

# **ESCUELA POLITÉCNICA DEL EJÉRCITO**



**SEDE LATACUNGA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA E  
INSTRUMENTACIÓN**

**DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DATA LOGGER  
PARA REGISTROS DE DATOS OBTENIDOS DE VARIABLES  
ANALÓGICAS Y/O DIGITALES MEDIANTE EL MÓDULO USB DEL  
PIC18F2550 Y EL SOFTWARE LABVIEW PARA COMUNICACIÓN  
CON UN PC**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA E INSTRUMENTACIÓN**

**RIVERA FÁREZ JORGE LUIS**

**Latacunga, Septiembre 2010**

ESCUELA POLITÉCNICA DEL EJÉRCITO  
CARRERA DE INGENIERÍA EN ELECTRÓNICA E  
INSTRUMENTACIÓN

**DECLARACIÓN DE RESPONSABILIDAD**

Yo, RIVERA FÁREZ JORGE LUIS

**DECLARO QUE:**

El proyecto de grado denominado "DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DATA LOGGER PARA REGISTROS DE DATOS OBTENIDOS DE VARIABLES ANALÓGICAS Y/O DIGITALES MEDIANTE EL MÓDULO USB DEL PIC18F2550 Y EL SOFTWARE LABVIEW PARA COMUNICACIÓN CON UN PC" ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, 5 de Septiembre del 2010.

-----  
RIVERA FÁREZ JORGE LUIS  
CI. No.- 180300252-4

ESCUELA POLITÉCNICA DEL EJÉRCITO  
CARRERA DE INGENIERÍA EN ELECTRÓNICA E  
INSTRUMENTACIÓN

**AUTORIZACIÓN**

Yo, RIVERA FÁREZ JORGE LUIS

Autorizo a la Escuela Politécnica del Ejército la publicación, en la biblioteca virtual de la Institución del trabajo "DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DATA LOGGER PARA REGISTROS DE DATOS OBTENIDOS DE VARIABLES ANALÓGICAS Y/O DIGITALES MEDIANTE EL MÓDULO USB DEL PIC18F2550 Y EL SOFTWARE LABVIEW PARA COMUNICACIÓN CON UN PC" cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Latacunga, 5 de Septiembre del 2010.

-----  
RIVERA FÁREZ JORGE LUIS  
CI. No.- 180300252-4

ESCUELA POLITÉCNICA DEL EJÉRCITO  
CARRERA DE INGENIERÍA EN ELECTRÓNICA E  
INSTRUMENTACIÓN

**CERTIFICADO**

ING. JULIO ACOSTA NÚÑEZ (DIRECTOR)

ING. AMPARO MEYTHALER NARANJO (CODIRECTOR)

**CERTIFICAN:**

Que el trabajo titulado "DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DATA LOGGER PARA REGISTROS DE DATOS OBTENIDOS DE VARIABLES ANALÓGICAS Y/O DIGITALES MEDIANTE EL MÓDULO USB DEL PIC18F2550 Y EL SOFTWARE LABVIEW PARA COMUNICACIÓN CON UN PC" realizado por el señor: RIVERA FÁREZ JORGE LUIS ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Escuela Politécnica del Ejército.

Debido a que constituye un trabajo de excelente contenido científico que coadyuvará a la aplicación de conocimientos y al desarrollo profesional, **SI** recomiendan su publicación.

El mencionado trabajo consta de UN empastado y UN disco compacto el cual contiene los archivos en formato portátil de Acrobat. Autorizan al señor: RIVERA FÁREZ JORGE LUIS que lo entregue al ING. ARMANDO ÁLVAREZ SALAZAR, en su calidad de Director de Carrera.

Latacunga, 5 de Septiembre del 2010.

-----  
Ing. Julio Acosta Núñez  
**DIRECTOR**

-----  
Ing. Amparo Meythaler Naranjo  
**CODIRECTOR**

## INTRODUCCIÓN

El presente trabajo tiene como objetivo principal el diseño de un módulo data logger, con interfaz USB 2.0.

El módulo consta de una interfaz gráfica desarrollada en LabVIEW, permite monitorear en tiempo real dos entradas analógicas y cuatro entradas digitales, ésta interfaz es la que permite cambiar parámetros de funcionamiento del data logger como el tiempo de adquisición, que es la frecuencia con que guarda en memoria no volátil los datos recolectados en ese instante, el tiempo puede variar desde un mínimo de un segundo hasta un máximo de 23 horas 59 minutos 59 segundos. Dado que el módulo ha sido desarrollado con un fin didáctico se considera que serán los rangos suficientes para probar su funcionamiento.

Con dicho software se realiza la descarga y el guardado de los datos recolectados por el data logger. Los datos que son descargados serán almacenados en un archivo .xls el mismo que tendrá un nombre definido por el usuario.

El valor de las entradas no solo será visualizado en el software de control sino también en una pantalla LCD que posee físicamente el módulo, con lo cual la conexión a la PC es opcional puesto que el data logger está diseñado para trabajar conectado o no a la misma.

El data logger recibe su voltaje de alimentación del puerto USB asignado en la PC para realizar comunicación con el dispositivo, pero también posee una entrada de alimentación de 9 a 12v que le permite trabajar de manera autónoma, recordando primero configurar el módulo con el software de control en base a las necesidades de medición de la variable que el usuario desee recolectar.

En lo referente al borrado de la memoria la única manera de realizar dicho proceso es mediante el software de control, el que dispone de un botón para dicho fin. Esta característica de diseño se debe a la gran importancia de la información y para

prevenir su pérdida por error, ya que inclusive después de ser descargada la memoria no volátil todos los datos permanecerán, en el dispositivo hasta que el operador decida lo contrario.

Para saber el estado de la memoria el software de control posee un indicador de estado de memoria que es una barra indicadora del porcentaje de uso de la misma.

En lo referente a la parte física del módulo, este dispone de un microcontrolador el que se encarga de la adquisición, registro y comunicaciones. El registro se realiza en una memoria I2C la cual es la encargada de conservar la integridad de los datos hasta que el usuario disponga de los mismos.

Las entradas análogas tienen un diseño interno que permite soportar de 0 a 10v con una precisión de 10 bits, mientras que las entradas digitales soportan de 0 a 5v que se entienden como 0 y 1 lógico respectivamente.

Para plasmar lo antes expuesto el presente documento se divide en cuatro capítulos.

El primero posee los fundamentos teóricos que sustentan el diseño desarrollado en el segundo capítulo. En el tercer capítulo se presentan los resultados y pruebas experimentales realizadas al prototipo, para concluir con el capítulo cuatro que incluye las conclusiones obtenidas a lo largo del desarrollo del proyecto, así como también las recomendaciones que se desprenden de las mismas.

## ÍNDICE

### CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

1.1. Data logger .....	1
1.1.1. Concepto.....	1
1.1.2. Ventajas y desventajas de un data logger.....	2
1.1.3. Campos de aplicación.....	3
1.2. Universal serial bus.....	4
1.2.1. Introducción .....	4
1.2.1.1. Concepto.....	5
1.2.1.2. Origen.....	7
1.2.1.3. Línea de tiempo.....	7
1.2.2. Características de la interface de comunicación USB.....	8
1.2.2.1. UBS 1.0.....	10
1.2.2.2. UBS 1.1.....	11
1.2.2.3. UBS 2.0.....	12
1.2.2.4. UBS 2.0 OTG.....	14
1.2.3. Especificaciones técnicas.....	17
1.2.3.1. La topología.....	17
1.2.3.2. El flujo de datos.....	18
1.2.3.3. La capa de protocolo.....	22
1.2.3.4. Eléctrica.....	25
1.2.3.5. Mecánica.....	28
1.3. Microcontrolador.....	31
1.3.1. Conceptos básicos.....	31
1.3.2. Características generales del pic18F2550.....	33
1.3.3. Programación de los microcontroladores.....	35
1.3.4. PIC's en internet.....	38

### CAPÍTULO II: DISEÑO E IMPLEMENTACIÓN DEL DATA LOGGER

2.1. Descripción de los sistemas.....	39
2.1.1. Sistema electrónico.....	39
2.1.2. Sistema de comunicación.....	40
2.1.2.1. Conversor analógico/digital.....	43
2.2. Programación del microcontrolador para el almacenamiento de datos del data logger y comunicación con la PC.....	44
2.3. Manipulación del modulo USB del pic18F2550.....	50
2.4. Comunicación del modulo data logger a la PC mediante LabVIEW.....	50
2.4.1. Preámbulo de comunicación.....	52
2.5. Acoplamiento del sistema total.....	54
2.5.1. Instalación del microcontrolador.....	56
2.5.2. Obtención de 48 MHz en el PIC 18FXX5X a partir de otros valores de Frecuencia.....	61

## **CAPÍTULO III: PRUEBAS EXPERIMENTALES Y ANÁLISIS DE RESULTADOS**

<b>3.1.</b> Pruebas de adquisición y almacenamiento del data logger.....	65
<b>3.2.</b> Pruebas de comunicación del data logger con la PC.....	75
<b>3.3.</b> Ventajas y limitaciones de la interface USB.....	77
<b>3.4.</b> Análisis de resultados.....	78
<b>3.5.</b> Análisis de costos.....	79

## **CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES**

<b>4.1.</b> Conclusiones.....	83
<b>4.2.</b> Recomendaciones.....	85
<b>4.3.</b> Bibliografía y enlaces.....	86

## **ANEXOS**

- A.** Fragmento del código de programación
- B.** Capturas de la programación en LabVIEW
- C.** Hojas técnicas del microcontrolador PIC18F2550



# CAPÍTULO I

## FUNDAMENTACIÓN TEÓRICA

### 1.1. DATA LOGGER

#### 1.1.1. CONCEPTO

Un registrador de datos (recolector de datos o grabador de datos) es un dispositivo electrónico que registra datos en el tiempo o en relación a la ubicación, ya sea utilizando un instrumento, un sensor o por medio de instrumentos exteriores. Cada vez más, pero no del todo, se basan en un procesador digital (o computadora); por lo general son pequeños, con pilas, portátiles y equipados con un microprocesador, memoria interna para almacenamiento de datos y sensores<sup>1</sup>. Algunos registradores de datos usan interfaz con un computador personal que posee software para activar el registrador de datos tomar y analizar los datos recogidos, mientras que otros tienen un dispositivo de interfaz local (teclado, pantalla LCD) y puede ser utilizado como un dispositivo independiente.

Los registradores de datos varían desde los modelos de uso general, para una amplia gama de aplicaciones, hasta los dispositivos de medición muy específicos según el medio ambiente o el tipo de aplicación. Es común que los tipos de propósito general puedan ser programables, sin embargo muchos siguen como máquinas estáticas con un número limitado o no de parámetros y/o variables. En la actualidad los registradores de datos electrónicos han reemplazado a los registradores de carta en muchas aplicaciones.

Uno de los principales beneficios del uso de registradores de datos es la capacidad para recopilar automáticamente datos las 24 horas del día, tras la activación. Una

---

<sup>1</sup> <http://www.picotech.com/data.html>

vez implementados los registradores de datos normalmente se dejan sin vigilancia para medir y registrar la información de la duración del período de seguimiento<sup>2</sup>.

Esto permite una visión global y precisa de las condiciones ambientales objeto de seguimiento, tales como la temperatura del aire y la humedad relativa.

Debido a la tecnología involucrada en la fabricación de algunos de los registradores de datos de mayor efecto, su implementación puede ser costosa. Los precios pueden variar desde \$ 400 - \$ 12.000 para un registrador de datos; pero, estas unidades de alta calidad suelen durar por muchos años.

### **1.1.2. VENTAJAS Y DESVENTAJAS DE UN DATA LOGGER**

Las ventajas del recolector de datos son:

- No hay lectura errónea de datos.
- El procesamiento de datos de alta velocidad está disponible.
- El ordenamiento de los datos se puede hacer fácilmente.
- El procesamiento de datos se puede efectuar con la Computadora Personal.
- El cambio de la variable medida a una unidad del elemento secundario es registrado.
- La recolección de datos para un periodo de tiempo está disponible. El sistema puede ser instalado en cualquier lugar.
- Lectura visual en gráficos en base a los datos recolectados.

Las ventajas de la lectura visual en gráficos de registradores son las siguientes:

- Bajo costo de los elementos secundarios de medición.
- Bajo costo de mantenimiento.
- Data Logger.

Las desventajas del recolector de datos son:

- Hardware y software para análisis de los datos.

---

<sup>2</sup> R. Ito, Tesis: Racing Data Logger, (UDLA-P Pue. México), (1999).

- Personal capacitado para trabajo de computadoras.
- Requiere asistencia técnica del diseñador o fabricante.

Las desventajas de la lectura visual en gráficos de registradores son las siguientes:

- Se requiere experiencia para interpretar los datos.
- La lectura errada de los datos es inevitable.
- Toma un mayor tiempo recolectar y analizar los datos.
- El manejo de los datos no puede ser simplificado.
- Se requiere que los datos sean procesados a mano.
- El cambio de la variable medida a una unidad del elemento secundario no es registrado.
- En el caso de recolectar los datos continuos por intervalos de tiempo cortos, se incurre en muchos gastos para instalar el sistema en un lugar.

### **1.1.3. CAMPOS DE APLICACIÓN**

Los registradores de datos están compuestos de un procesador programable, una memoria, puertos y de uno o varios canales de entrada para la conexión de diferentes sensores. Los registradores de datos suelen tener un abastecimiento propio de energía (una batería o acumulador), a través del sensor se reciben los datos de medición. Un convertidor analógico-digital convierte los datos en datos electrónicos y los graba en la memoria, ésta puede ser una tarjeta de memoria, una EEPROM, un disco duro o cualquier otro medio. Al memorizar se puede prevenir una posible pérdida de datos (batería descargada). Los datos registrados pueden ser transmitidos a través de los puertos (puerto de serie, USB, LAN, Bluetooth) y analizados por el software, a través de uno de los puertos se configuran el registro de datos para su uso (inicio y fin de la medición, intervalos, etc.).

Dependiendo de las mediciones, los registros de datos pueden tener uno o más canales para poder conectar diferentes sensores simultáneamente.

Los registradores de datos se utilizan en todos aquellos ámbitos en los que se deban realizar tomas de larga duración de un parámetro de medición sin la necesidad de contar con personal de control con una presencia continua en el lugar de la medición. Así los registradores de datos se emplean sobre todo en pruebas de campo, en el control de transportes (transportes en camiones), en pruebas dentro de la industria alimentaria, para el análisis de errores de sistemas (en tensiones oscilatorias de un sistema o presentes en el mismo), en estudios de calidad, en investigación y desarrollo y en formación. Los registradores de datos también resultan interesantes para uso privado.

Las posibilidades y los campos de aplicación son prácticamente ilimitados, tanto como monitores de variables en un proceso productivo industrial, como simples ensayos de laboratorios a nivel educacional.

## **1.2. UNIVERSAL SERIAL BUS**

### **1.2.1. INTRODUCCIÓN**

El *Universal Serial Bus*, abreviado comúnmente USB, es un puerto que sirve para conectar periféricos a una computadora, fue creado en 1996 por siete empresas: IBM, Intel, Northern Telecom, COMPAQ, Microsoft, Digital Equipment Corporation y NEC<sup>3</sup>.

El diseño del USB tenía en mente eliminar la necesidad de adquirir tarjetas separadas para poner en los puertos bus ISA o PCI y mejorar las capacidades plug-and-play permitiendo a esos dispositivos ser conectados o desconectados al sistema sin necesidad de reiniciar; sin embargo, en aplicaciones donde se necesita ancho de banda para grandes transferencias de datos o si se necesita una latencia baja, los buses PCI o PCIe salen ganando. Igualmente sucede si la aplicación requiere de robustez industrial. A favor del bus USB, cabe decir que cuando se conecta un

---

<sup>3</sup> USB - Hardware & Software; John Garney, Ed Solari, Kosar Jaff (Ed. Annabooks)

nuevo dispositivo, el servidor lo enumera y agrega el software necesario para que pueda funcionar.

El USB no puede conectar los periféricos porque sólo puede ser dirigido por el drive central así como: ratones, teclados, escáneres, cámaras digitales, teléfonos móviles, reproductores multimedia, impresoras, discos duros externos, entre otros ejemplos, tarjetas de sonido, sistemas de adquisición de datos y componentes de red. Para dispositivos multimedia como escáneres y cámaras digitales, el USB se ha convertido en el método estándar de conexión, para impresoras, el USB ha crecido tanto en popularidad que ha desplazado a un segundo plano a los puertos paralelos porque el USB hace mucho más sencillo el poder agregar más de una impresora a una computadora personal.

#### **1.2.1.1. Concepto**

El USB o Universal Serial Bus es una interfaz para la transmisión serie de datos y distribución de energía desarrollado por empresas líderes del sector de las telecomunicaciones y de la computación y que ha sido introducida en el mercado de los PC's y periféricos para mejorar las lentas interfaces serie (RS-232) y paralelo. Provee una mayor velocidad de transferencia (de hasta 100 veces más rápido) comparado con el puerto Paralelo de 25-pin y el Serial DB-9, DB-25, RS-232 que son los puertos que se encuentran en la mayoría de los computadores. Tenía en un principio como objetivo el conectar periféricos relativamente lentos (ratones, impresoras, cámaras digitales, unidades ZIP, etc.) de una forma realmente sencilla, rápida y basada en comunicaciones serie, aunque por sus características también podía conectarse con discos duros. En la figura 1.1 se puede constatar lo antes mencionado.



Fig. 1.1: Velocidad de transferencia de varias interfaces.

Esta interfaz de 4 hilos distribuye 5V para la alimentación y puede transmitir datos a una velocidad de hasta 480 Mbps en su versión 2.0. Es un bus serie que hace posible la conexión de hasta 127 periféricos a una única puerta de un PC, con detección y configuración automáticas, siendo esto posible con el PC conectado a la red y sin tener que instalar software adicional, ni reiniciar el ordenador (plug and play), algo que con los puertos convencionales serie y paralelo no sucedía.

Tampoco hay que preocuparse por conflictos de IRQ's o instalar tarjetas de adaptador para cada periférico. Estos periféricos pueden ser: Ratones, teclados, impresoras, escáneres, grabadoras, discos duros, módems, cámaras digitales, etc. Como se puede observar en la figura 1.2:

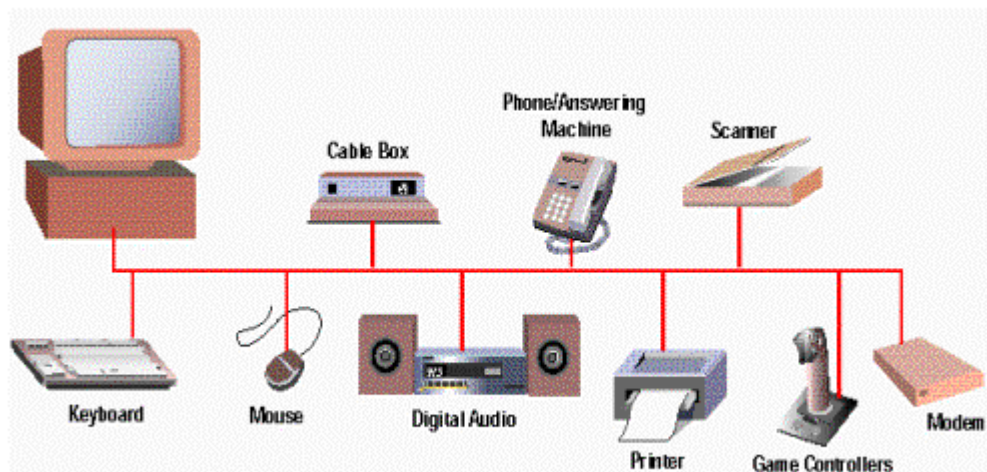


Fig. 1.2: Funciones típicas de un sistema USB.

El éxito de la interfaz USB ha sido tal que, actualmente todos los PC's tienen integrados al menos dos puertos USB para la conexión de dispositivos. Los PC's

modernos además de tener los dos traseros pueden tener más puertos en el frontal para facilitar su conexión. A falta de puertos se pueden acoplar hubs USB para ampliar el número de dispositivos siempre que no se sobrepase el límite soportado de 127.

Como desventaja principal es que el ancho de banda debe repartirse entre todos los dispositivos conectados a él, lo que quizás no tendría repercusión si se intentaran conectar varios ratones, pero sí que la tendría si se quisiera grabar CD's en varias grabadoras en paralelo.

#### **1.2.1.2. Origen**

Uno de los grandes problemas que tienen los PCs (problemas que se han heredado a través de los tiempos debido a un diseño deficiente en algunos aspectos) es la escasez de determinados recursos, básicamente líneas de interrupción IRQ's y canales de acceso directo a memoria DMA's. En ambos casos las capacidades del diseño inicial tuvieron que ser dobladas en 1984, tres años después de su lanzamiento, aprovechando la aparición de la gama AT.

La instalación de periféricos ha sido un gran problema constante para los ensambladores, que debían asignar los escasos recursos disponibles entre los dispositivos del sistema. Aunque el estándar PnP ("Plug and Play") vino a aliviar en parte las dificultades mecánicas de cambiar "jumpers" en las placas, el problema seguía ahí, ya que desde la aparición del AT el diseño del PC no había sufrido cambios sustanciales.

#### **1.2.1.3. Línea de Tiempo**

En la tabla 1.1 se describen cronológicamente los eventos más relevantes en el desarrollo del USB.

Tabla 1.1: Descripción cronológica y línea de tiempo del estándar USB.

Año	Mes	Suceso
1994	Noviembre	Compaq, Intel y el resto de empresas iniciaron las especificaciones del Universal Serial Bus
1994	30 Diciembre	USB 0.8 (Borrador)
1995	13 Abril	USB 0.9 (Borrador)
1996	15 Enero	<b>USB 1.0</b> - Velocidad Oficial 12 Mbps
1996	Febrero	Intel anuncia sus primeros chips controladores de USB
1996	Agosto	Microsoft saca la versión de Windows 95 OSR 2.1 que introduce los controladores correspondientes para el manejo de los dispositivos USB
1998	Marzo	Sale a la venta el iMac de Apple que utilizaba el USB 1.0 para la conexión de teclado y ratón
1998	24 Junio	Sale al mercado el Windows 98, el primer Sistema Operativo con completo soporte para USB
1998	23 Septiembre	<b>USB 1.1</b> - Revisión completada ( <b>USB-IF</b> ->Compaq, Intel, Microsoft, NEC)
1998	Noviembre	<b>USB-IF</b> supera el record mundial de periféricos conectados a un simple PC mediante USB (111 periféricos)
1999		Se forma el grupo promotor del USB 2.0
1999	Febrero	Se propone 240 Mbps como velocidad
1999	12 Octubre	Versión 0.79 del USB 2.0. Se aumenta la velocidad a 480 Mbps
1999	21 diciembre	Versión 0.9 del USB 2.0
2000	29 febrero	In-Systems muestra la primera unidad de almacenamiento USB 2.0, una unidad zip modificada
2000	17 Marzo	Microtech y Netchip muestran su scanner USB 2.0 en el foro de desarrollo de Tokio
2000	27 Abril	<b>USB 2.0</b> - Velocidad Oficial 480 Mbps ( <b>USB-IF</b> ->Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips)
2001	18 Diciembre	<b>USB 2.0 OTG-</b> (On-The-Go) Ampliación del USB 2.0.
2002	Mayo	Intel saca a la venta los primeros chipsets que integran Hi-speed USB 2.0 en el Controlador de Hub de I/O (ICH4). Estos chipsets fueron el Intel 845E, 845G y el 845GL.

### 1.2.2. CARACTERÍSTICAS DE LA INTERFACE DE COMUNICACIÓN USB

La especificación del USB proporciona una serie de características que pueden ser distribuidas en categorías. Estas características son comunes para todas las versiones (desde la 1.0 hasta la 2.0) y son las siguientes<sup>4</sup>:

<sup>4</sup> <http://www.usb.org>



Fácil uso para los usuarios:

- Modelo simple para el cableado y los conectores.
- Detalles eléctricos aislados del usuario (terminaciones del bus).
- Periféricos auto-identificativos.
- Periféricos acoplados y reconfigurados dinámicamente (Hot Swappable).

Flexibilidad:

- Amplio rango de tamaños de paquetes, permitiendo variedad de opciones de buffering de dispositivos.
- Gran variedad de tasas de datos de dispositivos acomodando el tamaño de buffer para los paquetes y las latencias.
- Control de flujo para el manejo del buffer construido en el protocolo.

Ancho de banda sincrónico:

- Se garantiza un ancho de banda y bajas latencias apropiadas para telefonía, audio, etc.
- Cantidad de trabajo sincrónico que puede usar el ancho de banda completo del bus.
- Control de flujo para el manejo del buffer construido en el protocolo.
- Amplia gama de aplicaciones y cargas de trabajo.
- Adecuando el ancho de banda desde unos pocos Kbps hasta varios Mbps.
- Soporta tanto el tipo de transferencia isócrono como el asíncrono sobre el mismo conjunto de cables.
- Conexiones múltiples, soportando operaciones concurrentes de varios dispositivos.
- Soporta hasta 127 dispositivos físicos.
- Soporta la transferencia de múltiples datos y flujos de mensajes entre el host y los dispositivos.

Robustez:

- Manejo de errores y mecanismos de recuperación ante fallos implementados en el protocolo.
- Inserción dinámica de dispositivos.
- Soporte para la identificación de dispositivos defectuosos.

Implementación de bajo costo

- Sub canal de bajo costo a 1.5 Mbps.
- Conectores y cables de bajo costo.
- Adecuado para el desarrollo de periféricos de bajo costo.

#### **1.2.2.1. USB 1.0**

Esta versión, publicada en enero 1996, fue iniciada por el USB-IF cuyos integrantes eran COMPAQ, Digital Equipment Corp., IBM, Intel Corp., Microsoft Corp., NEC y Northern Telecom. Esta es la primera versión oficial que salió a la luz y que reúne todas las características arriba mencionadas, con un límite máximo de velocidad de 12 Mbps. Inicialmente la especificación USB fue diseñada para conectar eficientemente teléfonos a PCs; sin embargo, este nuevo estándar de conectividad tuvo tanto éxito que el Foro de implementadores decidió empujar al USB como un estándar de PC, a pesar que el desarrollo de dispositivos USB para telefonía aún no se ha realizado.

En la tabla 1.2 se observa el alcance de aplicación del USB con respecto a la velocidad máxima establecida por las características del bus. Esto es también aplicable a las versiones 1.x del bus.

Tabla1.2: Alcance del uso del USB respecto a la velocidad máxima establecida

<u>RENDIMIENTO</u>	<u>APLICACIONES</u>	<u>ATRIBUTOS</u>
<b>VELOCIDAD BAJA</b>		
Dispositivos Comunes Dispositivos Interactivos 10-100 Kb/s	Teclado y ratón  Periféricos de entretenimiento Configuración del Monitor	Bajo costo  Des/Conexión en caliente Fácil uso Múltiples periféricos
<b>VELOCIDAD MEDIA</b>		
Teléfono Audio Video comprimido 500 Kb/s - 10 Mb/s	ISDN/RDSI  PBX POTS Audio	Bajo costo  Des/Conexión en caliente Fácil uso Múltiples dispositivos Garantizada la latencia Garantizado el ancho de banda

#### 1.2.2.2. USB 1.1

El objetivo de esta segunda versión, que salió a la luz en septiembre de 1998, era solucionar problemas de ambigüedad en la especificación 1.0 para facilitar el trabajo a los desarrolladores tanto de software como de hardware sin que hubiera que hacer cambios en los dispositivos para hacerlos funcionar bajo esta versión. No se hicieron cambios en los controladores de host para mantener esta compatibilidad.

El alcance de aplicación de esta versión coincide con la de su predecesora, así como sus características generales.

Dado que al USB se le dió una aplicación distinta a la de las telecomunicaciones, IBM y Northern Telecom no continuaron como miembros, para esta especificación sólo quedaron COMPAQ, Intel, Microsoft y NEC ya que Digital Equipment Corp. es ahora parte de COMPAQ.

### **1.2.2.3. USB 2.0**

Esta versión que salió al mercado a mitad del 2000, tras la unión al consorcio de otras 3 compañías (Hewlett Packard, Philips y Lucent), con lo que volvían a ser siete las integrantes del USB-IF.

Como principal característica está el aumento de velocidad hasta 480 Mbps (casi 40 veces la velocidad anterior) con una diferencia de costo casi mínimo. De este aumento de velocidad le viene el nombre de Hi-Speed, que es fácil de confundir con Full-Speed. Full-Speed era el término que se le daba a las versiones USB 1.x que alcanzaban 12 Mbps, por lo que se debe tener en cuenta a la hora de hablar sobre ello.

Este incremento de velocidad ha permitido sustituir en muchos casos el interfaz SCSI por el USB.

En total USB 2.0 soporta tres tipos distintos de velocidad:

- Hi-Speed 480 Mbps.
- Full-Speed 12 Mbps.
- Low-Speed 1.5 Mbps.

El alcance de uso para la versión USB 2.0 se ve incrementada notablemente respecto a sus predecesoras. Quedaría la tabla 1.3 que es una modificación de la tabla 1.2.

Tabla 1.3: Alcance de la versión USB 2.0 incrementada respecto a sus predecesoras

<b><u>RENDIMIENTO</u></b>	<b><u>APLICACIONES</u></b>	<b><u>ATRIBUTOS</u></b>
<b>VELOCIDAD BAJA</b>		
Dispositivos Comunes Dispositivos Interactivos 10-100 Kb/s	Teclado y ratón  Periféricos de entretenimiento Configuración del Monitor	Bajo costo  Des/Conexión en caliente Fácil uso Múltiples periféricos
<b>VELOCIDAD MEDIA</b>		
Teléfono Audio Video comprimido 500 Kb/s - 10 Mb/s	ISDN/RDSI  PBX POTS Audio	Bajo costo Des/Conexión en caliente Fácil uso Múltiples dispositivos Garantizada la latencia Garantizado el ancho de banda
<b>VELOCIDAD ALTA</b>		
Video Unidades de almacenamiento Video Comprimido 25-500 Mb/s	Video Discos duros y grabadoras	Gran ancho de banda Latencia garantizada Fácil uso

Básicamente, el USB 2.0 incluye todo lo que ofrece USB 1.1 y añade el modo de alta velocidad, el USB 2.0 también usa el mismo tipo de cables y conectores para conectar los dispositivos de alta velocidad, sin embargo los hubs USB clásicos ralentizarán los dispositivos USB 2.0. Otro requisito necesario es un controlador de

host para USB 2.0 si se quiere tener disponibles la conexión de alta velocidad con un dispositivo de este tipo.

Los hubs USB 2.0 tienen ahora mucho más trabajo que hacer que en el USB 1.1 ya que necesitan manejar todo el tráfico de tres tipos de dispositivos con velocidades distintas. Conectando un dispositivo USB 1.1 en uno USB 2.0 funcionaría bien, pero no lo haría si se hiciera al revés, además de ralentizarse a 12 Mbps, posiblemente, el sistema operativo avisaría de su mal uso.

Cabe destacar que el USB 2.0 es simplemente una extensión y nunca llegará a reemplazar completamente al 1.1 ya que hay productos como teclados genéricos, ratones, joysticks o altavoces que no requieren la gran velocidad que ofrece la tecnología USB 2.0. Sólo dispositivos de alta velocidad como webcams o sistemas de alta capacidad necesitarán el máximo de velocidad, aunque los ordenadores los fabriquen sólo con puertos USB 2.0.

#### **1.2.2.4. USB 2.0 OTG**

USB OTG o "USB-On-The-Go" es una variación de la especificación USB 2.0 que permite a un sólo puerto actuar como servidor o como dispositivo, más concretamente permitiría comunicarse a todo tipo de unidades con USB directamente entre ellas sin la necesidad de ordenador. Incluso después de que el cable está conectado y las unidades se están comunicando, las dos unidades pueden "cambiar" de rol bajo el control de un programa.

En resumen, USB On-The-Go define las siguientes características añadidas a la especificación USB:

- Un nuevo estándar para conectores USB y cables (Mini conectores).
- El hecho de que dispositivos que antes eran solamente periféricos ahora puedan ser hosts (dual-role devices)<sup>5</sup>.

---

<sup>5</sup> USB - Hardware & Software por John Garney, Ed Solari, Kosar Jaff

- La capacidad de ser host y periféricos y cambiar dinámicamente de rol según convenga (el uso del Host Negotiation Protocol o HNP para el cambio de rol).
- Protocolo de petición sesión (Session Request Protocol o SRP).
- Requisitos de bajo consumo para promocionar el USB en dispositivos alimentados con baterías (opción Micropower añadido a las ya existentes low y high power).

Esta tecnología es muy útil para dispositivos como el PDA, donde el enlace USB puede conectarse a un PC como un dispositivo o conectarse como servidor a un teclado o ratón.

Se puede imprimir fotos directamente desde la cámara digital o descargar canciones a través del móvil de tecnología WAP. En la tabla 1.4 se incluyen algunas aplicaciones posibles de este tipo de tecnología.

Tabla 1.4: Aplicaciones posibles de la tecnología USB-On-The-Go

<b>HOST</b>	<b>PERIFÉRICO</b>	<b>APLICACIÓN</b>
Teléfono móvil	Teléfono móvil	Intercambiar información de contactos
	Cámara digital	Mandar por mail fotos
	Reproductor MP3	Subir fotos a una web
	Dispositivo de almacenamiento	Subir/bajar música
	Scanner	Subir/bajar ficheros
		Escanear tarjetas de visita
Cámara Digital	Cámara digital	Intercambiar fotos
	Teléfono móvil	Mandar por mail fotos
	Impresora	Subir fotos a una web
	Dispositivo de almacenamiento	Imprimir fotos
		Almacenar fotos

Impresora	Cámara digital	Imprimir fotos
	Scanner	Imprimir imágenes escaneadas
	Dispositivo de almacenamiento	Imprimir ficheros almacenados
Reproductor MP3	Reproductor MP3	Intercambiar canciones
	Dispositivo de almacenamiento	Subir/bajar canciones
Osciloscopio	Impresora	Imprimir imágenes
PDA	PDA	Intercambiar ficheros
	Impresora	Imprimir ficheros
	Teléfono móvil	Subir/bajar ficheros
	Reproductor MP3	Subir/bajar música
	Escáner	Escanear fotos
	Dispositivo de almacenamiento	Obtener direcciones
	GPS	Información de mapas
	Cámara digital	Subir fotos
	Osciloscopio	Configurar un osciloscopio

Dependiendo de la velocidad de transmisión (low-speed o Hi-Speed) existen dos logotipos que identifica la existencia de un puerto USB On-The-Go en el dispositivo y que es fácilmente identificable al igual que con el resto de especificaciones.



## 1.2.3. ESPECIFICACIONES TÉCNICAS

### 1.2.3.1. La topología

El Universal Serial Bus conecta los dispositivos USB con el host USB. La interconexión física USB es una topología de estrellas apiladas donde un hub es el centro de cada estrella. Cada segmento de cable es una conexión punto-a-punto entre el host y los hubs o función, o un hub conectado a otro hub o función.

El número máximo de dispositivos que puede conectar USB es de 127, pero debido a las constantes de tiempo permitidas para los tiempos de propagación del hub y el cable, el número máximo de capas permitido es de siete (incluida la capa raíz), con un máximo de longitud de cable entre el hub y el dispositivo de 5 metros. Cabe destacar que en siete capas, sólo se soportan cinco hubs que no sean raíz en una ruta de comunicación entre el host y cualquier dispositivo. Un dispositivo compuesto ocupa dos capas, por eso, no puede ser activado si está acoplado en la última capa de nivel siete. Sólo las funciones pueden ponerse en este nivel. En la figura 1.3 se indica las diferentes capas antes mencionadas.

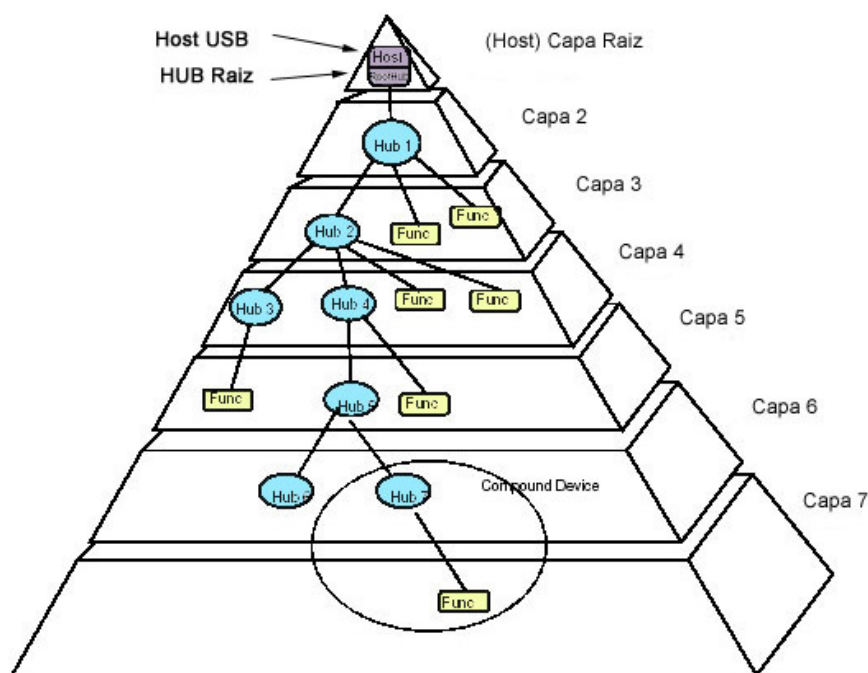


Fig. 1.3: Estructura de capas del bus USB.

La topología del bus USB se puede dividir en tres partes:

- La capa física: Aquí se indica cómo están conectados los elementos físicamente.
- La capa lógica: Los roles y las responsabilidades de los elementos USB.
- La relación software del cliente-función: Cómo se ven mutuamente el software del cliente y los interfaces de las funciones relacionadas. En la figura 1.4 se muestra la relación existente.

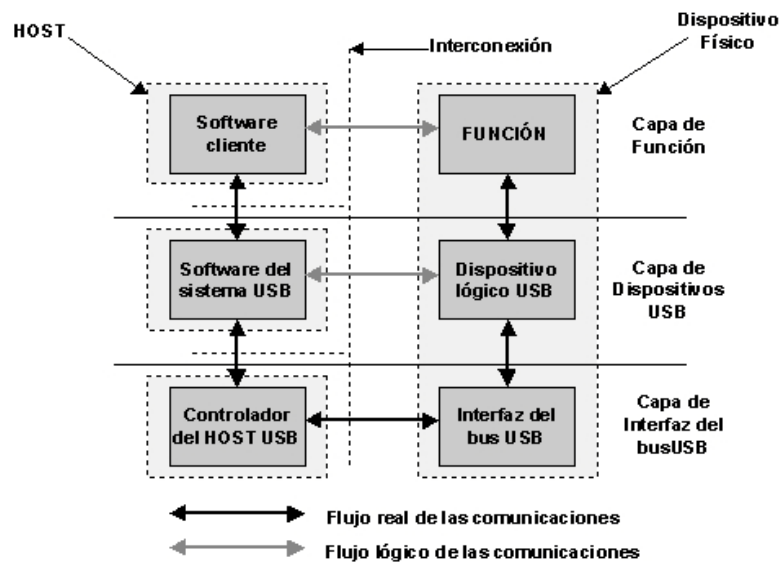


Fig. 1.4: Relación software del cliente-función

### 1.2.3.2. El flujo de datos

Un dispositivo USB desde un punto de vista lógico hay que entenderlo como una serie de endpoints, a su vez los endpoints se agrupan en conjuntos que dan lugar a interfaces, las cuales permiten controlar la función del dispositivo.

Como ya se ha visto la comunicación entre el host y un dispositivo físico USB se puede dividir en tres niveles o capas. En el nivel más bajo el controlador de host USB se comunica con la interfaz del bus utilizando el cable USB, mientras que en un nivel superior el software USB del sistema se comunica con el dispositivo lógico utilizando la tubería de control por defecto ("*Default Control Pipe*"). En lo que al nivel de función se refiere, el software cliente establece la comunicación con las interfaces de la función a través de tuberías asociadas a endpoints.

## Endpoints y direcciones de dispositivo

Cada dispositivo USB está compuesto por una colección de endpoints independientes y una dirección única asignada por el sistema en tiempo de conexión de forma dinámica. A su vez cada endpoint dispone de un identificador único dentro del dispositivo al que pertenece, a este identificador se le conoce como número de endpoint y viene asignado de fábrica. Cada endpoint tiene una determinada orientación de flujo de datos. La combinación de dirección, número de endpoint y orientación, permite referenciar cada endpoint de forma inequívoca.

Cada endpoint es por sí solo una conexión simple que soporta un flujo de datos en una única dirección, bien de entrada o bien de salida. Este endpoint se caracteriza por:

- Frecuencia de acceso al bus requerida.
- Ancho de banda requerido.
- Número de endpoint.
- Tratamiento de errores requerido.
- Máximo tamaño de paquete que el endpoint puede enviar o recibir.
- Tipo de transferencia para el endpoint.
- La orientación en la que se transmiten los datos.

Existen dos endpoints especiales que todos los dispositivos deben tener, los endpoints con número 0 de entrada y de salida, que deben de implementar un método de control por defecto al que se le asocia la tubería de control por defecto.

Estos endpoints están siempre accesibles mientras que el resto no lo estarán hasta que no hayan sido configurados por el host.

## Tuberías

Una tubería USB es una asociación entre uno o dos endpoints en un dispositivo y el software en el host. Las tuberías permiten mover datos entre software en el host, a través de un buffer y un endpoint en un dispositivo. Hay dos tipos de tuberías:

- Stream: Los datos se mueven a través de la tubería sin una estructura definida.
- Mensaje: Los datos se mueven a través de la tubería utilizando una estructura USB.

Además, una tubería se caracteriza por:

- Demanda de acceso al bus y uso del ancho de banda.
- Un tipo de transferencia.
- Las características asociadas a los endpoints.

Como ya se acotó, la tubería que está formada por dos endpoints con número cero se denomina tubería de control por defecto. Esta tubería está siempre disponible una vez se ha conectado el dispositivo y ha recibido un reseteo del bus.

El resto de tuberías aparecen después que se configure el dispositivo, la tubería de control por defecto es utilizada por el software USB del sistema para obtener la identificación y requisitos de configuración del dispositivo, y para configurar al dispositivo.

El software cliente normalmente realiza peticiones para transmitir datos a una tubería vía IRP's y entonces, o bien espera, o bien es notificado de que se ha completado la petición. El software cliente puede causar que una tubería devuelva todas las IRP's pendientes. El cliente es notificado de que una IRP se ha completado cuando todas las transacciones del bus que tiene asociadas se han completado correctamente, o bien porque se han producido errores. Una IRP puede necesitar de varias tandas para mover los datos del cliente al bus.

La cantidad de datos en cada tanda será el tamaño máximo de un paquete excepto el último paquete de datos que contendrá los datos que faltan. De modo que un paquete recibido por el cliente que no consiga llenar el buffer de datos de la IRP puede interpretarse de diferentes modos en función de las expectativas del cliente, si esperaba recibir una cantidad variable de datos considerará que se trata del último paquete de datos, sirviendo pues como delimitador de final de datos, mientras que si esperaba una cantidad específica de datos lo tomará como un error.

### Streams

No necesita que los datos se transmitan con una cierta estructura. Las tuberías stream son siempre unidireccionales y los datos se transmiten de forma secuencial: "*first in, first out*". Están pensadas para interactuar con un único cliente, por lo que no se mantiene ninguna política de sincronización entre múltiples clientes en caso de que así sea. Un stream siempre se asocia a un único endpoint en una determinada orientación.

### Mensajes

A diferencia de lo que ocurre con los streams, en los mensajes la interacción de la tubería con el endpoint consta de tres fases. Primero se realiza una petición desde el host al dispositivo, después se transmiten los datos en la dirección apropiada, finalmente un tiempo después se pasa a la fase estado. Para poder llevar a acabo este paradigma es necesario que los datos se transmitan siguiendo una determinada estructura.

Las tuberías de mensajes permiten la comunicación en ambos sentidos, de hecho la tubería de control por defecto es una tubería de mensajes. El software USB del sistema se encarga de que múltiples peticiones no se envíen a la tubería de mensajes concurrentemente. Un dispositivo ha de dar únicamente servicio a una petición de mensaje en cada instante por cada tubería de mensajes.

Una tubería de mensajes se asocia a un par de endpoints de orientaciones opuestas con el mismo número de endpoint.

### Frames y microframes

USB establece una unidad de tiempo base equivalente a 1 milisegundo denominada frame y aplicable a buses de velocidad media o baja, en alta velocidad se trabaja con microframes, que equivalen a 125 microsegundos. Los (micro) frames no son más que un mecanismo del bus USB para controlar el acceso a este, en función del tipo de transferencia que se realice. En un (micro) frame se pueden realizar diversas transacciones de datos.

### Tipos de transferencias

La interpretación de los datos que se transmitan a través de las tuberías, independientemente de que se haga siguiendo o no una estructura USB definida, corre a cargo del dispositivo y del software cliente. No obstante, USB proporciona cuatro tipos de transferencia de datos sobre las tuberías para optimizar la utilización del bus en función del tipo de servicio que ofrece la función.

Estos cuatro tipos son:

- Transferencias de control.
- Transferencias isócronas.
- Transferencias de interrupción.
- Transferencias de bultos ("Bulk").

### **1.2.3.3. La capa de protocolo**

La forma en la que las secuencias de bits se transmiten en USB es la siguiente; primero se transmite el bit menos significativo, después el siguiente menos significativo y así hasta llegar al bit más significativo. Cuando se transmite una

secuencia de bytes se realiza en formato "little-endian", es decir del byte menos significativo al byte más significativo<sup>6</sup>.

En la transmisión se envían y reciben paquetes de datos, cada paquete de datos viene precedido por un campo Sync y acaba con el delimitador EOP, todo esto se envía codificado además de los bits de relleno insertados. En este punto cuando se habla de datos se refiere a los paquetes sin el campo Sync ni el delimitador EOP, y sin codificación ni bits de relleno.

El primer campo de todo paquete de datos es el campo PID. El PID indica el tipo de paquete y por lo tanto, el formato del paquete y el tipo de detección de errores aplicado al paquete. En función de su PID podemos agrupar los diferentes tipos de paquetes en cuatro clases, las que podemos apreciar en la tabla 1.5:

Tabla 1.5: Diferentes tipos de paquetes agrupados en función de su PID

<b>TIPO PID</b>	<b>NOMBRE PID</b>	<b>PID</b>	<b>DESCRIPCIÓN</b>
Token	OUT	0001B	Dirección + número de endpoint en una transacción host a función.
	IN	1001B	Dirección + número de endpoint en una transacción función a host.
	SOF	0101B	Indicador de inicio de frame (Start Of Frame) y número de frame.
	SETUP	1101B	Dirección + número de endpoint en una transacción host a función para realizar un Setup de una tubería de control.

<sup>6</sup> <http://www.semiconductors.philips.com/buses/usb/>

Data	DATA0	0011B	PID de paquete de datos par.
	DATA1	1011B	PID de paquete de datos impar.
	DATA2	0111B	PID de paquete de datos de alta velocidad, elevado ancho de banda en una transferencia isócrona en un microframe.
Handshake	MDATA	1111B	PID de paquete de datos de alta velocidad para split y elevado ancho de banda en una transferencia isócrona.
	ACK	0010B	El receptor acepta el paquete de datos libre de errores.
	NAK	1010B	El dispositivo receptor no puede aceptar los datos o el dispositivo emisor no puede enviar los datos.
	STALL	1110B	Endpoint sin servicio o una petición de control sobre una tubería no está soportado.
Special			
	NYET	0110B	Aún no se ha recibido una respuesta del receptor.
	PRE	1100B	(Token) Habilita tráfico de bajada por el bus a dispositivos de velocidad baja.
	ERR	1100B	(Handshake) Error de transferencia Split.



	SPLIT	1000B	(Token) Transferencia de alta velocidad Split.
	PING	0100B	(Token) Control de flujo sobre endpoints de tipo control o bulk.
	Reservado	0000B	PID reservado.

#### 1.2.3.4. Eléctrica

##### Identificación de la velocidad del dispositivo

Para poder iniciar cualquier tipo de transacción cuando se conecta el dispositivo al host, es necesario que este conozca la velocidad a la que trabaja. Con esa finalidad existe un mecanismo a nivel eléctrico, la diferencia entre los dispositivos de velocidad media y los de velocidad baja, es que en velocidad media tiene una resistencia conectada al D+, en velocidad baja la misma resistencia se encuentra en D- y no en D+ como se puede observar en la figura 1.5:

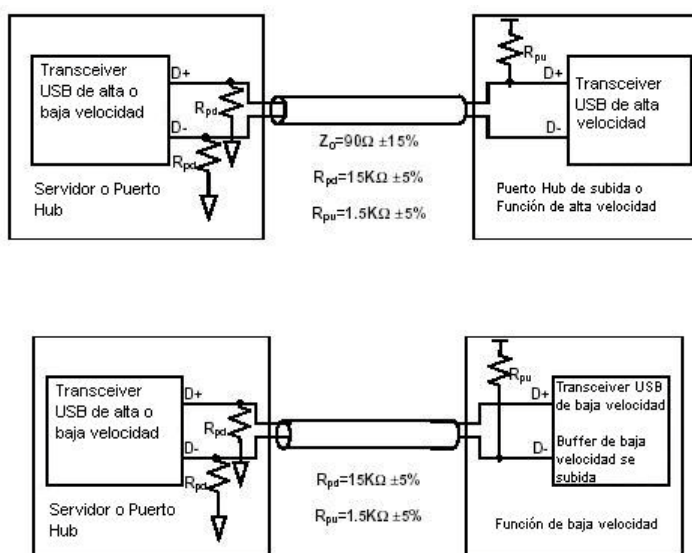


Fig. 1.5: Diferencia entre los dispositivos de velocidad media y los de velocidad baja

De forma que después del reset el estado de reposo de la línea es diferente si se trata de baja o media velocidad. En el caso de dispositivos de alta velocidad lo que se hace es que en un principio se conecta como un dispositivos de velocidad media y más tarde a través de un protocolo se pasa a velocidad alta.

### Codificación de datos

El USB utiliza la codificación NRZI para la transmisión de paquetes. En esta codificación los "0" se representan con un cambio en el nivel, y por el contrario los "1" se representan con un no cambio en el nivel. De modo que las cadenas de cero producen transiciones consecutivas en la señal, mientras que cadenas de unos produce largos periodos sin cambios en la señal. A continuación un ejemplo representado en la figura 1.6:

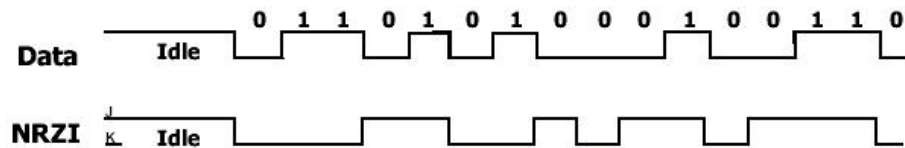


Fig. 1.6: Codificación de datos en USB

### Relleno de bits

Debido a que cadenas de unos pueden producir largos periodos en los que la señal no cambia dando lugar a problemas de sincronización, se introducen los bits de relleno. Cada 6 bits consecutivos a "1" se inserta un bit a "0" para forzar un cambio, de esta forma el receptor puede volverse a sincronizar. El relleno bits empieza con el patrón de señal Sync. El "1" que finaliza el patrón de señal Sync es el primer uno en la posible primera secuencia de seis unos.

En las señales a velocidad media o baja, el relleno de bits se utiliza a lo largo de todo el paquete sin excepción. De modo que un paquete con siete unos consecutivos será considerado un error y por lo tanto ignorado.

En el caso de la velocidad alta se aplica el relleno de bits a lo largo del paquete, con la excepción de los bits intencionados de error usados en EOP a velocidad alta.

### Sync

Teniendo en cuenta que K y J representan respectivamente nivel bajo y nivel alto, el patrón de señal Sync emitido, con los datos codificados, es de 3 pares KJ seguidos de 2 K para el caso de velocidad media y baja. Para velocidad alta es una secuencia de 15 pares KJ seguidos de 2 K. A continuación en la figura 1.7 el caso de velocidad media y baja:

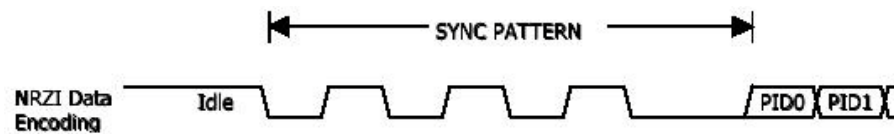


Fig. 1.7: Patrón de señal Sync emitido

El patrón de señal Sync siempre precede al envío de cualquier paquete, teniendo como objetivo que el emisor y el receptor se sincronicen y se preparen para emitir y recibir datos respectivamente.

Si se inicia en el estado de reposo de la señal es J, puede interpretarse Sync como una secuencia impar de "0's" y un "1" que se inserta antes de los datos.

### EOP ("End Of Packet")

A todo paquete le sigue EOP, cuya finalidad es indicar el final del paquete. En el caso de velocidad media y baja el EOP consiste en que, después del último bit de datos en el cual la señal estará o bien en estado J, o bien en estado K, se pasa al estado SE0 durante el periodo que se corresponde con el ocupado por dos bits, finalmente se transita al estado J que se mantiene durante 1 bit. Esta última transición indica el final del paquete.

En el caso de la velocidad alta se utilizan bits de relleno erróneos, que no están en el lugar correcto, para indicar el EOP. Concretamente, el EOP sin aplicar codificación consistiría en añadir al final de los datos la secuencia 0111 1111.

### 1.2.3.5. Mecánica

Como ya se revisó la topología física USB consiste en la conexión del puerto de bajada de un hub o host, con el puerto de subida de algún otro dispositivo o hub. Para facilitar la conexión de dispositivos de cara al usuario, USB utiliza dos tipos de conectores totalmente diferentes, los conectores de Serie A y los conectores de serie B. Los conectores de serie A permiten la conexión directa de dispositivos USB con el host o con el puerto de bajada de un host, y es obligatorio que estén presentes en todos los dispositivos y hubs USB. Los conectores de Serie B no son obligatorios y sirven para conectar un cable USB con el puerto de subida de un dispositivo, permitiendo por parte de los fabricantes de dispositivos la utilización de cables estándar USB. En la figura 1.8 se puede apreciar lo antes mencionado.

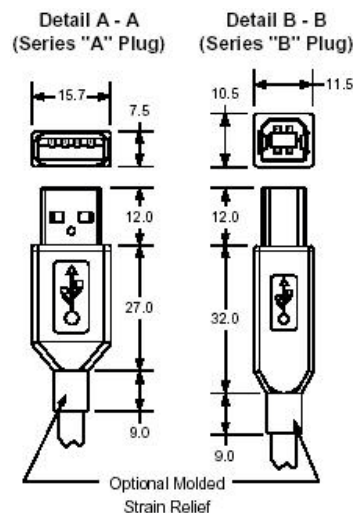


Fig. 1.8: Conectores tipo serie A y serie B

El cable USB consiste de cuatro conductores, dos conductores de potencia y dos de señal, D+ y D-. Los cables de media y alta velocidad están compuestos por un par trenzado de señal, además de GND (Tierra) y Vbus.

Existen tres tipos de cables USB: cable estándar de quita y pon, cable fijo de media y alta velocidad, y cable fijo de baja velocidad.

#### Cable estándar Plug and Play (quita y pon)

Se trata de un cable de velocidad alta y media, con un conector macho de Serie A en un extremo y un conector macho de Serie B en el otro extremo. Esto permite a los fabricantes de dispositivos fabricarlos sin cable y al usuario le facilita la sustitución del cable en caso de que se estropee. Es un requisito que los dispositivos que utilicen este cable sean de velocidad alta o media, el correcto funcionamiento del cable con dispositivos de velocidad baja no está garantizado porque podría superar la longitud máxima del cable de velocidad baja. En la figura 1.9 se muestra un cable USB estándar plug and play.

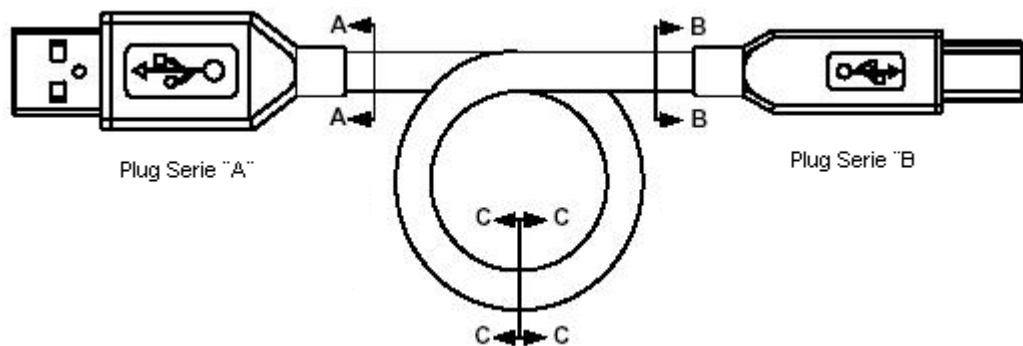


Fig. 1.9: Cable estándar USB

#### Cable fijo de velocidad alta y media

La denominación de fijo se refiere a los cables que son proporcionados por el fabricante del dispositivo fijos a éste, o bien sin ser fijos, con un conector específico del fabricante. Es obligatorio que en un extremo tenga un conector macho de Serie A. Dado que lo suministra el fabricante, puede ser utilizado por dispositivos tanto de velocidad alta y media, como de velocidad baja. En el caso de que se utilice para un dispositivo de velocidad baja, además de poder ser utilizado con dispositivos de velocidad media y alta, deberá cumplir con todos los requisitos propios de la velocidad baja.

## Cable fijo de velocidad baja

Al igual que el cable fijo de alta y media velocidad tiene un conector macho de Serie A en un extremo, mientras que el otro depende del fabricante. La diferencia es que este tipo de cables sólo funciona con dispositivos de velocidad baja, la tabla 1.6 hace referencia a la señal que provee cada uno de sus respectivos contactos.

Tabla 1.6: Señal que provee cada uno de los contactos y su color identificativo.

Contacto	Señal	Cable
1	VBUS	Rojo
2	Datos-	Blanco
3	Datos+	Verde
4	GND	Negro

## El ícono USB

El ícono USB es obligatorio que se encuentre en los conectores de serie A y B, esto facilita al usuario tanto el reconocimiento de que se trata de un cable USB, como la orientación a la hora de conectarlo. En la figura 1.10 se puede observar el diseño y la colocación del logotipo.

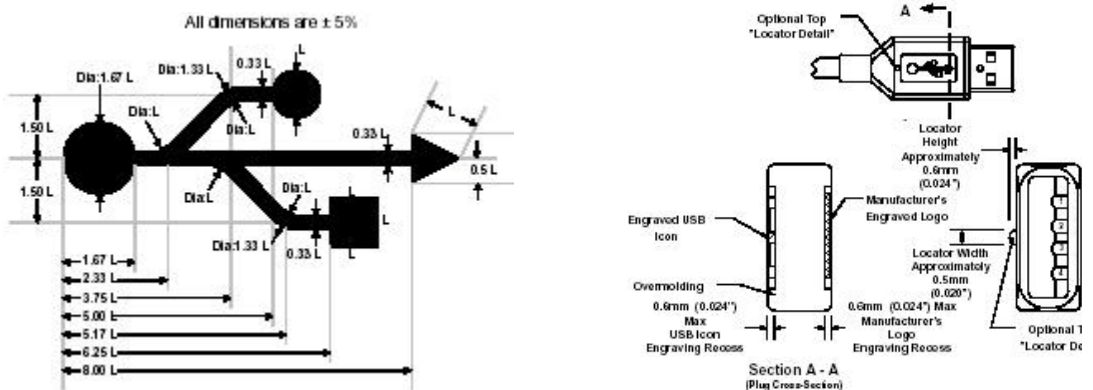


Fig. 1.10: Estándar y dimensiones de ícono USB

## **1.3. MICROCONTROLADOR**

### **1.3.1. CONCEPTOS BÁSICOS**

Un microcontrolador es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: unidad central de procesamiento, memoria y unidades de E/S (entrada/salida).

El “secreto” de los microcontroladores está en su tamaño, su precio y su diversidad.

Son diseñados para disminuir el costo económico y el consumo de energía de un sistema en particular, por eso el tamaño de la unidad central de procesamiento, la cantidad de memoria y los periféricos incluidos dependerán de la aplicación. El control de un electrodoméstico sencillo como una batidora, utilizará un procesador muy pequeño (4 u 8 bit) por que sustituirá a un autómata finito. En cambio un reproductor de música y/o vídeo digital (mp3 o mp4) requerirá de un procesador de 32 bit o de 64 bit y de uno o más Códec de señal digital (audio y/o vídeo). El control de un sistema de frenos ABS (Antilock Brake System) se basa normalmente en un microcontrolador de 16 bit, al igual que el sistema de control electrónico del motor en un automóvil.

Tanto la industria de los computadores comerciales como la de los microcontroladores están decantándose hacia la filosofía RISC (Computadores de Juego de Instrucciones Reducido). 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 hardware y el software del procesador.

Los PIC son una familia de microcontroladores tipo RISC, fabricados por Microchip Technology Inc. y derivados del PIC1650, originalmente desarrollado por la división de microelectrónica de General Instrument.

El nombre actual no es un acrónimo. En realidad, el nombre completo es PICmicro, aunque generalmente se utiliza como Peripheral Interface Controller (controlador de interfaz periférico).

El PIC original se diseñó para ser usado con la nueva CPU de 16 bits CP16000, siendo en general una buena CPU, ésta tenía malas prestaciones de entradas, y el PIC de 8 bits se desarrolló en 1975 para mejorar el rendimiento del sistema quitando peso de E/S a la CPU. El PIC utilizaba microcódigo simple almacenado en ROM para realizar estas tareas y aunque el término no se usaba por aquel entonces, se trata de un diseño RISC que ejecuta una instrucción cada 4 ciclos del oscilador.

En 1985 la división de microelectrónica de General Instrument se separa como compañía independiente que es incorporada como filial (el 14 de diciembre de 1987 cambia el nombre a Microchip Technology y en 1989 es adquirida por un grupo de inversores) y el nuevo propietario canceló casi todos los desarrollos, que para esas fechas la mayoría estaban obsoletos. El PIC, sin embargo, se mejoró con EPROM para conseguir un controlador de canal programable. Hoy en día multitud de PIC's vienen con varios periféricos incluidos (módulos de comunicación serie, UARTs, núcleos de control de motores, etc.) y con memoria de programa desde 512 a 32.000 palabras (una palabra corresponde a una instrucción en ensamblador, y puede ser 12, 14 o 16 bits, dependiendo de la familia específica de PICmicro).



### 1.3.2. CARACTERÍSTICAS GENERALES DEL PIC18F2550<sup>7</sup>

#### Características del USB:

- USB V2.0.
- Velocidad baja (1,5 Mb/s) y velocidad alta (12 Mb/s).
- Soporta Transferencias de control, interrupción, sincrónica y masiva.
- Soporta hasta 32 endpoints (16 bidireccional).
- 1-Kbyte de RAM de doble acceso para USB.
- On-Chip-receptor USB con On-Chip regulador de voltaje.
- Interfaz para off-chip receptor USB.

#### Modos de trabajo:

- Ejecución: CPU encendido, periféricos encendidos.
- Inactivo: COU apagado, periféricos encendidos.
- Sleep: CPU apagado, periféricos apagados.
- Las corrientes en modo inactivo por lo general bajan hasta 5,8  $\mu$ A y en modo de suspensión hasta 0,1 mA.
- Oscilador de Timer1: 1,1  $\mu$ A generalmente, 32 kHz, 2V.
- Watchdog Timer: 2,1  $\mu$ A.

#### Estructura de Oscilador Flexible:

- Cuatro modos de Cristal, incluyendo Alta Precisión PLL para USB.
- Dos modos de reloj externo, de hasta 48 MHz.
- Bloque de Oscilador interno.
- 8 frecuencias seleccionables por el usuario, a partir de 31 kHz a 8 MHz.
- Frecuencia ajustable por el usuario para compensar la desviación.
- Oscilador Secundario usando el Timer1 a 32 kHz.
- La opción Dual Oscillator permite al microcontrolador y al módulo USB funcionar a diferentes velocidades de reloj.

---

<sup>7</sup> Datasheet PIC 18F2550 de Microchip

- Fail-Safe Clock Monitor: Permite una parada segura si un reloj se detiene.

#### Periféricos:

- Corriente máxima Sink/Source: 25 mA/25 mA.
- Tres Interrupciones Exteriores.
- Cuatro módulos de temporizador (timer0 a Timer3).
- Hasta 2 módulos Captura/Comparación/PWM (CCP):
  - Captura 16-bit, a una resolución máxima de 5.2 ns. (TCY/16).
  - Compara 16-bit, a una resolución máxima de 83.3 ns. (TCY).
  - Salida PWM: resolución PWM es de 1 a 10-bit.
- Mejora en los módulos de Captura/Comparación/PWM (PECC):
  - Modo de salida múltiple.
  - Polaridad seleccionable.
  - Tiempo muerto programable.
  - Auto-apagado y reinicio automático.
- Mejora en el módulo USART: Soporte para bus LIN.
- Master Synchronous Serial Port (MSSP) soporta el Módulo SPI (los 4 modos) e I2C<sup>8</sup> en modo de maestro y esclavo.
- 10-bit, hasta 13 canales del módulo convertidor de analógico a digital (A/D), con tiempo programable adquisición.
- Comparadores Duales Análogos con la entrada de multiplexación.

#### Características especiales:

- Compilador C Arquitectura optimizada con opción de extensión del conjunto de instrucciones.
- 100.000 ciclos de borrado/escritura de memoria Flash.
- 1.000.000 ciclos de borrado/escritura de datos de memoria EEPROM.
- Retención de datos Flash/EEPROM > 40 años.
- Control Auto-programable por software.

---

<sup>8</sup> Bus bidireccional desarrollado por Philips para simplificar la interconexión de dispositivos al microprocesador, esta basado en dos hilos por el que se transmiten los datos vía serie.

- Niveles de prioridad para las interrupciones.
- Temporizador Watchdog Extendido (WDT):
  - Período programable de 41 ms a 131s.
  - Código de Protección de Programable.
- Alimentación simple de 5V en programación serie “In-Circuit Serial Programming”™ (ICSP™) a través de dos pines.
- Depuración “In-Circuit Debug” (CIE) a través de dos pines.
- Amplio rango de voltaje de operación (2.0V to 5.5V).

### Distribución de Pines:

En la figura 1.11 observamos la distribución de pines con sus respectivos nombres del PIC 18F2550.

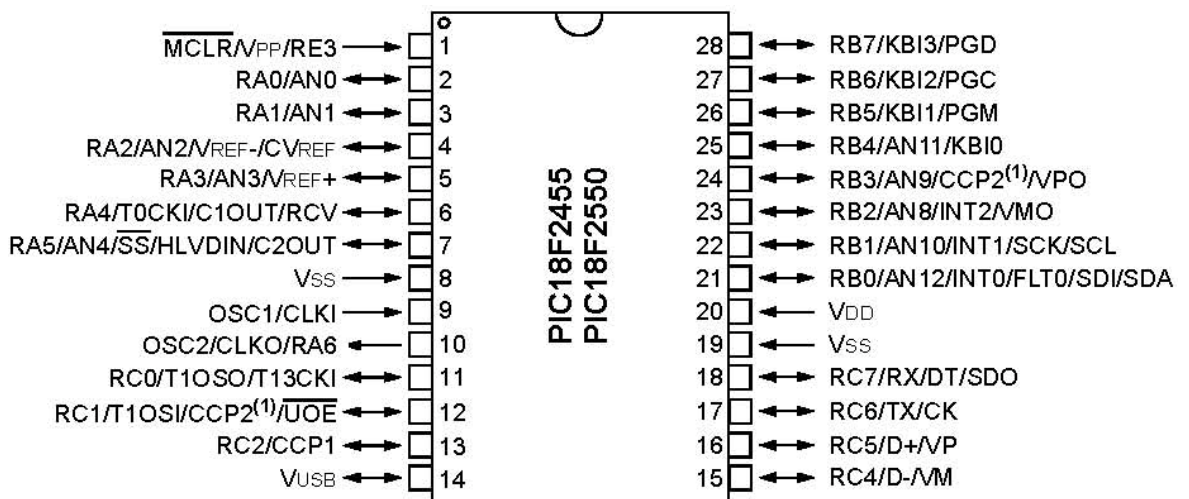


Fig. 1.11: Distribución de pines del PIC18F2550

### 1.3.3. PROGRAMACIÓN DE LOS MICROCONTROLADORES

El PIC usa un juego de instrucciones tipo RISC, cuyo número puede variar desde 35 para PIC's de gama baja a 70 para los de gama alta. Las instrucciones se clasifican entre las que realizan operaciones entre el acumulador y una constante, entre el acumulador y una posición de memoria, instrucciones de condicionamiento

y de salto/retorno, implementación de interrupciones y una para pasar a modo de bajo consumo llamada *sleep*<sup>9</sup>.

Microchip proporciona un entorno de desarrollo freeware llamado MPLAB que incluye un simulador software y un ensamblador. Otras empresas desarrollan compiladores C y BASIC. Microchip también vende compiladores para los PIC's de gama alta ("C18" para la serie F18 y "C30" para los dsPICs) y se puede descargar una edición para estudiantes del C18 que inhabilita algunas opciones después de un tiempo de evaluación.

Para el lenguaje de programación Pascal existe un compilador de código abierto, JAL, lo mismo que PicForth para el lenguaje Forth. GPUTILS es una colección de herramientas distribuidas bajo licencia GNU que incluye ensamblador y enlazador, y funciona en Linux, MacOS y Microsoft Windows. GPSIM es otra herramienta libre que permite simular diversos dispositivos hardware conectados al PIC.

Para transferir el código de un computador al PIC normalmente se usa un dispositivo llamado programador. La mayoría de PIC's que Microchip distribuye hoy en día incorporan ICSP (*In Circuit Serial Programming*, programación serie incorporada) o LVP (*Low Voltage Programming*, programación a bajo voltaje), lo que permite programar el PIC directamente en el circuito destino. Para la ICSP se usan los pines RB6 y RB7 (En algunos modelos pueden usarse otros pines como el GP0 y GP1 o el RA0 y RA1) como reloj y datos y el MCLR para activar el modo programación aplicando un voltaje de 13 voltios. Existen muchos programadores de PIC's, desde los más simples que dejan al software los detalles de comunicaciones, a los más complejos, que pueden verificar el dispositivo a diversas tensiones de alimentación e implementan en hardware casi todas las funcionalidades. Muchos de estos programadores complejos incluyen ellos mismos PIC's preprogramados como interfaz para enviar las órdenes al PIC que se desea programar. Uno de los programadores más simples es el TE20, que utiliza la línea TX del puerto RS232 como alimentación y las líneas DTR y CTS para mandar o

---

<sup>9</sup> <http://electronica-pic.blogspot.com/>

recibir datos cuando el microcontrolador está en modo programación. El software de programación puede ser el ICprog, muy común entre la gente que utiliza este tipo de microcontroladores.

A continuación se tiene una lista de los programadores más usados:

- PICStart Plus (puerto serie y USB).
- Promate II (puerto serie).
- MPLAB PM3 (puerto serie y USB).
- ICD2 (puerto serie y USB).
- PICKit 1 (USB).
- IC-Prog 1.06B.
- PICAT 1.25 (puerto USB2.0 para PIC's y Atmel).
- WinPic 800 (puerto paralelo, serie y USB).
- PICKit 2 (USB).
- PICKit 3 (USB).
- Terusb1.0.
- Eclipse (PIC's y AVR's. USB.).
- Además es posible hacer un programador de manera casera.

Y entre los software de emulación, preferidos por los usuarios, los más utilizados son:

- Proteus – ISIS.
- ICE2000 (puerto paralelo, convertidor a USB disponible).
- ICE4000 (USB).
- PIC EMU.
- PIC CDlite.

El tamaño de palabra de los microcontroladores PIC es fuente de muchas confusiones. Todos los PIC's (excepto los dsPIC) manejan datos en trozos de 8 bits, con lo que se deberían llamar microcontroladores de 8 bits. Pero a diferencia de la mayoría de CPUs, el PIC usa arquitectura Harvard, por lo que el tamaño de

las instrucciones puede ser distinto del de la palabra de datos. De hecho, las diferentes familias de PIC's usan tamaños de instrucción distintos, lo que hace difícil comparar el tamaño del código del PIC con el de otros microcontroladores.

Por ejemplo, un microcontrolador tiene 6144 bytes de memoria de programa: para un PIC de 12 bits esto significa 4096 palabras y para uno de 16 bits, 3072 palabras.

#### **1.3.4. PIC'S EN INTERNET**

Se puede encontrar mucha información y documentación sobre PIC's en Internet principalmente por dos motivos: el primero, porque han sido muy usados para romper los sistemas de seguridad de varios productos de consumo mayoritario (televisión de pago, Play Station...), lo que atrae la atención de los crackers y segundo, porque el PIC16C84 fue uno de los primeros microcontroladores fácilmente reprogramables para aficionados. Hay muchos foros y listas de correo dedicados al PIC en los que un usuario puede proponer sus dudas y recibir respuestas.

Pero también se puede enfocar el tema de internet a la posibilidad que se tiene de desarrollar con estos Sistemas SCADA vía Web, debido a que pueden adquirir y enviar datos al puerto serial de un computador utilizando transmisión UART y el protocolo RS232, o la posibilidad de implementar el protocolo TCP/IP directamente.

## CAPÍTULO II

# DISEÑO E IMPLEMENTACIÓN DEL DATA LOGGER

### 2.1. DESCRIPCIÓN DE LOS SISTEMAS

En la figura 2.1 se presenta un diagrama de bloques con las partes más relevantes del datalogger, las mismas se explicarán a continuación.

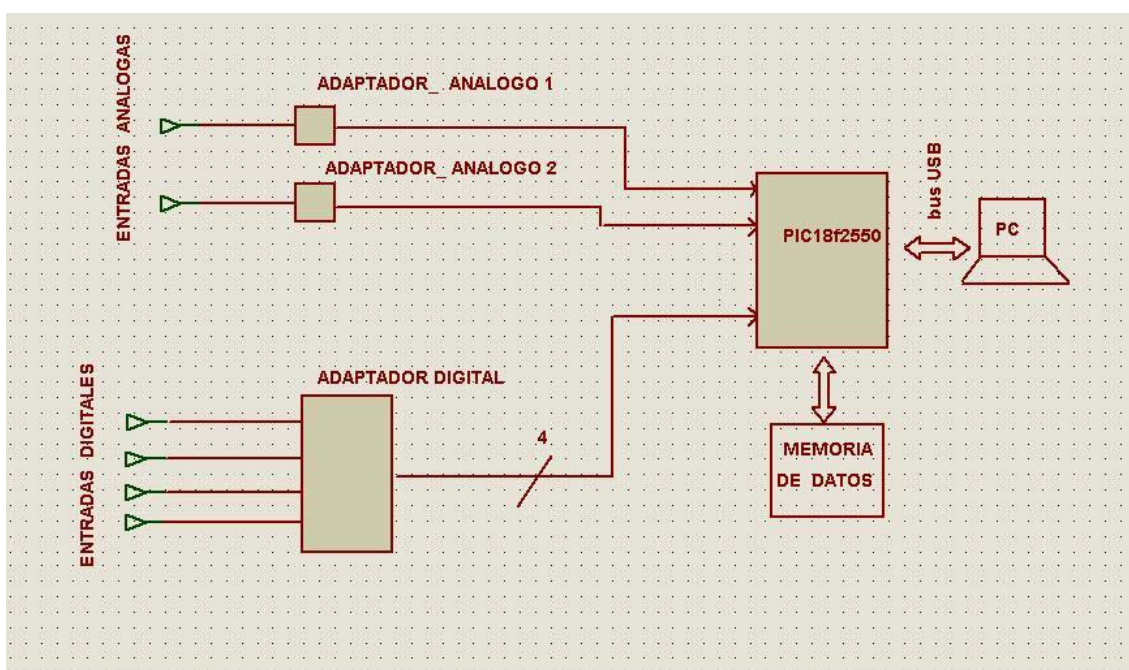


Fig. 2.1: Diagrama de bloques con las partes más relevantes del data logger

#### 2.1.1. SISTEMA ELECTRÓNICO

Entre los elementos más destacados que posee el sistema electrónico del data logger están los siguientes:

- Pantalla LCD: la misma que permite el funcionamiento del data logger independiente de un PC, debido a que en ella se puede observar el valor de las

variables análogas y de las 4 variables digitales que se están adquiriendo en tiempo real (voltios).

- Memoria de almacenamiento de datos no volátil 24LC512, la cual es del tipo serie de 64K x 8 E2PROM; es decir, es una memoria eléctricamente borrrable de 512 Kbits, capaz de operar en un amplio rango de tensiones (2.5 a 5.5v) y a una frecuencia máxima de reloj de 400KHz. Este dispositivo permite realizar lecturas aleatorias y secuenciales de hasta 512 Kbits, el espacio total direccionable puede ser ampliado hasta 4 Mbits mediante la utilización de tres líneas adicionales de direccionamiento que posibilitan la conexión de hasta ocho dispositivos en un mismo bus<sup>10</sup>.
- Acondicionador de señal el cual permite adecuar las señales recibidas para que pueda trabajar con el resto de los elementos internos del circuito, está diseñado con el circuito integrado TL27 el mismo que consta de dos amplificadores operacionales independientes de alta ganancia y con compensación interna de frecuencia, que fueron diseñados específicamente para trabajar con un solo suministro de energía y con una amplia gama de voltajes<sup>11</sup>.
- Microcontrolador PIC18F2550: que posee un cristal externo de 20MHz para su trabajo.
- Divisor de voltaje: para proteger las entradas análogas y el acondicionador de señal descartando diodos zener externos ya que el microcontrolador posee diodos internos de protección en los pines.

### **2.1.2. SISTEMA DE COMUNICACIÓN**

La configuración de la interfaz USB 2.0 se realiza con el PIC18F2550 que posee este tipo de comunicación y lo más importante es que la respuesta que proporciona lo hace en tiempo real.

---

<sup>10</sup> Datasheet 24LC512 de Microchip

<sup>11</sup> Datasheet LM124 de National Semiconductor



Previo la implementación física del dispositivo se usa el módulo ISIS del software PROTEUS el cual es un entorno integrado diseñado para la realización completa de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño, simulación y depuración. Además permite la realización de placas eléctricas mediante el módulo ARES.

En dicho módulo se realizó el circuito que se muestra en la figura 2.2.

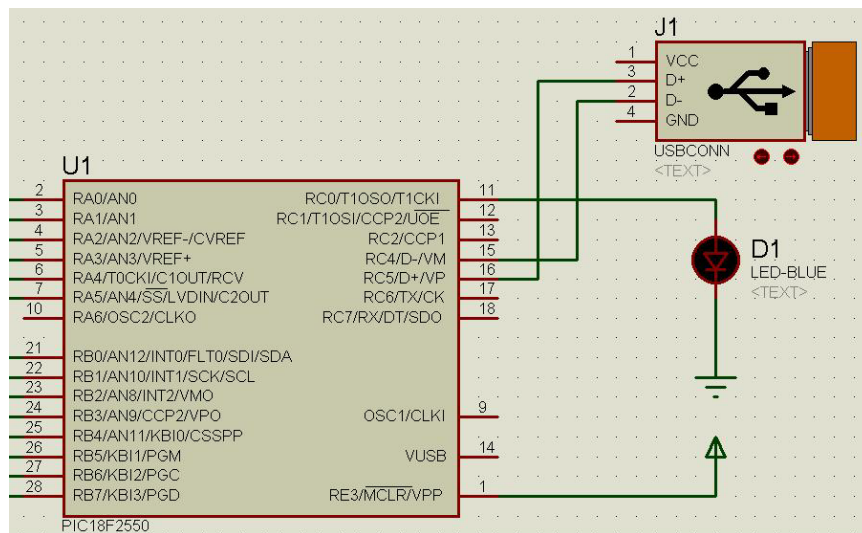


Fig. 2.2: Simulación del circuito en el módulo ISIS del software PROTEUS

En las propiedades del PIC es importante que en “Processor Clock Frequency” el valor sea de 20Mhz debido a que de entre la gama de frecuencias con las que se puede trabajar dicho valor ofrece la mayor velocidad sin comprometer la integridad de los datos y realizando todos los procesos necesarios esto se observa en la figura 2.3.

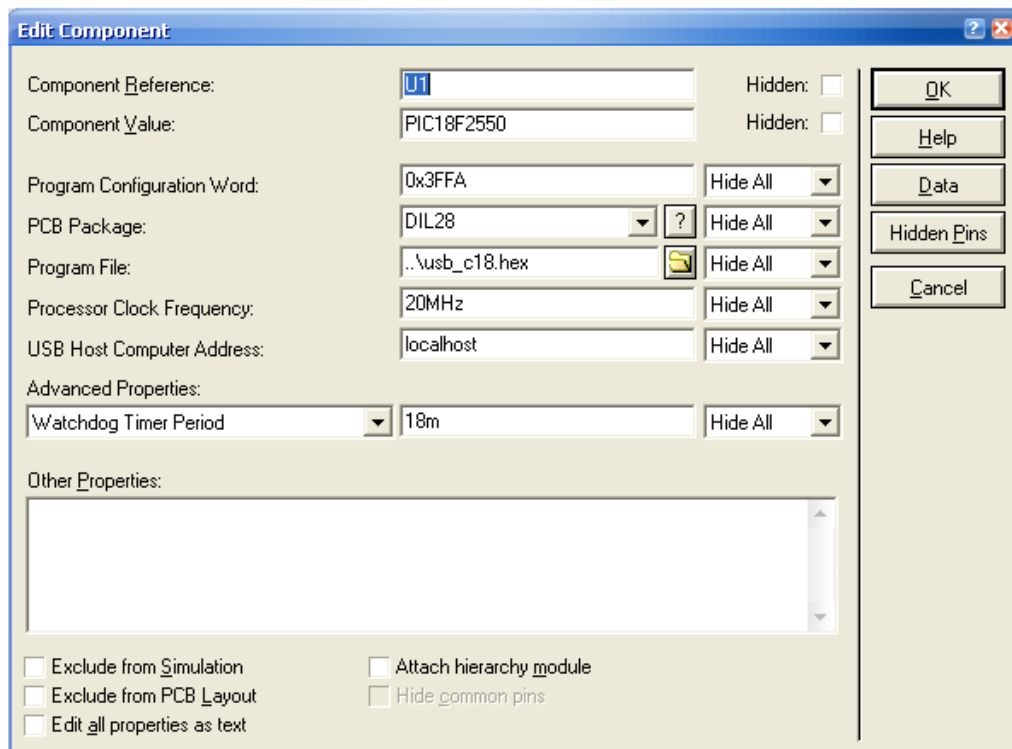


Fig. 2.3: Edición de las propiedades del PIC18F2550

Al ejecutar la simulación, el PC detectará un nuevo hardware, cabe destacar que esté debe utilizar Windows XP SP2 o superior, y solicitará el correspondiente driver.

El driver que permite identificar al dispositivo se maneja en base a archivos de extensión .dll<sup>12</sup>, la librería mpusbapi.dll específicamente; para ejecutarla es necesario tener previamente instalado Microsoft .NET Framework 2.0.

Este archivo dll fue creado por Microchip para brindar mayor facilidad de desarrollo de aplicaciones basadas en el bus USB debido que proporciona las funciones de acceso al puerto USB con un microcontrolador de la familia PIC18Fxx5x.

Para un funcionamiento correcto se necesita el driver mchusb.sys. Este archivo sirve entre otras aplicaciones para Visual Basic como para Visual C, etc.; más

<sup>12</sup> Biblioteca de enlace dinámico: son los archivos con código ejecutable que se cargan bajo demanda de un programa por parte del sistema operativo, esta denominación es exclusiva a los sistemas operativos Windows

importante aún, para LabVIEW que es el software de control que se usará en el proyecto.

La librería mpushapi.dll fue diseñada específicamente para Microsoft Windows XP SP2 esto obliga a la selección del mismo como sistema operativo descartando versiones más recientes como Vista o Siete y versiones anteriores.

### **2.1.2.1. Conversor Analógico/Digital**

El PIC 18F2550 posee un conversor analógico/digital (CAD). Una de las características de este conversor es que tiene dos pines donde pueden ingresarse la tensión de referencia para todos o algunos de los canales del CAD, aunque se toma uno a la vez. Debido a un arreglo RC interno, se toma un nivel de tensión y se lo almacena en el capacitor de carga Chold, el tiempo que tarda ese capacitor en cargarse se determina de 2 maneras:

- Mediante un retardo por software.
- Mediante un retardo por hardware configurando los bits: ACQT2:ACQT0 en el registro ADCON2.

El tiempo de carga del capacitor Chold es fundamental para obtener precisión en el resultado digital.

Otra opción es configurar el tiempo de adquisición de datos del CAD, pues tiene un registro habilitado especialmente para ello ya que en algunas ocasiones hay que esperar al interruptor de muestreo que se cierre y que el condensador (Chold) se descargue para poder hacer otra adquisición. En la figura 2.4 se indica el esquema interno de una de las entradas analógicas del puerto A.

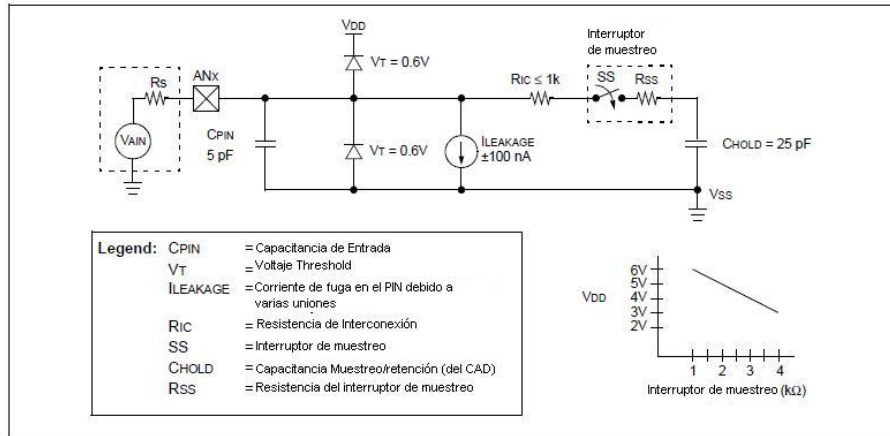


Fig. 2.4: Esquema interno de una de las entradas analógicas del puerto A.

Para habilitar el convertidor analógico se utiliza la función `OpenADC()`<sup>13</sup>, con los argumentos que la siguen.

## 2.2. PROGRAMACIÓN DEL MICROCONTROLADOR PARA EL ALMACENAMIENTO DE DATOS DEL DATA LOGGER Y COMUNICACIÓN CON LA PC

Para la programación del microcontrolador se utiliza el software MPLAB que es una herramienta para escribir y desarrollar código en lenguaje ensamblador para los microcontroladores PIC. MPLAB incorpora herramientas necesarias para la realización de cualquier proyecto, ya que además de un editor de textos cuenta con un simulador en el que se puede ejecutar el código paso a paso para ver así su evolución y el estado en el que se encuentran sus registros en cada momento.

Una vez instalado es posible comenzar a trabajar, para eso se deberá crear un nuevo proyecto utilizando el *Wizard* de MPLAB que se encuentra en el menú *Project -> Project Wizard*, así, aparecerá la pantalla que se observa en la figura 2.5.

<sup>13</sup> Función que permite configurar los 3 registros del PIC, se debe tomar en cuenta que los argumentos de la función `OpenADC()`; cambian con respecto al PIC a utilizar.

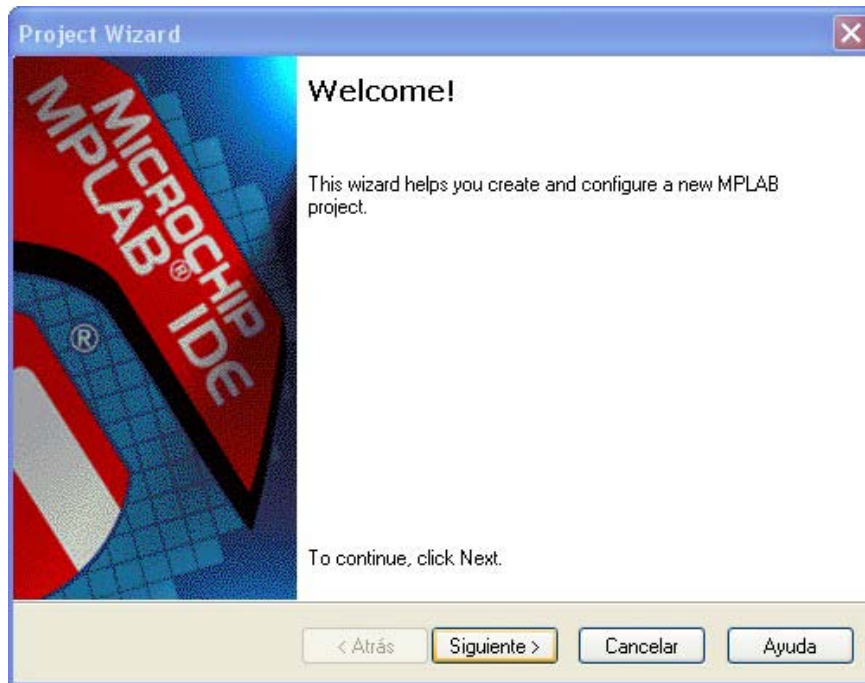


Fig. 2.5: Ventana inicial del Project Wizard de MPLAB.

Dando clic en *Siguiete*, luego aparecerá la ventana presentada en la figura 2.6, donde se escoge el PIC a usarse de una lista que aparece y se da clic en *Siguiete*.

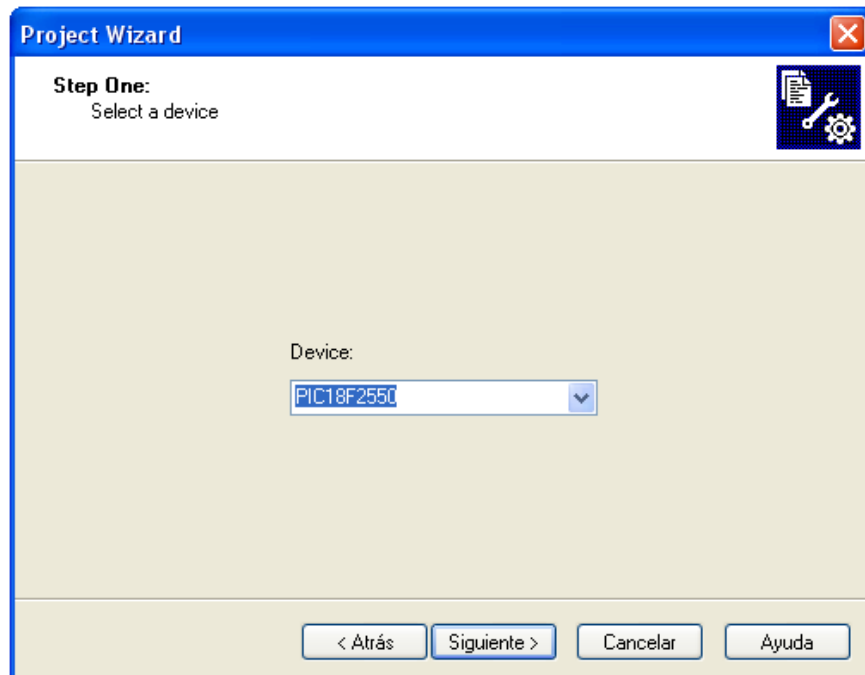


Fig. 2.6: Selección del PIC a utilizar.

El siguiente paso es definir el lenguaje que será usado. En este caso el lenguaje es *Microchip C18 Toolsuite* así que la opción mencionada será la elegida, aparecerá la figura 2.7 y nuevamente clic en *Siguiente*.

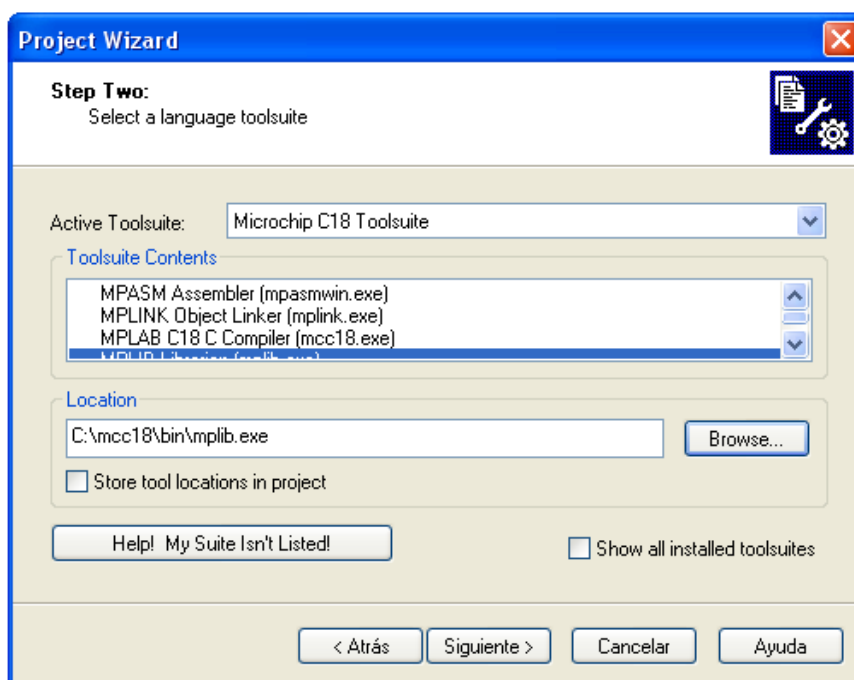


Fig. 2.7: Selección del programa de lenguaje.

En la ventana de la figura 2.8 se elige el nombre y el directorio en el que se guardará el proyecto. Es recomendable que la ruta de la carpeta donde se guarda el proyecto no sea muy larga ya que al compilarlo MPLAB marca un error, es por eso que en el ejemplo la ruta escogida se encuentra cerca de la raíz del disco duro, por ende se recomienda crear una carpeta directamente en el disco "C:\\" o en cualquiera que se use, pero que sea en la raíz del disco. Para este caso la ruta escogida fue *C:\Tesis\Datalogger\* pero se debe tener libertad de escoger cualquier otro nombre e incluso ruta para dicha carpeta.

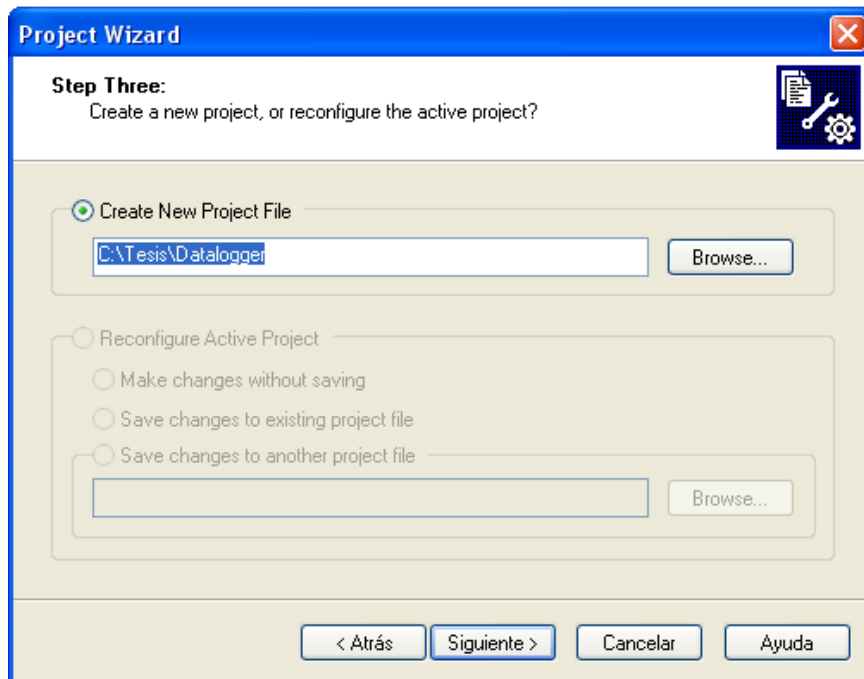


Fig. 2.8: Asignación del nombre y el directorio en el que se guardará el proyecto.

Una vez dado el nombre al proyecto al hacer clic en *Siguiente* se abrirá una nueva ventana, figura 2.9, que pedirá agregar archivos existentes al proyecto, como aún no se ha escrito ningún archivo simplemente clic en *Siguiente* y para terminar en la última ventana clic en *Finalizar*.

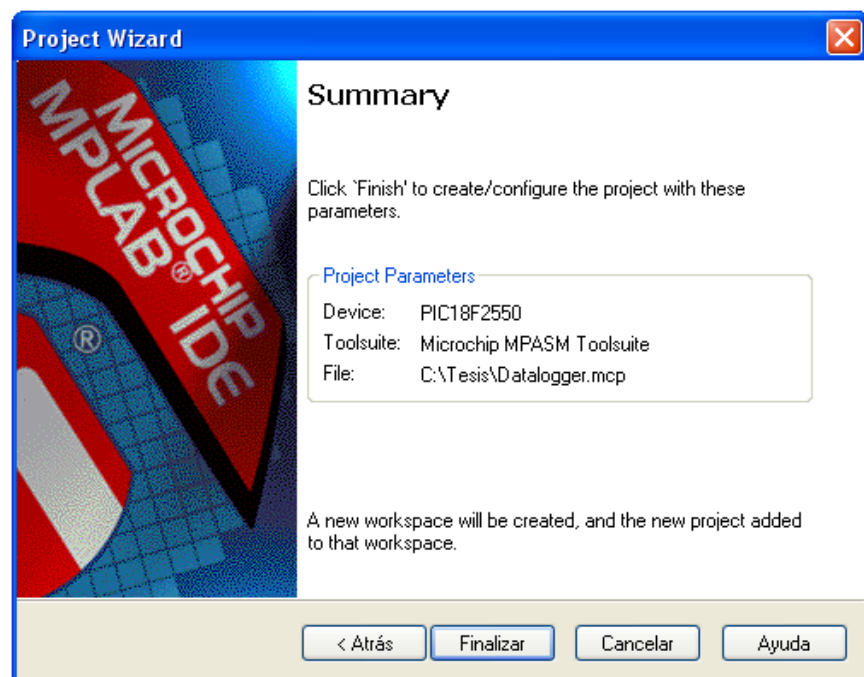


Fig. 2.9: Sumario del proyecto en el cual se visualiza los parámetros del mismo.

Ya creado el proyecto y habiendo escogido Finalizar, se verá la ventana del MPLAB con el aspecto que se observa en la figura 2.10.

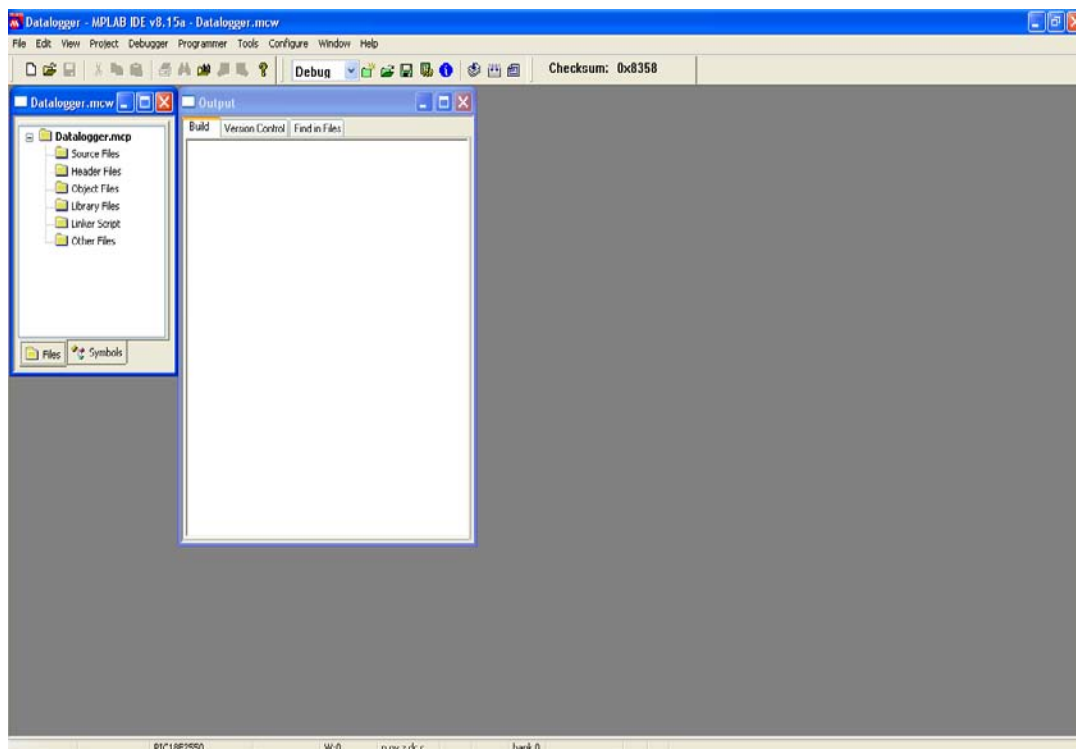


Fig. 2.10: Entorno de trabajo del software MPLab.

A continuación se procede a crear un archivo, es decir a escribir el código que debe ser guardado con extensión .c en la carpeta donde previamente se guardó el proyecto, para crear dicho archivo clic en *File -> New*, después y antes de escribir en el archivo se hará clic en *File -> Save As*. La ventana que se abrirá se indica la figura 2.11.

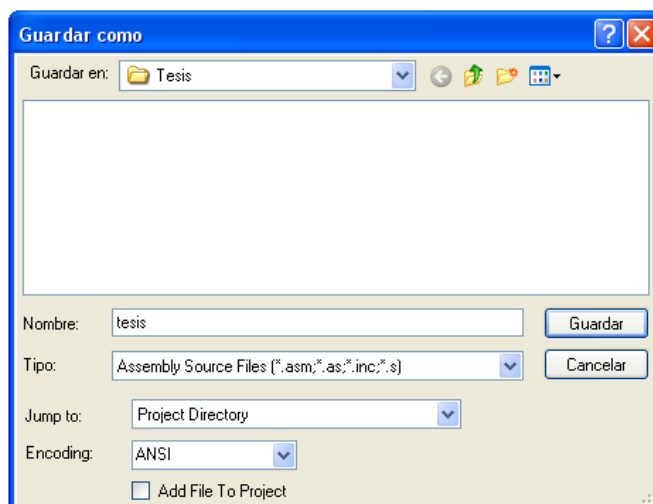


Fig. 2.11: Guardando el archivo con extensión .c.



Ahora el archivo creado tiene extensión .c, pero para el proyecto no sirve, se lo debe agregar al proyecto para después comenzar a trabajar en él, así que en la ventana del proyecto se hace clic derecho en *Source Files* y después seleccionar *Add File*, esto se observa en la figura 2.12.

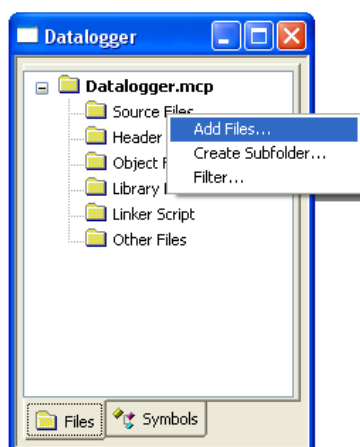


Fig. 2.12: Agregando el archivo con extensión .c al proyecto.

Posteriormente se abrirá una ventana donde se elegirá el archivo que se agregará al proyecto. Por defecto se abrirá la carpeta del proyecto creado, así que se escoge el archivo (en este caso tesis.c) y clic en *Abrir*. Hecho eso la ventana del proyecto debe verse como en la figura 2.13:

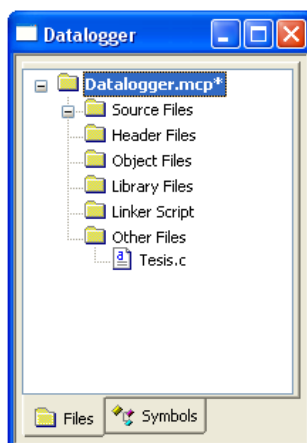


Fig. 2.13: Agregado el archivo con extensión .c al proyecto

Ahora si se puede escribir el código en el archivo tesis.c y todos los cambios que se hará en éste se verán reflejados en el proyecto.

Una vez escrito el código se debe compilar el programa, con esto se genera el archivo.hex con el que se grabará el PIC. Para compilar el programa puede

usarse el menú *Project ->Build All* o usar la combinación Ctrl + F10. El archivo HEX generado se encuentra en el mismo directorio que el proyecto y lleva el mismo nombre que el archivo con el código, en este caso sería tesis.hex.

### **2.3. MANIPULACIÓN DEL MÓDULO USB DEL PIC18F2550**

La comunicación entre el microcontrolador y el PC se dá por medio de la transmisión de datos en formato TTL<sup>14</sup>, los mismos que llegan al PIC18F2550 para luego ser enviados al PC.

El USB es un bus punto a punto: dado que el lugar de partida es el host (PC), el destino es un periférico. No hay más que un único host en una arquitectura USB. Los periféricos comparten la banda de paso del USB. El protocolo se basa en el llamado paso de testigo<sup>15</sup>. El ordenador proporciona el testigo, que es una serie especial de bits que contiene información de control, la posesión del testigo permite a un dispositivo de red transmitir datos a la red en este caso al periférico seleccionado y seguidamente, éste le devuelve el testigo en su respuesta.

Este bus permite la conexión y la desconexión en cualquier momento sin necesidad de apagar el equipo.

### **2.4. COMUNICACIÓN DEL MÓDULO DATA LOGGER A LA PC MEDIANTE LABVIEW**

La herramienta fundamental para conseguir comunicar el módulo datalogger con LabVIEW se llama mpusbapi.dll la misma que es suministrada por Microchip para realizar una comunicación PIC - PC de forma sencilla y eficaz.

La velocidad de estos dispositivos es teóricamente de 12Mbit/seg, pero para ello se necesita una programación compleja, debiendo abrir muchas “pipes” (tuberías) para poder alcanzar dicha velocidad.

---

<sup>14</sup> *Transistor-transistor logic*, es decir, "lógica transistor a transistor". Es una familia lógica o lo que es lo mismo, una tecnología de construcción de circuitos electrónicos digitales

<sup>15</sup> El método de transmisión de datos alrededor de un anillo se denomina paso de testigo (token passing).

Las tuberías son un enlace virtual entre el host (la PC) y el dispositivo USB, este enlace configura los parámetros asociados con el ancho de banda; el tipo de transferencia se va a utilizar (Control, Bulk, Asíncrona o de Interrupción), la dirección del flujo de datos y el máximo y/o mínimo tamaño de los paquetes/buffers. Cada enlace está caracterizado por su banda de paso, su tipo de servicio, el número de punto terminal (End Point) y el tamaño de los paquetes.

En el mundo de los PIC las transferencias más usadas son la tipo Bulk y la de Interrupción, mejor conocidas como CDC o HID respectivamente, básicamente se diferencian en la forma en que transmiten los datos hacia el host y en la forma en que los descriptores están contruidos.

Estos enlaces se definen y crean durante la inicialización del USB. Siempre existe un enlace virtual 0 que permite tener acceso a la información de configuración del periférico USB (estado, control e información).

La norma USB define 2 tipos de enlaces virtuales (pipe): *stream* y *message*.

- Stream Pipes: se trata de un flujo sin formato USB definido, esto significa que se puede enviar cualquier tipo de dato. Este tipo de pipe soporta las transferencias bulk, asíncrona e interrupt. Además, tanto el host como el dispositivo USB pueden controlar.
- Message Pipes: este tipo de enlace virtual si tiene un formato USB definido y sólo puede soportar la transferencia Control.

También se debe mencionar el Proceso de Enumeración que se produce cuando se conecta un dispositivo USB a la PC, el cual consiste en que el host le indica al dispositivo que se presente y diga cuáles son sus parámetros, tales como:

- Consumo de energía expresada en unidades de carga.
- Número y tipos de puntos terminales.
- Clase del producto.

- Tipo de transferencia.
- Razón de escrutinio, etc.

El proceso de enumeración es inicializado por el host cuando detecta que un nuevo dispositivo ha sido adjuntado al Bus. El host le asigna una dirección al dispositivo adjuntado y habilita su configuración permitiendo la transferencia de datos sobre el bus.

#### **2.4.1. PREÁMBULO DE COMUNICACIÓN**

El tipo de preámbulo define la longitud del bloque de checksum<sup>16</sup> para la comunicación entre el dispositivo USB y la PC. Se utiliza para detectar errores en la transmisión de datos.

En LabVIEW, para llamar a la librería necesaria para entablar la comunicación, primero se deberá identificar qué tipo de DLL es. Hay esencialmente dos tipos de DLL a considerar: C/C++ DLL y ensamblados de Microsoft.NET; mpubapi.dll pertenece al primer tipo, una vez hecho esto en el diagrama de bloques se utilizará la función “Call Library” esta función permite llamar directamente a un archivo DLL o a una función de librería compartida. Esta función soporta un gran número de tipos de datos también puede usarse para llamar a funciones en la mayoría de estándares, archivos DLL especialmente diseñados y bibliotecas compartidas. La función “Call Library” es expandible y muestra los tipos de datos para las entradas y salidas de cable como se puede observar en la figura 2.14.

---

<sup>16</sup> Una suma de verificación o checksum es una forma de control de redundancia, una medida muy simple para proteger la integridad de datos, verificando que no hayan sido corruptos. Es empleado para comunicaciones tanto como para datos almacenados.

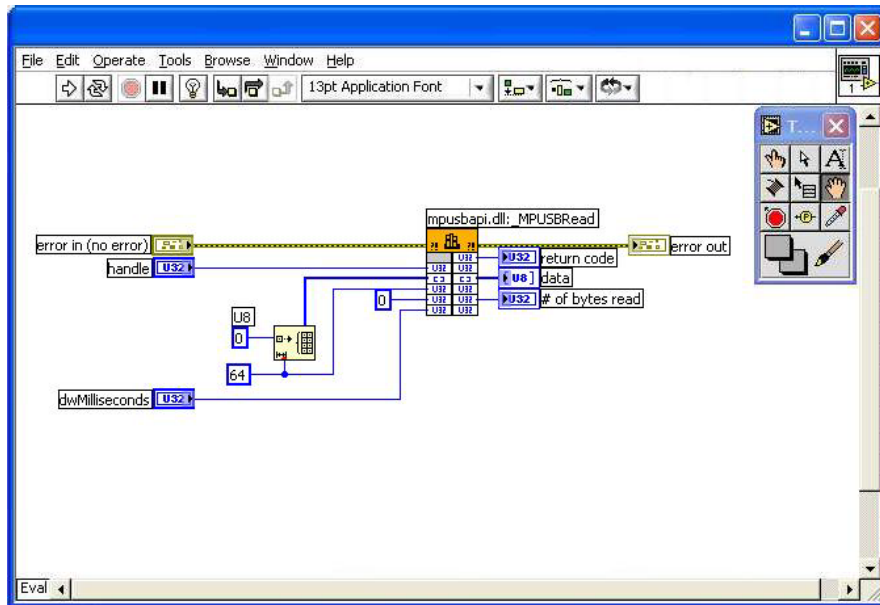


Fig. 2.14: Configuración de la función “Call Library”.

Al dar doble clic sobre dicha función aparecerá un menú para configurar la DLL a cargar, así como qué función se va exportar y cuál es su prototipo, esto se puede observar en las figuras 2.15 y 2.16. Para ello habrá que declarar los tipos de los distintos parámetros de entrada y de salida de la función, así como si el paso de los mismos se hace por valor o por referencia.

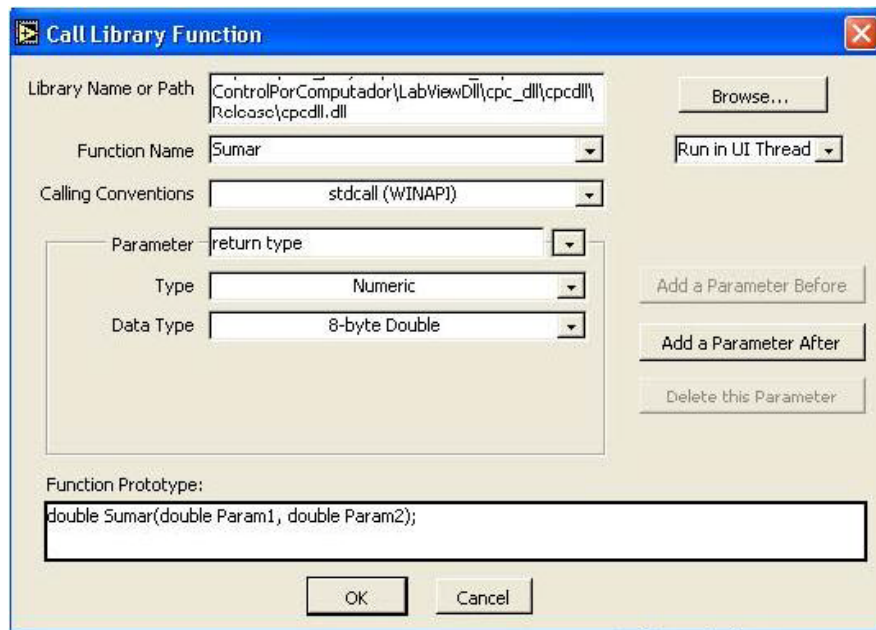


Fig. 2.15: Menú de configuración. Paso de parámetros por valor.

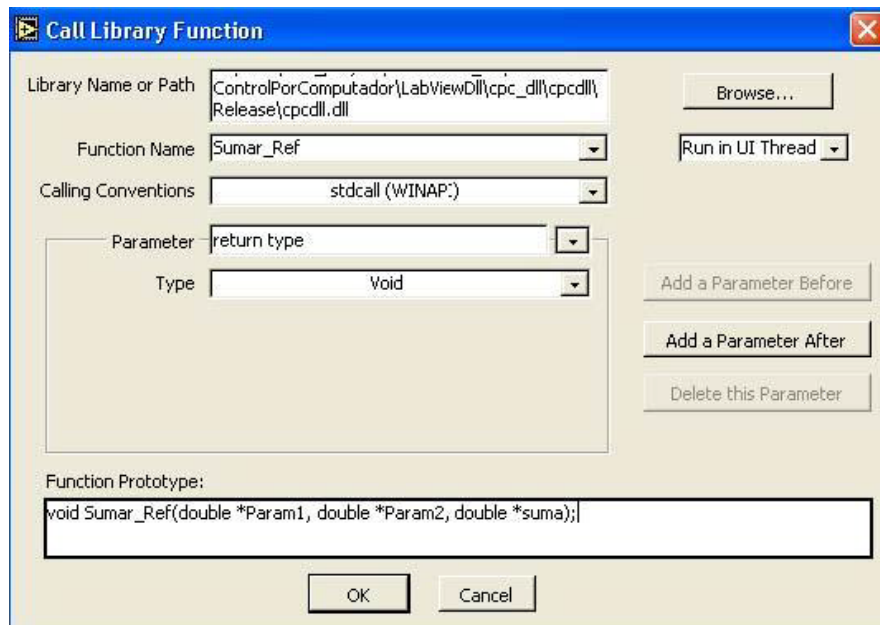


Fig. 2.16: Menú de configuración. Paso de parámetros por referencia.

## 2.5. ACOPLAMIENTO DEL SISTEMA TOTAL

Hablar sobre el Universal Serial Bus es complejo porque a diferencia de otras transmisiones seriales como la RS-232, RS-485, etc. la transmisión por este medio es más elaborada, desde la construcción de las tramas<sup>17</sup> hasta los descriptores que la componen; además de los métodos de transferencia que existen dentro del mismo USB.

Para usar la comunicación USB con los PIC, Microchip ha liberado una serie de microcontroladores que incluyen un módulo interno para trabajar con dicho protocolo. MPLAB posee un framework<sup>18</sup> liberado por Microchip para poder trabajar con estos modelos, para la realización de este proyecto se utilizará la versión 1.3 ya que no es muy compleja para trabajar y es sencilla de editar.

Igualmente se utilizarán los ejemplos que Microchip facilita, teniendo como base MCHPFSUSB que no es otra cosa que un paquete de distribución que contiene

<sup>17</sup> Una trama de datos, viene a ser lo mismo que un paquete de datos (la información se envía por trozos, no entera). Estos paquetes constan de cabecera (donde van los protocolos de enlace), datos (la información) y cola (donde suelen ir un chequeo de errores).

<sup>18</sup> Un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos de software concretos, mediante la cual otro proyecto de software puede ser organizado y desarrollado.

una variedad de proyectos "USB firmware" relacionados a las series PIC18, PIC24F, PIC32, junto con otros relacionados con los controladores USB y los recursos destinados para el uso en la PC.

Todos los archivos generados por Microchip que se ilustran en la figura 2.17 son los que se usarán, además se contempla la inclusión del archivo .lkr necesario para generar el archivo .hex que será el que se ejecute en el microcontrolador.

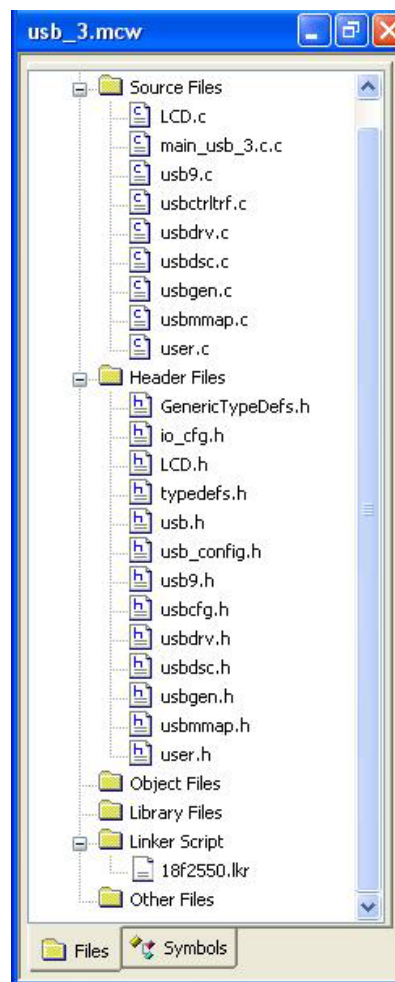


Fig. 2.17: Ejemplos de Microchip para usar la comunicación USB - PIC

Se usará un PIC 18F2550 que gestione mediante software la comunicación USB 2.0 con la PC, el cual se configurará de acuerdo a las necesidades del usuario, por ejemplo se pondrá el nombre de la empresa y el nombre de identificación de éste.

También hay que configurar dos parámetros esenciales en todo dispositivo que son el VID e ID.

VID (Vendor ID):

Es el identificador de la empresa. Cada empresa tiene un identificador, el cual puede vender diferentes productos a los que se les incluye el ID.

ID (Product ID):

Es el identificador del producto. Por tanto un dispositivo tendrá un VID e ID fijos que les hace únicos con lo que no pueden sufrir incompatibilidad con otros dispositivos.

Estos son únicos para cada dispositivo que existe en el mercado, ya que si existiera en el computador dos dispositivos con el mismo VID/ID causarían conflictos entre ellos, impidiendo el correcto funcionamiento de ambos.

Para evitar la existencia de conflictos entre dispositivos considerando la enorme cantidad de éstos en el mercado informático y electrónico se pagará por tener un VID/ID único e indivisible, claro está que al ser éste un proyecto que no se producirá en masa para venta al público no habrá necesidad de comprar un identificador para el dispositivo.

### **2.5.1. INSTALACIÓN DEL MICROCONTROLADOR**

Para el proyecto es fundamental implementar una comunicación bidireccional serie, Communications Devices Class<sup>19</sup>, entre el PIC y cualquier Software del PC, en este caso LabVIEW, que use un puerto COM Serie estándar pero emulándolo vía USB 2.0 para esto se realizarán ajustes necesarios en el Firmware del PIC para que sea reconocido por el sistema operativo como un dispositivo Serie estándar y lo registre como un puerto COM Virtual. Microchip provee un recurso que realiza las veces de intermediario, el driver mchpcdc.inf

---

<sup>19</sup> USB CDC es una clase de dispositivos USB compuestos el cual proporciona un rango único a cada dispositivo



para Windows por ende es imprescindible instalarlo pues es en este archivo donde se definirán los VID e ID a utilizar.

En la tabla 2.1 se observa el cuerpo del archivo mchpcdc.inf tal cual Microchip lo ofrece salvo la parte final donde se encuentra la información propia del proyecto.

Tabla 2.1: Contenido del archivo mchpcdc.inf de Microchip.

```
; Installation file for mchpusb driver
; Copyright (C) 2004 by Microchip Technology, Inc.
; All rights reserved

[Version]
Signature=$CHICAGO$
Class=Unknown
ClassGuid={4D36E97E-E325-11CE-BFC1-08002BE10318}
Provider=%ONEYSOFT%
CatalogFile=mchpusb.cat
DriverVer=12/13/2005

[Manufacturer]
%MFGNAME%=DeviceList

[DestinationDirs]
DefaultDestDir=10,System32\Drivers

[SourceDisksFiles]
mchpusb.sys=1
wdmstub.sys=1

[SourceDisksNames]
1=%\INSTDISK%,,,

[DeviceList]
%DESCRIPTION%=DriverInstall,USB\VID_04D8&PID_0011 ;
Modificable

;-----
; Windows 2000/XP Sections
;-----

[DriverInstall.ntx86]
CopyFiles=DriverCopyFiles

[DriverCopyFiles]
mchpusb.sys,,,2

[DriverInstall.ntx86.Services]
AddService=MCHPUSB,2,DriverService

[DriverService]
ServiceType=1
StartType=3
```

```
ErrorControl=1
ServiceBinary=%10%\system32\drivers\mchpubs.sys
AddReg=TraceFlags

;-----
; Windows 98/Me Sections
;-----

[DriverInstall]
AddReg=DriverAddReg
CopyFiles=DriverCopyFiles,StubCopyFiles
DriverVer=10/12/2009 ; Modifiable

[DriverAddReg]
HKR,,DevLoader,,*ntkern
HKR,,NTMPDriver,, "wdmstub.sys,mchpubs.sys"

[StubCopyFiles]
wdmstub.sys,,2

;-----
; String Definitions
;-----

[Strings] ; Modifiable
ONEYSOFT="J R "
MFGNAME="ESPE"
DESCRIPTION="ESPE JORGE RIVERA"
```

Una vez configurado el driver, lo siguiente será la instalación del hardware con el puerto USB. Lo primero será conectar el dispositivo a algún puerto libre, al hacerlo inmediatamente se abrirá el Asistente para Hardware nuevo encontrado, como se puede notar en la figura 2.18, donde se seleccionará “Instalar desde una lista o ubicación específica” y clic en siguiente.

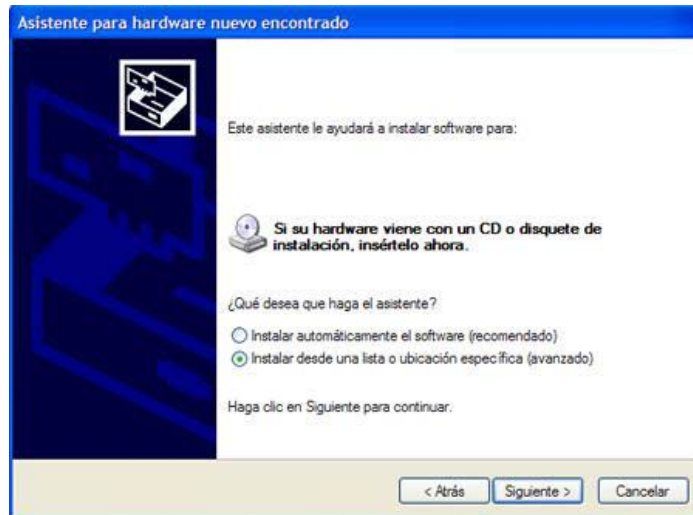


Fig. 2.18: Asistente para Hardware nuevo encontrado.

En la siguiente ventana para agilizar el proceso de detección del driver indicamos su ubicación utilizando la opción “Examinar” una vez hecho esto, dar clic en siguiente. Este proceso se observa en la figura 2.19.

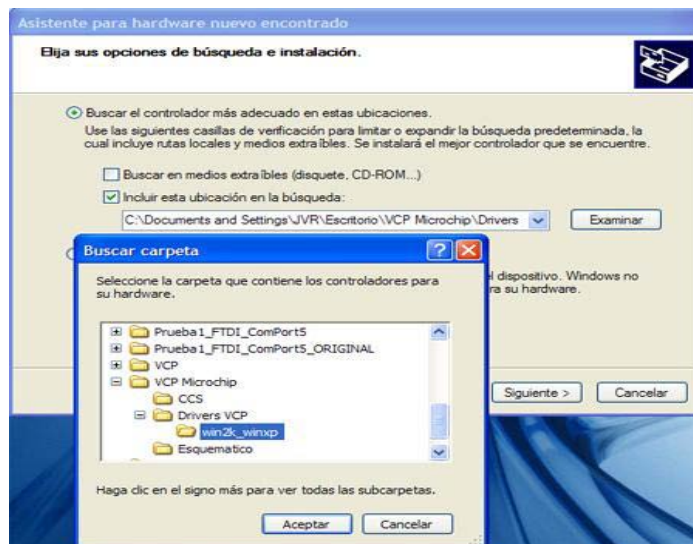


Fig. 2.19: Selección del driver desde su carpeta.

En este momento empezará la búsqueda de los archivos identificadores del dispositivo y una vez encontrados se podrá ver en la ventana el nombre asignado al dispositivo, pero para que se inicie la instalación Windows preguntará al usuario si desea realizar la misma ya que el software no posee ninguna firma digital y lo confunde con software mal intencionado, esto se nota en las figuras 2.20 y 2.21 respectivamente.

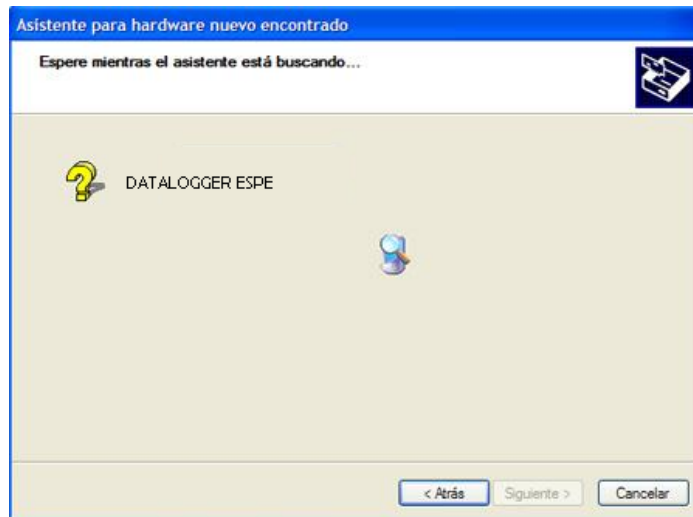


Fig. 2.20: Identificación del hardware conectado al puerto.

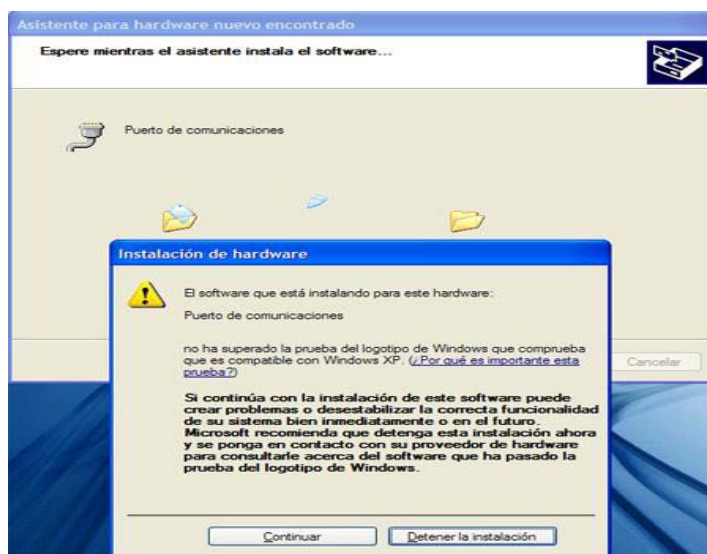


Fig. 2.21: Inicio del proceso de instalación del driver previa consulta.

Si el proceso se realizó de manera correcta en la última ventana, figura 2.21, se informará que el software ha sido instalado correctamente

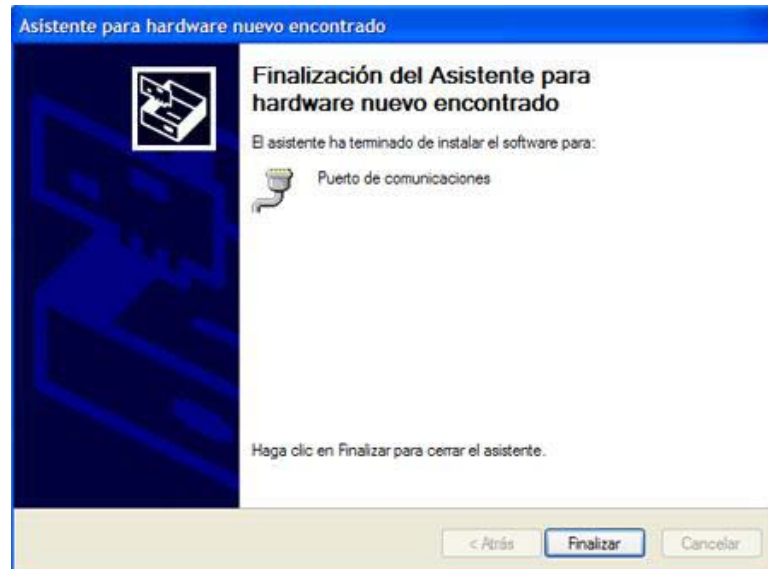


Fig. 2.22: Finalización de la instalación del software.

### **2.5.2. OBTENCIÓN DE 48MHz EN EL PIC 18FXX5X A PARTIR DE OTROS VALORES DE FRECUENCIA**

Los USB 2.0 utilizan 48 MHz, por lo que es necesario conseguir esta frecuencia internamente con la ayuda de recursos adicionales que poseen los PIC's de la familia 18Fxx5x.

Para empezar se tomará como referencia la figura 2.23 donde aparece la estructura de configuración de las opciones de los osciladores en los PIC de la familia 18Fxx5x que son los que soportan el USB 2.0.

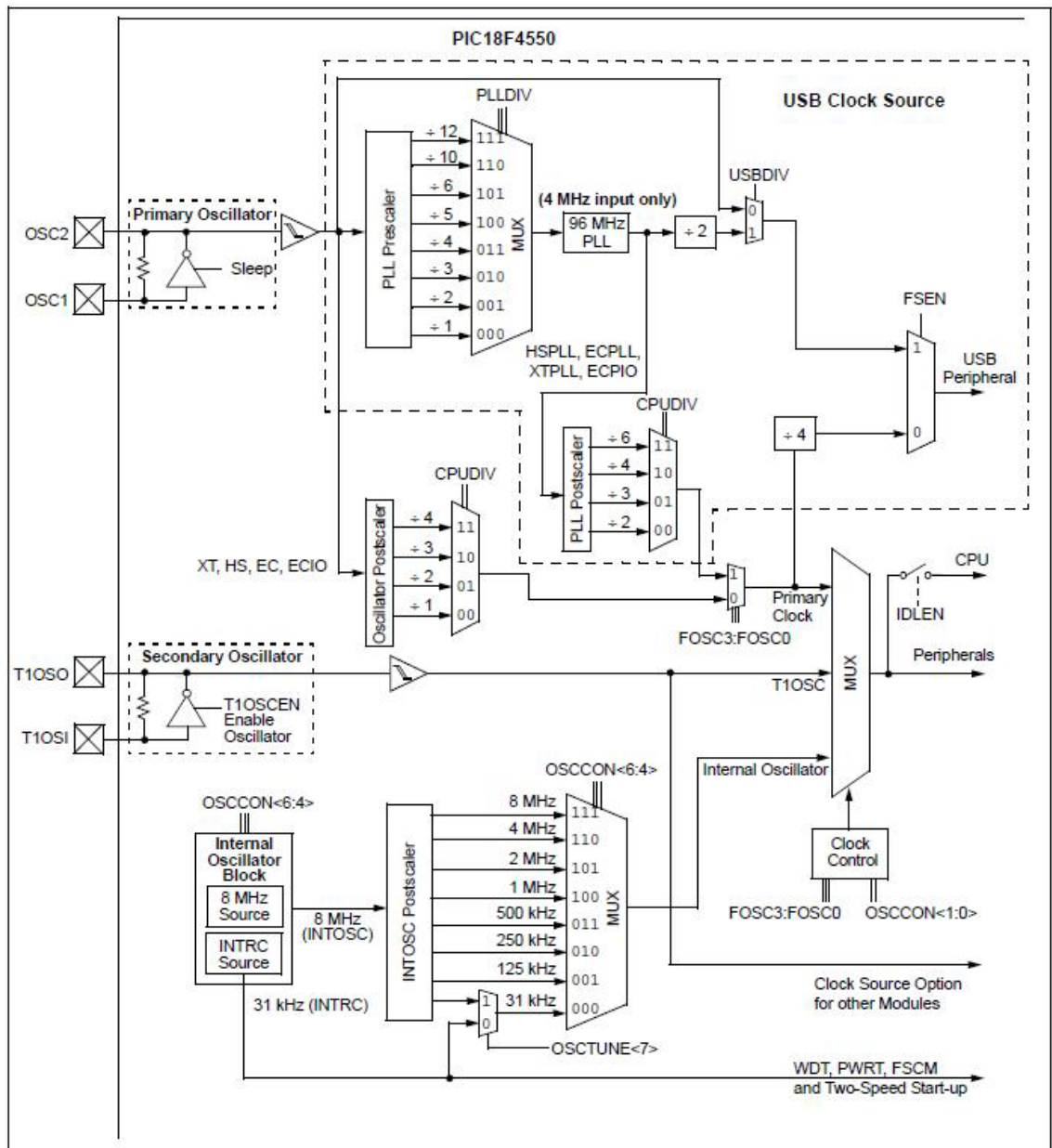


Fig. 2.23: Estructura de configuración de los osciladores en los PIC 18FXX5X.

Este sistema es complejo debido a la inmensa versatilidad y posibilidades que ofrece.

Como se puede apreciar en OSC1 y OSC2 se conecta el cristal, el cual puede variar entre: 4 MHz, 8 MHz, 12 MHz, 16 MHz, 20 MHz, 24 MHz, 40 MHz ó 48 MHz, que son los cristales válidos compatibles. Se debe considerar que, según la configuración final adoptada, este cristal puede ser el mismo para

generar la frecuencia de 48 MHz necesaria para el USB 2.0 y para el Clock<sup>20</sup> del PIC, o no; en la práctica esto quiere decir que podrá el USB funcionar a 48MHz mientras que el programa en el PIC funcionará a 12 MHz por ejemplo. Cualquier otro cristal debe ser tratado para conseguir los 48 MHz necesarios en el USB.

El módulo USB Clock Source tiene a su entrada un PLL Prescaler<sup>21</sup>, o sea un divisor de frecuencia. En cada una de sus salidas vamos a tener FOSC<sup>22</sup> dividida por 1, 2, 3, 4, 5, 6, 10 ó 12 y mediante el PLLDIV, que no es más que un multiplexor, se seleccionará la entrada que se desea usar.

El data logger fue diseñado para un cristal de 20 MHz y en el PLLDIV se colocó un 100, entonces se divide por 5 FOSC con lo que se obtiene también 4 MHz a la salida del multiplexor.

La salida del multiplexor es la que se utiliza para variar al PLL; quiere decir que si a la entrada se tiene 4 MHz se generará 96 MHz, esta capacidad de pasar de 4 MHz a 96 MHz brinda la posibilidad de usar cristales distintos.

Para alcanzar los 48 MHz requeridos se utiliza un divisor por 2 que es la segunda alternativa para llegar a USBDIV y en este caso se le puso un 1 para usar la señal proveniente del PLL.

Cabe recalcar que además de proveer la señal oscilante en USBDIV también se conecta la señal del PLL a 96 MHz en un Postscaler, otro divisor, en este caso por 2, 3, 4 ó 6 y cuyas señales van al CPUDIV, o sea podrá generar una señal de reloj para el PIC y no para el USB, sino para la velocidad de ejecución del programa tomándola del PLL y que puede ser de 16 MHz, 24 MHz, 32 MHz ó 48 MHz, la cual se selecciona con el switch FOSC3:FOSC0 de donde se obtendrá la frecuencia definitiva de ejecución de programas.

---

<sup>20</sup> Señal de reloj

<sup>21</sup> Es un circuito electrónico de conteo utilizado para reducir la alta frecuencia de la señal eléctrica a una frecuencia más baja.

<sup>22</sup> Es una señal generada internamente por el PIC tomada del circuito de reloj y que es igual a la frecuencia del oscilador dividida.

Como se puede ver toda esta gama de combinaciones permitirán al PIC trabajar a decenas de velocidades distintas; siendo capaz al mismo tiempo, de tener disponibles los 48 MHz imprescindibles para el USB 2.0.

Todo lo antes redactado se resume en la figura 2.24, en la cual se resalta con línea roja la configuración realizada para obtener 48MHz del USB 2.0 a partir de un cristal de 20MHz.

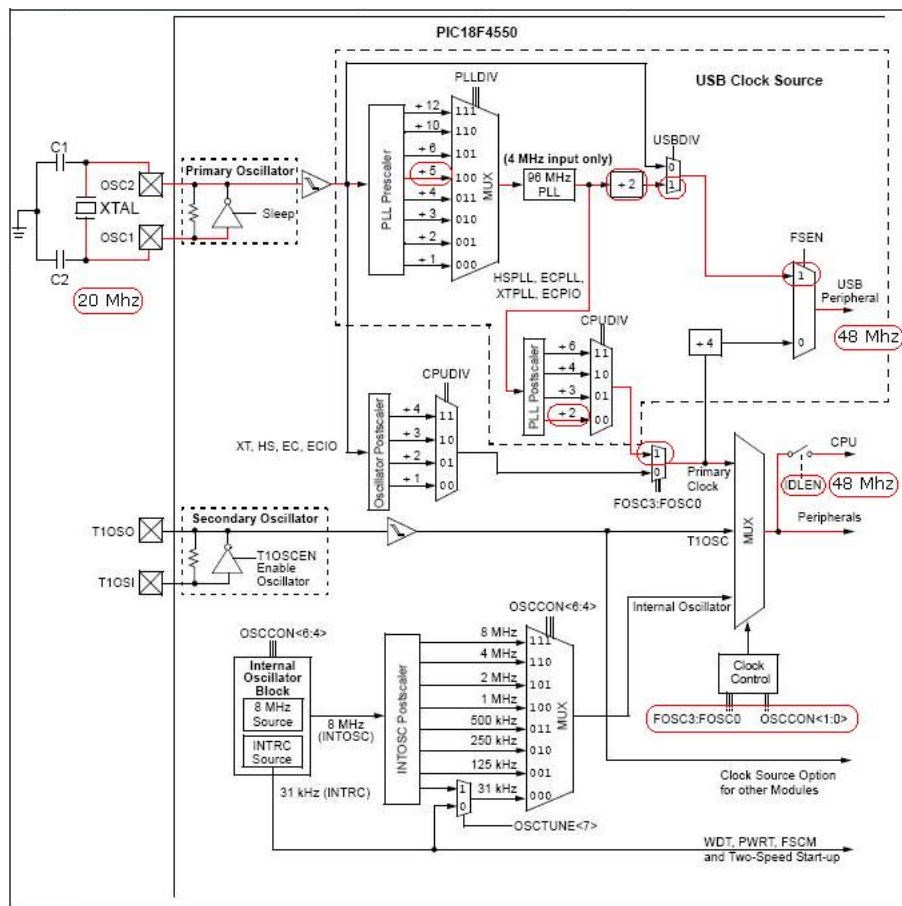


Fig. 2.24: Configuración realizada para obtener 48MHz a partir de un cristal de 20MHz.



# CAPÍTULO III

## PRUEBAS EXPERIMENTALES Y ANÁLISIS DE RESULTADOS

### 3.1. PRUEBAS DE ADQUISICIÓN Y ALMACENAMIENTO DEL DATA LOGGER

El data logger está diseñado para adquirir datos en tiempo real y la vez almacenarlos, el rango mínimo de almacenamiento es de 1 segundo y el máximo es de 23:59:59. Todo esto se puede configurar en la ventana principal del software de control, la representación de la misma se encuentra a continuación en la figura 3.1.

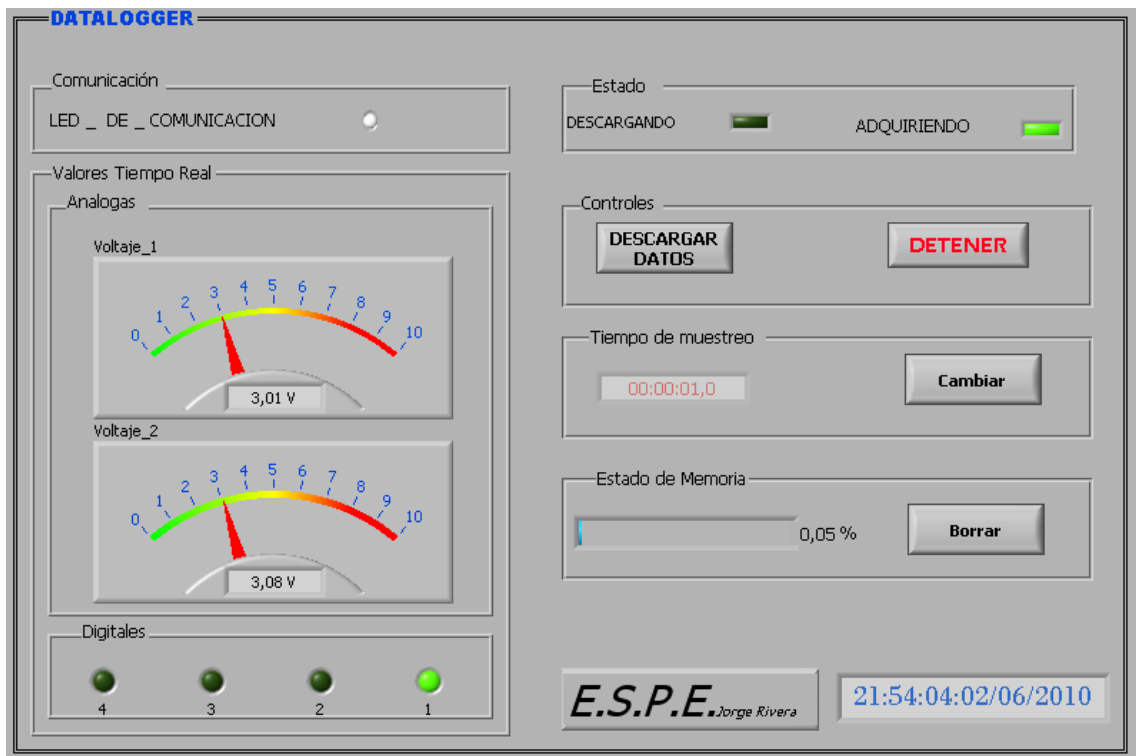


Fig. 3.1: Ventana principal de control y configuración.

En las pruebas de adquisición se visualiza en tiempo real la variación que existe entre la señal de entrada al data logger y la herramienta de visualización de señal representada en la figura 3.2.

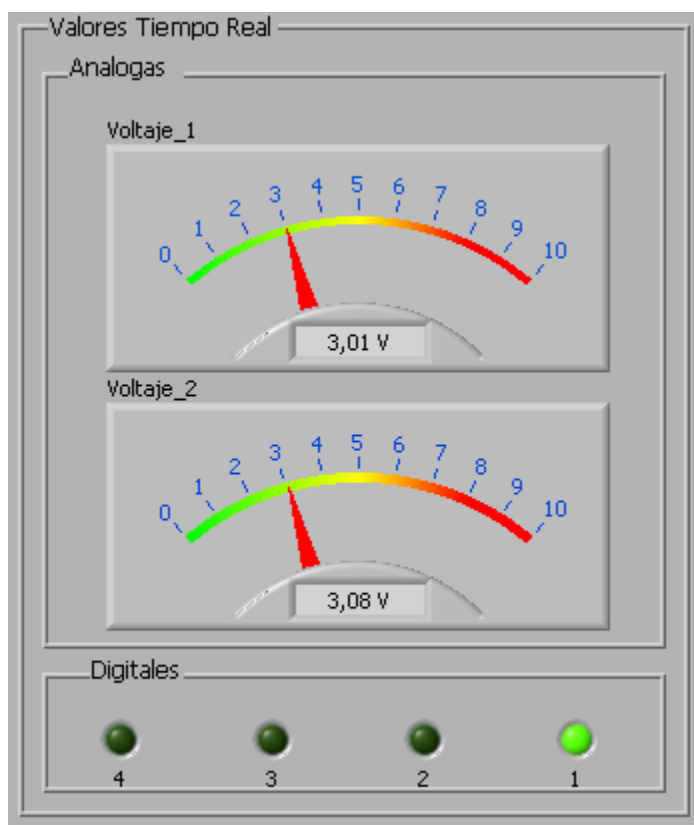


Fig. 3.2: Herramienta de visualización de señal en tiempo real.

Esta es la sección que está encargada de la adquisición de los dos tipos de señales: análogas y digitales, el valor medido y registrado de las entradas análogas poseen una exactitud del  $\pm 5\%$ , la diferencia se genera en el acondicionador de señal.

Al ser autónomo el dispositivo puede obtener y presentar los valores sin necesidad del PC, esto gracias al LCD que posee el módulo data logger, en el cual se visualiza la información de las dos entradas análogas y el estado de las cuatro entradas digitales, la presentación de dicha información es como se representa en la figura 3.3.

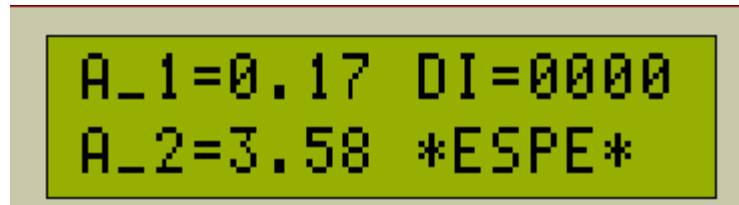


Fig. 3.3: Visualización de los estados de las entradas mediante el LCD.

La comunicación con la PC en tiempo real y la medición de los valores reales ha funcionado sin problemas por cortos lapsos de tiempo, así que la siguiente comprobación por realizar fue la adquisición y almacenamiento de la información para un mayor período de tiempo; para eso se tomaron varias muestras en un tiempo de 1 segundo, las mismas que se guardaron en un archivo de extensión .xls<sup>23</sup>, acto seguido se procedió a la descarga de los datos existentes en la memoria, los resultados obtenidos se reflejan en la tabla 3.1 que se presenta a continuación.

Tabla 3.1: Primeros datos obtenidos en las pruebas del data logger.

Archivo de Descarga de Datos					
fecha : 13/01/2010 : 10:45					
DIGITAL_1	DIGITAL_2	DIGITAL_3	DIGITAL_4	Voltaje_1 (V)	Voltaje_2 (V)
0	1	1	0	18,92	2,92
0	1	1	0	18,93	2,93
0	1	1	0	18,93	2,93
0	1	1	0	18,93	2,93
0	1	1	0	18,93	2,93
0	1	1	0	18,93	2,93
0	1	1	0	18,93	2,93
0	1	1	0	18,93	2,93
0	0	0	0	19,8	3,8
0	0	0	0	19,9	3,9
0	0	0	0	19,8	3,8
0	0	0	0	19,8	3,8
0	0	0	0	19,8	3,8
0	0	0	0	19,9	3,9
1	0	0	0	37,8	3,8
1	0	0	0	37,7	3,7
1	0	0	0	37,8	3,8

<sup>23</sup> Extensión para los archivos de hoja de cálculo de Microsoft Excel.

1	0	0	0	37,8	3,8
1	0	0	0	37,8	3,8
1	0	0	0	37,8	3,8
1	0	0	0	37,8	3,8
1	0	0	0	37,8	3,8
1	0	0	0	3,8	3,8
1	0	0	0	3,8	3,8
1	0	0	0	3,9	3,9
1	0	0	0	3,8	3,8
1	0	0	0	3,9	3,9
1	0	0	0	3,7	3,7
1	0	0	0	3,9	3,9
1	0	0	0	3,7	3,7
1	0	0	0	3,7	3,7
1	0	0	0	3,7	3,7
1	0	0	0	3,24	3,9
1	0	0	0	3,25	3,7
1	0	0	0	3,26	3,8
1	0	0	0	3,26	3,9
1	0	0	0	3,25	3,8
1	0	0	0	3,25	3,7
1	0	0	0	3,26	3,8
1	0	0	0	3,24	3,7
1	0	0	0	3,4	3,9
1	0	0	0	3,4	3,8
1	0	0	0	3,4	3,9
1	0	0	0	3,3	3,8
1	0	0	0	3,4	3,9
1	0	0	0	2,56	2,56
1	0	0	0	3,22	3,8
1	0	0	0	3,2	3,7
0	1	1	0	2,88	2,92
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,92
0	1	1	0	2,89	2,93

0	1	1	0	2,88	2,93
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,92
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,92
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,92
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,92
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,92
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,88	2,92
0	1	1	0	2,89	2,93
0	1	1	0	2,88	2,93
0	1	1	0	2,89	2,93
0	1	1	1	2,88	3,95
0	1	1	1	2,88	3,93
0	1	1	1	2,89	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,95
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,87	3,93
0	1	1	1	2,88	3,93

0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,95
0	1	1	1	2,87	3,93
0	1	1	1	2,88	3,95
0	1	1	1	2,87	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,87	3,93
0	1	1	1	2,88	3,95
0	1	1	1	2,87	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,95
0	1	1	1	2,88	3,93
0	1	1	1	2,89	3,95
0	1	1	1	2,87	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,87	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,89	3,95
0	1	1	1	2,87	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,87	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,89	3,95
0	1	1	1	2,87	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,87	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,89	3,95
0	1	1	1	2,87	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,87	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,93

0	1	1	1	2,88	3,93
0	1	1	1	2,89	3,95
0	1	1	1	2,87	3,93
0	1	1	1	2,88	3,95
0	1	1	1	2,87	3,93
0	1	1	1	2,89	3,95
0	1	1	1	2,88	3,93
0	1	1	1	2,88	3,95
0	1	1	0	2,84	2,95
0	1	1	0	2,82	2,93
0	1	1	0	2,84	2,93
0	1	1	0	2,82	2,93
0	1	1	0	2,84	2,95
0	1	1	0	2,82	2,92
0	1	1	0	2,82	2,93
0	1	1	0	2,82	2,92
0	1	1	0	2,84	2,93
0	1	1	0	2,81	2,92
0	1	1	0	2,82	2,93
0	1	1	0	2,81	2,92
0	1	1	0	2,84	2,93
0	1	1	0	2,81	2,93
0	1	1	0	2,82	2,95
0	1	1	0	2,82	2,93
0	0	0	0	3,13	37,13
0	0	0	0	3,13	37,15
0	0	0	0	3,15	37,12
0	0	0	0	3,12	37,13
0	0	0	0	3,13	37,12
0	0	0	0	3,12	37,15
0	0	0	0	3,15	37,13
0	0	0	0	3,13	37,13
0	0	0	0	3,13	37,13
0	0	0	0	3,13	37,15
0	0	0	0	3,15	37,12
0	0	0	0	3,12	37,13
0	0	0	0	3,13	37,12
0	0	0	0	3,12	37,15
0	0	0	0	3,15	37,13
0	0	0	0	3,13	37,13
0	0	0	0	3,13	37,13
0	0	0	0	3,13	37,15
0	0	0	0	3,15	37,12
0	0	0	0	3,12	37,13
0	0	0	0	3,13	37,12
0	0	0	0	3,12	37,15
0	0	0	0	3,15	37,13
0	0	0	0	3,13	37,13
0	0	0	0	3,13	37,15
0	0	0	0	3,15	37,12
0	0	0	0	3,12	37,13
0	0	0	0	3,13	37,12

0	0	0	0	3,12	37,15
0	0	0	0	3,15	37,13
0	0	0	0	3,13	37,13
0	0	0	0	3,13	37,13
0	0	0	0	3,13	37,15
0	0	0	0	3,15	37,12
0	0	0	0	3,12	37,13
0	0	0	0	3,13	37,12
0	0	0	0	3,12	37,15
0	0	0	0	3,15	37,13
0	0	0	0	3,13	37,13
0	0	0	0	3,13	37,13
0	0	0	0	3,13	37,15
0	0	0	0	3,15	37,12
0	0	0	0	3,12	37,13
0	0	0	0	3,13	37,12
0	0	0	0	3,12	37,15
0	0	0	0	3,15	37,13
0	0	0	0	3,13	37,13
0	0	0	0	3,13	37,15
0	0	0	0	3,15	37,12
0	0	0	0	3,12	37,13
0	0	0	0	3,13	37,12
0	0	0	0	3,12	37,15
0	0	0	0	3,15	37,13
0	0	0	0	3,13	38,18
1	1	1	1	4,18	38,38
1	1	1	1	4,21	38,35
1	1	1	1	4,19	38,38
1	1	1	1	4,21	38,35
1	1	1	1	4,18	38,38
1	1	1	1	4,21	38,35
1	1	1	1	4,19	38,38
1	1	1	1	4,21	0

Al analizar los resultados obtenidos en una de las primeras descargas del data logger se puede notar la existencia de datos errados, lo cual provocó una revisión de la programación del módulo y se identificó que dicho error se suscitaba por el mal direccionamiento a la localidad de memoria I2C ya que no se estaba leyendo el dato inicial en la localidad 000 en cada sub-bloque de 256 bytes.

Una vez detectada la falla se corrigió el software del microcontrolador y se realizó una nueva adquisición y almacenamiento de datos que dió como resultado los valores de la tabla 3.2.



Tabla 3.2: Datos obtenidos después de corregir errores en el data logger.

Archivo de Descarga de Datos					
fecha : 15/06/2010 : 19:58					
DIGITAL_1	DIGITAL_2	DIGITAL_3	DIGITAL_4	Voltaje_1 (V)	Voltaje_2 (V)
0	0	0	0	4,46	4,88
0	0	0	0	4,56	4,93
0	0	0	0	4,48	4,88
0	0	0	0	4,56	4,93
0	0	0	0	4,48	4,87
0	0	0	0	4,57	4,93
0	0	0	0	4,49	4,87
0	0	0	0	4,54	4,92
0	0	0	0	4,46	4,88
0	0	0	0	4,56	4,93
0	0	0	0	4,48	4,88
0	0	0	0	4,56	4,93
0	0	0	0	4,48	4,87
0	0	0	0	4,57	4,93
0	0	0	0	4,49	4,87
0	0	0	0	4,49	4,94
0	0	0	0	4,41	4,89
0	0	0	0	4,5	4,93
0	0	0	0	4,41	4,88
0	0	0	0	4,49	4,94
0	0	0	0	4,41	4,88
0	0	0	0	4,49	4,93
0	0	0	0	4,41	4,87
0	0	0	0	4,49	4,94
0	0	0	0	4,41	4,89
0	0	0	0	4,5	4,93
0	0	0	0	4,41	4,88
0	0	0	0	4,49	4,94
0	0	0	0	4,41	4,88
0	0	0	0	4,49	4,93
0	0	0	0	4,41	4,87
0	0	0	0	4,49	4,94
0	0	0	0	4,41	4,89
0	0	0	0	4,5	4,93
0	0	0	0	4,41	4,88
0	0	0	0	4,49	4,94
0	0	0	0	4,41	4,88
0	0	0	0	4,49	4,93
0	0	0	0	4,41	4,87
0	0	0	0	4,49	4,94
0	0	0	0	4,41	4,89
0	0	0	0	4,5	4,93
0	0	0	0	4,41	4,88
0	0	0	0	4,49	4,94
0	0	0	0	4,41	4,88
0	0	0	0	4,49	4,93
0	0	0	0	4,41	4,87
0	1	0	0	1,88	4,85

0	1	0	0	1,8	4,78
0	1	0	0	1,88	4,85
0	1	0	0	1,8	4,8
0	1	0	0	1,88	4,85
0	1	0	0	1,81	4,8
0	1	0	0	1,89	4,84
0	1	0	0	1,81	4,79
0	1	0	0	1,88	4,85
0	1	0	0	1,8	4,78
0	1	0	0	1,88	4,85
0	1	0	0	1,8	4,8
0	1	0	0	1,88	4,85
0	1	0	0	1,81	4,8
0	1	0	0	1,89	4,84
0	1	0	0	1,81	4,79
0	1	0	0	1,88	4,85
0	1	0	0	1,8	4,78
0	1	0	0	1,88	4,85
0	1	0	0	1,8	4,8
0	1	0	0	1,88	4,85
0	1	0	0	1,81	4,8
0	1	0	0	1,89	4,84
0	1	0	0	1,81	4,79
0	1	0	0	1,88	4,85
0	1	0	0	1,8	4,78
0	1	0	0	1,88	4,85
0	1	0	0	1,8	4,8
0	1	0	0	1,88	4,85
0	1	0	0	1,81	4,8
0	1	0	0	1,89	4,84
0	1	0	0	1,81	4,79
0	1	1	0	2,19	4,79
0	1	1	0	2,3	4,76
0	1	1	0	3,16	4,8
0	1	1	0	3,72	4,74
0	1	1	0	4,6	4,79
0	1	1	0	5,7	4,73
0	1	1	0	5,18	4,79
0	1	1	0	5,13	4,76
0	1	1	0	2,19	4,79
0	1	1	0	2,3	4,76
0	1	1	0	3,16	4,8
0	1	1	0	3,72	4,74
0	1	1	0	4,6	4,79
0	1	1	0	5,7	4,73
0	1	1	0	5,18	4,79
0	1	1	0	5,13	4,76
0	1	1	1	5,23	3,22
0	1	1	1	5,19	3,15
0	1	1	1	5,25	3,22
0	1	1	1	5,26	3,16
0	1	1	1	5,32	3,22
0	1	1	1	5,29	3,16

0	1	1	1	5,34	3,22
0	1	1	1	5,31	3,13
0	1	1	1	5,23	3,22
0	1	1	1	5,19	3,15
0	1	1	1	5,25	3,22
0	1	1	1	5,26	3,16
0	1	1	1	5,32	3,22
0	1	1	1	5,29	3,16
0	1	1	1	5,34	3,22
0	1	1	1	5,31	3,13
0	1	1	1	5,23	3,22
0	1	1	1	5,19	3,15
0	1	1	1	5,25	3,22
0	1	1	1	5,26	3,16
0	1	1	1	5,32	3,22
0	1	1	1	5,29	3,16
0	1	1	1	5,34	3,22
0	1	1	1	5,31	3,13

La tabla anterior demostró que la corrección realizada en el data logger fue exitosa puesto que los datos obtenidos son fieles a los medidos en tiempo real tanto en las entradas análogas como digitales.

### **3.2. PRUEBAS DE COMUNICACIÓN DEL DATA LOGGER CON LA PC**

Inicialmente, el desarrollo del módulo data logger fue pensado en PICC debido a que con dicho programa es posible escribir el código de programación en lenguaje Basic o lenguaje C, pero al avanzar surgieron varios problemas con el funcionamiento autónomo del módulo, especialmente para su arranque, ya que era estrictamente necesario que el módulo estuviera conectado a la computadora antes de iniciar el software de control y configuración para poder establecer la comunicación.

Este problema presentaba una falencia comparativa entre el módulo desarrollado y otros existentes en el mercado ya que no sólo evitaba el arranque sino que

también, si durante el funcionamiento del módulo, se desconectaba de la PC se perdía la comunicación y no se podía realizar una “conexión en caliente”<sup>24</sup>.

En la investigación realizada para corregir este inconveniente, la solución más viable fue utilizar el compilador C18, software de la misma Microchip, con lo que se logró solucionar el problema del funcionamiento autónomo; es decir, en el data logger se pueden configurar todos sus parámetros en PC y puede funcionar sin necesidad de ésta en cualquier entorno, sólo necesita alimentación. Además, sin necesidad de que esté apagado establece comunicación en cualquier instante con la PC para cambiar la configuración y descargar los datos. Las imágenes presentadas en las figuras 3.4 y 3.5 fueron tomadas una vez corregidos los problemas antes mencionados.



Fig. 3.4: Comprobación del funcionamiento del módulo data logger después de migrar el software del microcontrolador.

---

<sup>24</sup> Traducido del inglés *hot-plug*, es la capacidad que tienen algunos periféricos de poder enchufarse o desenchufarse al PC, sin apagar el mismo, y funcionar correctamente.



Fig. 3.5: Comprobación del funcionamiento autónomo del módulo data logger y conexión en caliente.

Solucionado todos esos inconvenientes la comunicación funciona eficientemente.

### **3.3. VENTAJAS Y LIMITACIONES DE LA INTERFACE USB**

El módulo data logger desarrollado, al igual que cualquier otro equipo electrónico, posee pros y contras. Entre las ventajas más notorias se pueden reconocer las mencionadas a continuación:

- La posibilidad de entablar comunicación en cualquier momento.
- Es capaz de trabajar con el voltaje proporcionado por el puerto USB, así también con el de una fuente externa.
- El funcionamiento es autónomo, pues la necesidad de conexión con la PC es sólo necesaria en el momento de descarga de datos.
- Posee una escala variable de tiempo de registro de datos, ya que dispone de su propia memoria de 512 Kb así como de convertidores análogos y 4 digitales, lo cual lo hace muy versátil.

En lo referente a la alimentación se recomienda trabajar preferentemente con fuente externa para evitar cualquier daño o conflicto con el puerto USB ya que consume alrededor de 300mA en plena carga.

Con referencia a las limitaciones la más notoria es el tiempo de descarga de los datos cuando la memoria se encuentra llena en su totalidad, ésto se produce debido a que el archivo mpusbapi.dll de Microchip utilizado para la comunicación sólo permite transferir bloques de datos de 64 bytes, haciendo imposible cualquier intento de corrección.

Además, se debe considerar que en la PC donde vaya a operar el módulo debe tenerse instalado el software LabVIEW versión 8,5 o posterior.

### **3.4. ANÁLISIS DE RESULTADOS**

Solucionados todos los problemas e inconvenientes presentados se procede a realizar otra prueba de análisis y comprobación. Los resultados obtenidos muestran el correcto funcionamiento del data logger en las tres áreas de interés a continuación detalladas:

- Funcionamiento Autónomo: El data logger, en su proceso de toma de datos, puede trabajar sin necesidad de estar conectado a la PC y trabajar con fuente externa sin comprometer los datos en el registrador, siendo opcional conectarse al PC para utilizar el puerto USB como fuente de alimentación en caso de no tener fuente externa.
- Comunicación con el PC: El módulo prueba ser un dispositivo “hot-plug” no importa si pierde la conexión con el PC o se desconecte del mismo, él seguirá en su función de recolección de datos mientras que el software de control permanecerá en estado de “stand-by” o estado de espera pudiéndose reanudar la conexión en cualquier momento sin comprometer la integridad de los datos o el correcto funcionamiento del data logger.

- Conservación de la información: La información grabada en la memoria del dispositivo permanece a salvo sin importar si el dispositivo permanece o no con alimentación por grandes períodos de tiempo o si se suspendió por un instante la toma de datos. La única manera de comprometer la integridad de los datos guardados es, comprometer la integridad total del dispositivo.

El data logger muestra un correcto funcionamiento tanto en hardware como en software lo que permite trasladar el circuito a placa, como se observa en la figura 3.6, sin perder ninguna de las características antes mencionadas.



Fig. 3.6: Circuito del data logger realizado en placa.

### 3.5. ANÁLISIS DE COSTOS

En las tablas 3.3 y 3.4 se detallan el costo de los elementos necesarios para la elaboración del módulo. La existencia de dos tablas se debe a que los elementos mencionados en la tabla 3.3 por su bajo costo fueron comprados en una electrónica común, mientras que los elementos de la tabla 3.4, debido a su mayor precio, fueron comprados en una distribuidora logrando abaratar los precios casi en un 20%.

Tabla 3.3: Primer grupo de elementos comprados.

ELEMENTO	CANTIDAD	VALOR	TOTAL
Zócalo de 8	2	0,05	0,1
Zócalo de 28	1	0,5	0,5
Memoria 24lc512	1	3,4	3,4
Resistencias 1/4 W 3.3 K	2	0,02	0,04
Resistencias 1/4 W 2.2 K	7	0,02	0,14
Resistencias 1/4 W 2.7 K	7	0,02	0,14
Fuente	1	10	10
Baquelita 10*15	1	1,5	1,5
Zenner 5,1V 1/2W	4	0,2	0,8
Bornera azul	4	0,25	1
Leds 3mm	7	0,15	1,05
Capacitor 104	12	0,08	0,96
Electrolítico 470/16	1	0,222	0,22
Caja metálica	1	2,5	2,5
Disipador	1	0,75	0,75
Potenciómetro	1	0,3	0,3
1 Jack 16 LCD	1	0,45	0,45
<b>TOTAL:</b>			<b>23,85</b>

Tabla 3.4: Elementos comprados a distribuidor para abaratar costos.

ELEMENTO	CANTIDAD	VALOR	TOTAL
Papel	1	2,8	2,8
PIC 18f2550	1	15	15
LCD 16*2	1	22	22
Conector USB	1	0,45	0,45
Resistencias	20	0,02	0,4
Relé	1	1,2	1,2
Cristal	1	0,6	0,6
7805	1	1	1
Disipador	1	1	1
Cable USB	1	2,5	2,5
<b>TOTAL:</b>			<b>41,85</b>

Al realizar un análisis de las tablas anteriores se puede observar que los costos no son muy elevados para el desarrollo, ya que el elemento más costoso el PC, lo poseen el elaborador del proyecto y los usuarios del mismo.



En caso de implementarlo para la industria, el sistema sigue siendo de bajo precio, ya que lo más costoso sería el software del data logger puesto que una licencia de software puede encontrarse en el mercado desde los \$200 en adelante, mientras que el software del proyecto al considerarse de código abierto, puede modificarse de acuerdo a las demandas del usuario y a bajo costo.

En el mercado existen dispositivos desde \$75 para medir variables específicas como temperatura o humedad, en la figura 3.7 se muestra un data logger de marca Lascar para medir las variables mencionadas y su valor es de \$87 mientras que en la figura 3.8 se aprecia un data logger marca 3M sólo para registrar la temperatura cuyo costo es de \$99, se debe tomar en cuenta que estos valores son sin los impuestos de ley.



Fig. 3.7: Data logger marca Lascar para medir temperatura y humedad.



Fig. 3.8: Data logger marca 3M para medir temperatura.

Los data logger diseñados para variables generales pueden tener precios que van desde los \$400, en la figura 3.9 se puede ver el data logger digital Lufft OPUS 10 TCE, cuyo valor en tres distribuidoras es diferente siendo estos: \$550, \$585 y \$615 sin contar el valor de impuesto.



Fig. 3.9: Data logger digital Lufft OPUS 10 TCE de propósito general.

Comparando los valores de los módulos data logger encontrados en el mercado, con los aproximadamente \$70 invertidos en el diseño de este dispositivo, se evidencia que es una alternativa de bajo costo y gran desempeño.

# CAPÍTULO IV

## CONCLUSIONES Y RECOMENDACIONES

Al término del desarrollo del presente proyecto de diseño e implementación de un módulo data logger, se ponen a consideración las conclusiones y recomendaciones alcanzadas en todo el trayecto de realización del mismo.

### 4.1. CONCLUSIONES

- Se diseñó y elaboró un dispositivo data logger no convencional, basado en tecnología reciente, con interfaz USB 2.0 usando el PIC 18F2550, totalmente funcional y que puede ser usado en el campo industrial y también en el educativo.
- La principal ventaja del data logger diseñado radica en que es un “dispositivo de código abierto”, puesto que en este documento se incluye toda la programación del mismo, lo que permitirá que cualquier persona realice modificaciones y mejoras, si las considera necesarias.
- La interfaz USB es de propósito general, logrando así facilitar el uso del data logger casi en cualquier PC, dicho de otra manera, el hardware no es una limitante para su funcionamiento.
- El microcontrolador PIC18F2550, elegido para desarrollar este proyecto, cumplió eficientemente su papel, ya que una de sus virtudes es que otorga respuesta en tiempo real en las aplicaciones en las que se lo usa.
- El dispositivo es autónomo, pues no siempre es necesaria la conexión con el PC para poder acceder a la información que se guarda en el mismo.

- Se utilizaron herramientas propias de Microchip las cuales tienen como función principal facilitar la comunicación entre la PC y el módulo USB del PIC 18F2550.
- Gracias a la integración de software y hardware, se considera al dispositivo conveniente para fines didácticos.
- El éxito de la interfaz USB ha sido tal que, actualmente todos los PC's tienen integrados al menos dos puertos USB para la conexión de dispositivos y prácticamente ha reemplazado a los puertos paralelos.
- La desventaja principal de la interfaz USB es que el ancho de banda debe repartirse entre todos los dispositivos conectados a dicho puerto.
- El rango de lectura de la trama, se la puede manipular con software para el mejor empleo o uso en las aplicaciones que así lo requieran.
- El software desarrollado, permite el control y cierto nivel de configuración del dispositivo por parte de los usuarios.
- El software que se utilizó para el desarrollo del proyecto de investigación fue PROTEUS v7.4, MPLAB v8.15a, LabVIEW v8.5, compilador MPLAB C18 v3.22, los cuales han sido de gran ayuda para el desarrollo de la aplicación.
- El proyecto puede ser el punto de inicio para otros de este mismo tipo que quisieran implementarse, ya que propone una solución alternativa en base al avance tecnológico y electrónico presentes en el ámbito industrial, en relación al manejo y adquisición de datos.
- La realización de los proyectos de grado es fundamental en el desarrollo profesional de los estudiantes, porque permite modernizar y actualizar los

conocimientos de muchos temas permitiendo, mejorar el nivel de preparación los egresados de la ESPEL.

## **4.2. RECOMENDACIONES**

- Entender el funcionamiento y manejo de los archivos .dll puesto que a pesar de ser muy útiles, también suelen ser causa de errores en Windows.
- Instalar todas las librerías y funciones en los programas utilizados para el desarrollo del módulo data logger, puesto que una instalación incompleta puede causar el incorrecto desempeño del mismo.
- Manipular con cuidado el cableado que va hacia la placa principal, ya que con movimientos bruscos estos se pueden romper con facilidad.
- El correcto funcionamiento del dispositivo se debe comprobar al armarlo en Proto Board ya que facilita la detección y corrección de errores que podría tener el circuito y también depurar los errores en el software.
- Se recomienda aprovechar la información de las hojas técnicas de los elementos a utilizar; además, de los manuales y tutoriales provistos por los desarrolladores de los programas utilizados, ya que en ellos se suelen encontrar las respuestas a los problemas técnicos que se presentan.
- Al realizar cambios en el software, ya sea en el programa del data logger como en el programa de control, se recomienda realizar respaldos documentados adecuadamente.
- Para un funcionamiento óptimo del dispositivo es recomendable que las entradas analógicas se encuentren en el rango de 0v a 10v mientras que las entradas digitales de 0v a 5v, que se entienden como 0 y 1 lógico respectivamente.

- Colocar el dispositivo data logger en un lugar libre de humedad, ya que contiene elementos electrónicos y estos a la larga pueden ser averiados.
- La pantalla LCD es una guía de los datos que adquiere el dispositivo.
- Para este tipo de aplicaciones el uso de microcontroladores de la Microchip es adecuado, ya que de ellos existe bastante información.
- Se recomienda el uso de herramientas de investigación que se encuentran en internet, foros, libros digitales y datasheets, ya que en ellos se puede encontrar información importante y confiable, de forma rápida y económica.

### **4.3. BIBLIOGRAFÍA Y ENLACES**

- The User Manual of Calling DLLs: Calling DLL in LabVIEW; National Instruments.
- Memoria 24LC512 con Interface I2C, Javier Antolí González, F. Manuel Díaz-Madroñero Boluda, Javier Moscardó Salido; Escuela Politécnica Superior de Alcoy
- [http://www.atc.ugr.es/~afdiaz/fich/bus\\_i2c.pdf](http://www.atc.ugr.es/~afdiaz/fich/bus_i2c.pdf)
- <http://www.epc.org.mx/view.php?id=1>
- <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>
- <http://www.ciciliani.com.ar/USB/introduccion.html>
- MICROCHIP MPLAB C18 C COMPILER, GUÍA DEL ESTUDIANTE; Laboratorio de Sistemas Basados en Microprocesador y Sistemas Electrónicos Digitales.
- GROOVER, Mikell. Robótica Industrial, Tecnología, programación y aplicaciones, México: McGRAW.HILL, 1994, p. 72 – 75.

- CAPT. DE MG. Capelo Badillo Aldo Grivaldy, Cruz Guanoluisa José Luis; Diseño y elaboración de un sistema de seguridad vehicular con tecnología rfid, con interfaz usb 2.0 y controles de acceso codificados.
- BARREIROS, Antonio. Fundamentos de Robótica, Madrid: Concepción Fernández Madrid, 1997, p. 26 -29, 37.
- [www.emmicroelectronic.com](http://www.emmicroelectronic.com)
- Angulo, José, Microcontroladores PIC: 88 - 123, México: Edigrafos.
- REYES, Carlos, “Aprenda a Programar Microcontroladores PIC”: 34 - 69, Ecuador.
- <http://es.wikipedia.org/wiki/USB>
- <http://www.ti.com/tiris/default.htm>
- BARRIETOS, A.; PEÑIN, L.; BALAGUER, C.; ARACIL, R. España: McGRAWHILL, 2007, p. 151 – 156
- Mackay, Park, Wright: Data Communication for instrumentation and Control. Techbooks. Sydney Australia. 2000.
- Horak Ray: Communication Systems and networks: MT Books, New York, 1999.
- Hallsall Fred: Data Communications, Computer Networks and Open Systems.
- <http://www.Mikroelectronica.com.es/>
- <http://www.semiconductors.philips.com/products/identification/index.html>
- PROTEUS VSM. Ingeniería Eléctrica Electrónica, S.A.
- [www.ieeproteus.com](http://www.ieeproteus.com)
- [http://www.olimex.cl/present.php?page=tut\\_pic\\_web\\_c18&c=1](http://www.olimex.cl/present.php?page=tut_pic_web_c18&c=1)
- Conversor analógico/Digital del PIC18F2550: Teoría de funcionamiento.; Moyano Jonathan Ezequiel.
- Librería del USB: mpushapi.dll; Microchip Electronics.

**ANEXO A**  
**FRAGMENTO DEL CÓDIGO DE PROGRAMACIÓN**



```

struct
{
    unsigned char    ACTUALIZAR_LCD:1;
    unsigned char    _ES_DIGITAL:1;
    unsigned char    _LEER_I2C:1;

}BANDERAS;
struct
{
    unsigned char    _USB_TX:1;
    unsigned char    _USB_RX:1;
    unsigned char    _DESCARGAR_DATOS:1;
    unsigned char    _PASAR_DATOS:1;
}BNADERAS_USB;

void main(void)
{
    OpenTimer1( TIMER_INT_ON &
                T1_16BIT_RW &
                T1_SOURCE_INT &
                T1_PS_1_8 &
                T1_OSC1EN_OFF &
                T1_SYNC_EXT_OFF );

    OpenADC      (ADC_FOSC_64 &
                  ADC_RIGHT_JUST    &
                  ADC_16_TAD    ,    &
                  ADC_CH0        &
                  ADC_INT_ON      &
                  ADC_VREFPLUS_VDD&
                  ADC_VREFMINUS_VSS,
                  0x0D);

    WriteTimer1(3035);
    TRISA=255;
    TRISC=0b00000000;
    LED_FUNCIONAMIENTO=1;
    TRISB=0b00000011;
    OpenI2C(MASTER, SLEW_OFF);
    SSPCON1=0X028;
    SSPADD=99;

    LCD_INI();
    LCD_INST(0x80);
    LCD_RomStr("*** ESPE--USB**");
    LCD_INST(0xc0);
    LCD_RomStr(" JORGE RIVERA ");
    //RETARDO DE ARRANQUE
    Delay10KTCYx(255);
    Delay10KTCYx(255);
    Delay10KTCYx(255);
    Delay10KTCYx(255);
}

```

```

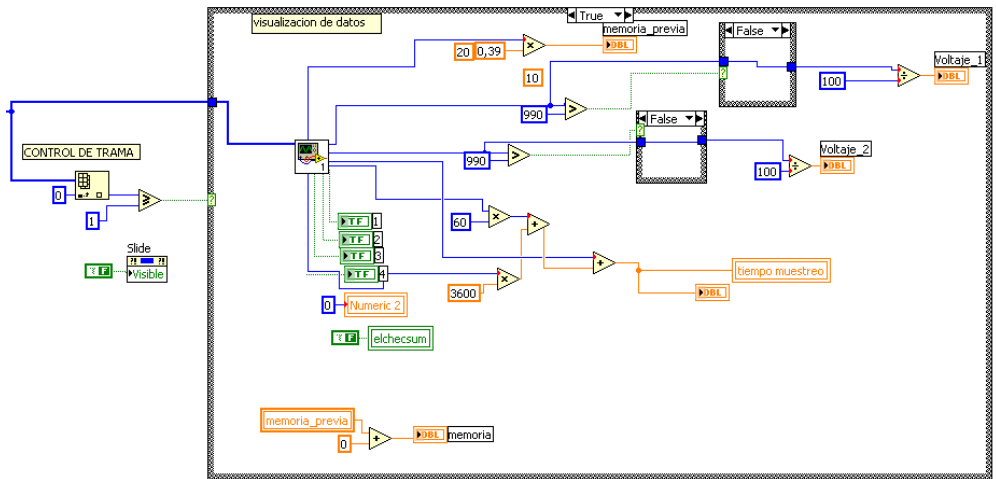
Delay10KTCYx(255);
Delay10KTCYx(255);
Delay10KTCYx(255);
Delay10KTCYx(255);
Delay10KTCYx(255);
Delay10KTCYx(255);
LED_FUNCIONAMIENTO=0;
LCD_INST(0xc0);
LCD_RomStr("      ");
LCD_INST(0x80);
LIMPIAR_LCD();
LCD_INST(0xc0);
LIMPIAR_LCD();
CMCON=0x7;
ADCON1      =0b00001101;
ADCON0bits.CHS0=0;
InitializeSystem();
INTCONbits.GIE=1;
INTCONbits.PEIE=1;
//MotorCount++;

while(1)
{
    USBTasks();
    if( BNADERAS_USB._USB_TX)
    {
        INTCONbits.GIE=0;
        INTCONbits.PEIE=0;
        USBTasks();
        USB_SALIDA();
        LED_FUNCIONAMIENTO=!LED_FUNCIONAMIENTO;
        BNADERAS_USB._USB_TX=0;
        INTCONbits.GIE=1
        INTCONbits.PEIE=1;
    }
}

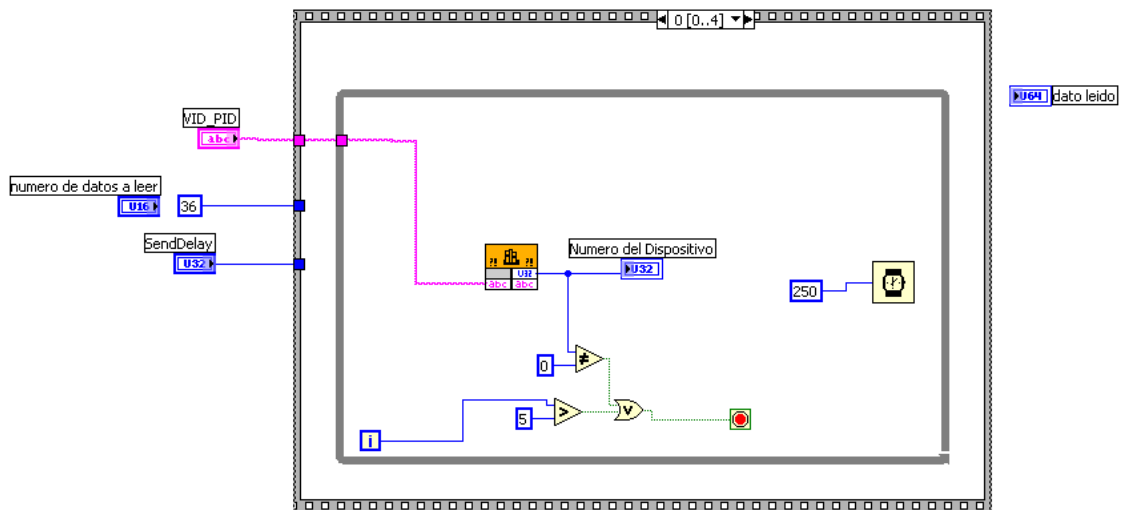
```

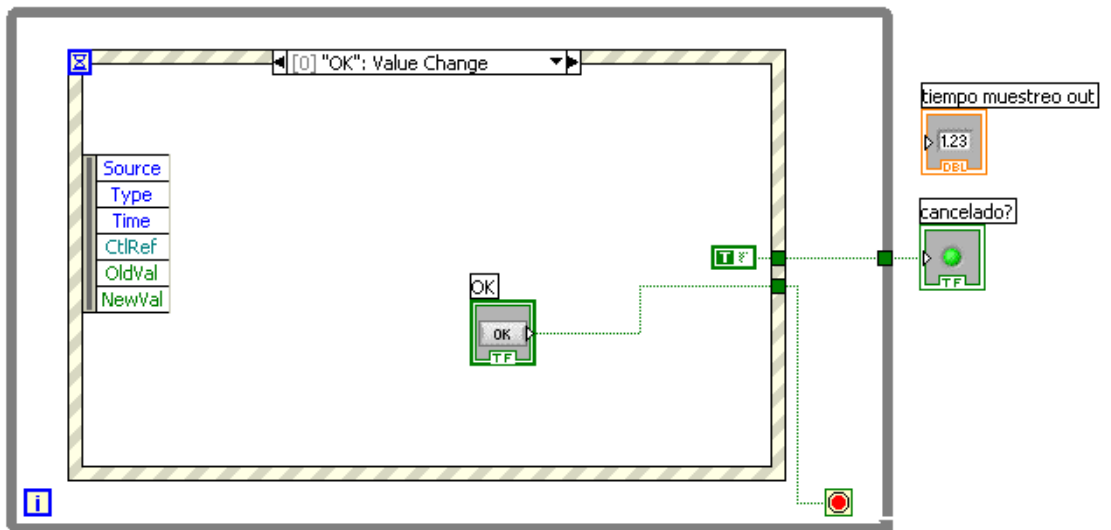
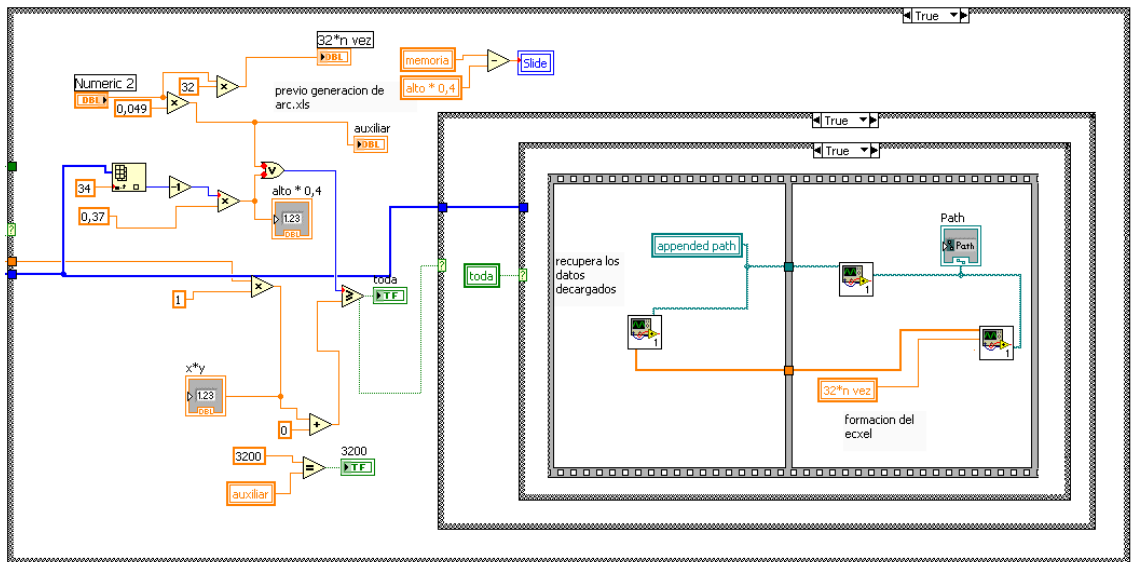
## **ANEXO B**

### **CAPTURAS DE LA PROGRAMACIÓN EN LABVIEW**



Jorge\_Rivera ESPE USB-DATALOGGR





**ANEXO C**  
**HOJAS TÉCNICAS DEL MICROCONTROLADOR**  
**PIC18F2550**



# **PIC18F2455/2550/4455/4550**

## **Data Sheet**

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

---

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

---

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

QUALITY MANAGEMENT SYSTEM  
CERTIFIED BY DNV  
**== ISO/TS 16949:2002 ==**



**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  $\mu$ A typical
- Sleep mode currents down to 0.1  $\mu$ A typical
- Timer1 Oscillator: 1.1  $\mu$ A typical, 32 kHz, 2V
- Watchdog Timer: 2.1  $\mu$ 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 (Tcy/16)
  - Compare is 16-bit, max. resolution 83.3 ns (Tcy)
  - 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 I<sup>2</sup>C™ 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:**

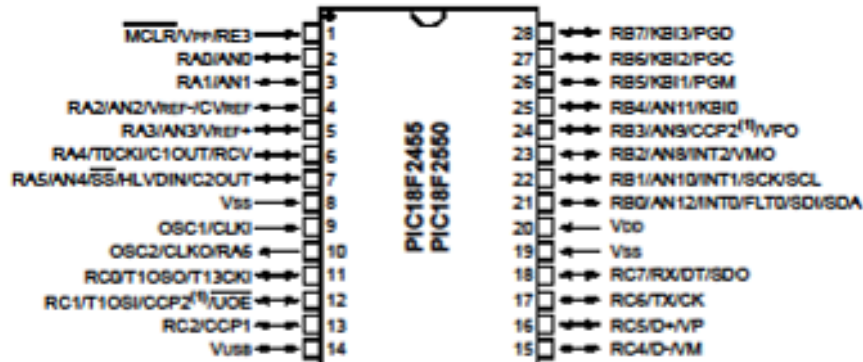
- C Compiler Optimized Architecture with optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-Programmable under Software Control
- Priority Levels 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		IO	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 <sup>2</sup> 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

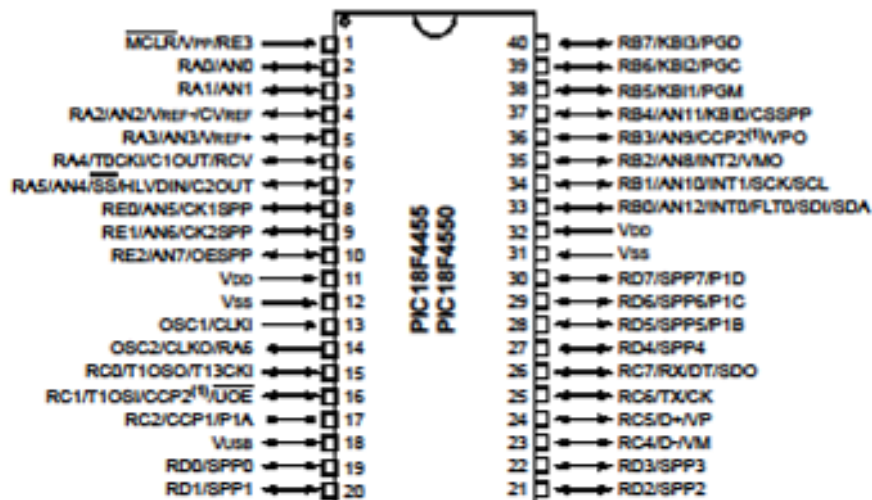
# PIC18F2455/2550/4455/4550

## Pin Diagrams

### 28-Pin PDIP, SOIC



### 40-Pin PDIP



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

# AUTORÍA

Latacunga, Septiembre 2010

Elaborado por:

---

Jorge Luis Rivera Fárez  
C.I. 180300252-4

EL DIRECTOR DE CARRERA DE INGENIERÍA ELECTRÓNICA E  
INSTRUMENTACIÓN

---

Ing. Armando Álvarez Salazar

EL SECRETARIO ACADÉMICO DE LA ESPE-L

---

Dr. Eduardo Vásquez Alcázar