Diseño de un practicario para instrumentación virtual a través del protocolo USB 2.0

TESIS
QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMUNICACIONES Y ELECTRÓNICA
PRESENTA:

C. Oswaldo Ulises Juárez Sandoval

ASESORES

Dr. Leobardo Hernández González
Ing. Sergio Núñez Pérez

México D.F. Enero 2013
INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
UNIDAD CULHUACAN

TESIS INDIVIDUAL

Que como prueba escrita de su Examen Profesional para obtener el Título de Ingeniero en Comunicaciones y Electrónica, deberá desarrollar el C:

OSWALDO ULISES JUAREZ SANDOVAL

"DISEÑO DE UN PRACTICARIO PARA INSTRUMENTACIÓN VIRTUAL A TRAVES DEL PROTOCOLO USB 2.0"

El diseño e implementación de un practicario para instrumentación virtual con protocolo USB 2.0 como bus de comunicación, con diferentes módulos de aplicación, permitirán al interesado hacerse del conocimiento tanto teórico como experimental de una forma práctica. En esta Tesis se mostrarán prácticas implementadas en el campo de la instrumentación virtual, dado que ésta área alterna de aprendizaje le será de gran ayuda tanto a: alumnos, egresados, profesores e interesados en el tema.

CAPITULADO

CAPITULO 1 INTRODUCCION A INSTRUMENTACION
CAPITULO 2 DISEÑO DE INTERFAZ USB 2.0
CAPITULO 3 PRUEBAS EXPERIMENTALES
CAPITULO 4 TRABAJOS FUTUROS
CONCLUSIONES

México D. F., a 16 de Noviembre del 2012

PRIMER ASESOR:  SEGUNDO ASESOR:
DR. LEOBARDO HERNANDEZ GONZALEZ  ING. SERGIO NUÑEZ PEREZ
Vo. Bo.

APROBADO

M. en C. ANTONIO ROMERO ROJANO  M. en C. HECTOR BECERRIL MENDOZA
JEFE DE LA CARRERA DE I.C.E.  SUBDIRECTOR ACADÉMICO
DEDICATORIA.

Dedico esta Tesis:

A mis padres, porque creyeron en mí y porque me sacaron adelante, dándome ejemplos dignos de superación y entrega, porque en gran parte gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera, y porque el orgullo que sienten por mí, fue lo que me hizo ir hasta el final. Va por ustedes, por lo que valen, porque admiro su fortaleza y por lo que han hecho de mí.

A mis hermanos, tíos, primos, abuelos, amigos y a los mejores profesores que en mi forjaron el ansia, deseo de superación y el anhelo de triunfo en la vida. "Los Profesores de mi querida ESIME Culhuacan".

A mis asesores que con amplio apoyo han hecho posible la culminación de esta tesis.

Mil palabras no bastarían para agradecerles su apoyo, su comprensión y sus consejos en los momentos difíciles.

A todos, espero no defraudarlos y contar siempre con su valioso apoyo, sincero e incondicional.
ÍNDICE

Justificación ........................................................................................................... I

CAPÍTULO I. INTRODUCCIÓN A INSTRUMENTACIÓN

1.1 Instrumentación .................................................................................................... 1
   1.1.1 Los instrumentos virtuales ........................................................................... 1
   1.1.2 Instrumentos Virtuales y Tradicionales ......................................................... 2
   1.1.3 Hardware de Instrumentación Virtual e Instrumentación Tradicional .......... 3
   1.1.4 Sistema de Desarrollo de LabVIEW ............................................................. 3
   1.1.5 Sistema EDIBON de Adquisición de Datos/Sistema de Instrumentación Virtual .. 6
   1.1.6 Frecuencímetro virtual ................................................................................ 7
   1.1.7 Termómetro para PT100 ............................................................................... 7
   1.1.8 Osciloscopio USB PoScope basic2 Kit completo ........................................... 8

1.2 Puertos para envío y adquisición de datos .............................................................. 8
   1.2.1 Puerto .......................................................................................................... 8
   1.2.2 Puertos Físicos ........................................................................................... 9
   1.2.3 Puertos de comunicación .............................................................................. 9
   1.2.4 Puerto paralelo ............................................................................................ 10
   1.2.5 Puerto paralelo IDE .................................................................................... 10
   1.2.6 Puerto paralelo SCSI ................................................................................... 10
   1.2.7 Puerto PCI .................................................................................................. 11
   1.2.8 PCI-Express ............................................................................................... 12
   1.2.9 Puerto infrarrojo .......................................................................................... 12

1.3 Microcontrolador .................................................................................................. 13
   1.3.1 Arquitectura de un Microcontrolador ............................................................ 13
   1.3.2 Comunicación con un Microcontrolador ....................................................... 14
   1.3.3 Puerto USB .................................................................................................. 15
   1.3.4 Cables y conectores ................................................................................... 17
   1.3.5 Diseño de instrumentos virtuales ................................................................ 19
   1.3.6 Las características de los instrumentos de medición ..................................... 20
   1.3.7 Adquisición de datos .................................................................................. 22
   1.3.8 Puente H .................................................................................................... 23
   1.3.9 Compilador PIC Basic Pro ......................................................................... 24

CAPÍTULO II. DISEÑO DE INTERFAZ USB 2.0

2.1 Desarrollo de la interfaz ....................................................................................... 26
2.2 Microcontrolador 18F4550 ................................................................................ 26
2.3 Diagrama de flujo para el programa del PIC18F4550 ............................................ 27
2.4 Sub rutinas de lectura y envío de datos DoUSBIn: y DoUSBOut ......................... 30
2.5 Diagrama eléctrico de la interfaz-USB2 ................................................................ 31
CAPÍTULO III. PRUEBAS EXPERIMENTALES

3.1 Pruebas y prácticas a realizar ................................................. 36
3.2 Comandos de uso frecuente ...................................................... 37
3.3 Conexión entre la interfaz y la PC ............................................... 37
3.4 Programa para el manejo de puertos ........................................... 41
3.5 Salida de datos ........................................................................... 41
3.6 Entrada de datos y despliegue de mensaje ....................................... 44
3.7 Salida de datos en código ASCII .................................................. 47
3.8 Control de un motor .................................................................. 50
3.9 Desarrollo de un puente H para el control de un motor .................... 50
3.10 Control de un motor por RF ......................................................... 55
3.11 Joystick ...................................................................................... 58
3.12 Notepad .................................................................................... 59
3.13 Elaboración de un Voltmetro ....................................................... 63

CAPÍTULO IV. TRABAJOS FUTUROS

4.1 Tendencias para USB 3.0 ............................................................ 65

Conclusiones .................................................................................... 68
Bibliografía ......................................................................................... 69
Agradecimientos ................................................................................ 70
Anexos ............................................................................................... 71
ÍNDICE DE FIGURAS

Figura 1.1 Instrumentos tradicionales de medición .......................................................... 1
Figura 1.2 Instrumentos tradicionales (izquierda), instrumentos virtuales (derecha) .................. 4
Figura 1.3 Sistema de adquisición de datos EDIBON ......................................................... 6
Figura 1.4 Pantallas del sistema de adquisición de datos EDIBON .......................................... 6
Figura 1.5 Modulo frecuencímetro virtual y su despliegue ..................................................... 7
Figura 1.6 Termómetro virtual con transmisión y receptor de la información de hasta 8 termómetros inalámbricamente y su despliegue ......................................................... 7
Figura 1.7 Osciloscopio virtual USB y su despliegue ............................................................ 8
Figura 1.8 Tipos de conectores ................................................................................................. 9
Figura 1.9 Conector RS 232 hembra .................................................................................... 9
Figura 1.10 Tipos de puertos paralelo SCSI .......................................................................... 11
Figura 1.11 Tipos de conectores PCI .................................................................................. 12
Figura 1.12 Esquema de un Microcontrolador ..................................................................... 13
Figura 1.13 Diagrama a bloques de una comunicación RS-232 .............................................. 14
Figura 1.14 Cronología del USB .......................................................................................... 16
Figura 1.15 Conectores USB y su configuración ................................................................. 19
Figura 1.16 Diagrama a bloques de la adquisición de datos .................................................. 22
Figura 1.17 Estructura de un puente H ................................................................................ 24
Figura 1.18 Los 2 estados básicos del circuito para el control de giro ...................................... 24
Figura 1.19 Pantalla principal del PIC-Basic Pro (PBP) .......................................................... 25
Figura 2.1 Diagrama de terminales del PIC 18F4550 ............................................................. 27
Figura 2.2 Diagrama de Flujo para la programación del PIC18F4550 ...................................... 29
Figura 2.3 Pistas de la interfaz USB2 .................................................................................. 32
Figura 2.4 Vista preliminar de interfaz-USB2 ..................................................................... 32
Figura 2.5 Vista de placa final con las pistas de cobre ........................................................... 33
Figura 2.6 Placa baquelita ferro-clorada y brocas de 1/64,3/64 y 1/32 ..................................... 34
Figura 2.7 Parte inferior de la placa una vez perforada, soldada y con mica protectora sujetada por tornillos ......................................................................................... 35
Figura 2.8 Parte superior de la placa .................................................................................... 35
Figura 3.1 Pantalla con la interfaz desconectada .................................................................... 38
Figura 3.2 Interfaz conectada .............................................................................................. 38
Figura 3.3 Despliegue de información introducida por el usuario ........................................... 42
Figura 3.4 Programa para la adquisición de datos, visualiza en pantalla el valor binario de ......................................................................................................................... 46
Figura 3.5  Programa que muestra un mensaje al insertar un dato por el puerto C ......................................................... 49
Figura 3.6  Análisis de los elementos y sus valores ............................................................................................................. 51
Figura 3.7  Programa de control del motor ......................................................................................................................... 54
Figura 3.8  Programa de control del motor de RF .................................................................................................................. 55
Figura 3.9  Programa de control del Shape1 (azul) por datos de entrada en el puerto “C” ...................................................... 59
Figura 3.10 Programa de un Notepad .................................................................................................................................. 62
Figura 3.11 Programa de un medidor de voltaje .................................................................................................................. 63
Figura 4.1  Cable y conector USB 3.0 .................................................................................................................................... 67

ÍNDICE DE TABLAS

Tabla I  Instrumentos Tradicionales vs. Virtuales .................................................................................................................. 4
Tabla II  Características de PIC 18F4550 .................................................................................................................................. 26

DIAGRAMAS

Diagrama 1: Diagrama eléctrico de la interfaz-USB2 ......................................................................................................... 31
Diagrama 2: Diagrama para desplegar información .............................................................................................................. 42
Diagrama 3: Diagrama para la adquisición de datos ............................................................................................................. 46
Diagrama 4: Circuito propuesto por el autor ......................................................................................................................... 50
Diagrama 5: Circuito propuesto para la elaboración del puente H para la interfaz ................................................................. 53
Diagrama 6: Diagrama para el control de un motor con puente H .......................................................................................... 53
Diagrama 7: Diagrama para el control de un motor con puente H y juego de relevadores ...................................................... 57
Diagrama 8: Medidor de voltaje .............................................................................................................................................. 63
Justificación:

El diseño e implementación de un practicario para instrumentación virtual con protocolo USB 2.0 como bus de comunicación, con diferentes módulos de aplicación, permitirán al interesado hacerse del conocimiento tanto teórico como experimental de una forma práctica.

En esta Tesis se mostrarán prácticas implementadas en el campo de la instrumentación virtual, dado que esta área alterna de aprendizaje le será de gran ayuda tanto a alumnos, egresados, profesores e interesados en el tema.
Capítulo I

INTRODUCCIÓN A LA INSTRUMENTACIÓN.

1.1 Instrumentación

Se define a la instrumentación como el conjunto de sistemas para el sensado y procesamiento de la información proveniente de variables físicas y químicas, a partir de los cuales se realiza el monitoreo y control de procesos, empleando dispositivos y tecnologías electrónicas. En la Figura 1.1 se pueden observar distintos tipos de instrumentos de medición utilizados en la industria electrónica.

Figura 1.1 Instrumentos tradicionales de medición.

1.1.1 Los instrumentos virtuales

Los instrumentos virtuales nacen a partir del uso del computador personal (PC) como "instrumento" de medición. Por medio de una interfaz de hardware se pueden reducir las señales de medición definidas por el usuario; son entre otras: temperatura, presión, caudal, etc., mientras que los instrumentos tradicionales tienen funcionalidades fijas, definidas y limitadas.
El inicio de los sistemas de instrumentación virtual, se sitúan a mediados de los años 60 cuando Hewlett Packard, desarrolló el bus para instrumentación HP-IB (Hewlett Packard Interface Bus) que permitía conectar su gama de instrumentos programables a un ordenador. Esta interfaz ganó rápidamente gran popularidad y en 1975 fue aceptada como un estándar denominado IEEE-488. Desde aquellos días hasta ahora el estándar ha sufrido varias modificaciones y el bus GPIB (acrónimo de General Purpose Interface Bus) se ha convertido en uno de los estándares más populares en el campo de la instrumentación programable. El concepto de instrumentación virtual fue introducido por la compañía National Instruments en el año 2001, los cuales crearon un software que le permitía a la computadora realizar mediciones externas.

El término "virtual" nace a partir del hecho de que cuando se utiliza la PC como "instrumento" es el usuario mismo quién, a través del software, define su aplicación y “apariencia” y por ello se dice que se "virtualiza" el instrumento, ya que este puede ser definida una y otra vez por el usuario y no por el fabricante.

La nueva tendencia en la instrumentación es la medición virtual. La idea es sustituir y ampliar elementos de "hardware" por otros de "software", para ello se emplea un procesador (normalmente un PC) que ejecute un programa específico el cual puede ser desarrollado por el usuario o previamente definido por el fabricante, este programa se comunica con los dispositivos para configurarlos y leer sus parámetros, y es capaz de automatizar las medidas, procesado de la información, visualización y actuación remotamente, etc.

Algunos programas especializados en este campo son LabVIEW y Agilent-VEE (antes HP-VEE). Y algunos bus de comunicación populares como; GPIB, RS-232, USB, etc.

1.1.2 Instrumentos Virtuales y Tradicionales

Muchos ingenieros y científicos tienen en sus laboratorios una combinación tanto de instrumentos virtuales como tradicionales. Adicionalmente, algunos instrumentos tradicionales proporcionan una medición especializada que los ingenieros o científicos
prefieren sean definidos por el vendedor que por ellos mismos. Esto genera la pregunta, “¿Existe compatibilidad entre los instrumentos virtuales y tradicionales?”.

Los instrumentos virtuales son compatibles con los instrumentos tradicionales casi sin excepción. El software de instrumentación virtual proporciona bibliotecas para crear interfaz con buses de instrumentos comunes u ordinarios como el serial GPIB, o Ethernet.

Adicional a las bibliotecas proporcionadas por el fabricante, los vendedores de instrumentos han contribuido con más de 4,000 controladores para la Biblioteca de Controladores de Instrumentos de National Instruments.

Los controladores proporcionan un conjunto de funciones de alto nivel para realizar interfaz con instrumentos de medición tradicional, donde cada controlador de instrumentos es específicamente diseñado para un modelo en particular y proporciona una interfaz con características únicas.

### 1.1.3 Hardware de Instrumentación Virtual e Instrumentación Tradicional

Una característica importante de la instrumentación virtual es la maximización del instrumento tanto del software y del hardware, véase la Tabla 1. Como ejemplo se tiene que National Instruments se enfoca en adaptarse o usar tecnologías de alta inversión de compañías como Microsoft, Intel, Analog Devices, Xilinx, y otros.

Para el software, utiliza la tremenda inversión de Microsoft en sistemas operativos y sus herramientas de desarrollo. Para hardware, aprovecha la inversión de Analog Devices en la tecnología de convertidores A/D.

### 1.1.4 Sistema de desarrollo de LabVIEW.

LabVIEW desarrollado por National Instruments, es un reconocido entorno de desarrollo optimizado para ingenieros y científicos que crean aplicaciones de pruebas, mediciones y control. Con LabVIEW, se puede adquirir rápida y fácilmente señales físicas, realizar análisis de datos y almacenar resultados en diferentes formatos. En la Figura 1.2 se observa la diferencia entre instrumentación tradicional e instrumentación virtual.
donde se puede observar que comparten a gran escala la misma arquitectura en componentes, pero con filosofías de desarrollo radicalmente diferentes.

**Tabla I** Instrumentos Tradicionales vs Virtuales.

<table>
<thead>
<tr>
<th>Instrumento Tradicional</th>
<th>Instrumento Virtual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definido por el fabricante</td>
<td>Definido por el usuario</td>
</tr>
<tr>
<td>Funcionalidad específica, con conectividad limitada.</td>
<td>Funcionalidad ilimitada, orientado a aplicaciones, conectividad amplia.</td>
</tr>
<tr>
<td>Hardware es la clave.</td>
<td>Software es la clave</td>
</tr>
<tr>
<td>Alto costo/función</td>
<td>Bajo costo/función, variedad de funciones, reusuable.</td>
</tr>
<tr>
<td>Arquitectura &quot;cerrada&quot;</td>
<td>Arquitectura &quot;abierta&quot;.</td>
</tr>
<tr>
<td>Lenta incorporación de nuevas tecnología.</td>
<td>Rápida incorporación de nuevas tecnologías, gracias a la plataforma PC.</td>
</tr>
<tr>
<td>Bajas economías de escala, alto costo de mantenimiento.</td>
<td>Altas economías de escala, bajos costos de mantenimiento.</td>
</tr>
</tbody>
</table>

**Figura 1.2.** Instrumentos tradicionales (izquierda), instrumentos virtuales (derecha).
El desarrollo de este tipo de instrumentación virtual en especial se basa en funciones de aplicación específica como son:

a) Monitoreo y control industrial. *LabVIEW* en una forma resumida cuenta con módulos y juegos de herramientas (*Toolkits*) como son:

- Módulo en tiempo real
- Ejecución de trazado en tiempo real
- Módulo de diseño FPGA
- Módulo de pantalla táctil
- Módulo de registro de datos y control de supervisión
- Módulo de diagrama de estado
- Módulo asistente de movimiento

Así como la integración de controladores de automatización programables (PACs), lo que permite crear sistemas distribuidos de monitoreo y control de sistemas existentes (PLCs).

b) Generación de reportes y almacenamiento de datos. Para generación de reportes y almacenamiento de datos se cuenta con los paquetes de herramientas:

- *LabVIEW SignalExpress*
- *Report Generation Toolkit para Microsoft Office*
- *LabVIEW Database Connectivity Toolkit*
- *LabVIEW DataFinder Toolkit*
- *LabVIEW Internet Toolkit*
1.1.5 Sistema EDIBON de Adquisición de Datos/Sistema de Instrumentación Virtual

Este módulo se encuentra enfocado a la adquisición de datos y al mismo tiempo abre la posibilidad de simular y trabajar tanto con parámetros de entrada como de salida, gracias a que cuenta con osciloscopio, generador de funciones, analizador de espectro, analizador de transitorios, analizador lógico, generador lógico, entre otros. En las figuras 1.3 y 1.4 se muestra la imagen de este tipo específico de sistema así como una pantalla del sistema de adquisición de datos.

Figura 1.3 Sistema de adquisición de datos EDIBON.

Figura 1.4 Pantallas del sistema de adquisición de datos EDIBON.
1.1.6 Frecuenciómetro virtual

Esta tarjeta de desarrollo se puede conectar a un puerto serial de la PC y mide frecuencias desde 1 hasta 80 MHz.

Figura 1.5 Modulo frecuencímetro virtual y su despliegue.

1.1.7 Termómetro para PT100

Instrumento para medir la temperatura: Recibe la señal del PT100 y transmite la temperatura mediante un poderoso sistema de radio frecuencia.

Se comunica directamente con el módulo transceptor Clave 875 que al conectarse a la PC facilita la lectura de la temperatura. Este sistema tiene un alcance de alrededor de 25 m. Se pueden conectar en red hasta 8 termómetros a un solo transceptor.

Figura 1.6 Termómetro virtual con transmisión y receptor de la información de hasta 8 termómetros inalámbricamente y su despliegue.
1.1.8 Osciloscopio **USB PoScope Basic2**

El Osciloscopio **USB PoScope Basic2** está especialmente diseñado para Ingenieros. Simplemente se conecta el equipo al puerto **USB** de la PC, es capaz de analizar señales del tipo: **I2C, SPI, 1-Wire, UART** etc.

Este equipo contiene osciloscopio, analizador lógico, analizador de espectros, grabador y generador de señales lógicas.

**Figura 1.7** Osciloscopio virtual USB y su despliegue.

1.2 Puertos para envío y adquisición de datos

1.2.1 Puerto

En informática, un puerto es una forma genérica de denominar a una interfaz de comunicación por la cual diferentes tipos de datos pueden ser enviados y recibidos. Dicha interfaz puede ser física, o puede ser a nivel software (los puertos permiten la transmisión de datos entre diferentes ordenadores).
1.2.2 Puertos Físicos

Los puertos físicos, son aquellos como el puerto "paralelo" de una computadora. En este tipo de puertos, se puede llegar a conectar: un monitor, la impresora, el escáner, etc. Actualmente el puerto paralelo es utilizado como una conexión para impresoras de tipo laser y de inyección de tinta aunque en los últimos años ha sido remplazado por el puerto USB.

1.2.3 Puertos de comunicación

Además del RS-232 con su conector DB-9, existen otros protocolos de comunicación bajo la arquitectura IBM como la comunicación por el puerto paralelo a través de un conector DB-25, este tipo de comunicación es más rápida que la comunicación serial, pero está restringida en cuanto al alcance que puede cubrir, existen diversos periféricos especializados para monitorear y muestrear parámetros, la gran mayoría de uso industrial, pero también equipos de oficina como impresoras, fax, que involucran micro-controladores en su diseño y que cabe mencionar dentro de los protocolos de comunicación que existen entre una computadora y un microprocesador. Además existen otros tipos de conectores como el DC-37, DD-50 que son más robustos y también están involucrados en la comunicación y/o transferencia de datos entre computadora y periféricos (figuras 1.8 y 1.9).

![Tipos de conectores](image1.png)

**Figura 1.8** Tipos de conectores. **Figura 1.9** Conector RS 232 hembra.
1.2.4 Puerto paralelo

Un puerto paralelo es una interfaz entre una computadora y un periférico cuya principal característica es que los bits de datos viajen juntos, enviando un paquete de byte a la vez. Es decir, se implementa un cable o una vía física para cada bit de datos formando un bus. Mediante el puerto paralelo se puede controlar también periféricos como focos, motores entre otros dispositivos, adecuados para automatización.

El cable paralelo es el conector físico entre el puerto paralelo y el dispositivo periférico. En un puerto paralelo habrá una serie de bits de control en vías aparte que fluyen en ambos sentidos por caminos distintos.

En contraposición al puerto paralelo está el puerto serie, que envía los datos bit a bit por el mismo hilo.

1.2.5 Puerto paralelo IDE

Existe otro puerto paralelo usado masivamente en las computadoras denominado puerto paralelo IDE, también llamado PATA (Parallel ATA), usado para la conexión de discos duros, unidades lectoras/grabadoras (CD-ROM, DVD), unidades magneto-ópticas, unidades ZIP y SuperDisk, entre la placa base del computador y el dispositivo.

1.2.6 Puerto paralelo SCSI

Un tercer puerto paralelo, muy usado en los ordenadores Apple Macintosh y en servidores es el SCSI (Small Computer System Interface). Al igual que IDE ha sido usado para la conexión de discos duros, unidades ópticas lectoras/grabadoras (CD-ROM, DVD), unidades magneto-ópticas y SuperDisk, pero también de otros dispositivos como escáneres e incluso otro computador de diferente plataforma hardware y sistema operativo. En la Figura 1.10 se muestra un conjunto de puertos SCSI de diferentes formas, observándose que se trata de un estándar no convencional.
Figura 1.10 Tipos de puertos paralelo SCSI.

1.2.7 Puerto PCI

Los puertos PCI (Peripheral Component Interconnect) son ranuras de expansión en las que se puede conectar tarjetas de sonido, de video, de red etc. El puerto PCI sigue usando hoy en día y se pueden encontrar diversos componentes en el formato PCI. Dentro de las ranuras PCI está el PCI-Express. Los componentes que suelen estar disponibles en este tipo de ranuras son:

- Capturadoras de televisión
- Controladoras RAID
- Tarjetas de red alámbricas e inalámbricas
- Tarjetas de sonido, video etc.

Cabe mencionar que existen diferentes versiones de puertos PCI al igual que en el puerto serie como en el paralelo, En la Fig.1.11 se muestra un ejemplo de puertos PCI.
1.2.8 PCI-Express

*PCI-Express* (anteriormente conocido por las siglas *3GIO, 3rd Generation Input/Output*) es un nuevo desarrollo del bus *PCI* que usa los conceptos de programación y los estándares de comunicación existentes, pero se basa en un sistema de comunicación en serie, mucho más rápido que *PCI* y *AGP*. Este sistema es apoyado principalmente por Intel, que empezó a desarrollar el estándar con nombre de proyecto *Arapahoe* después de retirarse del sistema *Infiniband*. Alcanza una velocidad de 16x (8GB/s) y es utilizado en tarjetas gráficas.

1.2.9 Puerto infrarrojo

Este tipo de puertos sirven para conectarse con otros dispositivos que cuenten con infrarrojos sin la necesidad de cables, los infrarrojos son como el Bluetooth (Tecnología de última generación). La principal diferencia es que la comunicación de Infrarrojos usa como medio la luz, en cambio el Bluetooth utiliza ondas de radio frecuencia.
1.3 Microcontrolador

1.3.1 Arquitectura de un Microcontrolador.

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

En la Figura 1.12, se observa al Microcontrolador dentro de un encapsulado de circuito integrado, con su procesador (CPU), buses, memoria, periféricos, puertos de entrada y salida. Fuera del encapsulado se ubican otros circuitos para completar periféricos internos y dispositivos que pueden conectarse a las terminales de entrada/salida. También se conectarán a las terminales del encapsulado la alimentación, tierra, circuito de completamiento del oscilador y otros circuitos necesarios para que el Microcontrolador pueda trabajar.

![Figura 1.12. Esquema de un Microcontrolador.](image)
1.3.2 Comunicación con un microcontrolador

Anteriormente una de las conexiones más populares para establecer una conexión entre una computadora y un Microcontrolador o microprocesador era por medio de la comunicación serial de una computadora, además, la comunicación serial puede ser utilizada para adquisición de datos si se usa en conjunto con un dispositivo remoto de muestreo.

De acuerdo a la Figura 1.13, la comunicación se establece entre la computadora y el Microcontrolador a través de una interfaz, que se encarga de hacer compatibles los voltajes y señales que se manejan entre los dos dispositivos, una vez realizada la conexión se puede ejecutar la aplicación deseada por medio de un programa, y estar enviando y recibiendo datos.

Pero este tipo de comunicación está limitada en cuanto a la conexión, ya que es punto a punto bien definido, entre el hardware y la computadora.

La longitud de los datos puede ser de 5, 7 u 8 bits, que se transfieren bit a bit, y aunque es un método relativamente lento, alcanza distancias más largas de comunicación que pueden llegar a los 1200m.

*Figura 1.13 Diagrama a bloques de una comunicación RS-232*
1.3.3 Puerto USB

En el año 1996 las empresas:

- *Digital Equipment Corporation*
- *NEC*
- *Compaq*
- *Intel*
- *Northern Telecom*
- *Microsoft*
- *IBM*

Se unieron para crear el *Universal Serial Bus* o *USB*. Las primeras unidades flash fueron fabricadas por *M-Systems* bajo la marca "Disgo" en tamaños de 8 MB, 16 MB, 32 MB y 64 MB. Estos fueron promocionados como los "verdaderos replazos del disquete", y su diseño continuó hasta los 256 MB. Los fabricantes asiáticos pronto fabricaron sus propias unidades más baratas que las de la serie *Disgo*.

La especificación *USB 1.0* (*denominada de baja velocidad*) se estableció por primera vez en 1996, y debido a su baja velocidad (0.192Kb/s-1.5Mb/s) sólo se utilizaba para dispositivos de interfaz humana como ratones, teclados, trackballs, etc.

En 1998, se lanzaría una nueva revisión (*USB 1.1*) que mejora la velocidad (1.5Mb/s-12Mb/s). Aunque es una mejora, se va volviendo cada vez más insuficiente para transferir información de varios megas de peso a medida que pasa el tiempo.

Entonces surge *USB 2.0*, un interfaz de alta velocidad (60Mb/s-480Mb/s) que consigue satisfacer las necesidades de transferencia y comienza a ser comercializado para discos duros externos, *pendrives*, memorias *plug & play*, etc. En la Figura 1.14 se puede observar la cronología descrita anteriormente.
USB es una nueva arquitectura de bus desarrollada que forman parte de los avances plug & play y permite instalar periféricos sin tener que abrir la máquina para instalarle hardware, es decir, basta con que se conecte dicho periférico en la parte posterior de la computadora.

Una característica importante es que permite a los dispositivos trabajar a velocidades mayores, en promedio a unos 12 Mbps, esto es más o menos de 3 a 5 veces más rápido que un dispositivo de puerto paralelo y de 20 a 40 veces más rápido que un dispositivo de puerto serial.

Trabaja como interfaz para transmisión de datos y distribución de energía, que ha sido introducida en el mercado de PC´s y periféricos para mejorar las lentas interfaces serie (RS-232) y paralelo. Esta interfaz de 4 hilos, 12 Mbps y "plug & play", distribuye 5V para alimentación, transmite datos y está siendo adoptada rápidamente por la industria informática.

Es un bus basado en el paso de un testigo, semejante a otros buses como los de las redes locales en anillo con paso de testigo y las redes FDDI (Fiber Distributed Data Interface). El controlador USB distribuye testigos por el bus. El dispositivo cuya dirección coincide con la que porta el testigo responde aceptando o enviando datos al controlador. Este también gestiona la distribución de energía a los periféricos que lo requieran.
Emplea una topología de estrellas apiladas que permite el funcionamiento simultáneo de 127 dispositivos a la vez. En la raíz o vértice de las capas, está el controlador anfitrión o host que controla todo el tráfico que circula por el bus. Esta topología permite a diversos dispositivos conectarse a un único bus lógico sin que los dispositivos que se encuentran más abajo en la pirámide sufran retardo. A diferencia de otras arquitecturas, USB no es un bus de almacenamiento y envío, de forma que no se produce retardo en el envío de un paquete de datos hacia capas inferiores.

Como detalle sorprendente es que cada puerto utiliza una única solicitud de interrupción (IRQ) independientemente de los periféricos que tenga conectados (sea 1 ó 127) por lo tanto no hay riesgo de conflictos entre una cantidad de dispositivos que de otra forma no podrían ser conectados por falta de recursos; de la misma manera tampoco utilizan DMA (asignación de memoria).

El sistema de bus serie universal USB consta de tres componentes principales y que son:

- Controlador.
- Hubs o Concentradores.
- Periféricos.

**1.3.4 Cables y conectores**

USB 1.1 transfiere señales y energía a los periféricos utilizando un cable de 4 hilos, apantallado para transmisiones a 12 Mbps y no apantallado para transmisiones a 1.5Mbps; con dos conductores para alimentación y los otros dos para señal, debiendo estos últimos ser trenzados o no según la velocidad de transmisión.

El calibre de los conductores destinados a alimentación de los periféricos varía desde 20 a 26 AWG, mientras que el de los conductores de señal es de 28 AWG. La longitud máxima de los cables es de 5 metros.

Por lo que respecta a los conectores son del tipo ficha (o conector) y receptáculo, se clasifican en dos tipos: serie A y serie B. Los primeros presentan las
cuatro terminales correspondientes a los cuatro conductores alineados en un plano. El color recomendado es blanco sucio y los receptáculos se presentan en cuatro variantes: vertical, en ángulo recto, panel y apilado en ángulo recto así como para montaje pasa muro. Se emplean en aquellos dispositivos en los que el cable externo, está permanentemente unido a los mismos, tales como teclados, ratones, y hubs o concentradores.

Los conectores de la serie B presentan los contactos distribuidos en dos planos paralelos, dos en cada plano, y se emplean en los dispositivos que deban tener un receptáculo al cual poder conectar un cable USB. Por ejemplo impresoras, scanner, y módem.

Aplicaciones actuales y posibilidades a futuro del estándar USB.

- Discos duros de estado sólido portátiles.
- Adaptadores de video para monitores de PC.
- Grabadores de audio y video sobre bus USB.
- Conexiones de PC a PC a través de puertos USB.
- Sustitución de los puertos serie y paralelo.

En la Figura 1.15 se muestran las vistas frontales y configuración de terminales para el conector USB tipo A y B.
1.3.5 Diseño de instrumentos virtuales

Para construir un instrumento virtual, sólo se requiere de una PC, una tarjeta de adquisición de datos con acondicionamiento de señales (PCMCIA, ISA, XT, PCI, etc.) y el software apropiado.

Un instrumento virtual debe realizar como mínimo las tres funciones básicas de un instrumento convencional: adquisición, análisis y presentación de datos.

La instrumentación virtual puede también ser implementada en equipos móviles (laptops), equipos distribuidos en campo (RS-485 o RS-232), equipos a distancia (conectados vía radio, Internet, etc.), o equipos industriales (NEMA 4X, etc.).

Actualmente existe una tarjeta de adquisición de datos para casi cualquier tipo de bus o canal de comunicación para PC (ISA, PCI, USB, serial RS-232, RS-422,
paralelo EPP, PCMCIA, CompactPCI, PC/104, VMEbus, CAMAC, PXI, VXI GPIB, etc.), y existe un driver para casi cualquier sistema operativo (WIN /3.1 /95 /2000 /XP /NT, DOS, Unix, Linux, MAC OS, etc.).

1.3.6 Las características de los instrumentos de medición

Como se ha indicado anteriormente, los instrumentos de medición hacen posible la observación de los fenómenos eléctricos y su cuantificación entre otros procesos. Ahora bien, estos instrumentos no son sistemas ideales sino reales, y por lo tanto tienen una serie de limitaciones que se deben de tomar en cuenta para poder juzgar si afectan de alguna manera las medidas que se están realizando, y poder determinar así mismo la veracidad de las anteriores. Las características que definen el comportamiento de los instrumentos son las siguientes:

- **Campo de Medida o Rango (Range)**

  El rango es el conjunto de valores en la escala de medición dentro de los límites superior e inferior, denota la capacidad del equipo, puede expresarse en unidades físicas.

- **Alcance (Span)**

  Es la diferencia algebraica entre los valores superior e inferior del rango del equipo.

- **Exactitud (Accuracy)**

  Capacidad del instrumento para acercarse y poder medir el valor real. Es el parámetro que representa o denota cuando una lectura puede ser incorrecta, generalmente se representa como un porcentaje de la escala completa de lectura, o en términos de +/- las unidades del instrumento.
• Precisión (Accuracy)

La precisión es la capacidad del instrumento para proporcionar datos y resultados iguales en diferentes medidas bajo las mismas condiciones.

• Zona Muerta (Dead band)

Intervalo de valores de la variable que no hace variar la indicación o la señal del instrumento, es decir que no se produce respuesta alguna.

• Umbral (Threshold)

Nivel mínimo necesario para que el instrumento empiece a indicar una medida, o para que empiece a ser registrado como un cambio.

• Sensibilidad (Sensitivity)

Está dada por la razón entre el incremento de la lectura y el incremento de la variable que la ocasiona, después de haber alcanzado el reposo.

• Repetibilidad (Repeatability)

Capacidad del instrumento de medir o indicar valor idénticos de la misma variable bajo las mismas condiciones de funcionamiento en todos los casos.

• Resolución

Mínima subdivisión de la escala. A mayor resolución el instrumento será más preciso.

• Incertidumbre (uncertainty)

Denota la inexactitud del instrumento o la tendencia al error que pueda tener.

• Linealidad (linearity)
Línea recta formada por la sucesión ordenada y constante que indica el grado de proporcionalidad entre la variable física y el valor medido, o entre la variable física y la acción ejercida.

- **Histéresis (Hysteresis)**

Diferencia máxima que se observa en los valores indicados por el instrumento para un mismo valor del campo de medida, cuando la variable recorre toda la escala en forma ascendente y luego en forma descendente.

### 1.3.7 Adquisición de datos.

La señal física pasa por una serie de etapas para poder ser leída por la computadora, como se muestra en la Figura 1.16.

![Diagrama a bloques de la adquisición de datos.](image)

**Figura 1.16** Diagrama a bloques de la adquisición de datos.
Etapa de transductores

Los transductores son dispositivos que convierten una señal física (como por ejemplo presión, temperatura, luz, etc.) en señales eléctricas de voltaje o corriente.

Etapa de transmisión:

Permite enviar las señales de salida de una etapa hacia otra situada en una localización remota. Para distancias no excesivas, es común emplear un bucle de corriente de 4 a 20 mA para la transmisión de las señales.

Etapa de acondicionamiento

Contiene circuitos electrónicos encargados de transformar las señales de sensado en nuevas variables eléctricas.

1.3.8 Puente H

El puente H es un hardware que se considera básicamente como un sistema de conmutación controlado por dos señales digitales de baja potencia, generalmente es utilizada para el control de giro de los motores. Cuando el sistema detecta un 1 lógico en una de sus dos entradas de control y un cero en la otra, este conecta el motor a la fuente de alimentación con determinada polaridad si la señal de control que estaba en 1 pasa a cero y la de cero a uno el PUENTE H conecta la fuente al motor con la polaridad invertida facilitando así el giro en sentido contrario.

El puente H es muy fácil de implementar, trabaja con tan solo 4 transistores 2 NPN y 2 PNP, y una fuente de alimentación. El funcionamiento básico pertenece de la Figura 1.17, Figura 1.18. y la explicación siguiente.

Al poner un transistor en corte y el otro en saturación, se logra que se envié un 1 y 0 lógico a la base de la otra pareja de transistores que permitirá que el motor se polarice con positivo y negativo, si se quiere que el motor gire en otro sentido se deberán de cambiar las condiciones de corte y saturación en los otros transistores y el motor invertirá su giro.
El tipo de transistores a usar dependerá de la potencia que demande el motor y la corriente máxima que se consuma.

![Figura 1.17. Estructura de un puente H.](image)

**Figura 1.17.** Estructura de un puente H.

![Figura 1.18. Los 2 estados básicos del circuito para control de giro.](image)

**Figura 1.18.** Los 2 estados básicos del circuito para control de giro.

### 1.3.9 Compilador *PIC Basic Pro*

El compilador *PicBasic Pro* (*PBP*) es el lenguaje de programación de nueva generación que hace más fácil y rápido el programar microcontroladores de *Microchip Technology* y es el que se usará para la compilación del programa que será alojado en el PiC-18F4550.

El lenguaje *Basic* es mucho más fácil de leer y escribir que el lenguaje ensamblador Microchip y menos complejo que programar en C.

El *PBP* es similar al “*BASIC STAMP II*” y tiene incorporadas varias de las librerías y funciones de las versiones *BASIC STAMP I* y *II*. Como es un compilador real los programas se ejecutan mucho más rápido y pueden ser mayores que sus equivalentes *STAMP*. En la Figura 1.19 se muestra una imagen del ambiente de programación PIC-Basic Pro.
Figura 1.19. Pantalla principal del PIC-Basic Pro (PBP).

*PBP* por defecto crea archivos que corren en un PIC 16F84-04/P con un reloj de 4 Mhz. Solamente muy pocas partes son necesarias como: capacitores de 22 pf para el cristal de 4Mhz un resistor de 4.7K en el terminal/MCLR y una fuente de 5 volt. Otros micros PIC además del 16F84, así como otros osciladores de frecuencias distintas pueden ser usados por este compilador.
Capítulo II

DISEÑO DE INTERFAZ USB 2.0.

Introducción

En este capítulo se describirá la metodología de diseño para implementar la interfaz USB que permitirá a través de un microcontrolador generar aplicaciones de instrumentación virtual, las cuales brindarán del conocimiento tanto práctico y teórico al estudiante para generar sus propias aplicaciones.

2.1 Desarrollo de la interfaz

2.1.1 Microcontrolador 18F4550

Por sus características y aplicaciones; el Microcontrolador que se usará es el PIC 18F4550, ver Tabla II, que pertenece a la familia de los 18FXX5X, entre los que se encuentran los 18F2455/4455/2550/4550; donde estos dos últimos tienen la opción de comunicarse por medio de USB 2.0 a una PC, sin embargo el 18F2550 es de 28 terminales y contiene menos puertos y el 18F4550 es de 40 terminales como se muestra en las Figura 2.1 contiene 4 puertos que permitirán realizar más aplicaciones.

Este micro controlador brinda la posibilidad de analizar señales analógicas debido a sus puertos ADC.

**TABLA II.** Características de PIC 18F4550.

<table>
<thead>
<tr>
<th>Características</th>
<th>Valores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memoria de programación</td>
<td>Flash de 32KB</td>
</tr>
<tr>
<td>Velocidad de CPU</td>
<td>12MIPS</td>
</tr>
<tr>
<td>RAM</td>
<td>2,048 bytes</td>
</tr>
<tr>
<td>EEPROM</td>
<td>256 bytes</td>
</tr>
<tr>
<td>Periféricos de comunicación</td>
<td>1-A/EUSART, 1-MSSP(SP1/I2C)</td>
</tr>
</tbody>
</table>


<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Timers</strong></td>
<td>1x8.bit, 3x16-bit</td>
</tr>
<tr>
<td><strong>ADC</strong></td>
<td>3 puertos</td>
</tr>
<tr>
<td><strong>USB</strong></td>
<td>USB 2.0</td>
</tr>
<tr>
<td><strong>Rango de temperatura</strong></td>
<td>-40 a 85 ºC</td>
</tr>
<tr>
<td><strong>Voltage de operación</strong></td>
<td>2--5.5 V</td>
</tr>
<tr>
<td><strong># de terminales</strong></td>
<td>40</td>
</tr>
</tbody>
</table>

**Figura 2.1** Diagrama de terminales del PIC 18F4550

2.3 **Diagrama de flujo para el programa del PIC18F4550**

La programación del PIC utilizado en el desarrollo del proyecto se realizó a través del diagrama de flujo que se presenta en la Figura 2.2. Las partes principales que integran al diagrama de flujo son:

a) Definición de variables
b) Selección de puertos
c) Declaración de puertos como entrada/salida
d) Inicialización del puerto **USB**
e) Inicio de programa de aplicación en instrumentación a partir de dos subrutinas principales

A continuación se describirá el principio de operación de las dos subrutinas principales.

```
DEFINE OSC 48
DEFINE LOADER_USED 1
USBBufferSizeMax
USBBufferSizeTX
USBBufferSizeRX
USBBuffer
USBBufferCount
ADBuffer
PUERTO_SEL
DATO_SET

USBBuffer
USBBufferCount
ADBuffer
PUERTO_SEL
DATO_SET

DECLARACION DE LOS PUERTOS A,B,C,D.
COMO ENTRADA O SALIDA SEGUN NUESTRAS NECESIDADES.

INICIALIZACION DE LA USB
usbinInit

ProgramStart:

usbservice
gosub DoUSBIn
PUERTO_SEL=USBBUFFER[7]
gosub DOUSBIN
DATO_SET=USBBUFFER[7]
```
Figura 2.2 Diagrama de Flujo para la programación del PIC18F4550
2.4 Subrutinas de lectura y envío de datos 

La subrutina `DoUSBIn` se usa para la interpretación de los datos insertados por medio de uno de los puertos de entrada seleccionados.

Mientras que la subrutina `DoUSBOut` se usa para desplegar la información de salida por medio del Software de la interfaz una vez analizada y procesada la información de entrada.

Ambos algoritmos son indispensables ya que son la base para realizar pruebas de conexión, entrada y salida y es la base del algoritmo que fue programado en el PIC.

En las líneas de abajo se presenta el código utilizado para el desarrollo de las dos subrutinas mencionadas anteriormente.

```c
DoUSBIn:
USBBufCount = USBBuffSizeRX ' TAMAÑO DEL BUFER DE RECEPCION.
USBServ
USBIn 1
USBBuff
USBBuffCount, DoUSBIn' LEE LOS DATOS, SI EL BUS ESTA DISPONIBLE return

DoUSBOut:
USBBuffCount = USBBuffSizeTX ' TAMAÑO DEL BUFER DE RECEPCION
USBServ
USBOut 1,
USBBuff
USBBuffCount, DoUSBOut' SI EL BUS ESTA ACTIVO TRANSMITE LOS DATOS.
```

2.5 Diagrama eléctrico de la **interfaz-USB2**.

En el Diagrama 1 se presentan las conexiones eléctricas para el diseño base de la Interfaz *USB*, en el cual se muestra la gran sencillez del circuito y la poca complejidad de los componentes, objetivo que se planteo al inicio del proyecto.

**Diagrama 1:** Diagrama eléctrico de la interfaz-**USB2**.

2.6 Desarrollo del circuito impreso

Utilizando el programa de diseño de *PCB`s, PCB Wizard* 3.50, se elaboraron las pistas del circuito impreso para la tablilla experimental que se usará como base de aplicaciones. En las Figuras 2.3 y 2.4 se presentan las vistas superior e inferior de la tablilla a implementar.
Figura 2.3 Pistas de la interfaz-USB2

Figura 2.4 Vista preliminar de la interfaz-USB2.
Un vez que se diseñaron las pistas del circuito impreso se procedió a su impresión en papel couche con una impresora LASER, ya que imprime de una forma texturizada apta para después planchar el cuche sobre la cara de cobre de la placa fenólica, la vista final se puede observar en la Figura 2.5.

2.8 Descobrización de placa

El material utilizado para la descobrización fue el cloruro férrico, el cual se preparó de la siguiente manera:

- 250g de percloruro férrico en polvo
- 1Lt de agua tibia
- 10mLt de acido clorhídrico al 10% (opcional)

Se mezcló el percloruro férrico en el litro de agua tibia, se agito la mezcla hasta que se disuelva el percloruro férrico y se tenga una mezcla uniforme.
Se limpió la baquelita con algún tipo de solvente (tiner) y se procedió a perforar la placa para poder insertar los componentes. En la Figura 2.6 se muestra el material para utilizado para este paso.

Figura 2.6 Placa baquelita ferro-clorada y brocas de 1/64,3/64 y 1/32.

Una vez realizado el paso descrito en líneas anteriores, se procedió a perforar la placa, armar y soldar los componentes como se muestra en la Figura 2.7, para evitar posibles cortocircuitos en el lado del cobre, se protegió con una mica protectora. En la Figura 2.8 se muestra la interfaz de adquisición ya terminada.
Figura 2.7 Parte inferior de la placa una vez perforada, soldada y con mica protectora sujetada por tornillos.

Figura 2.8 Parte superior de la placa

Como se puede observar la interfaz desarrollada físicamente, consta de pocos componentes y de tamaño pequeño, lo que garantiza su portabilidad y bajo costo de producción.
Capítulo III

PRUEBAS EXPERIMENTALES

Introducción.

En el capítulo anterior se describió el proceso para implementar la interfaz USB de adquisición de datos utilizando programación en Visual Basic 6.0 para su manejo. En este capítulo se procederá a presentar las diferentes pruebas realizadas, con el fin de comprobar el diseño y realización de la interfaz.

3.1 Pruebas y practicas a realizar

Para comprobar al diseño de la interfaz se procedió a realizar una serie de programas de aplicación, los cuales se resumen a continuación.

1. Conexión entre la interfaz y la PC.

2. Manejo de los puertos A, B, C, D.
   - Puerto A como entrada de datos y como puerto analógico.
   - Puerto B como puerto de salida.
   - Puerto C como puerto de entrada.
   - Puerto D como puerto de salida.

3. Salida de datos

4. Salida de datos en código ASCII

5. Secuencia de bits.

6. Entrada y despliegue del valor decimal del puerto.

7. Entrada de datos y despliegue de información.
8. Control de un motor conectado directamente a la interfaz

9. Control de un motor por RF.


11. Almacenamiento de datos.

12. Volmétro.

### 3.2 Comandos de uso frecuente.

Antes de presentar los resultados obtenidos para cada una de las aplicaciones propuestas se procederá a explicar en forma breve los comandos más frecuentes usados dentro del diseño del software en Visual Basic.

- Dim DATO As Integer: declaración de una variable de tipo entero.
- Dim DATO As String: declaración de una variable como cadena.
- BufferOut(8) = 4: Se selecciona el puerto con el que se trabajará.
- hidWriteEx VendorID, ProductID, BufferOut(0): Instrucciones para la captura o envío de datos
- DATO = BufferOut(8): Carga el valor que se obtuvo del puerto y se lo asigna a la variable dato.
- Label.Caption = DATO: Despliega el valor contenido en DATO
- BufferOut(8) = Asc(“DATO”): Instrucción para cargar a buffer con el valor ASCII de dato.

### 3.3 Conexión entre la interfaz y la PC

Para el despliegue de información, se utilizó el Visual Basic para el diseño del ambiente gráfico, lo que permite saber si la interfaz ha sido conectada o no. En las Figuras 3.1 y 3.2 se muestran las pantallas principales de dicho software.
Figura 3.1 Pantalla con la interfaz desconectada.

Figura 3.2 Interfaz conectada.
A continuación se presenta el código utilizado para determinar la correcta conexión de la interfaz.

********************************************************************************
LECTURA Y ESCRITURA DE LOS BUFFERS
********************************************************************************

Private Const BufferInSize = 8
Private Const BufferOutSize = 8
Dim BufferIn(0 To BufferInSize) As Byte
Dim BufferOut(0 To BufferOutSize) As Byte
Private Sub Form_Load()
   ConnectToHID (Me.hwnd)
End Sub
********************************************************************************
CAMBIA EL CONTROLADOR DE ACUERDO A LA NOTIFICACION DE LLAMADO
********************************************************************************

Public Sub OnChanged()
Dim DeviceHandle As Long
DeviceHandle = hidGetHandle(VendorID, ProductID)
hidSetReadNotify DeviceHandle, True
End Sub
********************************************************************************
EL DISPOSITIVO HID ESTA CONECTADO
********************************************************************************

Public Sub OnPlugged(ByVal pHandle As Long)
If hidGetVendorID(pHandle) = VendorID And hidGetProductID(pHandle) = ProductID Then
   Label1.Caption = "INTERFAZ CONECTADA CORRECTAMENTE"
End If
End Sub
EL DISPOSITIVO HID ESTÁ DESCONECTADO

Public Sub OnUnplugged(ByVal pHandle As Long)
If hidGetVendorID(pHandle) = VendorID And hidGetProductID(pHandle) = ProductID Then
Label1.Caption = "INTERFAZ DESCONECTADA"
End If
End Sub

EN CASO DE UNA LECTURA...

Public Sub OnRead(ByVal pHandle As Long)
If hidRead(pHandle, BufferIn(0)) Then
End If
End Sub

FORMA DE ESCRIBIR ALGUNOS DATOS

Public Sub WriteSomeData()
BufferOut(0) = 0
BufferOut(1) = 0
hidWriteEx VendorID, ProductID, BufferOut(0)
End Sub
3.4 Programa para el manejo de puertos

A continuación se presenta el principio de funcionamiento y el programa utilizado para el manejo de puertos: A, B, C, D. de acuerdo a la siguiente configuración.

- Puerto A como entrada de datos y como puerto analógico.
- Puerto B como puerto de salida.
- Puerto C como puerto de entrada.
- Puerto D como puerto de salida.

3.5 Salida de datos

Para el envío de datos por el puerto B; solo basta con colocar el nombre de la variable que contiene el dato a enviar que para este ejemplo es DATO. Se presenta el código utilizado para esta aplicación.

```vbs
Dim DATO As Integer
Private Sub Timer1_Timer()
  BufferOut(8) = 3
  hidWriteEx VendorID, ProductID, BufferOut(0)
  BufferOut(8) = DATO
  hidWriteEx VendorID, ProductID, BufferOut(0)
End Sub
```

En el Diagrama 2 se muestra el circuito usado para desplegar la información de salida y en la Figura 3.3 se muestra la imagen del software usado para desplegar datos al momento de ser insertados por el usuario.
Diagrama 2: Diagrama para desplegar información

Figura 3.3 Despliegue de información al ser introducido por el usuario.
Para sacar el ASCII de un carácter introducido basta con sustituir la línea
BufferOut(8) = DATO por BufferOut(8) = Asc(“DATO”).

Basándonos en lo anterior elaborar un programa que despliegue una secuencia
binaria, el cual se presentan a continuación.

Dim DATO As Integer
Private Sub Form_Load()
ConnectToHID (Me.hwnd)
Timer1.Interval = 300
DATO = 0
End Sub
Private Sub Timer1_Timer()
Select Case DATO
Case 1
BufferOut(8) = 1
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = 129
hidWriteEx VendorID, ProductID, BufferOut(0)
Case 2
Case 3
Case ####
DATO = 0
End Select
DATO = DATO + 1
End Sub
3.6 Entrada de datos y despliegue de mensaje

En el Diagrama 3 y Figura 3.4 se muestra el hardware y software usado para la entrada y despliegue de datos vía software.

Dim DATO As Integer
Dim COUNT As Integer

Private Sub Form_Load()
    ConnectToHID (Me.hwnd)
    Timer1.Interval = 1000
    Timer2.Interval = 2500
End Sub

EN CASO DE UNA LECTURA...

Public Sub OnRead(ByVal pHandle As Long)
If hidRead(pHandle, BufferIn(0)) Then
    DATO = BufferIn(8)
    Label2.Caption = DATO
    18F4550
End If
End Sub

adquisicion de datos

Private Sub Timer1_Timer()
BufferOut(8) = 3
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = 3
hidWriteEx VendorID, ProductID, BufferOut(0)
End Sub
Private Sub Timer2_Timer()
    contador = 0
    If DATO = 1 Then
        COUNT = COUNT + 1
        Label1.Caption = COUNT
        End If
    If DATO = 2 Then
        COUNT = COUNT + 2
        Label1.Caption = COUNT
        End If

    If DATO = 4 Then
        COUNT = COUNT + 4
        Label1.Caption = COUNT
        End If

    If DATO = 64 Then
        COUNT = COUNT + 64
        Label1.Caption = COUNT
        End If

    If DATO = 128 Then
        COUNT = COUNT + 128
        Label1.Caption = COUNT
        End If
    End Sub
Diagrama 3: Diagrama para la adquisición de datos

Figura 3.4. Programa para la adquisición de datos, visualiza en pantalla el valor binario de la tecla.
Consideraciones:

Para llevar a cabo este programa se tiene que declarar 2 timer’s, uno para la adquisición de los datos y el otro para la manipulación de los datos introducidos.

Recordar que solo se tienen 5 bits (1, 2, 4, 64, 128) de todo el puerto C, es por eso que con este programa solo se visualizarán los valores anteriores.

3.7 Salida de datos en código ASCII

Tomando como referencia las dos aplicaciones presentadas se desarrolló una aplicación para insertar un dato por el puerto C y se muestre un mensaje, o un dato de tipo carácter el cual se puede observar en el Diagrama 3, en la Figura 3.5 se puede observar la interfaz diseñada para esta aplicación.

```vba
Dim dato As Integer
Dim contador As String
Private Sub Form_Load()
  ConnectToHID (Me.hwnd)
  Timer1.Interval = 1000
  Timer2.Interval = 2500
End Sub

Public Sub OnRead(ByVal pHandle As Long)
  If hidRead(pHandle, BufferIn(0)) Then
    dato = BufferIn(8)
    Label2.Caption = dato
  End If
End Sub
```
Private Sub Timer1_Timer()
BufferOut(8) = 3
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = 3
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = 1
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = Asc("contador")
hidWriteEx VendorID, ProductID, BufferOut(0)
End Sub

Private Sub Timer2_Timer()
contador = "SALIDA"
If dato = 1 Then
contador = contador
Label1.Caption = contador
End If

If dato = 2 Then
contador = "DE"
Label1.Caption = contador
End If

If dato = 4 Then
contador = "DATOS"
contador = contador
Label1.Caption = contador
End If
If dato = 64 Then
    contador = "EN"
    contador = contador
    Label1.Caption = contador
End If

If dato = 128 Then
    contador = "CARACTER"
    contador = contador
    Label1.Caption = contador
End If
End Sub

*****TECLAS********** 1 * 2 * 4 * 64 * 128 ************************************
Mensaje a mostrar** SALIDA * DE * DATOS * EN * CARACTER ******************

**Figura 3.5.** Programa que muestra un mensaje al insertar un dato por el puerto C.
3.8 Control de un motor.

3.9 Desarrollo de un puente H

En el Diagrama 4 se muestra el circuito electrónico utilizado para la aplicación del puente H, cuando a las entradas A y B se les aplican “0”s y “1”s se logra controlar el giro del motor.

Diagrama 4: Circuito propuesto por el autor

Para la realización de esta parte del circuito; se propone usar los siguientes componentes electrónicos:

- TIP120 (NPN)
- TIP125 (PNP)
- Opto-acopladores PS2501-2
- Diodos 1N4005,
- Led rojo y ámbar.
Cálculo y propuesta de resistores de acuerdo a la Figura 3.6.

**Figura 3.6.** Análisis de los elementos y sus valores

Como se observa del lado del opto-acoplador de la Figura 3.6 se utiliza un resistor R₁ con valor recomendado por el fabricante de 220Ω.

Para el caso de R₂ y R₃ se observa que forman un divisor de voltaje y cuyos valores se calculan de acuerdo a:

Proponiendo R₂=10K Ω se puede calcular a R₃ de acuerdo a:

\[
V_{R_2} = \frac{R_2}{R_2 + R_3} \times V_{cc} = \frac{10K}{10K + 10K} \times 9 = 4.5V
\]

\[
I_{B} = \frac{V_{cc}}{R_2 + R_3} = \frac{9}{10K + 10K} = \frac{9}{20K} = 0.45mA
\]

Para corroborar el funcionamiento del circuito se procedió a conectar dos led`s 1 y 2 de diferente color para indicar el giro del motor. El circuito presenta los siguientes estados de operación.
Primer estado, cuando A=0 y B=0; el motor no gira ya que el voltaje que se aplica en los transistores Q2 y Q4 polarizan inversamente, no permitiendo flujo de corriente por el motor.

Segundo estado, cuando A=0 y B=1; A, se polariza directamente el transistor (Q1) lo que provoca que a la salida del transistor se tengan 9V que polarizan directamente al Led, al mismo tiempo el nivel alto en B provoca que se polarice directamente el transistor (Q4) y la corriente que viene del led y el motor entren en el colector; como esta polarizado directamente la corriente se va a tierra, provocando giro del motor

Tercer estado, cuando A=1 y B=0; como resultado del análisis se puede mencionar que el funcionamiento para esta parte es el mismo que en el segundo estado solo que en este caso conducen los transistores Q3 y Q2 lo que implica que el motor gire hacia en sentido contrario al caso uno.

Cuarto estado, cuando A=1 y B=1; en este caso se polarizan directamente los transistores Q2 y Q4 y se polarizan inversamente los transistores Q1 y Q3.

En el diagrama 5 se muestra el circuito eléctrico a desarrollar y en el Diagrama 6 se muestra la interconexión del puente H con la interfaz USB.
Diagrama 5: Circuito propuesto para la elaboración del puente H para la interfaz.

Diagrama 6: Diagrama para el control de un motor con puente H.
En la Figura 3.7 se presenta la pantalla de bienvenida para la aplicación de control de giro y se muestra el código utilizado para el manejo del puente H como control de giro de un motor.

![Figura 3.7. Programa de control del motor.](image)

```vbnet
Private Sub Command1_Click()
    BufferOut(8) = 1
    hidWriteEx VendorID, ProductID, BufferOut(0)
    BufferOut(8) = 1
    hidWriteEx VendorID, ProductID, BufferOut(0)
End Sub

Private Sub Command2_Click()
    BufferOut(8) = 1
    hidWriteEx VendorID, ProductID, BufferOut(0)
    BufferOut(8) = 2
    hidWriteEx VendorID, ProductID, BufferOut(0)
End Sub

Private Sub Command3_Click()
    BufferOut(8) = 1
    hidWriteEx VendorID, ProductID, BufferOut(0)
```
3.10 Control de un motor por RF

Tomando como referencia la aplicación de control de giro de motor, se plantea el controlar un motor por RF. En el Diagrama 7 se muestra la modificación en el circuito electrónico anterior para la realización de esta aplicación. En la Figura 3.8 se presenta la pantalla de aplicación y finalmente se muestra el código utilizado.

![Figura 3.8. Programa de control del motor de RF.](image)

Dim contador As Integer  
Dim ulises As Integer  
Private Sub Command1_Click()  
ulises = 255  
BufferOut(8) = 1  
hidWriteEx VendorID, ProductID, BufferOut(0)  
BufferOut(8) = ulises  
hidWriteEx VendorID, ProductID, BufferOut(0)  
End Sub
Private Sub Command2_Click()
ulises = 85
BufferOut(8) = 1
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = ulises
hidWriteEx VendorID, ProductID, BufferOut(0)
End Sub

Private Sub Command3_Click()
ulises = 170
BufferOut(8) = 1
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = ulises
hidWriteEx VendorID, ProductID, BufferOut(0)
End Sub

Private Sub Form_Load()
ConnectToHID (Me.hwnd)
Timer1.Interval = 250
contador = 0
End Sub
Diagrama 7: Diagrama para el control de un motor con puente H y juego de relevadores.
3.11 Joystick

Utilizando el programa para la captura de datos elaborar un programa que controle un Shape1 o círculo azul, con los datos de entrada del puerto C; tomar como referencia el código siguiente y Diagrama 3; cuya pantalla principal se muestra en la Figura 3.9.

```vbnet
Dim dato As Integer
Private Sub Form_Load()
    ConnectToHID (Me.hwnd)
    Timer1.Interval = 100
End Sub
Public Sub OnRead(ByVal pHandle As Long)
If hidRead(pHandle, BufferIn(0)) Then
dato = BufferIn(8)
End If
End Sub
Private Sub Timer1_Timer()
BufferOut(8) = 3
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = 3
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = 1
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = dato
hidWriteEx VendorID, ProductID, BufferOut(0)
Select Case dato
Case 1
Shape1.Top = Shape1.Top - 50
Case 2
Shape1.Top = Shape1.Top + 50
Case 4
Shape1.Left = Shape1.Left + 50
```
Case 64
Shape1.Left = Shape1.Left - 50
Case 128
Shape1.Left = 4080
Shape1.Top = 1920
End Select
End Sub

Figura 3.9 Programa de control del Shape1 (azul) por datos de entrada en el puerto C.

3.12 Notepad

Elaborar un programa que Guarde en un archivo TXT los datos de entrada por medio del teclado de la PC, tanto numérica como alfabéticamente; insertar datos por medio del puerto C de la interfaz, que también son guardados en el TXT y enviar el valor binario del bit por donde entran los datos del puerto C tomando como referencia el código Siguiente:

Dim dato As Integer
Dim temporal As String
Dim Karacters As String
Private Sub Form_Load()
    ConnectToHID (Me hwnd)
    Timer1.Interval = 100
    Timer2.Interval = 100
    End Sub

Public Sub OnRead(ByVal pHandle As Long)
    If hidRead(pHandle, BufferIn(0)) Then
        dato = BufferIn(8)
        Label1.Caption = dato
    End If
    End Sub

Private Sub Timer1_Timer()
    BufferOut(8) = 3
    hidWriteEx VendorID, ProductID, BufferOut(0)
    BufferOut(8) = 3
    hidWriteEx VendorID, ProductID, BufferOut(0)
    Text1.BackColor = &H80000014
End Sub

Private Sub Timer2_Timer()
    'Karacters = "PRIMER CARACTER A MOSTRAR"
    If dato <> 0 Then ' diferente de cero
        temporal = dato
        Text1.Text = Text1.Text + "," + temporal
        Open App.Path + "\documento.txt" For Output As #1
        Print #1, Text1.Text
        Close
    End If
End Sub
If dato = 1 Then
Karacters = "BUENOS DIAS"
Label5.Caption = Karacters
BufferOut(8) = 1
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = dato
hidWriteEx VendorID, ProductID, BufferOut(0)
End If

If dato = 2 Then
Karacters = " ESTA ES"
Label5.Caption = Karacters
BufferOut(8) = 1
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = dato
hidWriteEx VendorID, ProductID, BufferOut(0)
End If

If dato = 4 Then
Karacters = "UNA PRUEBA"
Label5.Caption = Karacters
BufferOut(8) = 1
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = dato
hidWriteEx VendorID, ProductID, BufferOut(0)
End If

If dato = 64 Then
Karacters = " DEL TECLADO"
Label5.Caption = Karacters
BufferOut(8) = 1
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = dato
hidWriteEx VendorID, ProductID, BufferOut(0)
End If

If dato = 128 Then
Karacters = "GRACIAS"
Label5.Caption = Karacters
BufferOut(8) = 1
hidWriteEx VendorID, ProductID, BufferOut(0)
BufferOut(8) = dato
hidWriteEx VendorID, ProductID, BufferOut(0)
End If
End Sub

En la Figura 3.10 se muestra la pantalla principal del software diseñado para la aplicación de esta práctica, en el cuadro se muestra lo que se inserta por medio de la PC y donde aparece PortC muestra lo insertado por medio de la interfaz.

**Figura 3.10.** Programa de un Notepad.
3.13 Elaboración de un Voltmetro

Elaborar un programa que permita la captura y despliegue de un voltaje en el puerto A, a partir del Diagrama 8, el cual muestra el uso del puerto analógico del PIC, que previamente fue configurado.

Diagrama 8. Medidor de voltaje.

Figura 3.11. Programa de un medidor de voltaje.
Prácticas propuestas

Finalmente se presentan algunas prácticas sugeridas para desarrollar, lo que permitirá fortalecer diversos aspectos técnicos y académicos de todo estudiante de ingeniería con interés en el protocolo USB 2.0.

1. Programar un PIC que use una LCD y un teclado matricial y que al momento de introducir un dato éste sea desplegado en la LCD y enviado por uno de los puertos del Microcontrolador y sea recibido por el puerto C de la interfaz, éste almacenará los datos en una variable e irá conformando una cadena que será visualizada en la pantalla del programa.

2. Utilizando un LM35 elaborar un termómetro de -55ºC a 150ºC considerando una conversión de 10mv/ºC.

3. Utilizando el programa que guarda los datos en un archivo TXT, elaborar uno que mande a llamar dicho archivo y ejecute su contenido, el programa evaluará si el dato es carácter o número entero y los mostrará en pantallas diferentes del programa y contará cuantos caracteres y cuantos números se tienen, sacar el valor de cada digito por el puerto B.

4. Utilizando el programa que guarda los datos en un archivo TXT, modificar dicho programa de tal forma que los datos insertados sean enviados a un PIC por el puerto B, programar el PIC de tal forma que guarde los datos enviados por la interfaz y los despliegue en una LCD.

5. Acoplar la práctica 2 y 4 para que se visualice la temperatura tanto en la pantalla de la computadora como en la LCD del PIC.

6. Apoyándose del diagrama 6 y el código de programación para su uso, modificar dicho código de tal forma que permita colocar el tiempo de giro de motor para
cada sentido e intercalarlos los sentidos en 20s a la derecha, 20s parado y 20s a la izquierda.

7. Con el programa del Voltmetro colocar una resistencia de 10kΩ y mediante el programa calcular la corriente que circula en dicho resistor.

4.1 Tendencias para USB 3.0

El 18 de septiembre del 2007 se anunció en la IDF (Intel Development Forum) de San Francisco, el desarrollo de la próxima versión de la interfaz de conexión universal USB (Universal Serial Bus) 3.0, que promete incrementar notablemente el ancho de banda y transferencia de la actual especificación USB 2.0. Hace poco también se anunció que las especificaciones estaban cerca, e incluso se mostraron imágenes del USB 3.0.

En San José, California, en el marco de la “USB Developers Conference”, el USB 3.0 Promoter Group, a cargo de desarrollar las especificaciones de la nueva interfaz y compuesto por compañías como:

- Hewlett-Packard Company
- Intel Corporation
- Microsoft Corporation
- +NEC Corporation
- ST-NXP Wireless
- Texas Instruments

Anunciaron que han completado las especificaciones en su versión 1.0 para la interfaz USB 3.0, y que serán entregadas al USB Implementers Forum (USB-IF), que es la organización encargada de regular y administrar el estándar. La finalización de las especificaciones y la aprobación del USB-IF, permite a los desarrolladores de hardware y dispositivos, el poder acceder a las especificaciones para la implementación en sus futuros productos.
La nueva interfaz (USB 3.0) incrementa por 10 veces el ancho de banda de la especificación actual 480Mbit/s (60MB/s) para USB 2.0, al llevarla hasta los 5Gbps (625MB/s), que data desde ya hace 8 años, cuando fuera lanzado en abril de 2000 y que vino a reemplazar a la especificación USB 1.1 (12Mb/s) introducida en el 1998, la que a su vez reemplazo al USB 1.0 (1.5Mb/s) lanzada en 1998.

Respecto a otros temas de arquitectura, USB 3.0 se conforma por cuatro capas (layer), denominadas “Protocol Layer” capa del protocolo, “Link Layer” Capa de enlace, “Physical Layer” (Capa física) y Device Framework, esto y un diagrama para el cable USB 3.0

USB 3.0 además mantendrá la compatibilidad con la especificación actual y el mismo diseño de conectores, aunque tecnológicamente será bastante diferente, ya que mientras USB 2.0 está basado en la administración de flujo de datos de manera unidireccional, (con negociación para el envío de datos por el bus direccional), USB 3.0 soporta flujo de datos simultáneos de manera bidireccional, esto a través de un cable de tipo “dual-simplex four-wire” con señal diferenciada, en lugar de cableado “half-duplex two wire differential” de la especificación actual. A nivel de consumo, USB 3.0 incluye una nueva característica de administración de energía que soporta estados: Idle (sin carga), Sleep (Reposo) y Suspend (Suspendido), esto con el fin de hacer un óptimo uso de la energía de la interfaz. En este mismo tema, la nueva interfaz proveerá la energía suficiente para que dispositivos como discos duros externos de alta velocidad, no requieran de fuente de alimentación adicional, sino sólo el propio cable USB, algo que sin duda simplificará las cosas.
USB 3.0, será conocido comercialmente como SuperSpeed USB y los primeros controladores aparecerán en la segunda mitad del 2009, mientras que productos comerciales aparecerán en el 2010. Los primeros dispositivos USB SuperSpeed serán dispositivos de almacenamiento Flash, discos duros externos, cámaras digitales, y reproductores de música digitales.
Conclusiones

Durante el desarrollo de esta tesis surgieron varios contratiempos, que a lo largo del desarrollo de cada aplicación se fueron subsanando, lo que me deja satisfecho como profesionista y egresado del I.P.N.

Cada una de estas aplicaciones desarrolladas (y prácticas propuestas) ha sido elaborada de tal forma que sean lo más sencillas y económicas para que cualquier persona que desee utilizar este practicario como soporte a sus clases en el área de ingeniería en comunicaciones y electrónica y afines.

Uno de los retos más interesantes fue el acoplamiento de los relevadores de tal forma que manipularan el Transmisor del motor RF mediante un puente H, ya que se tenía pensado utilizar otro tipo de circuito de control.

El número de prácticas y aplicaciones depende de la imaginación que tenga la persona para el uso de la interfaz-USB2, ya que al realizar un análisis más a fondo su uso, se pueden desarrollar diferentes módulos de potencia e implementar diferentes funciones de aplicación tanto en el área electrónica, instrumentación, comunicaciones y otras disciplinas.

Como trabajos futuros se pueden realizar las siguientes aplicaciones:

- Monitoreo de temperatura en una empresa y el manejo del aire acondicionado mediante la manipulación de extractores.
- Apagado y encendido de equipo industrial
- Adquisición de información.
- Medición de presión, velocidad, torsión.
- Manejo de compuertas, electro válvulas.
Bibliografía.

Previa autorización del titular del Copyright; Henry Antonio Mendiburu Díaz.

- “Instrumentación Virtual Industrial”, Copyright by Henry Mendiburu Díaz

Derechos reservados conforme a Ley (Indecopi/Perú), Julio/2006, 180 páginas (http://hamd.galeon.com)


- Helfrick, A. D. Cooper, W.D. “Modern electronic instrumentation and measurement techniques"Prentice-Hall, USA, 1990


- Wolf, S. Smith, R. “Guía para mediciones electrónicas y prácticas de laboratorio” Prentice Hall Hispanoamericana, 1992


Agradecimientos.

Se destaca el apoyo del Ing. Joel Castillo por sus valiosos comentarios y apoyo a la realización de este proyecto.

Henry Mendiburu Díaz

MICROCHIP

PIC18F2455/2550/4455/4550

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

Universal Serial Bus Features:
- USB V2.0 Compliant SIE
- Low-speed (1.5 Mbps) and full-speed (12 Mbps)
- Supports control, interrupt, isochronous and bulk transfers
- Supports up to 32 endpoints (16 bidirectional)
- 1-kbyte dual access RAM for USB
- On-board USB transceiver with on-chip voltage regulator
- Interface for off-chip USB transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

Peripheral Highlights:
- High current sink/source: 25 mA/25 mA
- Three external interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
  - Capture is 16-bit, max. resolution 8.25 ns (Tcy/16)
  - Compare is 16-bit, max. resolution 100 ns (Tcy)
- PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
  - Multiple output modes
  - Selectable polarity
  - Programmable dead-time
  - Auto-Shutdown and Auto-Restart
- Addressable USART module:
  - LIN bus support
  - Master Synchronous Serial Port (MSSP) module
    - supporting 3-wire SPI™ (all 4 modes) and PC™
    - Master and Slave modes
  - 10-bit, up to 13-channels Analog-to-Digital Converter
    - module (A/D) with programmable acquisition time
  - Dual analog comparators with input multiplexing

Special Microcontroller Features:
- C compiler optimized architecture with optional extended instruction set
- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle data EEPROM memory typical
- Flash data EEPROM retention: > 40 years
- Self-programmable under software control
- Priority levels for interrupts
- 8 x 8 Single Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 41 ns to 131s
  - Programmable Code Protection
- Single-supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Wide operating voltage range (2.0V to 5.5V)

<table>
<thead>
<tr>
<th>Device</th>
<th>Program Memory</th>
<th>Data Memory</th>
<th>I/O</th>
<th>10-bit AD (ch)</th>
<th>CCP/PWM (P)</th>
<th>SPP</th>
<th>MSSP</th>
<th>Master/PC</th>
<th>BAUDART</th>
<th>Timers bit/16-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC18F2455</td>
<td>24k</td>
<td>2238</td>
<td>2048</td>
<td>256</td>
<td>24</td>
<td>10</td>
<td>20</td>
<td>No</td>
<td>Y</td>
<td>1</td>
</tr>
<tr>
<td>PIC18F2550</td>
<td>32k</td>
<td>16384</td>
<td>2048</td>
<td>256</td>
<td>24</td>
<td>10</td>
<td>20</td>
<td>No</td>
<td>Y</td>
<td>1</td>
</tr>
<tr>
<td>PIC18F4455</td>
<td>24k</td>
<td>2238</td>
<td>2048</td>
<td>256</td>
<td>35</td>
<td>13</td>
<td>1/1</td>
<td>Yes</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PIC18F4550</td>
<td>32k</td>
<td>16384</td>
<td>2048</td>
<td>256</td>
<td>35</td>
<td>13</td>
<td>1/1</td>
<td>Yes</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
TIP120/121/122
TIP125/126/127

COMPLEMENTARY SILICON POWER DARLINGTON TRANSISTORS

- STMicroelectronics PREFERRED SALESTYPES

DESCRIPTION
The TIP120, TIP121 and TIP122 are silicon Epitaxial-Base NPN power transistors in monolithic Darlington configuration mounted in Jedeo TO-220 plastic package. They are intended for use in power linear and switching applications. The complementary PNP types are TIP125, TIP120 and TIP127, respectively.

INTERNAL SCHEMATIC DIAGRAM

ABSOLUTE MAXIMUM RATINGS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCEO</td>
<td>Collector-Base Voltage (IE = 0)</td>
<td>60</td>
<td>V</td>
</tr>
<tr>
<td>VCEO</td>
<td>Collector-Emitter Voltage (IE = 0)</td>
<td>80</td>
<td>V</td>
</tr>
<tr>
<td>VCEO</td>
<td>Emitter-Base Voltage (IE = 0)</td>
<td>100</td>
<td>V</td>
</tr>
<tr>
<td>IC</td>
<td>Collector Current</td>
<td>5</td>
<td>A</td>
</tr>
<tr>
<td>ICM</td>
<td>Collector Peak Current</td>
<td>8</td>
<td>A</td>
</tr>
<tr>
<td>IB</td>
<td>Base Current</td>
<td>0.1</td>
<td>A</td>
</tr>
<tr>
<td>P(T)</td>
<td>Total Dissipation at Tcase ≤ 25 °C</td>
<td>65</td>
<td>W</td>
</tr>
<tr>
<td>T(T)</td>
<td>Storage Temperature</td>
<td>-65 to 150</td>
<td>°C</td>
</tr>
<tr>
<td>Tj</td>
<td>Max. Operating Junction Temperature</td>
<td>150</td>
<td>°C</td>
</tr>
</tbody>
</table>

* For PNP types voltage and current values are negative.
DATA SHEET

PHOTOCOUPLER

PS2501-1,-2,-4, PS2501L-1,-2,-4

HIGH ISOLATION VOLTAGE
SINGLE TRANSISTOR TYPE
MULTI PHOTOCOUPLER SERIES

DESCRIPTION

The PS2501-1, -2, -4 and PS2501L-1, -2, -4 are optically coupled isolators containing a GaAs light emitting diode and an NPN silicon phototransistor.

The PS2501-1, -2, -4 are in a plastic DIP (Dual In-line Package) and the PS2501L-1, -2, -4 are lead bending type (Gull-wing) for surface mount.

FEATURES

- High isolation voltage \((B_V = 5 \text{ kV})\)
- High collector to emitter voltage \((V_{CEO} = 80 \text{ V})\)
- High-speed switching \((\beta = 3 \mu s \text{ TYP.}, \beta = 5 \mu s \text{ TYP.})\)
- Ordering number of taping product: PS2501L-1-E3, E4, F3, F4, PS2501L-2-E3, E4
- UL approved: File No. E72422 (S)

APPLICATIONS

- Power supply
- Telephone/FAX
- FA/OA equipment
- Programmable logic controller
Features
- Low forward voltage drop.
- High surge current capability.

General Purpose Rectifiers (Glass Passivated)

Absolute Maximum Ratings

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{RSM}$</td>
<td>Peak Repetitive Reverse Voltage</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>$I_{AK}$</td>
<td>Average Rectified Forward Current</td>
<td>1.0</td>
<td>A</td>
</tr>
<tr>
<td>$I_{RSM}$</td>
<td>Non-repetitive Peak Forward Surge Current</td>
<td>30</td>
<td>A</td>
</tr>
<tr>
<td>$T_{MH}$</td>
<td>Storage Temperature Range</td>
<td>-55 to +175</td>
<td>°C</td>
</tr>
<tr>
<td>$T_{J}$</td>
<td>Operating Junction Temperature</td>
<td>-55 to +175</td>
<td>°C</td>
</tr>
</tbody>
</table>

These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

Thermal Characteristics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_D$</td>
<td>Power Dissipation</td>
<td>3.0</td>
<td>W</td>
</tr>
<tr>
<td>$R_{JJA}$</td>
<td>Thermal Resistance, Junction to Ambient</td>
<td>50</td>
<td>°C/W</td>
</tr>
</tbody>
</table>

Electrical Characteristics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Device</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_F$</td>
<td>Forward Voltage @ 1.0 A</td>
<td>1.1</td>
<td>V</td>
</tr>
<tr>
<td>$I_L$</td>
<td>Maximum Full Load Reverse Current, Full Cycle</td>
<td>30</td>
<td>µA</td>
</tr>
<tr>
<td>$I_R$</td>
<td>Reverse Current @ rated $V_F$, $T_A = 25°C$</td>
<td>5.0</td>
<td>µA</td>
</tr>
<tr>
<td>$C_T$</td>
<td>Total Capacitance $V_F = 4.0 V$, $f = 1.0$ MHz</td>
<td>15</td>
<td>pF</td>
</tr>
</tbody>
</table>