

**Universidad de Pinar del Río**  
**Facultad de Informática y Telecomunicaciones**  
**Departamento de Telecomunicaciones**



**Tesis presentada en opción al Título Académico de Máster  
en Sistemas de Telecomunicaciones**

**Título:**

Diseño y evaluación de soluciones técnicas para la comunicación USB 2.0 entre la PC y una interfaz electrónica con el microcontrolador PIC18F4550.

**Autor:** Ing. José Rafael Corrales de la Cruz.

**Tutor:** MSc. Rolando Rodríguez Henríquez  
Profesor Auxiliar UPR

**2009**

**"Año del 50 Aniversario del Triunfo de la Revolución"**

“Nunca consideres al estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber”

**Albert Einstein**

# **AGRADECIMIENTOS**

Agradezco:

A los que me animaron en los momentos de debilidad y flaqueza por los que todos los seres humanos transitamos en el camino de la vida.

A los que nos aconsejaron, y nos brindaron su solidaridad sin la cual nos hubiera sido imposible concluir.

A todas las personas que tuvieron confianza en que podía y debía seguir adelante, y me alentaron dulce y cariñosamente para conjugar ideas, elaborar criterios, perfeccionar conocimientos y procurar que todo cuanto se ha logrado con paciencia y dedicación, cumpla el objetivo que se proyectó y por el que se ha trabajado denodadamente.

A todos mis familiares, amigos y compañeros, pues interminable sería la lista de nombres que se esmeraron en ayudarme y alentarme para cumplir con los objetivos y metas de este proyecto.

Muy especialmente al tutor Msc. Rolando Rodríguez Henríquez, quien de manera profesional, me ha ayudado en todo momento y con sus esmerados e incansables dotes de educador, sin su apoyo y diligente conducción no hubiera sido posible la organización de ideas, la consecución de los objetivos ni la terminación de esta obra.

A mis compañeros de la UCT LACETEL cuya colaboración necesariamente tengo que apreciar y sin la cual no habría logrado de manera práctica la conclusión de esta maestría. Muy especialmente al Ing. Pablo González, quien de manera incondicional me ha ayudado, su eterno agradecimiento por dedicarme su precioso tiempo y abnegado legado de sus conocimientos y su labor.

A todos Muchas Gracias

# **DEDICATORIA**

Dedico esta tesis:

- A la sabiduría de la naturaleza.
- A la eterna búsqueda de la verdad.
- A todos que apremian el amor y la ternura.
- A toda obra que buena que por demás es bella.
- A mi familia en especial, que han recorrido junto a mí el camino hasta hoy y además porque cada éxito obtenido lo sienten como suyos.
- Especialmente a mi hija.

José Rafael

## **OPINIÓN DEL TUTOR**

El ingeniero José Rafael Corrales de la Cruz, trabajador de la UCT LACETEL, ha concluido con resultados satisfactorios el Trabajo de Tesis por el grado científico de MSc en Sistemas de Telecomunicaciones, en el cual ha puesto de manifiesto sus conocimientos técnicos, su laboriosidad y consagración a la investigación científico técnica.

El desarrollo de este trabajo se caracterizó por:

- Un elevado rigor científico en las soluciones de hardware y software obtenidas durante el desarrollo de la aplicación.
- Una detallada recopilación bibliográfica sobre los microcontroladores PIC de la familia 18Fxxx.
- Capacidad de integrar conocimientos.
- La óptima aplicación de las herramientas para el diseño con microcontroladores PIC

En el desarrollo de esta investigación, el maestrante ha manifestado independencia, creatividad y capacidad para aplicar los conocimientos adquiridos durante los estudios de post-grado.

Por las razones antes expuestas, por la calidad técnica del trabajo y el cumplimiento de los objetivos propuestos, consideramos que el Maestrante es merecedor de la calificación de \_\_\_\_\_ puntos.

Dado en la ciudad de Pinar del Río, a los \_\_\_\_ días del mes de \_\_\_\_\_ de 2009.

---

Firma del Tutor



## **DECLARACIÓN DE AUTORÍA**

Declaro que soy autor del presente Trabajo de Tesis por el grado científico de MSc y otorgo el derecho al centro de trabajo donde me desempeñó: la Unidad de Ciencia y Técnica LACETEL, a hacer uso del mismo con la finalidad que estime conveniente, y a la Universidad de Pinar del Río “Hermanos Saiz Montes de Oca” a utilizar este resultado solo con fines docentes.

---

José Rafael Corrales de la Cruz

## INDICE

Índice	Paginas
Resumen	
Summary	
Introducción	1
CAPÍTULO 1 Antecedentes y Estado Actual de la Temática	7
1.0 Antecedentes y Estado Actual	7
1.1 Fundamentación del Proyecto.	9
1.2 Herramientas de diseño, metodologías, y métodos de investigación a emplear en el Proyecto. Medios	10
1.3 Aportes novedosos.	11
1.4 Impactos: Económicos, Sociales y Tecnológicos.	12
1.5 Resultados esperados.	12
CAPÍTULO 2 EL ESTÁNDAR USB Y SU APLICACIÓN EN LA COMUNICACIÓN DIGITAL ENTRE LOS SISTEMAS DE ADQUISICIÓN DE DATOS Y LA PC	14
2.0 El estándar USB	14
2.1 La comunicación por puerto USB. Tipos y Especificaciones	15
2.2 Especificaciones USB	15
2.3 Cables y conectores para puerto USB	16
2.4 Velocidad de transferencia del puerto USB	16
2.5 Funcionamiento del Protocolo de comunicación USB.	17
2.6 La especificación USB 2.0	
2.7 Evolución y Características técnicas de los microcontroladores Pic dotados de recursos para realizar la comunicación USB 2.0.	18
2.7.1 Los PICs de Microchip.	18
2.7.2 La familia PIC18Fxx50 y PIC18Fxx55	20
2.7.3 Modos de Trabajo	21
2.7.4 Conexiones del Oscilador	22
2.7.5 Otras características de los Microcontroladores	22
2.7.6 Operación del módulo de USB	28
2.7.7 RAM USB	32
2.7.8 Buffer Descriptor y el Buffer Descriptor de Tabla	33
2.7.9 Conceptos generales de la estructura de la comunicación.	35
CAPÍTULO 3 OBTENCIÓN DE LA COMUNICACIÓN POR PUERTO USB ENTRE UNA APLICACIÓN CON MICROCONTROLADORES PIC Y UNA PC SEGÚN LA ESPECIFICACIÓN USB 2.0	
3.0 Soluciones de Hardware para la realización de la comunicación con la PC por puerto USB.	39
3.1 Requerimientos técnicos para el desarrollo del Software para la comunicación con la PC por puerto USB.	39
3.2 Una aplicación demostrativa para la comunicación USB 2.0, sobre el Kit de desarrollo PICDEM FS.	40
3.2.1 Programa Demostrativo residente en la PC, para comunicar por puerto USB con la interfaz electrónica basada en la aplicación del PIC18F4550, según la especificación USB 2.0	41
3.2.2 Descripción del hardware del Kit de Desarrollo PICDEM FS USB.	44
3.2.3 Estructura del microprograma del PIC18F4550 para obtener la	46

	aplicación demostrativa de la comunicación USB 2.0, con el Kit de desarrollo PICDEM FS.	
3.2.4	Descripción de la aplicación desarrollada para el sistema.	57
3.2.5	Descripción de la aplicación desarrollada en Delphi.	65
3.2.6	Descripción del funcionamiento del programa demostrativo.	66
	Conclusiones	69
	Recomendaciones	70
	Bibliografía	71
	Referencias Bibliográficas	73

## **RESUMEN**

En el presente Trabajo de Diploma se diseña, modela a escala de maqueta y evalúa en la práctica soluciones de hardware y de software para el diseño de aplicaciones empleando microcontroladores del tipo PIC y además para de forma general, integrar los resultados obtenidos al diseño de sistemas automatizados de control de parámetros tecnológicos y desarrollar equipos electrónicos que requieran la comunicación por puerto USB con una PC cumpliendo con las especificaciones USB 2.0. En el desarrollo de la investigación se emplea como herramienta de diseño el software MPLAB 7.5, el depurador programador ICD2, el compilador CCSC para aplicaciones con microcontroladores PIC en lenguaje C y un software demostrativo de la comunicación que está residente en la PC. Las aplicaciones construidas y evaluadas se basan en la especificación USB 2.0 se emplea el kit de desarrollo PICDEM FS. Se obtienen resultados satisfactorios en la evaluación de las soluciones de hardware y software para la comunicación por puerto USB 2.0 con el microcontrolador PIC18F4550.

## **SUMMARY**

In this term paper we design, scale model and implement different hardware and software solutions for the design of applications using PIC microcontrollers, and resides, to integrate the obtained results to the design of parameter-controlling automated systems, and also to develop electronic devices that require communications by USB port with a PC, for USB 2.0 specifications. In the development of this research, we use, as a design tool, the MPLAB 7.5 software, ICD 2 debugger programmer, CCSC compilers for applications with PIC microcontrollers in C language for the development and implementation of communication programs residing in the PC with C language. The built and evaluated applications are based for the USB 2.0 specification; we use PICDEM FS development kit, based in the PIC18F4550 microcontroller. Satisfactory results have been obtained in the evaluation of hardware and software solutions for communication by USB port for both kinds of specifications.

## Introducción.

La comunicación por puerto USB (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 los ordenadores y que ha sido introducida en el mercado por las computadoras personales y periféricos para mejorar las lentas interfases serie RS-232 y paralelo. La comunicación por puerto USB provee una mayor velocidad de transferencia (de hasta 100 veces más rápido) comparado con la comunicación por puerto Paralelo y con la RS-232, el objetivo inicial era conectar periféricos relativamente lentos (mouse, impresoras y cámaras digitales) de una forma rápida y eficiente, basada en comunicaciones serie, posteriormente su uso se extendió a la conexión de interfaces más complejas como son discos duros, tarjetas de adquisición de datos, equipos de medición en tiempo real, memorias flash, scanner y otros. La interfaz USB posee 4 líneas de conexión, donde se encuentran las líneas de 5 Voltios y tierra para la alimentación, puede transmitir datos con diferentes velocidades, lo cual ha sido estandarizado por las especificaciones internacionales USB 1.0 y USB 2.0. La primera computadora que incluyó un puerto **USB** de forma estándar fue el iMac de Apple, en 1998, para conectar el teclado y el Mouse, y Microsoft lo introdujo en los controladores correspondientes en la versión OSR 2.1 de Windows 95C cuando los sistemas de MS incorporan de forma estándar soporte para este bus y en el ámbito de servidores se incorpora en Windows 2000. Actualmente la comunicación USB se ha convertido en el método universal de conexión de periféricos [2].

Los microcontroladores de la familia PIC son componentes electrónicos, cuyas características técnicas y funcionales, brindan la posibilidad de soportar tecnológicamente las soluciones técnicas de hardware y software que de forma personalizada se requieren diseñar, fabricar y evaluar como resultados científico técnicos a introducir en la práctica social sostenible, lo cual brinda actualidad científico-técnica, fiabilidad, novedad y elevado rigor científico a los resultados de la investigación.

Además en el contexto actual, existen entidades en el ámbito nacional que solicitan el desarrollo y la introducción rápida, oportuna y sostenible de estas nuevas tecnologías en el campo de la defensa del país y el control de procesos tecnológicos, con el objetivo de satisfacer los requerimientos de calidad pedidos e imponer a estas soluciones, características técnicas distintivas y personalizadas que aseguren la soberanía tecnológica.

Teniendo en cuenta la naturaleza general del equipo de comunicación, existen dos especificaciones que definen la arquitectura que es capaz de soportar cualquier dispositivo de comunicación.

El desarrollo de los sistemas de adquisición, procesamiento y control de parámetros tecnológicos se obtiene diseñando y construyendo una aplicación de hardware y software basada en el empleo de microcontroladores, principalmente del tipo PIC. Un microcontrolador, es un componente electrónico encapsulado en un chip, capaz de ejecutar un microprograma, para lo cual cuenta con un conjunto de recursos tales como: microprocesador, memoria de programa, memoria de datos y puertos de entrada/salida. Además, también dispone de otros elementos especiales como son: puertos serie, comparadores, convertidores analógico-digitales y otros. Además poseen un conjunto de instrucciones las cuales se ejecutan según el orden programado. Mediante estas instrucciones básicas podemos realizar operaciones más complejas y así llegar al objetivo de la aplicación. Microchip Technology, es decir los PICs. Este tipo de microprocesadores están muy extendidos actualmente en el mercado gracias a su gran variedad y bajo coste. Otra razón del éxito de los PICs es su utilización, ya que una vez se ha aprendido a utilizar uno, conociendo su arquitectura y juego de instrucciones, es muy fácil emplear otro modelo diferente.

La necesidad de investigar en el desarrollo de aplicaciones con microcontroladores PIC para desarrollar procesamiento de parámetros en procesos tecnológicos (SAPC), que tengan la comunicación con la PC por puerto USB, por lo cual surge el siguiente problema a resolver.

### **Problema a resolver.**

En el diseño personalizado de sistemas de adquisición, procesamiento y control de parámetros tecnológicos. **¿Como contribuir a obtener y evaluar soluciones técnicas de hardware y software para la comunicación por puerto USB entre la interfaz electrónica del sistema y una computadora personal (PC)?**, que permitan garantizar la transmisión y la recepción de paquetes digitales de datos, soportado tecnológicamente por la aplicación de los microcontroladores de la familia PIC y una PC, donde reside un Software propio de monitoreo y control de procesos tecnológicos.

### **Objeto de la investigación:**

La familia de los microcontroladores PIC y sus aplicaciones en las comunicaciones por puerto USB en sistemas automatizados de adquisición y procesamiento de datos para el control de procesos tecnológicos empleando una PC.

### **Objetivos.**

#### **Objetivo General:**

Establecer soluciones de diseño propias, soportadas tecnológicamente por la aplicación de los microcontroladores de la familia PIC, para obtener la comunicación de paquetes de datos digitales entre el puerto USB de la PC y la tarjeta electrónica principal que constituye el sistema automatizado de adquisición, procesamiento y control de parámetros tecnológicos, dando cumplimiento a la especificación internacional USB 2.0.

#### **Objetivos Específicos:**

1. Diagnosticar la evolución y el estado actual de los aportes científico técnicos y de los procedimientos tecnológicos empleados para implementar la comunicación por puerto USB mediante las especificación internacional USB 2.0 y su aplicación al diseño de sistemas automatizados para la adquisición y procesamiento de parámetros en procesos tecnológicos (SAPC), empleando como soporte técnico los microcontroladores PIC y una PC.
2. Diseñar y evaluar a nivel de maqueta, las soluciones técnicas creadas (mediante tarjetas electrónicas y microprogramas PIC) para obtener la comunicación entre la interfaz electrónica del sistema y una PC por puerto USB según las especificaciones USB 2.0 [11] con un protocolo propio, que sea factible de integrar al diseño personalizado de sistemas automatizados de adquisición y procesamiento de parámetros tecnológicos empleando los microcontroladores PIC.
3. Ejecutar un Programa demostrativo residente en la PC, que permita mostrar que se establece la comunicación vía puerto USB entre la interfaz electrónica del sistema soportada con microcontroladores PIC y una PC, empleando los protocolos de comunicación obtenidos.
4. Establecer los requerimientos tecnológicos para obtener la fabricación a mayor escala de las soluciones técnicas de hardware y software diseñadas,

con el objetivo de aplicarlas en la generalización de las salidas planificadas de otros Proyectos de Desarrollo y de Innovación Tecnológica en Programas de alcance territorial, ramal y nacional.

### **Campo de acción.**

El diseño y la construcción de aplicaciones con microcontroladores PIC para obtener protocolos propios de comunicación por puerto USB en los sistemas automatizados de adquisición y procesamiento de parámetros tecnológicos y el desarrollo de programas personalizados para estos sistemas en la PC.

### **Hipótesis**

Si se logra desarrollar y evaluar, un conjunto de soluciones técnicas para el diseño del hardware y el software, entonces se contribuirá a establecer la comunicación por puerto USB entre una micro computadora personal (PC) y, las aplicaciones construidas con los microcontroladores PIC, e integrar estas soluciones en el diseño de sistemas de adquisición y procesamiento de parámetros tecnológicos.

### **Marco contextual.**

El estado de la ciencia y la tecnología a nivel mundial, muestra en este tema, la existencia de sistemas automatizados que emplean sensores para el control de los parámetros tecnológicos, logrando medir y transmitir en tiempo real a una PC, la lectura de múltiples datos medidos en diferentes puntos, en un ambiente de red, como por ejemplo son: los sistemas empleados en redes domóticas, en redes inmóticas para edificios inteligentes, en las redes de control de procesos tecnológicos industriales, entre otras aplicaciones construidas y mantenidas en la práctica social.

Estos sistemas incluyen la comunicación inalámbrica o cableada desde cada uno de los nodos sensores inteligentes con un nodo central o coordinador de la red, el cual a su vez, se comunica con la PC mediante el puerto USB, y en el ordenador reside un software supervisor que monitorea y controla el sistema. Para normar esta comunicación existen los estándares USB1.0/1.1 y USB2.0 [12], para implementar estos protocolos se han desarrollado componentes electrónicos específicos para ser empleados en el soporte tecnológico de esta comunicación, los cuales permiten reconfigurar estos en correspondencia con los requerimientos propios del sistema y su diseño personalizado.

## **Marco Teórico.**

Para el control de los procesos tecnológicos en diferentes ramas de la industria y los servicios se requieren diseñar sistemas de adquisición, procesamiento y control (SAPC) que permitan las siguientes funciones:

- Medir la variable de control en el rango y la exactitud pedida.
- Indicar digitalmente la lectura de la medición
- Procesar las lecturas de las mediciones.
- Programar los valores de los parámetros de control.
- Almacenar en una memoria externa los resultados.
- Controlar el proceso tecnológico mediante los actuadores.
- La Comunicación con una computadora mediante puerto serie o puerto USB, empleando un protocolo estandarizado o propio.

La interfaz de comunicación se selecciona considerando aspectos técnicos como son: la velocidad de la transmisión, la distancia entre los nodos del sistema y la computadora, las condiciones de trabajo del sistema, entre otras. Algunas de las vías de comunicación alámbrica que se emplean son: RS232, RS485, USB 1.0, USB 2.0, Ethernet, entre otras.

Desde finales de la década de los 90 y hasta la actualidad la firma Microchip ha puesto en el mercado las series de dispositivos PIC denominadas PIC16Cxx, PIC16Fxx, PIC18Fxxx, y otros del tipo PICRF y PICDSP para la comunicación inalámbrica y el procesamiento digital de señales.

La familia PIC incluye dispositivos en diferentes gamas considerando la capacidad de memoria EEPROM, la capacidad de memoria de datos, las líneas de puerto, el número de temporizadores, el número de interrupciones externas y la posibilidad de obtener mediante estos la programación y el control de periféricos (teclados, indicadores del tipo LCD, convertidores análogos-digital, entre otros) y de interfaces de comunicación como son: bus I2C, puerto USB, protocolo puerto serie RS232, y otros [8] [9].

Inicialmente la obtención de la comunicación USB para los sistemas de adquisición, procesamiento y control de parámetros tecnológicos empleando microcontroladores del tipo PIC evolucionó de la siguiente forma:

1. Debido a que la interfaz RS232 es factible de obtener directamente de las líneas de puerto de los microcontroladores PIC de la serie 16Cxx en adelante y no existía esta posibilidad tecnológica para la comunicación USB. Primero

se desarrolló el diseño de soluciones de Hardware y Software para a partir de la interfaz RS232 que brinda el microcontrolador PIC y mediante una tarjeta electrónica externa obtener la comunicación USB del tipo 1.0 con la PC, principalmente con el circuito integrado FT232BM, luego en esta primera etapa las soluciones técnicas buscaban trasladar los programas desarrollados para RS232 al protocolo USB 1.0. Esta opción aún se emplea, pero está limitada por la velocidad de transmisión y por ser menos compacta y fiable que las posteriores soluciones desarrolladas.

2. Con el avance experimentado en el diseño de microcontroladores PIC, la firma Microchip lanzó al mercado en el año 2002 el microcontrolador PIC16C745/765 el cual tiene 28/25 pines de conexión y por primera vez ofrece desde 2 de estas líneas de puerto la posibilidad de comunicación con una PC mediante la interfaz USB 1.1 con una velocidad de hasta 12 Mbps Además este microprocesador posee un conversor análogo-digital con una resolución de 8 bits y hasta 8 canales de medición multiplexados, además posee una capacidad de 8 Kbytes de memoria de programa y 256 bytes de memoria de datos. Así como otras características técnicas que permiten diseñar con éxito sistemas de adquisición, y procesamiento de datos.
3. A partir del 2005 aparecen en el mercado Microcontroladores PIC que permiten la comunicación por puerto USB 2.0, como son la familia 18F2455/2550/4555/4550. Estos dispositivos cuentan con el hardware necesario para realizar una conexión USB a LS o FS, las operaciones de protocolo deben ser programadas y consumen importantes recursos del dispositivo, y que tiene la ventaja de ofrecer calidad, flexibilidad e invaluable respaldo y soporte técnico para el diseño de sistemas de adquisición, y procesamiento de datos [1], [2].

# **CAPÍTULO 1**

## **Antecedentes y Estado Actual de la Temática**

### **1.0 Antecedentes y Estado Actual**

Se tienen como antecedentes cognoscitivos sobre el tema de investigación, la dirección y ejecución de otros Proyectos de I+D e Innovación Tecnológica en la UCT a nivel Territorial, Ramal y Nacional, donde ha participado el maestrante en calidad de coautor de los resultados obtenidos, algunos de estos logros se enumeran a continuación:

- Prototipo del sistema automatizado con comunicación con PC por puerto USB según la especificación USB 1.0 basado en la aplicación del circuito integrado FT232BM y sus accesorios, y otros programas en bibliotecas imprescindibles para el diseño de aplicaciones con microcontroladores de la familia sobre la base de una aplicación del microcontrolador PIC18Fxxx [1,7,19].
- Prototipo de sistema automatizado con comunicación RS232 [2] con PC a nivel de prototipo, para medir, indicar y controlar la temperatura de forma multiplexada en 8 canales de medición con destino al proceso de fermentación del tabaco en cámaras cerradas, empleando como sensores los transmisores de temperatura con salida por corriente, sobre la base de una aplicación con el microcontrolador Motorola [20].
- Prototipos para el control del secado del tabaco rubio TC07M empleando sondas PT100 y el cálculo de la humedad relativa por el método psicrométrico, construido sobre la base del diseño con el microcontrolador PIC 16F74 [1,3,7,19,20] con comunicación por protocolo RS232.
- También se cuenta con el desarrollo de una metodología propia para el diseño de aplicaciones con microcontroladores PIC, la cual se ha aplicado con éxito en el diseño de equipos y sistemas automatizados de adquisición, procesamiento y control de parámetros [1, 7, 14, 15,38].
- Se han creado conocimientos y habilidades en el trabajo con las herramientas de diseño para el trabajo con los microcontroladores PIC de la firma Microchip [1,3,9,10,11,12,11,13,14,15,16,17], que permiten simular, compilar, depurar errores y programar los componentes de la familia como son: el depurador/programador ICD2: software compilador CCSC (PCW) con lenguaje C para ensamblar código de los microcontroladores PIC, el Software

especializado MPLAB 7.0 para la simulación, puesta a punto y programación con los circuitos integrados PIC, el software de simulación de aplicaciones con microcontroladores PIC, denominado Proteus ISIS 7.0 [26], el Software especializado ORCAD 9.0 [27] para el diseño de circuitos impresos y la simulación de los circuitos electrónicos analógicos y de los circuitos lógicos, el Software especializado MatLab [25] para el tratamiento estadístico de las mediciones, Kit para el desarrollo de aplicaciones, muestras de los microcontroladores y CD-ROM de instalación de los software especializado para la familia de microcontroladores PIC Serie 18Fxxx [7,10]

- Además se cuenta con tarjetas electrónicas para el desarrollo de aplicaciones, como es el Kit demostrativo PICDEM FS sobre la base del circuito integrado microcontrolador PIC18F4550 diseñado especialmente para facilitar el diseño de la comunicación por puerto USB según la especificación USB 2.0,
- Se tienen conocimientos y habilidades en la programación de sistemas de monitoreo y control empleando como herramienta de diseño el Software LabWindows CVI para crear aplicaciones en la PC empleando lenguaje de alto nivel con programación objeto C++ [17].
- No existen reportes técnicos en publicaciones especializadas cubanas y en páginas WEB nacionales, sobre la construcción de sistemas automatizados con ambiente de red, cuyo nodo coordinador soporte su comunicación según las especificación estandarizada USB 2.0 con soluciones técnicas basadas en el empleo del desarrollo de aplicaciones con componentes de la familia de microcontroladores PIC.

Para desarrollar esta investigación se encuentra disponible un conjunto de modernas herramientas para el diseño [15,16,17,25,26,27] de aplicaciones empleando los microcontroladores PIC de la empresa Microchip, que facilitan el desarrollo de los microprogramas en lenguaje C y permiten simular, ensamblar, compilar, depurar errores y programar los componentes PIC, tales como: el depurador/programador ICD2, los compiladores CCSC (PCW) y C30, el Software MPLAB 7.0 para simular, poner a punto y programar los circuitos integrados PIC, el software de simulación de aplicaciones con microcontroladores PIC, denominado Proteus ISIS 6.0, y la simulación de los circuitos electrónicos analógicos y de los circuitos lógicos, el Software MatLab para el tratamiento estadístico de las mediciones.

Además, para realizar la investigación se propone desarrollar, construir y evaluar soluciones técnicas propias para obtener la comunicación USB entre un sistema de

adquisición y procesamiento de datos diseñado con un microcontrolador PIC y una PC según la siguiente especificación:

- La USB 2.0 empleando el kit de desarrollo PICDEM FS sobre la base de la aplicación del microprocesador PIC18F4550 para obtener la comunicación entre esta tarjeta y una PC con un microprograma de alto nivel.

Por lo que el aporte novedoso de la investigación radica en la obtención de microprogramas para el establecimiento de un protocolo de comunicación personalizado para este tipo de interfaz según las especificaciones antes descritas. Además, como puede apreciarse, estas soluciones tienen un carácter general que permite su integración al diseño de los sistemas de adquisición de datos. La reciente aparición del soporte técnico para el diseño de estas interfaces empleando microcontroladores PIC garantiza la actualidad y el elevado rigor científico de la investigación.

### **1.1 Fundamentación del Proyecto.**

Considerando las características técnicas del sistema a desarrollar, es necesario diseñar, construir, poner en funcionamiento y evaluar un conjunto de soluciones de hardware y software (micro programas) para obtener un Sistema Automatizado de adquisición de datos, procesamiento de la información y control de parámetros tecnológicos con comunicación con la PC por puerto USB según la norma 2.0. Además se requiere desarrollar y poner a punto un Software programado en lenguaje de alto nivel residente en la computadora (denominado Sistema de Monitoreo y Control), con el objetivo de controlar la transmisión y la recepción de la información digital, su almacenamiento, procesamiento y presentación interactiva al usuario.

El Proyecto de I+D solicitado incluye, diseñar, construir y evaluar las soluciones técnicas creadas, a nivel de maquetas, la simulación de los programas desarrollados con ayuda de Software especializados, y establecer las condiciones tecnológicas para su introducción rápida y sostenible en la práctica social.

Como parte integrante de este Proyecto de I+D presentado al Programa Ramal de la Electrónica del MIC, se crea la necesidad de desarrollar un conjunto de módulos jerárquicos que conduzcan con su integración, a lograr todas las soluciones técnicas que conducen a un exitoso diseño del sistema automatizado antes descrito. Entre estos módulos jerárquicos a desarrollar se identifica y presenta la necesidad de realizar un Proyecto de Investigación desarrollo, cuyo alcance técnico esté acotado

a: diseñar, construir y evaluar diversas soluciones de hardware y software para obtener la comunicación con la PC por puerto USB, con características propias y aplicarla en el diseño de sistemas de adquisición, procesamiento y control de parámetros tecnológicos, empleando como soporte técnico: el diseño y la construcción de aplicaciones con micro controladores del tipo PIC [1,4,5,6,7 ].

Además de forma general, se necesita integrar los resultados obtenidos al diseño de otros sistemas automatizados similares al descrito, y al desarrollo de tarjetas electrónicas y equipos que requieran establecer la comunicación por puerto USB con una computadora para cumplir las especificaciones de USB 2.0, empleando un protocolo propio o estandarizado.

Se prevé obtener resultados científico técnicos novedosos que permitan alcanzar en la práctica, soluciones de diseño con protocolos de comunicación que garanticen un sello distintivo para el producto y aseguren la soberanía tecnológica de estos sistemas de adquisición, procesamiento y control de parámetros tecnológicos en tiempo real, empleando el diseño y la construcción de aplicaciones con micro controladores, lo cual implica conducir la investigación con el desarrollo y la aplicación de soluciones técnicas muy actuales, de tecnología de punta y garantizar un elevado rigor científico en los resultados.

El contenido y alcance del módulo descrito constituye el tema de investigación que se propone en este Proyecto para obtener el grado científico de Master, los resultados científico técnicos y las salidas planificadas en el mismo son parte integrante del Proyecto de I+D presentado y aprobado en el Programa Ramal.

## **1.2 Herramientas de diseño, metodologías, y métodos de investigación a emplear en el Proyecto. Medios**

Para la realización del Proyecto se emplearán las siguientes metodologías y técnicas, con las que se espera garantizar la calidad de los resultados durante la ejecución del Proyecto, estas son:

1. Metodología para el desarrollo de aplicaciones con microcontroladores PIC, elaborada en la UCT LACETEL.
2. Sistema de gestión de la ciencia e innovación tecnológica. Procedimientos de diseño y Manual de Diseño elaborado en la UCT LACETEL, según el SG de calidad empleando las Normas Internacionales ISO9001 DEL 2001.
3. ICD2: Herramientas de diseño con microcontroladores PIC para detectar errores de programación y programar el PIC (Microchip).

4. Kit de desarrollo y herramientas de diseño para construir aplicaciones con microcontroladores PIC del tipo FS, que incluye: ensamblador, compilador, detección de errores de programación. Documentación técnica de las herramientas de diseño y notas de aplicación de la Microchip, para la construcción del nodo coordinador.
5. Software compilador CCSC (PCW) con lenguaje C++ para ensamblar código del PIC.
6. Software MPLAB 7.0 para la simulación, detección de errores y programación con PIC.
7. Software Borland Delphi para crear aplicaciones residentes en la PC.
8. Software MatLab para el tratamiento estadístico de las mediciones.
9. Proteus ISIS
10. Metodologías para evaluar los parámetros de calidad.
11. Además se prevé el intercambio técnico con expertos y la cooperación con otros centros universitarios y de investigación del país que investigan en temas afines.

### **1.3 Aportes novedosos.**

El aporte novedosos de la investigación radica en el desarrollo, la obtención y la evaluación de soluciones técnicas de hardware/software para obtener la comunicación por puerto USB según las especificación internacional USB 2.0, y factibles de aplicarse al diseño personalizado de sistemas automatizados de adquisición, procesamiento y control de parámetros tecnológicos, sobre la base de la implementación de estas soluciones técnicas novedosas con tecnología avanzada, en nuestras condiciones de trabajo a pedido de la entidad introductora, con el empleo de los microcontroladores: PIC18Fxxx y otros componentes electrónicos.

Las soluciones técnicas obtenidas pueden aplicarse de forma práctica al diseño de diferentes tipos de sistemas de adquisición y procesamiento de datos que empleen como soporte técnico los microcontroladores PIC.

Todo ello genera una base tecnológica con visión estratégica, que va a elevar la capacitación técnica y la capacidad nacional de crear nuevos productos y potenciar la innovación tecnológica en el campo de la electrónica y la automatización, permitiendo a mediano plazo obtener la diversificación de las ofertas de nuevos productos dirigidos a otras aplicaciones para el mercado nacional e internacional y de esta forma contribuir de forma eficiente a mejorar la producción, los servicios, la

economía, el desarrollo de la enseñanza universitaria, la cultura y la calidad de vida del pueblo.

#### **1.4 Impactos: Económicos, Sociales y Tecnológicos.**

1. Se espera lograr un impacto económico por concepto de ahorro de recursos en moneda convertible al dejar de importar estos sistemas y tenerlos en el mercado nacional a un costo menor mediante el diseño de estos.
2. La introducción y la aplicación de tecnologías de punta en el diseño de sistemas automatizados se espera que constituya un muy significativamente positivo impacto en el desarrollo tecnológico y la generalización de estos sistemas, mejorando la los costos de los procesos tecnológicos que automatiza. Con la automatización del control de procesos industriales se espera un mejoramiento de la calidad del producto, al realizarse las mediciones en forma continua, lo que traerá como consecuencia mayor eficiencia industrial, mayor productividad del trabajo y un ahorro de energía eléctrica, al mejorar el control de los parámetros del proceso tecnológico.
3. Se espera un muy positivo aporte en lo social, con la posibilidad de aplicarse en la defensa de nuestro país.
4. Además se espera un muy positivo impacto tecnológico al contribuir a alcanzar la soberanía tecnológica en el campo de la electrónica y aplicar esto a la defensa del país, a la producción y los servicios.
5. Debido a que se asimila y aplica una tecnología con un nivel alto de complejidad técnica, se espera lograr un salto cualitativo en la capacitación técnica del colectivo de investigadores y del personal que trabaja con el sistema.

#### **1.5 Resultados esperados.**

1. Se obtiene durante el proyecto de investigación, un resumen de la evolución y el estado actual de los aportes científico técnicos y de los procedimientos tecnológicos empleados para implementar la comunicación por puerto USB mediante la especificación internacional USB 2.0 empleando como soporte técnico los microcontroladores PIC y una PC y sobre la aplicación de este tipo de comunicación con la PC al diseño de sistemas automatizados para la adquisición y procesamiento de parámetros en procesos tecnológicos (SAPC),
2. Se presentan y evalúan, a nivel de maqueta, con el empleo de las tarjetas electrónicas propiamente diseñadas para esta investigación y la interfaz

electrónica que brinda el Kit de desarrollo FS PICDEM (todos estos hardware están soportados tecnológicamente por la construcción de aplicaciones basadas en los microcontroladores PIC: series 16Fxxx y 18Fxxx y la familia 18F2455 / 18F2450 / 18F4555 / 18F4550); para demostrar la validez de las soluciones técnicas de software desarrolladas, estableciendo la comunicación entre la interfaz electrónica del sistema y una PC por puerto USB según la especificación USB 2.0 con un protocolo personalizado que sea factible de aplicar al diseño de sistemas automatizados de adquisición y procesamiento de parámetros tecnológicos. Para mostrar el correcto funcionamiento del Protocolo, se presenta como resultado la ejecución de Programas demostrativos residente en la PC, elaborado en lenguaje de alto nivel u otro software factible, que permita validar satisfactoriamente que se establece y se controla la comunicación a través del puerto USB de la PC con la interfaz electrónica de la maqueta.

3. Se presenta una valoración técnico económica de los resultados obtenidos para introducir de forma sostenible esta tecnología en la fabricación a mayor escala de equipos electrónicos y SAPC personalizados.

# CAPÍTULO 2

## EL ESTÁNDAR USB Y SU APLICACIÓN EN LA COMUNICACIÓN DIGITAL ENTRE LOS SISTEMAS DE ADQUISICIÓN DE DATOS Y LA PC

### 2.0 El estándar USB

El USB (Universal Serial Bus), es un nuevo estándar de entrada/salida para periféricos de PC que aporta a los usuarios, conexiones simples, fáciles, y funciones de conectar-y-funcionar (plug and play). Desarrollado inicialmente por un consorcio de compañías líderes dentro de la industria del PC, el USB puede acomodar simultáneamente hasta 127 dispositivos periféricos. Con un único conector USB en la parte posterior de los PC reemplaza a los usuales puertos series y paralelos. Para añadir un periférico, el usuario ya no tendrá que establecer IRQs, o apagar el PC, abrir la carcasa del equipo, colocar el nuevo periférico y volver a ponerlo en marcha. Con el USB, el usuario puede conectar simplemente el periférico al Bus USB - incluso con el PC conectado- y el trabajo queda terminado. El periférico será detectado, caracterizado, configurado y listo para su uso, automáticamente, sin interacción del usuario. También se está utilizando para aplicaciones industriales.

Las transacciones USB, generalmente son isócronas, con dos velocidades de transmisión: un rango de baja velocidad hasta 1.5 Mbps, y un rango de media velocidad hasta 12.5 Mbps. El USB está centrado en aplicaciones de bajo costo y alto volumen. En el rango de baja velocidad, el USB se centra en dispositivos interactivos, tales como los "mouse" y "trackballs", teclados, juegos, sistemas de realidad virtual, etc. El rango de velocidad media se centra en aplicaciones ISDN y PBX, audio, transferencias de datos a "granel" (bulk) y vídeo limitado. La telefonía digital es el mayor objetivo para USB, debería de tener un rango de alta velocidad de 50 hasta 1000 Mbps todavía no disponible para las necesidades de vídeo completo y Redes Locales. Dicho de otra manera, el USB permite una conectividad extremadamente amplia con señales de velocidad media, utilizando circuitos convencionales.

Los dispositivos se incluyen al USB en una topología en estrella: varios periféricos se pueden incluir en un concentrador llamado "hub", otros periféricos se pueden incluir en otro "hub", etc.; por turno, los "hubs" se conectan al estilo de una cadena de margarita, alcanzando finalmente el "hub" raíz, incorporado en el procesador principal (host).

## **2.1 La comunicación por puerto USB. Tipos y Especificaciones. [2]**

La comunicación por puerto USB es una interfaz para la transmisión serie de datos y distribución de energía, desarrollado por empresas líderes del sector de las telecomunicaciones y de los ordenadores y que ha sido introducida en el mercado por las computadoras personales y periféricos para mejorar las lentas interfaces serie RS232 y paralelo. La comunicación por puerto USB provee una mayor velocidad de transferencia (de hasta 100 veces más rápido) comparado con la comunicación por puerto Paralelo y con la RS232, el objetivo inicial era conectar periféricos relativamente lentos (mouse, impresoras y cámaras digitales) de una forma rápida y eficiente, basada en comunicaciones serie, posteriormente su uso se extendió a la conexión de interfaces más complejas como son discos duros, tarjeta de adquisición de datos, equipos de medición en tiempo real, memorias flash, scanner y otros.

La primera computadora que incluyó un puerto USB de forma estándar fue el iMac de Apple, en 1998, para conectar el teclado y el Mouse, y Microsoft lo introdujo en los controladores correspondientes en la versión OSR 2.1 de Windows 95C cuando los sistemas de MS incorporan de forma estándar soporte para este bus y en el ámbito de servidores se incorpora en Windows 2000. Actualmente la comunicación USB se ha convertido en el método universal de conexión de periféricos [2].

## **2.2 Especificaciones USB**

Teniendo en cuenta la naturaleza general del equipo de comunicación, existen especificaciones que definen la arquitectura que es capaz de soportar cualquier dispositivo de comunicación. A continuación se relacionan en orden cronológico la aparición de estas especificaciones:

- Especificación USB 0.9:** Primer borrador, publicado en Noviembre de 1995.
- Especificación USB 1.0:** Publicada en 1996 establece dos tipos de conexión: La primera, denominada velocidad baja ("Low speed"), ofrece 1.5 Mbps, y se aplica en periféricos que no requieren un gran ancho de banda. La segunda, denominada velocidad completa ("Full speed"), es de 12 Mbps, y está destinada a los dispositivos más rápidos.
- Especificación USB 1.1:** Publicada en 1998, añade detalles y precisiones a la norma inicial; es el estándar mínimo que debe cumplir un dispositivo USB.

- **Especificación USB 2.0:** Su versión final fue publicada en Abril del 2000; es una extensión de la norma compatible con las anteriores. Permite velocidades de hasta 480 Mbps, denominada alta velocidad ("High speed").

En los sistemas de adquisición de datos se emplean con mayor frecuencia las especificaciones internacionales USB 1.0 y USB 2.0 para transmitir los datos

### 2.3 Cables y conectores para puerto USB

El cable del bus USB es de 4 hilos, y comprende líneas de señal (datos) y alimentación, con lo que las funciones pueden utilizar un único cable.

Pin	Nombre	Descripción	Color
1	VBUS	+ 5 V	rojo
2	D-	Data -	azul
3	D+	Data +	amarillo
4	GND	Tierra	verde

**TABLA 1**

### 2.4 Velocidad de transferencia del puerto USB

La velocidad de un periférico en el puerto USB depende del tipo de periférico y de la cantidad de dispositivos USB que comparten el puerto. Cuantos más dispositivos haya, menor será la velocidad de transferencia de datos. En general, USB proporciona una velocidad de transferencia de datos mayor que una conexión de puerto paralelo estándar con la ventaja real de que USB permite conectar varios dispositivos (hasta 127) en el mismo puerto sin conflictos, como se muestra en la siguiente tabla comparativa.

Tipo de conexión	Velocidad de transferencia (Megabits/segundo)
Puerto serie	0,92 Mbps
Puerto paralelo estándar	0,92 Mbps
USB 1.0/1.1	12 Mbps
USB 2.0	12 Mbps
USB 2.0 de alta velocidad	480 Mbps

**TABLA 2**

## 2.5 Funcionamiento del Protocolo de comunicación USB.

El bus serie USB es síncrono, y utiliza el algoritmo de codificación NRZI ("Non Return to Zero Inverted"). En este sistema existen dos voltajes opuestos; una tensión de referencia corresponde a un "1", pero no hay retorno a cero entre bits, de forma que una serie de "1" lógicos corresponde a un voltaje uniforme; en cambio los "0" lógicos se marcan como cambios del nivel de tensión, de modo que una sucesión de ceros produce sucesivos cambios de tensión entre los conductores de señal.

El protocolo de comunicación utilizado guarda cierta similitud con el sistema Token-Ring de IBM. Puesto que todos los periféricos comparten el bus y pueden funcionar de forma simultánea. La información es enviada en paquetes; cada paquete contiene una cabecera que indica el periférico a que va dirigido. Existen cuatro tipos de paquetes distintos: Token; Datos; Handshake, y Especial; el máximo de datos por paquete es de 8; 16; 32 y 64 Bytes. Se utiliza un sistema de detección y corrección de errores bastante robusto tipo CRC ("Cyclical Redundancy Check").

El funcionamiento está centrado en el host, todas las transacciones se originan en él. Es el controlador host el que decide todas las acciones, incluyendo el número asignado a cada dispositivo (esta asignación es realizada automáticamente por el controlador "host" cada vez que se inicia el sistema, se añade, o elimina, un nuevo dispositivo en el bus), su ancho de banda, entre otros. Cuando se detecta un nuevo dispositivo es el host el encargado de cargar los drivers oportunos sin necesidad de intervención por el usuario.

El sistema utiliza cuatro tipos de transacciones que resuelven todas las posibles situaciones de comunicación. Cada transacción utiliza un mínimo de tres paquetes, el primero es siempre un Token que avisa al dispositivo que puede iniciar la transmisión.

- Transferencia de control ("Control transfer"): Ocurre cuando un dispositivo se conecta por primera vez. En este momento el controlador de host envía un paquete "Token" al periférico notificándole el número que le ha asignado.
- Transferencia de pila de datos ("Bulk data transfer"): Este proceso se utiliza para enviar gran cantidad de datos de una sola vez. Es útil para dispositivos que tienen que enviar gran cantidad de datos cada vez, como escáneres o máquinas de fotografía digital.

- Transferencia por interrupción ("Interrupt data transfer"): Este proceso se utiliza cuando se solicita enviar información por el bus en una sola dirección (de la función al host).
- Transferencia de datos isócrona ("Isochronous data transfer"): Este proceso se utiliza cuando es necesario enviar datos en tiempo real. Los datos son enviados con una cadencia precisa ajustada a un reloj, de modo que la transmisión es a velocidad constante.

## **2.6 La especificación USB 2.0 [4]**

Esta versión salió al mercado a mitad del 2000, tras la unión al consorcio USB-IF de de las compañías: Hewlett, Packard, Philips y Lucent. La principal característica es el aumento de velocidad hasta 480 Mbps (casi 40 veces la velocidad anterior) con una diferencia de costo casi mínima. Soporta tres rangos distintos de velocidad, estos son:

- 1.Hi-Speed a 480 Mbps
- 2.Full-Speed a 12 Mbps.
- 3.Low-Speed a 1.5 Mbps.

Básicamente, USB 2.0 incluye todo lo que ofrece USB 1.1 y añade el modo de alta velocidad. USB 2.0 también usa el mismo tipo de cables y conectores para conectar los dispositivos de alta velocidad. Otro requisito es que es necesario un controlador de host para USB 2.0 para tener disponible la conexión de alta velocidad con un dispositivo de este tipo.

De lo anterior se infiere, que este tipo de comunicación (USB) es frecuentemente empleada en el diseño de los sistemas de adquisición, procesamiento y control de parámetros tecnológicos por su velocidad y fiabilidad de su protocolo., para ello se han desarrollado microcontroladores que poseen los recursos para efectuar la comunicación según USB 2.0, ejemplo de ello son los microcontroladores Pic de la Microchip

## **2.7 Evolución y Características técnicas de los microcontroladores Pic dotados de recursos para realizar la comunicación USB 2.0.**

### **2.7.1 Los PICs de Microchip. [10] [11]**

La evolución de la electrónica desde la aparición del circuito integrado ha ido en constante ascenso tecnológico, manifestándose con un aumento de la complejidad

funcional, de la velocidad de trabajo, el incremento de la cantidad de transistores que se integran en un chip, la diversidad de las aplicaciones donde se emplea y la mejora de sus características técnicas, entre otras cualidades. Actualmente podemos hallar componentes electrónicos con un muy alto grado de integración que realizan muy complejas operaciones y se presentan en encapsulados cada vez más pequeños, un ejemplo de esto son los microcontroladores PIC.

Un microcontrolador, es un dispositivo electrónico encapsulado en un chip, capaz de ejecutar un programa previamente almacenado en la memoria ROM del dispositivo, el cual puede ser reprogramado según cada aplicación deseada. El microcontrolador reúne en un solo chip, los siguientes recursos internos: unidad lógica y aritmética, unidad de procesamiento central (para ejecutar el conjunto de instrucciones del microcontrolador), memoria de programa (ROM), la memoria de datos (RAM y del tipo E2PROM) y los puertos de entrada/salida. Además, también disponen de otras estructuras funcionales con características especiales como: puertos serie, comparadores analógicos, convertidores analógico-digitales, temporizadores, contadores, entre otros bloques internos del microcontrolador.

En el mercado existen gran variedad de empresas que fabrican microcontroladores con características técnicas distintivas, por ello la selección del uso de estos depende de la aplicación a realizar, entre estos se encuentra la familia PIC desarrollada por la empresa Microchip. Existe una cantidad de aplicaciones que podemos realizar con los microcontroladores PICs, algunas muy sencillas en las cuales no necesitamos muchos recursos y otras más complejas en las cuales necesitamos microcontroladores muy potentes, por ello y siguiendo esta filosofía, la empresa Microchip produce una familia de tres tipos de gamas de microcontroladores PIC para facilitar el diseño de aplicaciones con diferentes grados de complejidad, con distintos tipos y cantidades de recursos internos, con cápsulas exteriores de los tipos (8DIP, 12DIP, 14DIP, 16DIP, 18DIP, 20DIP, 40DIP, cápsulas cerámicas SOIC y cápsula SMD para el montaje superficial) y con diferentes costos, estas gamas son: microcontroladores de gama baja (las series PIC10Fxxx y PIC12Fxxx), la gama media (la serie PIC16Fxxx) y la gama alta (la series PIC18Fxxx , PIC del tipo RF para la comunicación inalámbrica y PIC del tipo DSP para el procesamiento digital de señales, por lo que están disponibles microcontroladores sencillos y baratos para diseñar aplicaciones simples y otros componentes más complejos y costosos para desarrollar aplicaciones de mayor requerimiento técnico,

y es posible seleccionar el chip que mejor satisface las necesidades del diseño teniendo en cuenta el tipo y capacidad de las memorias, el número de líneas de E/S y las funciones auxiliares precisas. Sin embargo, todos los componentes PIC están contruidos con una arquitectura interna común, un conjunto de instrucciones compatibles para toda la familia, con frecuencia de trabajo de hasta 40 MHz, con un bajo consumo de potencia y un amplio margen del voltaje de alimentación entre 3 y 5 Voltios. Todas estas características técnicas lo hacen idóneo para ser empleado en el diseño de equipos electrónicos, sistemas automáticos de adquisición de datos, procesamiento y control y en otras interfaces electrónicas.

### **2.7.2 La familia PIC18Fxx50 y PIC18Fxx55**

Entre la gama alta de los microcontroladores PIC de la empresa Microchip, se encuentra la familia PIC18Fxx50 y PIC18Fxx55, a la cual pertenecen los componentes: PIC18F2455, PIC18F2550, PIC18F4555 y PIC18F4550, los cuales poseen entre sus múltiples recursos, la novedad tecnológica de poseer una interfaz de registros internos para el control de la comunicación por puerto USB con una PC según el estándar internacional USB 2.0, con la posibilidad de establecer velocidades de transmisión digital de datos según los tipos denominados Low Speed y Full Speed, ello permite efectuar este tipo de comunicación con la PC de forma eficiente y muy útil para el desarrollo de aplicaciones, como es: el diseño de sistemas automatizados de adquisición de datos, procesamiento y control de variables tecnológicas.

Estos dispositivos PIC cuentan con los recursos de hardware necesarios para realizar una conexión USB con las velocidades antes descritas. Las operaciones del protocolo de comunicación se realizan según un conjunto de rutinas de programas creadas por el diseñador para cada aplicación en específico y que emplean importantes recursos internos del microprocesador PIC, como son: el tratamiento de interrupciones, un conjunto de registros internos para el control de la comunicación USB, la memoria RAM del PIC para el almacenamiento del Buffer de datos a transmitir y/o recibir por puerto USB, la memoria ROM para almacenar el microprograma para el control del puerto USB, líneas de puerto de E/S para la conexión exterior con la PC y el empleo de los temporizadores internos del PIC para lograr un eficiente y exacto control del tiempo, lo cual garantiza el sincronismo en la transmisión y/o recepción de los paquetes digitales de datos entre la interfaz electrónica PIC y la PC.

Todas estas características técnicas brindan enormes ventajas para desarrollar la comunicación por puerto USB según el estándar USB 2.0 empleando el

microprocesador PIC, entre estas se hallan: una elevada fiabilidad y calidad en el control de la comunicación por puerto USB, flexibilidad para permitir el diseño personalizado del protocolo de comunicación según el programa específico generado por el diseñador para cada aplicación a diseñar, y un valioso soporte técnico para el desarrollo de esta aplicación con un relativo bajo costo.

Para el desarrollo de la comunicación por puerto USB se selecciona en esta investigación el microcontrolador PIC18F4550 cuyas características principales son:

### **2.7.3 Modos de Trabajo**

La familia de dispositivos: PIC18F2455 / PIC18F2550 / PIC18F4455 / PIC18F4550 presenta mejoras de diseño que hacen de estos microcontroladores una elección lógica para muchas aplicaciones de alto rendimiento y sus homólogos PIC18LF2455/ PIC18LF2550/ PIC18LF4455/ PIC18LF4550 se emplean en aplicaciones que requieran bajo consumo de energía, siendo elaborados con la tecnología Nanowatt.

Para reducir significativamente el consumo de energía durante la operación, estos dispositivos poseen los siguientes modos de trabajo.

- **Modo de corrida alternada:** Empleando el reloj mediante el Timer 1 o un oscilador interno es posible disminuir en un 90 % el consumo de energía durante la ejecución de las instrucciones.
- **Modo múltiple inactivo:** El controlador puede ejecutar el programa con el núcleo de CPU inactivos pero con los dispositivos periféricos activos. En este estado, el consumo de energía puede ser reducido a un 4 % de los requerimientos de operación normal.
- **Modo de conmutación durante la corrida:** Los modos de control de la energía son programados por el usuario durante la ejecución del programa, permitiendo a este incorporar recursos para lograr el ahorro de energía durante el diseño de las aplicaciones de software.
- **Modo de bajo consumo en el módulo de teclado:** En este caso los requerimientos de energía son minimizados para el Timer 1 y el perro guardián (Watchdog).

Los dispositivos de la familia PIC18F2455 / 2550 / 4455 / 4550 incorporan un módulo para la comunicación por puerto USB que es compatible con la especificación USB 2.0. El módulo permite tasas de transferencia de datos de los tipos: low-speed y full-speed. También incluye en el chip, un transmisor/receptor integrado y un regulador

de voltaje de 3.3V para la comunicación, brindando la posibilidad de conectar externamente al PIC, circuitos integrados del tipo “transceiver” (transmisor/receptor) y para regular el voltaje durante la comunicación.

#### **2.7.4 Conexiones del Oscilador**

La familia PIC18F2455 / 2550 / 4455 / 4550 permite doce opciones diferentes de conexión del oscilador. Éstos son:

- Cuatro tipos de cristales de cuarzo o resonador cerámico.
- Cuatro tipos de reloj externos, que brindan la opción de usar dos líneas (para la entrada del oscilador y para la salida de la señal de reloj de frecuencia dividida por 4) o emplear una línea para la entrada del oscilador y la segunda se asigna como una línea de entrada y salida general.
- Un bloque oscilador interno que suministra una señal de reloj de 8 MHz con una exactitud de  $\pm 2\%$  y obtiene una señal de interrupción aproximadamente de 31 kHz, estable con la temperatura y la alimentación de voltaje. También ofrece señales de reloj seleccionables por el usuario, con frecuencia entre 125 kHz y 4 MHz, para un total de 8 frecuencias de reloj. Esta opción libera una línea del oscilador que puede ser empleada como una línea adicional de entrada y salida con propósito general.
- Un multiplicador de frecuencia del tipo (PLL), disponible para modos de trabajo que empleen cristales de cuarzo de alta velocidad y osciladores externos, lo cual admite un amplio rango de frecuencias de reloj de 4 MHz a 48 MHz.
- La operación de dos relojes asincrónicos, que permiten que el módulo de USB se ejecute a partir de un oscilador de alta frecuencia mientras el resto del microcontrolador emplea un reloj mediante un oscilador interno de baja potencia. Además está disponible una fuente de reloj mediante un bloque oscilador interno que suministra una fuente de referencia estable que brinda ventajas adicionales para la ejecución de programas con estos dispositivos.

#### **2.7.5 Otras características de los Microcontroladores**

- **Memoria EEPROM:** Se emplea para el almacenamiento del programa fuente y de los datos, la cual admite más 100,000 ciclos de borrado/escritura para la memoria de programa, y 1, 000,000 de ciclos para la retención de los datos sin tener que refrescar la información con un tiempo de vida de 40 años.
- **Auto-Programmable:** El código fuente puede ser grabado en las memorias de programa mediante el control de un software interno, empleando una rutina bootloader, ubicada en un bloque de inicialización protegido en la zona superior de

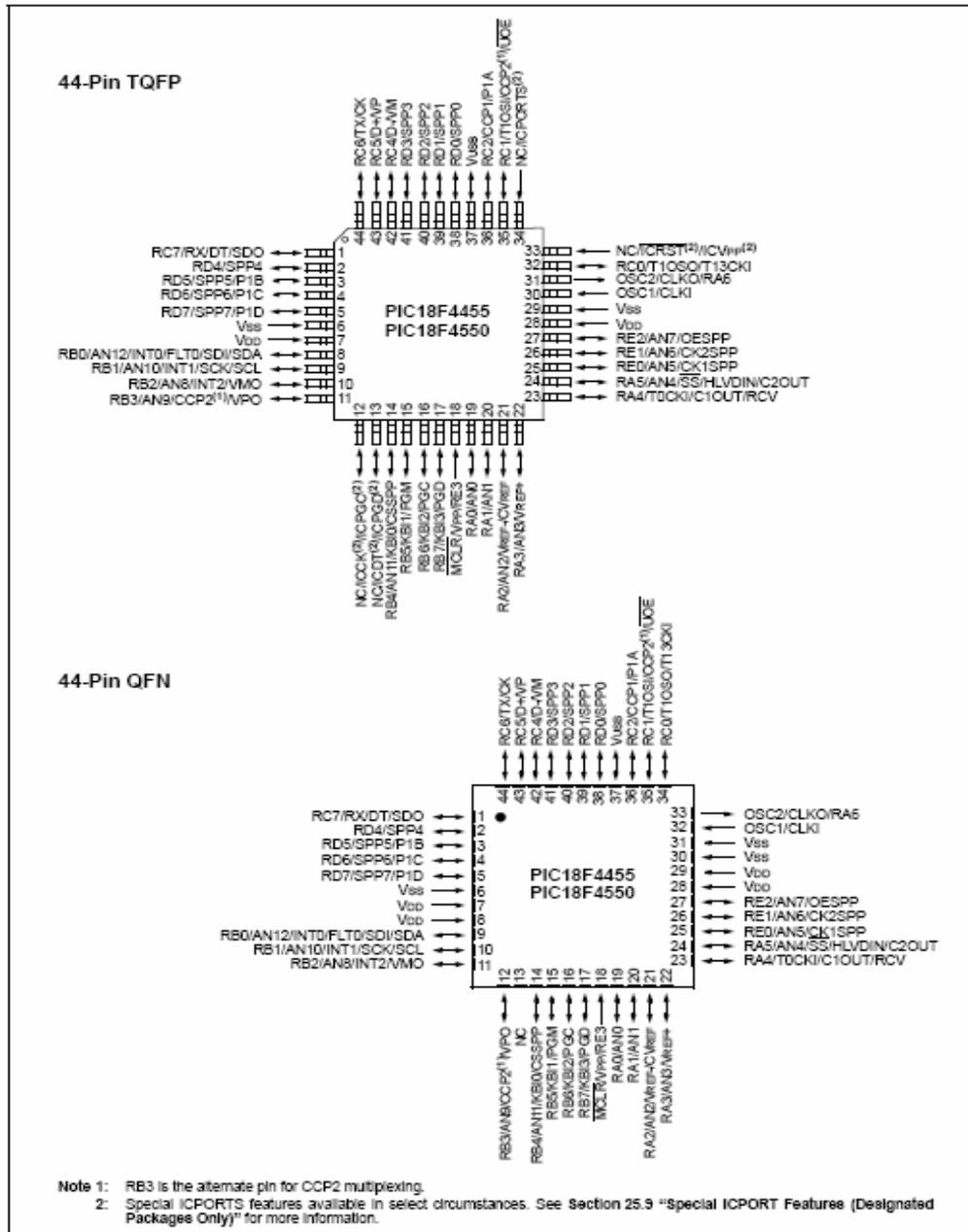
la memoria de programa, el cual hace posible crear una aplicación que puede actualizarse a sí misma en el área.

- **Conjunto de instrucciones extendidas:** Esta familia PIC introduce una extensión opcional del conjunto de instrucciones de la serie PIC18Fxxx, que añade 8 nuevas instrucciones y un modo de direccionamiento Indexado. Esta extensión, habilita una opción de configuración del dispositivo, que ha sido diseñada específicamente para optimizar aplicaciones con código reusable desarrollada en lenguaje de alto nivel.
- **Modulo USART:** Es un módulo que permite la comunicación según la operación del protocolo RS-232 y con la ventaja de incluir la detección automática de la velocidad de transmisión en baudios y además posee un generador de 16 bits para mejorar la resolución de la velocidad de transmisión.
- **Conversor analógico-digital de 10 bit:** Este módulo incorpora la ventaja de tener el tiempo de adquisición programable para un canal seleccionado y además permite que la conversión A/D se inicie sin esperar un período de muestreo, reduciendo la posibilidad de un código digital erróneo.
- **Líneas de puertos dedicadas a la herramienta ICD/ICSP:** Estas líneas de puertos permiten el uso de una herramienta externa de diseño denominada ICD/ICSP para depurar errores en el código fuente y programar los dispositivos, esta característica permite que los usuarios desarrollen aplicaciones de entrada y salida mientras conservan la habilidad de programar y depurar en el circuito. A continuación se presenta una tabla con las características técnicas de este grupo de la familia de los microcontroladores PIC, en la columna 1 de la misma se relaciona en forma consecutiva los siguientes parámetros técnicos: Frecuencia de operación en MHz, capacidad de memoria de programa EEPROM en bytes, capacidad de memoria RAM de datos en bytes, capacidad de memoria de programa EEPROM de datos en bytes, cantidad y tipo de fuente de interrupción al PIC, cantidad de líneas de puerto de entrada/salida del PIC, cantidad de temporizadores internos, cantidad y tipos de módulos de capturas (CCP) y de bloques de modulación del ancho del pulso (PWM), modulo de comunicación del puerto serie, módulo de comunicación para el control del puerto paralelo, cantidad de líneas del conversor análogo digital, tipos de reset del PIC, módulo programable de detección de bajo voltaje de alimentación (LVV), cantidad de instrucciones disponibles para la ejecución de microprogramas y tipos de encapsulados.

Features	PIC18F2455	PIC18F2550	PIC18F4455	PIC18F4550
Operating Frequency	DC – 48 MHz			
Program Memory (Bytes)	24576	32768	24576	32768
Program Memory (Instructions)	12288	16384	12288	16384
Data Memory (Bytes)	2048	2048	2048	2048
Data EEPROM Memory (Bytes)	256	256	256	256
Interrupt Sources	19	19	20	20
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	2	2	1	1
Enhanced Capture/Compare/PWM Modules	0	0	1	1
Serial Communications	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART
Universal Serial Bus (USB) Module	1	1	1	1
Streaming Parallel Port (SPP)	No	No	Yes	Yes
10-Bit Analog-to-Digital Module	10 Input Channels	10 Input Channels	13 Input Channels	13 Input Channels
Comparators	2	2	2	2
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Programmable Low-Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled
Packages	28-pin PDIP 28-pin SOIC	28-pin PDIP 28-pin SOIC	40-pin PDIP 44-pin QFN 44-pin TQFP	40-pin PDIP 44-pin QFN 44-pin TQFP

**TABLA 3:** Características técnicas de la familia de los microcontroladores PIC.

Los dispositivos en la familia del PIC18F4550 están disponibles en los paquetes de 40/44 pines. En la figura 1 se muestra el diagrama de terminales de entradas y salidas del dispositivo:



**FIGURA 1:** Diagrama de terminales de entradas y salidas del PIC18F4550.

El diagrama de bloques para este grupo es mostrado en la figura siguiente:

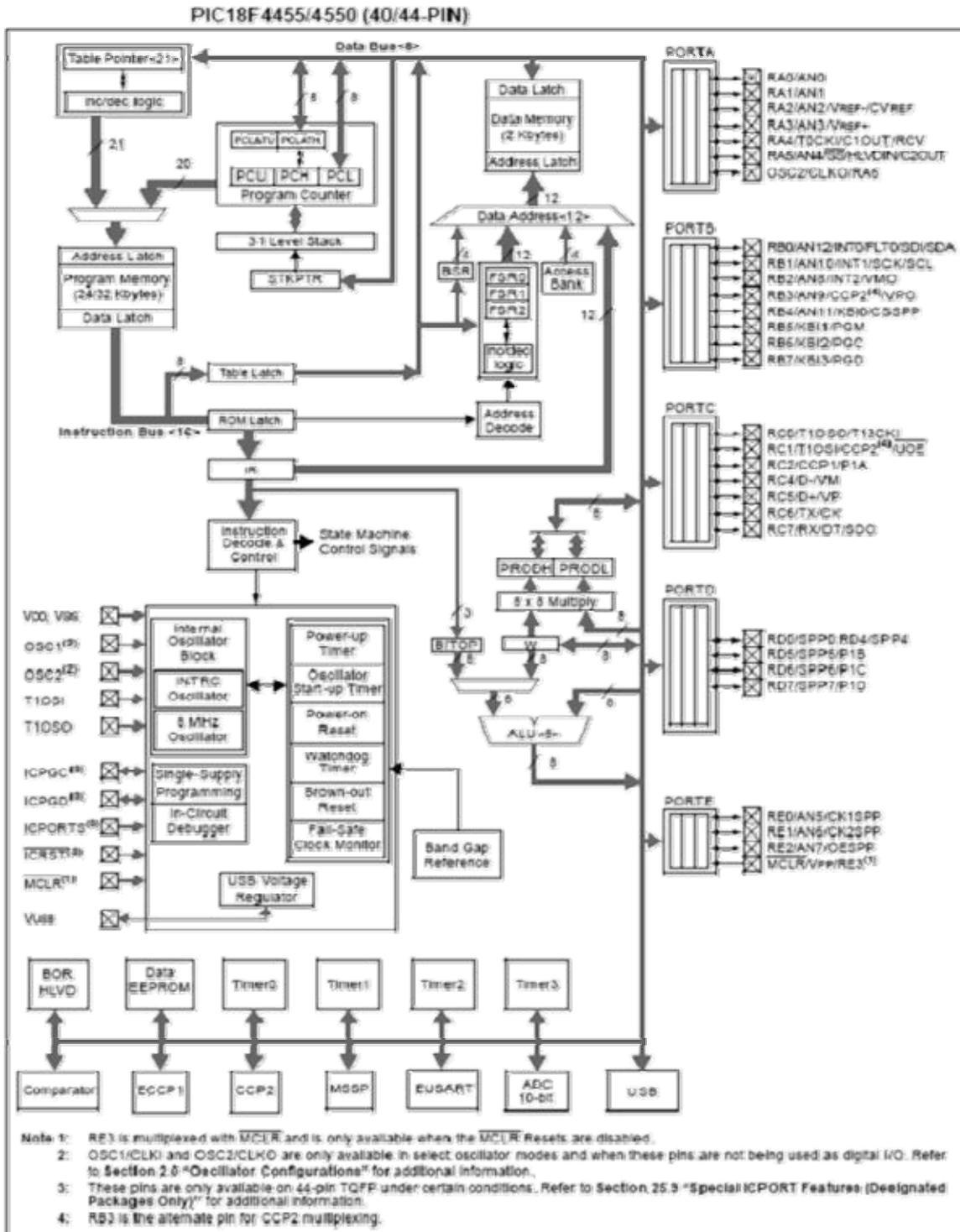


FIGURA 2: Diagrama de bloques de la estructura interna del PIC18F4550.

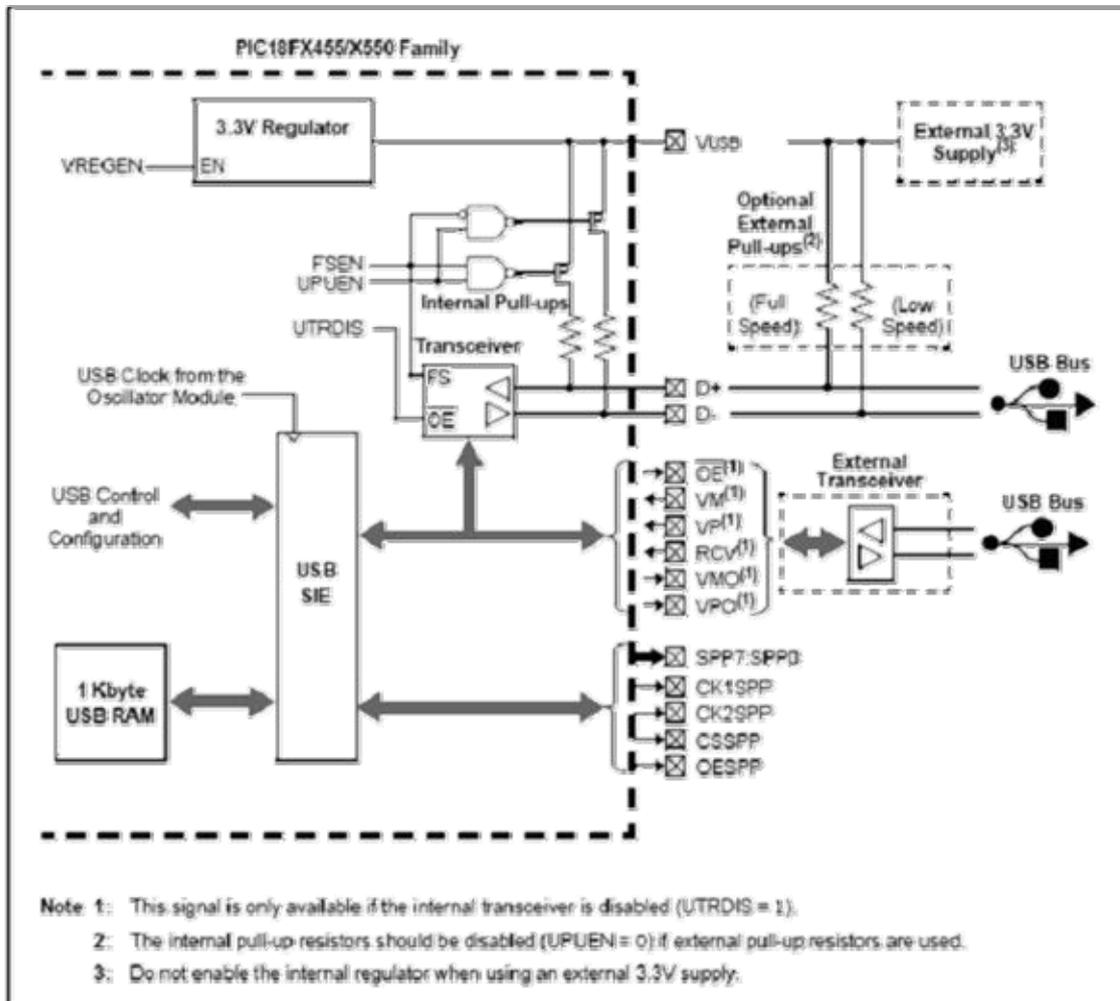
El microcontrolador para la tarjeta PICDEM FS USB es el PIC18F4550, principal dispositivo de la familia PIC18F2455/2550/4455/4550. Todos estos dispositivos implementan tecnología de Nanowatt, presentan un ahorro de energía en su fuente de alimentación, memorias de programa mejoradas y características técnicas nuevas para controlar la comunicación a través del puerto USB, éstas son:

- Cumplimiento de la especificación USB 2.0.
- Máxima Velocidad (12Mbit/s) y Baja Velocidad de operación (1.5Mbit/s).
- El soporte tecnológico para el control y la interrupción de la comunicación.
- Soporte de hasta 32 puntos finales.
- 1 Kbyte de doble acceso de RAM para USB.
- Las características del circuito permiten la realización de la comunicación por puerto USB, incluye: Transmisor de USB, Regulador de voltaje de USB y Resistores pull-up en las líneas de puerto para la comunicación USB.

Esta familia de dispositivos de PIC posee un hardware interno que realiza la función de interfaz de control del puerto USB, denominada USB Serial Interface Engine (SIE) la cual permite alcanzar todo el rango de velocidades de transmisión compatibles con este tipo de comunicaciones entre cualquier periférico con puerto USB y el microcontrolador PIC. El SIE puede ser conectado directamente al USB, utilizando el transmisor/receptor interno, o puede ser conectado a través de un circuito integrado transmisor/receptor externo. El microcontrolador PIC posee un regulador de voltaje interno de 3.3V, es también disponible para suministrar energía al circuito interno del transmisor/receptor en caso de que las aplicaciones se alimenten con 5V.

El SIE esta compuesto por bloques circuitales de hardware con características especiales que mejoran la velocidad de la transmisión, éstas son: Una memoria RAM con doble registro para el acceso a la zona de memoria de datos (denominada como RAM de USB) y ello permite acceder y operar sobre esta memoria de forma compartida por el CPU del microcontrolador y el SIE, zonas de memoria para ser configuradas como buffer con información descriptiva que permiten ser programadas libremente dentro del espacio de la RAM de USB y un circuito electrónico para realizar las funciones de un puerto paralelo de transmisión continua que soporta la transferencia ininterrumpida de grandes volúmenes de datos, enviados y recibidos desde los buffers en las memorias externas. A continuación se presenta un

diagrama en bloque del hardware interno del PIC18F4550 para dar solución circuital a la comunicación por puerto USB según la especificación 2.0.



**FIGURA 3:** Diagrama en bloque del hardware interno del PIC18F4550 para atender la comunicación con dispositivos periféricos USB.

### 2.7.6 Operación del módulo de USB

La operación del módulo de USB es configurada y dirigida a través de tres registros de control y otros 19 registros que son usados para dirigir las operaciones actuales de USB, algunos de ellos son: [10]

1. Registro de Control USB (UCON).
2. Registro de configuración USB (UCFG).
3. Registro de transferencia de estado USB (USTAT).
4. Registro de dirección del dispositivo USB (UADDR).
5. Registros de números enmarcados (UFRMH: UFRML).
6. Registros de punto final permitidos (de 0 a 15) (UEPn).

7. Registro BDnSTAT (Modo CPU).
8. Registro BDnSTAT (Modo de SIE).
9. Registro de estado de interrupción USB (UIR).
10. Registro permitido de interrupción USB (UIE).
11. Registro permitido por la interrupción de error USB (UEIE).

A continuación se describen algunos de ellos.

### **Registro de Control USB (UCON)**

#### **Composición del registro:**

U-0	R/W-0	R-x	R/C-0	R/W-0	R/W-0	R/W-0	U-0
—	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	—
bit 7							bit 0

#### **Bit 6: PPBRST:** Ping-Pong Buffers Reset bit

**1** = Reset todos los punteros a los buffers de los bancos de memoria intermedia par

**0** = No realiza el reset

#### **Bit 5: SE0:** Bit de bandera de cero terminada por Live Single-Ended.

**1** = Detecta Single-Ended en el bus de USB con cero activo por cero activo

**0** = El cero no detectar Single-Ended.

#### **Bit 4: PKTDIS:** Transferencia de paquete de bit

**1** = SIE simbólico y procesamiento de paquete desactivado, fijado automáticamente cuando una ficha de configuración es recibida.

**0** = Ficha de SIE y paquete que el procesamiento permitió.

#### **Bit 3: USBEN:** Módulo permitido de bit de USB

**1** = Módulo de USB y circuitería de soporte permitida (dispositivo adjuntado).

**0** = Módulo de USB y circuitería de soporte desactivada (dispositivo separado).

#### **Bit 2: Resumen:** Resumen Disponible para indicar bit.

**1** = Resumen indica activo.

**0** = Resumen indica desactivado.

#### **Bit 1: SUSPND:** Suspend Bit USB.

**1** = Módulo de USB y reloj de SIE inactivo.

**0** = Módulo de USB y reloj de SIE registrado al configurar la frecuencia.

#### **Bit 0: Unimplemented:** Leer como "0".

El registro de Control USB contiene bits necesarios para controlar el comportamiento del módulo durante la transferencia. El registro contiene las partes de configuración que controlan lo siguiente:

1. Principal dispositivo periférico USB.
2. Reset del puntero del buffer Ping-Pong.
3. Control del modo de Reposo.
4. Desactivar Transferencia de Paquete.

El registro de Control USB contiene un bit de estado, que es usado para demostrar la ocurrencia de un cero single-ended en el bus. Cuando el módulo de USB está activado, este bit debe ser monitoreado para determinar si las líneas de datos del

diferencial han salido de la condición de cero single-ended. La operación en conjunto del módulo de USB es controlada por el bit de USBEN. Este bit también activa al regulador de voltaje y conecta resistores de pull -up internos, si están activados.

## Registro de configuración USB (UCFG)

### Composición del registro:

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UTEYE	UOEMON <sup>(1)</sup>	—	UPUEN <sup>(2,3)</sup>	UTRDIS <sup>(2)</sup>	FSEN <sup>(2)</sup>	PPB1	PPB0
bit 7							bit 0

**Bit 7: UTEYE:** USB Eye Pattern Test Enable bit.

**Bit 6: UOEMON:** USB OE Monitor Enable bit.

**1** = OE señal activa; indica los intervalos en que las líneas D+/D- conducen.

**0** = OE señal inactiva.

**Bit 5: Unimplemented:** Leer como "0".

**Bit 4: UPUEN:** USB On-Chip Pull-up Enable bit.

**1**= On-Chip Pull-up habilitado (Pull-up sobre D+ con FSEN = 1o D- con FSEN = 0).

**0**= On-chip pull-up inhabilitado.

**Bit 3: UTRDIS:** On-Chip Transceiver Disable bit.

**1** = On-chip transmisor inhabilitado; interfaz del transmisor digital permitida.

**0**= On-chip transmisor activo.

**Bit 2: FSEN:** Velocidad Máxima de bit permitida.

**1** = Dispositivo de máxima velocidad: requiere un reloj de entrada a 48 MHz.

**0** = Dispositivo de baja velocidad: requiere un reloj de entrada a 6 MHz.

**Bit 1-0: PPB1:PPB0:** protege los bits de configuración del buffer Ping-Pong.

**11** = Reservar.

**10**= Even/Odd ping-pong permite proteger para todos los puntos finales.

**01** = Even/Odd ping-pong permite proteger para OUT el punto final 0.

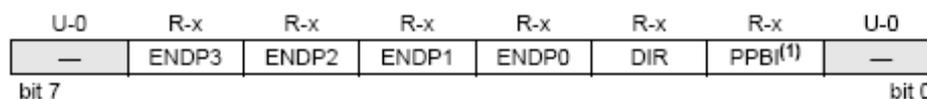
**00** = Even/Odd ping-pong protege desactivado.

Antes de la comunicación sobre USB, el módulo interno y externo asociado al hardware debe ser configurado con el registro de UCFG. El regulador de voltaje de USB es controlado a través de los registros de configuración. El registro UCFG contiene la mayoría de los bits que controlan el comportamiento del nivel del sistema del módulo de USB. Éstos son:

- La velocidad del bus.
- Resistor disponible On-chip pull-up.
- Transceiver disponible On-chip.
- Uso del buffer Ping-pong.

## Registro de estado USB (USTAT)

### Composición del registro:



**Bit 7: Unimplemented:** Leer como '0'.

**Bit 6-3: ENDP3:ENDP0:** Codificar la cantidad de la última actividad del punto final.

**1111** = Punto final 15

**1110** = Punto final 14

**0001** = Punto final 1

**0000** = Punto final 0

**Bit 2: DIR:** Último bit del indicador de Dirección de BD.

**1** = La última transacción existe en la señal IN.

**0** = La última transacción existe en la señal OUT o SETUP.

**Bit 1: PPBI:** Ping-Pong BD bit indicador.

**1** = La última transacción existe en el bank Odd BD.

**0** = La última transacción existe en el bank Even BD.

El registro de estado USB informa sobre el estado de la transacción dentro del SIE. Cuando el SIE emite una transferencia de interrupción completa de USB, el registro USTAT debe ser leído para determinar el estado de la transferencia y el registro USTAT contiene el número del punto final de transferencia, la dirección y el valor del puntero al buffer ping-pong.

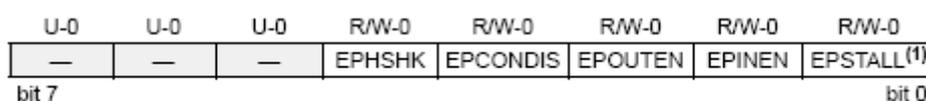
## Registro de dirección USB (UADDR)

El registro de dirección USB contiene la dirección única de USB que el dispositivo periférico descifrará cuando esté activo. UADDR es vuelto a poner a 00h cuando es recibido un Reset de USB, demostrado por URSTIF, o cuando es recibido un Reset del microcontrolador. La dirección de USB debe ser escrita por el microcontrolador durante la fase de configuración de USB.

## Registros de números enmarcados (UFRMH:UFRML)

Los registros de números enmarcados contienen el número de marco de 11 bits. El byte low-order es contenido en UFRML, mientras que los tres bits high-order están contenidos en UFRMH. El par de registro es actualizado con el número de marco en curso siempre que una ficha de SOF es recibida. Para el microcontrolador, estos registros son de solo lectura. El registro de números enmarcados se usa para los traslados isócronos principalmente.

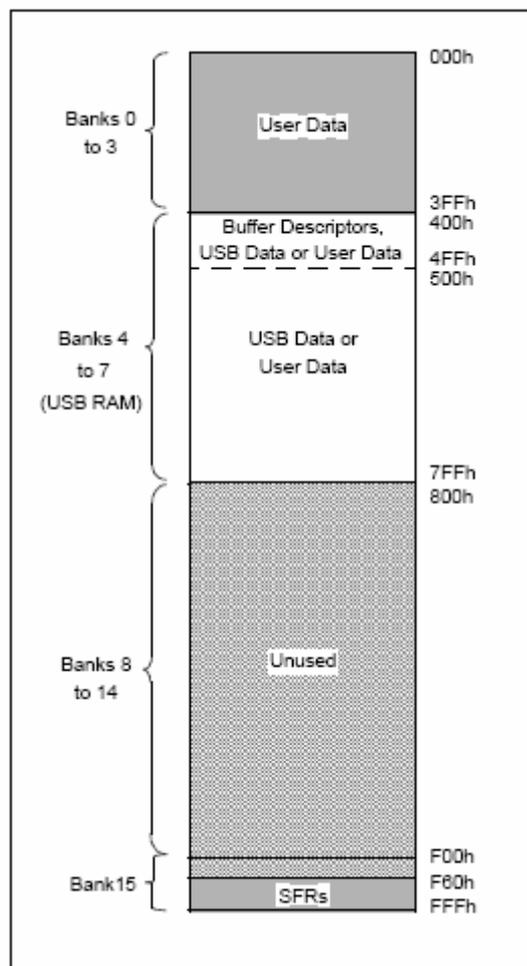
## Control del punto final de USB (UEPn)



Cada uno de los 16 puntos finales bidireccionales posibles tienen su propio registro de control independiente UEPn. El bit de EPHSHK controla el intercambio de indicadores de control para el punto final; poner este bit en 1 permite el intercambio de indicadores de control de USB.

### 2.7.7 RAM USB

Los datos de USB se mueven entre el núcleo del microcontrolador y el SIE por un espacio de memoria conocido como RAM USB (ver figura 4). Ésta es una memoria especial con un doble puerto de acceso. Aunque la RAM de USB está asequible al microcontrolador como la memoria de datos, las secciones que están siendo accedidas por el SIE no deben ser accedidas por el microcontrolador.



**FIGURA 4:** Estructura de la memoria RAM USB.

## 2.7.8 Buffer Descriptor y el Buffer Descriptor de Tabla

Los registros en el Bank 4 son usados específicamente para el control del buffer del punto final en una estructura conocida como el Cuadro (BDT) del descriptor del buffer. El BDT está compuesto del buffer descriptor (BD) que se usa para definir y controlar los verdaderos buffers hacia dentro del espacio de la RAM USB. Cada BD consta de cuatro registros, donde n representa uno de los 64 BDs posibles:

BDnSTAT: Registro de estado de BD, BDnCNT: Registro Contador de byte de BD, BDnADRL: Registro de Dirección nible bajo de BD, BDnADRH: Registro de dirección nible alto de BD.

### Registro del estado de interrupción USB (UIR)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0
—	SOFIF	STALLIF	IDLEIF <sup>(1)</sup>	TRNIF <sup>(2)</sup>	ACTVIF <sup>(3)</sup>	UERRIF <sup>(4)</sup>	URSTIF
bit 7							bit 0

**Bit 7: Unimplemented:** Leer como '0'.

**Bit 6: SOFIF:** START-OF-FRAME Bit interrupción de señal.

**1** = Una ficha de START-OF-FRAME recibida por el SIE.

**0** = Ninguna ficha de START-OF-FRAME recibida por el SIE.

**Bit 5: STALLIF:** Una interrupción de bit STALLIF: A STALL Handshake.

**1** = Una STALL Handshake fue enviada por el SIE.

**0** = Una STALL Handshake no fue enviada por el SIE.

**Bit 4: IDLEIF:** Bit de interrupción detectada inactiva.

**1** = Condición inactiva detectada (Estado vacante constante de 3 milésimas de segundo o más).

**0** = No detectada condición inactiva.

**Bit 3: TRNIF:** Bit de interrupción Completa de transacción.

**1** = El procesamiento de la transacción pendiente está completo.

**0** = El procesamiento de la transacción pendiente no está completo.

**Bit 2: ACTVIF:** Bit de interrupción detectada de actividad del bus.

**1** = La actividad sobre las líneas de D + / D- fue detectada.

**0** = Ninguna actividad detectada sobre las líneas de D + / D-.

**Bit 1: UERRIF:** Bit de interrupción de condición de error de USB.

**1** = Una condición de error desenmascarada ha ocurrido.

**0** = Ninguna condición de error desenmascarada ha ocurrido.

**Bit 0: URSTIF:** Bit de interrupción de Reset de USB.

**1** = Reset de USB válido ocurrió; 00h es cargado en el registro de UADDR.

**0** = Reset de No USB ha ocurrido.

El registro de estado de interrupción USB contiene las partes del indicador de bits para cada una de las fuentes de interrupción de estado de USB. Cada uno de estos recursos tiene su interrupción correspondiente, al permitir bit en el registro de UIE.

## Registro de habilitación de interrupción USB (UIE)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE
bit 7							bit 0

**Bit 7: Unimplemented:** Leer como '0'.

**Bit 6: SOFIE:** START-OF-FRAME Bit interrupción de señal.

1 = Una ficha de START-OF-FRAME recibida por el SIE.

0 = Ninguna ficha de START-OF-FRAME recibida por el SIE.

**Bit 5: STALLIF:** STALL Handshake bit de interrupción permitida.

1 = Una STALL fue enviada por el SIE.

0 = Una STALL no fue enviada por el SIE.

**Bit 4: IDLEIF:** Bit de interrupción detectada inactiva.

1 = Idle detecta la interrupción permitida.

0 = Idle detecta la interrupción desactivada.

**Bit 3: TRNIF:** Bit de interrupción Completa de transacción.

1 = La interrupción de transacción permitida.

0 = La interrupción de transacción desactivada.

**Bit 2: ACTVIF:** Bit de interrupción detectada de actividad del bus.

1 = La actividad del bus detecta la interrupción permitida.

0 = La actividad del bus detecta la interrupción desactivada.

**Bit 1: UERRIF:** Bit de interrupción de error de USB.

1 = La interrupción de error de USB permitida.

0 = La interrupción de error de USB desactivada.

**Bit 0: URSTIF:** Bit de interrupción permitida del Reset de USB.

1 = La interrupción del Reset de USB permitida.

0 = La interrupción de Reset de USB desactivada.

La interrupción que el registro USB permite contiene las partes para los orígenes de la interrupción del estado de USB. Colocar algunos de estos bits en uno habilitará el origen de interrupción respectiva en el registro de UIR.

## Registro de estado de interrupción de error USB (UEIR)

R/C-0	U-0	U-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
BTSEF	—	—	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF
bit 7							bit 0

El registro de estado de interrupción de error USB contiene los indicadores para cada uno de los orígenes de error dentro del dispositivo periférico de USB.

## Registro permitido por la interrupción de error USB (UEIE)

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BTSEE	—	—	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE
bit 7							bit 0

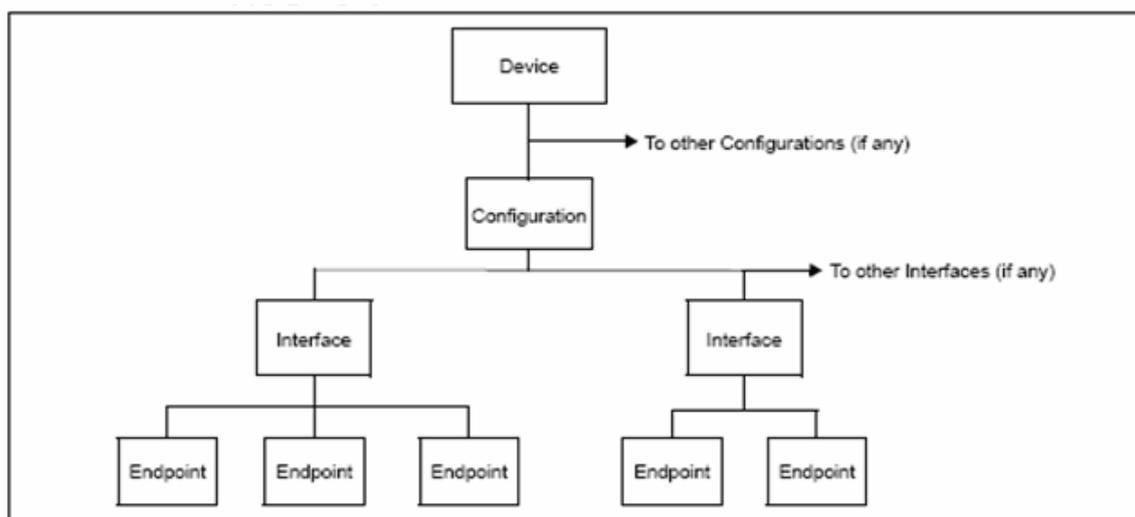
El error de USB que el registro de interrupción habilita cada uno de los orígenes de interrupción de error de USB.

### 2.7.9 Conceptos generales de la estructura de la comunicación.

#### Entramado en capas

La funcionalidad del dispositivo de USB está estructurada en un entramado dividido en capas como se muestra en la figura 5. Cada nivel se relaciona con un nivel funcional dentro del dispositivo. La capa más alta, aparte del dispositivo, es la configuración. Un dispositivo podría tener múltiples configuraciones; por ejemplo, un dispositivo especial podría requerir múltiples tipos de alimentación como son: los modos Self-Power Only o Bus Power Onlys. Para cada configuración, pueden existir múltiples interfaces. Cada interfaz podría soportar un modo específico de esa configuración.

El nivel inferior de la interfaz está formado por los dispositivos denominados endpoint(s) o puntos finales. Los datos son directamente movidos en este nivel. Puede existir hasta un total de 16 dispositivos de este tipo con comunicación bidireccionales. El dispositivo final denominado Endpoint 0 es siempre el punto final de control, y por defecto cuando el dispositivo está en el bus, Endpoint 0 está disponible para configurar el dispositivo.

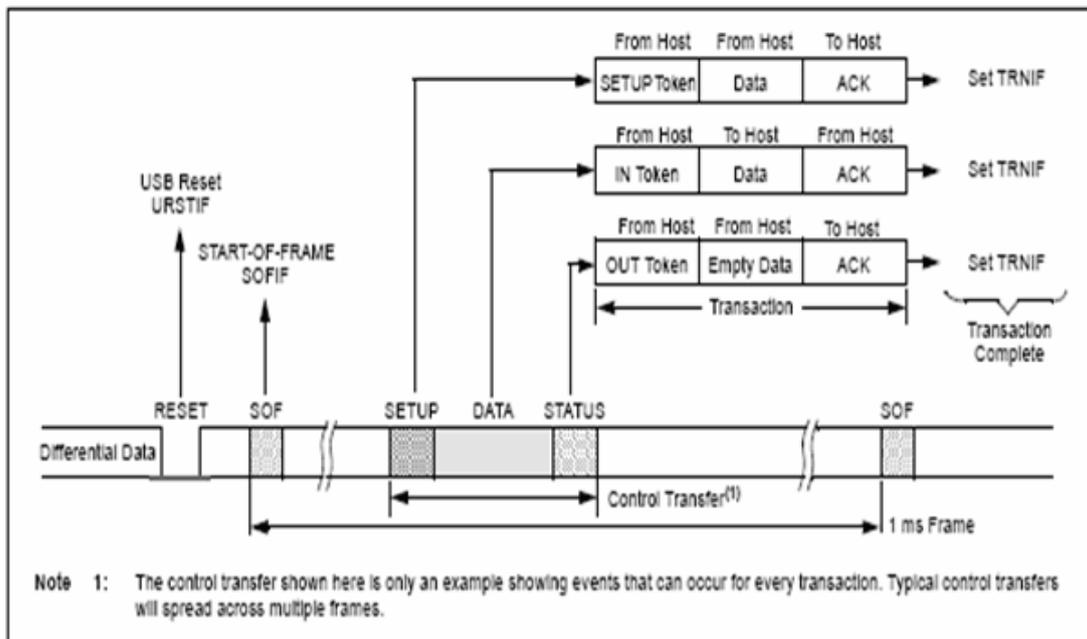


**FIGURA 5:** Estructura de la comunicación USB.

#### Marcos

La información a comunicar en el bus se agrupa en una milésima de segundo en una ranura de la trama. Cada trama puede contener transacciones a varios dispositivos y

a puntos finales. La siguiente figura indica un ejemplo de una transacción dentro de una trama.



**FIGURA 6:** Estructura de las transacciones de las tramas.

### **Transferencia**

Hay cuatro tipos de transferencia definidos en la especificación de USB.

- **Isócrono:** Este tipo suministra un método de transferencia para grandes cantidades de datos (hasta 1023 bytes) con entrega oportuna y fiable; sin embargo, la integridad de datos no está garantizada asegurada. Esto es bueno para las aplicaciones de transmisión continua donde la pérdida de datos pequeña no es crítica, como el audio.
- **Voluminoso:** Este tipo de método de transferencia permite enviar grandes cantidades de datos, garantizando la integridad de datos; sin embargo, esta entrega no se realiza de forma oportuna, no se asegura su transferencia en un tiempo óptimo.
- **Interrupción:** Este tipo de transferencia proporciona una entrega en el momento oportuno y con fiabilidad de bloques pequeños de datos. Además la integridad de datos está garantizada.
- **Control:** Este tipo de transferencia proporciona el control de la configuración de los dispositivos.

Los dispositivos de alta velocidad soportan todos los tipos de transferencia de datos, los dispositivos de baja velocidad están limitados a los tipos de transferencia por interrupción y del tipo control.

## **Energía de alimentación**

La energía para la alimentación de los dispositivos USB está disponible en el bus USB. La especificación de la comunicación USB define los requisitos de potencia del bus. Los dispositivos pueden ser auto alimentados desde una fuente externa (self-powered) o alimentados desde el propio bus (bus powered).

La especificación del puerto USB limita la potencia que se obtiene del bus. Cada dispositivo tiene asegurada una corriente de 100 mA a 5V, por unidad de carga. Una potencia adicional puede ser solicitada hasta un máximo de 500 mA. Las especificaciones también definen el modo de trabajo Suspendidos. En esta situación, la corriente promedio es limitada a 500  $\mu$ A, durante 1 segundo.

## **Enumeración**

Cuando el dispositivo es inicialmente fijado al bus, el periférico entra en un proceso de enumeración para identificar el dispositivo. Esencialmente, el periférico interroga el dispositivo, obteniendo la información sobre el consumo de energía, el tamaño de los datos, el protocolo y otras informaciones descriptivas.

## **Descriptores**

Existen 8 tipos de descriptor diferentes, de ellos cinco son los más importantes para este dispositivo, y éstos son:

### **Descriptor de dispositivo**

El descriptor de dispositivo suministra la información general sobre el fabricante, el producto, el número de serie, la clase de dispositivo y el número de configuraciones. Hay solamente un descriptor por dispositivo.

### **Descriptor de configuración**

El descriptor de configuración provee la información sobre los requisitos de alimentación del dispositivo y cuantas interfaces diferentes pueden ser soportadas con esta configuración. Puede haber más de una configuración para un dispositivo.

### **Descriptor de interfaz**

Los detalles del descriptor de interfaz describen los puntos finales que emplea esta interfaz, tanto como la clase de la interfaz. Puede haber más de una interfaz para una configuración.

### **Descriptor de punto final**

Para el punto final este descriptor identifica los tipos de transferencia y dirección, también como algunos otros detalles concretos el punto final. Puede haber muchos puntos finales en un dispositivo y los puntos finales pueden ser compartidos en configuraciones diferentes.

### **Descriptor de cadena**

Muchos de los descriptores previos mencionan a uno o más descriptores de una cadena de estos tipos y proveen la información legible sobre la capa que describen. A menudo estas cadenas aparecen en el anfitrión para ayudar al usuario a identificar el dispositivo. Estos son en general opcionales para salvar la memoria y son codificados en un formato de código único.

# **CAPÍTULO 3**

## **OBTENCIÓN DE LA COMUNICACIÓN POR PUERTO USB ENTRE UNA APLICACIÓN CON MICROCONTROLADORES PIC Y UNA PC SEGÚN LA ESPECIFICACIÓN USB 2.0**

### **3.0 Soluciones de Hardware para la realización de la comunicación con la PC por puerto USB.**

En este Proyecto de Investigación se propone desarrollar, construir y evaluar soluciones técnicas propias para obtener la comunicación USB 2.0 entre un sistema de adquisición y procesamiento de datos diseñado con un microcontrolador PIC18F4550, el cual fue descrito en el capítulo anterior y que está preparado para este tipo de comunicación por puerto USB 2.0 y con una PC.

La USB 2.0 empleando el kit de desarrollo PICDEM FS sobre la base de la aplicación del microprocesador PIC18F4550 para obtener la comunicación entre esta tarjeta y una PC.

Por lo que el aporte novedoso de la investigación radica en la obtención de los microprogramas para el establecimiento de un protocolo de comunicación personalizado para este tipo de interfaz según la especificación USB 2.0 antes descritas. Además como puede apreciarse estas soluciones tienen un carácter general que permite su integración al diseño de los sistemas de adquisición de datos. La reciente aparición del soporte técnico para el diseño de estas interfaces empleando microcontroladores PIC garantiza la actualidad y el elevado rigor científico de la investigación.

### **3.1 Requerimientos técnicos para el desarrollo del Software para la comunicación con la PC por puerto USB. [6]**

Para el desarrollo de los microprogramas en la comunicación a través del puerto USB empleando como soporte tecnológico los microcontroladores PIC se necesita satisfacer los siguientes requerimientos técnicos:

- Para la comunicación mediante la especificación USB 2.0, se establece un protocolo de comunicación entre la interfaz PIC y la PC, en este caso el microprograma a desarrollar para el control de estas acciones se realiza en lenguaje C y se compila para obtener un código fuente en lenguaje ensamblador compatible con los circuitos integrados de la familia PIC. Este microprograma tiene entre sus acciones configurar todos los registros internos del microprocesador PIC18F4550 que intervienen en el

proceso de comunicación, así como las rutinas de interrupción y de transmisión recepción del microcontrolador, la configuración del buffer de entrada y salida, de los temporizadores que intervienen y del conteo de la base de tiempo (reloj interno) necesaria para la velocidad de transmisión seleccionada.

Además en la PC se ejecuta un programa demostrativo para establecer la comunicación con la interfaz, mostrar y graficar parámetros recepcionados desde la interfaz PIC y permitir la programación de acciones desde la PC que son ejecutadas en la interfaz PIC al efectuarse la transmisión desde la PC.

Como puede apreciarse, para establecer la comunicación por puerto USB es necesario ejecutar un programa en lenguaje de alto nivel residente en la PC y un microprograma en lenguaje ensamblador almacenado en la EEPROM del microcontrolador PIC de manera que ambos software de forma simultánea puedan establecer un protocolo de comunicación confiable y eficaz que asegure el éxito de la comunicación a través del puerto USB y puedan cumplirse las especificaciones técnicas de calidad para este tipo de comunicación.

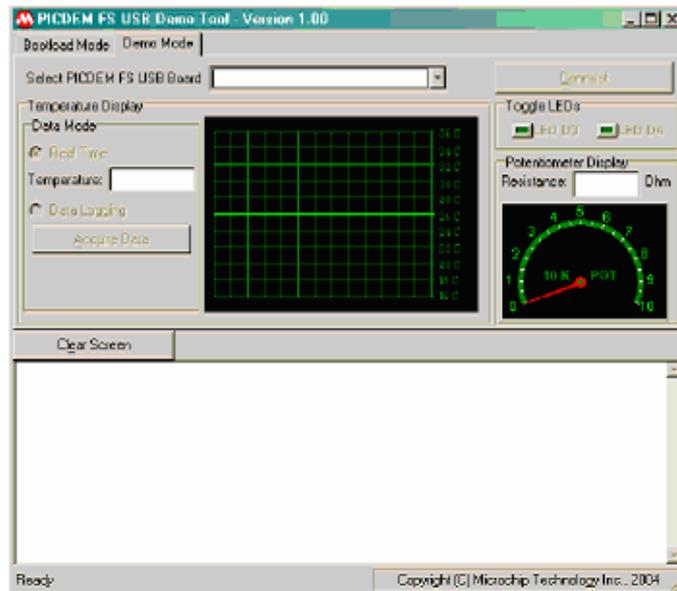
### **3.2. Una aplicación demostrativa para la comunicación USB 2.0, sobre el Kit de desarrollo PICDEM FS.**

La tarjeta PICDEM FS USB permite diseñar una aplicación de software para desarrollar y mostrar las soluciones de software para el control de la comunicación por puerto USB con velocidad máxima de transferencia. Usando este software, se pueden valorar las características de este tipo de comunicación a través del puerto USB y el rendimiento brindado por el microcontrolador PIC18F4550.

Hay dos modos disponibles para el uso de la misma:

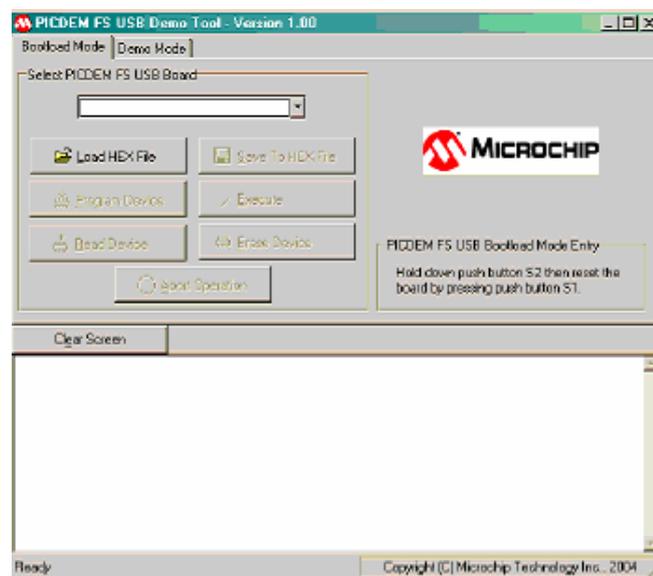
- Modo de presentación del programa demostrativo.
- Modo Bootload como plataforma de desarrollo de nuevas aplicaciones.

El modo demostrativo permite la comunicación de la interfaz electrónica con la PC a través del puerto USB, en una configuración típica del sistema. La aplicación indicada incluye: mostrar las lecturas obtenidas en tiempo real del sensor de temperatura y de la resistencia del potenciómetro, así como ejercer el control de los componentes del dispositivo periférico. El modo de Bootload permite que diferentes usuarios descarguen y valoren un microprograma diferente sin usar un programador externo adicional, además de que pueden crear su propio programa utilizando un programador externo como MPLAB ICD2. Para poner a funcionar la aplicación en el modo de demostración, se ejecuta el programa PDFSUSB.exe el cual mostrará la siguiente ventana que se presenta a continuación:



**FIGURA 7:** Panel Inicial del Programa Demostrativo.

En la figura siguiente se presenta el Panel de Mediciones del Programa demostrativo, donde se indican las lecturas de los parámetros y se ejemplifica la transmisión y recepción de los buffer de datos mediante la comunicación por puerto USB según la especificación 2.0.



**FIGURA 8:** Panel de Mediciones.

### 3.2.1. Programa Demostrativo residente en la PC, para comunicar por puerto USB con la interfaz electrónica basada en la aplicación del PIC18F4550, según la especificación USB 2.0

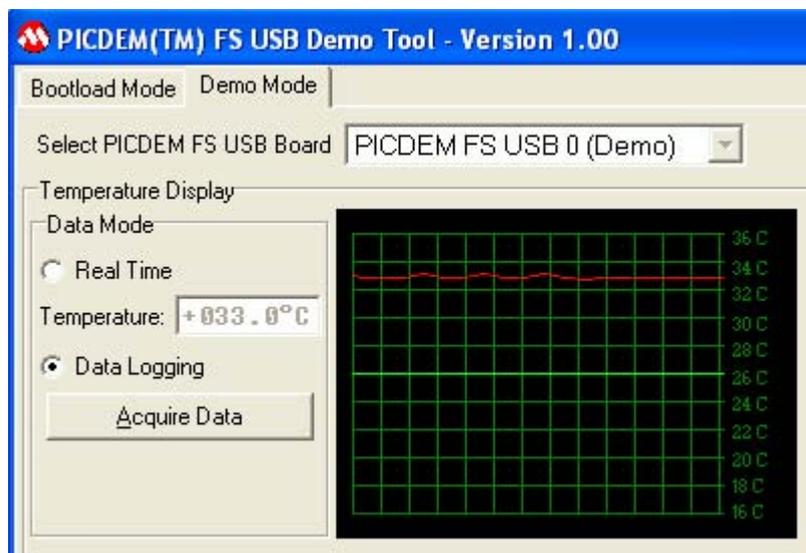
Éste facilita la comunicación entre la tarjeta electrónica y la PC para demostrar la conectividad por puerto USB. Para correr la aplicación, se selecciona " PICDEM FS

de USB 0 (DEMO) " del ""Select PICDEM FS USB Board" y se presiona el botón de "Conexión", cuando se conectan ambos de forma exitosa, aparece el mensaje "USB Demo Firmware Versión 2.0". A continuación se realizan dos ejemplos donde se comprueba la transmisión de un buffer de datos desde la interfaz PIC a la PC a través del puerto USB.

### **Ejemplo No.1: Visualización de temperatura**

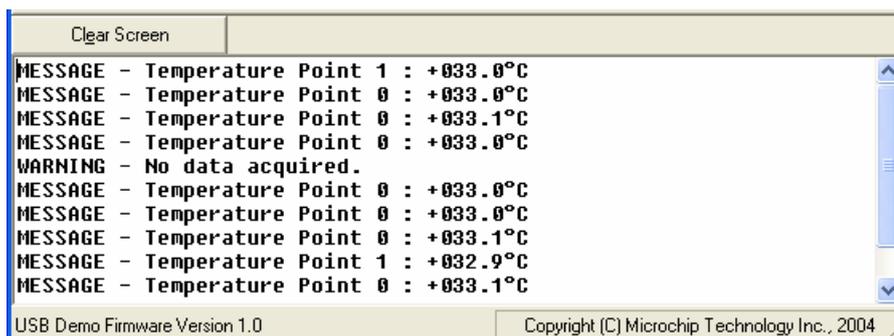
Hay dos modos de mostrar la temperatura:

**1.Mostrar los datos de tiempo real:** En este modo los datos de temperatura son transmitidos de la tarjeta a la PC constantemente. Tanto el gráfico de temperatura como el valor de temperatura se actualizan periódicamente.



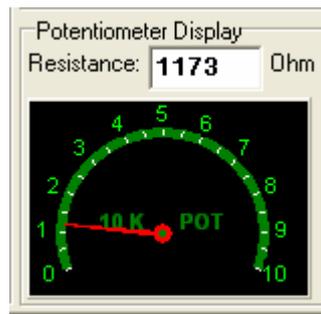
**FIGURA 9:** Panel de medición de temperatura en tiempo real.

**2.Mostrar los últimos datos almacenados:** El sensor de temperatura es muestreado cada 1 segundo y la lectura se guarda en la memoria de datos hasta seleccionar el botón de "Adquirir datos". En ese momento, el microprograma envía todos los datos a la PC. El microprograma almacena hasta 30 datos; por lo que tarda 30 segundos en llenar la memoria del buffer. Las lecturas en tiempo real se muestran en el panel de la **FIGURA 10:**



### **Ejemplo No.2: Visualización del Potenciómetro**

La visualización del Potenciómetro de forma gráfica, refleja el valor actual de la resistencia del potenciómetro de la tarjeta de demostración. Al rotar el potenciómetro en la tarjeta, éste cambia su valor de resistencia, ello se visualiza de forma analógica en el gráfico, y se muestra su valor digital en el monitor de la PC en tiempo real.



**FIGURA 11:** Panel de medición del potenciómetro.

### **Ejemplo No.3: Conmutación de diodos emisores de luz (LED).**

Este panel interactivo que se muestra en la PC contiene dos botones para la conmutación ON/OFF del LED D3 y el LED D4 sobre la tarjeta electrónica. Cada LED correspondiente se enciende cuando se presiona el botón que le corresponde y se apaga cuando el mismo botón es presionado nuevamente. Con este ejemplo se comprueba la transmisión de un buffer de datos desde la interfaz PIC a la PC a través del puerto USB.



**FIGURA 12:** Panel de diodos emisores de luz.

A continuación se describe el Kit de desarrollo, donde se construye esta aplicación.

### 3.2.2. Descripción del hardware del Kit de Desarrollo PICDEM FS USB.

El Kit PICDEM FS USB consiste en una interfaz electrónica basada en el uso del microcontrolador PIC18F4550 y otros componentes, que permiten obtener a modo de ejemplo un sistema de adquisición y procesamiento de datos dotado de la comunicación USB. Además ello permite desarrollar, poner a punto y en ejecución un programa demostrativo elaborado en código fuente en lenguaje C basado en una plataforma de software donde se encuentran programadas algunas de las rutinas necesarias para la comunicación USB para dispositivos clase de interfaz humana o HID, dispositivos de comunicación o CCD y otros periféricos específicos de la empresa Microchip. Estas rutinas facilitan la tarea de configurar y controlar la comunicación por puerto USB mediante funciones disponibles en bibliotecas.

Además para ilustrar demostrativamente la llegada de los paquetes digitales de datos transmitidos y/o recibidos en la comunicación entre el PIC y la PC por puerto USB el KIT de desarrollo adiciona un conjunto de controladores para el sistema operativo Windows y una librería de acceso dinámico **mpusbapi.dll** que permiten la conexión entre las aplicaciones para PC y el microcontrolador, y un ejemplo de programa residente en la PC desarrollado en lenguaje C++ para ilustrar el uso de la librería. En esta investigación se emplea este soporte de software para presentar interactivamente en el monitor de la PC la comprobación del arribo de los paquetes digitales transmitidos desde el PIC, pero esta aplicación también puede ser obtenida mediante la simulación con el Programa ISIS 7, quien ofrece una demostración de esta aplicación.

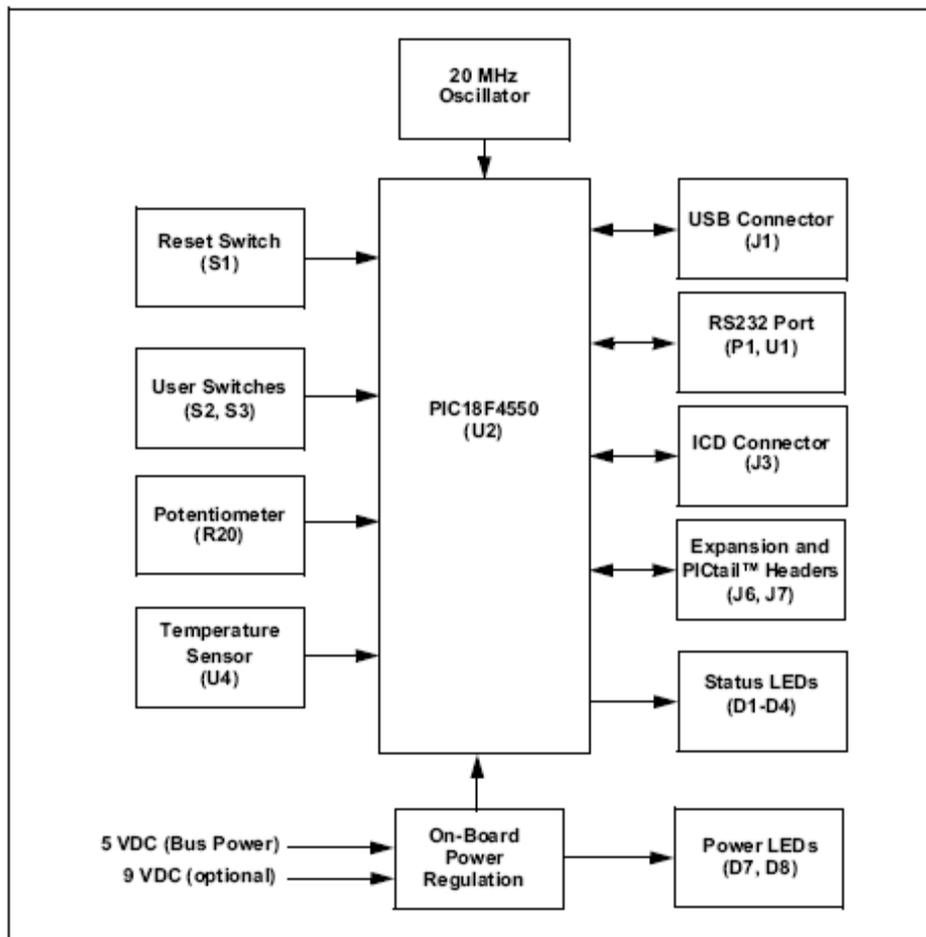
El Kit de desarrollo consta de una placa de circuito impreso con un conector USB en la cual se encuentra el microcontrolador PIC18F4550 el cual es programado con la aplicación de software demostrativo. Además cuenta con una aplicación de software residente en la PC que permite visualizar en la PC los siguientes datos transmitidos en un paquete digital desde el PIC, estos son: la lectura digital de la temperatura captada por el sensor TC77 ubicado en la placa, la lectura del valor de la resistencia según la posición de un potenciómetro, y el estado de la conexión de algunos interruptores y diodos emisores de luz presentes en la placa.

Una de las características funcionales de este Kit de desarrollo es que el PIC18F4550 puede ser fácilmente reprogramado a través del mismo puerto USB con la ayuda de otra aplicación de software residente en la PC que se incluye en esta herramienta de diseño y que no requiere del uso de un programador externo. Esta aplicación pregrabada en la memoria EEPROM del PIC se denomina BOOT LOADER y ocupa

las primeras direcciones de memoria del microcontrolador PIC y prepara a éste para recibir el programa a través del puerto USB y grabarlo en las posiciones restantes de la memoria flash. De lo anterior se infiere que la tarjeta PICDEM FS USB DEMONSTRATION BOARD no es solo una tarjeta de demostración como su nombre lo indica, sino que es un sistema de desarrollo y entrenamiento muy cómodo para el diseñador. La tarjeta PICDEM FS USB posee dos importantes ventajas:

1. Brindar la posibilidad de realizar un sencillo programa demostrativo para la comunicación por puerto USB 2.0 y además la herramienta de diseño para reprogramar la tarjeta con aplicaciones propias de software según las necesidades del diseñador.
2. Ofrece una plataforma expansible que permite ampliar las capacidades de hardware de la tarjeta a través de conectores de expansión para satisfacer las necesidades del diseñador.

Un diagrama en bloques de la tarjeta electrónica se muestra a continuación:



**FIGURA 13:** Diagrama en bloque de la tarjeta electrónica PICDEM FS.

Los principales componentes de la tarjeta son:

1. Microcontrolador PIC18F4550: Con 44 pines según el encapsulado TQFP (U2).
2. Oscilador: Cristal de cuarzo de 20 MHz (Y1) para generar las señales del reloj para el controlador USB y para el microcontrolador.
3. Configuración de los Jumpers para emplear la herramienta de diseño ICD2: Estos tres jumpers deshabilitados permiten que el diseñador elija las líneas de puertos para emplear el depurador/programador ICD2 en su diseño.
4. Conectores de expansión: Permiten acceder a las líneas de puerto de entrada y salida y lograr la expansión para la conexión de otras tarjetas electrónicas y el empleo del Kit una plataforma de prueba y de interfaz de comunicación USB.
5. Potenciómetro: El potenciómetro simula una entrada analógica al conversor A/D y su valor real se muestra como ejemplo en el software de la PC.
6. Sensor de temperatura: Un sensor de temperatura digital de TC77, monitorea la temperatura ambiental de la tarjeta. Esta lectura es transmitida al controlador vía una interfaz SPI y mostrada en el tiempo real mediante el software residente en la PC.
7. Power LED (verde): Estos diodos emisores de luz se encienden para mostrar el suministro de energía a la tarjeta. El LED D7 indica que la tarjeta está suministrando energía del bus mientras que D8 indica que la tarjeta se le está suministrando energía desde un suministro eléctrico distinto.
8. Interruptor: Para efectuar el reset del dispositivo.
9. Power Conector: Conector para proporcionar a la tarjeta un voltaje de (9 VDC) a través de un adaptador externo. Puede también energizarse a través de USB.
10. Conector de USB: Es un usual receptáculo tipo "B" de la serie USB.
11. Conector de ICD: Conector RJ11 que provee la conexión del microcontrolador PIC con la herramienta de diseño ICD2.
12. Conjunto de los LEDs: Un grupo de cuatro LED verdes para mostrar el estado de funcionamiento de la tarjeta, dos de ellos se emplean por el microprograma de aplicación para demostrar el estado de la conexión de USB y los otros (D3 y D4) pueden ser definidos por el diseñador según su aplicación.
13. Dos interruptores: Permiten simular las entradas de control digitales.
- 14 Puerto RS - 232 (DB9F): Provee la comunicación RS232.

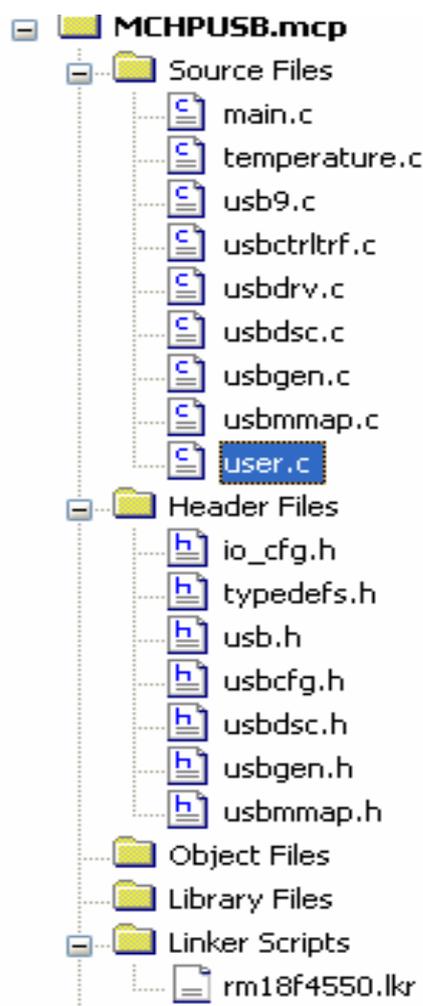
### 3.2.3. Estructura del microprograma del PIC18F4550 para obtener la aplicación demostrativa de la comunicación USB 2.0, con el Kit de desarrollo PICDEM FS.

La estructura del microprograma para la comunicación USB con este microcontrolador de Microchip es un sistema de ficheros que realiza las operaciones para dotar de esta comunicación a las aplicaciones creadas. El proyecto contiene un conjunto de ficheros de microprogramas necesarios para la operación de USB y un fichero reservado para el microprograma de aplicación. Éste está contenido en un directorio raíz con subdirectorios para la organización del código fuente.

Las herramientas de desarrollo y puesta a punto del programa son:

- El software MPLAB IDE 7.5 para simular, errores y programar.
- El software Microchip C18 v 2.30.01 para compilar y ensamblar el programa.

A continuación se muestra el proyecto del programa demostrativo de la comunicación USB en el ambiente del software MPLAB IDE 7.5.



**FIGURA 14:** Estructura de los Ficheros de USB.

El proyecto del programa de la aplicación se denomina MCHPUSB.mcp y tiene además su correspondiente zona de trabajo definida por MCHPUSB.mcw.

El proyecto está constituido por un conjunto de ficheros de código fuente (.C) entre los que se encuentra el fichero **main.c**: contiene la aplicación. La función **main** es un bucle infinito que atiende diferentes tareas, ya sea la comunicación USB o las acciones propias de la aplicación, en este caso contenidas en el fichero **user.c**.

- El fichero **user.c** realiza las operaciones propias del programa demostrativo y que pueden ser cambiadas en función del diseño de nuevas aplicaciones.
- El resto de los ficheros.c y ficheros.h (que se muestra en la figura 14) constituyen la estructura de microprogramas de control de la comunicación por puerto USB, formada por un grupo de microprogramas que implementan de forma modular este tipo de comunicación.

Este directorio contiene tres archivos de la configuración, importantes para el funcionamiento de USB:

- **usbcfg.h**: Configuración global del USB
- **usbdsc.c**: Descriptor del fichero USB
- **usbdsc.h**: Descriptor del fichero cabecera del USB

### **El fichero USBCFG.H**

Este archivo de encabezamiento es muy importante en el proyecto pues define los puntos finales, trazando un mapa de sus funciones de clase. También define el número de documentos de identidad (MUID) de USB, que es usado para determinar la transferencia de control de la clase de USB, y el tamaño máximo de la memoria buffer del dispositivo en el punto final 0. En la configuración del USB de forma centralizada es importante que todos los ficheros del USB relacionados se compilen todos al mismo tiempo. El archivo se construye como una serie de #define de directivas del compilador. Las opciones para la configuración son:  
EPO\_BUFF\_SIZE [buffer\_size], MAX\_NUM\_INT [max], MODE\_PP[\_PPBMn]  
,UCFG\_VAL[option1|option2...],USE\_USB\_BUS\_SENSE\_IO  
USE\_SELF\_POWER\_SENSE\_IO, MAX\_EP\_NUMBER [max\_ep]

- **USB\_USE\_GEN** or **USB\_USE\_CDC** or **USB\_USE\_HID**
- **EPO\_BUFF\_SIZE** Define el tamaño del buffer para el punto final 0. Éste Es válido para los valores de 8, 16, 32 ó 64 bytes. Esta definición es usada globalmente en el proyecto en muchas acciones, tales como: durante la compilación del proyecto para la asignación apropiada del tamaño del buffer para el punto final 0, para el USB descriptor para notificar el USB host del tamaño del punto final 0 y también es usado durante el control de la operación de transferencia.

Cuando en la definición de variables, se configura una baja velocidad de transmisión, el USB puede solo usar 8 bytes en el Buffer, si es Full-Speed o velocidad máxima, el equipo USB puede usar 8, 16, 32 ó 64 bytes en el buffer.

**MAX\_NUM\_INT** [max]: Define el tamaño máximo del número de direcciones que guarda la traza de la interrupción activa para cada interface y que puede cambiarse durante el funcionamiento. Los valores válidos son los enteros que pertenecen al intervalo [0,1, 2...]. Si el equipo tiene múltiples configuraciones, el número máximo de interfaces de una la configuración es el que debe configurarse. Por ejemplo, un dispositivo tiene: dos configuraciones, pero posee tres interfaces en la primera configuración y dos interfaces para la segunda. En este caso, el parámetro MAX\_NUM\_INT debe ser tres.

**MODE\_PP [\_PPBMn]**: Define el modo buffer para ser usado durante el tiempo de la corrida (runtime). Los valores permitidos son \_PPBM0, \_PPBM1 y \_PPBM2; cada uno de éstos se define a su vez en el fichero usbdv.h. La función de cada modo se explica en el fichero de las características del PIC18F4550. La versión actual del firmware sólo permite el modo de \_PPBM0.

**UCFG\_VAL [option1 | option2...]**: Define el valor inicial por el registro UCFG de la función especial. Los valores permitidos son:

- \_LS or \_FS: Selección del modo de velocidad baja o alta.
- \_TRINT or \_TREXT: Selección del modo de Transferencia Externo o Interno.
- \_PUEN: Uso interno USB pull-up resistencia
- \_OEMON: Uso del indicador de salida del bloque SIE
- \_UTEYE: Habilita la salida de la señal o patrón de prueba para evaluar el proceso.

Las opciones pueden ser al mismo tiempo ORed (Por ejemplo #define UCFG\_VAL \_PUEN|\_TRINT|\_FS). Este valor se usa por el firmware para inicializar el registro de UCFG. Una explicación completa de las opciones pueden encontrarse en la descripción del registro de UCFG dada en el capítulo anterior.

**USE\_SELF\_POWER\_SENSE\_IO**: Indica que el microcontrolador está detectando la presencia de energía en la tarjeta a través del pin de I/O (entrada y salida). Si el diseño de la aplicación no usa el pin de I/O para detectar la presencia de la energía en la misma (Self-Power), ésta se obtiene por conexión externa.

**USE\_USB\_BUS\_SENSE\_IO**: Indica que los firmware usa el pin definido en el fichero io\_cfg.h para determinar cuándo se habilitó el módulo de USB. Si el diseño de la aplicación no usa un pin de I/O para detectar la presencia del bus de USB, esta definición debe hacerse externa.

Cuando `USE_USB_BUS_SENSE_IO` es indefinido, el módulo de USB siempre será habilitado. Tener esta opción ayuda a mejorar la eficiencia de la energía del sistema porque el módulo de USB solo se habilita cuando el bus está presente.

**USB\_USE\_GEN, USB\_USE\_CDC and USB\_USE\_HID** se usa para indicar qué clases de USB deben ser incluidas en el código del proyecto. Cuando cada una de estas clases se definen, se comunica al fichero global USB (`usb.h`) cuáles son las clase-específicas presentes y que están en los ficheros `.h` que deben incluirse en el proyecto. El fichero `usb.h` se usa globalmente como un requisito pues incluye los ficheros a usar en la biblioteca de USB. Si la clase HID es usada, entonces los ficheros `hid.c` y `hid.h` también deben agregarse al proyecto en la corrida de la herramienta MPLAB 7.5. Si la clase CDC es usada, entonces los ficheros `cdc.c` y `cdc.h` también se adicionan al proyecto en la corrida de MPLAB 7.5.

Esta definición es también usada cuando se recibe una petición de `SET_CONFIGURATION` y puede ser modificada en la función `USBStdSetCfgHandler` del fichero `usb9.c`. Las llamadas localizadas en la sección del modificable inicializan los puntos finales (endpoints) empleados para cada clase específica de USB y éstas son mapeadas por sus trazas en el fichero `usbcfg.h`.

**MAX\_EP\_NUMBER [max\_ep]:** Es el número mayor asignado a un punto final (endpoint) en el proyecto. Por ejemplo, si el número del punto final (endpoint) más alto usado es el punto final 5, entonces `MAX_EP_NUMBER` debe ser igual a cinco. Esta definición se usa principalmente en el fichero `usbmmap.c` para asignar los registros de la descripción del buffer.

### **Los ficheros USBDSC.C y USBDSC.H**

Estos archivos contienen la información del descriptor de USB para el dispositivo, éstas son: la estructura, el tamaño y la configuración. El usuario puede cambiar la estructura del descriptor en el fichero `usbdsc.h` que se usa para calcular el descriptor, clasificarlo según el tamaño con el parámetro `sizeof ()` en la declaración. Cuando un descriptor se agrega o se elimina de la configuración principal, el usuario también debe cambiar la estructura del descriptor definida en el fichero `usbdsc.h`.

Un descriptor de la configuración típico consiste en estos componentes:

- Por lo menos un descriptor de la configuración (`USB_CFG_DSC`)
- Uno o más descriptores de la interface (`USB_INTF_DSC`)
- Uno o más descriptores del endpoint (`USB_EP_DSC`)

- **Personalizando USB DSC.C**

En el fichero `usb dsc.c` se incluye la estructura del firmware donde pueden definirse los parámetros, y sirven de plantilla para desarrollar el nuevo dispositivo. Los siguientes parámetros son explicados a continuación:

- **USB\_CFG\_DSC** : El atributo de la configuración debe siempre tener como mínimo la definición de `_DEFAULT`. Dos opciones adicionales, `_SELF` y `_RWU` son: `ORed` con `_DEFAULT`. `_SELF` que informa al organizador de USB que este dispositivo es auto energizado, mientras `_RWU` le dice al organizador de USB que el dispositivo es inicializado externamente (Remote Wakeup). Las definiciones para estas opciones se proporcionan en el fichero `usbdefs_std_dsc.h`.

- **USB\_EP\_DSC**: Es un descriptor del punto final (endpoint= que tiene una forma similar a: `sizeof(USB_EP_DSC),DSC_EP,_EP01_OUT,_BULK,64,0x00,`

Los primeros dos parámetros especifican la longitud de este descriptor del endpoint (7) y los tipos de descriptors. El próximo parámetro identifica el endpoint. Toma el formato: `_EP <##>_ <EL DIR>` donde `##` es la numeración del endpoint y el `dir` es la dirección de traslado. Es decir `dir` tiene el valor de cualquiera 'OUT' o 'IN'. Las definiciones se proporcionan en el fichero `usbdefs_std_dsc.h`.

El próximo parámetro identifica el tipo del endpoint. Las opciones disponibles son `_BULK`, `_INT`, `_ISO` y `_CTRL` (para el Volumen, Interrupción, isócrono y Control de endpoints, respectivamente). El `_CTRL` no se usa típicamente porque el valor por defecto es el endpoint de traslado de control, y no se define en el descriptor de USB. Cuando `_ISO` es usado, las opciones adicionales pueden ser `ORed` con él. Por ejemplo: `_ISO |_AD |_FE` describe el endpoint como un flujo consecutivo isócrono con adaptabilidad y regeneración de los atributos. Se puede observar en el fichero `usbdefs_std_dsc.h` la especificación en detalles para la USB. Los últimos parámetros definen el tamaño máximo del endpoint y registran los intervalos.

- **Descripción del USB String**

En lugar de una cadena simple de texto, el descriptor se estructura en una forma de datos particular. Los descriptors de la cadena toman el formato:

El rom struct { byte bLength;byte bDscType;word string[size]; } sdxxx= {el sizeof(sdxxx),DSC\_STR,<text>};

Esta estructura permite al compilador de C calcular la longitud de la cadena `sdxxx` del descriptor donde el `xxx` es el número de índice de la cadena. Los primeros dos bytes de los descriptors son la longitud y el tipo del descriptor.

- **El fichero usbdsc.h**

En el fichero usbdsc.h, las variables para cada uno de los componentes del descriptor son nombradas con las convenciones siguientes:

- Se nombran los tipos de USB\_CFG\_DSC cdxx, dónde xx es el número de la configuración. Este número debe ser compatible o similar al valor del índice real de la configuración.
- Se nombran los tipos de USB\_INTF\_DSC i <el yy>a <el zz>, dónde yy es el número de la interface y el zz es el número de la interface alternado.
- Se nombran los tipos de USB\_EP\_DSC el ep <##> <el d>\_i <el yy>a <el zz>, dónde ## es el endpoint el número y d es la dirección de traslado. El nombre de la interface también debe listarse como un sufijo para identificar a qué interface de endpoint pertenece.

- **El fichero USBMMAP.C**

Este archivo es el gestor de la memoria para la comunicación USB; el mismo asigna la memoria durante el tiempo de compilación para cada uno de los destinos finales de las transferencias de datos. Los puntos finales de la comunicación son definidos mediante direcciones y la enumeración de éstos. El fichero usbmmmap.c se encarga de asignar y registrar las direcciones y la numeración de éstos.

En el USB el gestor de la memoria usa MAX\_EP\_NUMBER, como está definido en el fichero usbcfg.h, se definen los puntos finales para crear las instancias. Esto representa que los endpoint con los números mayores se asignan, cuándo los endpoints se usan. Los BDTs para el endpoints tiene las direcciones hardware asignadas en Banco 4 de la RAM, si se asigna un valor demasiado alto puede llegar a ser ineficiente el uso de RAM de los datos.

- **El fichero USBDRV.C**

Este archivo provee servicios de comunicación USB de bajo nivel de prioridad, y maneja todas las interrupciones de este tipo. Mediante la función de USB CheckBusStatus se detecta en el terminal de entrada salida la solicitud de comunicación, además permite desactivar el módulo de USB para incrementar en consecuencia la eficiencia del motor del sistema.

- **El fichero USBCTRLTRF.C**

Este archivo efectúa la transferencia de control, y provee servicios a todas las clases de USB que tienen que manejar los pedidos de USB. Debido a que el servicio de transferencia de control es compartido entre muchas clases de USB y la mayoría de los traslados crecen sobre las transacciones múltiples, es importante estar al tanto

de qué clase USB posee la acción de transferencia de control en curso. Cada transferencia de control tiene tres etapas: la instalación, los datos y el estado.

Si se trata de una transferencia del tipo de instalación, la función `USBCtrlTrfSetupHandler` revisa el paquete de configuración en tres pasos:

1. Inicializar la rutina que controla la transferencia.
2. Ejecuta la rutina de procesamiento de la transferencia según la clase del dispositivo USB que la solicita.
3. Mediante la rutina `USBCtrlEPServiceComplete` se verifica la dirección del punto final de la transferencia de los datos.

### **USBCtrlEPService()**

Es llamado por el fichero `usbdrv.c`. Sólo procesa las transacciones que pasen por endpoint 0. Verifica para el tipo de la transacción, si es diferente llama al apropiado controlador según el tipo de transacción (Endpoint 0 SETUP, Endpoint 0 OUT or Endpoint 0 IN). Si la fase del traslado es el SETUP, entonces actúa `USBCtrlTrfSetupHandler()` que se llama para procesar el paquete del arreglo. Cada USB controla que el paquete de arreglo de traslado siempre sea 8 bytes. Hay tres pasos para procesar un paquete del arreglo:

1. La rutina inicializa la transferencia de control de estado a la máquina.
2. Llama entonces en cada una de las clases de los controladores de la demanda específica, en un esfuerzo por encontrar una solución para ella, si algún controlador conoce como procesar la demanda.
3. Una vez que todos los controladores de la demanda han tenido la oportunidad para analizar el paquete del arreglo, la función `USBCtrlEPServiceComplete()` chequea la dirección del traslado de los datos organizadamente para determinar cómo procesar el endpoint 0.

### **USBCtrlEPServiceComplete()**

Incluye tareas restantes para procesar un arreglo de paquete. Su tarea principal es poner el endpoint controlando apropiadamente para una situación dada. Existen 3 posibles resultados: Endpoint 0 se detiene si la demanda no puede procesarse, Endpoint 0 se inicializa para la transferencia de los datos al host durante la etapa de los datos y Endpoint 0 se inicializa para recibir datos desde el host durante la etapa de datos.

La etapa fase de los datos puede extenderse para múltiples transacciones de USB. Es importante guardar la traza de la fuente de los datos, el destino de los datos, el

conteo de los datos y el tipo de los datos. Éstos se enrutan y se actualizan usando cuatro variables especializadas:

- pSrc es puntero del origen del dato
- pDst es puntero del destino del dato
- wCount es el contador de los datos
- Es el dato tipo booleano o tipo bits denominado usb\_stat.ctrl\_trf\_mem que puede tener el valor de \_ROM o \_RAM según sea el tipo de memoria a usar.

Al notificarse o escribirle a un controlador la demanda de envío de datos de traslado de control de un dispositivo periférico a un host, se realizan los siguientes pasos para asegurar esto:

**1-Direccionar la fuente de los datos:** Para las localizaciones de RAM: pSrc.bRam = <Localización para buffer de la RAM >; Para las localizaciones de ROM: pSrc.bRom = <Localización de memoria de programa>;

**2-El tipo de memoria de los datos:** usb\_stat.ctrl\_trf\_mem = <\_ROM or \_RAM>;

**3-Poner el tamaño de datos a transferir:** wCount.\_word = < tamaño de datos para transferir >;

**4- Dar la dirección de destino de datos,** para ello se ocupa automáticamente el fichero usbctrltrf.c; el destino de CtrlTrfData es el espacio del buffer definido en el fichero usbmmap.c

Al notificarse o escribirle a un controlador la demanda de recepción de datos de traslado de control de un dispositivo periférico a un host con múltiples transacciones, se deben realizar los siguientes pasos para asegurar esto:

**1-Direccionar el destino de los datos:** pDst.bRam = <Localización para el buffer de datos en la RAM >

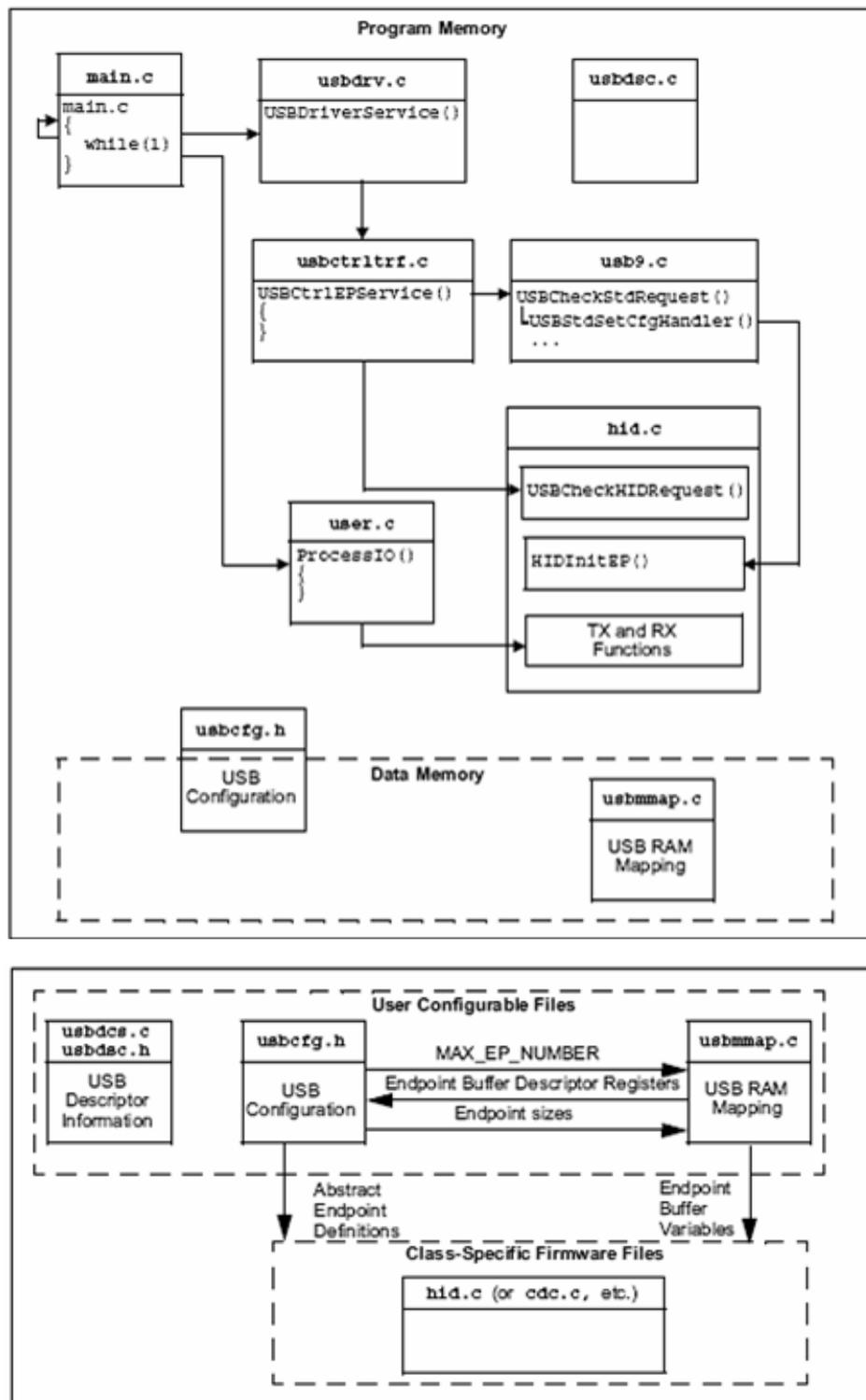
**2-Inicializar el contador de los datos recibidos,** wCount.\_word, en cero.

**3-Dar la dirección de destino de datos,** de lo que se ocupa automáticamente el fichero usbctrltrf.c; el destino de CtrlTrfData es el espacio del buffer definido en el fichero usbmmap.c

#### • El fichero USB9.C

Este archivo maneja los pedidos de USB según la especificación de USB, y realiza el proceso de enumeración, la función USBStdSetCfgHandler() puede ser modificada para corresponder a la configuración y el comportamiento de la aplicación de los usuarios.

En la figura 15 se muestran las relaciones entre los ficheros de código fuente que configuran y controlan la comunicación por puerto USB., cuyo funcionamiento se describe a continuación



**FIGURA 15:** Relaciones entre los ficheros de código fuente que configuran y controlan la comunicación por puerto USB.

Las tareas de la comunicación USB son controladas por la función `USBDriverService()`, que detecta y ejecuta las interrupciones de este tipo y efectúa acciones de control según el protocolo de intercambio de datos de control descrito en la especificación de USB. El servicio de transferencia de control es proveído por las funciones del fichero `usbctrltrf.c`.

Cuando una transacción es recibida se corre la función `USBCtrlEPService()`. Esta función chequea y da servicio a tres tipos de transacciones: de entrada, de salida y de instalaciones. La función `USBCtrlTrfSetupHandler()` realiza las siguientes funciones: inicializa el control de la transferencia, clasifica y atiende el tipo de transferencia según el dispositivo que la solicita. Se tienen tres diferentes etapas para el control de la transferencia, éstas son: transmisión, recepción y espera. Además se tiene en cuenta la ruta de los datos y la cantidad de datos.

El proceso de enumeración de los dispositivos USB es tratado en `usb9.c`. Uno de los pasos más importantes en el proceso de enumeración es el manejo de un pedido, que es hecho por la función `USBStdSetCfgHandler()`. Esta función puede ser modificada por el usuario para llamar a otras funciones para inicializar los puntos finales de la aplicación.

En la aplicación usuario incluido en el programa principal se tiene la función `Process()` que facilita que ésta pueda transmitir y recibir una transacción USB, mediante una llamada a las funciones `HIDRxReport()` y `HIDTxReport()` que son específicas para cada microprograma. La configuración del dispositivo USB se controla de forma modular, modificando variables en alguno de estos archivos; la información está disponible durante el tiempo de compilación para crear la configuración de USB.

A continuación vamos a presentar los resultados obtenidos en el desarrollo personalizado y la construcción de un sistema automatizado de adquisición de datos con comunicación por puerto USB, según la especificación USB2.0, el cual se obtiene empleando el PIC18F4550 y la tarjeta electrónica que brinda el Kit de desarrollo PICDEM FS.

Esta aplicación consta de los siguientes aspectos:

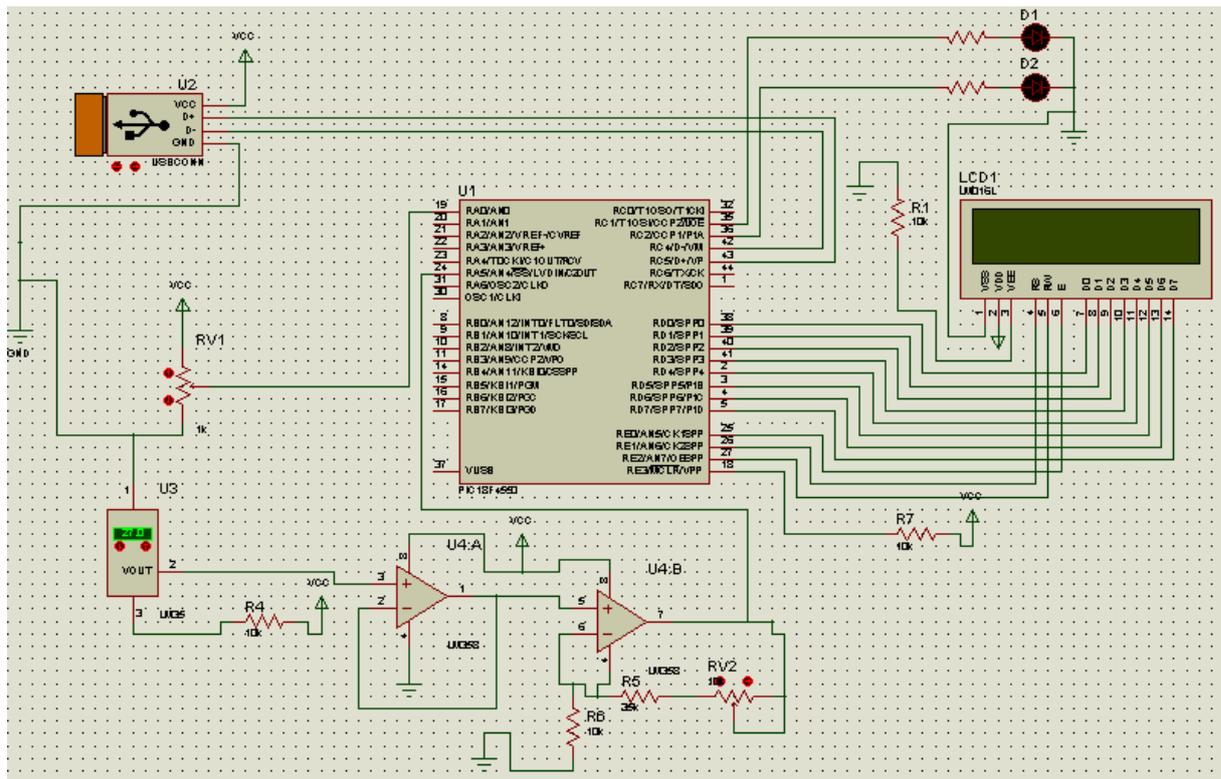
- 1) Las conexiones de Hardware del Sistema sobre el KIT.
- 2) Un Microprograma elaborado en código fuente en lenguaje C, el cual es ensamblado por el MPLAB 7.5 y puesto a punto utilizando la simulación lógica con el Proteus ISIS en combinación con las herramientas MPLAB e ICD2 para depurar y grabar el código en el PIC.

3) Un programa residente en la PC para controlar la comunicación y demostrar la comunicación entre la interfaz PIC del Sistema y la PC, éste se ha desarrollado en lenguaje Delphi y emplea la **picusbci.dll**. Para esta aplicación, además se tiene el fichero **Driver** (con sus respectivos valores de **pid** y **vid**) para el reconocimiento de la conexión de la tarjeta electrónica en el puerto USB de la PC.

### 3.2.4. Descripción de la aplicación desarrollada para el sistema.

- **Descripción del Hardware**

Esta aplicación se simula con el Proteus ISIS, teniendo el siguientes esquema eléctrico de la figura 16, donde se aprecia el uso de 2 sensores conectados al Conversor A/D del PIC, un indicador LCD alfanumérico de 16 caracteres x 2 renglones, y además consta de 2 actuadores representados por un diodo emisor de luz (led) de color rojo y otro de color amarillo. Además se tiene la conexión del PIC18F4550, con un cristal de cuarzo de 20 MHz y con el conector para la comunicación USB.

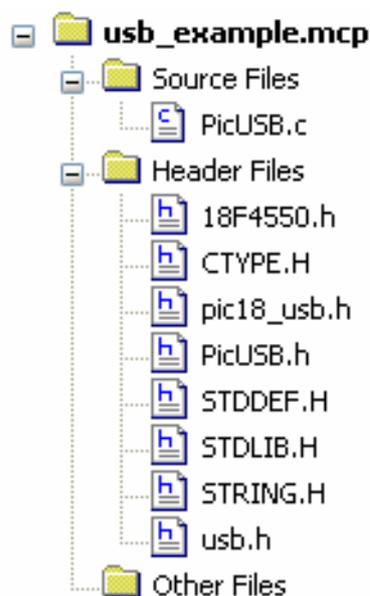


**FIGURA 16:** Circuito eléctrico de la simulación en el Proteus.

Este hardware se construye en la práctica empleando la tarjeta electrónica del Kit de desarrollo, en la zona de expansión de ésta, empleando conectores Panduit en su construcción. Además la tarjeta se alimenta de la propia PC a través del puerto USB y posee la posibilidad de conectar la herramienta de depuración y programación ICD2 para ajustar y poner a punto el sistema en la práctica.

- **El Microprograma de la aplicación.**

La simulación lógica del hardware es exitosa y permite comprobar que es válido el funcionamiento del programa almacenado en el microcontrolador PIC y el programa residente en la PC, para obtener la comunicación por puerto USB según la especificación 2.0. El microprograma PIC consta de un Proyecto USB\_example.mcp, con los siguientes ficheros:



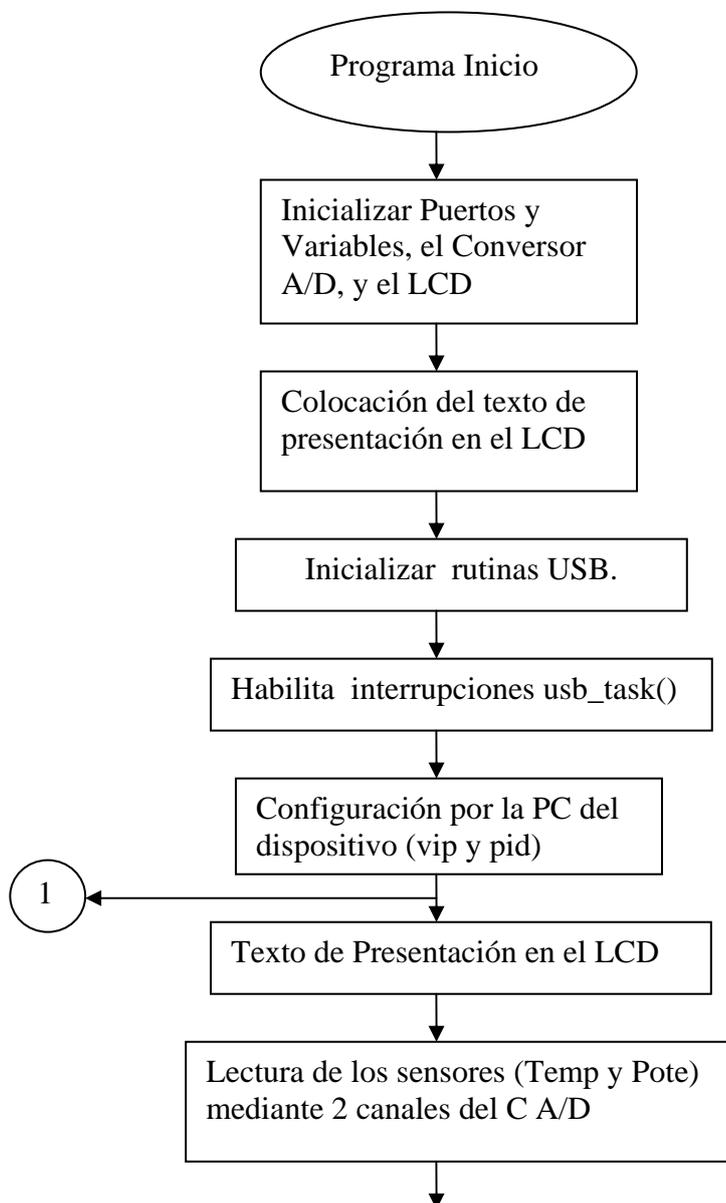
Los ficheros: STDEF.H, STDLIB.H, STRING.H, CTYPE.H pertenecen a la biblioteca de funciones de Microchip para el manejo de variables de cadenas de caracteres y otras aplicaciones. El fichero 18F4550.h es el fichero del dispositivo microcontrolador que da Microchip.

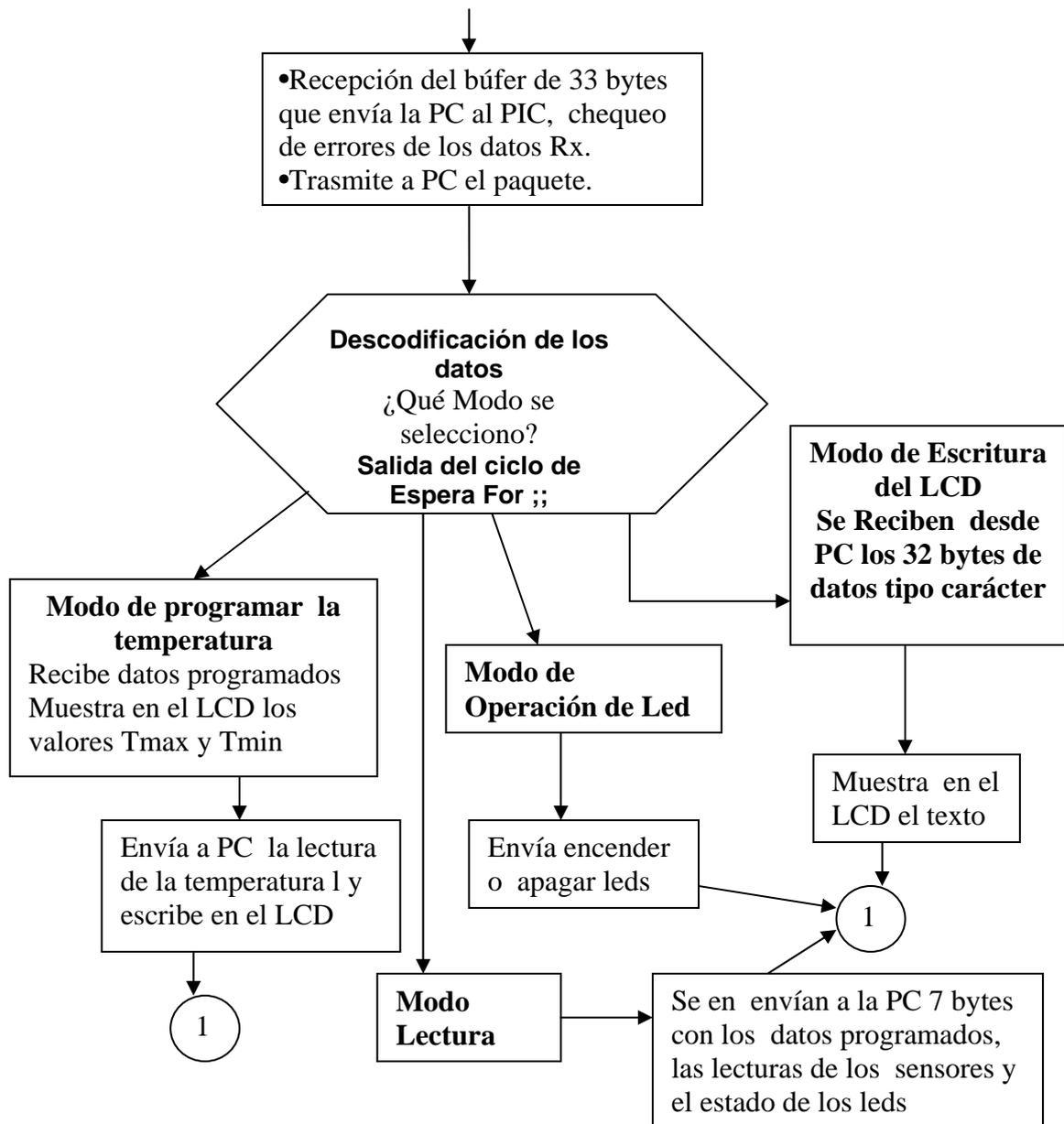
El fichero PICUSB.h, permite configurar el dispositivo USB y sus descriptores. En el mismo, se realiza en la rutinas **start device descriptors** y **start string descriptors**, la definición de las cadenas de caracteres que identifican: el valor del **vid** (firma fabricante) y del **pid** (nombre del dispositivo o producto), lo cual personaliza la aplicación.

Los ficheros: pic18\_USB.h, USB.c y usb.h, contienen un conjunto de funciones y configuraciones usadas por el compilador CCSC para controlar la comunicación USB, éstas son:

- 1) **usb\_init()** y **usb\_init\_cs()** : Inicializa la pila USB, periféricos y las interrupciones.
- 2) **usb\_put\_packet()** y **usb\_puts()**: Envían datos a la PC de 1 byte o más.
- 3) **usb\_kbhit()** – Retorna un 1 si el dispositivo USB de salida contiene datos enviados desde la PC.
- 4) **usb\_get\_packet()**: Obtiene un paquete de datos enviados desde la PC.
- 5) **usb\_gets()**: Obtiene un paquete de datos enviados desde la PC.
- 6) **usb\_task()**: Realiza la conexión USB, emplea las rutinas: **usb\_detach()** y **usb\_attach()**.

El fichero PicUsb.c contiene la aplicación desarrollada. A continuación se presenta en la figura 17, el **Diagrama en Bloques del Programa**, donde se halla el Protocolo de Comunicación entre la interfaz PIC y la PC





**FIGURA 17:** Diagrama en Bloques del Programa

Un ejemplo del código fuente se describe a continuación:

Descripción	Segmentos del Código Fuente
<p>Inicio del programa principal. Inicialización del conversor, del LCD, y se muestra un texto en el LCD al inicio</p>	<pre>void main (void) {     set_tris_d(0x00);//todo salida     set_tris_e(0x00);//todo salida     bit_clear(tric,1);// c1 es salida     trie=0; trid=0;     adc_init ();     lcd_init(); //inicio del LCD     lim();     demora(500);     pos(1);     printf(tolcd,"PROGRAMA USB2.0 ");     pos(17);     printf(tolcd,"TESIS MSc UPR ");     demora(3000); }</pre>
<p>Inicialización del USB</p>	<pre>//////////////////////////////////USB////////////////////////////////// usb_init();           //inicializamos el USB usb_task();           //habilita periférico USB e interrupciones usb_wait_for_enumeration(); //esperamos hasta que el PicUSB sea configurado por el host</pre>
<p>Aplicación principal lecturas del los canales del C A/D para potenciómetro y temperatura  Rx del paquete de 33 bytes que envía PC al PIC. Con los diferentes Modo de trabajo. <b>Modo de Programación</b></p>	<pre>////////////////////////////////// el principal////////////////////////////////// for (;;) {     pos(1);    printf(tolcd,"PROGRAMA USB2.0 ");     pos(17);   printf(tolcd,"TESIS MSc UPR ");     //temperatura     delay_ms(10);     set_adc_channel (4); ///estaba el 0 cogiendo el valor del pote     delay_us(100);     lect_adc = read_adc ();     delay_ms(10);     temperatura= lect_adc;     //pote     delay_ms(10);     set_adc_channel (0);     delay_us(100);     lect_adc = read_adc ();     delay_ms(10);     pote = lect_adc;      if(usb_enumerated()) //si el PicUSB está configurado     {         if (usb_kbhit(1)) //si el endpoint de salida contiene datos del host         {             af=0;             z: recibe[af]=32;             Te[af]=32;             af++;             if(af!=34) goto z;         }     }      //cogemos el paquete de tamaño 33 bytes del EP1 y almacenamos en recibe     usb_gets(1, recibe, 33, 1000);      if (modo == 0) //Modo_Programación     {         TMAX=param1;         TMIN=param2;         envia[0]=1;         resultado=temperatura;         usb_put_packet(1, envia, 1, USB_DTS_TOGGLE); //enviamos el paquete de</pre>

<p>Confirma a la PC la operación</p>	<pre>tamaño 1byte del EP1 al PC lim(); pos(1);    printf(tolcd,"TempMax=%U",param1); pos(17);   printf(tolcd,"TempMin=%U",param2); param1=0;  param2=0;  resultado=0; DEMORA(5000); lim(); }</pre>
<p><b>Modo de Leds</b></p>	<pre>if (modo == 1) //Modo_Led { if (param1 == 0) { output_low (LEDA); output_low (LEDR); } //apagamos los leds  if (param1 == 2) { output_low (LEDA); output_high (LEDR); } //encendemos led rojo if (param1 == 3) { output_high (LEDA); output_high (LEDR); } //encendemos ambos leds if (param1 == 4) { output_low (LEDR); output_high (LEDA); } //encendemos led amarillo</pre>
<p>Muestra en LCD</p>	<pre>pos(1); printf(tolcd,"COMUNICACION PC"); pos(17); DEMORA(5000); DEMORA(5000); lim(); param1=0; }</pre>
<p><b>Modo de LECTURA</b></p>	<pre>if (modo == 6) //Modo ADC { enviat[0]=0; enviat[1]=0; enviat[2]=0; enviat[3]=0; enviat[4]=0; //enviat[4] //temperatura enviat[0] = temperatura; //pote enviat[1] = pote; if(IED1==1) enviat[2]=1; //enviat[2] if(IED2==1)enviat[3]=1; //enviat[2] enviat[4]=TMAX; enviat[5]=TMIN; enviat[6]=1; usb_puts(1, enviat, 7, 550); pos(1); printf(tolcd,"Lectura del PIC "); aff=0;</pre>
<p>Confirma a la PC la operación Muestra en LCD <b>Modo LCD</b></p>	<pre>aff=0; }</pre>

<p>Confirma a la PC la operación Muestra en LCD</p>	<pre> if (modo == 7) //Modo LCD ojo     esultado= 1;     //if (recibe[17]==2) resultado=1;     usb_put_packet(1, envia, 1, USB_DTS_TOGGLE); //enviamos el paquete de tamaño 1byte del EP1 al PC     aj=1;     ty: pos(aj);     printf(tolcd,"%C",recibe[aj]);     aj++;     if (aj&lt;17) goto ty;      DEMORA(5000);     //////////////////////////////////////     aj=17;     ty1: pos(aj);     printf(tolcd,"%C",recibe[aj]);     aj++;     //DEMORA(5000);DEMORA(500);     if (aj&lt;33) goto ty1;         } </pre>
<p><b>Modo Inicio</b> Confirma a la PC la operación Muestra en LCD</p>	<pre> if (modo == 8) //Modo INICIO     {         lim();         TMAX=25;         TMIN=15;         output_low (LEDA);         output_low(LEDR);         resultado= 2;         usb_put_packet(1, envia, 1, USB_DTS_TOGGLE);         DEMORA(5000);DEMORA(5000); DEMORA(5000);DEMORA(5000);     } } } } } } </pre>

- **El programa NEVIS residente en la PC para esta aplicación**

El protocolo de comunicación que se establece entre la PC y la interfaz PIC ya descrito anteriormente requiere que el programa desarrollado en Delphi cumpla satisfactoriamente con el intercambio de señales entre ambos equipos, la transmisión y la recepción de buffer de datos con una configuración propia y el chequeo de errores durante el proceso de comunicación. A continuación presentamos fragmentos del código fuente desarrollados para esta aplicación y se comentan algunas de estas acciones.

Para establecer la comunicación se utilizó la función **GetSummary()**, que lleva implícito la api de Windows MPUSBAPI.DLL la cual utiliza varias funciones, como son:  
MPUSBGetDLLVersión para ver la versión de la dll  
MPUSBGetDeviceCount; Lo que es el vid y pid  
MPUSBOpen: Abre el puerto USB

MPUSBRead: Lectura de puerto USB

MPUSBWrite: Escribir el puerto USB

MPUSBReadInt: Lectura de USB de enteros

MPUSBClose: Cerrar puerto USB

En la siguiente tabla se muestran segmentos de código de la programación de dos funciones, donde se emplean las diferentes funciones del api para lenguaje Delphi:

Descripción	Segmentos del Código Fuente
Inicialización de la función GetSummary (); para comprobar si ya un USB con vid y pid para empezar la Tx y Rx	<pre>procedure TfrmUsbTest.btnGetSummaryClick(Sender: TObject); begin   Button2.SetFocus;   timer1.Enabled:=False;   GetSummary(); end;  function TfrmUsbTest.GetSummary():Integer; var   tempPipe:THandle;   count:DWORD;   max_Count:DWORD;   i:Byte; begin   tempPipe := INVALID_HANDLE_VALUE;   count:=0;   max_count := MPUSBGetDeviceCount(vid_pid);   if(max_count=0) then   begin     result:= max_count;     Memo1.Lines.add('No hay dispositivo conectado');     exit;   end   else     memo1.lines.add(IntToStr(max_Count) + ' dispositivo(s) con ' + vid_pid + ' actualmente conectado');     count := 0;     for i:=0 to MAX_NUM_MPUSB_DEV-1 do       begin         tempPipe := MPUSBOpen(i,vid_pid,NIL,MP_READ,0);         if(tempPipe &lt;&gt; INVALID_HANDLE_VALUE) then           begin             memo1.lines.add('Indice de instancia ' + IntToStr(i));             MPUSBClose(tempPipe);             Inc(count);           end;         if(count = max_count) then break;       end;     result:= max_count;   end;</pre>

<p>Esta es para enviar al PIC un byte y recibir la confirmación de que llegó con éxito en el PIC</p>	<pre> function TfrmUsbTest.SendReceivePacket(SendData: PByteBuffer;     SendLength: DWORD;     ReceiveData: PByteBuffer;     var ReceiveLength: DWORD;     SendDelay: Word;     ReceiveDelay:Word):DWORD;  var     SentDataLength: DWORD ;     ExpectedReceiveLength: DWORD; begin     ExpectedReceiveLength := ReceiveLength;     if(myOutPipe &lt;&gt; INVALID_HANDLE_VALUE) and ( myInPipe &lt;&gt; INVALID_HANDLE_VALUE) then         begin             if MPUSBWrite(myOutPipe,SendData,SendLength,@SentDataLength, SendDelay) &lt;&gt; 0 then                 begin                     if(MPUSBRead(myInPipe,ReceiveData, ExpectedReceiveLength,@ReceiveLength,ReceiveDelay)) &lt;&gt; 0 then                         begin                             if (ReceiveLength = ExpectedReceiveLength) Then                                 begin                                     result:=1;                                     exit;                                 end                             else                                 begin                                     if (ReceiveLength &lt; ExpectedReceiveLength) then   begin   result:=2;   end;   end                                     end                                 end                             else                                 CheckInvalidHandle();                             end                         else                             CheckInvalidHandle();                     end;                     result:=0;                 end;             end;         end;     end; </pre>
--	--

### 3.2.5. Descripción de la aplicación desarrollada en Delphi.

La tarjeta PICDEM FS USB permite desarrollar y diseñar diferentes aplicaciones para comunicación por puerto USB con velocidad máxima de transferencia, desarrollando un protocolo propio de comunicación. Para desarrollar y mostrar las

soluciones de software hechas se elaboró una aplicación desarrollada en Delphi para el control de un sistema. Usando este software, se pueden valorar las características de este tipo de comunicación a través del puerto USB y el rendimiento brindado por el microcontrolador PIC18F4550. Este provee una interfaz simple para la comunicación entre la tarjeta electrónica y la PC para demostrar la conectividad de USB. A continuación se describe el funcionamiento del programa NEVIS, donde se comprueba la transmisión y recepción de un buffer de datos desde la interfaz PIC a la PC (o viceversa) a través del puerto USB.

### 3.2.6. Descripción del funcionamiento del programa demostrativo.

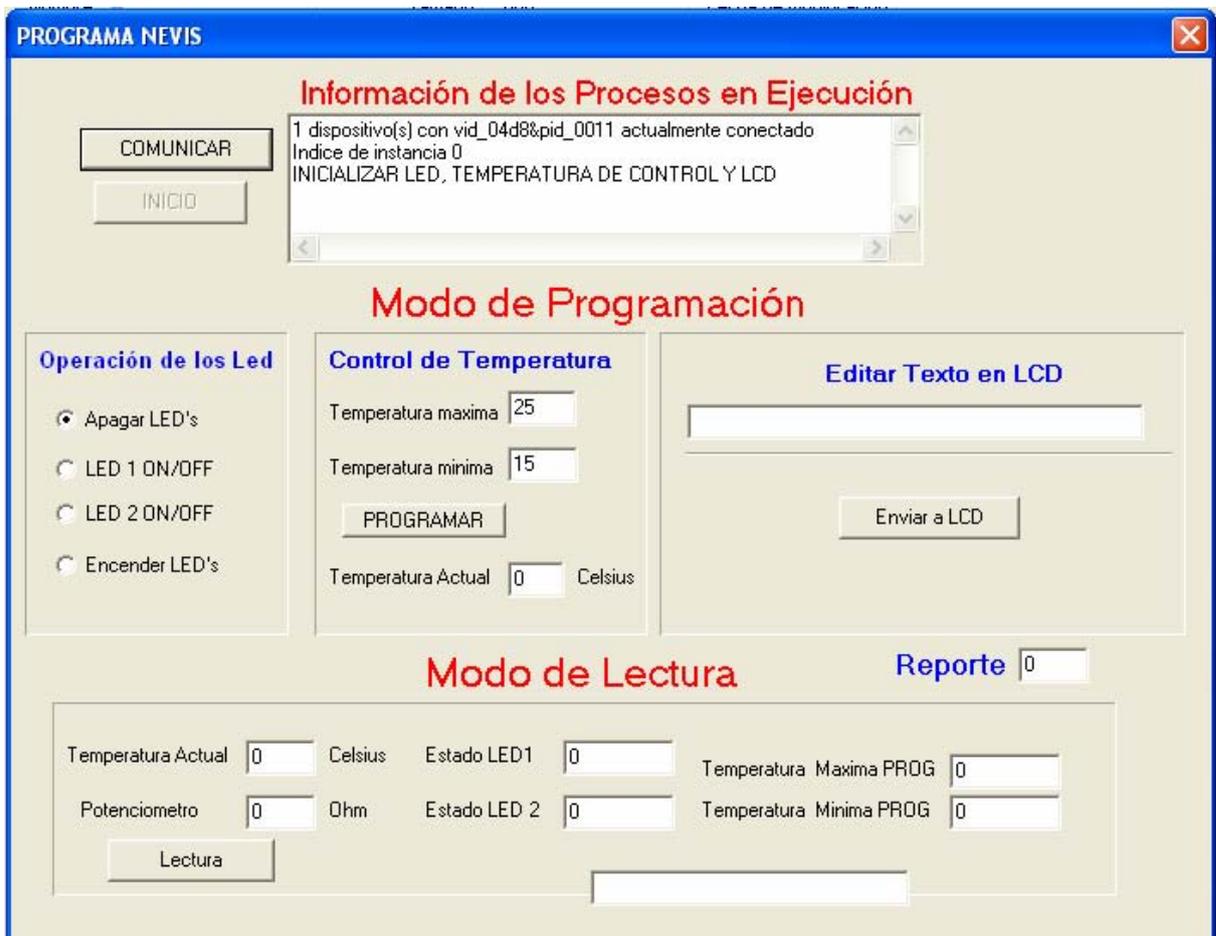
Una vez instalado el driver (con su respectiva dll picusbci.dll para que el microcontrolador y la PC se puedan comunicar entre sí), utilizando como aplicación un software hecho en Delphi, se da doble click sobre la aplicación que tiene como

nombre PicUSBDelphi.exe y como icono  que identifica al programa. A continuación se abrirá el siguiente panel:



**FIGURA 18:** Panel activado esperando comunicación.

Para comenzar la comunicación entre la PC y el dispositivo se pulsa el botón **COMUNICAR**, el cuál establecerá una comunicación a través del puerto USB con el microcontrolador, teniendo en cuenta las especificaciones USB 2.0. Cuando es establecida la misma aparecerá en el menú de Información de los procesos en ejecución, si el dispositivo está conectado o no a la PC, este menú informa también todos los procesos en ejecución del programa. Una vez establecida la comunicación con el dispositivo se pulsa el botón de **INICIO** donde se tomarán todos los valores por defecto, como se muestra en la figura 17, además de poner a funcionar un temporizador el cual estará supervisando si hay algún cambio en los diferentes parámetros para su actualización, donde podrá apreciarse el número del reporte del mismo.



**FIGURA 19:** Muestra la comunicación

En el Modo de Programación, se programan los diferentes parámetros que le darán órdenes al microcontrolador, éstos son:

1)En la Operación de Led se pueden manipular el apagado y el encendido de los mismos, con acción de click sobre los radiobutton que controlan cada uno de los leds y de esta forma es posible programar desde la PC una comunicación interactiva con la interfaz PIC.

2)En el Control de Temperatura se programará la temperatura máxima y mínima del sistema, al ser presionado el botón **Programar**, los datos son enviados para el hardware y se mostrarán en el LCD estos valores. En el Editor de texto en LCD se podrá escribir hasta 32 caracteres, los mismos serán enviados al pulsar el botón **Enviar al LCD** y se mostrarán en el LCD, todas estas operaciones en el Modo de Programación, su correcta comunicación con el hardware o no, se mostrará en el menú de Información de los Procesos en Ejecución. En el Modo de Lectura al pulsar el botón **LECTURA**, del estado de los diferentes parámetros del sistema, se obtiene la medición de la resistencia del potenciómetro que tiene un rango de 0 a 10 Kohm y una exactitud de  $\pm 1$  ohm, así como se indica el valor numérico de esta medición que se encuentra en la interfaz PIC, la temperatura actual del sistema, el estado de los led (1 encendido ó 0 apagado), así como las temperaturas mínimas y máximas programadas.

# CONCLUSIONES

En este Trabajo de Diploma se cumplen satisfactoriamente los objetivos inicialmente trazados, obteniéndose los siguientes resultados:

1. En el Proyecto se realiza una investigación teórica profunda y detallada sobre las especificaciones para la comunicación por puerto USB y su empleo en sistemas automatizados de control de parámetros tecnológicos, la cual constituye una fuente bibliográfica actual para profesionales, que estén interesados en diseñar estos sistemas empleando esta comunicación en el desarrollo de aplicaciones con los microcontroladores PIC.
2. Se realiza el diseño de las soluciones de hardware y software para la comunicación por puerto USB según las especificaciones 2.0, empleando como base tecnológica las aplicaciones con microcontroladores PIC, y las siguientes herramientas de diseño: Software Borland Delphi 6 para crear aplicaciones residentes en la PC empleando lenguaje de alto nivel con programación orientada a objeto, Software MPLAB 7.5 para la simulación, puesta a punto y la programación con el PIC, el compilador CCSC para ensamblar el código fuente en lenguaje C y la herramienta ICD2 para depurar errores y programar los microcontroladores PIC.
3. Se diseña una aplicación demostrativa de la comunicación por puerto USB según la especificación USB 2.0 entre la tarjeta electrónica PICDEM FS basada en el microcontrolador PIC18F4550 y una microcomputadora. Se obtiene el microprograma en lenguaje C a nivel del microcontrolador para configurar, controlar y efectuar la transmisión del buffer de datos en ambas direcciones, mostrándose los resultados mediante una aplicación de software residente en la PC.
4. Las soluciones técnicas de hardware y software quedan validadas y listas para ser empleadas en el diseño de sistemas automatizados para el control de procesos tecnológicos que en general requieran implementar la comunicación por puerto USB según esta especificación.

## **RECOMENDACIONES**

Se recomienda que se aplique de forma inmediata las soluciones técnicas obtenidas para la comunicación por puerto USB en el diseño de un sistema automatizado, solicitado por un Proyecto de I+D inscrito en el Programa Ramal de Ciencia y técnica del MIC.

## BIBLIOGRAFIA

1. Angulo Martínez J., Angulo Usategui J. Ma., y Martín Cuenca E. Aplicaciones de los Microcontroladores PIC de Microchip Editorial McGraw Hill, 1998.
2. Campos Cerda Manuel Fernando, Castañeda Pérez Ramiro, Contreras Torres Arturo Cesar. Implementación de un Sistema de Desarrollo Utilizando Microcontroladores PIC, Microchip Technology. Universidad de Guadalajara. 1998.
3. Compresión y comunicación de Datos con un microcontrolador PIC. [pdf].
4. Carlos Fernando Jiménez. Desarrollando para el puerto USB con la familia 18F2455/2550/4555/4550 y la PICDEM FS USB DEMONSTRATION BOARD de Microchip. Universidad Industrial de Santander, Enero de 2005 [pdf].
5. Gómez Ponce Marco Antonio, Mateos Maestro Jorge Alfredo, Medina Alatorre José Luís. Programación y Aplicaciones de los Microcontroladores PIC. Microchip Technology. Universidad de Guadalajara. 1998.
6. Grebene, Alan B. "Analog Integrated Circuit Design". 1994.
7. Introducción a los microcontroladores. [pdf]
8. Jordi Mayné Grau. Field Application Engineer. [pdf].
9. Milman J" Microelectronics 1993"
10. Manual de Programación de las Herramientas de Diseño C18. Software Ensamblador de lenguaje C al lenguaje nemotécnico del PIC familia 18XX. CD Rom Microchip. 2002.
11. Manual de Programación de las Herramientas de Diseño MPLAB. Software de Simulación y Debug del programa para aplicaciones con PIC. CD Rom Microchip 2002.
12. Manual de Programación de las Herramientas de Diseño ICEPIC. Software de Emulación de aplicaciones con PIC. CD Rom Microchip. 2003.
13. Manual de Programación de las Herramientas de Diseño PICSTAR. Software de Programación del circuito integrado PIC.
14. Manuel Rodríguez Padrón .PROYECTO PIC 18F452 DUPLEX RF. [pdf].
15. PIC18F2455/2550/4455/4550 Data Sheet 28/40/44-Pin High-Performance, Enhanced Flash USB Microcontrollers with nanoWatt Technology [pdf].

16. PICDEM™ FS USB DEMONSTRATION BOARD USER'S GUIDE, 2004 Microchip Technology Inc [pdf].
17. PIC'School: Módulo PIC18FXXXX "Un paso adelante. El fascinante mundo de los potentes PIC18" V 1.00 Marzo 2007. [pdf].
18. Universal Serial Bus Class Definitions for Communication Devices Version 1.1, January 19, 1999, [pdf].
19. USB 2.0 Specification Engineering, Date: 10/20/2000 [pdf].
20. Universal Serial Bus System Architecture (USB 2.0).Second Editions. [pdf]
21. [www.basicmicro.com](http://www.basicmicro.com)
22. [www.chessdirect.com](http://www.chessdirect.com)
23. [www.dhacel.com.ar](http://www.dhacel.com.ar)
24. [www.datasheet.com](http://www.datasheet.com)
25. [www.dgtprojects.com](http://www.dgtprojects.com)
26. [www.fide.com](http://www.fide.com)
27. [www.farnell.com](http://www.farnell.com)
28. [www.microchip.com](http://www.microchip.com)
29. [www.peonderey.com.es](http://www.peonderey.com.es)
30. [www.webelectronica.com.ar](http://www.webelectronica.com.ar)
31. [www.familia de productos HP/Información general del USB 1.0 y USB 2.0.html](http://www.familia de productos HP/Información general del USB 1.0 y USB 2.0.html)
32. [www.MicroPIC.Enero2005.html](http://www.MicroPIC.Enero2005.html)
33. [www.Puertos USB.html](http://www.Puertos USB.html)

# REFERENCIAS BIBLIOGRÁFICAS

- [1] Jordi Mayné Grau. Field Application Engineer. pdf].
- [2] Carlos Fernando Jiménez. Desarrollando para el puerto USB con la familia 18F2455/2550/4555/4550 y la PICDEM FS USB DEMONSTRATION BOARD de Microchip. Universidad Industrial de Santander, Enero de 2005 [pdf].
- [3] Universal Serial Bus Class Definitions for Communication Devices Version 1.1, January 19, 1999, [pdf].
- [4] USB 2.0 Specification Engineering, Date: 10/20/2000 [pdf].
- [5] Introducción a los microcontroladores. [pdf]
- [6] Gómez Ponce Marco Antonio, Mateos Maestro Jorge Alfredo, Medina Alatorre José Luís. Programación y Aplicaciones de los Microcontroladores PIC. Microchip Technology. Universidad de Guadalajara. 1998.
- [7] FT232BM Designers Guide, Future Technology Devices Intl. Ltd. 2002/2003 [pdf].
- [8] FT232BM Designers Guide Version 2.0. [pdf].
- [9] [www.webelectronica.com.ar](http://www.webelectronica.com.ar)
- [10] PIC18F2455/2550/4455/4550 Data Sheet 28/40/44-Pin High-Performance, Enhanced Flash USB Microcontrollers with nanoWatt Technology [pdf].
- [11] PIC'School: Módulo PIC18FXXXX "Un paso adelante. El fascinante mundo de los potentes PIC18" V 1.00 Marzo 2007. [pdf].
- [12] PICDEM™ FS USB DEMONSTRATION BOARD USER'S GUIDE, 2004 Microchip Technology Inc [pdf].