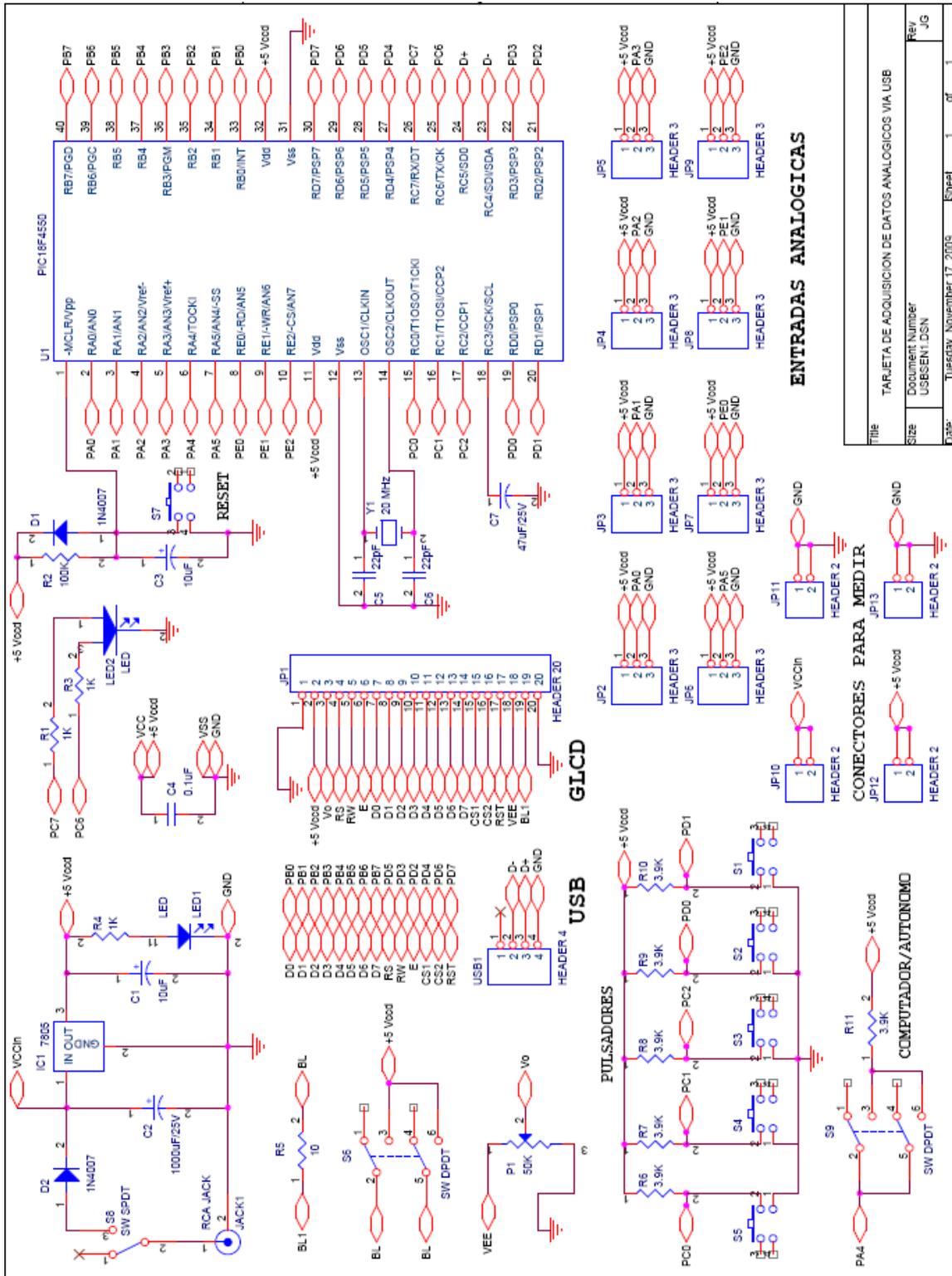


Figura 2.14: Diagrama lógico del circuito procesador de señal.

| | |
|-------|--|
| TITLE | TARJETA DE ADQUISICION DE DATOS ANALOGICOS VIA USB |
| Size | Document Number USBSEN1.DSN |
| Date: | Tuesday, November 17, 2009 |
| Sheet | 1 of 1 |
| Rev | JG |

2.4.2 DESARROLLO DE LOS CIRCUITOS ADAPTADORES DE SEÑAL

2.4.2.1 Circuito adaptador del sensor de presión

Los materiales que se han utilizado en el ensamblaje del circuito adaptador de señal, del sensor de presión, se enlistan a continuación:

- Tres condensadores de 0.1uF.
- Un condensador de 10uF / 25V.
- Dos diodos LED.
- Un regulador de voltaje en CI LM1117Txx.
- Dos Headers de 2 pines.
- Un header de 4 pines.
- Un conector tipo bornera de 2 bornes.
- Dos potenciómetros de precisión de 1K Ω .
- Un potenciómetro de precisión de 100K Ω .
- Una resistencia de precisión de 15K Ω .
- Dos resistencias de precisión de 33K Ω .
- Dos resistencias de precisión de 2.2K Ω .
- Un CI AD620AN.
- Un zócalo de 8 pines.

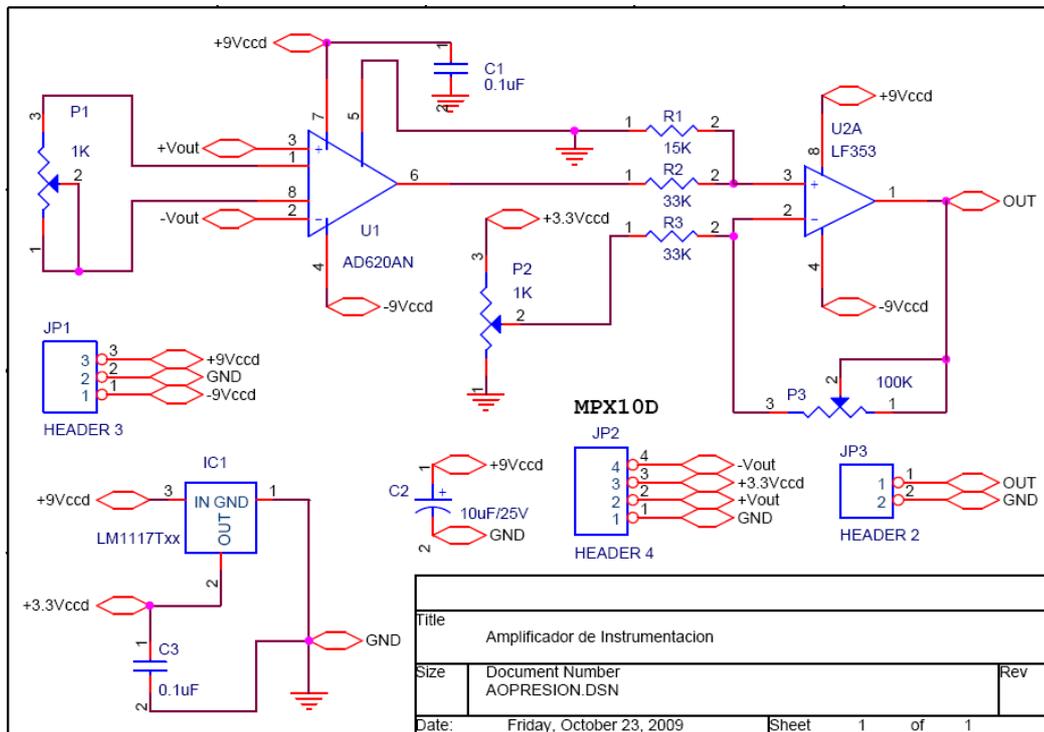


Figura 2.15: Diagrama lógico del circuito adaptador de señal del sensor de presión.

2.4.2.2 Circuito adaptador del sensor de temperatura

Los materiales que se han utilizado en el ensamblaje del circuito adaptador de señal del sensor de temperatura, se enlistan a continuación:

- Dos condensadores de 10uF.
- Un diodo LED
- Un header de 3 pines.
- Un conector tipo bornera de 3 bornes.
- Un potenciómetro de precisión de 20K Ω
- Una resistencia de 4.99K Ω
- Una resistencia de 1K Ω
- Un zócalo de 8 pines
- Un CI LM358

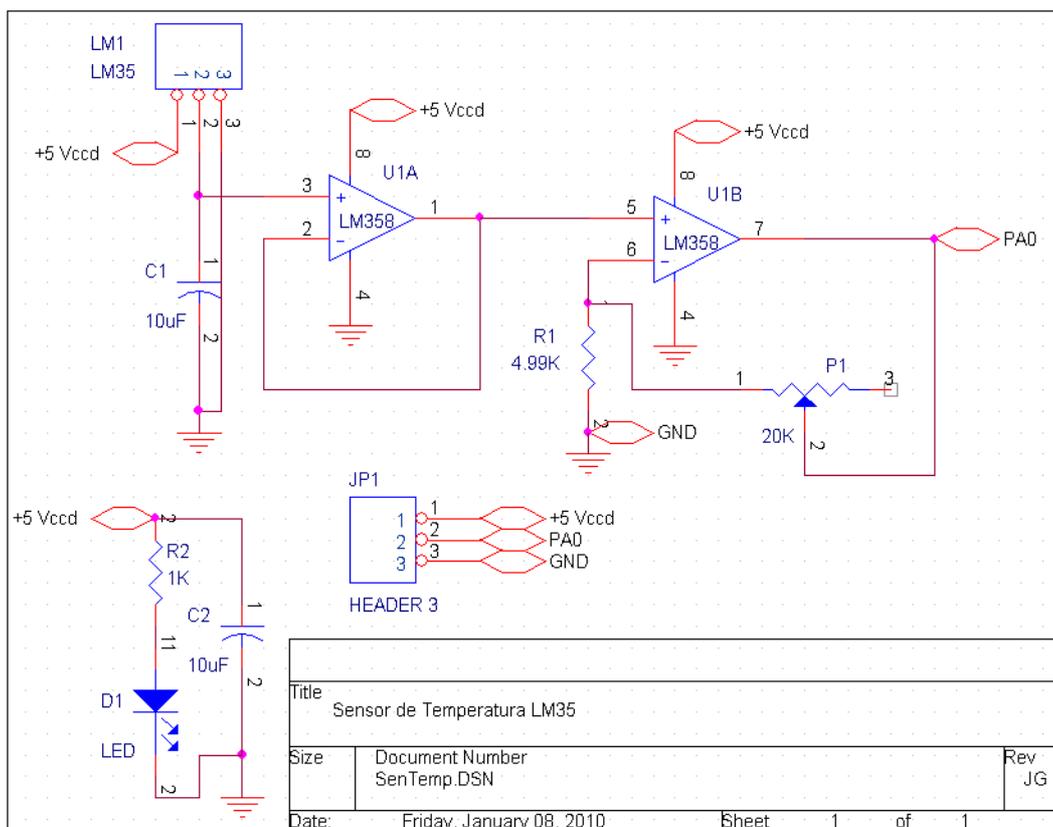


Figura 2.16: Diagrama lógico del circuito adaptador de señal del sensor de temperatura.

2.4.2.3 Circuito adaptador del sensor de humedad

Los materiales que se han utilizado en el ensamblaje del circuito adaptador de señal del sensor de humedad, se enlistan a continuación:

- Dos condensadores de 10uF.
- Un diodo LED.
- Un header de 3 pines.
- Un conector tipo bornera de 3 bornes.
- Un potenciómetro de precisión de 100KΩ.
- Una resistencia de 33KΩ.
- Una resistencia de 1KΩ.
- Una resistencia de 47KΩ.

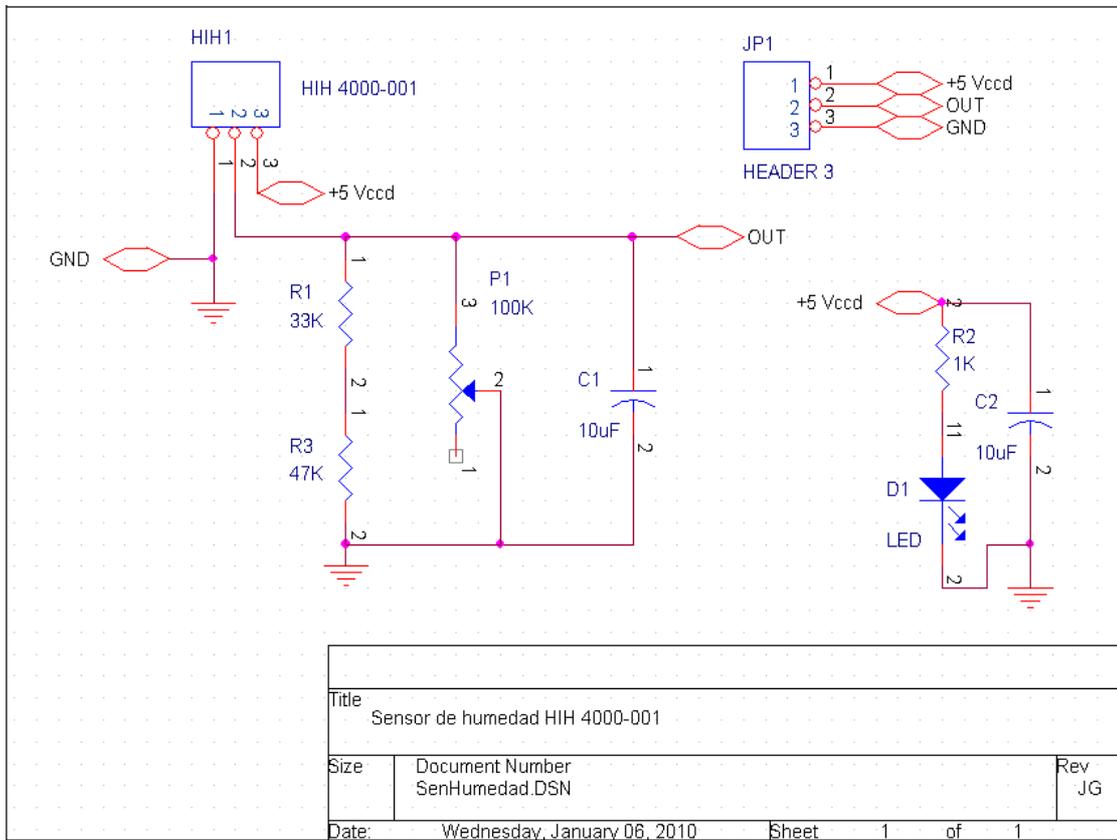


Figura 2.17: Diagrama lógico del circuito adaptador de señal del sensor de humedad.

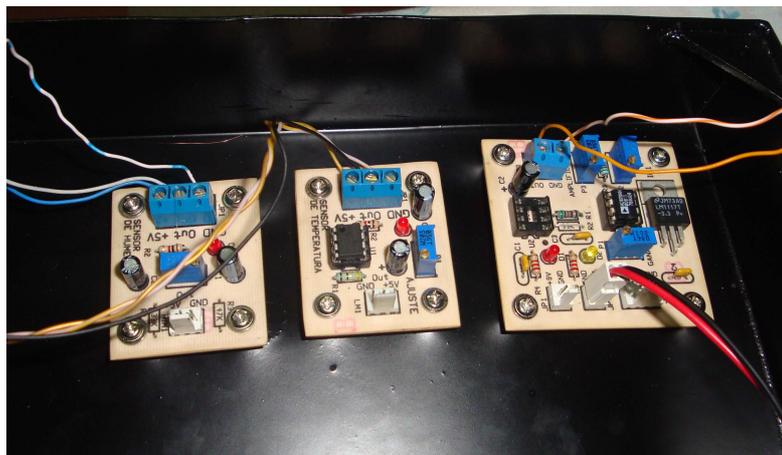


Figura 2.18: Componentes electrónicos de los circuitos adaptadores de señal.

2.5 DISEÑO DE LOS CIRCUITOS IMPRESOS DEL SISTEMA

2.5.1 CIRCUITO IMPRESO DEL PROCESADOR DE SEÑAL

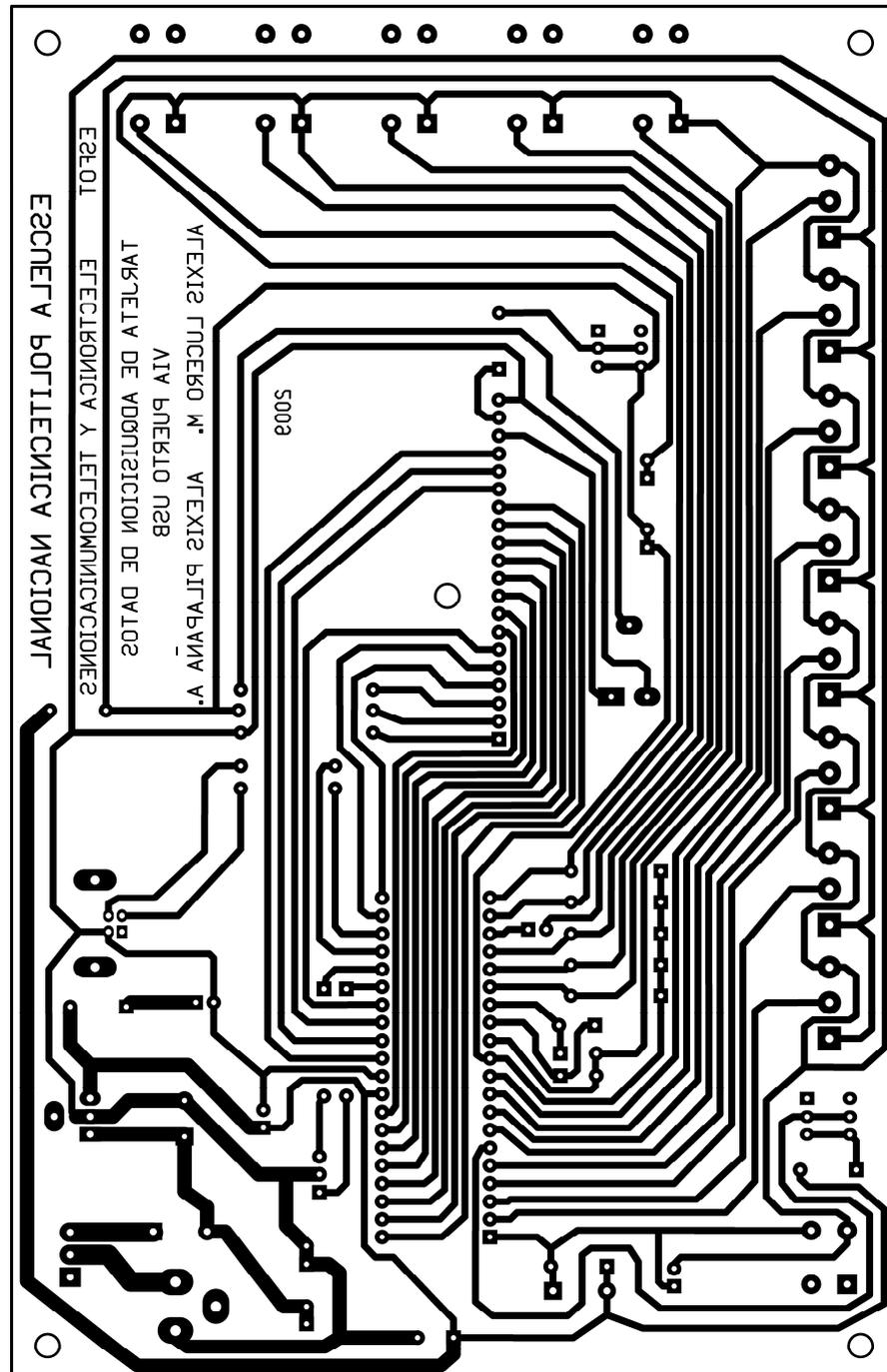


Figura 2.19: Pistas del circuito procesador de señal.

2.5.2 CIRCUITOS IMPRESOS DE LOS ADPTADORES DE SEÑAL

2.5.2.1 Circuito impreso del adaptador de señal del sensor de presión

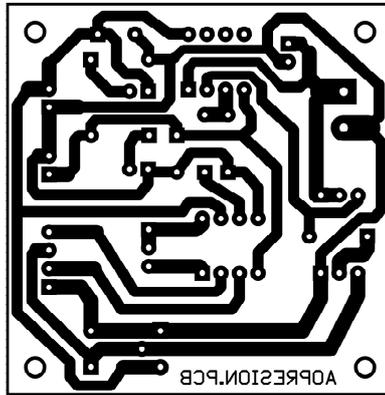


Figura 2.22: Pistas del circuito adaptador de señal del sensor de presión.

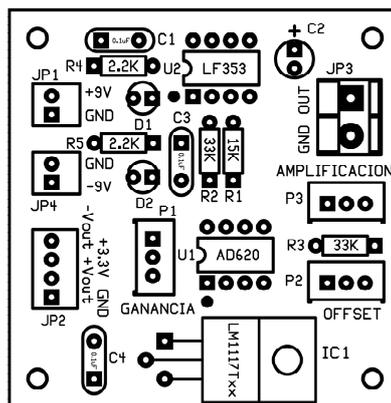


Figura 2.23: Distribución de elementos del circuito adaptador de señal del sensor de presión.

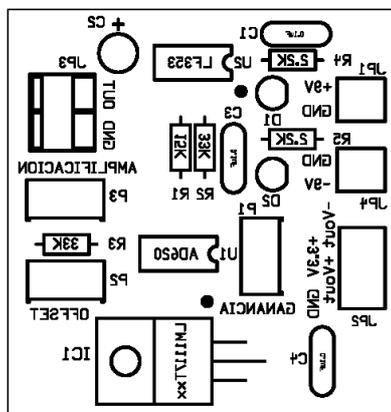


Figura 2.24: Screen de los elementos del circuito adaptador de señal del sensor de presión.

2.5.2.2 Circuito impreso del adaptador de señal del sensor de temperatura

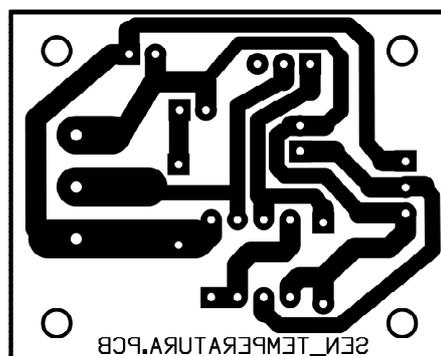


Figura 2.25: Pistas del circuito adaptador de señal del sensor de temperatura.

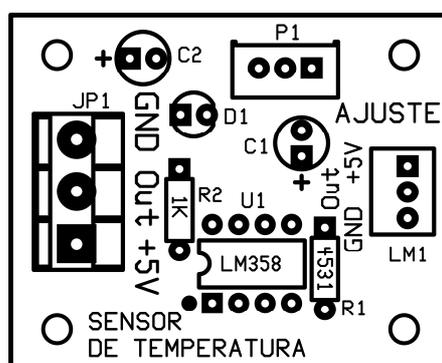


Figura 2.26: Distribución de elementos del circuito adaptador de señal del sensor de temperatura.

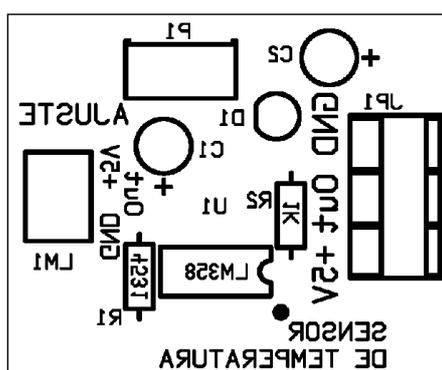


Figura 2.27: Screen de los elementos del circuito adaptador de señal del sensor de temperatura

2.5.2.3 Circuito impreso del adaptador de señal del sensor de humedad

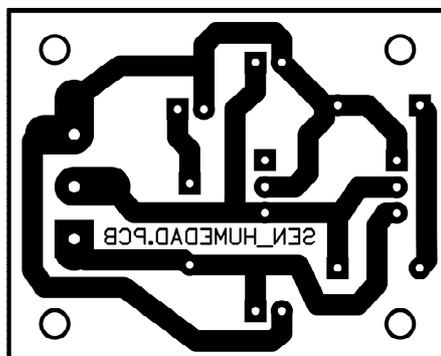


Figura 2.28: Pistas del circuito adaptador de señal del sensor de humedad.

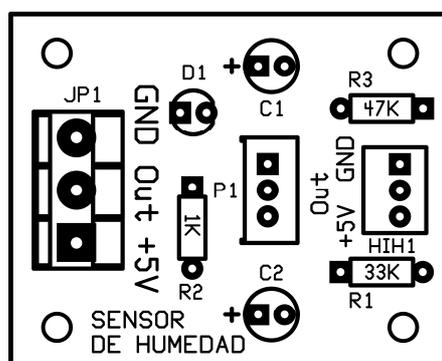


Figura 2.29: Distribución de elementos del circuito adaptador de señal del sensor de humedad.

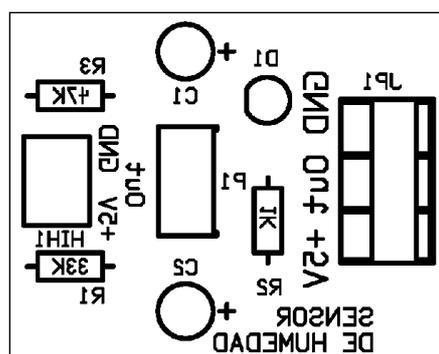


Figura 2.30: Screen de los elementos del circuito adaptador de señal del sensor de humedad.

2.6 PRINCIPIO DE FUNCIONAMIENTO

El sistema de adquisición de datos propuesto se compone de tres sensores que miden variables físicas tales como la presión, temperatura y humedad. Para cada uno de los anteriores se tiene un circuito adaptador de señal, que envían las mismas hacia el circuito procesador de señales, la cual después de realizar el procesamiento que le indique su software guardado en la memoria, envía estos datos hacia un GLCD para poder visualizar la información obtenida de la medición de variables físicas, o caso contrario, si el usuario así lo desea y con solo presionar un interruptor, puede visualizar los mismos datos del GLCD en una computadora con la ayuda de una interfaz grafica diseñada en Visual Basic.

En la Figura 2.31, se observa la conexión completa de todos los elementos antes mencionados que forman parte del sistema.



Figura 2.31: Conexión completa del sistema de adquisición de datos.

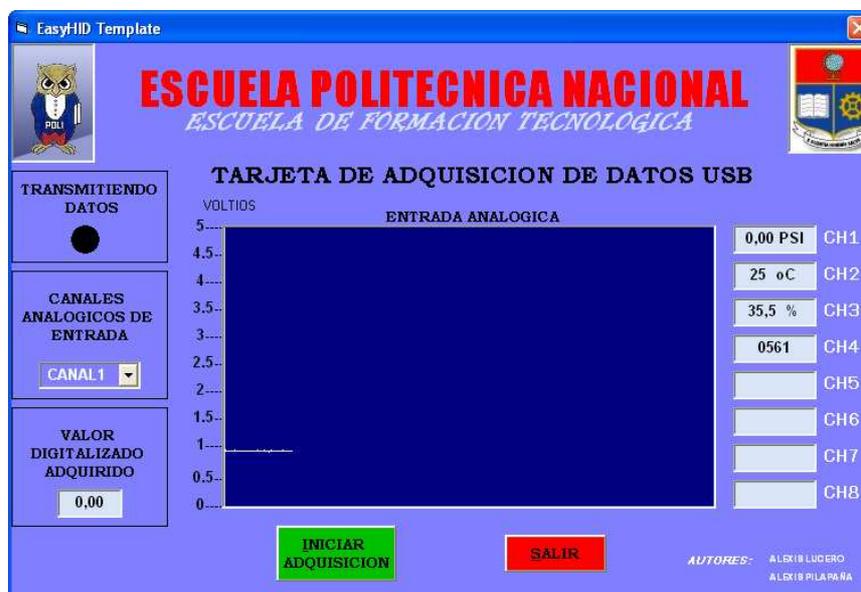


Figura 2.32: Interfaz graficas del sistema de adquisición de datos.

Una vez conectado todos los circuitos, comprobamos el correcto funcionamiento del mismo, es decir que debemos visualizar los valores de las variable físicas medidas en la pantalla del GLCD o en la pantalla de un computador (conectado al sistema mediante un cable USB), cualquiera de estas dos opciones dependerá tan solo de presionar o no el interruptor PC/AUTO que se encuentra en el circuito principal. Cuando visualizamos los datos en la pantalla de un computador, tenemos un plus adicional en la interfaz grafica, además de observar los datos de medición en números, también lo podemos hacer en forma gráfica (como un osciloscopio), lo cual nos indica la variación de voltaje en función del tiempo presente en las entradas analógicas del PIC 18F4550. Dichos voltajes son las señales acondicionadas provenientes de los sensores.

Además, tanto para la visualización en el GLCD como en la pantalla del computador, el sistema tiene funcionalidades de presentación de datos que son controlados con solo pulsar cualquiera de 5 pulsadores en un orden determinado.

Caso 1: para este caso utilizamos los pulsadores S1, S2 y S3.

S3: Al pulsarlo primero activa la función de poder elegir de cuantas de las ocho entradas analógicas puedo visualizar su respectivo dato digital procesado.

S1: Con cada pulso sirve para aumentar en una unidad el numero de entradas analógicas que deseo elegir.

S2: Con cada pulso sirve para disminuir en una unidad el numero de entradas analógicas que deseo elegir.

Luego de pulsar S1 o S2, debemos pulsar nuevamente S3 para poder hacer efectiva la acción de S1 o S2, es similar a un ENTER.

Caso 2: Para este caso utilizamos los pulsadores S4 y S5.

S5: Al pulsarlo primero activa la función de poder elegir de cual de las ocho entradas analógicas puedo visualizar su respectivo dato digital en toda la pantalla del GLCD O en la interfaz grafica.

S4: Con cada pulso sirve para elegir en orden ascendente cual entrada analógica se visualizara.

Luego de pulsar S4, debemos pulsar S5 nuevamente para poder hacer efectiva la acción de S4, es similar a un ENTER.

CAPITULO 3

3 DESARROLLO DEL SOFTWARE

3.1 INTRODUCCIÓN

Este capítulo tiene por objetivo indicar paso a paso el desarrollo del programa generado para el Pic 18F4550, como también el diseño del software para la presentación de la interfaz gráfica.

En nuestro proyecto, el programa que se encuentra grabado en el PIC 18F4550, está diseñado para trabajar básicamente de dos maneras. La primera forma de trabajo del sistema de adquisición de datos consiste en tomar las muestras desde las entradas analógicas, estas son digitalizadas mediante el conversor ADC interno del PIC, y se visualizan en un GLCD. La segunda manera se basa en la comunicación USB, en la cual se conecta el Hardware al computador por medio del puerto USB, se inicializa el archivo ejecutable generado en Visual Basic (Interfaz gráfica), aquí se puede seleccionar el canal analógico que se desea visualizar, estos datos se observarán tanto en forma gráfica como en su correspondiente valor decimal.

Para visualizar los datos en forma correcta se requiere de transductores, que son circuitos que se encargan de tomar los valores de las variables físicas (humedad, presión, temperatura, etc) y convertirlas en señales eléctricas, las cuales son entregadas al microcontrolador en forma de corriente o en forma de voltaje, éstas a su vez son digitalizadas y enviadas a los dispositivos de salida para poder visualizar, en nuestro caso los dispositivos de salida son el GLCD y la interfaz gráfica.

El sistema de adquisición de datos desarrollado en nuestro caso consiste de 8 entradas analógicas, de las cuales tres están programadas y probadas con variables físicas, estas variables son temperatura, humedad y presión.

Las variables físicas deben ser tratadas y estudiadas previamente, es decir, primero se debe diseñar el transductor para el correspondiente sensor, este a su vez debe ser calibrado y probado, luego se debe obtener varios puntos con la finalidad de obtener la ecuación correspondiente, la ecuación obtenida será VALOR DE LA VARIABLE FÍSICA vs. VALOR DE LA SEÑAL ELÉCTRICA (ya sea voltaje o corriente), una vez obtenido la ecuación se lo debe incluir de alguna manera en el programa del PIC, el cual leerá esa variable y lo presentará en los dispositivos de salida.

En este capítulo también se detallarán las herramientas computacionales utilizadas para cumplir con los objetivos.

3.2 DESCRIPCION DE LAS HERRAMIENTAS COMPUTACIONALES

En el mercado existen diversos software dedicados exclusivamente al diseño de aplicaciones basadas en microcontroladores, muchos de estos pueden ser descargados en forma gratuita desde Internet. Una de las ventajas que tienen estos programas es su bajo costo, por lo que está al alcance de todos y su costo al consumidor final resulta más cómodo, además de la versatilidad.

Los criterios de búsqueda de software para el sistema de adquisición de datos con comunicación al computador por medio de puerto usb se basa en los siguientes puntos:

- USB.- por el hecho de usar la comunicación USB entre Hardware y PC, se necesitó conseguir un software de programación para PICs tal que sea capaz de generar los archivos necesarios para dispositivos HID.
- GLCD.- en este caso se requirió además de un software, que, a parte del enunciado anterior (USB), permita de una manera fácil y amigable programar imágenes y datos para visualizar en un display gráfico de cristal líquido (GLCD).

- INTERFAZ GRÁFICA.- en este software se escribe el código para visualizar los datos analógicos en una interfaz gráfica en el computador.

Una vez analizado estos puntos importantes, se ha optado por utilizar los siguientes software de programación:

3.2.1 PROTON IDE

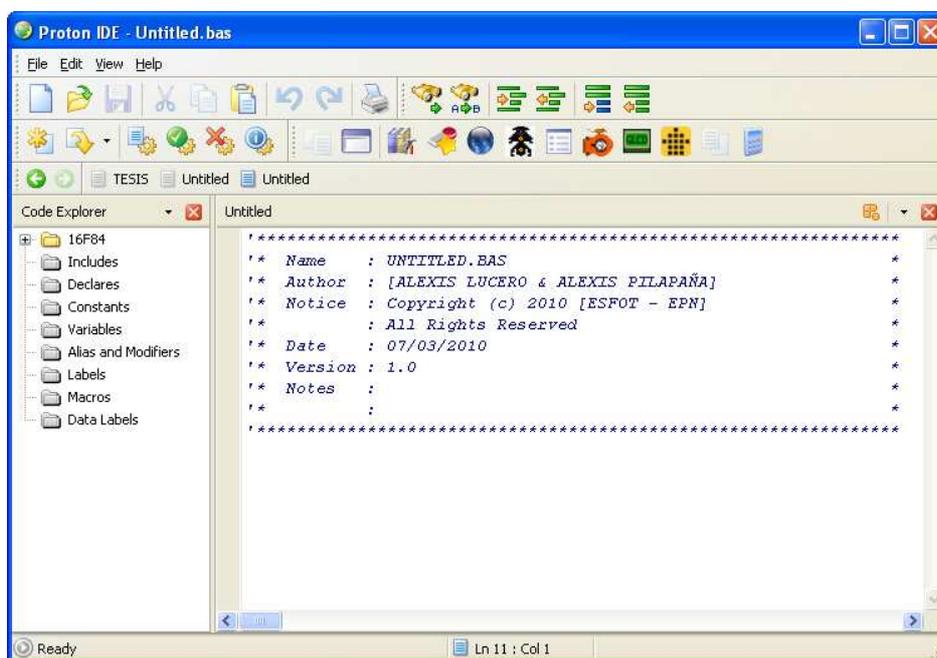


Figura 3.1: Ventana principal de proton ide

Este software permite escribir el código de programa en lenguaje BASIC, dentro de su librería incluye el PIC 18F4550, que es el que nos interesa para desarrollar el programa, por el hecho de incluir en su librería al PIC de nuestro interés, tiene la capacidad de generar los archivos necesarios para comunicar el hardware con el PC por medio del puerto USB.

Además, nos permite de una manera sencilla generar los códigos binarios requeridos de los gráficos previamente programados a ser visualizados en el GLCD.

Este software tiene su propio set de instrucciones, en el que se puede utilizar comandos conocidos comúnmente en el lenguaje BASIC.

3.2.2 VISUAL BASIC

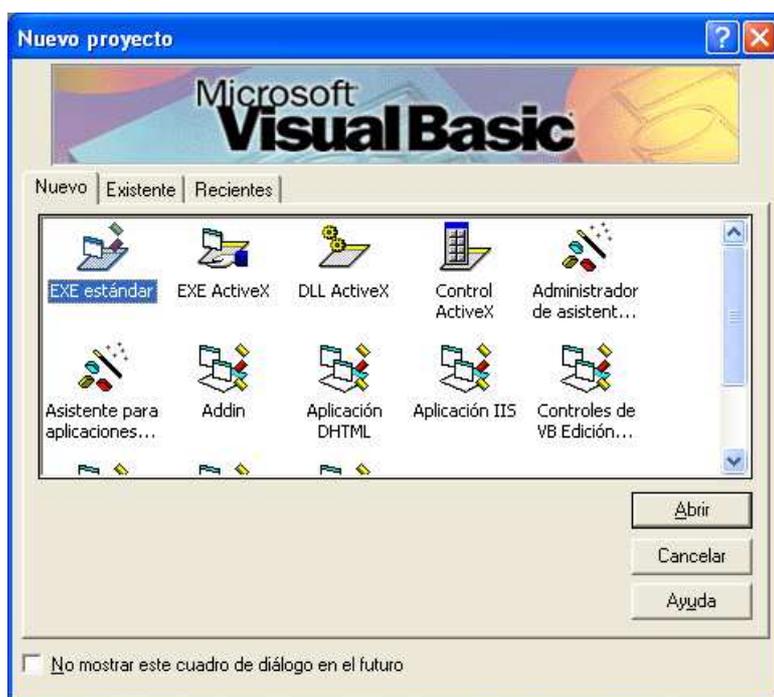


Figura 3.2: Ventana principal de Visual Basic 6.0

Este software es bien conocido y usado a nivel general para programación de aplicaciones para Windows. Se basa en el lenguaje Basic y en la programación orientada a objetos.

Es muy fácil de utilizar, una de las ventajas es que simplemente se arrastra los diversos objetos a la interfaz gráfica según la necesidad del usuario, estos objetos serán programados según sea el caso y finalmente serán presentados los datos de acuerdo a la programación de cada uno de estos objetos.

Finalmente se puede generar un archivo ejecutable, el cual puede ser grabado en diversas máquinas y ejecutado desde un simple acceso directo.

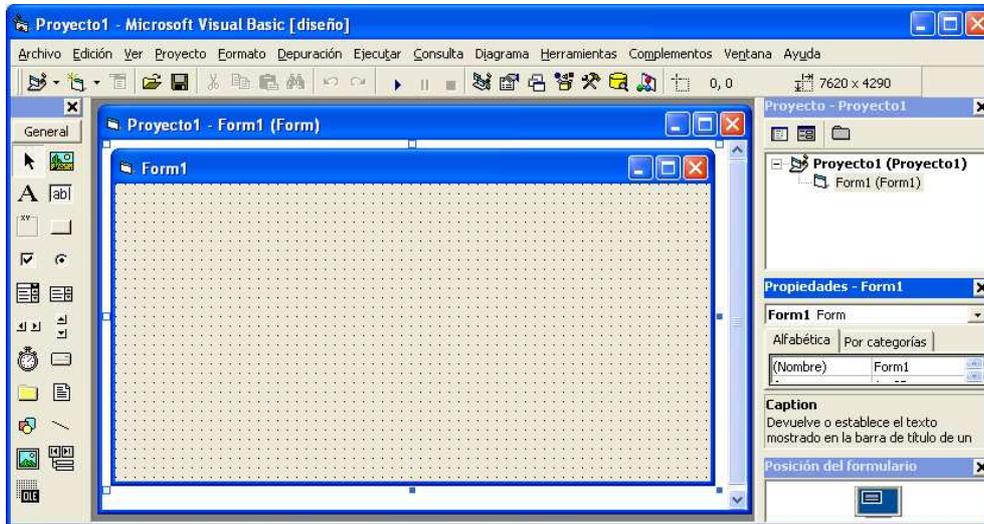


Figura 3.3: Forma inicial para comenzar a generar la interfaz gráfica.

3.2.3 WINPIC

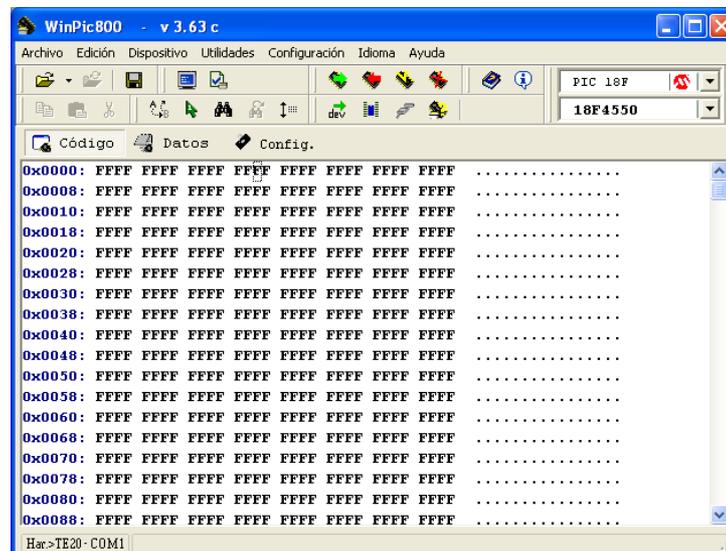


Figura 3.4: Ventana principal del programa Winpic 800

Este software nos ayuda para grabar el archivo .hex en la memoria del PIC 18F4550, razón por la que no puede trabajar sin antes conectar el grabador de pics con el PIC a trabajar.

La ventaja de este software es que genera automáticamente la configuración necesaria al escoger el modelo de PIC, se debe tomar en cuenta que este PIC utiliza el PLL interno, razón por la que se encuentra activada la casilla USBPLL.

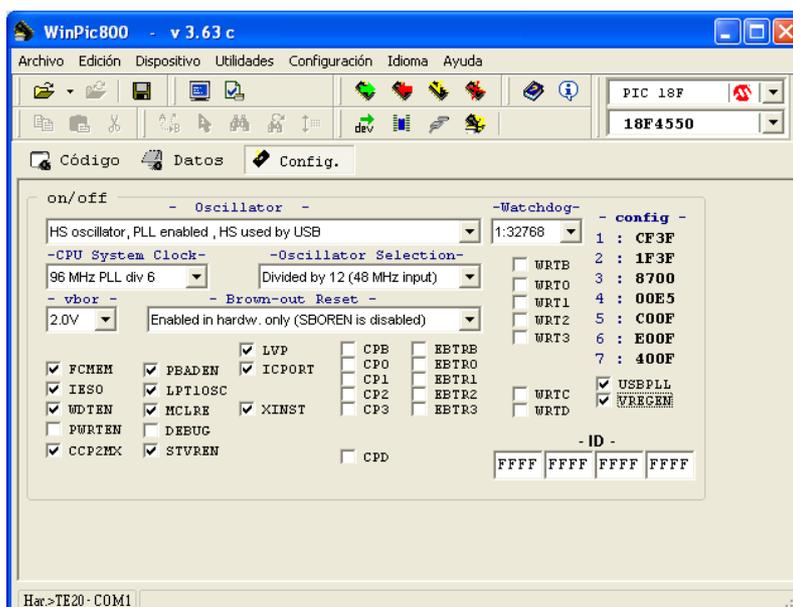


Figura 3.5: Configuración correcta para grabar el PIC18F4550

3.3 DESARROLLO DEL PROGRAMA PARA EL MICROCONTROLADOR PIC 18F4550

El código del programa está escrito en lenguaje Basic, este código es compilado por el programa proton ide, el cual genera el archivo .hex necesario para poder grabar en el PIC.

3.3.1 COMUNICACIÓN USB PIC – COMPUTADOR

Para lograr establecer una comunicación entre el PIC18F4550 y el computador, es necesario generar un código dentro del programa, este código es generado automáticamente siguiendo los siguientes pasos:

- Una vez abierto el programa proton IDE, ejecutamos el comando EasyHid USB Wizard, este nos permitirá generar el código requerido para establecer la comunicación USB entre el sistema de adquisición de datos y el PC

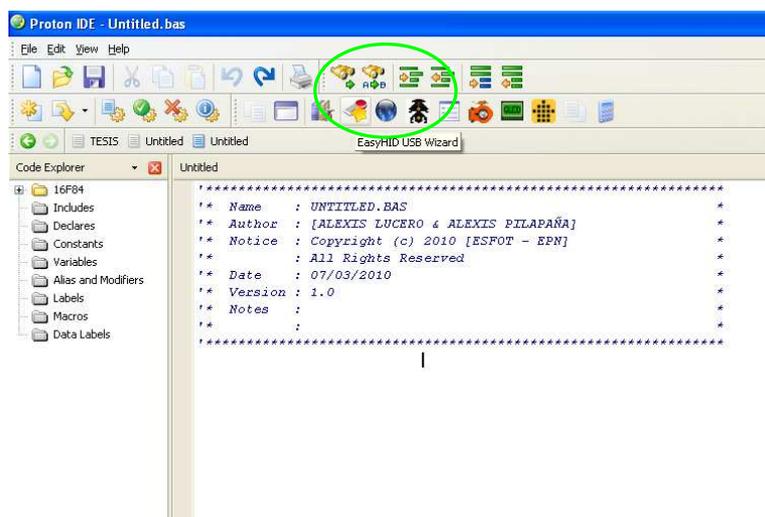


Figura 3.6: Comando utilizado para generar el código para la comunicación USB.

- Luego procedemos a dar el nombre con el que deseamos generar los diferentes archivos para establecer la comunicación USB.

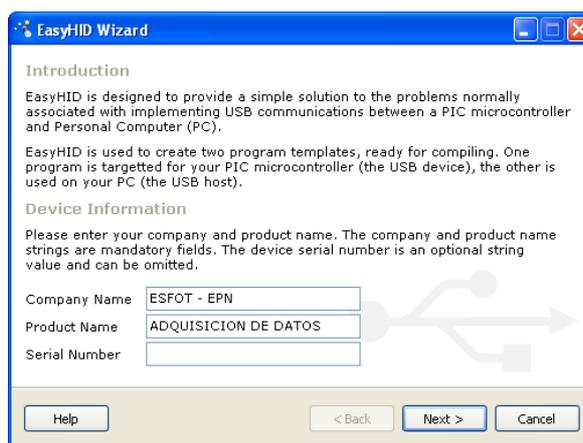


Figura 3.7: Nombres con los que se generan los archivos para establecer la comunicación USB.

- Luego escogemos la identificación del producto y del vendedor, estos valores tienen un rango, los cuales es preferible dejar en los valores por default.

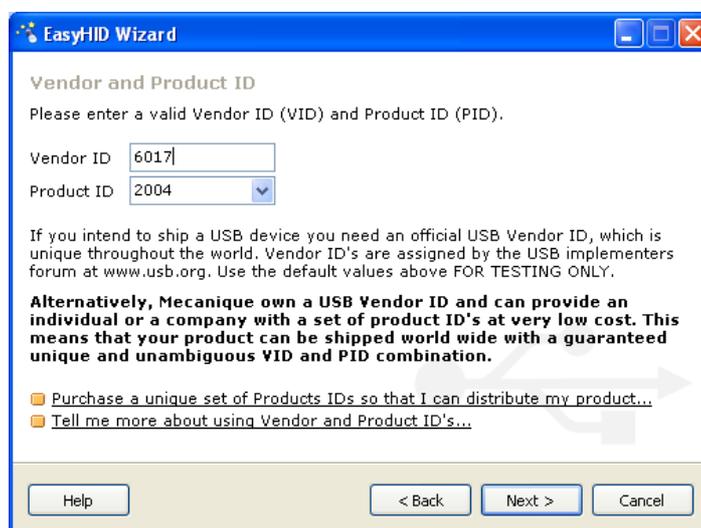


Figura 3.8: Identificación del producto y del vendedor

- En el siguiente paso se debe configurar los parámetros de tiempos requeridos, y los buffers de entrada y salida necesarios para la transmisión y recepción de datos. Estos parámetros se debe dejar en los valores por default.

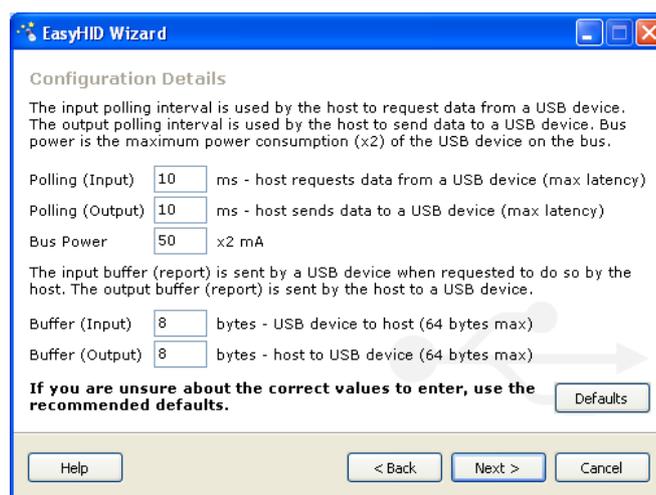


Figura 3.9: Configuración de parámetros para transmisión y recepción de datos

- En la siguiente ventana se debe ingresar el nombre del archivo y la ubicación donde se desea guardar. Además se debe escoger el modelo de PIC, el compilador donde se generará también la interfaz gráfica (en este caso se trabajó con Visual Basic). En esta ventana, algo que se debe poner énfasis es en la habilitación de la casilla para generar interrupciones USB, ya que sin estas interrupciones se tiene errores en la transmisión y recepción de datos.

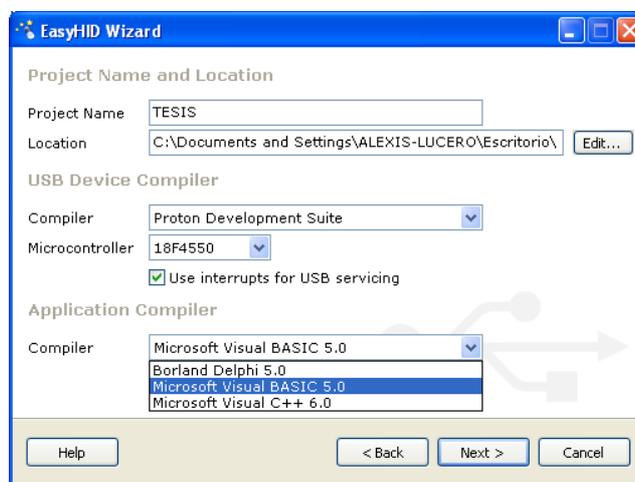


Figura 3.10: Elección del nombre y dirección de archivo, modelo del PIC y compilador

- Al dar click en siguiente, aparece una ventana, la cual se cerrará automáticamente si todo está bien configurado.

Al abrir la carpeta donde se guardó el archivo, aparecen dos carpetas, la una con el nombre PDS y la otra como VISUAL BASIC.

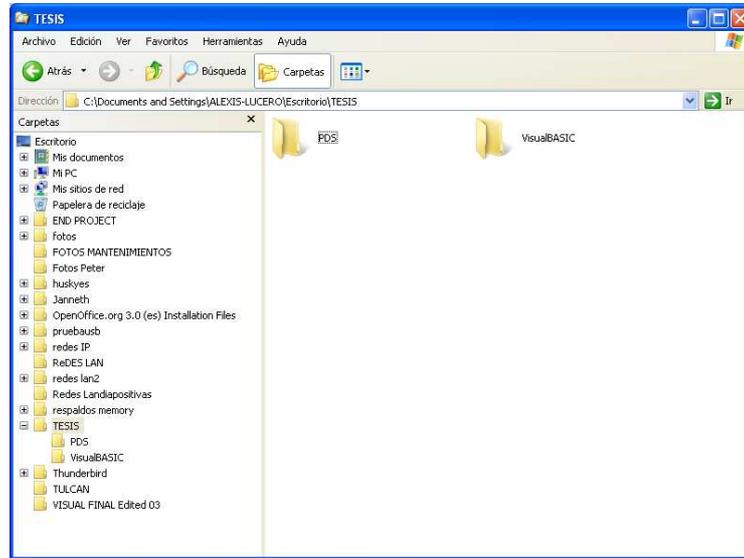


Figura 3.11: Carpetas generadas por EasyHid USB wizard

Abrimos la carpeta PDS y encontramos el archivo .BAS, este se abre con el programa proton y tenemos ya el código para establecer la comunicación USB, sobre este código empezamos a escribir el resto del programa para la lectura de datos.

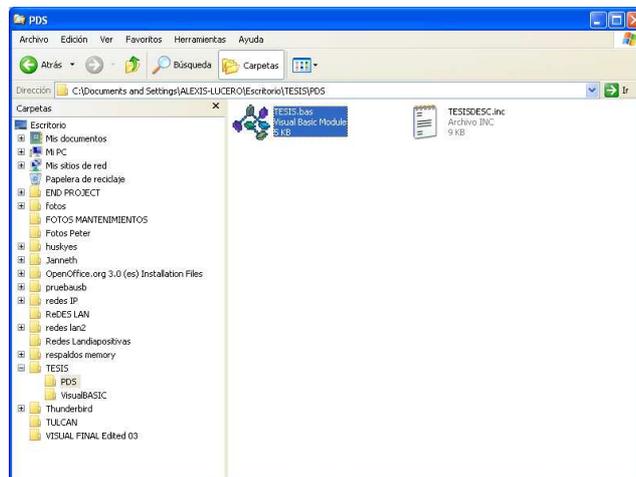


Figura 3.12: Archivo .bas donde se escribirá el código del programa en el software proton para el microcontrolador

Al abrir la carpeta Visual Basic, encontramos el archivo con extensión .vbp sobre este se escribirá el código para el diseño de la presentación de datos en la interfaz gráfica del computador.

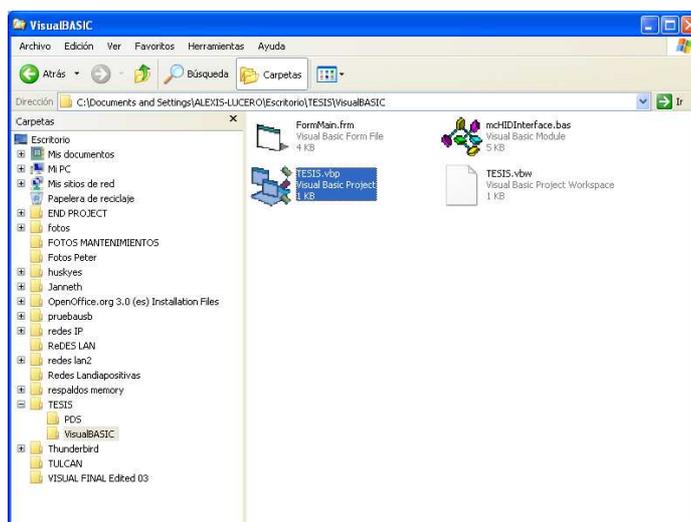


Figura 3.13: Archivo .vbp, donde se escribirá el código del programa en Visual Basic para la interfaz gráfica

3.3.2 DEFINICION DE PUERTOS

Para comenzar a programar, es necesario definir cuál de los puertos existentes en el PIC van a ser utilizados como entradas, y cuales van a ser salidas. Para ello se debe tomar en cuenta que debemos tener 8 entradas analógicas, 14 pines para el GLCD, 5 pines para control y selección de canales en el GLCD, un pin para seleccionar si se desea ver en el PC o solo en el GLCD, 2 pines para un led bicolor, el que indicará si está trabajando el computador o el GLCD.

Entonces se eligió los pines de la siguiente manera:

- **ENTRADAS ANALOGICAS.**- las entradas analógicas disponibles en el PIC 18F4550 son 8, es decir, que para nuestro proyecto se ha utilizado todas las entradas analógicas, estas son:

Puerto A.0
Puerto A.1
Puerto A.2
Puerto A.3
Puerto A.5
Puerto E.0
Puerto E.1
Puerto E.2

Por lo tanto estos puertos deben estar definidos en el programa como entradas, entonces se puede observar estas definiciones de la siguiente manera:

TRISA=%00111111

TRISE=%00000111

Se puede observar de derecha a izquierda que existen seis 1L en el caso del Puerto A y tres 1L en el caso de Puerto E, esto indica que desde el puerto A.0 hasta el Puerto A.5, y desde el Puerto E.0 hasta el Puerto E.2 se encuentran configurados como entradas, es decir todo el Puerto A y todo el Puerto E hemos configurado como entradas.

El Puerto A.4 también está definido como entrada, pero al observar el datasheet del PIC se aprecia que este cumple con otras funciones, razón por la que no es tomado en cuenta como entrada analógica.

- **PINES PARA EL GLCD.-** como se indicó anteriormente, el GLCD requiere de 14 pines del microcontrolador para su funcionamiento, de los cuales están configurados todo el Puerto B (8 pines), y desde el Puerto D2 hasta el puerto D7 (6 pines).

Estos puertos se encuentran configurados en nuestro programa de la siguiente manera:

TRISB=%00000000

TRISD=%00000011

El GLCD es un dispositivo de salida, razón por la que se puede observar que los 14 pódicos están definidos como salidas (0L).

El Puerto D.0 y D.1 están definidos como entradas ya que estos cumplen otras funciones.

- **PINES PARA CONTROL DE LOS CANALES.-** estos pines están definidos como entradas, ya que se conectarán 5 pulsadores para seleccionar que canal y cuales de las 8 entradas existentes deseo observar en el GLCD.

Los puertos que se han elegido para este efecto son:

Puerto C.0

Puerto C.1

Puerto C.2

Puerto D.0

Puerto D.1

Los cuales se encuentran configurados en nuestro programa de la siguiente manera:

TRISC=%00000111

TRISD=%00000011

Donde se puede observar los puertos enunciados configurados como salidas (1L).

- **OTROS PINES.-** dentro del programa existen otros puertos utilizados los cuales son de menor importancia que los anteriores pero cumplen ciertas funciones para un óptimo funcionamiento de nuestro sistema. Estos se detallan a continuación:

Puerto A.4-----TRISA=%00111111

Se encuentra definido como entrada, se conecta un pulsador que será manejado por el usuario para seleccionar si desea observar los datos en el GLCD o en el computador.

Puerto C.6 y C.7----- TRISC=%00000111

Se encuentra definido como salida, se conecta un Led bicolor que indicará si está trabajando el GLCD o el Computador.

3.3.3 DEFINICIONES PARA EL GLCD

Las definiciones necesarias para visualizar los datos en el GLCD se pueden observar en la Fig. 3.3 – 9.

Aquí se puede apreciar que los datos se van a enviar al GLCD a través de los puertos B y D, como ya se indicó en el subtema definición de puertos.

Además existen 2 declaraciones:

Declare GLCD_CS_INVERT On. - Esta instrucción es necesaria para visualizar los datos ya que el GLCD tiene las entradas CS (Chip Select) invertidas, caso contrario no se puede visualizar ningún dato.

Declare INTERNAL_FONT On. - Esta instrucción indica que se usará el tipo de fuente interna existente para imprimir los datos en el display.

```

TRISC=00000111
' *****
' * Definiciones para el GLCD
' *****
TRISD.7=0           'Pin del Reset del GLCD
PORTD.7=1           'Estado inicial IL
Declare LCD_TYPE 1
LCD_DTPORT = PORTB
LCD_RSPIN = PORTD.5
LCD_ENPIN = PORTD.2
LCD_RWPIN = PORTD.3
LCD_CS1PIN = PORTD.4
LCD_CS2PIN = PORTD.6
Declare GLCD_CS_INVERT On
Declare INTERNAL_FONT On

```

Figura 3.14: Definiciones para visualizar los datos en el GLCD de nuestro sistema

3.3.4 DEFINICIONES DE ALIAS

Los Alias son utilizados para facilitar la programación ya que a lo largo del programa se utilizará repetidas veces un mismo p rtico en diferentes etapas de programaci n, es aqu  donde se le da un sobrenombre al p rtico.

Ejm: Si un pulsador est  conectado al puerto A.0, a  ste le damos el sobrenombre BOTON 1, y se escribir  en el programa de la siguiente manera:

Symbol BOTON 1 = PORTA.0

Con esto, cuando se desee programar el pulsador, se escribir  BOTON 1 en lugar de PORTA.0, esto hace m s f cil la programaci n.

En el presente proyecto se tiene definido los sobrenombres o Alias a los siguientes elementos:

- Pulsadores para control de los canales a visualizar en el GLCD (Tecla1 a Tecla5)
- Pulsador para elegir si se desea observar los datos en el GLCD o en el PC (Pcauto)
- Led Bicolor (Led rojo y Led Verde)

```
' * Definiciones para los ALIAS
' *****
Symbol Tecla1 = PORTD.1
Symbol Tecla2 = PORTD.0
Symbol Tecla3 = PORTC.2
Symbol Tecla4 = PORTC.1
Symbol Tecla5 = PORTC.0
Symbol Pcauto = PORTA.4
Symbol LedRojo = PORTC.6
Symbol LedVerde = PORTC.7
' *****
```

Figura 3.15: Definiciones de Alias en el sistema de Adquisici n de datos

3.3.5 DEFINICIONES DEL CONVERTOR A/D

Para configurar el Conversor Analógico / Digital, se debe tener en cuenta cuántas entradas analógicas se van a utilizar en el programa, con cuántos bits se va a realizar la conversión analógica a digital, la fuente de reloj, el tiempo de muestreo de la señal.

```
' *****
' * Definiciones para Conversor Analogico Digital *
' *****
ADIN_RES    10           'Resolucion del ADC 10 bits
ADIN_TAD    FRC         'Elige RC oscilador de muestreo para ADC
ADIN_STIME  100        'Permite 100us para tiempo de Carga
ADCON1 = %00000111     'Setea PORTA Y PORTE como analogicos con justificacion del resultado a la derecha
ADCON2.7=1
' *****
```

Figura 3.16: Definiciones para el conversor ADC

En el programa se define entonces el puerto A y el puerto E como entradas analógicas, con una resolución del muestreo de 10 bits, se elige una fuente de reloj interna configurada a 100us para tiempo de carga.

3.3.6 PRESENTACION DE IMÁGENES EN EL GLCD

Para lograr visualizar una imagen en el GLCD, es necesario primero diseñar imágenes en blanco y negro de 128 pixeles de ancho por 64 pixeles de alto, este sería el tamaño máximo de las imágenes que se desea presentar.

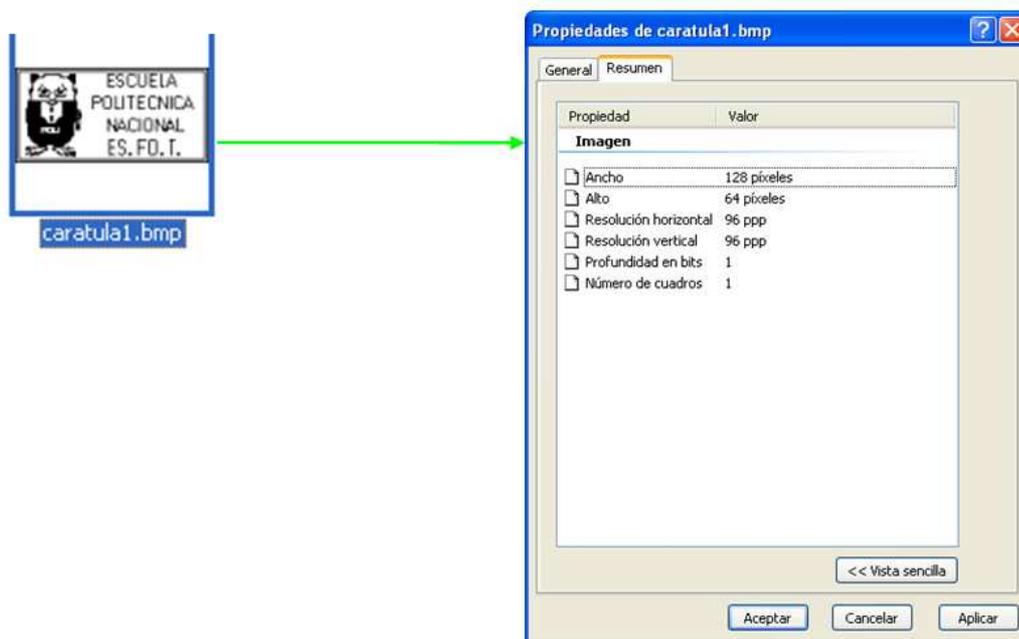


Figura 3.17: Tamaño de imágenes a grabar a visualizar en el GLCD

Se obtiene un código el cual será visualizado en el display, este código es generado en el software Proton siguiendo los siguientes pasos:

- Abrir el software Proton
- Aquí encontramos un icono de la siguiente forma . Damos click sobre este icono.
- En la ventana que se abre elegimos la imagen que ha sido previamente guardada para visualizar en el GLCD. Figura 3.17.



Figura 3.18: Imagen que se visualizará en el GLCD

Si la figura seleccionada no cumple con las características requeridas el programa rechazará la imagen, por lo que no se podrá obtener código alguno.

- Damos clic en el botón code para visualizar el código generado de acuerdo a la imagen seleccionada. (Ver figura 3.3 – 13)

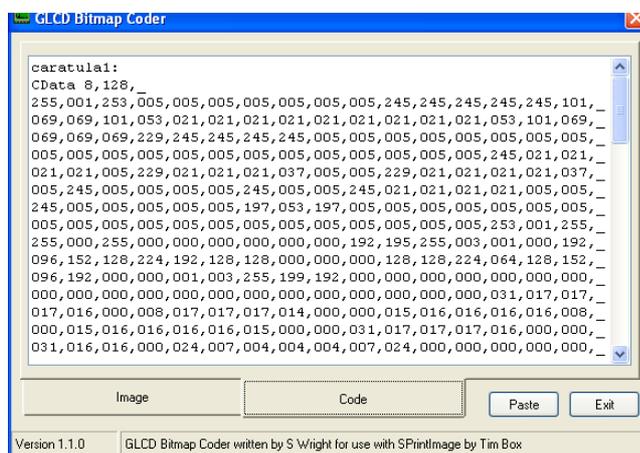


Figura 3.19: Código generado de acuerdo a la imagen seleccionada

Este código se debe copiar en un archivo de texto (Bloc de notas), con la extensión .inc y guardar en la carpeta PDS generada anteriormente en el subcapítulo 3.3.1.

Este archivo se debe incluir en el programa de la siguiente manera:

Include "CARATULA1.inc"

Luego de haber incluido el archivo dentro del programa, se debe escribir las correspondientes líneas de código para presentar la imagen.

A continuación detallamos las instrucciones para presentar una imagen en el GLCD, está será visualizada por un tiempo de 4s para este caso.

```

Y = 0
  PUNTERO = 0
  Cls
  Repeat
    Repeat
      DATO = CRead caratula1 + PUNTERO
      LCDWrite Y,X,[DATO]
      Inc X
      Inc PUNTERO
    Until X = 128
  X = 0
  Inc Y
  Until Y = 8
  DelayMS 4000

```

3.3.7 PRESENTACIÓN DE DATOS EN EL GLCD

Para visualizar datos provenientes de variables físicas como temperatura, resistencia, etc. Se debe seguir un minucioso proceso, este consiste en la consecución del sensor, acoplamiento del mismo mediante un transductor, finalmente este será el que envíe los datos analógicos a los pórnicos del PIC, el cual se encargará de convertirlos a datos digitales y finalmente estos podrán ser visualizados en un dispositivo de salida de acuerdo a una programación previa.

La señal analógica debe ser tratada con el fin de obtener una curva característica de la magnitud física en función de una magnitud eléctrica, debemos obtener una ecuación de esta curva para poder ingresar al programa.

3.3.7.1 Diseño del programa para el sensor de temperatura

Una vez adquirido físicamente el sensor de temperatura, y luego de revisar las características técnicas del mismo, se conoce como dato para motivos de cálculo que este sensor entrega 10mV por cada °C esto nos condujo a diseñar un amplificador con una ganancia tal que duplique el voltaje de entrada, con esto, al dividir este valor para 4 obtenemos directamente el valor de la temperatura sin necesidad de recurrir a cálculos matemáticos complejos.

Ejm: si tenemos una temperatura ambiente de 25° C, esto quiere decir que a la salida del sensor obtenemos 250 mV, como la señal esta amplificada con una ganancia al doble que la entrada tendremos 500 mV si realizamos una regla de tres obtenemos lo siguiente:

5V-----1024 bits

0.5V-----X

$X = 102,4$ bits

Si este valor dividimos para 4 obtendremos otro bastante aproximado al dato adquirido de la variable física, en este caso, tenemos 25.6 que es un valor bastante cercano a 25° C, este es el valor que se visualizará en los dispositivos de salida. Para corregir este error, tenemos en el circuito transductor un potenciómetro de precisión que regula la ganancia de voltaje, la variación será mínima.

Por lo tanto, la ecuación para este caso será:

Dato a visualizar = Dato leído / 4

3.3.7.2 Diseño del programa para el sensor de humedad

Luego de adquirir nuestro sensor de humedad, se procede a acoplarlo por medio del circuito transductor, este incluye únicamente un potenciómetro que sirve para calibrar el sensor.

Como es de conocimiento general, los datos digitales trabajan en código binario, es decir, con 0L y 1L, el siguiente paso es obtener el valor del voltaje en su correspondiente valor binario, para esto, sabemos que el muestreo de la señal es a 10 bits, pues así se lo ha definido al inicio del programa para mayor precisión, de lo que podemos sacar mediante una regla de tres simple el siguiente cálculo:

¿Si 5V es equivalente a 1024, X voltios (valor de voltaje) a cuántos bits equivalen?

5V-----1024

XV-----?

| % humedad | Voltaje | Valor en binario |
|-----------|---------|------------------|
| 32 | 2,60 | 532,48 |
| 37 | 2,80 | 573,44 |
| 41,80 | 2,93 | 600,064 |
| 42,10 | 2,94 | 602,112 |
| 42,90 | 2,97 | 608,256 |
| 46,80 | 3,12 | 638,976 |
| 55 | 3,60 | 737,28 |
| 62,02 | 4,05 | 829,44 |
| 67,40 | 4,10 | 839,44 |

Tabla 3.1: Datos obtenidos para la curva humedad Vs Voltaje y su equivalente en binario.

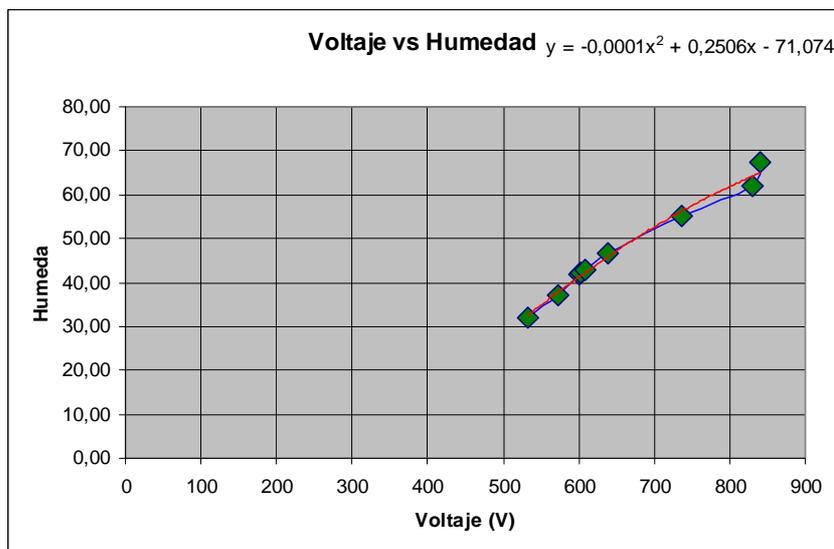


Figura 3.20: Curva de humedad en función del voltaje

La ecuación obtenida, como se puede apreciar en la Fig. 3.3 -14, es la que está ingresada en las líneas de la programación del PIC para poder presentar en forma digital en los dispositivos de salida.

3.3.7.3 Desarrollo del programa para medición de presión

Al igual que en los sensores descritos anteriormente, se debe acoplar este sensor a un circuito amplificador, una vez amplificada la señal se obtiene una tabla de valores de forma similar al sensor de humedad, la tabla obtenida es la siguiente:

| Presión (PSI) | Voltaje-pin6 del AD620AN (V) | Voltaje en binario |
|---------------|------------------------------|--------------------|
| 0,0000 | 1,00 | 204,8 |
| 0,1813 | 1,25 | 256 |
| 0,3625 | 1,50 | 307,2 |
| 0,5437 | 1,74 | 356,352 |
| 0,7250 | 1,98 | 405,504 |
| 0,9060 | 2,25 | 460,8 |
| 1,0870 | 2,49 | 509,952 |
| 1,2680 | 2,74 | 561,152 |
| 1,4500 | 3,00 | 614,4 |

Tabla 3.21: Datos obtenidos para la curva Presión Vs Voltaje y su equivalente en binario

De esta tabla obtenemos la gráfica de la Figura 3.3 -15 y obtenemos la ecuación respectiva.

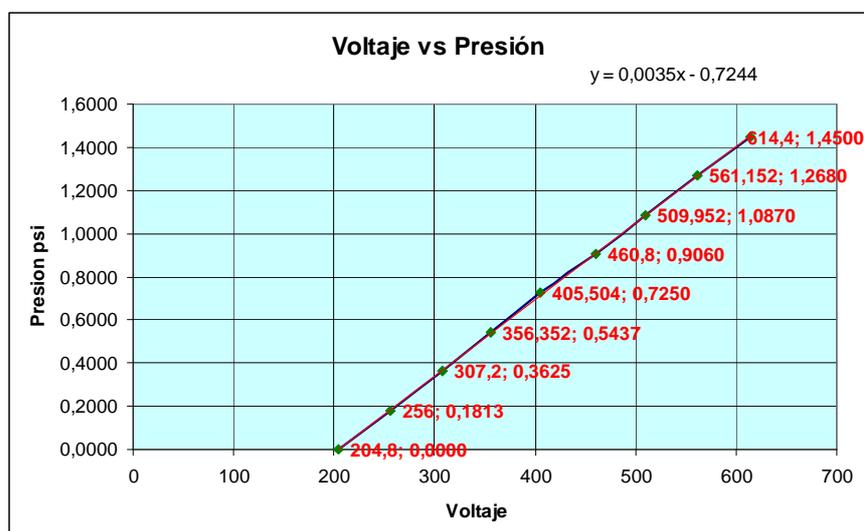


Figura 3.22: Curva de presión en función del voltaje y su respectiva ecuación

3.4 DESARROLLO DEL SOFTWARE PARA LA INTERFAZ GRÁFICA

Otro dispositivo de salida en el sistema de adquisición de datos es la interfaz gráfica, está desarrollada en el software Visual Basic. En esta interfaz gráfica se puede observar el voltaje en forma gráfica y su correspondiente valor de la variable física.

El programa está diseñado para indicar una variable a la vez según sea seleccionada por el usuario.

3.4.1 PROGRAMA PARA LA INTERFAZ GRÁFICA

3.4.1.1 Código para comunicación Usb Pic - Pc

En el capítulo 3.3 se generaron dos carpetas, la una con el nombre PDS y la otra llamada Visual Basic.

Dentro de la carpeta Visual Basic se encuentra un archivo con la extensión .vbp (Ver Fig. 3.3 – 8), abrimos este archivo y encontramos un código que fue generado por EasyHid Usb Wizard, éste es el código para comunicar al PIC con el PC, sobre este código empiezo a escribir las líneas de programación para la interfaz gráfica.

3.4.1.2 Código para lectura de datos desde el Pic.

Una vez abierto el archivo TESIS .vbp se observa que en la esquina superior derecha existen dos ventanas, la una ventana contiene el código de la interfaz gráfica y la otra la forma en la que será presentada al usuario.

En el código están definidas las variables, los buffers de entrada y de salida, la programación de los diferentes objetos existentes en la interfaz gráfica y como se va a visualizar los datos en forma gráfica y numérica.

3.4.2 DISEÑO DE VISUALIZACIÓN DE LA INTERFAZ

Al dar click en el botón ver objeto de la parte superior derecha ingresamos a la programación de la interfaz gráfica. Esta programación es sencilla ya que esta orientada a objetos, es decir, se puede arrastrar diferentes objetos como botones de comando, imágenes, cuadros de texto, etc, a la ventana principal. Ver figura 3.4 -1

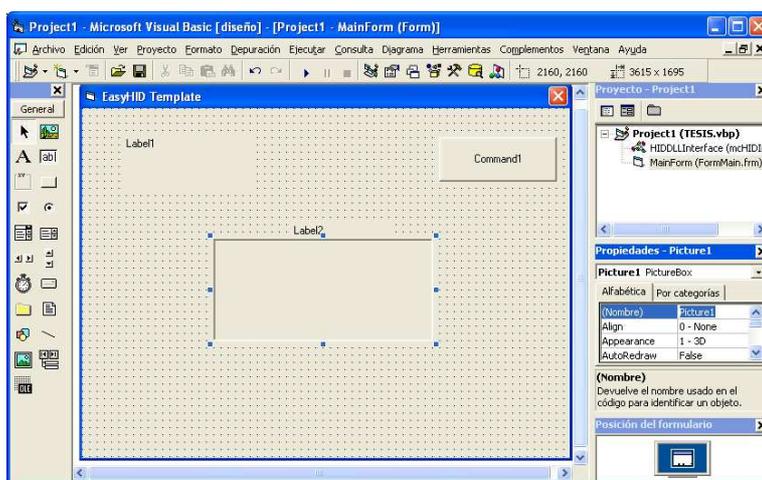


Figura 3.23: Ventana donde se va generando la presentación de la interfaz gráfica

Estos botones pueden ser programados de acuerdo a las necesidades del usuario con solo dar clic sobre el objeto: Ver figura 3.4 -2

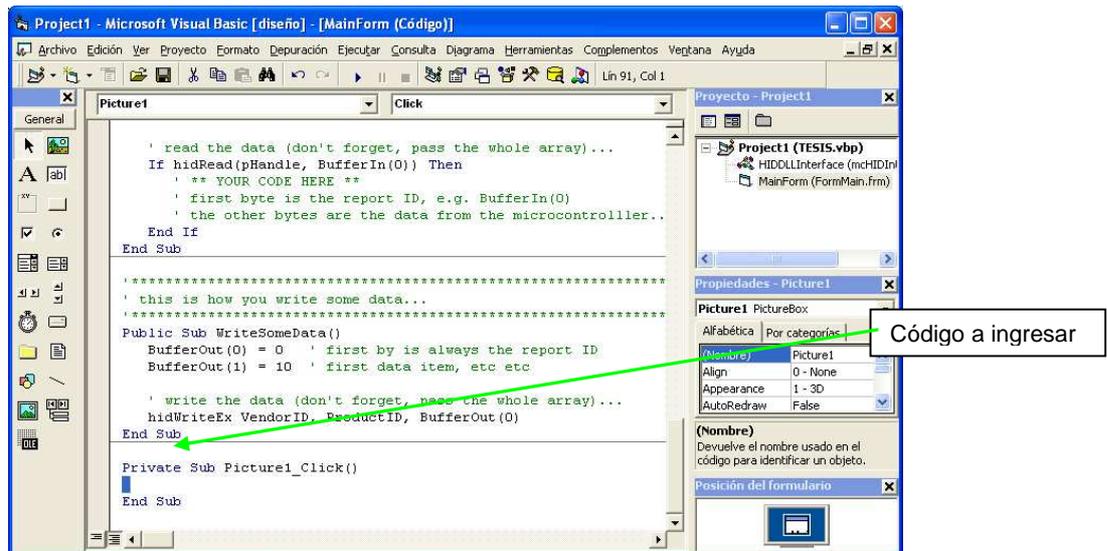


Figura 3.24: Rutina generada al dar doble clic en el botón Picture 1

3.4.3 GENERACIÓN DE ARCHIVO EJECUTABLE

Cuando ya se ha terminado de diseñar la presentación de la interfaz gráfica, damos clic en el botón iniciar, si todo el programa, tanto en el código como en el objeto está bien, no aparece ningún mensaje de error, caso contrario, indica un mensaje de error que hay que corregir para que el programa pueda correr.

Una vez que no tenemos errores en el programa procedemos a generar un archivo ejecutable, procedemos de la siguiente manera:

- Luego de guardar el archivo, nos ubicamos en la opción Generar proyecto .exe del menú archivo (Figura 3.25).

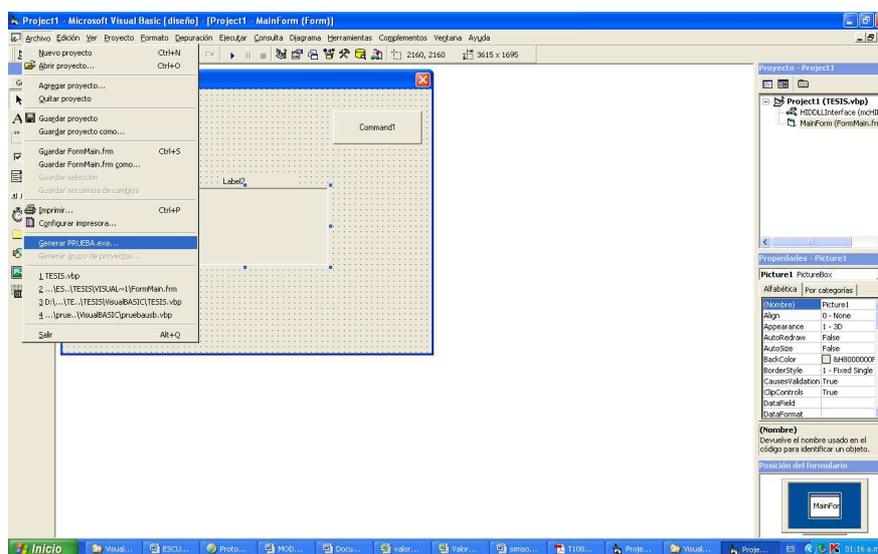


Figura 3.25: Opción para generar el archivo ejecutable

- Guardamos en alguna ubicación deseada, y ya está listo el archivo ejecutable.

Este archivo es de vital importancia ya que si no se genera el ejecutable, la interfaz gráfica se podrá abrir solo en las máquinas que tengan instalados Visual Basic. El ejecutable puede correr en cualquier máquina sin necesidad de grabar el programa y podrá trabajar el Sistema de Adquisición de datos sin ningún inconveniente.

3.4.4 PRESENTACION FINAL DE LA INTERFAZ GRÁFICA



Figura 3.26: Presentación de la interfaz gráfica

3.4.5 DESCRIPCION BREVE DE LAS FUNCIONES DE LA INTERFAZ GRAFICA

Al dar clic en el botón iniciar adquisición, empieza a graficar el voltaje de la entrada analógica seleccionada, estas se pueden elegir desde el canal 1 hasta el canal 8 (una a la vez), a su vez indica el valor de la variable física en la casilla de valor digitalizado adquirido.

En la parte lateral derecha se pueden visualizar el valor de las 8 variables físicas existentes.

CAPITULO 4

4 PRUEBAS Y RESULTADOS

El objetivo de este subcapítulo es comprobar el correcto funcionamiento de los sensores, como se visualizan los datos en los dispositivos de salida (Interfaz gráfica y GLCD) y el comportamiento del sistema cuando existen cambios de niveles en las entradas analógicas.

4.1 PRUEBAS CON EL SENSOR DE PRESION

De acuerdo a la programación de nuestro sistema, encontramos que el canal 1 está diseñado para conectar el sensor de presión. Las pruebas realizadas son las siguientes:

- PRUEBA DE VISUALIZACIÓN EN EL GLCD

En esta prueba se puede observar sin ningún problema el dato tomado por el sensor. En la fotografía se puede observar un poco distorsionado el dato ya que los decimales no están fijos por las variaciones existentes de presión.



Figura 4.1: Visualización de presión en el GLCD.

- PRUEBAS CON LA INTERFAZ GRAFICA

Se puede visualizar el valor de presión en el canal 1 y la curva de voltaje obtenida de acuerdo a los cambios de presión realizados.



Figura 4.2: Visualización de presión en la interfaz gráfica

4.2 PRUEBAS CON EL SENSOR DE TEMPERATURA

- PRUEBAS CON EL GLCD

Para este caso se verifica el dato de temperatura obtenido y se compara con otro equipo, de esta comparación se puede concluir que nuestro sistema de adquisición de datos está trabajando en forma correcta.

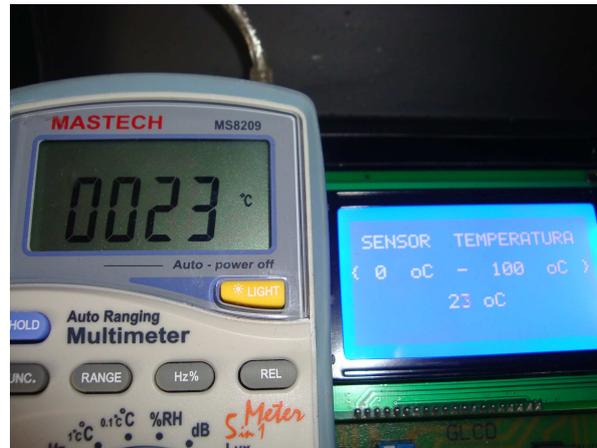


Figura 4.3: Visualización de temperatura en el GLCD.

- PRUEBAS CON LA INTERFAZ GRAFICA

Para esta prueba se varía la temperatura, y se puede visualizar los datos tanto de temperatura como la curva de voltaje. Este dato se visualiza en el canal 2 ya que el sistema está programado para que se visualice por este canal.



Figura 4.4: Visualización de temperatura en la interfaz gráfica

4.3 PRUEBAS CON EL SENSOR DE HUMEDAD

- PRUEBAS EN EL GLCD

Al comparar el valor obtenido con otro equipo de medición, podemos observar que los valores obtenidos son muy parecidos, el porcentaje de error es mínimo. De lo que podemos concluir que nuestro equipo está procesando los datos en forma correcta.

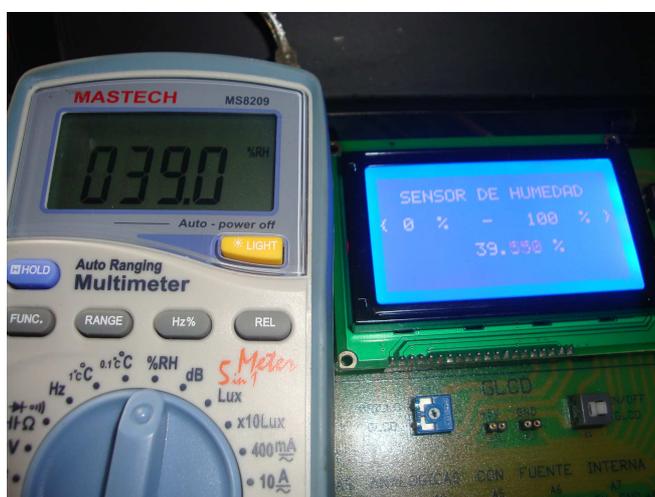


Figura 4.5: Visualización del valor de humedad en el GLCD

- PRUEBAS CON LA INTERFAZ GRAFICA

En el canal 3 podemos visualizar el valor obtenido del porcentaje de humedad relativa y en la ventana gráfica se puede visualizar una curva de variación de voltaje de acuerdo a la variación de humedad.



Figura 4.6: Visualización del valor de humedad en la interfaz gráfica.

4.4 PRUEBAS CON EL POTENCIOMETRO

En esta prueba se puede observar como varía el dato obtenido en el canal 4. Este valor no está en ohmios sino en bits. Se observa también la curva de variación al girar el potenciómetro.



Figura 4.7: Dato obtenido al variar el potenciómetro.

4.5 COSTO DEL PROYECTO

La tabla 4.1 generada a continuación, indica el costo de cada uno de los elementos utilizados en el proyecto, y el costo final de todo el sistema de adquisición de datos.

Los costos en este estudio están basados en los precios reales generados al adquirir los diferentes elementos.

El costo de los elementos detallados en esta tabla puede variar según la tienda donde se adquiera los mismos.

Tabla 4.1: Costos de los elementos electrónicos utilizados en el proyecto

| Ítem | Elementos | Cantidad | C. Unitario | C. Total |
|------|---|----------|-------------|----------|
| 1 | Baquelita 10x20 | 1 | 3.00 | 3.00 |
| 2 | Acido sulfúrico | 2 | 0.50 | 1.00 |
| 3 | Papel para circuitos impresos | 1 | 2.50 | 2.50 |
| 4 | Condensador de 10uF / 25V. | 7 | 0.20 | 1.40 |
| 5 | Condensador de 100uF / 25V. | 1 | 0.20 | 0.20 |
| 6 | Condensador de 0.21uF. | 1 | 0.20 | 0.20 |
| 7 | Condensador de 22pF. | 2 | 0.20 | 0.40 |
| 8 | Condensador de 47uF / 25V. | 1 | 0.20 | 0.20 |
| 9 | Condensador de 0.1uF. | 3 | 0.20 | 0.60 |
| 10 | Resistencia de 1K Ω . | 5 | 0.05 | 0.25 |
| 11 | Resistencia de 100K Ω . | 1 | 0.05 | 0.05 |
| 12 | Resistencia de 10 Ω . | 1 | 0.05 | 0.05 |
| 13 | Resistencia de 3.9K Ω . | 6 | 0.05 | 0.30 |
| 14 | Resistencia de 4.99K Ω | 1 | 0.05 | 0.05 |
| 15 | Resistencia de 33K Ω . | 1 | 0.05 | 0.05 |
| 16 | Resistencia de 47K Ω . | 1 | 0.05 | 0.05 |
| 17 | Resistencia de precisión de 15K Ω . | 1 | 0.35 | 0.35 |
| 18 | Resistencia de precisión de 33K Ω . | 2 | 0.35 | 0.70 |
| 19 | Resistencia de precisión de 2.2K Ω . | 2 | 0.35 | 0.70 |
| 20 | Potenciómetro de precisión de 20K Ω | 1 | 1.50 | 1.50 |
| 21 | Potenciómetro de precisión de 50K Ω . | 1 | 1.50 | 1.50 |
| 22 | Potenciómetro de precisión de 1K Ω . | 2 | 1.50 | 3.00 |
| 23 | Potenciómetro de precisión de 100K Ω . | 2 | 1.50 | 3.00 |
| 24 | Conector tipo bornera de 3 bornes. | 10 | 0.35 | 3.50 |
| 25 | Un conector tipo bornera de 2 bornes. | 1 | 0.30 | 0.30 |
| 26 | Diodos LED. | 6 | 0.30 | 1.80 |
| 27 | Diodos 1N4007. | 2 | 1.00 | 2.00 |
| 28 | Header de 20 pines. | 1 | 1.00 | 1.00 |
| 29 | Header de 2 pines. | 6 | 0.30 | 1.80 |
| 30 | Header de 4 pines. | 1 | 0.50 | 0.50 |

| | | | | |
|----|---|---|------------------------|---------------|
| 31 | Header de 3 pines. | 2 | 0.40 | 0.80 |
| 32 | Zócalo de 8 pines. | 2 | 0.20 | 0.40 |
| 33 | Zócalo de 40 pines | 1 | 1.00 | 1.00 |
| 34 | CI AD620AN. | 1 | 20.00 | 20.00 |
| 35 | CI LM358 | 1 | 0.80 | 0.80 |
| 36 | PIC 18F4550. | 1 | 12.00 | 12.00 |
| 37 | Regulador de voltaje en CI 7805. | 1 | 0.50 | 0.50 |
| 38 | Regulador de voltaje en CI LM1117Txx. | 1 | 1.00 | 1.00 |
| 39 | Disipador de calor | 1 | 1.00 | 1.00 |
| 40 | Pulsadores N/A. | 6 | 0.50 | 3.00 |
| 41 | Switch | 3 | 0.30 | 0.90 |
| 42 | Conector USB | 1 | 1.00 | 1.00 |
| 43 | Cristal de 20MHz. | 1 | 0.60 | 0.60 |
| 44 | Display grafico de cristal líquido (GLCD) | 1 | 35.00 | 35.00 |
| 45 | Jack de energía. | 1 | 0.20 | 0.20 |
| 46 | Sensor de presión MPX10D | 1 | 40.00 | 40.00 |
| 47 | Sensor de temperatura LM35 | 1 | 4.00 | 4.00 |
| 48 | Sensor de humedad HIH 4000-001 | 1 | 40.00 | 40.00 |
| 49 | Conector A/C | 1 | 1.00 | 1.00 |
| 50 | Porta Fusibles | 1 | 0.50 | 0.50 |
| 51 | Fusible de 1A | 1 | 0.20 | 0.20 |
| 52 | Interruptor | 1 | 1.00 | 1.00 |
| 53 | Cable de energía | 1 | 2.00 | 2.00 |
| 54 | Cable plano (m) | 2 | 4.50 | 9.00 |
| 55 | Cable USB | 1 | 3.00 | 3.00 |
| 56 | Caja metálica para los circuitos | 1 | 55.00 | 55.00 |
| 57 | Fuentes switching | 2 | 11.50 | 23.00 |
| 58 | Barniz | 1 | 3.50 | 3.50 |
| 59 | Mano de obra (placa principal) | 1 | 50.00 | 50.00 |
| 60 | Suelda | 1 | 5.00 | 5.00 |
| 61 | Otros | 1 | 15.00 | 15.00 |
| | | | TOTAL (USD) | 362.35 |

CAPITULO 5

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES:

- El trabajo se culmina en su totalidad de acuerdo a los objetivos planteados dentro del Plan de proyecto, esto permite que el usuario se familiarice más con temas referentes a aplicaciones tecnológicas dentro del área de control con PICs.
- La culminación del sistema de adquisición de datos fue posible gracias a la importación de sensores de alta tecnología como es el de presión y humedad, y, a la acertada investigación de temas de interés para el proyecto, esto permitió enriquecer los conocimientos teóricos adquiridos en el transcurso de la carrera.
- Nuestro sistema de adquisición de datos está diseñado para lectura de datos y no para escritura, es decir, el sistema simplemente leerá los datos adquiridos y los presentará en los dispositivos de salida (GLCD ó INTERFAZ GRÁFICA), pero no tomará acciones en el caso de que las variables sobrepasen los rangos permitidos.
- En el mercado existen varios software dedicados exclusivamente al desarrollo de aplicaciones basadas en microcontroladores PICs, pero se debió seleccionar los adecuados basándonos en los requerimientos de diseño de nuestro sistema. Una de las razones fundamentales por la que se utilizó el software Protón es por que incluye dentro de sus librerías el display gráfico de cristal líquido (GLCD) y por la fácil creación del código necesario para la comunicación USB.

- Si bien es cierto que el sistema está basado en un estándar muy reconocido y utilizado en la actualidad (USB), es necesario indicar que una limitación muy fuerte de este es la longitud máxima del cable que se puede utilizar (5m), por lo que se concluye que el equipo debe estar siempre conectado a una distancia máxima de 5m cuando se necesite trabajar con la computadora, sin embargo, este problema es solucionable hasta cierto punto gracias al GLCD ya que este trabaja en forma independiente y se puede movilizar el equipo sin necesidad de conectarlo al computador.
- Para lograr visualizar los datos en alguna PC, basta con cargar el archivo .exe de Visual Basic y ejecutarlo una vez que se conecte el módulo de adquisición de datos al computador. No es necesario tener cargado el programa Visual Basic en el computador.
- Durante el tratamiento de los diferentes tipos de señales analógicas se requiere una etapa de amplificación, esta etapa puede resultar costosa si el sensor en cuestión entrega magnitudes muy bajas.
- Para lograr establecer la comunicación USB entre el computador y el Pic, es necesario generar un código de programa que será cargado en el PIC, este código es generado en forma automática por Easy Hid USB Wizard, opción que se encuentra en el software proton Ide. Una vez que se carga este código en el Pic, entonces automáticamente es reconocido el Hardware como un dispositivo Hid.

- En el módulo existen 4 entradas disponibles para que sean programadas de acuerdo al requerimiento del usuario, para que esto sea posible se requiere obtener una ecuación de la magnitud física en función del voltaje, tomando en cuenta que el valor del voltaje debe ser convertido en valor binario para que el programa pueda entender como un dato digital.

5.2 RECOMENDACIONES

- Si se desea usar otra variable física, se recomienda usar un sensor que no sea complejo al momento de adecuarlo a nuestro sistema, esto ya que el Pic debe recibir señales eléctricas entre 0 y 5V. Si el sensor adquirido entrega voltajes muy pequeños entonces se deberá diseñar una etapa de amplificación muy alta para que el PIC pueda tratar la señal, esto puede hacer que se requiere de elementos electrónicos costosos.
- Para manejar el equipo, se deberá primero recurrir al manual de usuario para tener conocimiento de las funciones de cada uno de los elementos manipulables en el sistema. Una mala conexión o manejo inadecuado de la tarjeta, hará que el sistema presente errores en el momento de transmisión de datos, ó a su vez, puede dañarse algún elemento y dejará de trabajar la tarjeta.
- Se recomienda que el diseño de los circuitos para acoplar la señal al Pic sean lo más pequeños posibles, ya que estos quedarán en la parte interna de la caja, y el espacio físico disponible es mínimo.

- Se debe tener mucho cuidado en la manipulación de los sensores, en especial el de humedad y el de presión, estos no existen en el país y su consecución fue muy difícil.
- Se recomienda tener cuidado al colocar objetos pesados cerca al GLCD, este elemento es costoso.
- Un tema que se debe tener muy en cuenta es que existen potenciómetros de precisión en las tarjetas del sistema, estos no deben ser manipulados por ningún motivo ya que estos sirven para la calibración de los sensores, si se mueven los potenciómetros provocará que los sensores se descalibren, y la calibración de estos requieren mucho tiempo e investigación.

BIBLIOGRAFIA

Libros:

- REYES, Carlos; Microcontroladores PIC, Editorial Rispergraf C.A, Segunda edición; Ecuador 2006.
- Curso de Pics Ing. Alcivar Costales

Páginas Web:

- http://www.gii.upv.es/personal/gbenet/dispositivos_ctrl/adquisicion%20de%20datos/adquisicion%20de%20datos.pdf
- http://www.forpas.us.es/aula/hardware/dia2_USB.pdf
- http://www.usb.org/developers/devclass_docs/HID1_11.pdf
- <http://www.i-micro.com/pdf/articulos/usb.pdf>
- <http://www.beyondlogic.org/usbnutshell/usb-in-a-nutshell.pdf>
- <http://www.robokits.co.in/datasheets/PIC18F4550.pdf>
- <http://ww1.microchip.com/downloads/en/DeviceDoc/39632b.pdf>
- http://www.eeplace.com/dm/agilent/admf/2008/papers/common/ADMF2008_USB_Microcontrollers_ByMicrochips.pdf
- <http://www.ate.uniovi.esfernandoDoc2004SEDPresentacionesIntroduccion%20a%20los%20MCUs%20v2.pdf>
- <http://lc.fie.umich.mx/~jrincon/manual%20PICs%20Ruddy.pdf>
- <http://www.monografias.com/trabajos13/fire/fire.shtml>
- <http://www.electronicafacil.net/tutoriales/Codigos-colores-resistencias.php>
- <http://www.scribd.com/doc/2940003/PIC18F4550>

ANEXOS

ANEXO A: PROGRAMAS DEL SISTEMA

❖ PROGRAMA EN PROTÓN

```

*****
!* Name : TESIS.BAS *
!* Author : [ALEXIS LUCERO & ALEXIS PILAPAÑA] *
!* Notice : Copyright (c) 2010 [ESFOT - EPN] *
!* : All Rights Reserved *
!* Date : 02/03/2010 *
!* Version : 1.0 *
!* Notes : *
!* : *
*****

```

```
' select MCU and clock speed
```

```
Device = 18F4550
```

```
XTAL = 48
```

```
TRISA=%00111111
```

```
TRISB=%00000000
```

```
TRISC=%00000111
```

```
TRISD=%00000011
```

```
TRISE=%00000111
```

```
' *****
```

```
' * Definiciones para el GLCD *
```

```
' *****
```

```
TRISD.7=0 'Pin del Reset del GLCD
```

```
PORTD.7=1 'Estado inicial 1L
```

```
Declare LCD_TYPE 1
```

```
LCD_DTPORT = PORTB
```

```
LCD_RSPIN = PORTD.5
```

```
LCD_ENPIN = PORTD.2
```

```
LCD_RWPIN = PORTD.3
```

```
LCD_CS1PIN = PORTD.4
```

```
LCD_CS2PIN = PORTD.6
```

```
Declare GLCD_CS_INVERT On
```

```
Declare INTERNAL_FONT On
```

```
' *****
```

```
' * Definiciones para los ALIAS *
```

```
' *****
```

```
Symbol Tecla1 = PORTD.1
```

```
Symbol Tecla2 = PORTD.0
```

```
Symbol Tecla3 = PORTC.2
```

```
Symbol Tecla4 = PORTC.1
```

```
Symbol Tecla5 = PORTC.0
```

```
Symbol Pcauto = PORTA.4
```

```
Symbol LedRojo = PORTC.6
```

```
Symbol LedVerde = PORTC.7
```

```
' *****
```

```
' * Definiciones para Conversor Analogico Digital *
```

```
' *****
```

```
ADIN_RES 10
```

```
'Resolucion del ADC 10 bits
```

```
ADIN_TAD FRC
```

```
'Elige RC oscilador de muestreo para ADC
```

```
ADIN_STIME 100 'Permite 100us para
```

```
tiempo de Carga
```

```
ADCON1 = %00000111 'Setea PORTA Y
```

```
PORTE como analogicos
```

```
ADCON2.7=1
```

```
' *****
```

```
' * Definiciones para Variables *
```

```
' *****
```

```
Dim Raw As Word
```

```
Dim Value As Word
```

```
Dim Volts As Byte
```

```
Dim nins As Byte
```

```
Dim Tiempo As Word
```

```
Dim X1 As Byte
```

```
Dim Sigs As Byte
```

```
Dim y1 As Word
```

```
Dim Flag1 As Bit
```

```
Dim Flag2 As Bit
```

```
Dim Flag3 As Bit
```

```
Dim Flag4 As Bit
```

```
Dim Flag5 As Bit
```

```
Dim Valor As Float
```

```
Dim Valor1 As Float
```

```
Dim Valor2 As Float
```

```
Dim Valor3 As Float
```

```
Symbol Rebote = 100
```

```
' *****
```

```
' * Definiciones para Presentar una Imagen *
```

```
' *****
```

```
Dim X As Byte '<-----POSICION X DE 0 A 128
```

```
Dim Y As Byte '<-----POSICION Y DE 1 A 8
```

```
Dim DATO As Byte '<-----VALOR A ESCRIBIR EN POSICION X, Y
```

```
Dim PUNTERO As Word '<-----POSICIONAMIENTO DE VALOR
```

```
' descriptor file, located in \includb_18 - a copy
```

```
' is located in the same folder as this file
```

```
USB_DESCRIPTOR = "TESISDESC.inc"
```

```
' we are going to use an interrupt to keep the USB
```

```
' connection alive, so disable auto polling from within
```

```
' the USBIn and USBOut commands
```

```
USBIN_AUTO_POLL = OFF
```

```
USBOUT_AUTO_POLL = OFF
```

```
' USB buffer...
```

```
Symbol USBBufferSizeMax = 8
```

```
Symbol USBBufferSizeTX = 8
```

```
Symbol USBBufferSizeRX = 8
```

```
Dim USBBuffer[USBBufferSizeMax] As Byte
```

```
' some useful flags...
```

```
Dim PP0 As Byte SYSTEM ' USBPOLL status return
```

```
Symbol CARRY_FLAG = STATUS.0 ' high if microcontroller does not have
```

```
control over the buffer
```

```
Symbol ATTACHED_STATE = 6 ' is USB attached
```

```
Symbol TRNIF = UIR.3 ' low if USB Busy
```

```
Dim TIMER1 As TMR1L.Word ' access 16 bits of TMR1
```

```
' use an interrupt to keep USB connection alive...
```

```
Symbol TimerPreload = $A23F ' approx 2ms
```

```
ON_HARDWARE_INTERRUPT GoTo USBServiceInterrupt
```

```
GoTo ProgramStart
```

```
' *****
```

```
' * the USB interrupt service routine maintains the *
```

```
' connection to the bus - without this routine, a *
```

```
' call to USBPoll, USBIn or USBOut must be made *
```

```
' every couple of milliseconds or so *
```

```
' *****
```

```
USBServiceInterrupt:
```

```
' don't poll USB if it's busy...
```

```
btfs TRNIF
```

```
bra ExitInterrupt
```

```
' poll the USB interface...
```

```
Call (Check@BusStatus)
```

```
Call (Driver@Service)
```

```
' clear interrupt flag and exit...
```

```
ExitInterrupt:
```

```
bcf PIR1,0
```

```

TIMER1 = TimerPreload
retfie fast
' *****
' * program starts here... *
' *****
ProgramStart:
If Pcauto=0 Then
  LedRojo=1
  LedVerde=0
  GoSub AttachToUSB
  ' turn on interrupt timer...
  TIMER1 = TimerPreload
  T1CON = %10000001          'Habilita 16 bits, Timer1 ON

  PIE1 = %00000001          'Habilita TMR1 interrupcion por
overflow
  INTCON = %11000000        'Habilita interrupcion Global e
Interrupcion de Perifericos
  Flag5=0
Else
  INTCON = %00000000        'DesHabilita interrupcion Global e
Interrupcion de Perifericos
  LedRojo=0
  LedVerde=1
  Flag5=1
EndIf
nins=4
Flag1=0
Flag2=0
Flag3=0
Flag4=0
Sig5=1

DelayMS 50
Y = 0
PUNTERO = 0
Cls
Repeat
  Repeat
    DATO = CRead caratula1 + PUNTERO
    LCDWrite Y,X,[DATO]
    Inc X
    Inc PUNTERO
  Until X = 128
  X = 0
  Inc Y
Until Y = 8
DelayMS 4000

Cls
Y = 0
PUNTERO = 0
Cls
Repeat
  Repeat
    DATO = CRead caratula1A + PUNTERO
    LCDWrite Y,X,[DATO]
    Inc X
    Inc PUNTERO
  Until X = 128
  X = 0
  Inc Y
Until Y = 8
DelayMS 4000
Cls

' *****
' * main program loop *
' *****
ProgramLoop:
' your code in here - to send data, load USBBuffer
' and then call DoUSBOut. To receive data, call DoUSBIn...

' GoSub DoUSBIn
Select Case nins
  Case 1
    GoSub Retardo
    If Flag2=0 Then
      Raw = ADIn 0
      If Flag5=0 Then
        USBBuffer[1]=Raw.LowByte
        USBBuffer[2]=Raw.HighByte
        USBBuffer[3]=1
        USBBuffer[4]=nins
        GoSub DoUSBOut
      EndIf
      Valor=RAW*35
      Valor=Valor/10000
      Valor=Valor - 0.7244
      Print At 0,0,"PRE = ",Dec Valor," PSI", " " ' Display the
RAW data
    EndIf
  Case 2
    GoSub Retardo
    If Flag2=0 Then
      Raw = ADIn 0
      If Flag5=0 Then
        USBBuffer[1]=Raw.LowByte
        USBBuffer[2]=Raw.HighByte
        USBBuffer[3]=1
        USBBuffer[4]=nins
        GoSub DoUSBOut
      EndIf
      Valor=Raw*35
      Valor=Valor/10000
      Valor=Valor - 0.7244
      Print At 0,0,"PRE = ",Dec Valor," PSI", " " ' Display the
RAW data
    EndIf
    GoSub Retardo
    If Flag2=0 Then
      Raw = ADIn 1
      If Flag5=0 Then
        USBBuffer[1]=Raw.LowByte
        USBBuffer[2]=Raw.HighByte
        USBBuffer[3]=2
        USBBuffer[4]=nins
        GoSub DoUSBOut
      EndIf
      Raw=Raw>>2
      Print At 1,0,"TEM = ",DEC2 Raw," oC", " " ' Display the
RAW data
    End If
  Case 3
    GoSub Retardo
    If Flag2=0 Then
      Raw = ADIn 0
      If Flag5=0 Then
        USBBuffer[1]=Raw.LowByte
        USBBuffer[2]=Raw.HighByte
        USBBuffer[3]=1
        USBBuffer[4]=nins
        GoSub DoUSBOut
      EndIf
      Valor=Raw*35
      Valor=Valor/10000
      Valor=Valor - 0.7244
      Print At 0,0,"PRE = ",Dec Valor," PSI", " " ' Display the
RAW data
    EndIf
    GoSub Retardo
    If Flag2=0 Then
      Raw = ADIn 1
      If Flag5=0 Then
        USBBuffer[1]=Raw.LowByte
        USBBuffer[2]=Raw.HighByte
        USBBuffer[3]=2
        USBBuffer[4]=nins
        GoSub DoUSBOut
      EndIf
      Raw=Raw>>2
      Print At 1,0,"TEM = ",DEC2 Raw," oC", " " ' Display the
RAW data
    End If
    GoSub Retardo
    If Flag2=0 Then

```

```

Raw = ADIn 2
Read the ADC
If Flag5=0 Then
  USBBuffer[1]=Raw.LowByte
  USBBuffer[2]=Raw.HighByte
  USBBuffer[3]=3
  USBBuffer[4]=nins
  GoSub DoUSBOut
Endif
Valor1=RAW*raw
Valor1=Valor1/(10000.0)
Valor2=25.06*Raw
Valor2=Valor2/100
Valor1=valor2-valor1
Valor3=Valor1 - 71.074
Print At 2,0,"HUM = ",Dec Valor3," %", " " ' Display
the RAW data
Endif
Case 4
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 0
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=1
      USBBuffer[4]=nins
      GoSub DoUSBOut
    Endif
    Valor=Raw*35
    Valor=Valor/10000
    Valor=Valor - 0.7244
    Print At 0,0,"PRE = ",Dec Valor," PSI", " " ' Display
the RAW data
  Endif
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 1
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=2
      USBBuffer[4]=nins
      GoSub DoUSBOut
    Endif
    Raw=Raw>>2
    Print At 1,0,"TEM = ",Dec2 Raw," oC", " " ' Display
the RAW data
  Endif
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 2
    Read the ADC
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=3
      USBBuffer[4]=nins
      GoSub DoUSBOut
    Endif
    Valor1=RAW*raw
    Valor1=Valor1/(10000.0)
    Valor2=25.06*Raw
    Valor2=Valor2/100
    Valor1=valor2-valor1
    Valor3=Valor1 - 71.074
    Print At 2,0,"HUM = ",Dec Valor3," %", " " ' Display
the RAW data
  Endif
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 3
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=4
      USBBuffer[4]=nins
      GoSub DoUSBOut
    Endif
    Print At 3,0,"POT = ",Dec4 Raw," " ' Display the RAW data
  Endif
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 4
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=5
      USBBuffer[4]=nins
      GoSub DoUSBOut
    Endif
    Print At 4,0,"SEN5= ",Dec4 Raw," " ' Display the RAW data
  Endif
Case 6
  GoSub Retardo
  Print At 3,0,"POT = ",Dec4 Raw," " ' Display the RAW data
Endif
Case 5
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 0
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=1
      USBBuffer[4]=nins
      GoSub DoUSBOut
    Endif
    Valor=Raw*35
    Valor=Valor/10000
    Valor=Valor - 0.7244
    Print At 0,0,"PRE = ",Dec Valor," PSI", " " ' Display the
RAW data
  Endif
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 1
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=2
      USBBuffer[4]=nins
      GoSub DoUSBOut
    Endif
    Raw=Raw>>2
    Print At 1,0,"TEM = ",Dec2 Raw," oC", " " ' Display the
RAW data
  Endif
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 2
    Read the ADC
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=3
      USBBuffer[4]=nins
      GoSub DoUSBOut
    Endif
    Valor1=RAW*raw
    Valor1=Valor1/(10000.0)
    Valor2=25.06*Raw
    Valor2=Valor2/100
    Valor1=valor2-valor1
    Valor3=Valor1 - 71.074
    Print At 2,0,"HUM = ",Dec Valor3," %", " " ' Display the
RAW data
  Endif
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 3
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=4
      USBBuffer[4]=nins
      GoSub DoUSBOut
    Endif
    Print At 3,0,"POT = ",Dec4 Raw," " ' Display the RAW data
  Endif
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 4
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=5
      USBBuffer[4]=nins
      GoSub DoUSBOut
    Endif
    Print At 4,0,"SEN5= ",Dec4 Raw," " ' Display the RAW data
  Endif
Case 6
  GoSub Retardo

```

```

If Flag2=0 Then
  Raw = ADIn 0
  If Flag5=0 Then
    USBBuffer[1]=Raw.LowByte
    USBBuffer[2]=Raw.HighByte
    USBBuffer[3]=1
    USBBuffer[4]=nins
    GoSub DoUSBOut
  EndIf
  Valor=Raw*35
  Valor=Valor/10000
  Valor=Valor - 0.7244
  Print At 0,0,"PRE = ",Dec Valor," PSI", " " ' Display
the RAW data
EndIf
GoSub Retardo
If Flag2=0 Then
  Raw = ADIn 1
  If Flag5=0 Then
    USBBuffer[1]=Raw.LowByte
    USBBuffer[2]=Raw.HighByte
    USBBuffer[3]=2
    USBBuffer[4]=nins
    GoSub DoUSBOut
  EndIf
  Raw=Raw>>2
  Print At 1,0,"TEM = ",Dec2 Raw," oC", " " ' Display
the RAW data
EndIf
GoSub Retardo
If Flag2=0 Then
  Raw = ADIn 2
  Read the ADC
  If Flag5=0 Then
    USBBuffer[1]=Raw.LowByte
    USBBuffer[2]=Raw.HighByte
    USBBuffer[3]=3
    USBBuffer[4]=nins
    GoSub DoUSBOut
  EndIf
  Valor1=RAW*raw
  Valor1=Valor1/(10000.0)
  Valor2=25.06*Raw
  Valor2=Valor2/100
  Valor1=valor2-valor1
  Valor3=Valor1 - 71.074
  Print At 2,0,"HUM = ",Dec Valor3," %", " " ' Display
the RAW data
EndIf
GoSub Retardo
If Flag2=0 Then
  Raw = ADIn 3
  If Flag5=0 Then
    USBBuffer[1]=Raw.LowByte
    USBBuffer[2]=Raw.HighByte
    USBBuffer[3]=4
    USBBuffer[4]=nins
    GoSub DoUSBOut
  EndIf
  Print At 3,0,"POT = ",Dec4 Raw," " ' Display the RAW
data
EndIf
GoSub Retardo
If Flag2=0 Then
  Raw = ADIn 4
  If Flag5=0 Then
    USBBuffer[1]=Raw.LowByte
    USBBuffer[2]=Raw.HighByte
    USBBuffer[3]=5
    USBBuffer[4]=nins
    GoSub DoUSBOut
  EndIf
  Print At 4,0,"SEN5= ",Dec4 Raw," " ' Display the RAW
data
EndIf
GoSub Retardo
If Flag2=0 Then
  Raw = ADIn 5
  If Flag5=0 Then
    USBBuffer[1]=Raw.LowByte
    USBBuffer[2]=Raw.HighByte
    USBBuffer[3]=6
    USBBuffer[4]=nins
    GoSub DoUSBOut
  EndIf
  Print At 5,0,"SEN6= ",Dec4 Raw," " ' Display the RAW data
EndIf
Case 7
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 0
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=1
      USBBuffer[4]=nins
      GoSub DoUSBOut
    EndIf
    Valor=Raw*35
    Valor=Valor/10000
    Valor=Valor - 0.7244
    Print At 0,0,"PRE = ",Dec Valor," PSI", " " ' Display the
RAW data
  EndIf
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 1
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=2
      USBBuffer[4]=nins
      GoSub DoUSBOut
    EndIf
    Raw=Raw>>2
    Print At 1,0,"TEM = ",Dec2 Raw," oC", " " ' Display the
RAW data
  EndIf
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 2
    Read the ADC
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=3
      USBBuffer[4]=nins
      GoSub DoUSBOut
    EndIf
    Valor1=RAW*raw
    Valor1=Valor1/(10000.0)
    Valor2=25.06*Raw
    Valor2=Valor2/100
    Valor1=valor2-valor1
    Valor3=Valor1 - 71.074
    Print At 2,0,"HUM = ",Dec Valor3," %", " " ' Display the
RAW data
  EndIf
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 3
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=4
      USBBuffer[4]=nins
      GoSub DoUSBOut
    EndIf
    Print At 3,0,"POT = ",Dec4 Raw," " ' Display the RAW data
  EndIf
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 4
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=5
      USBBuffer[4]=nins
    EndIf
  EndIf

```

```

    GoSub DoUSBOut
  EndIf
  Print At 4,0,"SEN5= ",DEC4 Raw," " ' Display the RAW
data
EndIf
GoSub Retardo
If Flag2=0 Then
  Raw = ADIn 5
  If Flag5=0 Then
    USBBuffer[1]=Raw.LowByte
    USBBuffer[2]=Raw.HighByte
    USBBuffer[3]=6
    USBBuffer[4]=nins
    GoSub DoUSBOut
  EndIf
  Print At 5,0,"SEN6= ",DEC4 Raw," " ' Display the RAW
data
EndIf
GoSub Retardo
If Flag2=0 Then
  Raw = ADIn 6
  If Flag5=0 Then
    USBBuffer[1]=Raw.LowByte
    USBBuffer[2]=Raw.HighByte
    USBBuffer[3]=7
    USBBuffer[4]=nins
    GoSub DoUSBOut
  EndIf
  Print At 6,0,"SEN7= ",DEC4 Raw," " ' Display the RAW
data
EndIf
Case 8
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 0
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=1
      USBBuffer[4]=nins
      GoSub DoUSBOut
    EndIf
    Valor=Raw*35
    Valor=Valor/10000
    Valor=Valor - 0.7244
    Print At 0,0,"PRE = ",Dec Valor," PSI"," " ' Display
the RAW data
  EndIf
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 1
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=2
      USBBuffer[4]=nins
      GoSub DoUSBOut
    EndIf
    Raw=Raw>>2
    Print At 1,0,"TEM = ",DEC2 Raw," oC," " ' Display
the RAW data
  End If
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 2
  Read the ADC
  If Flag5=0 Then
    USBBuffer[1]=Raw.LowByte
    USBBuffer[2]=Raw.HighByte
    USBBuffer[3]=3
    USBBuffer[4]=nins
    GoSub DoUSBOut
  EndIf
  Valor1=RAW*raw
  Valor1=Valor1/(10000.0)
  Valor2=25.06*Raw
  Valor2=Valor2/100
  Valor1=valor2-valor1
  Valor3=Valor1 - 71.074

```

```

    Print At 2,0,"HUM = ",Dec Valor3," %," " ' Display the
RAW data
  EndIf
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 3
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=4
      USBBuffer[4]=nins
      GoSub DoUSBOut
    EndIf
    Print At 3,0,"POT = ",DEC4 Raw," " ' Display the RAW data
  EndIf
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 4
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=5
      USBBuffer[4]=nins
      GoSub DoUSBOut
    EndIf
    Print At 4,0,"SEN5= ",DEC4 Raw," " ' Display the RAW data
  EndIf
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 5
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=6
      USBBuffer[4]=nins
      GoSub DoUSBOut
    EndIf
    Print At 5,0,"SEN6= ",DEC4 Raw," " ' Display the RAW data
  EndIf
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 6
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=7
      USBBuffer[4]=nins
      GoSub DoUSBOut
    EndIf
    Print At 6,0,"SEN7= ",DEC4 Raw," " ' Display the RAW data
  EndIf
  GoSub Retardo
  If Flag2=0 Then
    Raw = ADIn 7
    If Flag5=0 Then
      USBBuffer[1]=Raw.LowByte
      USBBuffer[2]=Raw.HighByte
      USBBuffer[3]=8
      USBBuffer[4]=nins
      GoSub DoUSBOut
    EndIf
    Print At 7,0,"SEN8= ",DEC4 Raw," " ' Display the RAW data
  EndIf
End Select
+++++
If Tecla3=0 Then
  DelayMS Rebote
  While Tecla3=0
  Wend
  DelayMS Rebote
  Flag1=1
  Print At 0,16,Inverse 0,"#IN's" ' Display the RAW data
  Print At 1,18,Inverse 1," ",DEC1 nins," "
EndIf
While Flag1=1
  If Tecla1=0 Then
  DelayMS Rebote
  While Tecla1=0

```

```

Wend
DelayMS Rebote
nins=nins+1
If nins=9 Then
  nins=8
End If
Print At 1,18,Inverse 1," ",DEC1 nins," " ' Display the RAW
data
EndIf
If Tecla2=0 Then
  DelayMS Rebote
  While Tecla2=0
  Wend
  DelayMS Rebote
  nins=nins-1
  If nins=0 Then
    nins=1
  End If
  Print At 1,18,Inverse 1," ",DEC1 nins," " ' Display the RAW
data
EndIf
If Tecla3=0 Then
  DelayMS Rebote
  While Tecla3=0
  Wend
  DelayMS Rebote
  Flag1=0
  Flag2=0
  Print At 1,18,Inverse 0," ",DEC1 nins," "
  CIs
EndIf
Wend
+++++
+++++
If Tecla5=0 Then
  DelayMS Rebote
  While Tecla5=0
  Wend
  DelayMS Rebote
  Flag3=1
  Print At 6,15,Inverse 0,"SENSOR"," "
  Print At 7,15,Inverse 1," ",DEC1 Sigs," "
EndIf
While Flag3=1
  If Tecla4=0 Then
    DelayMS Rebote
    While Tecla4=0
    Wend
    DelayMS Rebote
    Sigs=Sigs+1
    If Sigs=9 Then
      Sigs=1
    End If
    Print At 7,15,Inverse 1," ",DEC1 Sigs," "
  EndIf
  If Tecla5=0 Then
    DelayMS Rebote
    While Tecla5=0
    Wend
    DelayMS Rebote
    Flag3=0
    Flag2=0
    Flag4=1
    Print At 7,15,Inverse 0," ",DEC1 Sigs," "
    CIs
  EndIf
Wend
+++++
+++++
While Flag4=1
  Select Case Sigs
  Case 1
    GoSub Retardo
    If Flag2=0 Then
      Raw = ADIn 0
      If Flag5=0 Then
        USBBuffer[1]=Raw.LowByte
        USBBuffer[2]=Raw.HighByte
        USBBuffer[3]=1
      End If
    End If
  Case 2
    GoSub Retardo
    If Flag2=0 Then
      Raw = ADIn 1
      If Flag5=0 Then
        USBBuffer[1]=Raw.LowByte
        USBBuffer[2]=Raw.HighByte
        USBBuffer[3]=2
        USBBuffer[4]=10
      End If
    End If
  Case 3
    GoSub Retardo
    If Flag2=0 Then
      Raw = ADIn 2
      ' Read the ADC
      If Flag5=0 Then
        USBBuffer[1]=Raw.LowByte
        USBBuffer[2]=Raw.HighByte
        USBBuffer[3]=3
        USBBuffer[4]=11
      End If
    End If
  Case 4
    GoSub Retardo
    If Flag2=0 Then
      Raw = ADIn 3
      If Flag5=0 Then
        USBBuffer[1]=Raw.LowByte
        USBBuffer[2]=Raw.HighByte
        USBBuffer[3]=4
        USBBuffer[4]=12
      End If
    End If
  Case 5
    GoSub Retardo
    If Flag2=0 Then
      Raw = ADIn 4
      If Flag5=0 Then
        USBBuffer[1]=Raw.LowByte
        USBBuffer[2]=Raw.HighByte
        USBBuffer[3]=5
      End If
    End If
  End Select
End While

```

```

USBBuffer[4]=9
GoSub DoUSBOut
EndIf
Valor=RAW*35
Valor=Valor/10000
Valor=Valor - 0.7244
Print At 1,0," SENSOR DE PRESION " ' Display the
RAW data
Print At 3,0,"{ 0 PSI - 1.5 PSI }" ' Display the RAW data
Print At 5,7, Dec Valor," PSI"," " ' Display the RAW data
EndIf
Case 2
GoSub Retardo
If Flag2=0 Then
  Raw = ADIn 1
  If Flag5=0 Then
    USBBuffer[1]=Raw.LowByte
    USBBuffer[2]=Raw.HighByte
    USBBuffer[3]=2
    USBBuffer[4]=10
  End If
EndIf
Raw=Raw>>2
Print At 1,0," SENSOR TEMPERATURA " ' Display the
RAW data
Print At 3,0,"{ 0 oC - 100 oC }" ' Display the RAW data
Print At 5,8,DEC2 Raw," oC"," " ' Display the RAW data
EndIf
Case 3
GoSub Retardo
If Flag2=0 Then
  Raw = ADIn 2
  ' Read the ADC
  If Flag5=0 Then
    USBBuffer[1]=Raw.LowByte
    USBBuffer[2]=Raw.HighByte
    USBBuffer[3]=3
    USBBuffer[4]=11
  End If
EndIf
Valor1=RAW*raw
Valor1=Valor1/(10000.0)
Valor2=25.06*Raw
Valor2=Valor2/100
Valor1=valor2-valor1
Valor3=Valor1 - 71.074
Print At 1,0," SENSOR DE HUMEDAD " ' Display the
RAW data
Print At 3,0,"{ 0 % - 100 % }" ' Display the RAW data
Print At 5,8,Dec Valor3," %"," " ' Display the RAW data
EndIf
Case 4
GoSub Retardo
If Flag2=0 Then
  Raw = ADIn 3
  If Flag5=0 Then
    USBBuffer[1]=Raw.LowByte
    USBBuffer[2]=Raw.HighByte
    USBBuffer[3]=4
    USBBuffer[4]=12
  End If
EndIf
Print At 1,0," POTENCIOMETRO " ' Display the
RAW data
Print At 3,0,"{ 0 - 1024 }" ' Display the RAW data
Print At 5,8,DEC4 Raw," " ' Display the RAW data
EndIf
Case 5
GoSub Retardo
If Flag2=0 Then
  Raw = ADIn 4
  If Flag5=0 Then
    USBBuffer[1]=Raw.LowByte
    USBBuffer[2]=Raw.HighByte
    USBBuffer[3]=5
  End If
End If

```

```

        USBBuffer[4]=13
        GoSub DoUSBOut
    EndIf
    Print At 1,0,"  SENSOR No. 5 " ' Display the RAW
data
    Print At 3,0,"{ 0 - 1024 }" ' Display the RAW
data
    Print At 5,8,DEC4 Raw," " ' Display the RAW
data
EndIf
Case 6
GoSub Retardo
If Flag2=0 Then
    Raw = ADIn 5
    If Flag5=0 Then

        USBBuffer[1]=Raw.LowByte
        USBBuffer[2]=Raw.HighByte
        USBBuffer[3]=6
        USBBuffer[4]=14
        GoSub DoUSBOut
    EndIf
    Print At 1,0,"  SENSOR No. 6 " ' Display the RAW
data
    Print At 3,0,"{ 0 - 1024 }" ' Display the RAW
data
    Print At 5,8,DEC4 Raw," " ' Display the RAW
data
EndIf
Case 7
GoSub Retardo
If Flag2=0 Then
    Raw = ADIn 6
    If Flag5=0 Then

        USBBuffer[1]=Raw.LowByte
        USBBuffer[2]=Raw.HighByte
        USBBuffer[3]=7
        USBBuffer[4]=15
        GoSub DoUSBOut
    EndIf
    Print At 1,0,"  SENSOR No. 7 " ' Display the RAW
data
    Print At 3,0,"{ 0 - 1024 }" ' Display the RAW
data
    Print At 5,8,DEC4 Raw," " ' Display the RAW
data
EndIf
Case 8
GoSub Retardo
If Flag2=0 Then
    Raw = ADIn 7
    If Flag5=0 Then

        USBBuffer[1]=Raw.LowByte
        USBBuffer[2]=Raw.HighByte
        USBBuffer[3]=8
        USBBuffer[4]=16
        GoSub DoUSBOut
    EndIf
    Print At 1,0,"  SENSOR No. 8 " ' Display the RAW
data
    Print At 3,0,"{ 0 - 1024 }" ' Display the RAW
data
    Print At 5,8,DEC4 Raw," " ' Display the RAW
data
EndIf
End Select
If Tecla5=0 Then
    DelayMS Rebote
    While Tecla5=0
    Wend
    DelayMS Rebote
    Flag3=1
    Print At 6,15,Inverse 0,"SENSOR", " "
    Print At 7,15,Inverse 1," ",DEC1 Sigs," "
EndIf
While Flag3=1

If Tecla4=0 Then
    DelayMS Rebote
    While Tecla4=0
    Wend
    DelayMS Rebote
    Sigs=Sigs+1
    If Sigs=9 Then
        Sigs=1
    End If
    Print At 7,15,Inverse 1," ",DEC1 Sigs," "
EndIf
If Tecla5=0 Then
    DelayMS Rebote
    While Tecla5=0
    Wend
    DelayMS Rebote
    Flag3=0
    Flag2=0
    Flag4=1
    Print At 7,15,Inverse 0,"#IN's" ' Display the RAW data
    Print At 1,18,Inverse 1," ",DEC1 nins," "
EndIf
Wend

GoTo ProgramLoop

' *****
' * receive data from the USB bus *
' *****
DoUSBIn:
USBIn 1, USBBuffer, USBBufferSizeRX, DoUSBIn
Return

' *****
' * transmit data *
' *****
DoUSBOut:
USBOut 1, USBBuffer, USBBufferSizeTX, DoUSBOut
Return

' *****
' * wait for USB interface to attach *
' *****
AttachToUSB:
Repeat

    USBPoll
    Until PP0 = ATTACHED_STATE
Return

' *****
' * Subrutina de RETARDO *
' *****
Retardo:
For y1=0 To 50
    DelayMS 1
    If Tecla3=0 Or Tecla5=0 Then
        Flag2=1
        y1=65535
    EndIf
Next y1
Return
' *****
Include "FONT.inc"
Include "CARATULA1.inc"
Include "CARATULA1A.inc"

```

❖ PROGRAMA DESARROLLADO EN VISUAL BASIC

```
' vendor and product IDs
Private Const VendorID = 6017
Private Const ProductID = 2000
```

```
' read and write buffers
Private Const BufferInSize = 8
Private Const BufferOutSize = 8
Dim BufferIn(0 To BufferInSize) As Byte
Dim BufferOut(0 To BufferOutSize) As Byte
Dim Z(999) As Double
Dim CHVal As Byte
Dim X As Single
Dim Q As Single
Dim P As Single
Dim DAQINAN As Integer ' ENTRADA VOLTAJE ANALOGICO
Dim OPENHID As Byte
Dim Valor1 As Double
```

```
Private Sub Clear()
    Q = 0
    PICTAN.Scale (0, 1023)-(1000, 0)
    PICTAN.Cls
    PICTAN.Line (0, 1023)-(0, 0)
    PICTAN.Line (0, 1)-(1000, 1)
    PICTAN.Line (0, PICTAN.Height / 2)-(0, PICTAN.Height / 2)
End Sub
```

```
Private Sub CMBCH_CLICK()
    Select Case CMBCH.Text
        Case "CANAL1"
            CHVal = 1
        Case "CANAL2"
            CHVal = 2
        Case "CANAL3"
            CHVal = 3
        Case "CANAL4"
            CHVal = 4
        Case "CANAL5"
            CHVal = 5
        Case "CANAL6"
            CHVal = 6
        Case "CANAL7"
            CHVal = 7
        Case "CANAL8"
            CHVal = 8
    End Select
    Clear
End Sub
```

```
Private Sub CMDEXIT_Click()
    DisconnectFromHID
End Sub
```

```
Private Sub CMDSTART_Click()
    CMDSTART.Visible = False
    Command1.Visible = True
    CMDEXIT.Enabled = False
    Shape1.FillColor = &HFF00&
    OPENHID = ConnectToHID(Me.hwnd)
    Clear
End Sub
```

```
Private Sub Command1_Click()
    Command1.Visible = False
    CMDSTART.Visible = True
    CMDEXIT.Enabled = True
    Shape1.FillColor = &H0&
```

```
DisconnectFromHID
```

```
End Sub
```

```
' *****
' when the form loads, connect to the HID controller - pass
' the form window handle so that you can receive notification
' events...
' *****
```

```
Private Sub Form_Load()
```

```
' do not remove!
    P = 0
    Q = 0
    X = 0
    CMBCH.AddItem "CANAL1"
    CMBCH.AddItem "CANAL2"
    CMBCH.AddItem "CANAL3"
    CMBCH.AddItem "CANAL4"
    CMBCH.AddItem "CANAL5"
    CMBCH.AddItem "CANAL6"
    CMBCH.AddItem "CANAL7"
    CMBCH.AddItem "CANAL8"
    CHVal = 1
```

```
End Sub
```

```
' *****
' disconnect from the HID controller...
' *****
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    DisconnectFromHID
End Sub
```

```
End Sub
```

```
' *****
' a HID device has been plugged in...
' *****
```

```
Public Sub OnPlugged(ByVal pHandle As Long)
    If hidGetVendorID(pHandle) = VendorID And hidGetProductID(pHandle) =
    ProductID Then
```

```
        '** YOUR CODE HERE **
```

```
End If
```

```
End Sub
```

```
' *****
' a HID device has been unplugged...
' *****
```

```
Public Sub OnUnplugged(ByVal pHandle As Long)
    If hidGetVendorID(pHandle) = VendorID And hidGetProductID(pHandle) =
    ProductID Then
```

```
        '** YOUR CODE HERE **
```

```
        Lblvendedor.Caption = ""
```

```
        lblproducto.Caption = ""
```

```
        Shape1.FillColor = &H0&
```

```
End If
```

```
End Sub
```

```
' *****
' controller changed notification - called
' after ALL HID devices are plugged or unplugged
' *****
```

```
Public Sub OnChanged()
    Dim DeviceHandle As Long
```

```
        ' get the handle of the device we are interested in, then set
```

```
        ' its read notify flag to true - this ensures you get a read
```

```
        ' notification message when there is some data to read...
```

```
        DeviceHandle = hidGetHandle(VendorID, ProductID)
```

```
        hidSetReadNotify DeviceHandle, True
```

```
End Sub
```

```
' *****
' on read event...
' *****
```

```
Public Sub OnRead(ByVal pHandle As Long)
```

```
        ' read the data (don't forget, pass the whole array)...
```

```
        If hidRead(pHandle, BufferIn(0)) Then
```

```
            '** YOUR CODE HERE **
```

```
            ' first byte is the report ID, e.g. BufferIn(0)
```

```
            ' the other bytes are the data from the microcontroller...
```

```
            Select Case BufferIn(5)
```

```
                Case 1
```

```
                    Text2.Text = ""
```

```
                    Text3.Text = ""
```

```
                    Text4.Text = ""
```

```
                    Text5.Text = ""
```

```

Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
CMBCH.Visible = True
Case 2
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
CMBCH.Visible = True
Case 3
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
CMBCH.Visible = True
Case 4
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
CMBCH.Visible = True
Case 5
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
CMBCH.Visible = True
Case 6
Text7.Text = ""
Text8.Text = ""
CMBCH.Visible = True
Case 7
Text8.Text = ""
CMBCH.Visible = True
Case 8
CMBCH.Visible = True
Case 9
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
CHVal = 1
CMBCH.Visible = False
Case 10
Text1.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
CHVal = 2
CMBCH.Visible = False
Case 11
Text1.Text = ""
Text2.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
CHVal = 3
CMBCH.Visible = False
Case 12
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
CHVal = 4
CMBCH.Visible = False

```

```

Case 13
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
CHVal = 5
CMBCH.Visible = False
Case 14
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text7.Text = ""
Text8.Text = ""
CHVal = 6
CMBCH.Visible = False
Case 15
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text8.Text = ""
CHVal = 7
CMBCH.Visible = False
Case 16
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
CHVal = 8
CMBCH.Visible = False
End Select
DAQINAN = BufferIn(3) * 256 + BufferIn(2)
Select Case BufferIn(4)
Case 1
Valor1 = 0.0035 * DAQINAN - 0.7244
Text1.Text = Format(Valor1, "0.00") + " PSI"
Case 2
Text2.Text = Format(DAQINAN / 4, "00") + " oC "
Case 3
Valor1 = (-0.0001 * DAQINAN * DAQINAN) + (0.2506 * DAQINAN) -
71.074
Text3.Text = Format(Valor1, "00.0") + " % "
Case 4
Text4.Text = Format(DAQINAN, "0000")
Case 5
Text5.Text = Format(DAQINAN, "0000")
Case 6
Text6.Text = Format(DAQINAN, "0000")
Case 7
Text7.Text = Format(DAQINAN, "0000")
Case 8
Text8.Text = Format(DAQINAN, "0000")
End Select
Select Case CHVal
Case 1
If BufferIn(4) = 1 Then
DAQINAN = BufferIn(3) * 256 + BufferIn(2)
Valor1 = 0.0035 * DAQINAN - 0.7244
TXTADVAL = Format(Valor1, "0.00")
Z(Q) = DAQINAN
PICTAN.LINE -(Q, Z(Q))
Q = Q + 1
If Q = 1000 Then
Clear
End If
End If
Case 2
If BufferIn(4) = 2 Then

```

```

    DAQINAN = BufferIn(3) * 256 + BufferIn(2)
    TXTADVAL = Format(DAQINAN / 4, "00")
    Z(Q) = DAQINAN
    PICTAN.Line -(Q, Z(Q))
    Q = Q + 1
    If Q = 1000 Then
        Clear
    End If
End If
Case 3
If BufferIn(4) = 3 Then
    DAQINAN = BufferIn(3) * 256 + BufferIn(2)
    Valor1 = (-0.0001 * DAQINAN * DAQINAN) + (0.2506 * DAQINAN) - 71.074
    TXTADVAL = Format(Valor1, "00.0")
    Z(Q) = DAQINAN
    PICTAN.Line -(Q, Z(Q))
    Q = Q + 1
    If Q = 1000 Then
        Clear
    End If
End If
Case 4
If BufferIn(4) = 4 Then
    DAQINAN = BufferIn(3) * 256 + BufferIn(2)
    TXTADVAL = Format(DAQINAN, "0000")
    Z(Q) = DAQINAN
    PICTAN.Line -(Q, Z(Q))
    Q = Q + 1
    If Q = 1000 Then
        Clear
    End If
End If
Case 5
If BufferIn(4) = 5 Then
    DAQINAN = BufferIn(3) * 256 + BufferIn(2)
    TXTADVAL = Format(DAQINAN, "0000")
    Z(Q) = DAQINAN
    PICTAN.Line -(Q, Z(Q))
    Q = Q + 1
    If Q = 1000 Then
        Clear
    End If
End If
Case 6
If BufferIn(4) = 6 Then
    DAQINAN = BufferIn(3) * 256 + BufferIn(2)
    TXTADVAL = Format(DAQINAN, "0000")
    Z(Q) = DAQINAN
    PICTAN.Line -(Q, Z(Q))
    Q = Q + 1
    If Q = 1000 Then
        Clear
    End If
End If
Case 7
If BufferIn(4) = 7 Then
    DAQINAN = BufferIn(3) * 256 + BufferIn(2)
    TXTADVAL = Format(DAQINAN, "0000")
    Z(Q) = DAQINAN
    PICTAN.Line -(Q, Z(Q))
    Q = Q + 1
    If Q = 1000 Then
        Clear
    End If
End If
Case 8
If BufferIn(4) = 8 Then
    DAQINAN = BufferIn(3) * 256 + BufferIn(2)
    TXTADVAL = Format(DAQINAN, "0000")
    Z(Q) = DAQINAN
    PICTAN.Line -(Q, Z(Q))
    Q = Q + 1
    If Q = 1000 Then
        Clear
    End If
End If
End Select
End If
End Sub

```

ANEXO B: DATASHEETS

DATASHEET AD620AN

FEATURES

EASY TO USE

Gain Set with One External Resistor
(Gain Range 1 to 1000)

Wide Power Supply Range (± 2.3 V to ± 18 V)

Higher Performance than Three Op Amp IA Designs

Available in 8-Lead DIP and SOIC Packaging

Low Power, 1.3 mA max Supply Current

EXCELLENT DC PERFORMANCE ("B GRADE")

50 μ V max, Input Offset Voltage

0.6 μ V/ $^{\circ}$ C max, Input Offset Drift

1.0 nA max, Input Bias Current

100 dB min Common-Mode Rejection Ratio (G = 10)

LOW NOISE

9 nV/ $\sqrt{\text{Hz}}$, @ 1 kHz, Input Voltage Noise

0.28 μ V p-p Noise (0.1 Hz to 10 Hz)

EXCELLENT AC SPECIFICATIONS

120 kHz Bandwidth (G = 100)

15 μ s Settling Time to 0.01%

APPLICATIONS

Weigh Scales

ECG and Medical Instrumentation

Transducer Interface

Data Acquisition Systems

Industrial Process Controls

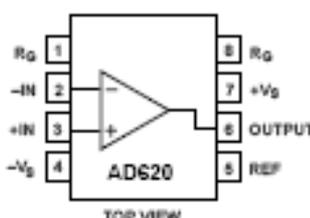
Battery Powered and Portable Equipment

PRODUCT DESCRIPTION

The AD620 is a low cost, high accuracy instrumentation amplifier that requires only one external resistor to set gains of 1 to

CONNECTION DIAGRAM

8-Lead Plastic Mini-DIP (N), Cerdip (Q)
and SOIC (R) Packages



1000. Furthermore, the AD620 features 8-lead SOIC and DIP packaging that is smaller than discrete designs, and offers lower power (only 1.3 mA max supply current), making it a good fit for battery powered, portable (or remote) applications.

The AD620, with its high accuracy of 40 ppm maximum nonlinearity, low offset voltage of 50 μ V max and offset drift of 0.6 μ V/ $^{\circ}$ C max, is ideal for use in precision data acquisition systems, such as weigh scales and transducer interfaces. Furthermore, the low noise, low input bias current, and low power of the AD620 make it well suited for medical applications such as ECG and noninvasive blood pressure monitors.

The low input bias current of 1.0 nA max is made possible with the use of Superbeta processing in the input stage. The AD620 works well as a preamplifier due to its low input voltage noise of 9 nV/ $\sqrt{\text{Hz}}$ at 1 kHz, 0.28 μ V p-p in the 0.1 Hz to 10 Hz band, 0.1 pA/ $\sqrt{\text{Hz}}$ input current noise. Also, the AD620 is well suited for multiplexed applications with its settling time of 15 μ s to 0.01% and its cost is low enough to enable designs with one in-amp per channel.

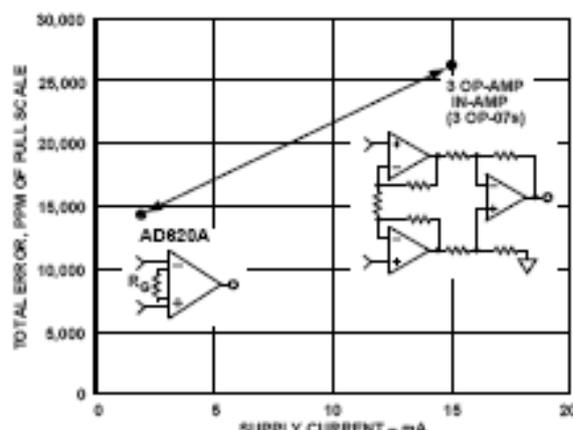


Figure 1. Three Op Amp IA Designs vs. AD620

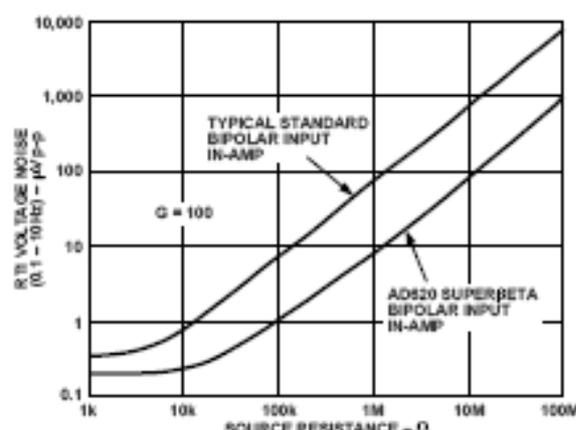


Figure 2. Total Voltage Noise vs. Source Resistance

REV. E

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 World Wide Web Site: <http://www.analog.com>
Fax: 781/326-8703 © Analog Devices, Inc., 1999

AD620—SPECIFICATIONS

(Typical @ +25°C, $V_S = \pm 15$ V, and $R_L = 2$ k Ω , unless otherwise noted)

| Model | Conditions | AD620A | | | AD620B | | | AD620S ¹ | | | Units |
|--|---|--------------|----------|--------------|--------------|----------|--------------|---------------------|----------|--------------|----------------|
| | | Min | Typ | Max | Min | Typ | Max | Min | Typ | Max | |
| GAIN | | | | | | | | | | | |
| Gain Range | $G = 1 + (49.4 \text{ k}\Omega/R_G)$ | 1 | | 10,000 | 1 | | 10,000 | 1 | | 10,000 | |
| Gain Error ² | $V_{OUT} = \pm 10$ V | | | | | | | | | | |
| G = 1 | | | 0.03 | 0.10 | | 0.01 | 0.02 | | 0.03 | 0.10 | % |
| G = 10 | | | 0.15 | 0.30 | | 0.10 | 0.15 | | 0.15 | 0.30 | % |
| G = 100 | | | 0.15 | 0.30 | | 0.10 | 0.15 | | 0.15 | 0.30 | % |
| G = 1000 | | | 0.40 | 0.70 | | 0.35 | 0.50 | | 0.40 | 0.70 | % |
| Nonlinearity, G = 1–1000 | $V_{OUT} = -10$ V to $+10$ V, $R_L = 10$ k Ω | | 10 | 40 | | 10 | 40 | | 10 | 40 | ppm |
| G = 1–100 | $R_L = 2$ k Ω | | 10 | 95 | | 10 | 95 | | 10 | 95 | ppm |
| Gain vs. Temperature | G = 1 | | | 10 | | | 10 | | | 10 | ppm/°C |
| | Gain > 1 ² | | | -50 | | | -50 | | | -50 | ppm/°C |
| VOLTAGE OFFSET | | | | | | | | | | | |
| Input Offset, V_{OSI} | (Total RTI Error = $V_{OSI} + V_{OSO}/G$) $V_S = \pm 5$ V to ± 15 V | | 30 | 125 | | 15 | 50 | | 30 | 125 | μ V |
| Over Temperature | $V_S = \pm 5$ V to ± 15 V | | | 185 | | | 85 | | | 225 | μ V |
| Average TC | $V_S = \pm 5$ V to ± 15 V | | 0.3 | 1.0 | | 0.1 | 0.6 | | 0.3 | 1.0 | μ V/°C |
| Output Offset, V_{OSO} | $V_S = \pm 15$ V | | 400 | 1000 | | 200 | 500 | | 400 | 1000 | μ V |
| Over Temperature | $V_S = \pm 5$ V | | | 1500 | | | 750 | | | 1500 | μ V |
| Average TC | $V_S = \pm 5$ V to ± 15 V | | 5.0 | 15 | | 2.5 | 7.0 | | 5.0 | 15 | μ V/°C |
| Offset Referred to the Input vs. Supply (PSR) | $V_S = \pm 2.3$ V to ± 18 V | | | | | | | | | | dB |
| G = 1 | | 80 | | 100 | 80 | | 100 | 80 | | 100 | dB |
| G = 10 | | 95 | | 120 | 100 | | 120 | 95 | | 120 | dB |
| G = 100 | | 110 | | 140 | 120 | | 140 | 110 | | 140 | dB |
| G = 1000 | | 110 | | 140 | 120 | | 140 | 110 | | 140 | dB |
| INPUT CURRENT | | | | | | | | | | | |
| Input Bias Current | | | 0.5 | 2.0 | | 0.5 | 1.0 | | 0.5 | 2 | nA |
| Over Temperature | | | | 2.5 | | | 1.5 | | | 4 | nA |
| Average TC | | | 3.0 | | | 3.0 | | | 8.0 | | pA/°C |
| Input Offset Current | | | 0.3 | 1.0 | | 0.3 | 0.5 | | 0.3 | 1.0 | nA |
| Over Temperature | | | | 1.5 | | | 0.75 | | | 2.0 | nA |
| Average TC | | | 1.5 | | | 1.5 | | | 8.0 | | pA/°C |
| INPUT | | | | | | | | | | | |
| Input Impedance | | | | | | | | | | | G Ω /pF |
| Differential | | | 10 | 2 | | 10 | 2 | | 10 | 2 | G Ω /pF |
| Common-Mode | | | 10 | 2 | | 10 | 2 | | 10 | 2 | G Ω /pF |
| Input Voltage Range ³ | $V_S = \pm 2.3$ V to ± 5 V | $-V_S + 1.9$ | | $+V_S - 1.2$ | $-V_S + 1.9$ | | $+V_S - 1.2$ | $-V_S + 1.9$ | | $+V_S - 1.2$ | V |
| Over Temperature | $V_S = \pm 5$ V to ± 18 V | $-V_S + 2.1$ | | $+V_S - 1.3$ | $-V_S + 2.1$ | | $+V_S - 1.3$ | $-V_S + 2.1$ | | $+V_S - 1.3$ | V |
| Over Temperature | | $-V_S + 1.9$ | | $+V_S - 1.4$ | $-V_S + 1.9$ | | $+V_S - 1.4$ | $-V_S + 1.9$ | | $+V_S - 1.4$ | V |
| Over Temperature | | $-V_S + 2.1$ | | $+V_S - 1.4$ | $-V_S + 2.1$ | | $+V_S - 1.4$ | $-V_S + 2.3$ | | $+V_S - 1.4$ | V |
| Common-Mode Rejection Ratio DC to 60 Hz with 1 k Ω Source Imbalance | $V_{CM} = 0$ V to ± 10 V | | 73 | 90 | | 80 | 90 | | 73 | 90 | dB |
| G = 1 | | | 93 | 110 | | 100 | 110 | | 93 | 110 | dB |
| G = 10 | | | 110 | 130 | | 120 | 130 | | 110 | 130 | dB |
| G = 1000 | | | 110 | 130 | | 120 | 130 | | 110 | 130 | dB |
| OUTPUT | | | | | | | | | | | |
| Output Swing | $R_L = 10$ k Ω , $V_S = \pm 2.3$ V to ± 5 V | $-V_S + 1.1$ | | $+V_S - 1.2$ | $-V_S + 1.1$ | | $+V_S - 1.2$ | $-V_S + 1.1$ | | $+V_S - 1.2$ | V |
| Over Temperature | $V_S = \pm 5$ V to ± 18 V | $-V_S + 1.4$ | | $+V_S - 1.3$ | $-V_S + 1.4$ | | $+V_S - 1.3$ | $-V_S + 1.6$ | | $+V_S - 1.3$ | V |
| Over Temperature | | $-V_S + 1.2$ | | $+V_S - 1.4$ | $-V_S + 1.2$ | | $+V_S - 1.4$ | $-V_S + 1.2$ | | $+V_S - 1.4$ | V |
| Over Temperature | | $-V_S + 1.6$ | | $+V_S - 1.5$ | $-V_S + 1.6$ | | $+V_S - 1.5$ | $-V_S + 2.3$ | | $+V_S - 1.5$ | V |
| Short Current Circuit | | | ± 18 | | | ± 18 | | | ± 18 | | mA |

DATASHEET SENSOR DE HUMEDAD HIH 4000 - 001

HIH-4000-001

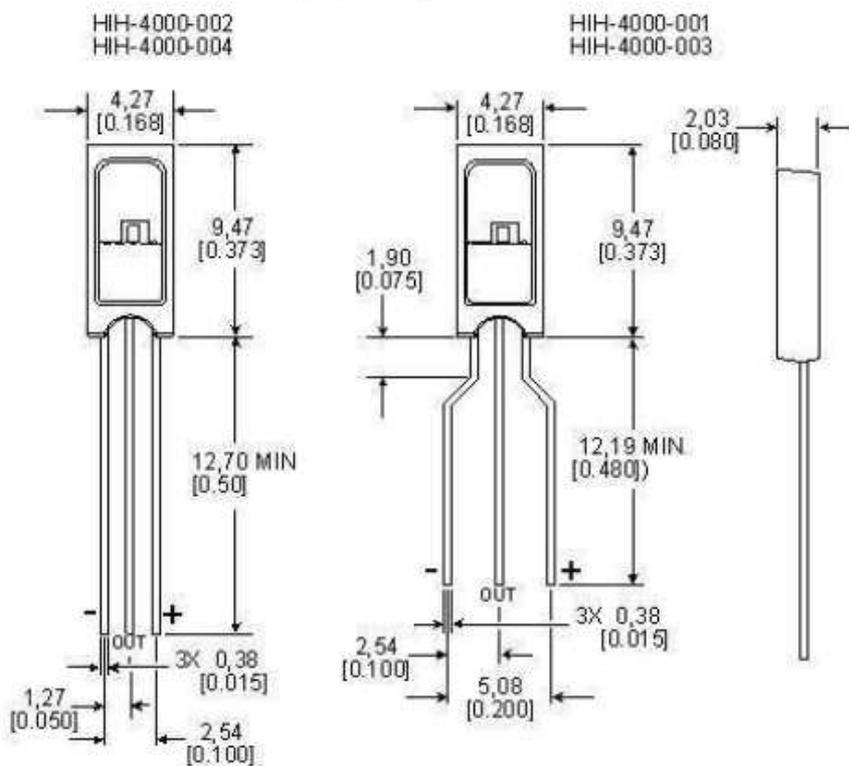
HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

| Product Specifications | |
|-----------------------------|--|
| Package Style | Solderable SIP |
| Termination Details | 2,54 mm [0.100 in] Lead Pitch |
| Series Name | HIH-4000 Series |
| RH Accuracy | ± 3.5% RH, 0-100 % RH non-condensing, 25 °C, 5 Vdc supply |
| RH Interchangeability | ± 5% RH, 0-60% RH; ± 8% @ 60-100% RH Typical |
| RH Hysteresis | ± 3% of RH Span Maximum |
| RH Repeatability | ± 0.5% RH |
| RH response time, 1/e | 15 s in slowly moving air @ 25 °C |
| RH Stability | ± 0.2% RH Typical at 50% RH in 1 Year |
| Supply Voltage | 4.0 Vdc to 5.8 Vdc |
| Supply Current | 500 µA Max. |
| Operating Humidity Range | 0 to 100% RH, non-condensing |
| Operating Temperature Range | -40 °C to 85 °C (-40 °F to 185 °F) |
| Temperature Compensation | True RH = Sensor RH/(1.0305+0.000044T-0.0000011T ²) T in °C (True RH = Sensor RH/(0.9237-0.0041T+0.000040T ²) T in °C) |
| Availability | Global |
| Comment | Light sensitive, shield from bright light. |
| UNSPSC Code | 411121 |
| UNSPSC Commodity | 411121 Transducers |

HIH-4000-001

HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

Mounting Dimensions
For Reference Only [mm/in]



DATASHEET SENSOR DE TEMPERATURA LM35

LM35 Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in $^{\circ}$ Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of $\pm 1/4^{\circ}\text{C}$ at room temperature and $\pm 1/2^{\circ}\text{C}$ over a full -55 to $+150^{\circ}\text{C}$ temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only $60\ \mu\text{A}$ from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a -55° to $+150^{\circ}\text{C}$ temperature range, while the LM35C is rated for a -40° to $+110^{\circ}\text{C}$ range (-10° with improved accuracy). The LM35 series is available pack-

aged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features

- Calibrated directly in $^{\circ}$ Celsius (Centigrade)
- Linear $+10.0\ \text{mV}/^{\circ}\text{C}$ scale factor
- 0.5°C accuracy guaranteeable (at $+25^{\circ}\text{C}$)
- Rated for full -55° to $+150^{\circ}\text{C}$ range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than $60\ \mu\text{A}$ current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only $\pm 1/4^{\circ}\text{C}$ typical
- Low impedance output, $0.1\ \Omega$ for $1\ \text{mA}$ load

Typical Applications

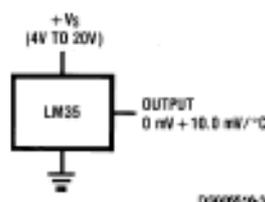
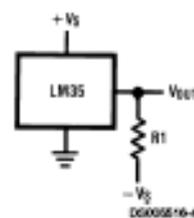


FIGURE 1. Basic Centigrade Temperature Sensor
($+2^{\circ}\text{C}$ to $+150^{\circ}\text{C}$)



Choose $R_1 = -V_S/50\ \mu\text{A}$
 $V_{\text{OUT}} = +1,500\ \text{mV}$ at $+150^{\circ}\text{C}$
 $= +250\ \text{mV}$ at $+25^{\circ}\text{C}$
 $= -550\ \text{mV}$ at -55°C

FIGURE 2. Full-Range Centigrade Temperature Sensor

Connection Diagrams

TO-46
Metal Can Package*



BOTTOM VIEW
DS000018-1

*Case is connected to negative pin (GND)

Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH

See NS Package Number H03H

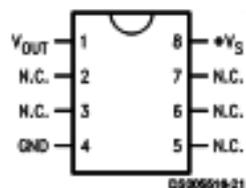
TO-92
Plastic Package



BOTTOM VIEW
DS000018-2

Order Number LM35CZ,
LM35CAZ or LM35DZ
See NS Package Number Z03A

SO-8
Small Outline Molded Package

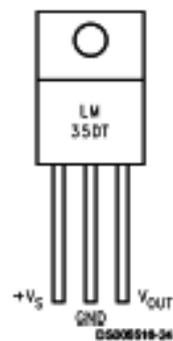


DS000018-3

N.C. = No Connection

Top View
Order Number LM35DM
See NS Package Number M08A

TO-220
Plastic Package*



DS000018-4

*Tab is connected to the negative pin (GND).

Note: The LM35DT pinout is different than the discontinued LM35DP.

Order Number LM35DT
See NS Package Number TA03F

Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|---|-----------------|
| Supply Voltage | +35V to -0.2V |
| Output Voltage | +6V to -1.0V |
| Output Current | 10 mA |
| Storage Temp.: | |
| TO-46 Package, | -60°C to +180°C |
| TO-92 Package, | -60°C to +150°C |
| SO-8 Package, | -65°C to +150°C |
| TO-220 Package, | -65°C to +150°C |
| Lead Temp.: | |
| TO-46 Package, (Soldering, 10 seconds) | 300°C |

| | |
|---|-----------------|
| TO-92 and TO-220 Package, (Soldering, 10 seconds) | 260°C |
| SO Package (Note 12) | |
| Vapor Phase (60 seconds) | 215°C |
| Infrared (15 seconds) | 220°C |
| ESD Susceptibility (Note 11) | 2500V |
| Specified Operating Temperature Range: T_{MIN} to T_{MAX} (Note 2) | |
| LM35, LM35A | -55°C to +150°C |
| LM35C, LM35CA | -40°C to +110°C |
| LM35D | 0°C to +100°C |

Electrical Characteristics

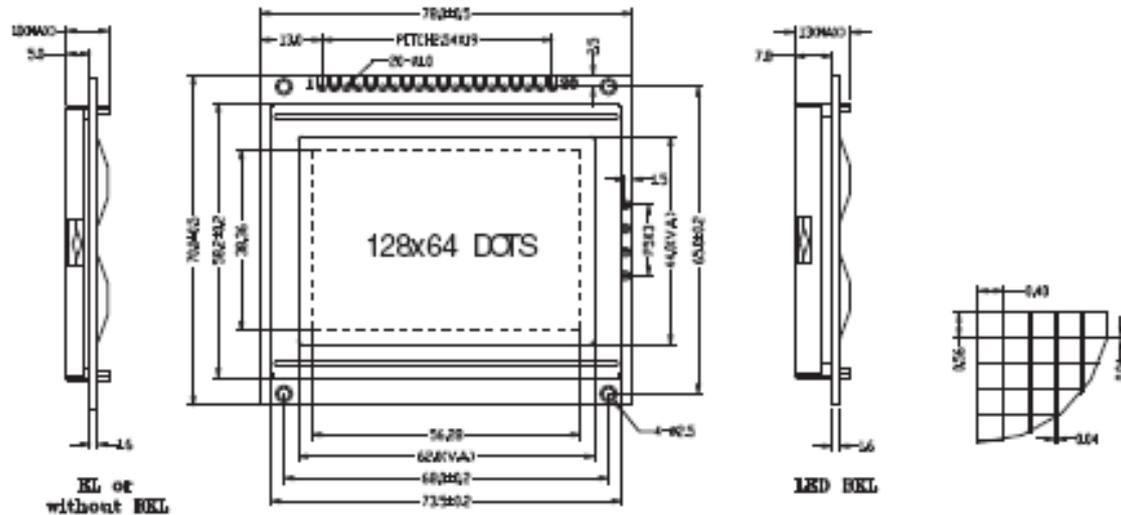
(Notes 1, 6)

| Parameter | Conditions | LM35A | | | LM35CA | | | Units (Max.) |
|--|---|------------|-----------------------------|-----------------------------|------------|-----------------------------|-----------------------------|------------------------------|
| | | Typical | Tested Limit (Note 4) | Design Limit (Note 5) | Typical | Tested Limit (Note 4) | Design Limit (Note 5) | |
| Accuracy (Note 7) | $T_A = +25^\circ\text{C}$ | ± 0.2 | ± 0.5 | | ± 0.2 | ± 0.5 | ± 1.0 | $^\circ\text{C}$ |
| | $T_A = -10^\circ\text{C}$ | ± 0.3 | | | ± 0.3 | | ± 1.0 | $^\circ\text{C}$ |
| | $T_A = T_{MAX}$ | ± 0.4 | ± 1.0 | | ± 0.4 | ± 1.0 | | $^\circ\text{C}$ |
| | $T_A = T_{MIN}$ | ± 0.4 | ± 1.0 | | ± 0.4 | | ± 1.5 | $^\circ\text{C}$ |
| Nonlinearity (Note 8) | $T_{MIN} \leq T_A \leq T_{MAX}$ | ± 0.18 | | ± 0.35 | ± 0.15 | | ± 0.3 | $^\circ\text{C}$ |
| Sensor Gain (Average Slope) | $T_{MIN} \leq T_A \leq T_{MAX}$ | +10.0 | +9.9, +10.1 | | +10.0 | | +9.9, +10.1 | mV/ $^\circ\text{C}$ |
| Load Regulation (Note 3) $0 \leq I_L \leq 1$ mA | $T_A = +25^\circ\text{C}$ | ± 0.4 | ± 1.0 | | ± 0.4 | ± 1.0 | | mV/mA |
| | $T_{MIN} \leq T_A \leq T_{MAX}$ | ± 0.5 | | ± 3.0 | ± 0.5 | | ± 3.0 | mV/mA |
| Line Regulation (Note 3) | $T_A = +25^\circ\text{C}$ | ± 0.01 | ± 0.05 | | ± 0.01 | ± 0.05 | | mV/V |
| | $4\text{V} \leq V_O \leq 30\text{V}$ | ± 0.02 | | ± 0.1 | ± 0.02 | | ± 0.1 | mV/V |
| Quiescent Current (Note 9) | $V_O = +5\text{V}, +25^\circ\text{C}$ | 56 | 67 | | 56 | 67 | | μA |
| | $V_O = +5\text{V}$ | 105 | | 131 | 91 | | 114 | μA |
| | $V_O = +30\text{V}, +25^\circ\text{C}$ | 56.2 | 68 | | 56.2 | 68 | | μA |
| | $V_O = +30\text{V}$ | 105.5 | | 133 | 91.5 | | 116 | μA |
| Change of Quiescent Current (Note 3) | $4\text{V} \leq V_O \leq 30\text{V}, +25^\circ\text{C}$ | 0.2 | 1.0 | | 0.2 | 1.0 | | μA |
| | $4\text{V} \leq V_O \leq 30\text{V}$ | 0.5 | | 2.0 | 0.5 | | 2.0 | μA |
| Temperature Coefficient of Quiescent Current | | +0.39 | | +0.5 | +0.39 | | +0.5 | $\mu\text{A}/^\circ\text{C}$ |
| Minimum Temperature for Rated Accuracy | In circuit of Figure 1, $I_L = 0$ | +1.5 | | +2.0 | +1.5 | | +2.0 | $^\circ\text{C}$ |
| Long Term Stability | $T_J = T_{MAX}$, for 1000 hours | ± 0.08 | | | ± 0.08 | | | $^\circ\text{C}$ |

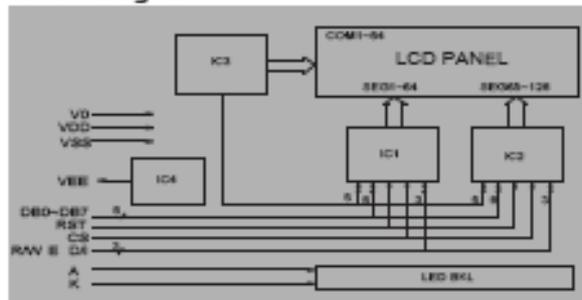
DATASHEET GLCD GDM12864B



Outline Dimension



Block diagram



Interface pin connections

| PIN NO | Symbol | Function |
|---------|------------|----------------------------|
| 1 | /CSA | Chip select for IC1 |
| 2 | /CSB | Chip select for IC2 |
| 3 | VSS □ | GND |
| 4 | VDD □ | +5V |
| 5 | V0 □ | Constrast adjustment |
| 6 | D/I | H/L Register select signal |
| 7 | R/W | H/L Read/Write signal |
| 8 | E | H/L Enable signal |
| 9 to 18 | D80 to D87 | H/L Data bus line |
| 17 | RST | Reset signal |
| 18 | VEE | Negative voltage output |
| 19 | A | Power supply for BKL(4.2V) |
| 20 | K | Power supply for BKL(GND) |

Feature

1. 128x64 dots graphic LCD module
2. Built-in controller (KS0108B)
3. 5.0V power supply
4. STN; 1/64 duty; LED BKL or EL BKL

Mechanical Data

| Item | Standard | Unit |
|------------------|-----------|------|
| Module dimension | 78.0x70.0 | mm |
| Viewing area | 62.0x44.0 | mm |
| Dot size | 0.40x0.56 | mm |
| Dot pitch | 0.44x0.60 | mm |

Absolute Maximum Rating

| Item | Symbol | Standard | | | Unit |
|---------------|---------|----------|-------|-----|------|
| | | Min | Typ | Max | |
| Power supply | VDD-VSS | -0.3 | ----- | 5.5 | V |
| Input voltage | VI | -0.3 | ----- | VDD | |

Electronical characteristics

| Item | Symbol | Condition | Standard | | | Unit |
|--|-----------------|-----------------------------------|----------|-------|-------|------|
| | | | Min | Typ | Max | |
| Input voltage | VDD | +5V | 4.7 | 5.0 | 5.5 | V |
| Supply current | I _{DD} | VDD=5V | ----- | 8 | ----- | mA |
| Recommended LCD driving voltage for normal temp version module | VDD-V0 | -20°C | ----- | ----- | ----- | V |
| | | 0 °C | ----- | 9.8 | ----- | |
| | | 25°C | ----- | 9.5 | ----- | |
| | | 50°C | ----- | 9.3 | ----- | |
| LED forward voltage | V _F | 25°C | ----- | 4.2 | 4.5 | V |
| LED forward current | I _F | 25°C | ----- | 360 | ----- | mA |
| EL power supply current | I _{EL} | V _{EL} =110V AC 400Hz | ----- | ----- | ----- | mA |

DATASHEET DEL SENSOR DE PRESION MPX10D



10 kPa Uncompensated Silicon Pressure Sensors

The MPX10 series device is a silicon piezoresistive pressure sensor providing a very accurate and linear voltage output — directly proportional to the applied pressure. This standard, low cost, uncompensated sensor permits manufacturers to design and add their own external temperature compensating and signal conditioning networks. Compensation techniques are simplified because of the predictability of Motorola's single element strain gauge design.

Features

- Low Cost
- Patented Silicon Shear Stress Strain Gauge Design
- Ratiometric to Supply Voltage
- Easy to Use Chip Carrier Package Options
- Differential and Gauge Options

Application Examples

- Air Movement Control
- Environmental Control Systems
- Level Indicators
- Leak Detection
- Medical Instrumentation
- Industrial Controls
- Pneumatic Control Systems
- Robotics

Figure 1 shows a schematic of the internal circuitry on the stand-alone pressure sensor chip.

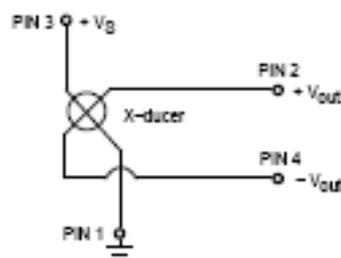


Figure 1. Uncompensated Pressure Sensor Schematic

VOLTAGE OUTPUT versus APPLIED DIFFERENTIAL PRESSURE

The differential voltage output of the X-ducer is directly proportional to the differential pressure applied.

The output voltage of the differential or gauge sensor increases with increasing pressure applied to the pressure side (P1) relative to the vacuum side (P2). Similarly, output voltage increases as increasing vacuum is applied to the vacuum side (P2) relative to the pressure side (P1).

Senseon and X-ducer are trademarks of Motorola, Inc.

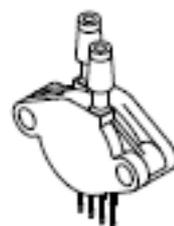
REV 5

MPX10 SERIES

0 to 10 kPa (0–1.45 psi)
35 mV FULL SCALE SPAN
(TYPICAL)



BASIC CHIP
CARRIER ELEMENT
CASE 344-16, STYLE 1



DIFFERENTIAL
PORT OPTION
CASE 344C-01, STYLE 1

NOTE: Pin 1 is the notched pin.

| PIN NUMBER | | | |
|------------|------------|---|------------|
| 1 | Gnd | 3 | V_B |
| 2 | $+V_{out}$ | 4 | $-V_{out}$ |



MOTOROLA

MPX10 SERIES

MAXIMUM RATINGS

| Rating | Symbol | Value | Unit |
|---|--------------------|-------------|------|
| Overpressure ⁽⁸⁾ (P1 > P2) | F _{max} | 75 | kPa |
| Burst Pressure ⁽⁸⁾ (P1 > P2) | F _{burst} | 100 | kPa |
| Storage Temperature | T _{stg} | -40 to +125 | °C |
| Operating Temperature | T _A | -40 to +125 | °C |

OPERATING CHARACTERISTICS (V_B = 3.0 Vdc, T_A = 25°C unless otherwise noted, P1 > P2)

| Characteristic | Symbol | Min | Typ | Max | Unit |
|---|--------------------|-------|------|-------|-----------------------|
| Differential Pressure Range ⁽¹⁾ | P _{OP} | 0 | — | 10 | kPa |
| Supply Voltage ⁽²⁾ | V _B | — | 3.0 | 6.0 | Vdc |
| Supply Current | I _B | — | 6.0 | — | mAdc |
| Full Scale Span ⁽³⁾ | V _{FSS} | 20 | 35 | 50 | mV |
| Offset ⁽⁴⁾ | V _{OFF} | 0 | 20 | 35 | mV |
| Sensitivity | ΔV/ΔP | — | 3.5 | — | mV/kPa |
| Linearity ⁽⁵⁾ | — | -1.0 | — | 1.0 | %V _{FSS} |
| Pressure Hysteresis ⁽⁵⁾ (0 to 10 kPa) | — | — | ±0.1 | — | %V _{FSS} |
| Temperature Hysteresis ⁽⁵⁾ (-40°C to +125°C) | — | — | ±0.5 | — | %V _{FSS} |
| Temperature Coefficient of Full Scale Span ⁽⁵⁾ | TCV _{FSS} | -0.22 | — | -0.16 | %V _{FSS} /°C |
| Temperature Coefficient of Offset ⁽⁵⁾ | TCV _{OFF} | — | ±15 | — | μV/°C |
| Temperature Coefficient of Resistance ⁽⁵⁾ | TCR | 0.21 | — | 0.27 | %Z _{IN} /°C |
| Input Impedance | Z _{IN} | 400 | — | 550 | Ω |
| Output Impedance | Z _{OUT} | 750 | — | 1250 | Ω |
| Response Time ⁽⁶⁾ (10% to 90%) | t _R | — | 1.0 | — | ms |
| Warm-Up | — | — | 20 | — | ms |
| Offset Stability ⁽⁹⁾ | — | — | ±0.5 | — | %V _{FSS} |

MECHANICAL CHARACTERISTICS

| Characteristic | Symbol | Min | Typ | Max | Unit |
|--|--------|-----|-----|-----|-------|
| Weight (Basic Element, Case 344-15) | — | — | 2.0 | — | Grams |
| Common Mode Line Pressure ⁽⁷⁾ | — | — | — | 690 | kPa |

NOTES:

- 1.0 kPa (kiloPascal) equals 0.145 psi.
- Device is ratiometric within this specified excitation range. Operating the device above the specified excitation range may induce additional error due to device self-heating.
- Full Scale Span (V_{FSS}) is defined as the algebraic difference between the output voltage at full rated pressure and the output voltage at the minimum rated pressure.
- Offset (V_{OFF}) is defined as the output voltage at the minimum rated pressure.
- Accuracy (error budget) consists of the following:
 - Linearity: Output deviation from a straight line relationship with pressure, using end point method, over the specified pressure range.
 - Temperature Hysteresis: Output deviation at any temperature within the operating temperature range, after the temperature is cycled to and from the minimum or maximum operating temperature points, with zero differential pressure applied.
 - Pressure Hysteresis: Output deviation at any pressure within the specified range, when this pressure is cycled to and from the minimum or maximum rated pressure, at 25°C.
 - TcSpan: Output deviation at full rated pressure over the temperature range of 0 to 85°C, relative to 25°C.
 - TcOffset: Output deviation with minimum rated pressure applied, over the temperature range of 0 to 85°C, relative to 25°C.
 - TCR: Z_{IN} deviation with minimum rated pressure applied, over the temperature range of -40°C to +125°C, relative to 25°C.
- Response Time is defined as the time for the incremental change in the output to go from 10% to 90% of its final value when subjected to a specified step change in pressure.
- Common mode pressures beyond specified may result in leakage at the case-to-lead interface.
- Exposure beyond these limits may cause permanent damage or degradation to the device.
- Offset stability is the product's output deviation when subjected to 1000 hours of Pulsed Pressure, Temperature Cycling with Bias Test.

DATASHEET PIC18F4550



MICROCHIP PIC18F2455/2550/4455/4550

28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology

Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- 1-Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

Power-Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8 μ A typical
- Sleep mode currents down to 0.1 μ A typical
- Timer1 Oscillator: 1.1 μ A typical, 32 kHz, 2V
- Watchdog Timer: 2.1 μ A typical
- Two-Speed Oscillator Start-up

Flexible Oscillator Structure:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns ($T_{CV}/16$)
 - Compare is 16-bit, max. resolution 83.3 ns (T_{CV})
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I²C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

Special Microcontroller Features:

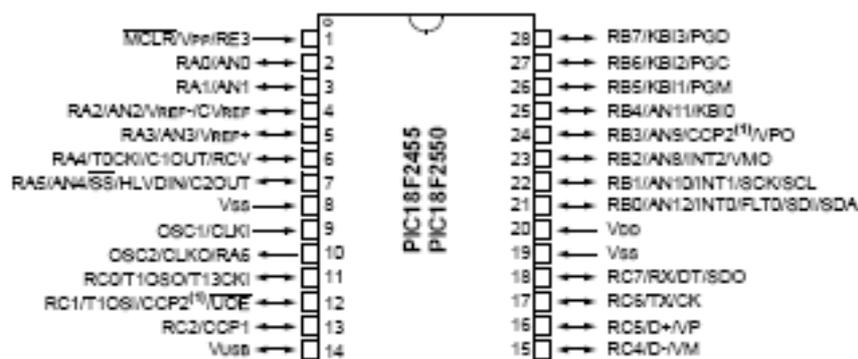
- C Compiler Optimized Architecture with optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
 - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Optional dedicated ICD/ICSP port (44-pin devices only)
- Wide Operating Voltage Range (2.0V to 5.5V)

| Device | Program Memory | | Data Memory | | I/O | 10-Bit A/D (ch) | CCP/ECCP (PWM) | SPP | MSSP | | EA/USART | Comparators | Timers 8/16-Bit |
|------------|----------------|----------------------------|--------------|----------------|-----|-----------------|----------------|-----|------|--------------------------|----------|-------------|-----------------|
| | Flash (bytes) | # Single-Word Instructions | SRAM (bytes) | EEPROM (bytes) | | | | | SPI | Master I ² C™ | | | |
| PIC18F2455 | 24K | 12288 | 2048 | 256 | 24 | 10 | 2/0 | No | Y | Y | 1 | 2 | 1/3 |
| PIC18F2550 | 32K | 16384 | 2048 | 256 | 24 | 10 | 2/0 | No | Y | Y | 1 | 2 | 1/3 |
| PIC18F4455 | 24K | 12288 | 2048 | 256 | 35 | 13 | 1/1 | Yes | Y | Y | 1 | 2 | 1/3 |
| PIC18F4550 | 32K | 16384 | 2048 | 256 | 35 | 13 | 1/1 | Yes | Y | Y | 1 | 2 | 1/3 |

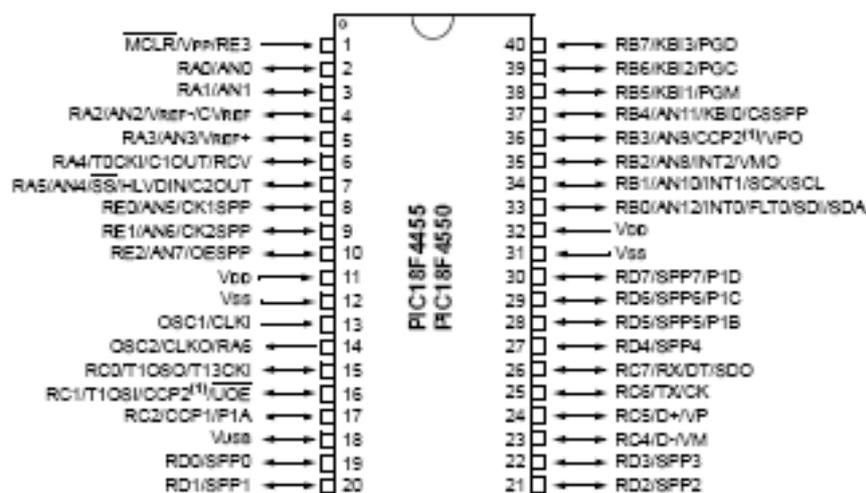
PIC18F2455/2550/4455/4550

Pin Diagrams

28-Pin PDIP, SOIC



40-Pin PDIP



Note 1: RB3 is the alternate pin for CCP2 multiplexing.

PIC18F2455/2550/4455/4550

1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- PIC18F2455
- PIC18F2550
- PIC18F4455
- PIC18F4550
- PIC18LF2455
- PIC18LF2550
- PIC18LF4455
- PIC18LF4550

This family of devices offers the advantages of all PIC18 microcontrollers – namely, high computational performance at an economical price – with the addition of high endurance, Enhanced Flash program memory. In addition to these features, the PIC18F2455/2550/4455/4550 family introduces design enhancements that make these microcontrollers a logical choice for many high-performance, power sensitive applications.

1.1 New Core Features

1.1.1 nanoWatt TECHNOLOGY

All of the devices in the PIC18F2455/2550/4455/4550 family incorporate a range of features that can significantly reduce power consumption during operation. Key items include:

- **Alternate Run Modes:** By clocking the controller from the Timer1 source or the internal oscillator block, power consumption during code execution can be reduced by as much as 90%.
- **Multiple Idle Modes:** The controller can also run with its CPU core disabled but the peripherals still active. In these states, power consumption can be reduced even further, to as little as 4% of normal operation requirements.
- **On-the-Fly Mode Switching:** The power-managed modes are invoked by user code during operation, allowing the user to incorporate power-saving ideas into their application's software design.
- **Low Consumption in Key Modules:** The power requirements for both Timer1 and the Watchdog Timer are minimized. See Section 28.0 "Electrical Characteristics" for values.

1.1.2 UNIVERSAL SERIAL BUS (USB)

Devices in the PIC18F2455/2550/4455/4550 family incorporate a fully featured Universal Serial Bus communications module that is compliant with the USB Specification Revision 2.0. The module supports both low-speed and full-speed communication for all supported data transfer types. It also incorporates its own on-chip transceiver and 3.3V regulator and supports the use of external transceivers and voltage regulators.

1.1.3 MULTIPLE OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC18F2455/2550/4455/4550 family offer twelve different oscillator options, allowing users a wide range of choices in developing application hardware. These include:

- Four Crystal modes using crystals or ceramic resonators.
- Four External Clock modes, offering the option of using two pins (oscillator input and a divide-by-4 clock output) or one pin (oscillator input, with the second pin reassigned as general I/O).
- An internal oscillator block which provides an 8 MHz clock ($\pm 2\%$ accuracy) and an INTRC source (approximately 31 kHz, stable over temperature and V_{DD}), as well as a range of 6 user-selectable clock frequencies, between 125 kHz to 4 MHz, for a total of 8 clock frequencies. This option frees an oscillator pin for use as an additional general purpose I/O.
- A Phase Lock Loop (PLL) frequency multiplier, available to both the High-Speed Crystal and External Oscillator modes, which allows a wide range of clock speeds from 4 MHz to 48 MHz.
- Asynchronous dual clock operation, allowing the USB module to run from a high-frequency oscillator while the rest of the microcontroller is clocked from an internal low-power oscillator.

Besides its availability as a clock source, the internal oscillator block provides a stable reference source that gives the family additional features for robust operation:

- **Fail-Safe Clock Monitor:** This option constantly monitors the main clock source against a reference signal provided by the internal oscillator. If a clock failure occurs, the controller is switched to the internal oscillator block, allowing for continued low-speed operation or a safe application shutdown.
- **Two-Speed Start-up:** This option allows the internal oscillator to serve as the clock source from Power-on Reset, or wake-up from Sleep mode, until the primary clock source is available.

PIC18F2455/2550/4455/4550

1.2 Other Special Features

- **Memory Endurance:** The Enhanced Flash cells for both program memory and data EEPROM are rated to last for many thousands of erase/write cycles – up to 100,000 for program memory and 1,000,000 for EEPROM. Data retention without refresh is conservatively estimated to be greater than 40 years.
- **Self-Programmability:** These devices can write to their own program memory spaces under internal software control. By using a bootloader routine, located in the protected Boot Block at the top of program memory, it becomes possible to create an application that can update itself in the field.
- **Extended Instruction Set:** The PIC18F2455/2550/4455/4550 family introduces an optional extension to the PIC18 instruction set, which adds 8 new instructions and an Indexed Literal Offset Addressing mode. This extension, enabled as a device configuration option, has been specifically designed to optimize re-entrant application code originally developed in high-level languages such as C.
- **Enhanced CCP Module:** In PWM mode, this module provides 1, 2 or 4 modulated outputs for controlling half-bridge and full-bridge drivers. Other features include auto-shutdown for disabling PWM outputs on interrupt or other select conditions and auto-restart to reactivate outputs once the condition has cleared.
- **Enhanced Addressable USART:** This serial communication module is capable of standard RS-232 operation and provides support for the LIN bus protocol. Other enhancements include Automatic Baud Rate Detection and a 16-bit Baud Rate Generator for improved resolution. When the microcontroller is using the internal oscillator block, the EUSART provides stable operation for applications that talk to the outside world without using an external crystal (or its accompanying power requirement).
- **10-Bit A/D Converter:** This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated, without waiting for a sampling period and thus, reducing code overhead.
- **Dedicated ICD/ICSP Port:** These devices introduce the use of debugger and programming pins that are not multiplexed with other microcontroller features. Offered as an option in select packages, this feature allows users to develop I/O intensive applications while retaining the ability to program and debug in the circuit.

1.3 Details on Individual Family Members

Devices in the PIC18F2455/2550/4455/4550 family are available in 28-pin and 40/44-pin packages. Block diagrams for the two groups are shown in Figure 1-1 and Figure 1-2.

The devices are differentiated from each other in six ways:

1. Flash program memory (24 Kbytes for PIC18FX455 devices, 32 Kbytes for PIC18FX550).
2. A/D channels (10 for 28-pin devices, 13 for 40/44-pin devices).
3. I/O ports (3 bidirectional ports and 1 input only port on 28-pin devices, 5 bidirectional ports on 40/44-pin devices).
4. CCP and Enhanced CCP implementation (28-pin devices have two standard CCP modules, 40/44-pin devices have one standard CCP module and one EOCPP module).
5. Streaming Parallel Port (present only on 40/44-pin devices).

All other features for devices in this family are identical. These are summarized in Table 1-1.

The pinouts for all devices are listed in Table 1-2 and Table 1-3.

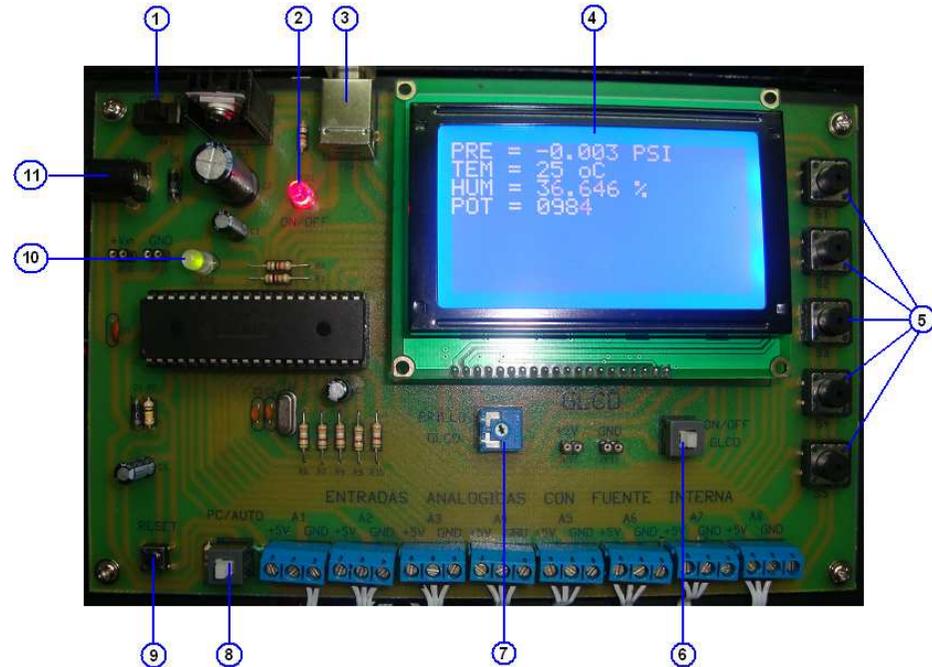
Like all Microchip PIC18 devices, members of the PIC18F2455/2550/4455/4550 family are available as both standard and low-voltage devices. Standard devices with Enhanced Flash memory, designated with an "F" in the part number (such as PIC18F2550), accommodate an operating V_{DD} range of 4.2V to 5.5V. Low-voltage parts, designated by "LF" (such as PIC18LF2550), function over an extended V_{DD} range of 2.0V to 5.5V.

ANEXO C: MANUAL DE USUARIO

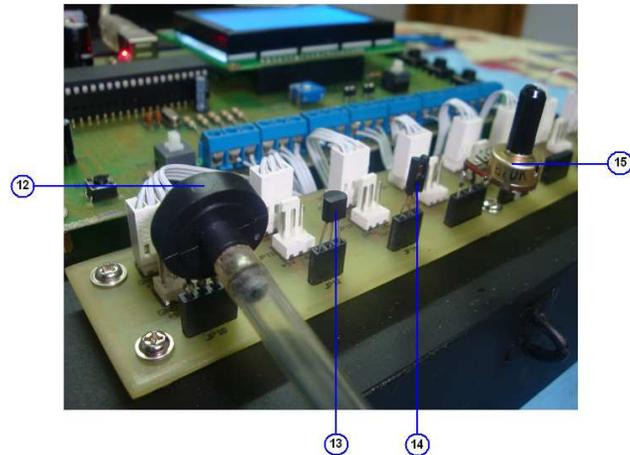
El sistema de adquisición de datos conforma un sistema de chequeo de variables físicas muy fácil de utilizar, puede ser manejado por el mismo usuario y no requiere de personal calificado.

1.- CARACTERISTICAS DEL SISTEMA

- Chequeo de 4 variables físicas por default a través de sensores, de las cuales una es simulada por un potenciómetro.
- Visualización de resultados de chequeo de dos maneras: en modulo GLCD y en pantalla de computador (interfaz grafica).
- Posee comunicación USB hacia el computador para visualizar los resultados de chequeo en el mismo con la ayuda de la interfaz grafica (USB1).
- Posee pulsadores para elegir opciones de visualización de resultados, tanto en el GLCD como en la pantalla del computador (S1, S2, S3, S4, S5).
- Selección de modo de funcionamiento automático ó con ayuda de un computador (PC/AUTO).
- Selección de apagado y encendido de luz de fondo del GLCD (ON/OFF GLCD).
- Selección de intensidad de brillo de luz de fondo del GLCD (BRILLO GLCD).
- Selección de reset del circuito (RESET).
- Posee terminales para extender alcance de lugar de trabajo de los sensores.



- 1: Interruptor de Encendido y apagado ON/OFF.
- 2: LED indicador de encendido y apagado (LED1).
- 3: Puerto de comunicación USB (USB1).
- 4: Módulo GLCD.
- 5: Pulsadores de opciones de visualización.
- 6: Switch de encendido y apagado de luz de fondo GLCD (ON/OFF GLCD).
- 7: Control de intensidad de brillo de luz de fondo del GLCD (BRILLO GLCD).
- 8: Switch de modo de funcionamiento (PC/AUTO).
- 9: Pulsador de reset del circuito (RESET).
- 10: LED indicador de modo de funcionamiento (LED2).
- 11: Conector de energía tipo hembra.



12: Sensor de presión.

13: Sensor de temperatura.

14: Sensor de Humedad.

15: Sensor simulado con potenciómetro.



16: Terminales de extensión de alcance de los sensores.

2.- MODOS DE FUNCIONAMIENTO.

Funcionamiento Autónomo:

Para poder trabajar con el sistema de adquisición de datos en este modo, debemos observar que se cumplan los siguientes parámetros:

- El interruptor de encendido y apagado debe estar en ON.
- LED1 (ON/OFF) debe emitir luz roja
- LED2 debe emitir luz verde
- Switch PC/AUTO debe estar ON.

En este caso queda a libre albedrío del usuario según lo requiera, la manipulación del control BRILLO GLCD, el switch ON/OFF GLCD y la conexión o no del cable UBS al puerto USB1.



Los datos de las variables físicas medidas se visualizaran en el GLCD tal y como lo muestra la figura anterior, ya que por default de programación aparecen solo 4 variables medidas.

Funcionamiento con PC:

Al igual que en el caso anterior, para poder trabajar con el sistema de adquisición de datos en este modo, debemos observar que se cumplan los siguientes parámetros:

- Conectar el cable USB en sus 2 extremos, tanto hacia el circuito en el puerto USB1, como al puerto USB de la computadora.
- El interruptor de encendido y apagado debe estar en ON.
- LED1 (ON/OFF) debe emitir luz roja
- LED2 debe emitir luz roja.
- Switch PC/AUTO debe estar OFF.
- Debemos dar un pulso de RESET para empezar a trabajar en este modo de funcionamiento.

En este caso debemos utilizar la interfaz grafica generada en Visual Basic para poder visualizar los datos de las variables medidas.



1: Indicador de transmisión de datos desde el circuito.

2: Ventana de selección de canal de entrada que deseo visualizar.

3: Valor digitalizado (binario) del canal de entrada seleccionado.

4: Opción de inicio de visualización del valor adquirido por el canal.

5: Área de visualización del valor adquirido por el canal (similar a osciloscopio).

6: Opción de salida de la interfaz grafica.

7: Ventanas de visualización de los valores en decimal con sus respectivas unidades, de los valores adquiridos por cualquiera de los canales.

La visualización de los valores adquiridos en cada canal de entrada del circuito esta representada en voltios. Cabe

resaltar que para este caso, los datos se visualizan también en el GLCD simultáneamente.

Los datos de las variables físicas medidas se visualizaran en la interfaz grafica tal y como lo muestra la siguiente figura:

3.- OPCIONES DE VISUALIZACION EN EL GLCD

El sistema, tiene varias opciones de visualizar los datos en el GLCD, las cuales están determinadas por el orden y el número de veces que activemos los pulsadores (S1, S2, S3, S4, S5).

Las opciones de visualización son las siguientes:

Opción 1:

En este caso utilizamos los pulsadores S1, S2 y S3, para poder visualizar en el GLCD un cierto número de variables medidas, según lo requiera el usuario.

Paso1: Al pulsar por primera vez S3, se visualiza en el GLCD lo siguiente:



S1 y S3 nos sirven para aumentar y disminuir respectivamente el número de variables medidas que se visualizan en el GLCD.

Paso 2: Si pulsamos S1 una vez, subiremos a 5 el valor # IN's que se muestra en el GLCD, y se visualizará lo siguiente:



Paso 3: Al pulsar por segunda ocasión S3, podremos visualizar en el GLCD 5 valores de las variables medidas (similar a dar un ENTER).



Si queremos disminuir el número de variables medidas que se visualizan en el GLCD, solo debemos pulsar S2 en el paso 2 y continuar con el paso 3.

Opción 2:

En este caso utilizamos los pulsadores S4 y S5, para poder visualizar de manera completa en el GLCD una sola variable medida, según lo requiera el usuario.

Paso1: Al pulsar S5 por primera vez, se visualiza en el GLCD lo siguiente:



Paso 2: Pulsamos S4 para elegir de manera ascendente cual es la variable medida que se visualizará en el GLCD.

Paso 3: Al pulsar por segunda ocasión S5, podremos visualizar en el GLCD un solo valor de las variable medid (similar a dar un ENTER), según lo requiera el usuario.

A continuación detalle una secuencia de elecciones que pueden suceder.

Variable Elegida:



Visualización en el GLCD:



4.- SOLUCION DE PROBLEMAS.

| PREGUNTA | RESPONDER |
|--|--|
| ¿Qué hacer cuando los datos no se visualizan en el GLCD? | Revisar la conexión de los sensores, con el circuito apagado. |
| ¿Qué hacer cuando mi sistema no funciona correctamente? | <p>Verificar que tanto el microcontrolador como el GLCD se encuentren bien posicionados sobre sus zócalos.</p> <p>Revisar que los switch se encuentre en su estado lógico de trabajo para un cierto modo u otro.</p> <p>Comprobar que los Sensores no se encuentren averiados.</p> |
| ¿Funciona el sistema si no tengo energía eléctrica de 110 Vac? | No funciona. |
| ¿Puedo cambiar algún elemento dañado a averiado del sistema? | Se puede realizar el cambio de algún elemento tomando en cuenta las características del mismo, ya que todos los elementos utilizados son comerciales. Por ejemplo: los circuitos integrados, los sensores, las fuentes de energía DC. |