



MICROCHIP

Regional Training Centers

Diseñando aplicaciones USB

Objectives

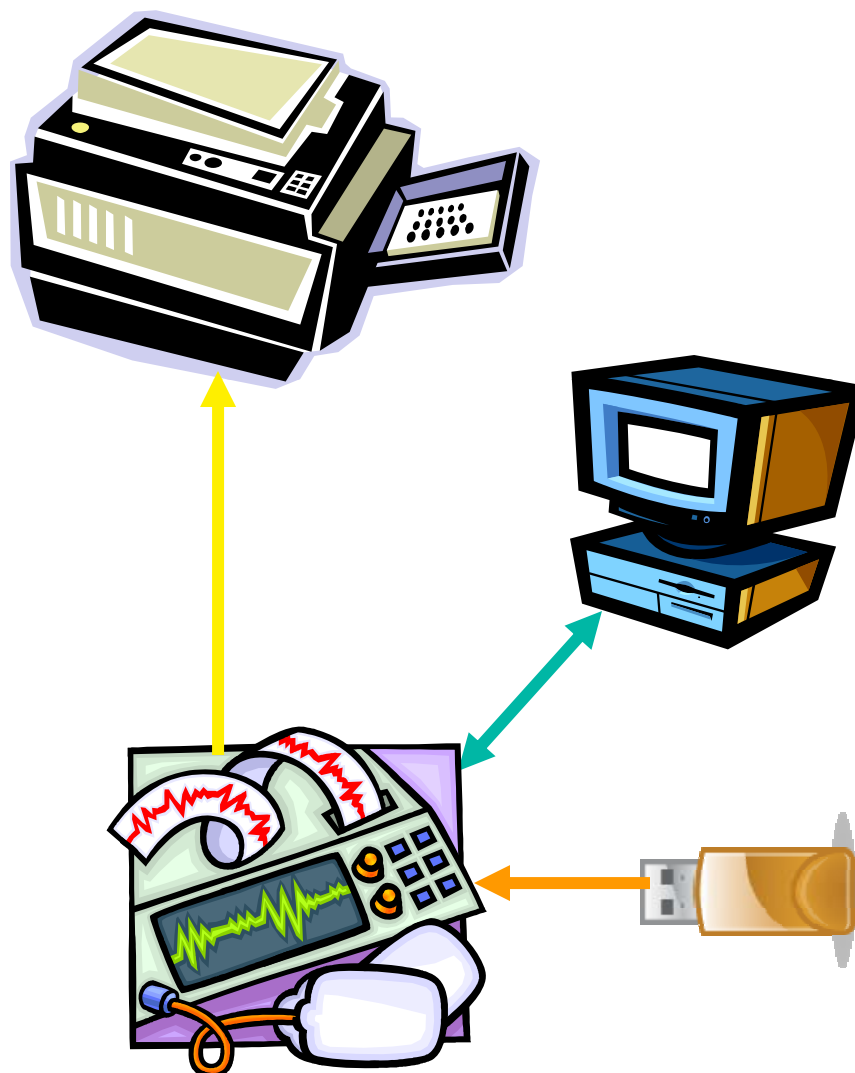
- **Despues de esta clase usted podrá**
 - seleccionar una apropiada solución de conectividad USB para su aplicación de control embebida (incrustada)
 - seleccionar el MCU PIC mas conveniente para ese propósito
 - seleccionar la plataforma de desarrollo apropiada para su aplicación
 - decida que manos de Microchip en el entranamiento le asistirán en el proceso de desarrollo.

Agenda

- **El desafío de conectividad para Diseñar Sistemas Embebidos**
- **Revisión sobre USB**
- **Principios de Operación**
- **Soluciones en Microcontroladores PIC® para USB de MCHP**
- **Usando el Microchip USB Framework**
- **Drivers de PC para USB**
- **Microchip USB Herramientas de desarrollo**
- **Algunas Soluciones USB**
- **Resumen y Recusos**

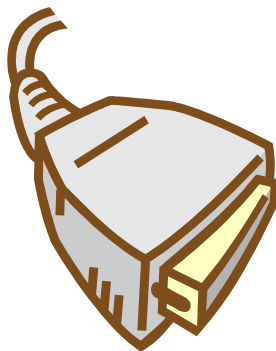
El Desafío de la Conectividad

- **Muchas aplicaciones embebidas requieren conectividad para equipo externo**
 - señales de control de intercambio
 - descarga de datos
 - carga de datos y nuevo firmware
 - diagnósticos



El desafío de la conectividad

- **Ultima Solución: Conectividad RS232**
- **Nuevo desafío: No RS232 o puertos paralelos o laptops o computadoras o impresoras**
 - Solo USB, Ethernet y/o WLAN



Acceso a Periféricos de PC

- **Vía antigua**
- **Periféricos de PC:**
 - Espacio de direcciones I/O mapeado en la memoria del 80X86
 - Asigna una línea específica de IRQ
 - Asigna un canal de DMA específico
- **Acceso Directo (ISA, PCI, PCMCIA)**
- **Vía USB**
- **Periféricos de PC:**
 - Mapeado de 127 dispositivos dentro de un espacio virtual de direccionamiento
 - No utiliza ningún recurso del PC I/O, IRQ o DMA
- **Acceso indirecto usando la interfaz programable provista por el driver del dispositivo**

El desafío de la conectividad

- **Alternativa ?**
 - Subsistir usando viejas computadoras
 - Usar adaptadores RS232 a USB
- **Solución: Implementar conectividad USB en su aplicación**



Agenda

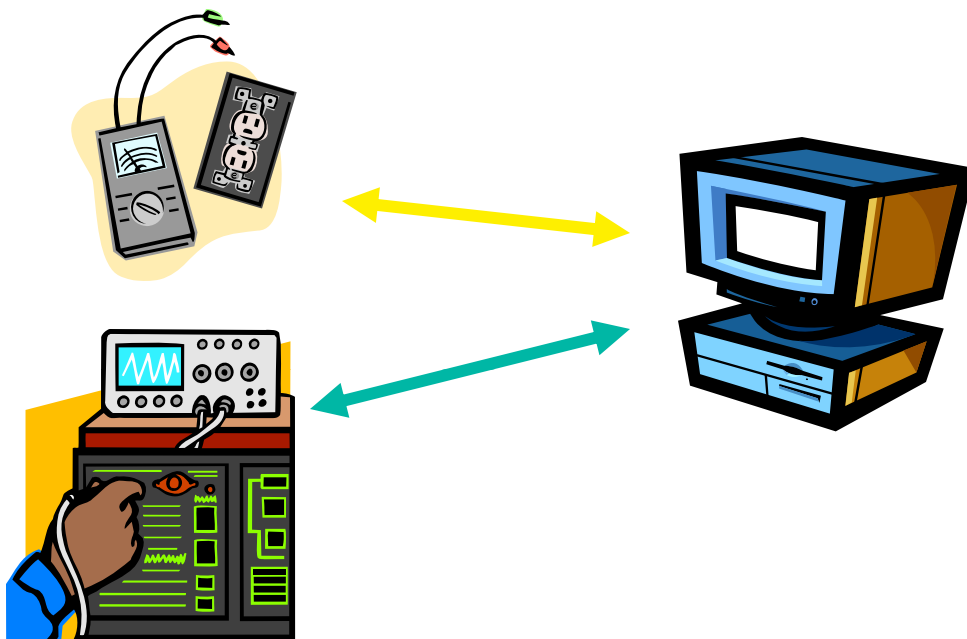
- El desafío de conectividad para Diseñar Sistemas Embebidos
- **Revisión sobre USB**
- Principios de Operación
- Soluciones en Microcontroladores PIC® para USB de MCHP
- Usando el Microchip USB Framework
- Drivers de PC para USB
- Microchip USB Herramientas de desarrollo
- Algunas Soluciones USB
- Resumen y Recusos

USB Descripción

- **USB fué co-desarrollado por un consocio de companias....**
 - Compaq/HP, Intel, Microsoft, LSI, NEC etc.
 - ...Buscando crear un solo Bus que ...
 - **Conecte muchos dispositivos sobre el mismo bus**
 - **Fuerafacil para adicionar/remover dispositivos periféricos conectados sobre el bus**



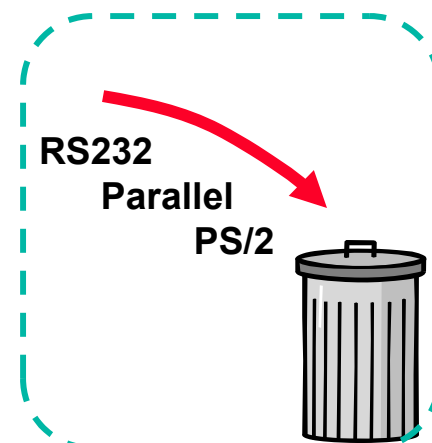
Ventajas de la conectividad USB



**Extienden la funcionalidad
de su computadora!**

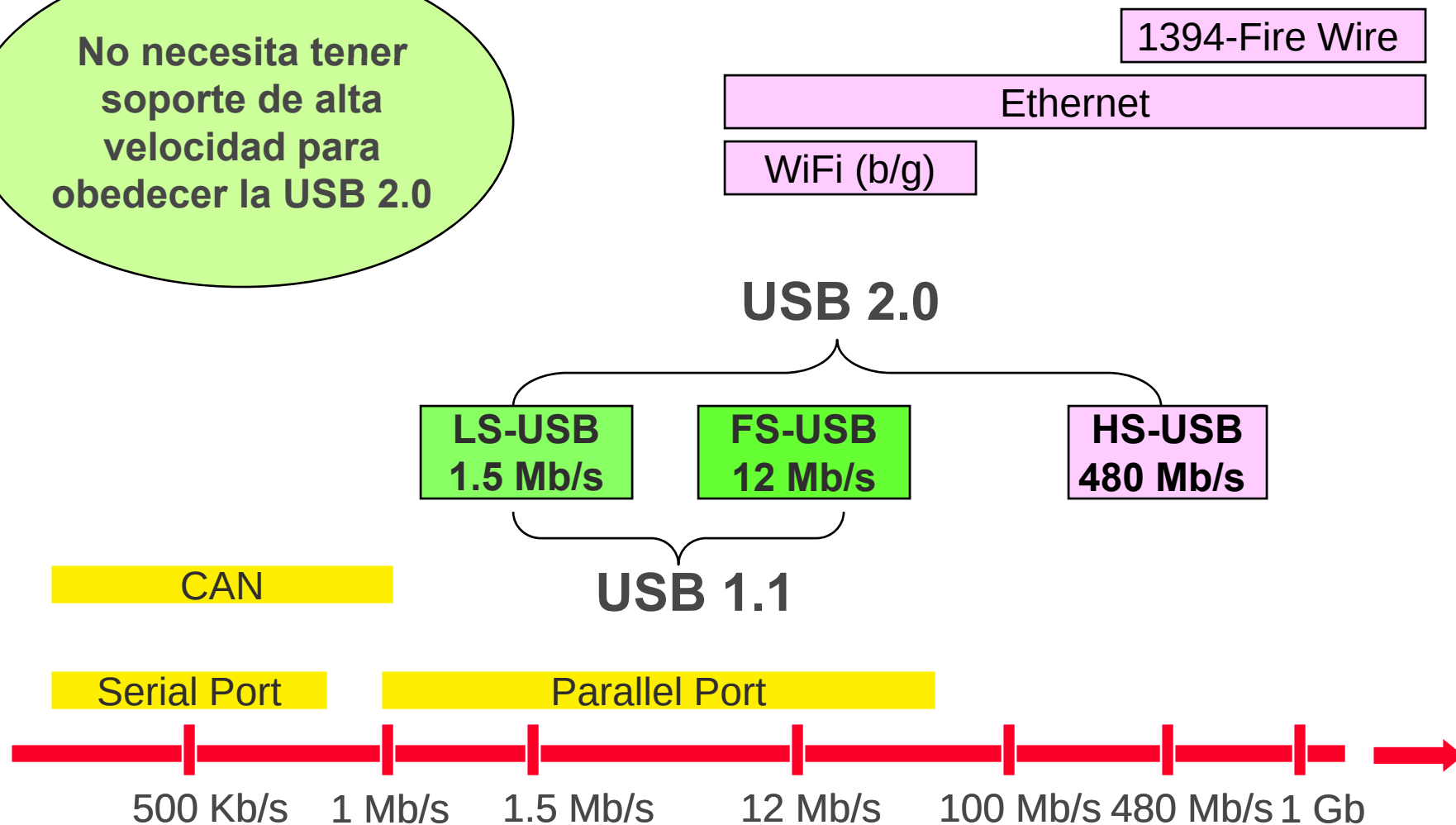
**Data Analysis,
Data Logging,
Firmware Updates,
Diagnostics,
Embedded Applications!**

- Auto detección & configuración (Plug&Play)
- Facil expansión usando hubs
- Bus alimentado – reduce los transformadores de pared
- Datos protegidos por CRC , paquetes malos son vueltos a enviar.
- Tres Velocidades:
Low- 1.5, Full- 12, High- 480 Megabits / second

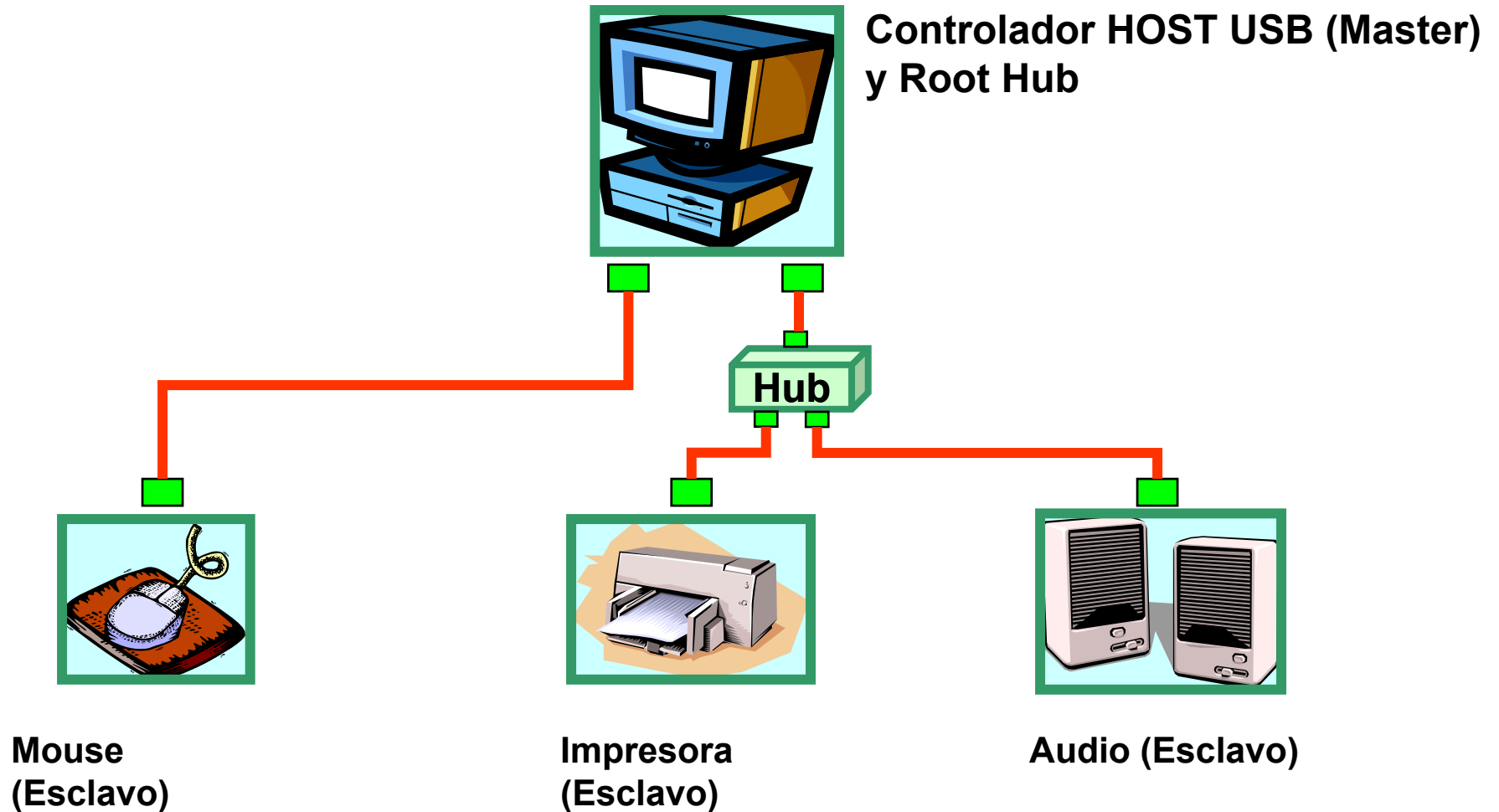


Comparación de Buses y Velocidad

No necesita tener soporte de alta velocidad para obedecer la USB 2.0

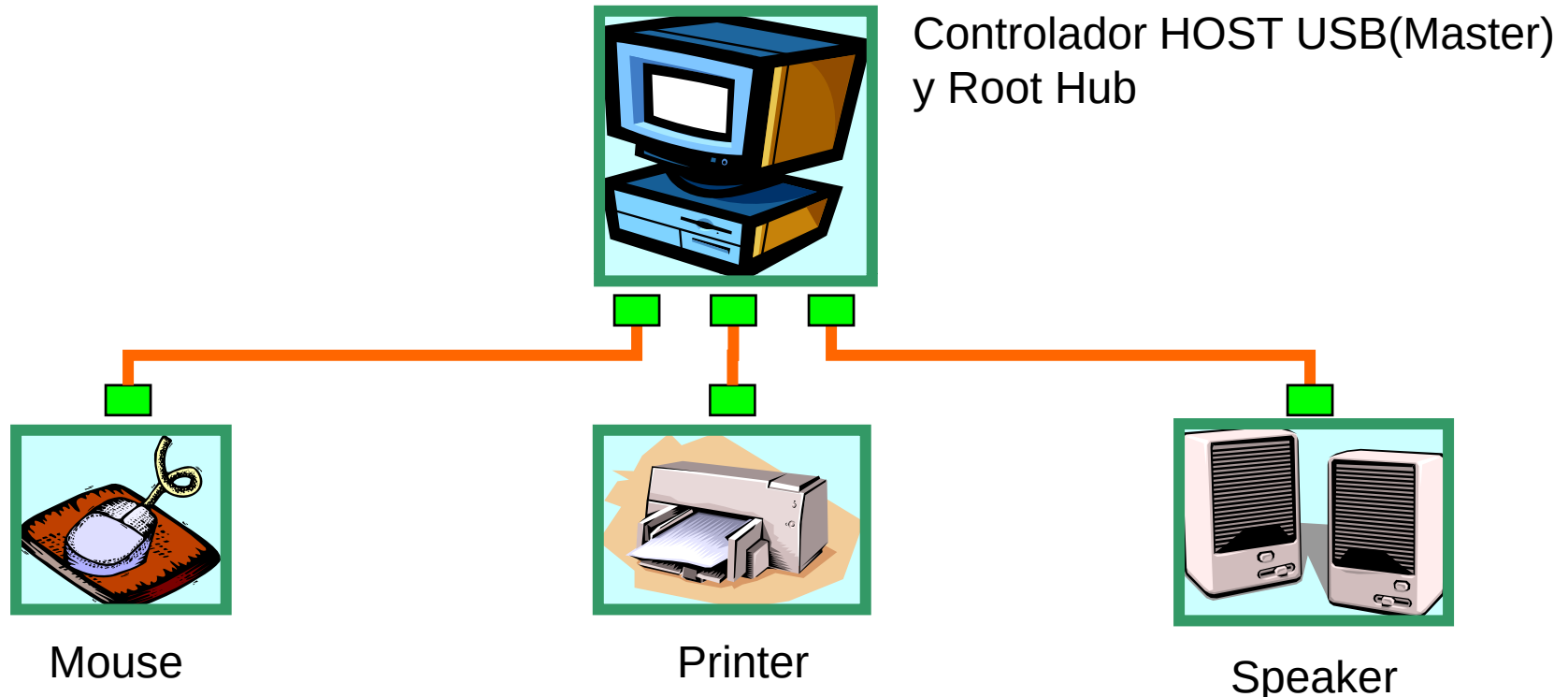


Elementos de un sistema USB



Mecanismo de transferencia de datos USB

- USB es un “Single Master + Multiple Slaves” Bus consultado



Todas las transacciones son iniciadas por el Host

Dispositivos no pueden llamar a otro

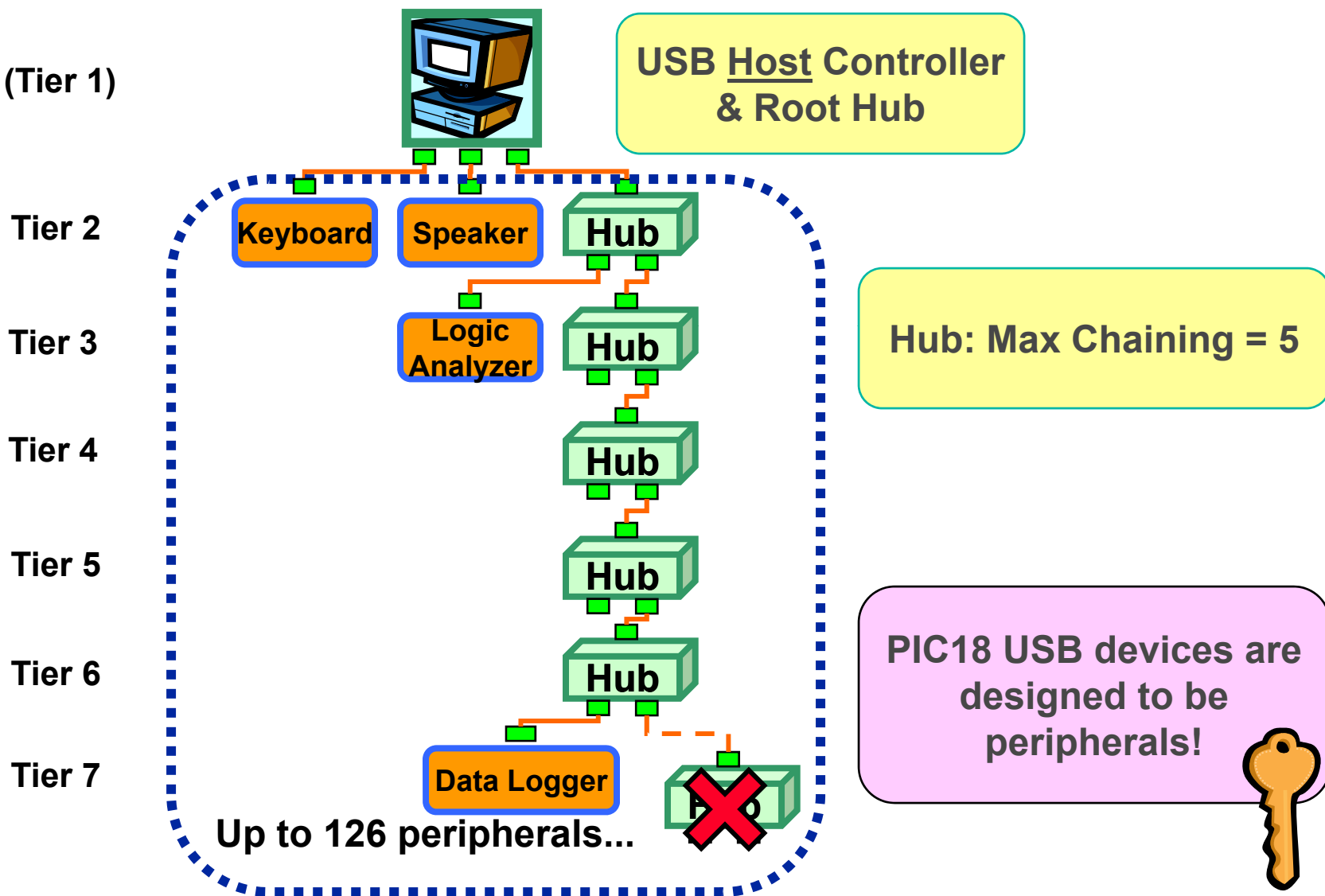
El ancho de banda del USB es compartido

Dispositivos responden al HOST

Cada dispositivo es consultado por el Host

Topología Física del Bus

Host (Tier 1)



USB Host

- **Maneja y controla el BUS**
- **Inicializa todas las transacciones**
- **Automaticamente detecta todos los dispositivos removidos e insertados en el BUS**
- **Enumera todos los dispositivos conectados y los conecta con los drivers**

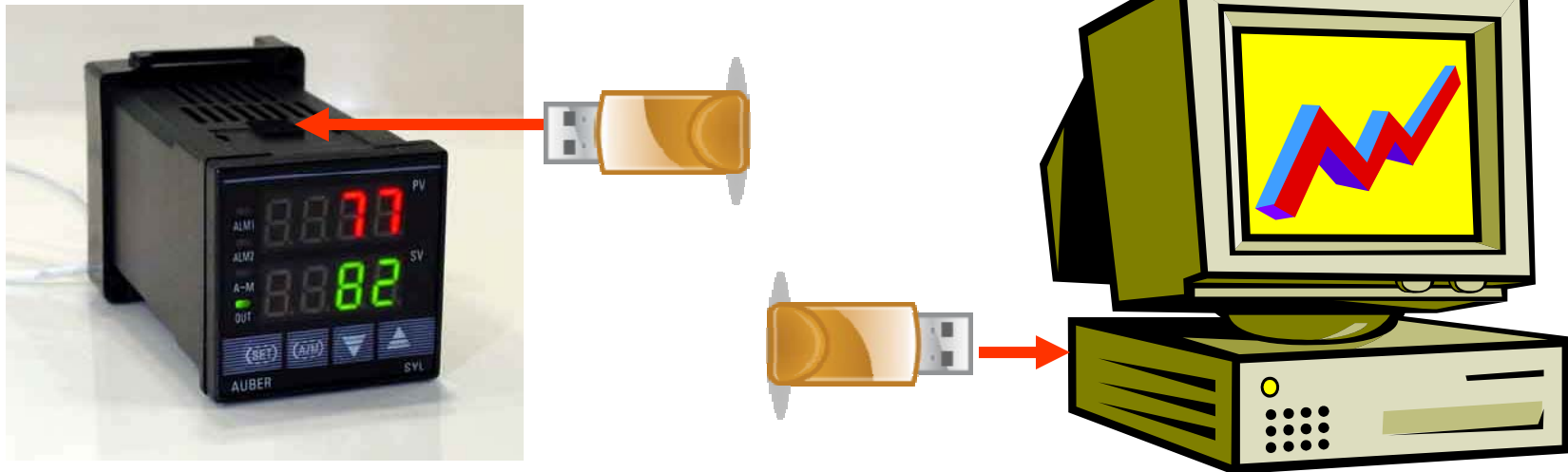


Requerimientos típicos del MCU para ser HOST

- **Procesamiento de alto rendimiento con MCU de 32/64 bit funcionando con un sistema operativo en tiempo real como Windows**
- **Aplicaciones con interfaz grafica de usuario**
- **Identificación de Drivers USB y enumeración de dispositivos USB**
- **USB Host Controller, funcionando a Full Speed (12Mbps) or High Speed(480Mbps).**

Host Embebido

- Aplicación Embebida conectada a uno dispositivo periférico USB
- Aplicación Embebida no conectad a otros Host
- Ventajas: firmware embebido mas pequeño, menos complejo
- Ejemplo: Data Logger de temperatura remoto
 - Descarga datos al USB Flash Drive
 - Actúa como Host cuando es conectado al Flash Drive but ...
 - No se conecta directamente al PC Host



Requerimientos típicos para el Host Embebido

- **Peripherals to service real world inputs and outputs**
 - A/D converters, Comparators etc
 - Serial Interfaces like SPI, I2C, UART
 - PWMs, Timers and I/O lines
- **High throughput 16/32 bit MCU**
- **Full Speed (12Mbps) USB OTG transducer with data input/output buffers**
- **Host Software Drivers to identify and enumerate USB Peripheral Devices**

Dispositivos Periféricos USB

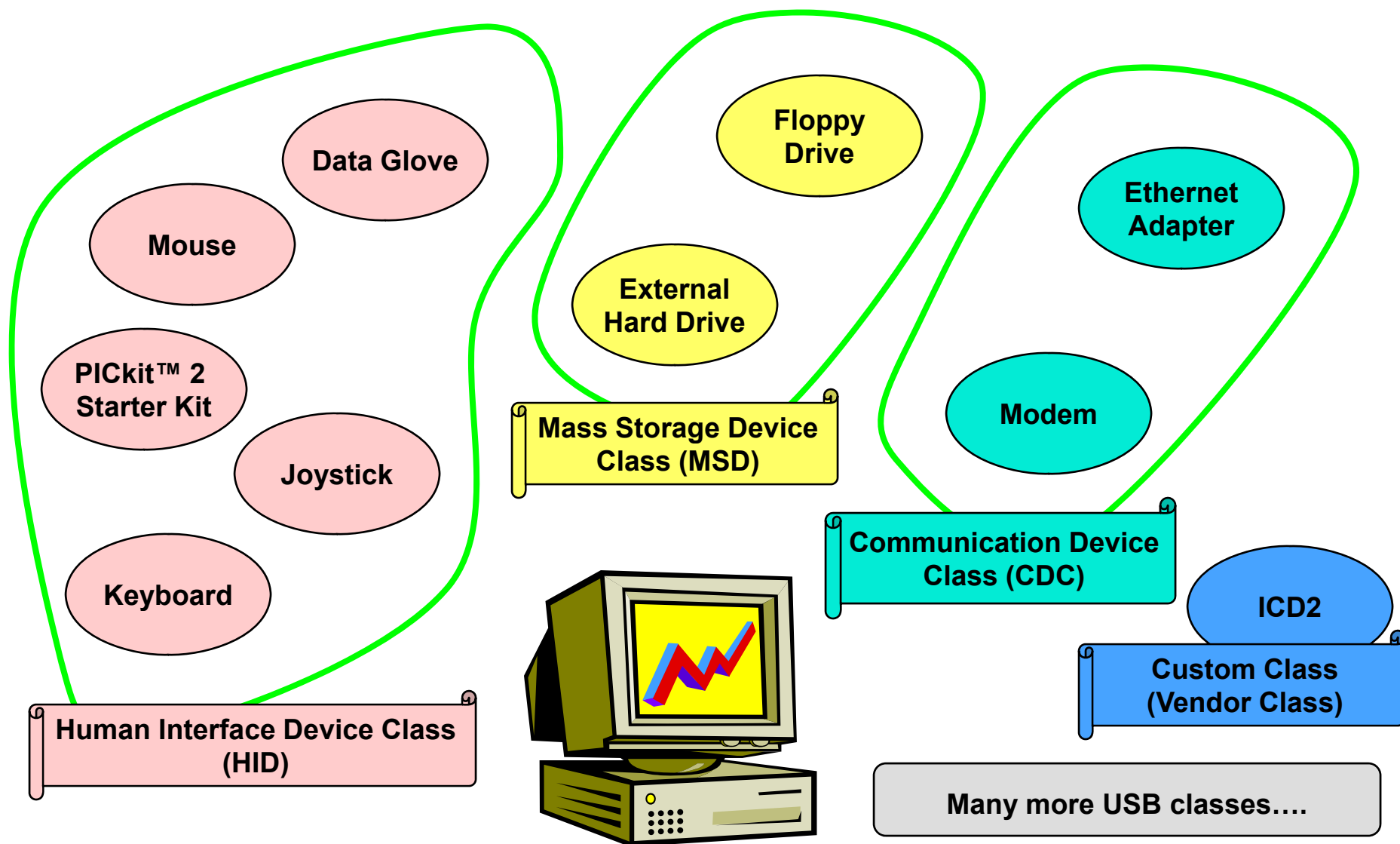
- **Responde al Host, no puede iniciar transacciones**
- **Requiere drivers para ser reconocido por el Host**
- **Hardware/Firmware para responder al Host**



Características típicas de los periféricos del MCU

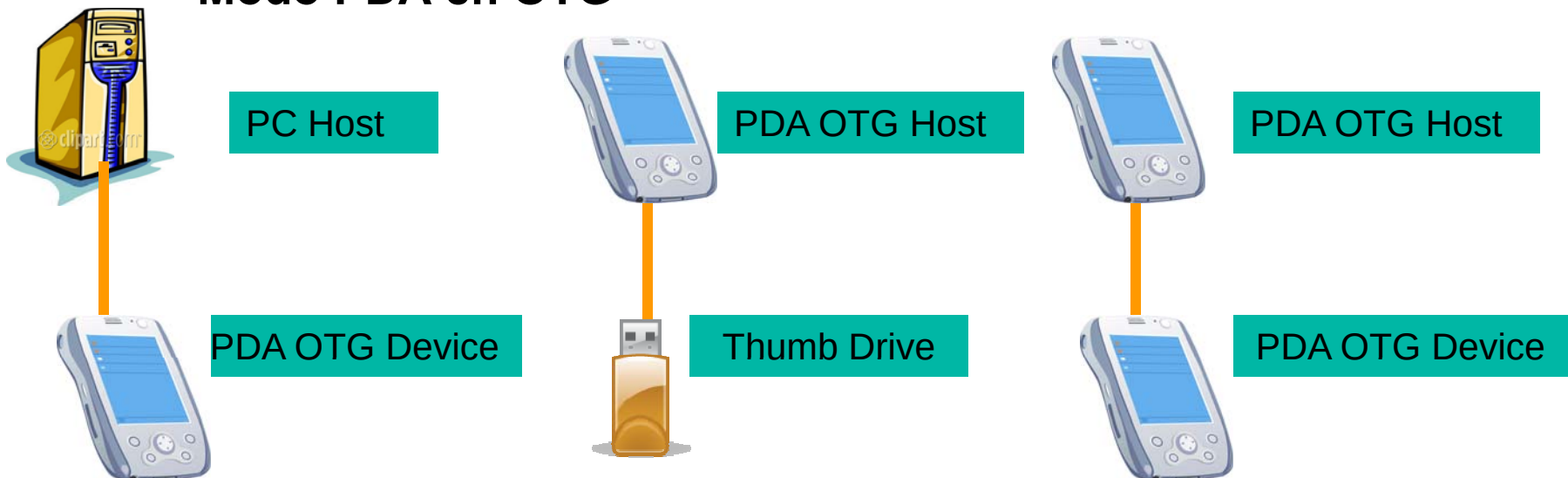
- **Periféricos para el servicio de entrada y salida con el mundo real**
 - conversores A/D, Comparadores etc
 - Interfaces Serie como SPI, I2C, UART
 - PWMs, Timers y líneas I/O
 - Full Speed (12Mbps) USB transducer with data input/output buffers
- **Dispositivos USB y Firmware para interface de periféricos**
- **Procesamiento para el servicio requerido por el USB Full speed**

Clases de dispositivos periféricos USB



USB On-The-Go (OTG) y Doble papel

- **USB On-The-Go (OTG) le permite al Dispositivo operar como Host o Dispositivo**
 - PDA (Dispositivo) conectado al PC (Host)
 - PDA (Host) conectado al Thumb drive (Dispositivo)
 - PDA en Modo Dual Role
- **PDA conectado a PDA, Host y dispositivo pueden cambiar papeles**
 - **Modo PDA en OTG**

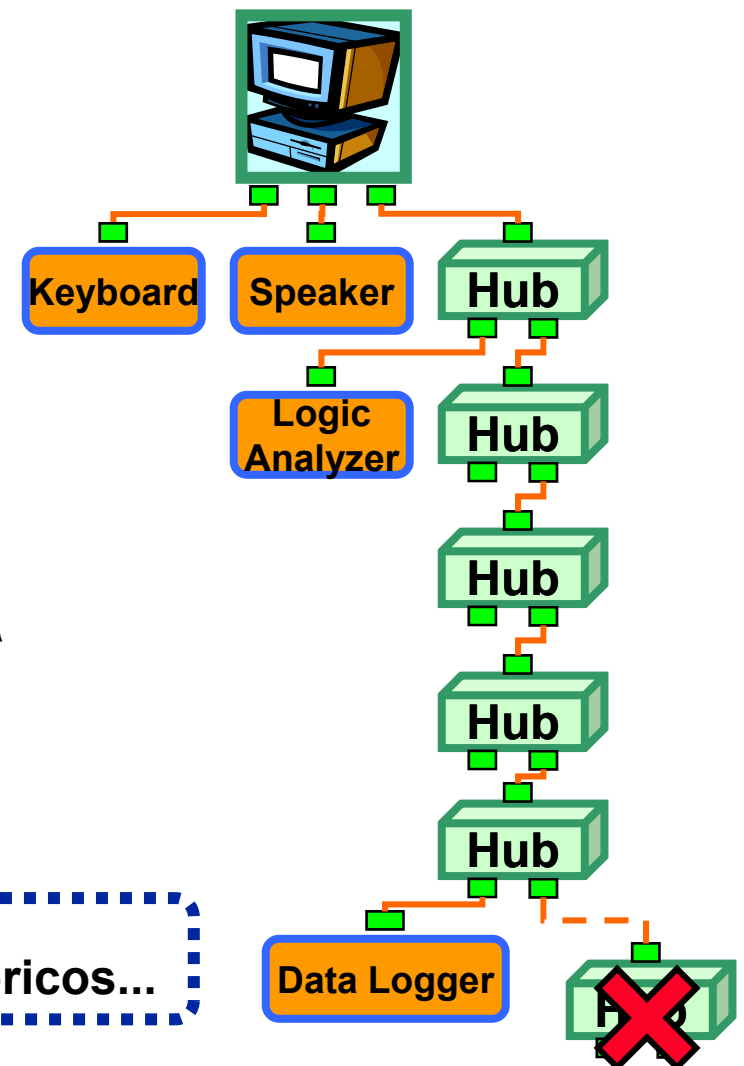


Requerimientos típicos para Dispositivos OTG

- **Periféricos para servicio de entradas y salidas al mundo real**
 - A/D converters, Comparators etc
 - Serial Interfaces like SPI, I2C, UART
 - PWMs, Timers and I/O lines
- **MCU de alto procesamiento 16/32 bit**
- **Full Speed (12Mbps) USB OTG transductor con buffers de datos entrada/salida**
- **Host Software Drivers para identificar y enumerar Dispositivos Periféricos USB**
- **Software para funcionar como Dispositivo cuando se conecta al Host**

Hub USB

- Conecta dispositivos adicionales al Bus USB
- Maximo 5 Hubs sobre Root Hub
- Provee alimentación al conectar dispositivos
- La mayoría de Hubs usa controladores ASIC



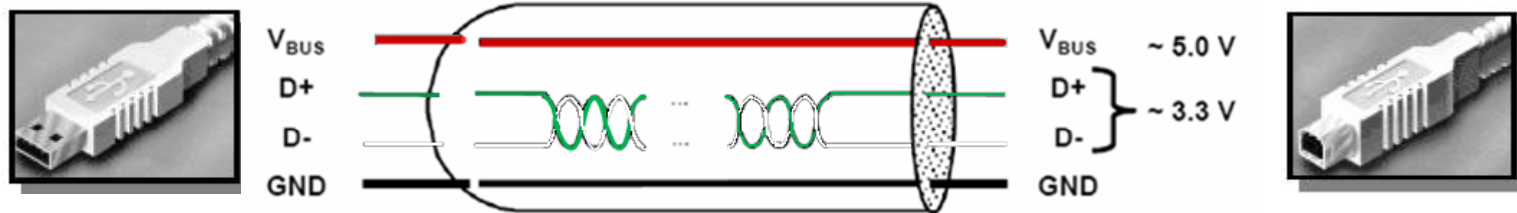
hasta 126 periféricos...

Resumen de Elementos USB

- **Full USB Host**
 - PC
- **Embedded USB Host**
 - Drivers No Instalables
 - Requerimientos de Firmware reducidos
- **Dispositivos periféricos USB**
 - Mouse, Keyboard, Thumb drive, etc.
- **Doble papel /OTG**
 - Puede switchear Roles
 - Puede ser OTG o apenas dos roles

USB Interfaz física

5V Power and GND Lines



Host Connector

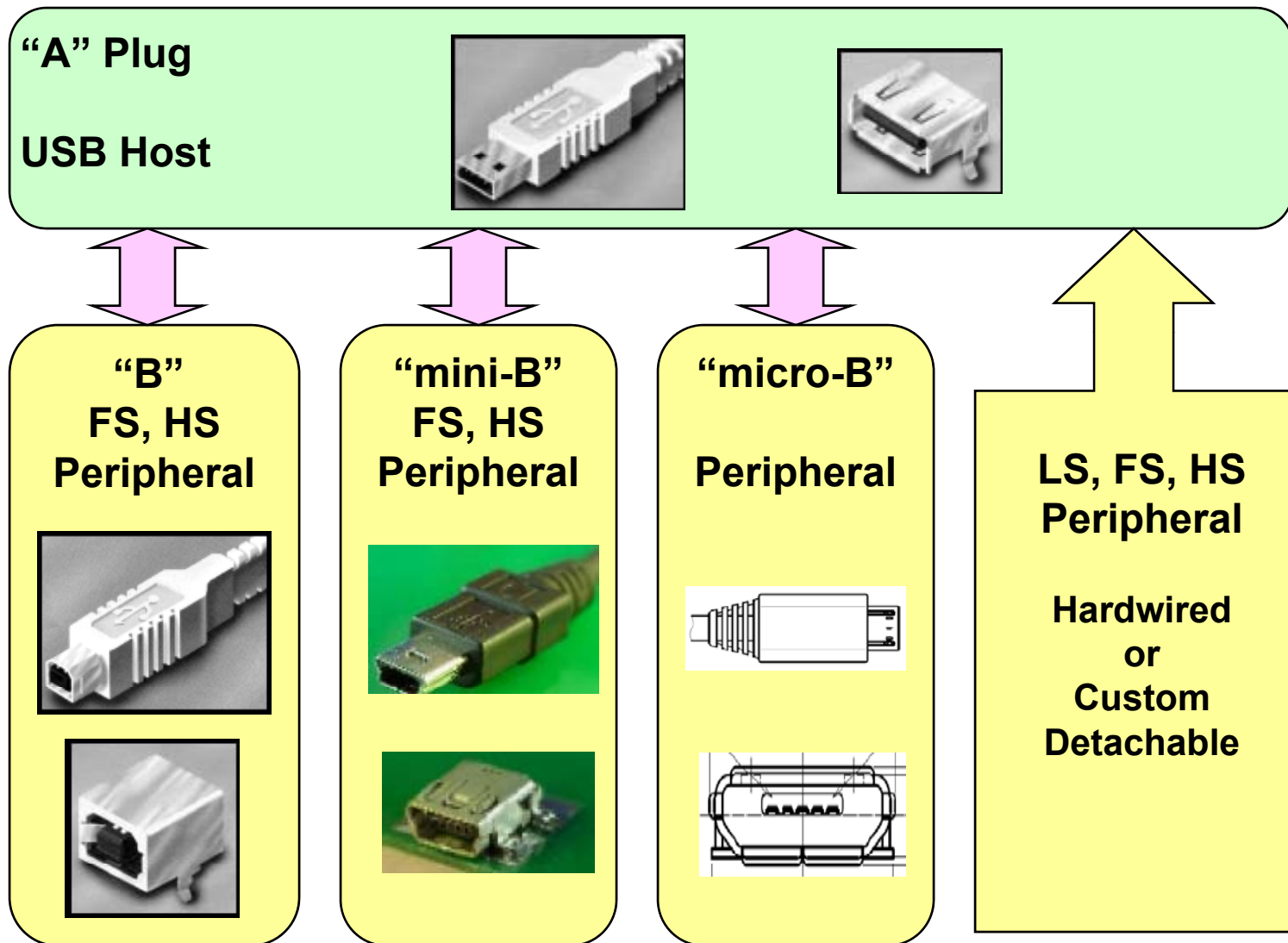
D+ and D- Twisted Pair, Differential mode
Communication Lines

Device Connector

● Interfaz de cuatro cables

- Línea de 5 V de alimentación y de GND desde donde el dispositivo USB puede ser alimentado
- Par trenzado, líneas de comunicaciones en modo diferencial para una excelente inmunidad al ruido de datos
- Conectores especiales para el Host y Dispositivo

Conectores USB



El Mito mas grande

- **Mito:** una Low-Speed periféricos USB pueden transferir datos de aplicaciones hasta 187.5 KB/s (1.5 Mbps)
- **FACT:** Imposible, por las restricciones de la especificación USB :
 - 8 byte de transferencia de datos cada 10 ms
 - = 800 Bytes/segundo

El siguiente gran Mito

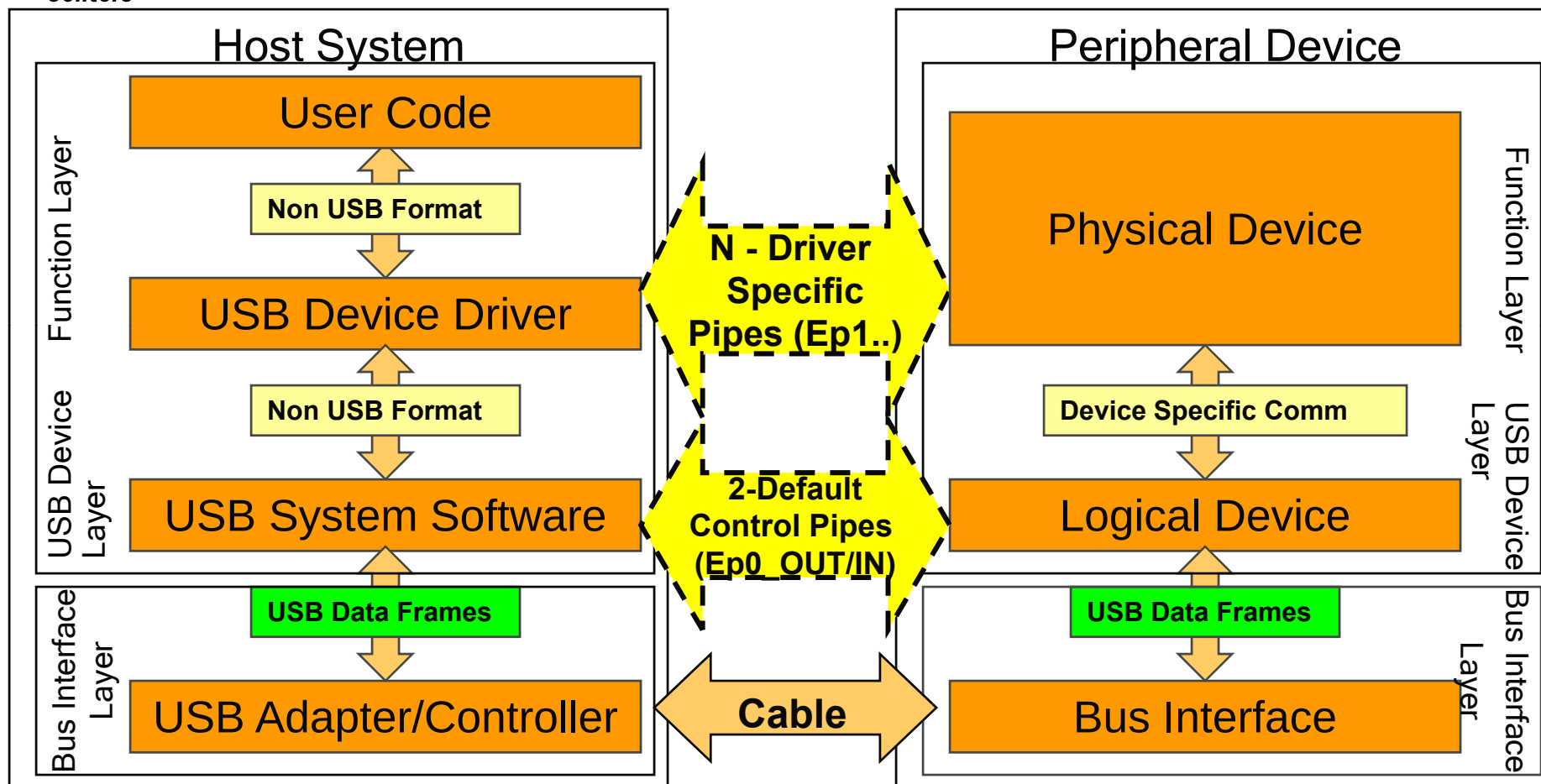
- **Mito:** Una Full-Speed periférico USB puede transferir datos hasta 1.5 MB/s (12 Mb/s)
- **Hecho :** Imposible, el ancho de banda total es de 1.5 MB/s
 - Debe ser compartido entre los periféricos
 - Overhead del protocolo
 - Restricciones del protocolo
 - El procesamiento de datos crudos realista para a periférico simple es ~1.0 MB/s
 - Solo 64KB/s en algunos casos.

Agenda

- El desafío de conectividad para Diseñar Sistemas Embebidos
- Revisión sobre USB
- Principios de Operación
- Soluciones en Microcontroladores PIC® para USB de MCHP
- Usando el Microchip USB Framework
- Drivers de PC para USB
- Microchip USB Herramientas de desarrollo
- Algunas Soluciones USB
- Resumen y Recusos

Dispositivo USB Framework

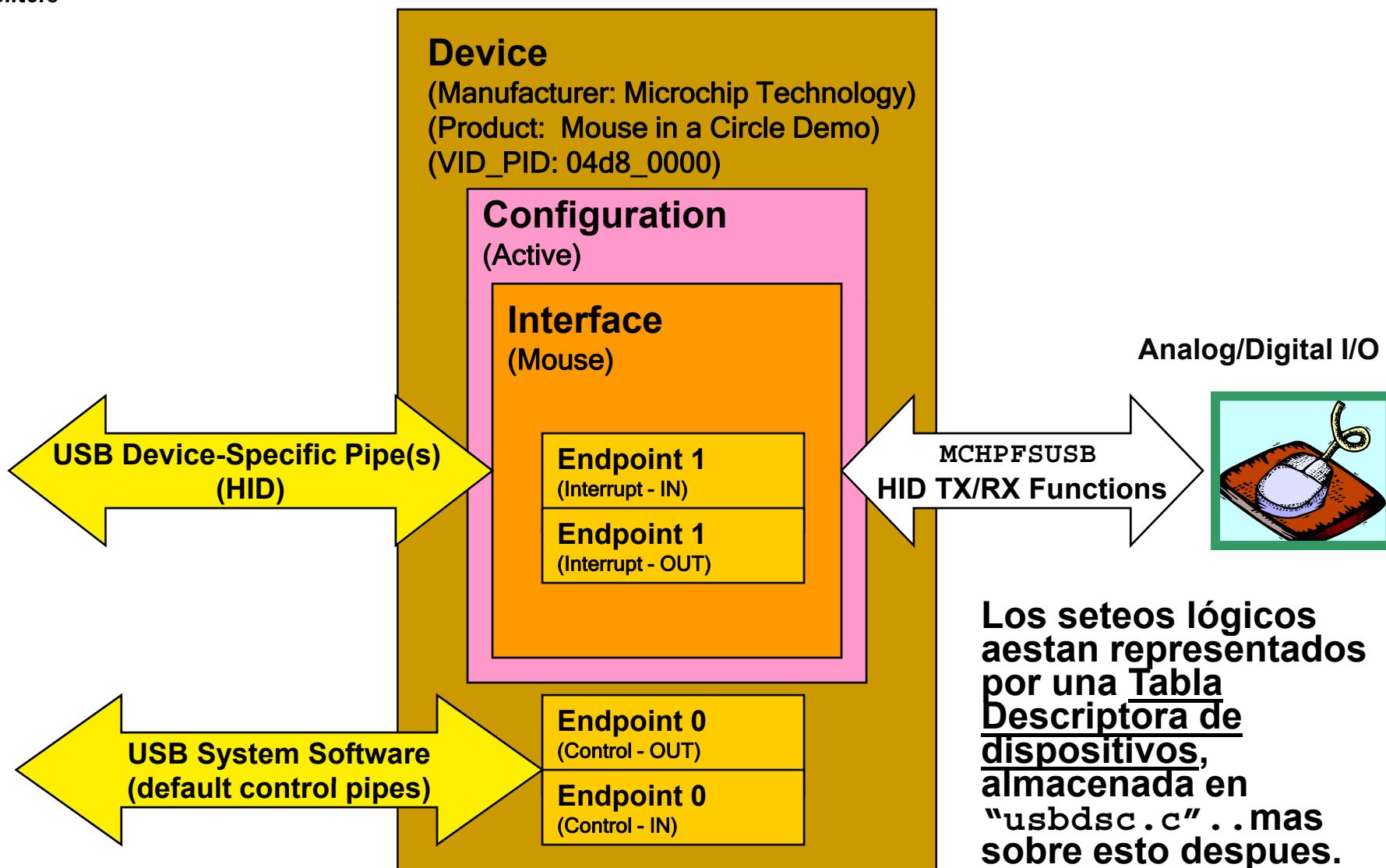
- Software View of Hardware -



Camino de las comunicaciones física

Camino de las comunicaciones lógicas ("Pipe")

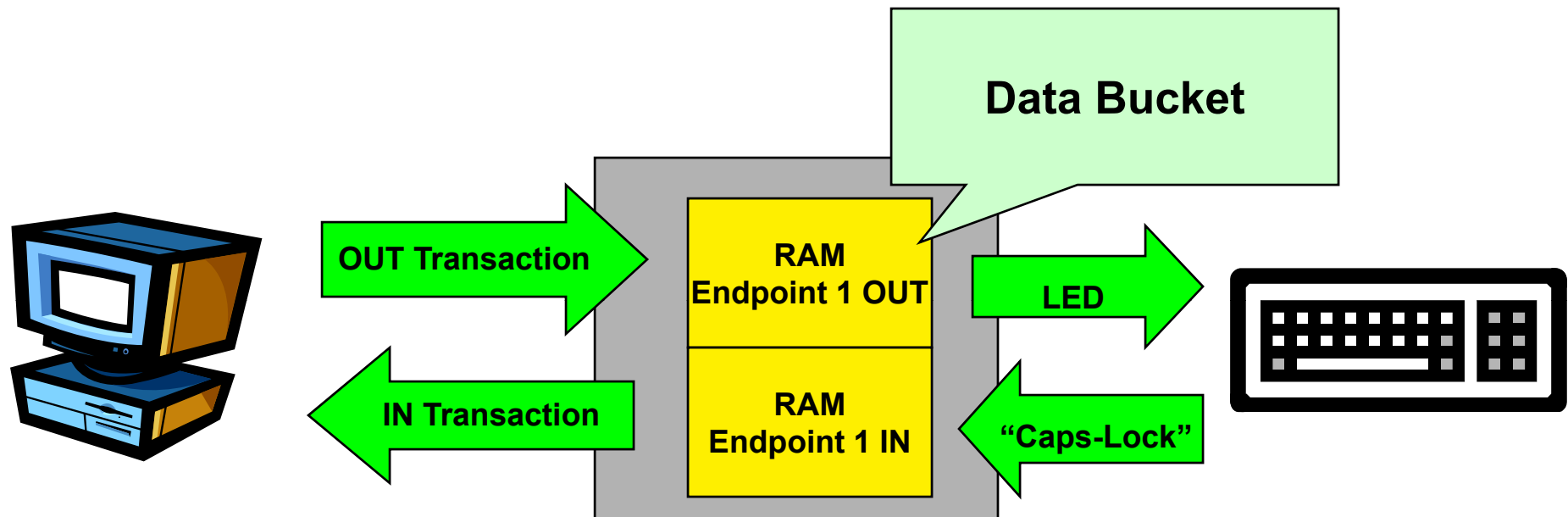
El Dispositivo USB “Lógico”



Como hacen las comunicaciones el host y el dispositivo?

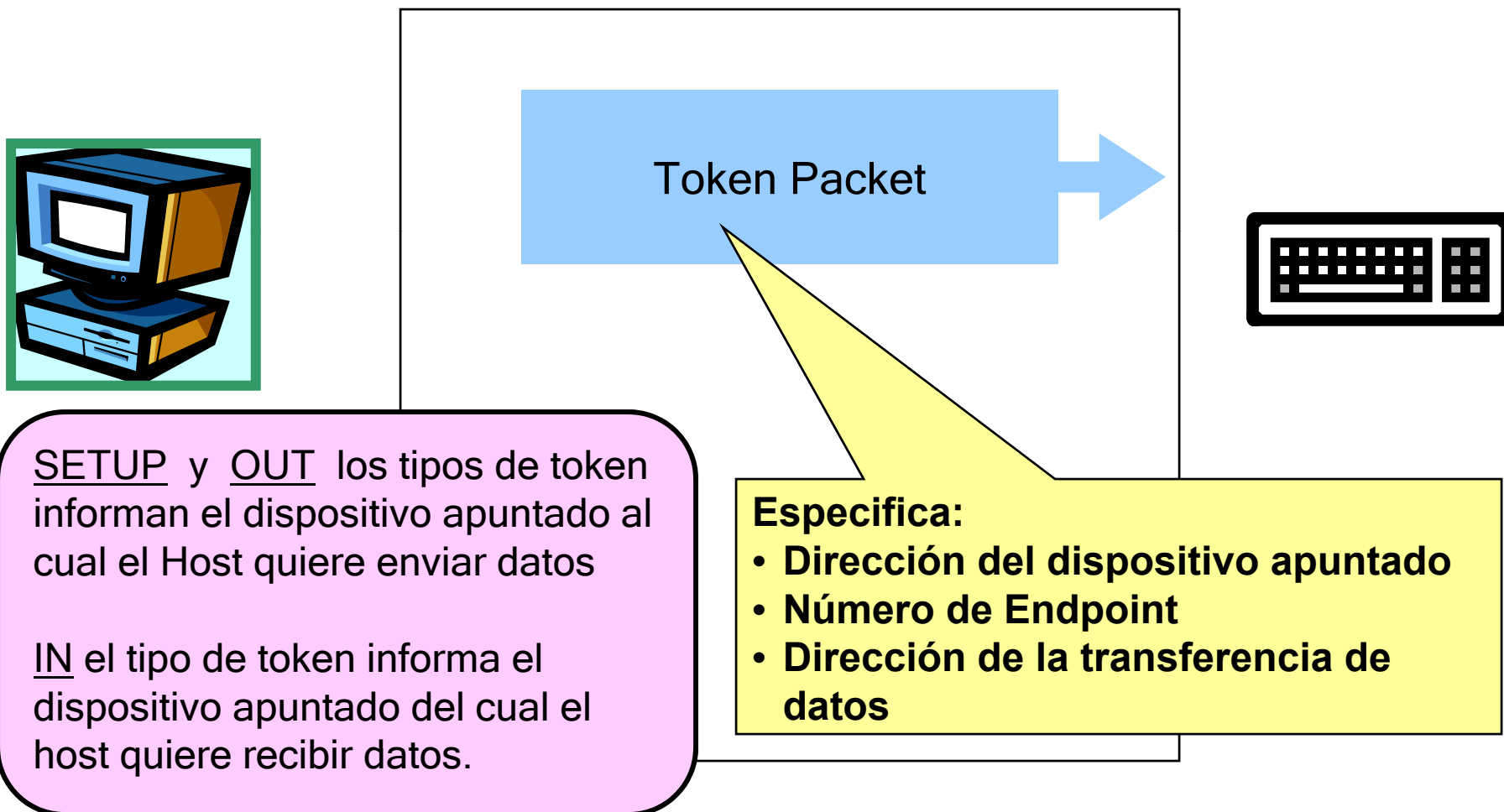
Transacciones...

Un Endpoint es un Buffer

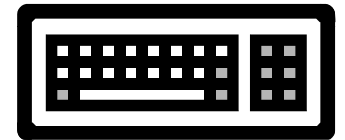
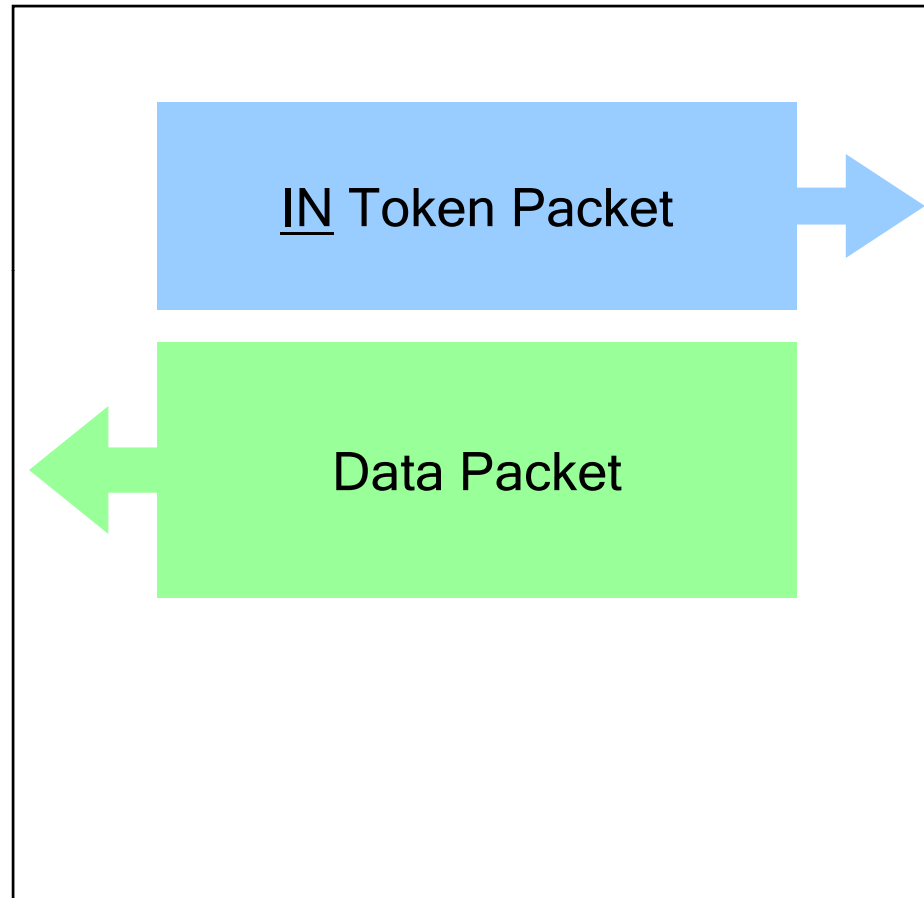


- **El número máximo de endpoints por dispositivo especificado por la especificación USB:**
 - 16 endpoints de salida + 16 endpoints de entrada = 32 endpoints
 - PIC18F87J50, PIC18F4550 suportan hasta 32 endpoints
- **EP0 = Es la comunicación Pipe por Default**

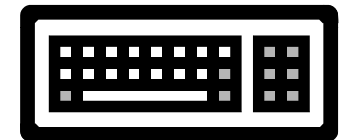
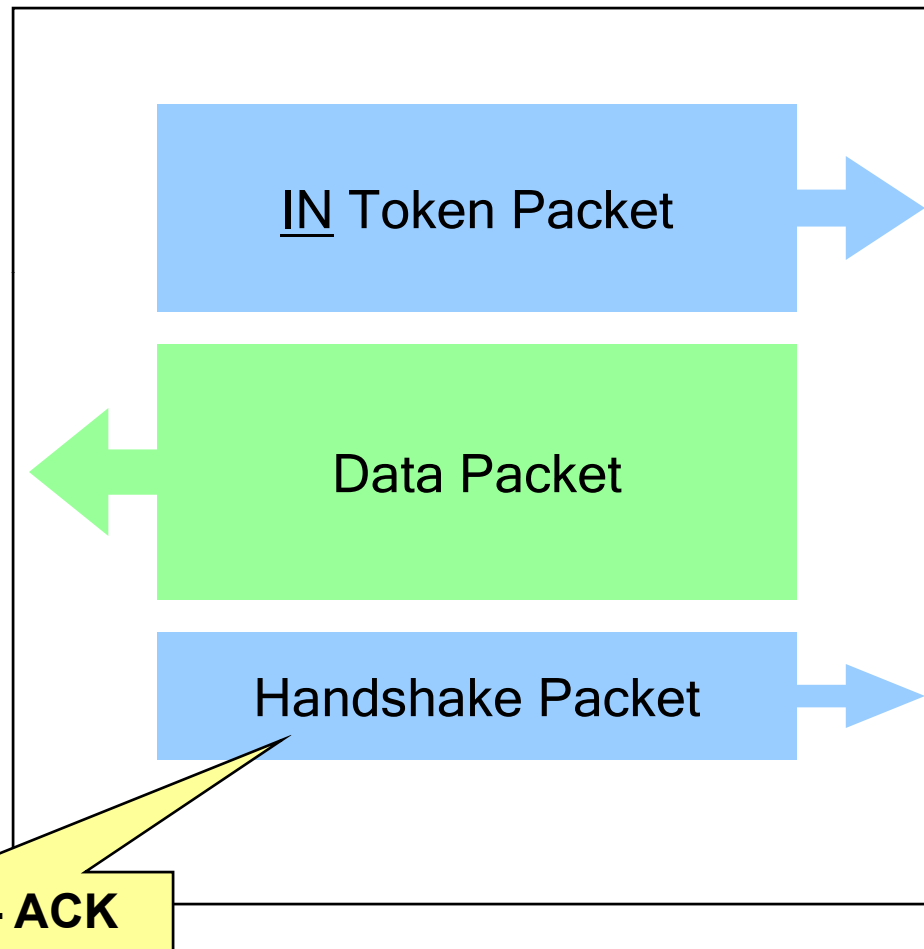
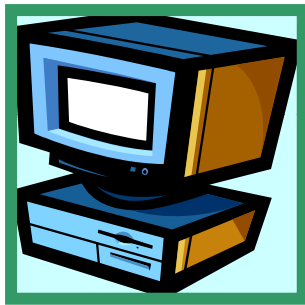
Transacción USB



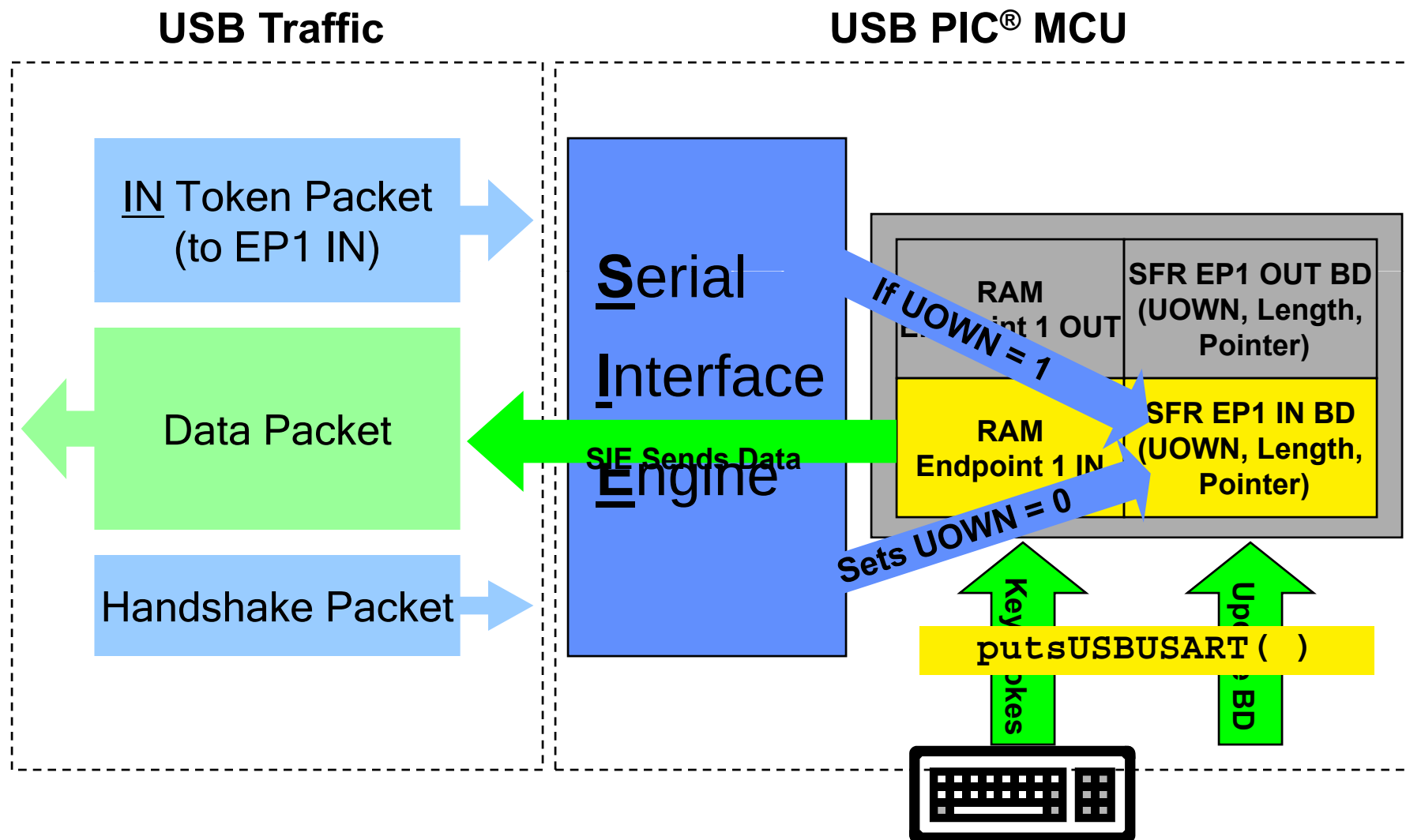
USB Transacción- IN



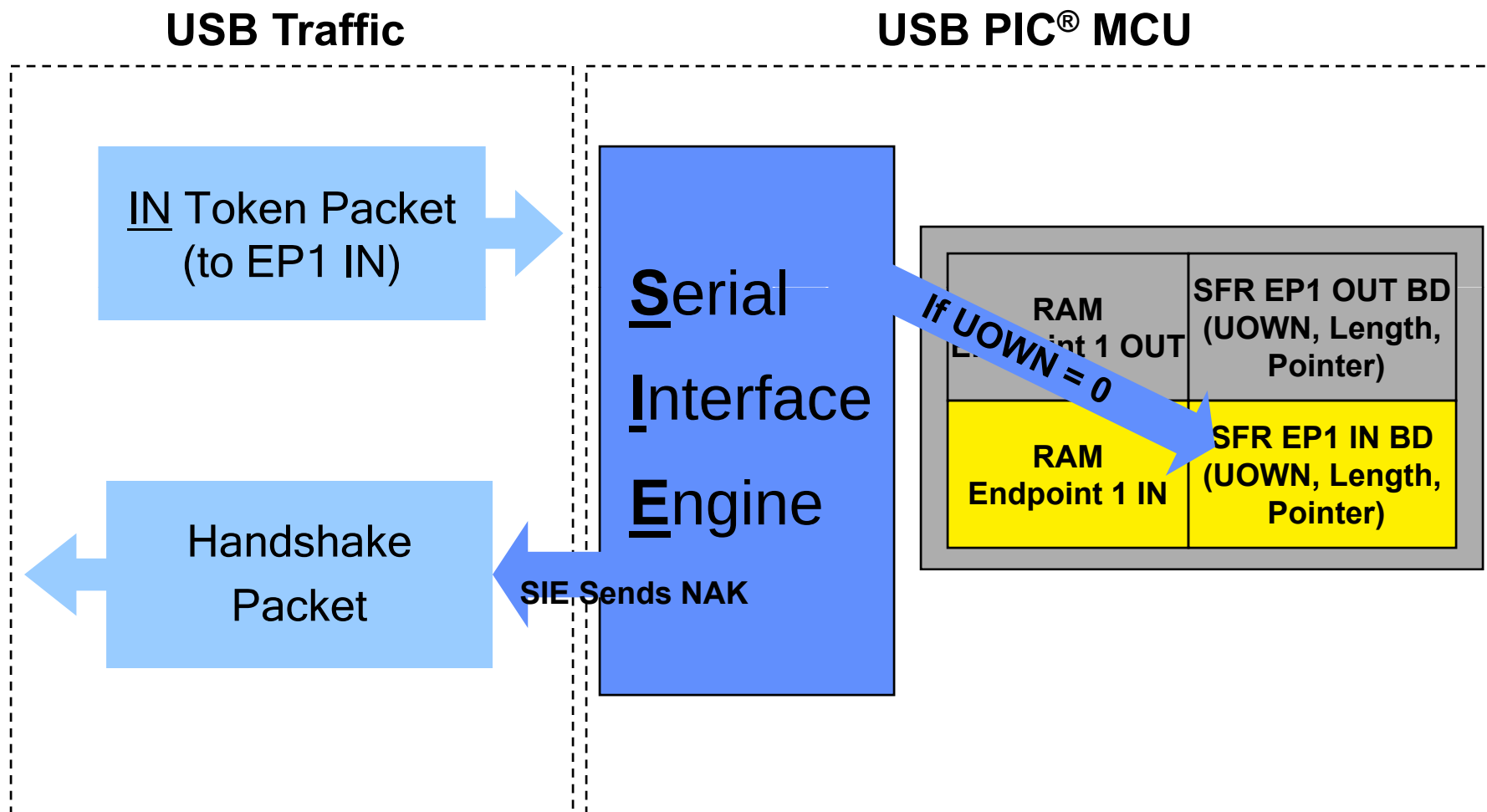
USB Transacción - IN



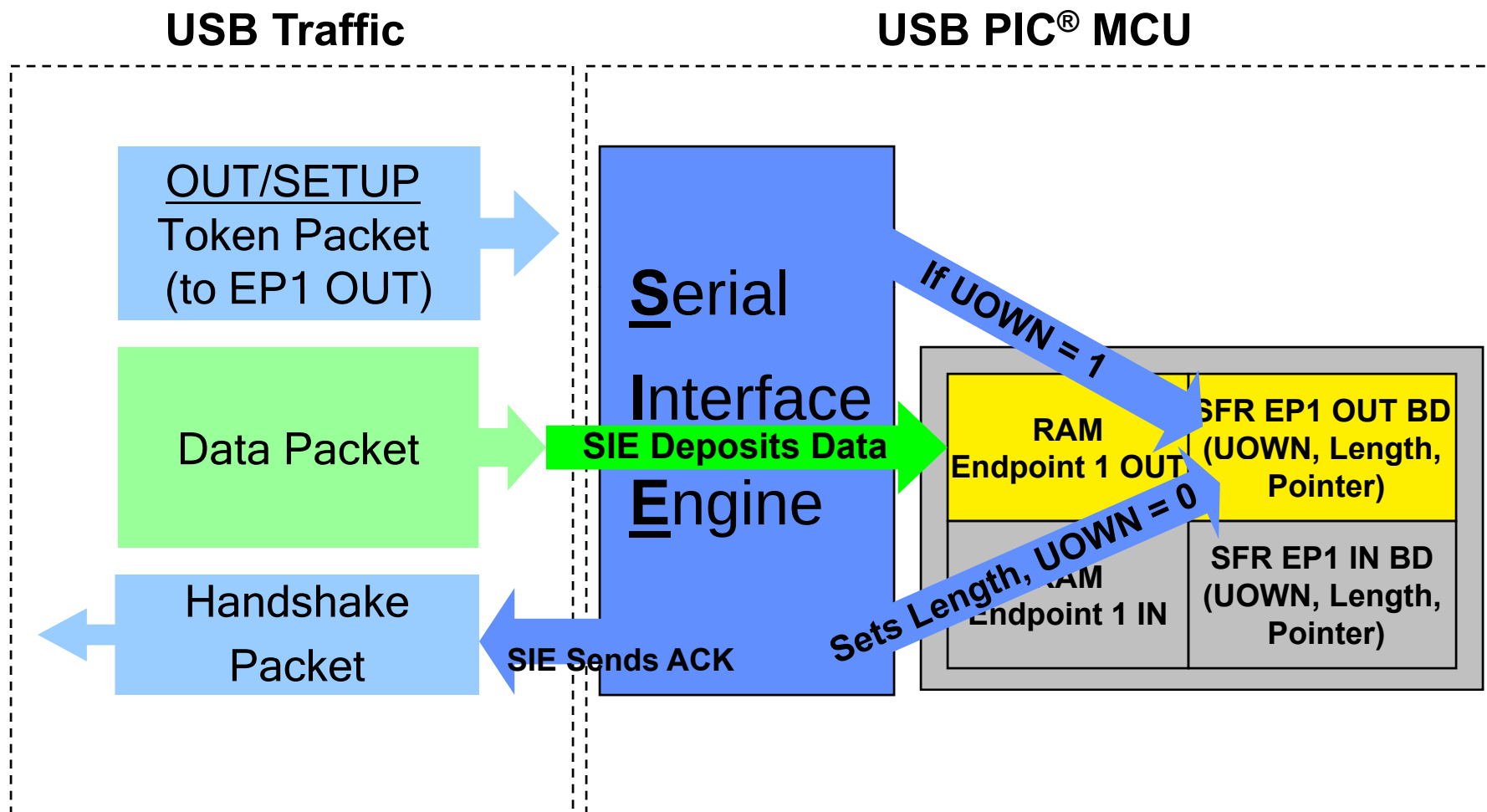
IN Transacción ACK



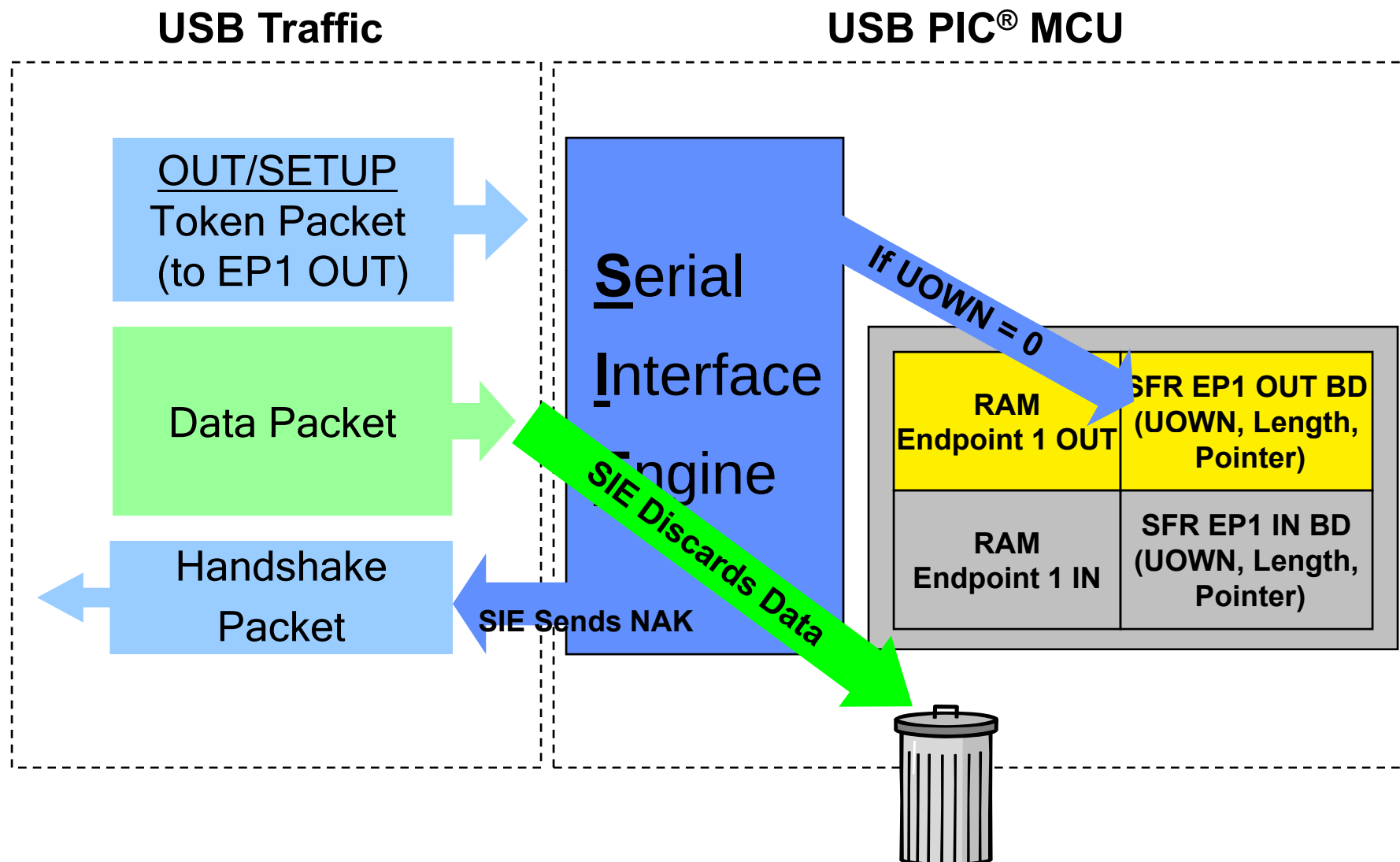
IN Transacción NAK



OUT/SETUP Transacción ACK

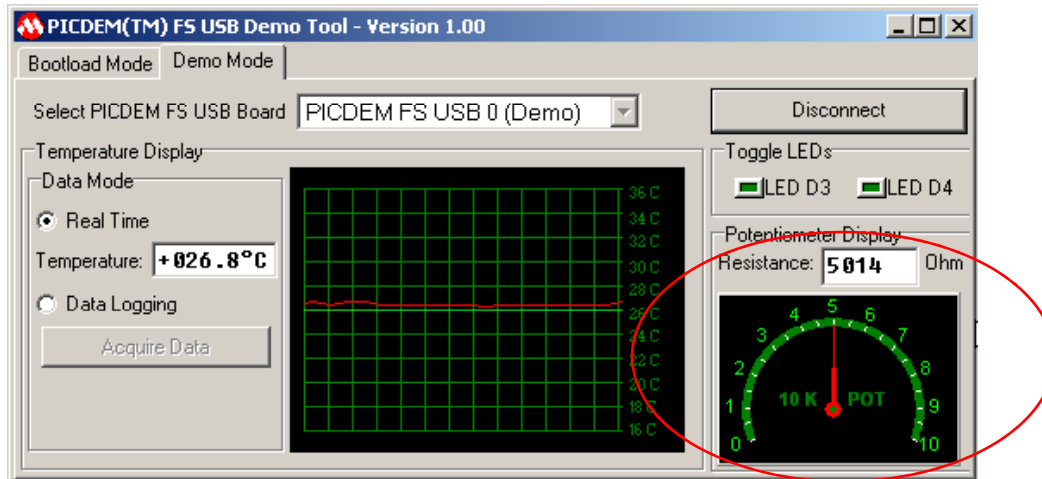


OUT/SETUP Transacción NAK



Transacciones en Acción

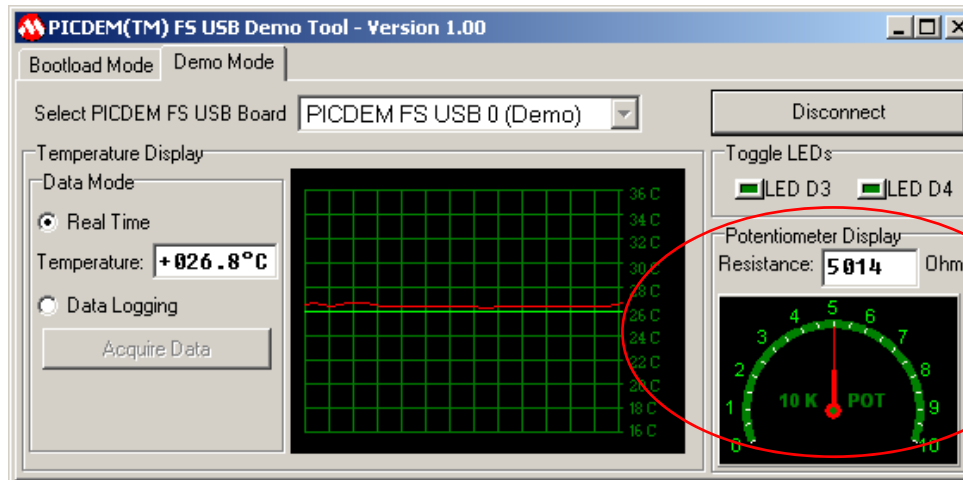
- Función PDFSUSB READ_POT -



- Aplicación PDFSUSB continuamente envía el comando “0x37”
- El dispositivo retorna el comando + el valor del ADC de 10-bit del Potenciómetro: <0x37><ADRESL><ADRESH>

Transacciones en Acción

- Beagle12 Capturando READ_POT -



Referirse al apendice A en Handout para el PDFSUSB completo Referirse al Command/Response

Total Phase - Beagle Data Center - Untitled (modified)

File Beagle Help

Connection 1112-108475 USB Settings 21 ns Capture

Index	m:s.ms.us	Dur	Len	Err	PID	CRC	Packet
0	0:00.000.000						Capture started [Fri Feb 22 05:59:00]
21	0:00.101.276						[110 SOF] [23 PRE/ERR]
31	0:00.125.275						[24 SOF] [6 PRE/ERR]
37	0:00.133.285						[8 SOF] [2 PRE/ERR]
38	0:00.141.181	2.81 us			OUT 00		DEV=0x03 EP=01
39	0:00.141.184	3.47 us	1		DATA1 6901		37
40	0:00.141.188	1.47 us			ACK		
47	0:00.157.274						[24 SOF] [6 PRE/ERR]
48	0:00.165.180	2.83 us			IN 00		DEV=0x03 EP=81
49	0:00.165.183	4.81 us	3		DATA1 30BE		37 00 02
50	0:00.165.189	1.47 us			ACK		
53	0:00.165.284						[8 SOF] [2 PRE/ERR]
63	0:00.189.277						[24 SOF] [6 PRE/ERR]

Llaves: Tipos de Token



SETUP

OUT

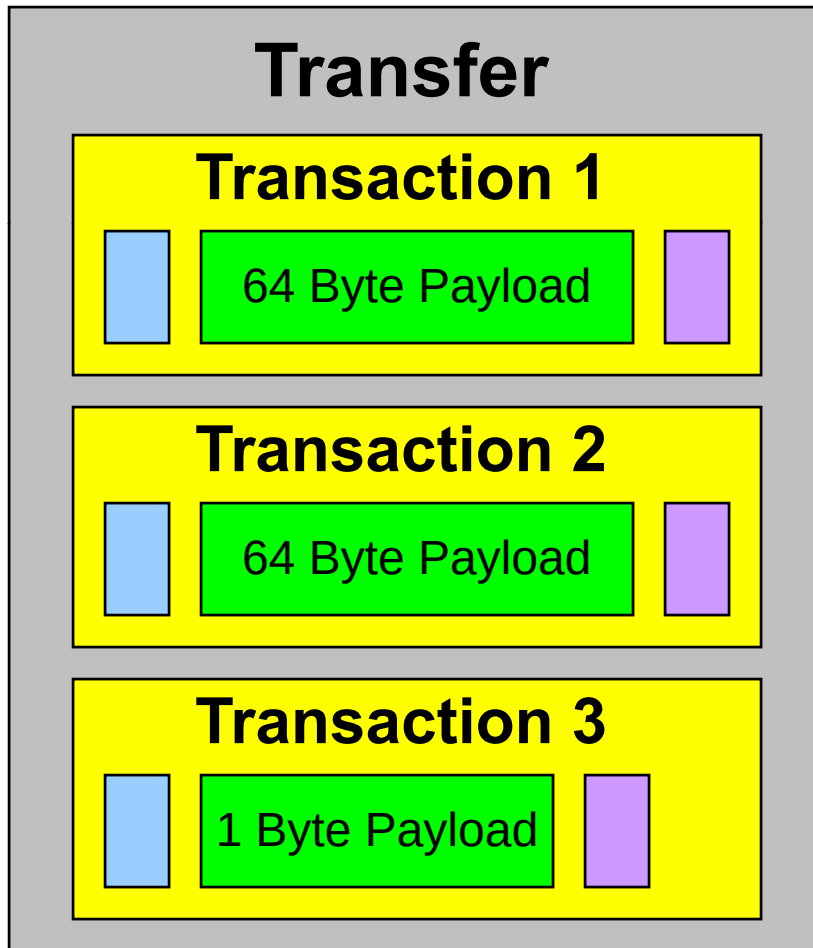
IN

**Como hace la comunicación el host y
el dispositivo?**

Transferencias...

Transferencias Vs. Transacciones

`MPUSBWrite(EP7, Pointer, Size = 129, Timeout)`



Key:



OUT Token Packet



Data Packet



ACK Handshake Packet

Transferencia: Grupo de transacciones relacionadas.

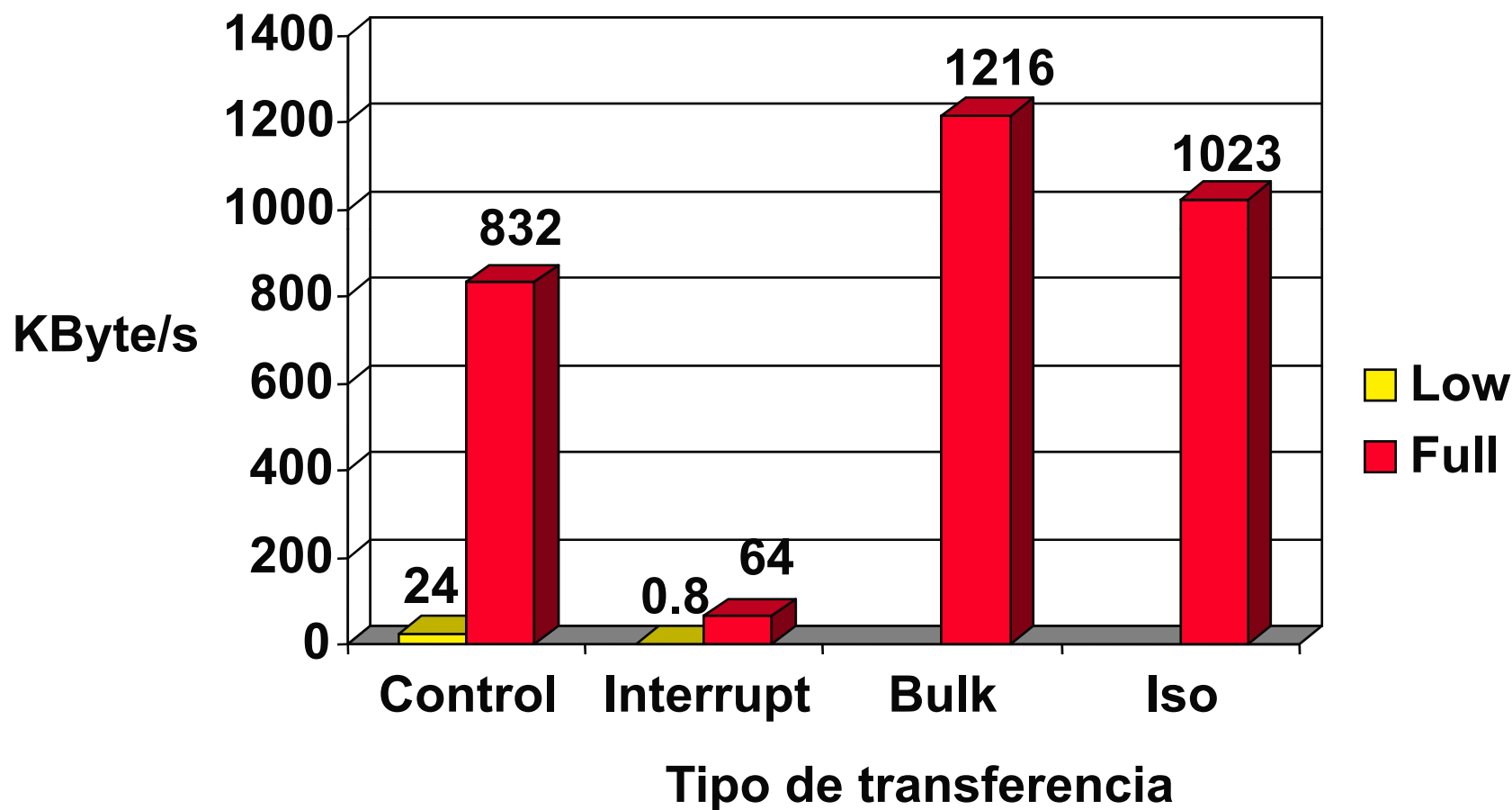
Resumen - Tipo de transferencia de Dato

- Full Speed USB -

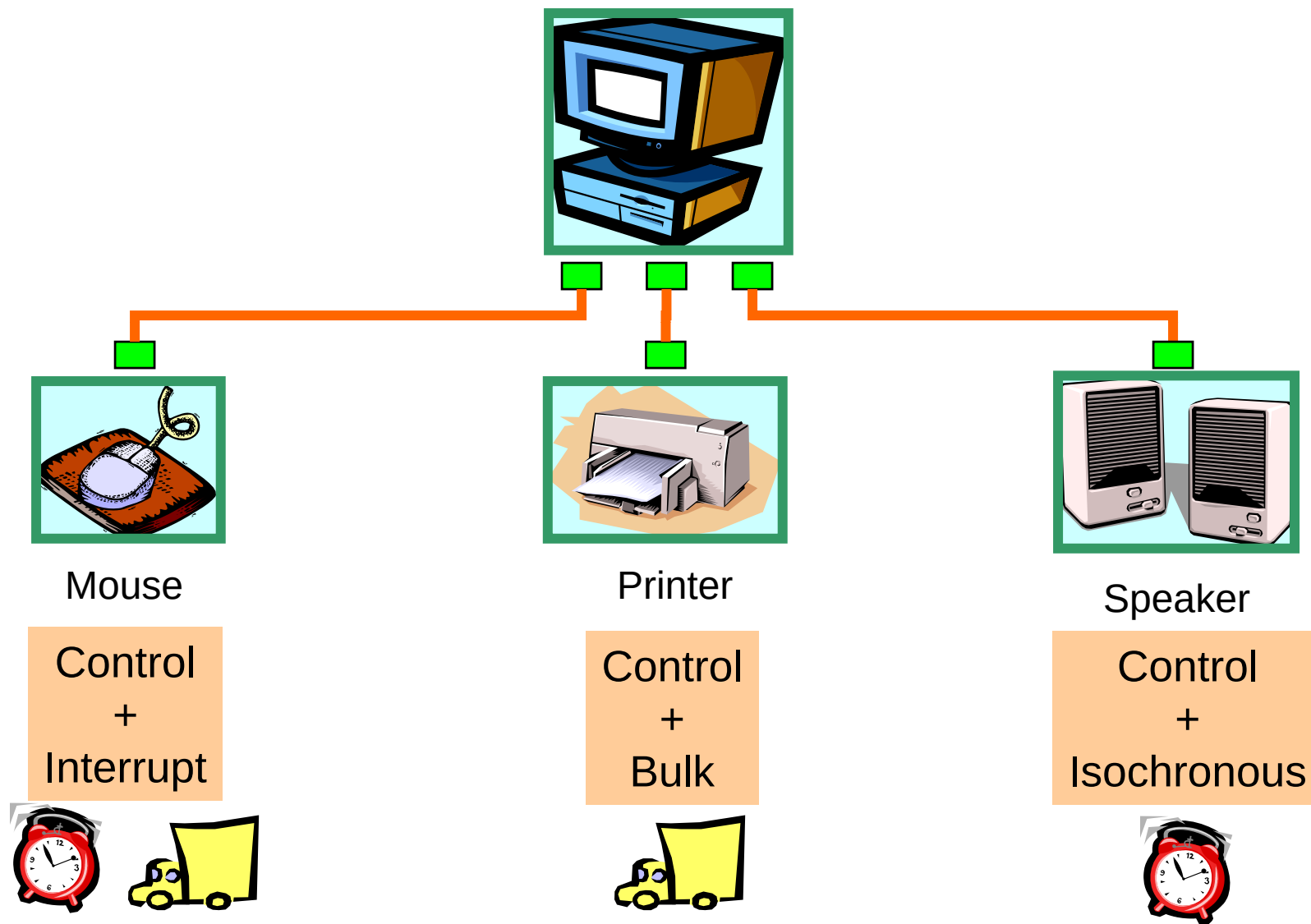
<u>Transfer/ Endpoint Type</u>	<u>Polling Interval</u>	<u>% Reserved BW/Frame for all transfers of this type</u>	<u>Max. # Data Bytes/Frame/Endpoint (Max# transactions per frame @ Max Ep Size)*</u>	<u>Data Integrity</u>
Interrupt	Fixed, Periodic	90	64 (1 x 64)	Yes
Isochronous	Fixed, Periodic	90	1023 (1 x 1023)	No
Bulk	Variable, Uses Free Bandwidth	0	1216 (19 x 64)	Yes
Control	Variable	10	832 (13 x 64)	Yes

*Assumes transfers use maximum packet sizes allowed per Ep type

Rango máximo de transferencias teórica Por Endpoint



Tipos de Transferencias - Ejemplos



Consideraciones cuando se usan las clases standar

- **CDC (Virtual Com Port)**

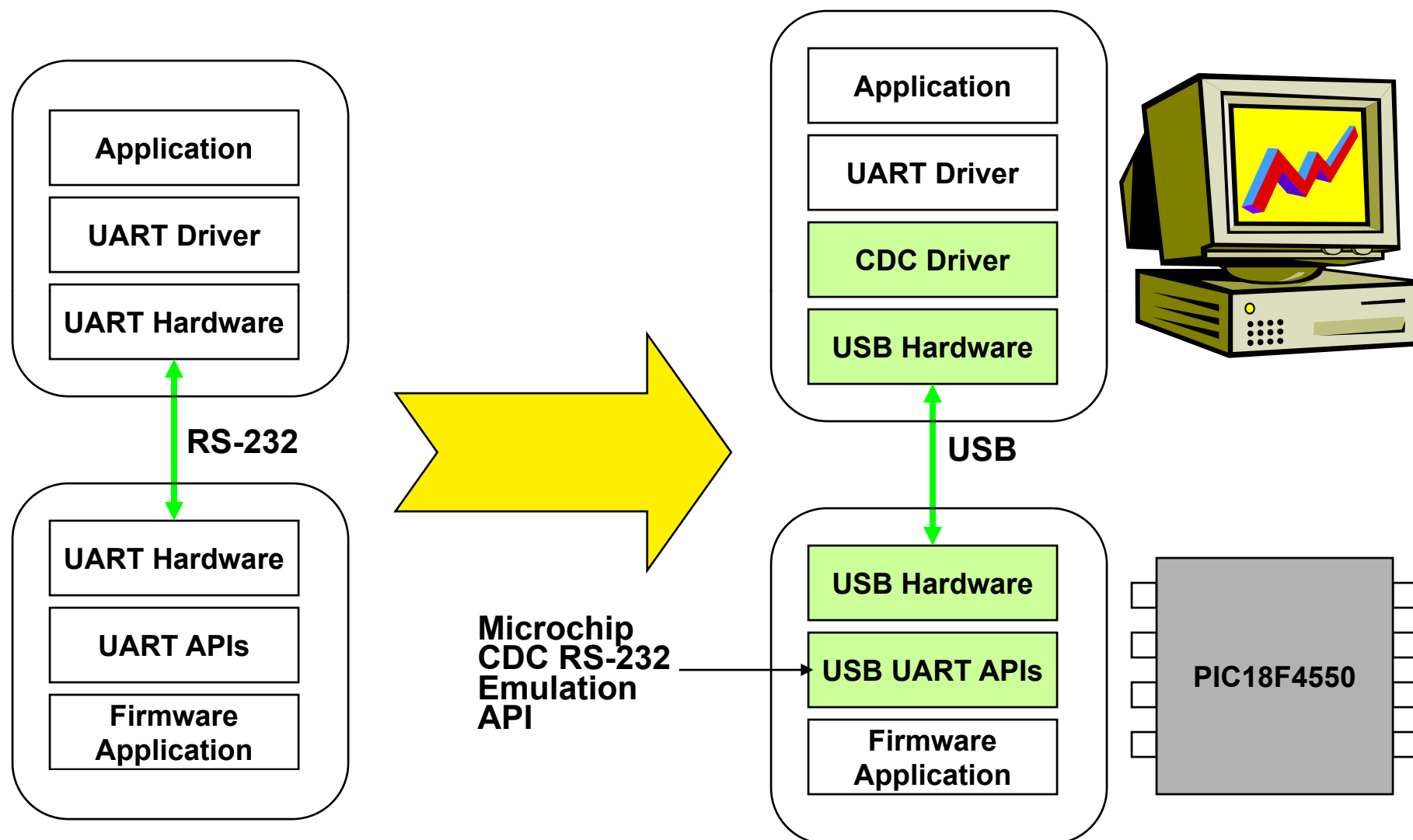
- Protocolo de comunicaciones del dispositivo definido
 - El PIC mira bien un modem, o un terminal conectado al puerto serie
 - El PC Host intercambia con la aplicación PIC string ASCII (default)
 - Llamar al manejador de packets enviando además los paquetes de datos
 - Lower data rate than custom class (80kByte/s vs 1MByte/s)

- **Custom**

- Protocolo de comunicaciones del dispositivo indefinido
 - PC application talks directly to the endpoint in the PIC
 - The PC application/PIC exchange raw bytes

**Mas sobre el USB
CDC Device Class
en la Parte 2**

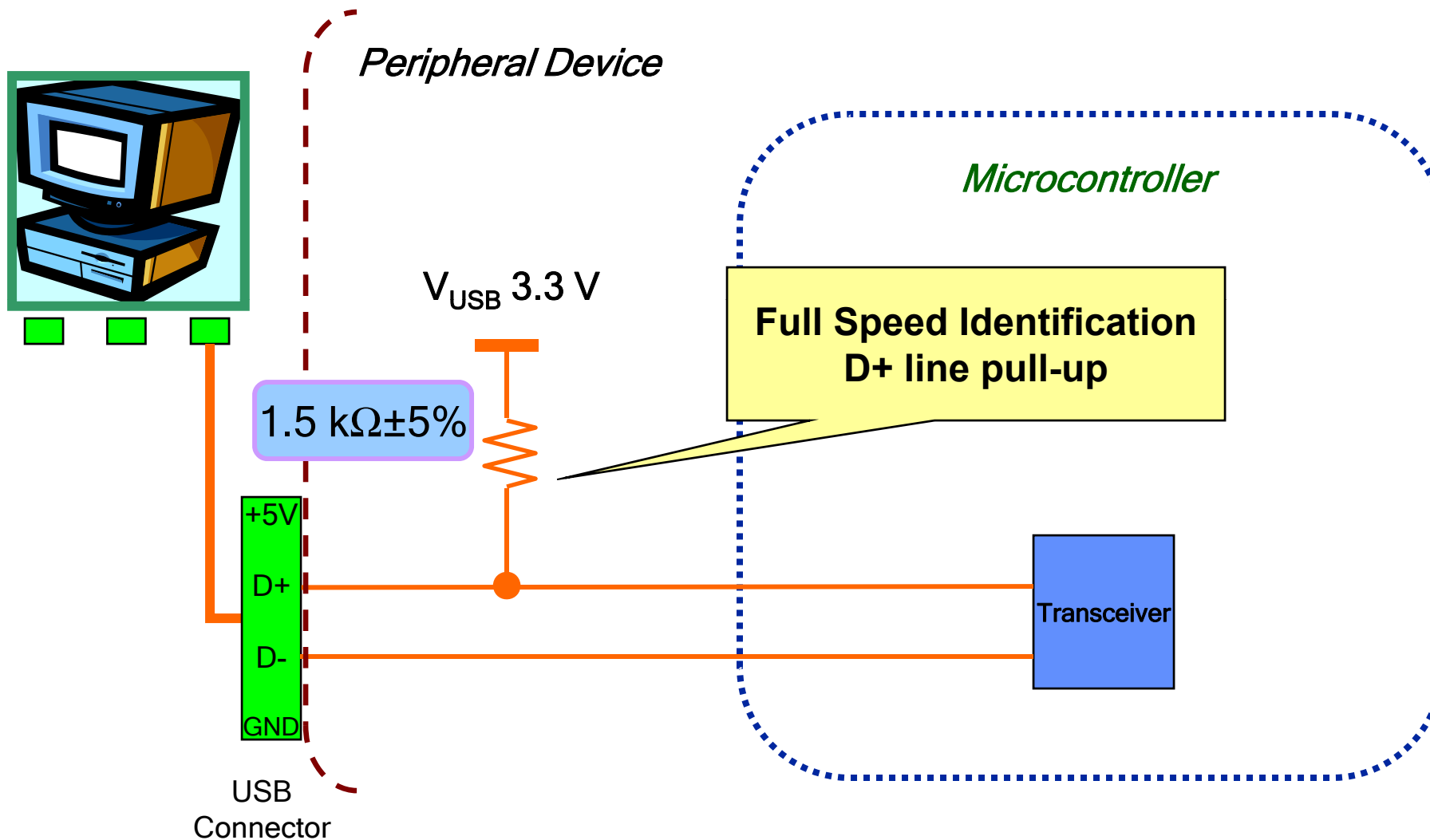
Ejemplo – Usando CDC



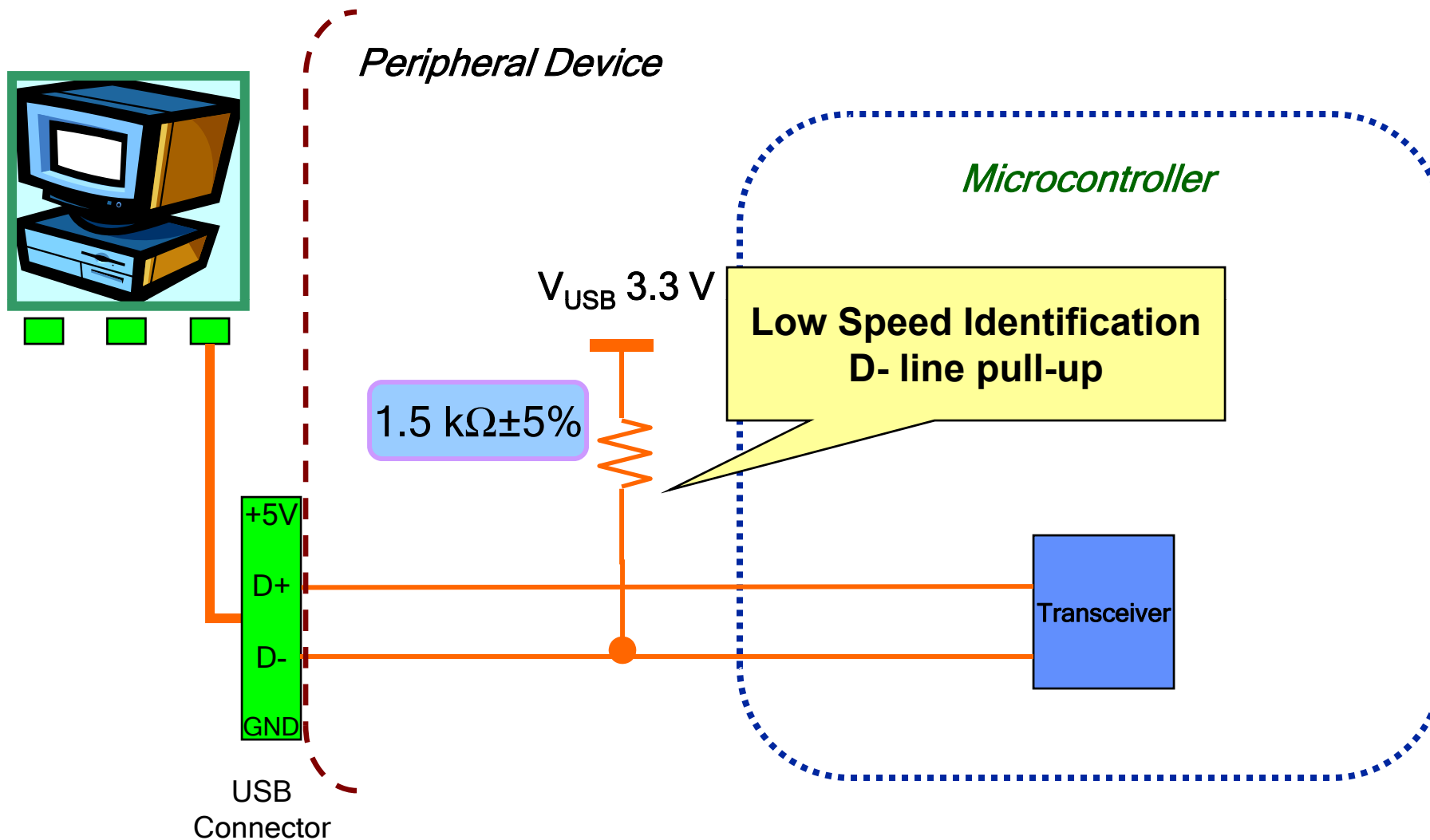
Enumeración Auto-Detección & Auto-Configuración

***Detras de la “Magia” del
“Plug&Play”***

Auto-Detección: Full-Speed

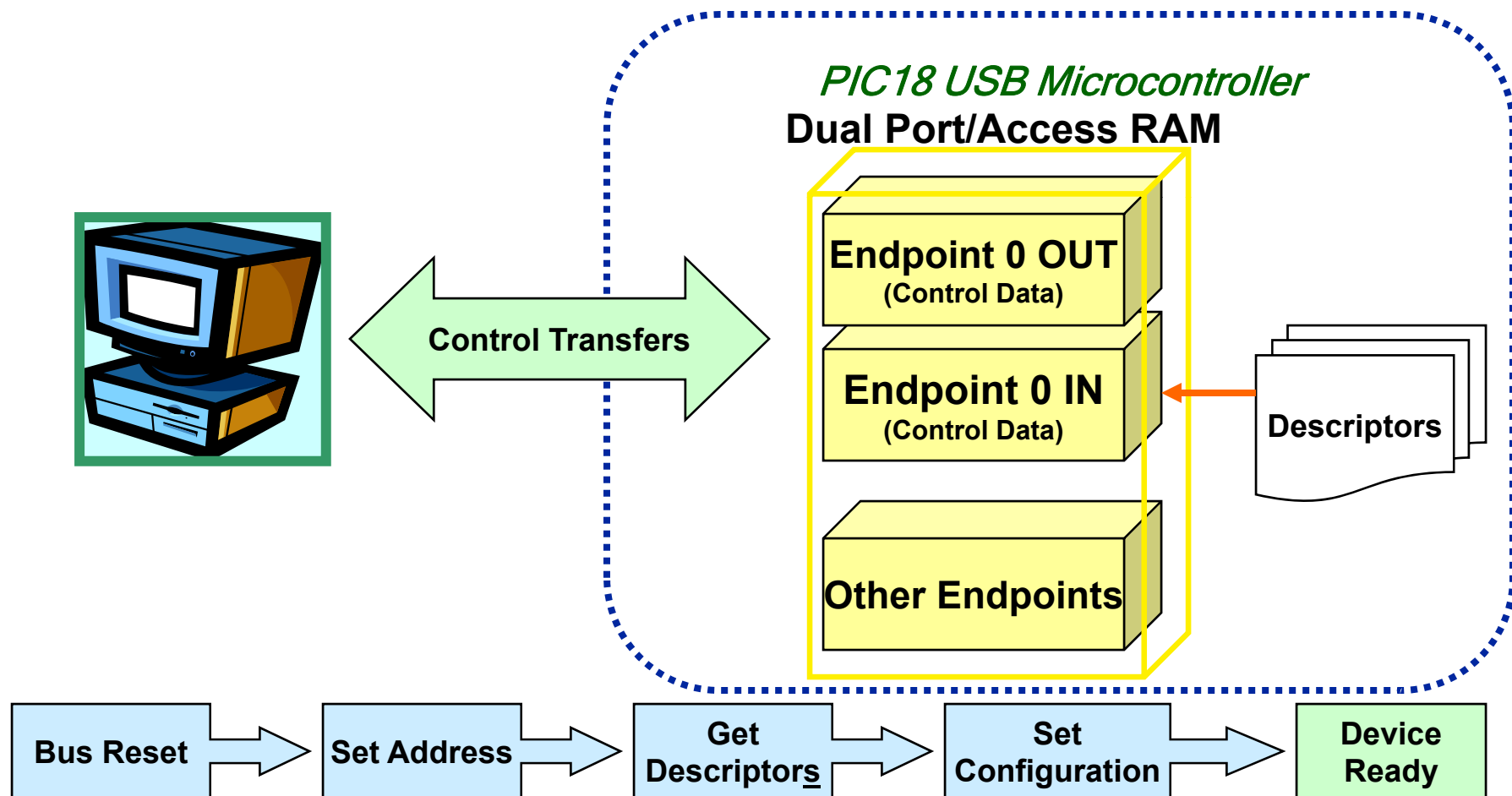


Auto-Detección: Low-Speed

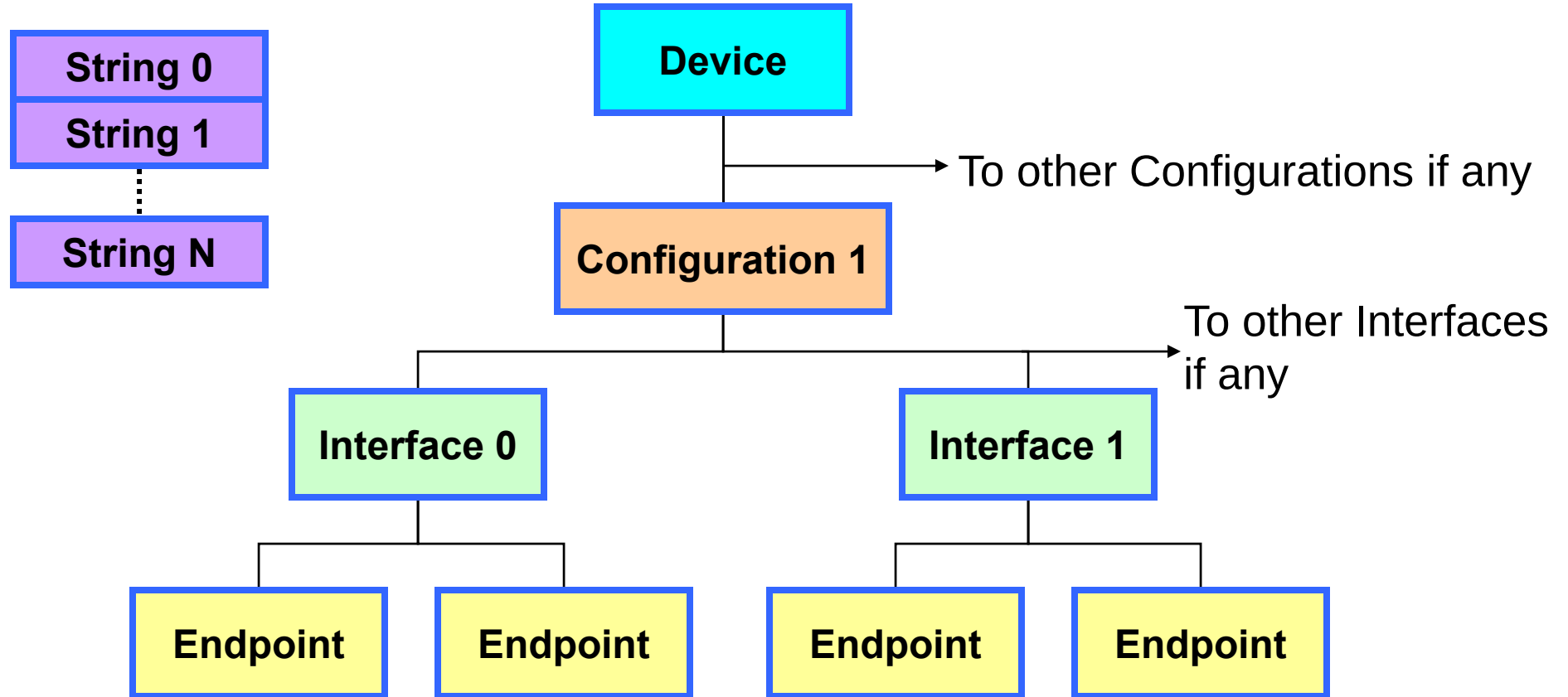


Endpoint 0 y Enumeración

- See Chapter 9 in USB 2.0 Spec for more info.

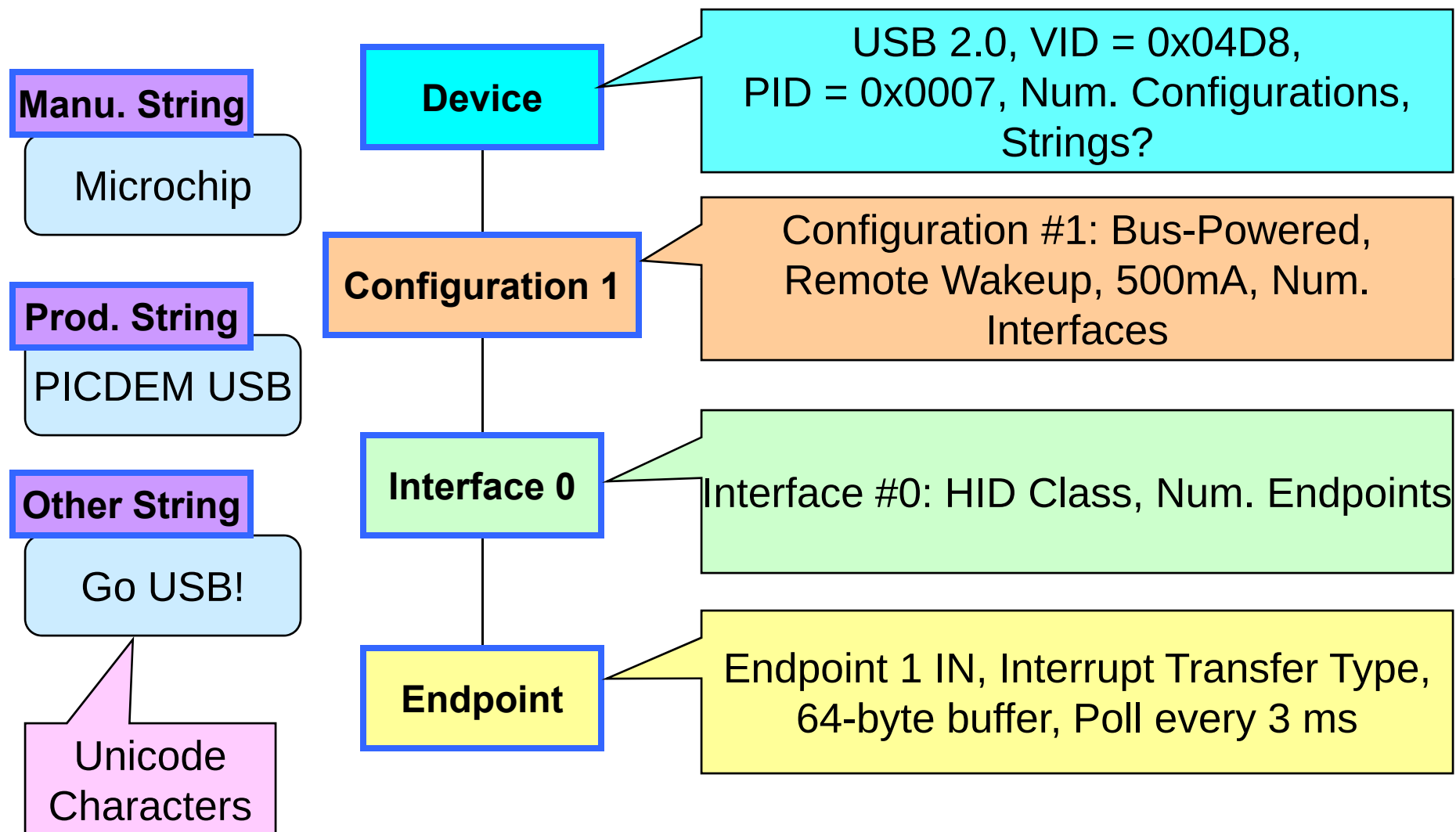


Descriptores



- **Los Descriptores son tipicamente almacenados en memoria no volatil/Flash**

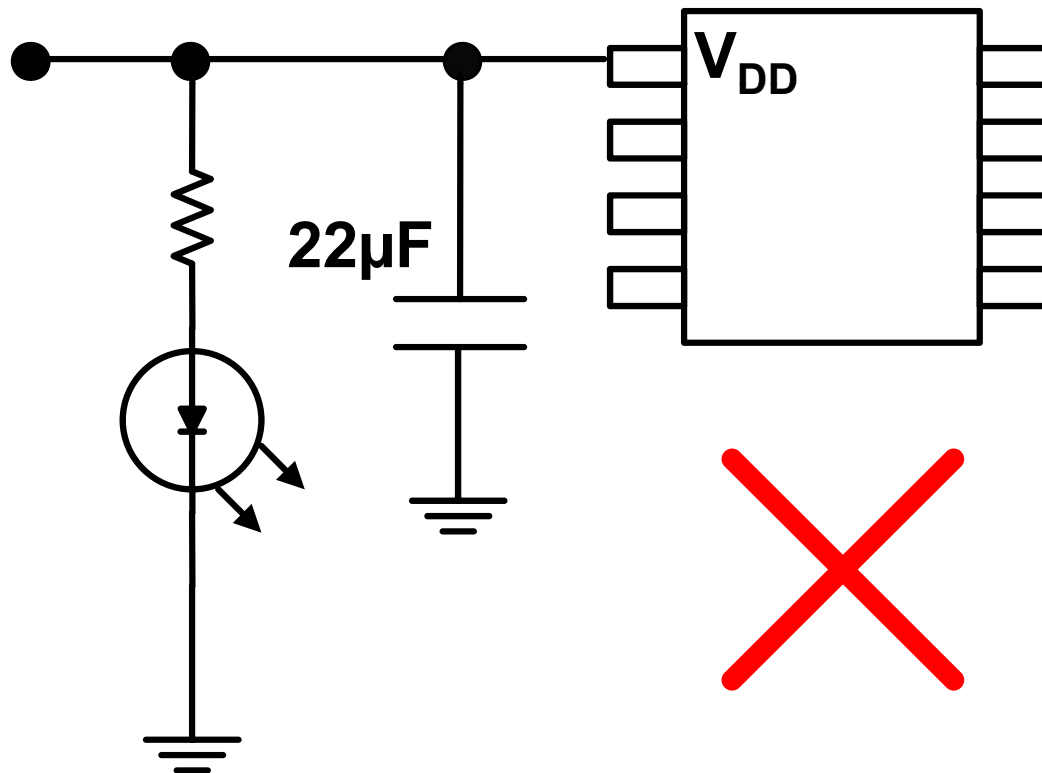
Descriptores - Ejemplo



Planeando la alimentación

- Max USB suspend current is 0.5/2.5mA
- Don't:

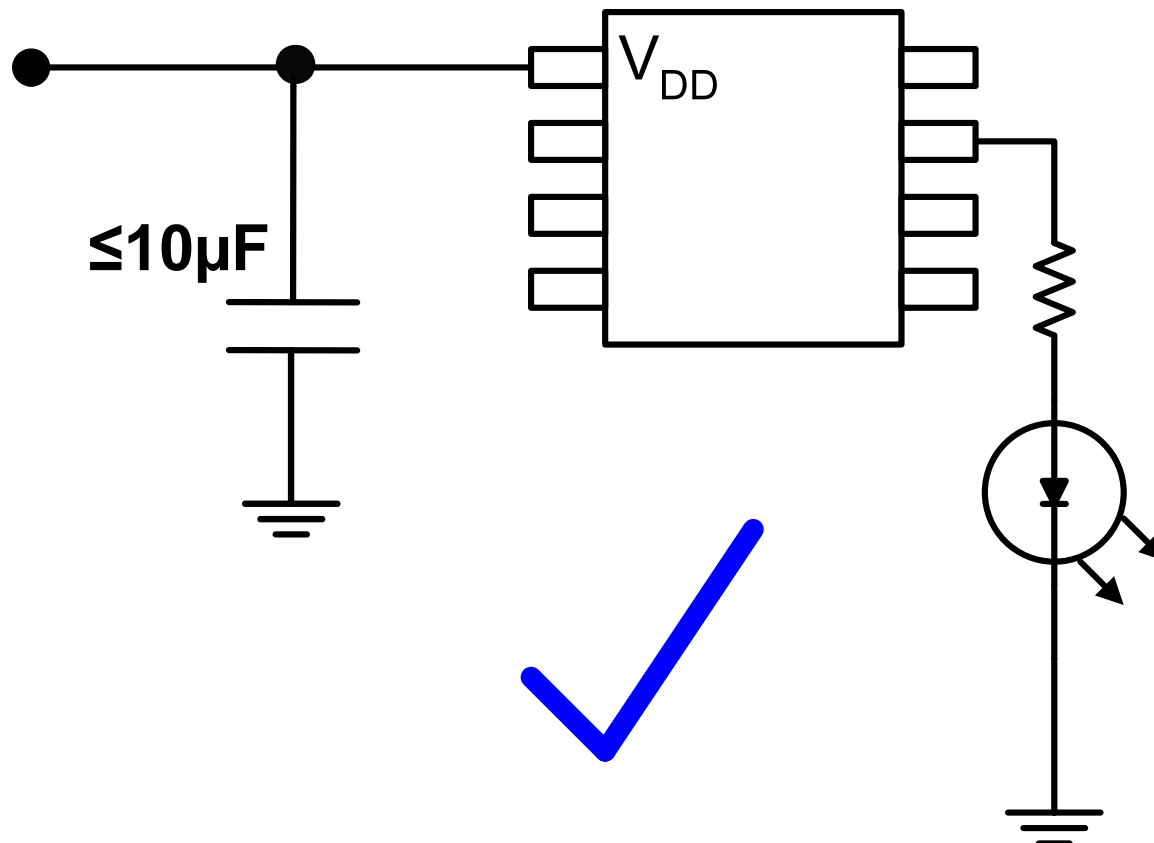
Power from
USB Cable



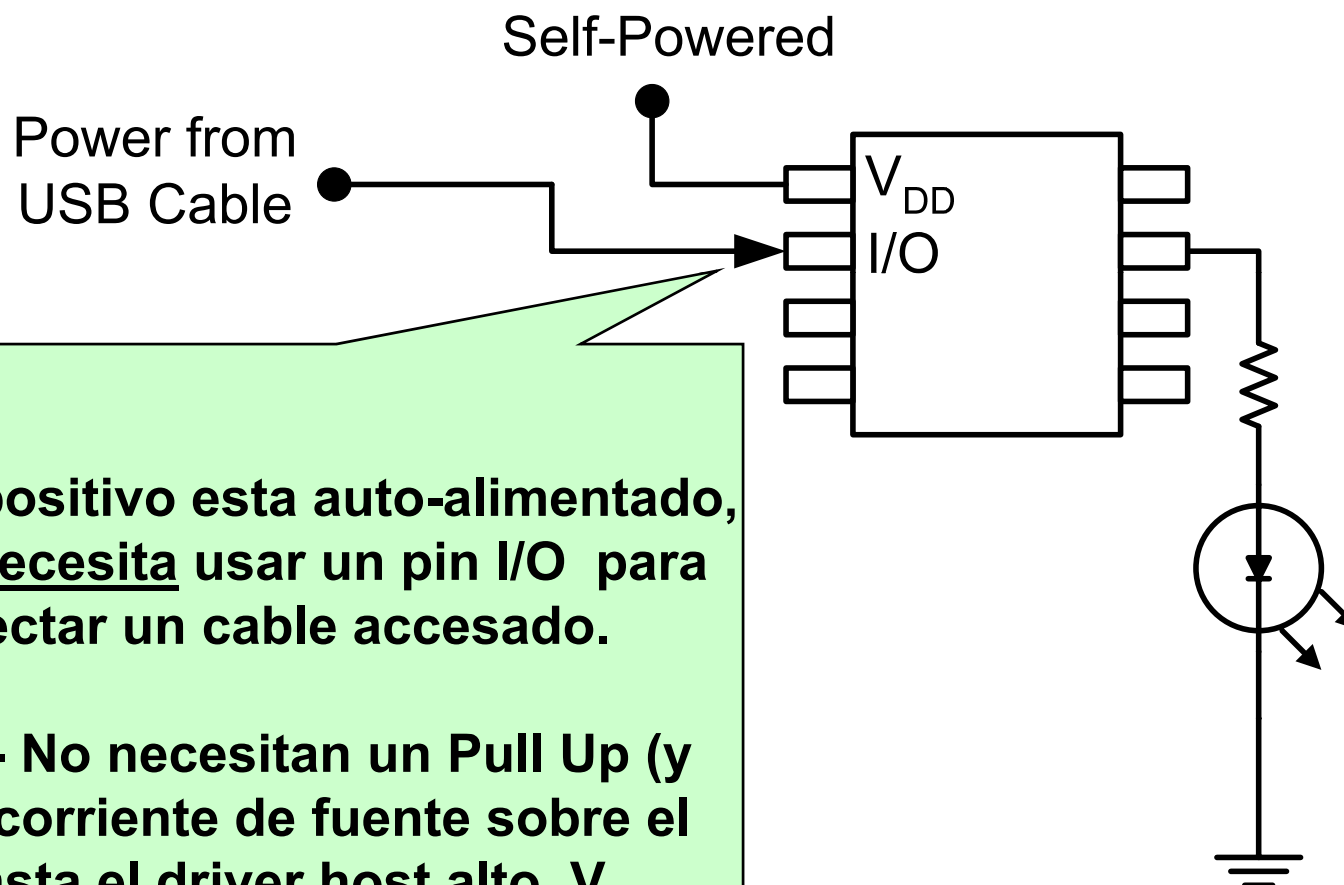
Planeando la alimentación

- Max USB suspend current is 0.5/2.5mA
- Do:

Power from
USB Cable



PIC® MCU: Detectando un accesorio USB



Si el dispositivo esta auto-alimentado, usted Necesita usar un pin I/O para detectar un cable accesado.

D+ o D- No necesitan un Pull Up (y nunca corriente de fuente sobre el V_{BUS}) hasta el driver host alto V_{BUS} .

Chequeando el conocimiento del USB

- **Número máximo de dispositivos puede soportar el USB ?**
- **Número de Pines en un conector USB?**
- **Que es FS USB data rate?**
- **Que direcció es Data-In?**
- **Tipo de transferencia de datos USB ?**
- **Que es enumeración USB ?**

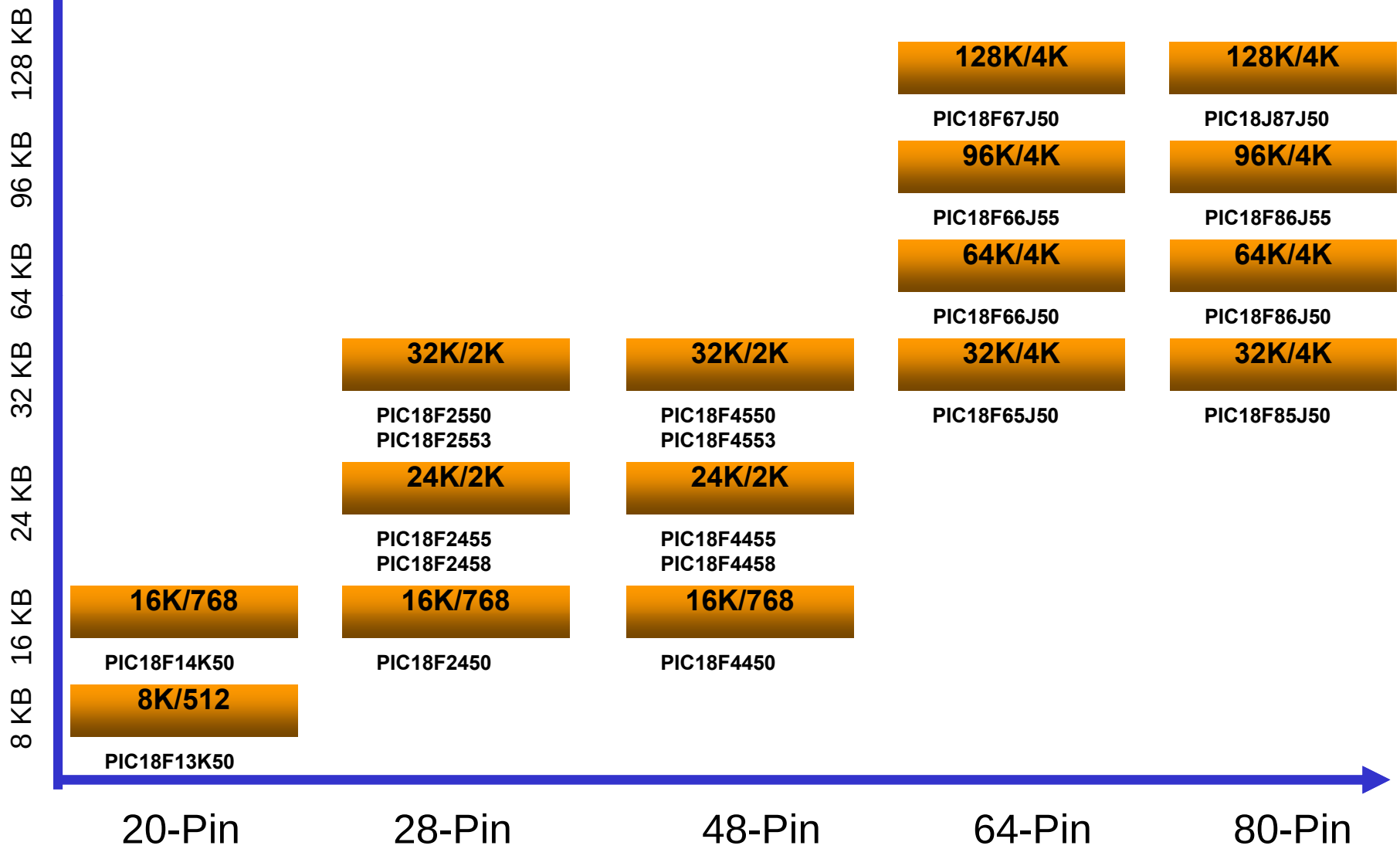
Agenda

- El desafío de conectividad para Diseñar Sistemas Embebidos
- Revisión sobre USB
- Principios de Operación
- Soluciones en Microcontroladores PIC® para USB de MCHP
- Usando el Microchip USB Framework
- Drivers de PC para USB
- Microchip USB Herramientas de desarrollo
- Algunas Soluciones USB
- Resumen y Recusos

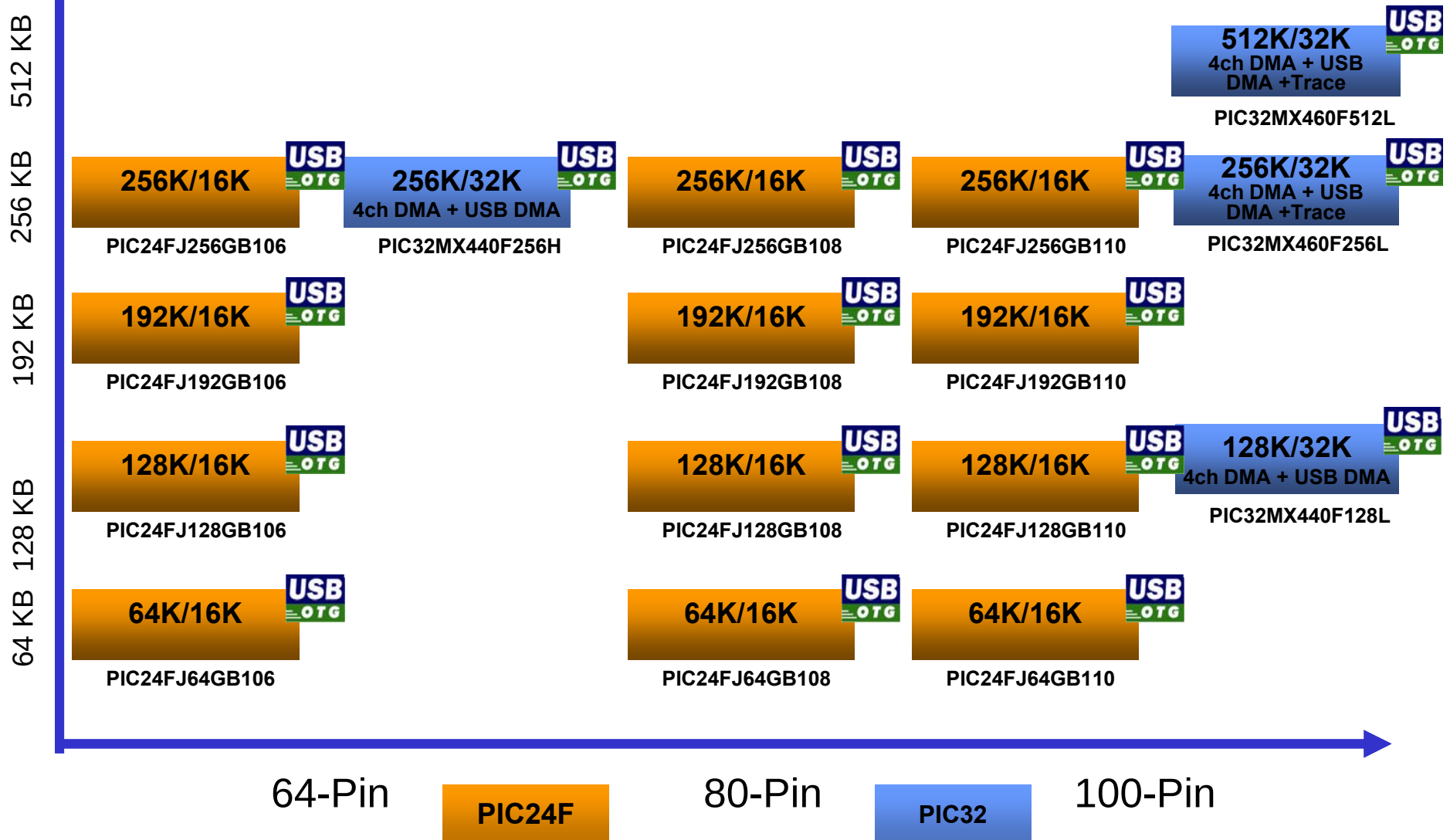
MCUs PICs de Microchip

- Ofrece MCU PIC con arquitectura de 8, 16, and 32 bit
- Dispositivos de 20 - 100 Pines
- 8Kb a 256Kb de Memoria de Programa en Flash
- 512 bytes a 16Kb Ram de Datos
- Voltaje de Operación:1.8V to 5.5V
- Amplio Rango de Periféricos para Aplicaciones en el mundo Real
 - Conversores A/D , Comparadores
 - SPI, I2C, UART
 - PWMs, Timers y lineas I/O

Productos USB Ofrecidos

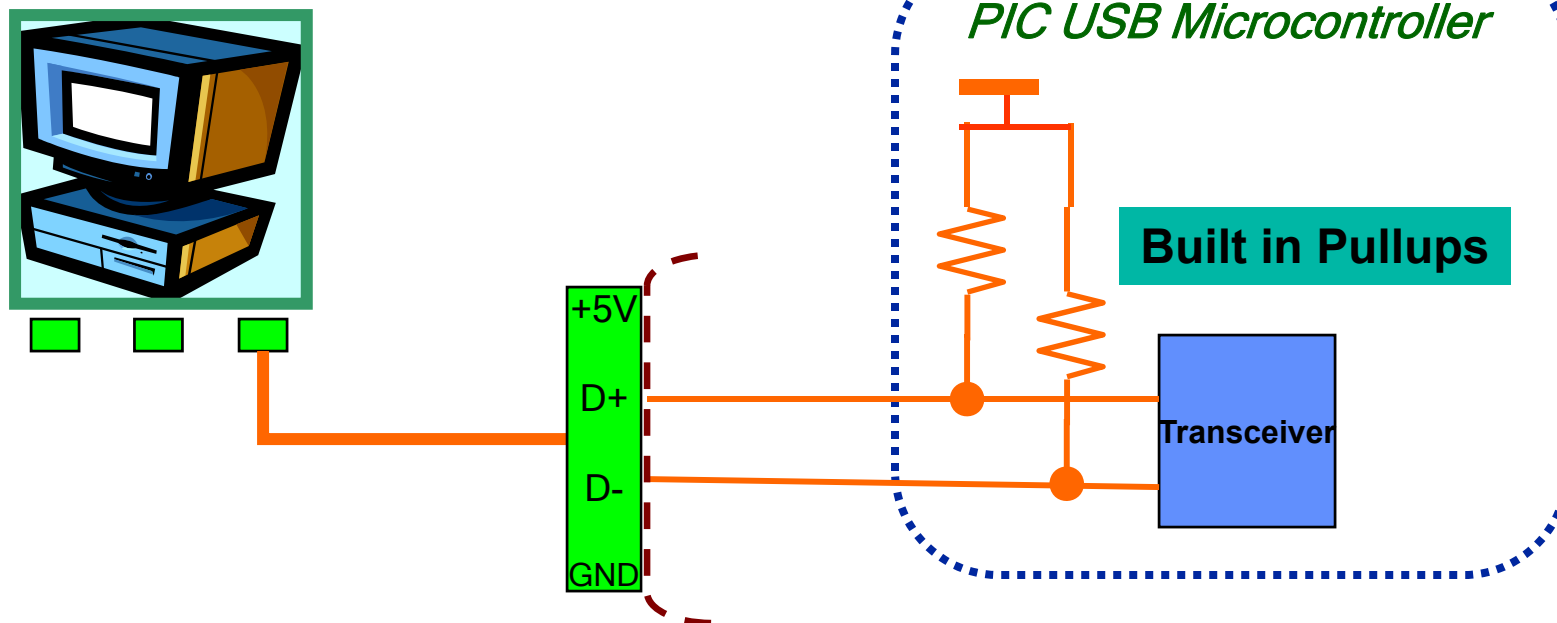


Productos USB OTG Ofrecidos

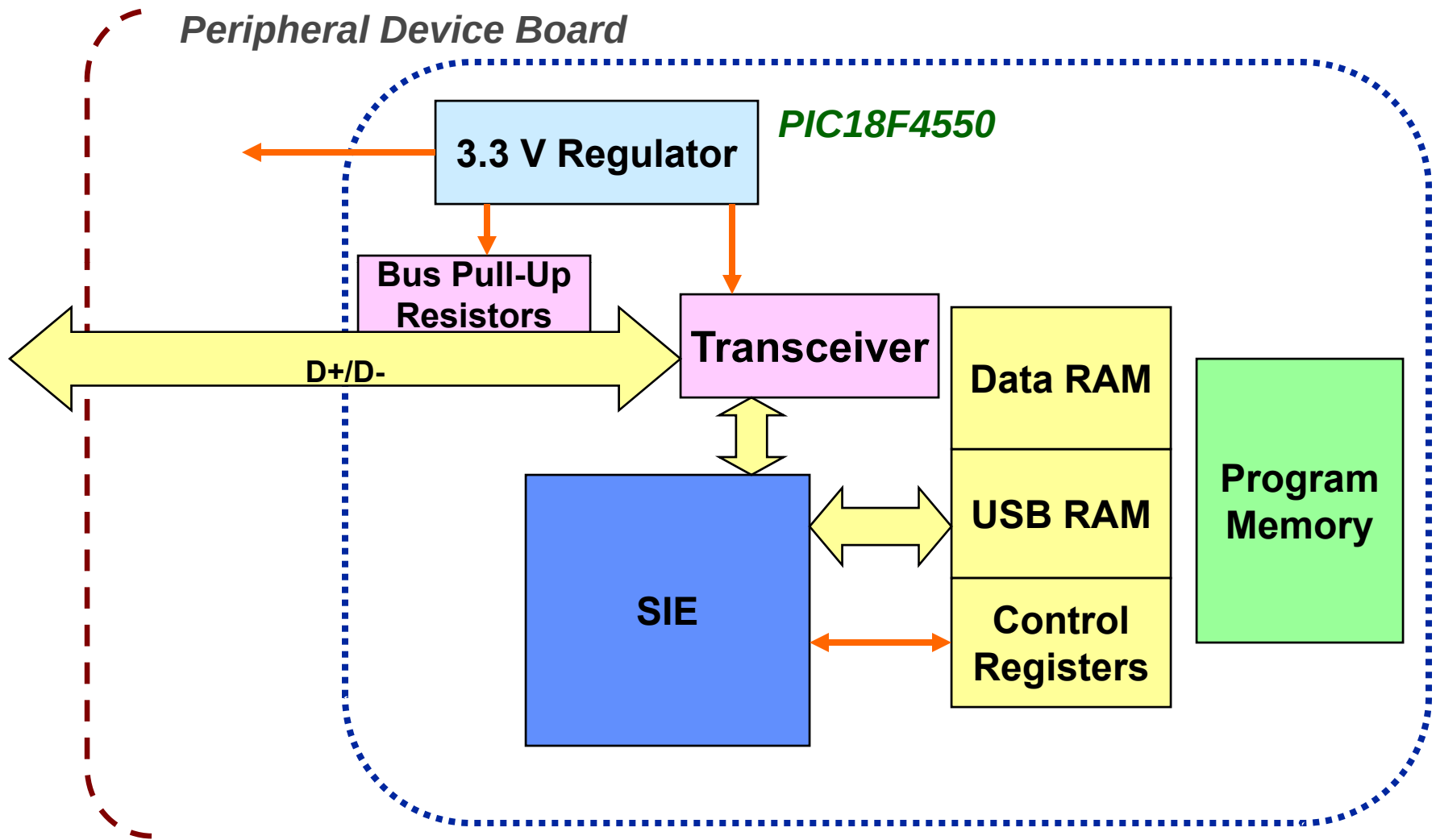


USB de MCU PIC de Microchip's

- Suporta USB 2.0 Low y Full Speed
- Transceptores, Reguladores de Voltaje y resistores de pull up sobre el chip
- PLL para el Clock USB sobre el chip
- USB Dual Port RAM
- no requiere hardware externo adicional para soportar USB



Módulo USB

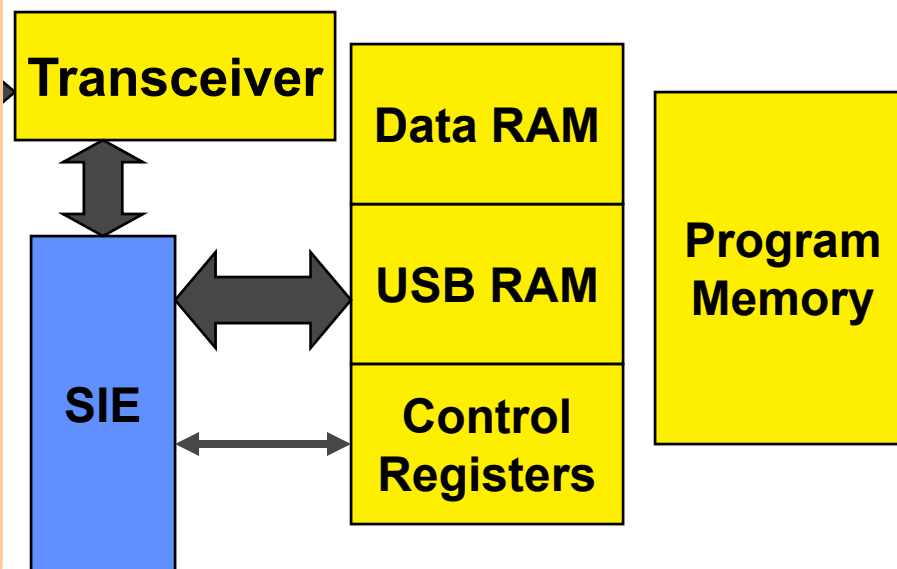


Motor de la Interfaz Serie

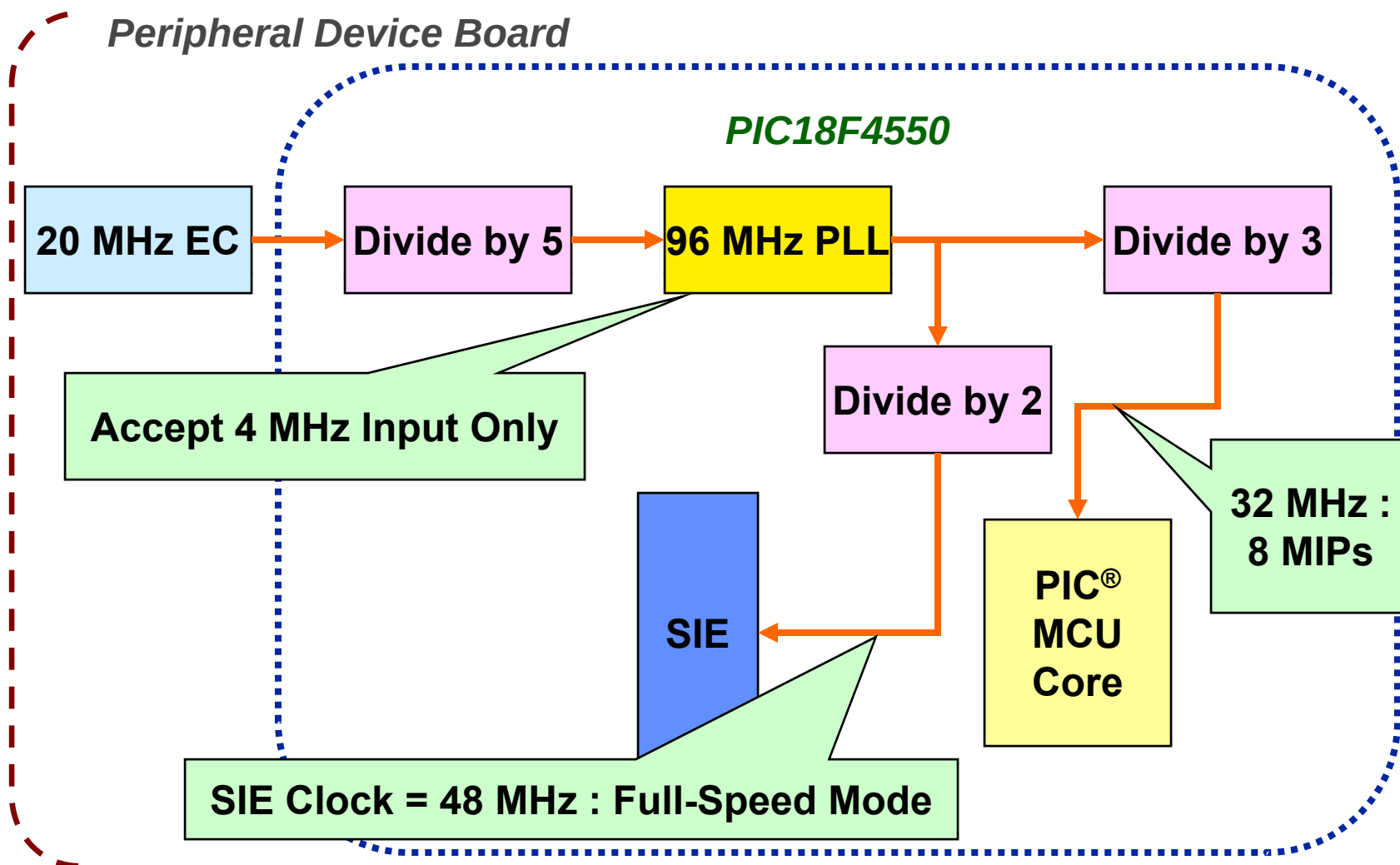
SIE ...

- Serializa y deserializa datos USB
- Codifica y decodifica datos NRZI
- Handles bit stuffing
- Cheque CRC para validar paquetes de datos
- Detecta señalizaciones de eventos en el Bus y notifica a la CPU para procesar interrupción
- Manejo de Transacciones USB
- Manejo del protocolo de handshaking

PIC18 USB MCU



Ejemplo de la configuración de Clock



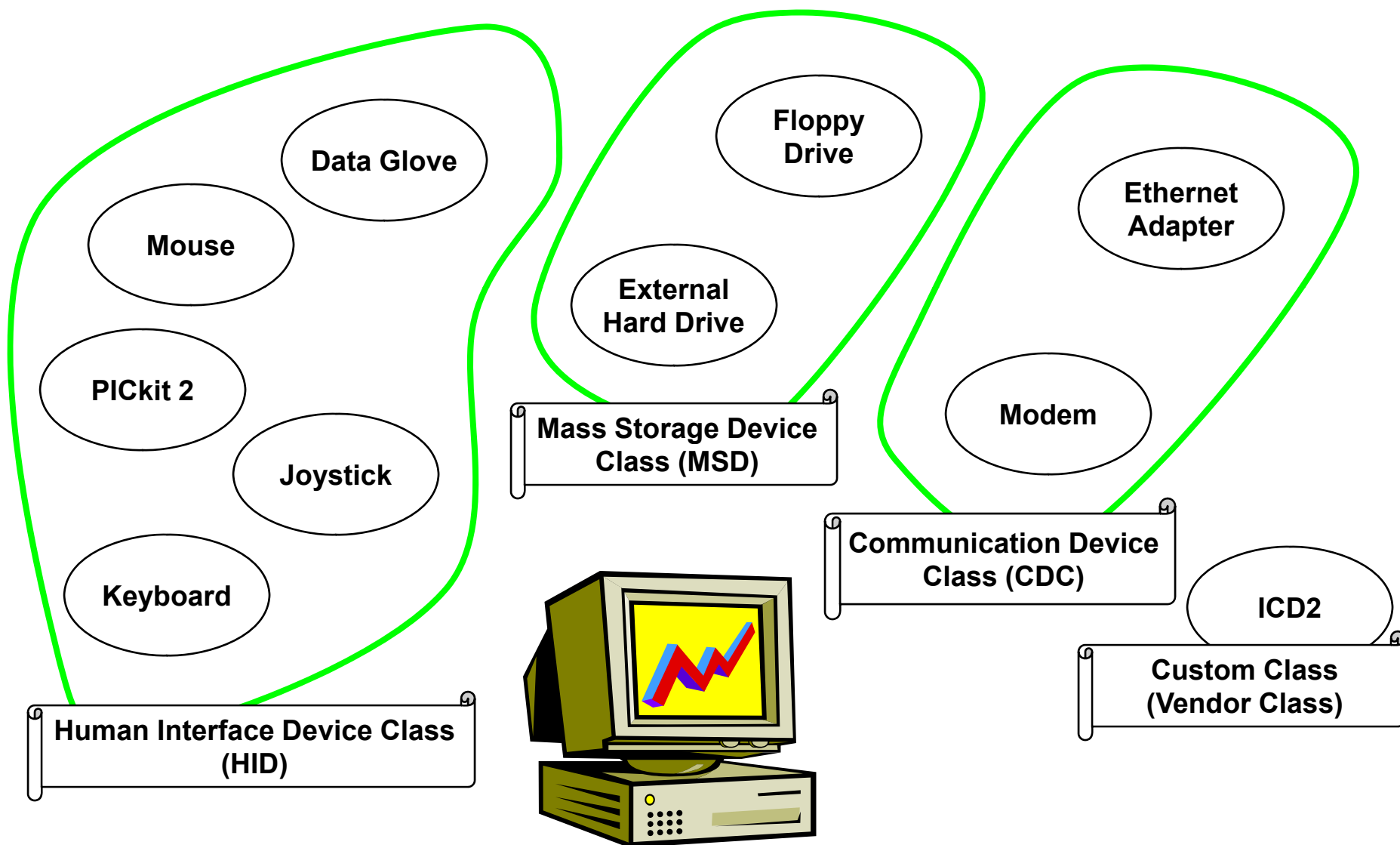
Agenda

- **El desafío de conectividad para Diseñar Sistemas Embebidos**
- **Revisión sobre USB**
- **Principios de Operación**
- **Soluciones en Microcontroladores PIC® para USB de MCHP**
- **Usando el Microchip USB Framework**
- **Drivers de PC para USB**
- **Microchip USB Herramientas de desarrollo**
- **Algunas Soluciones USB**
- **Resumen y Recusos**

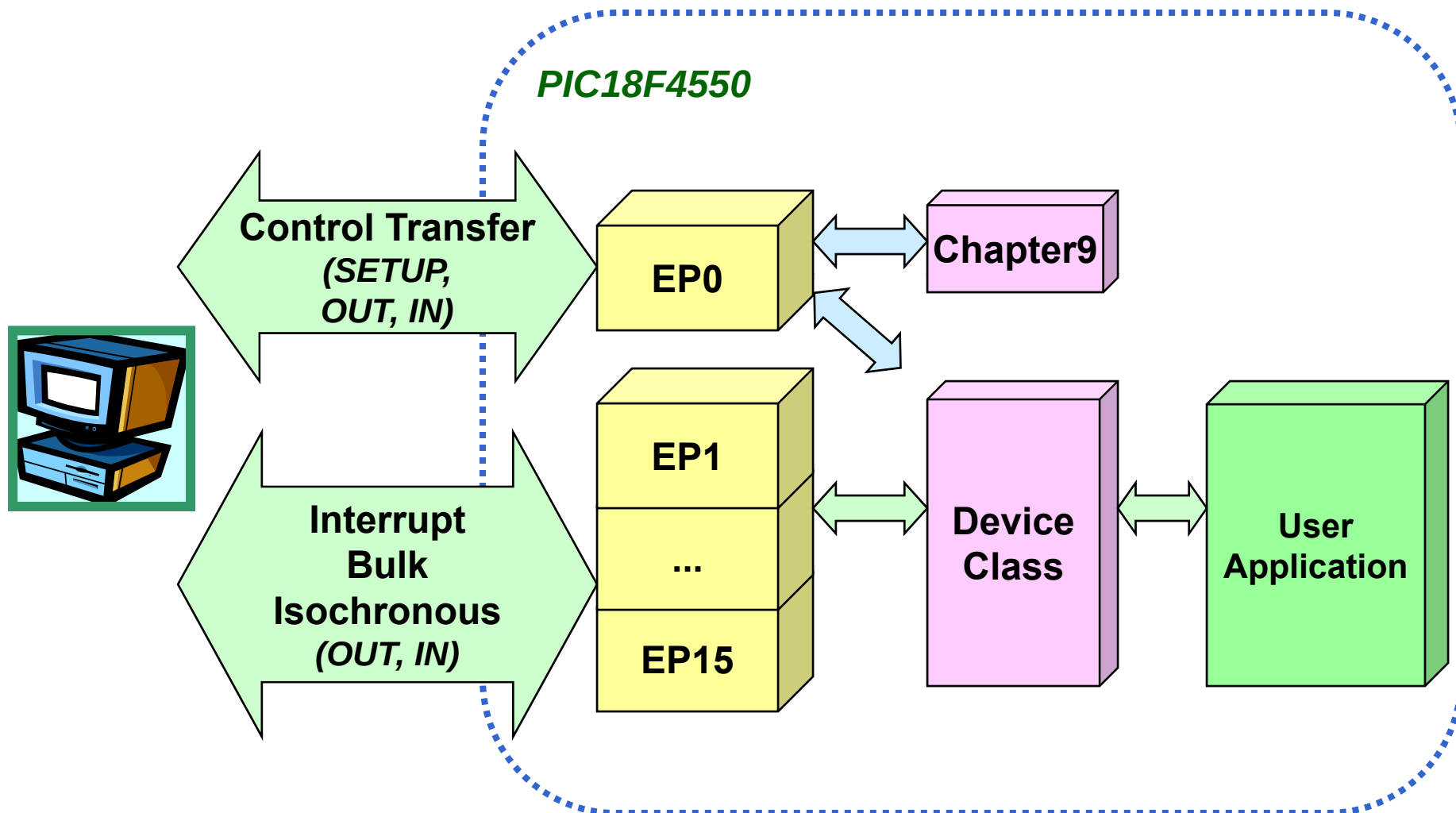
Microchip USB Firmware Características

- **C18 Compatible**
- **MPLAB® IDE Project Centric**
- **Polling Scheme**
- **Cooperative Multi-Tasking (No Blocking Routines)**
- **Program Memory Usage**
 - USB Enumeration (Chapter 9) - 3 KB
 - HID - 1 KB
 - CDC (RS-232 Emulation) - 1 KB
 - USB Mass Storage - 4 KB

MCHPFSUSB Clases de Dispositivos Frameworks Disponibles



El usar vs. Creando clase de Dispositivo



Dispositivo de Interfaz Humana (HID)

PIC[®] Microcontroller

HID\Mouse

PC Computer

Standard Windows Drivers

HID App

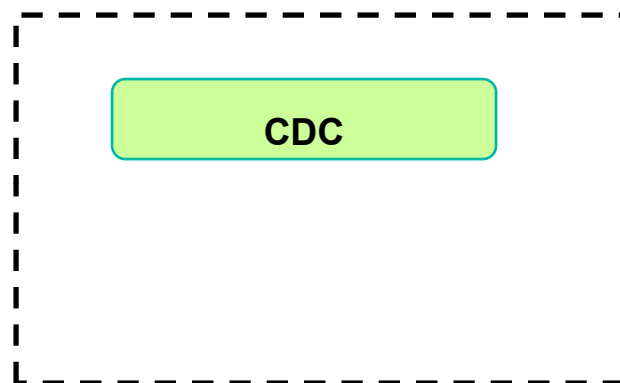
USB Cable

Design Considerations:

- 64 KB/s max
- Interrupt Transfer Type
- Standard Windows driver
- Custom PC application can access HID data through Win32 APIs

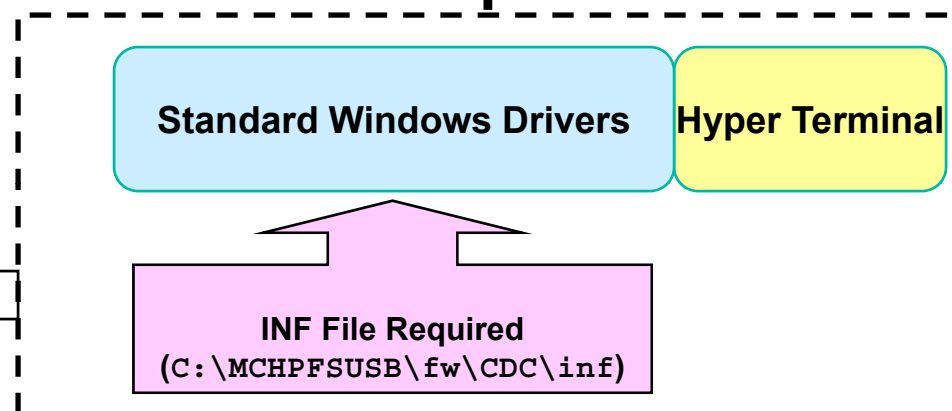
CDC - RS-232 Emulación

PIC[®] Microcontroller



USB Cable

PC Computer

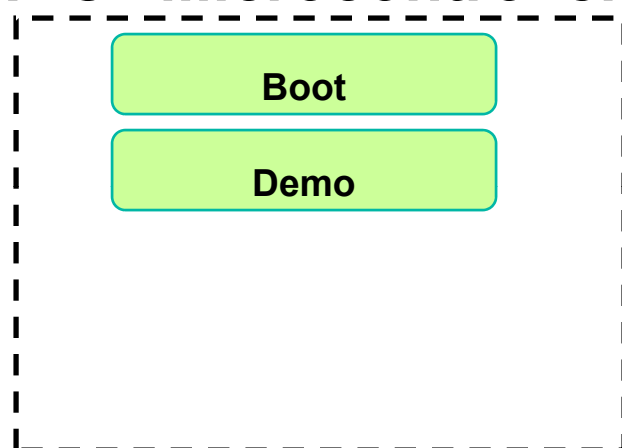


Design Considerations:

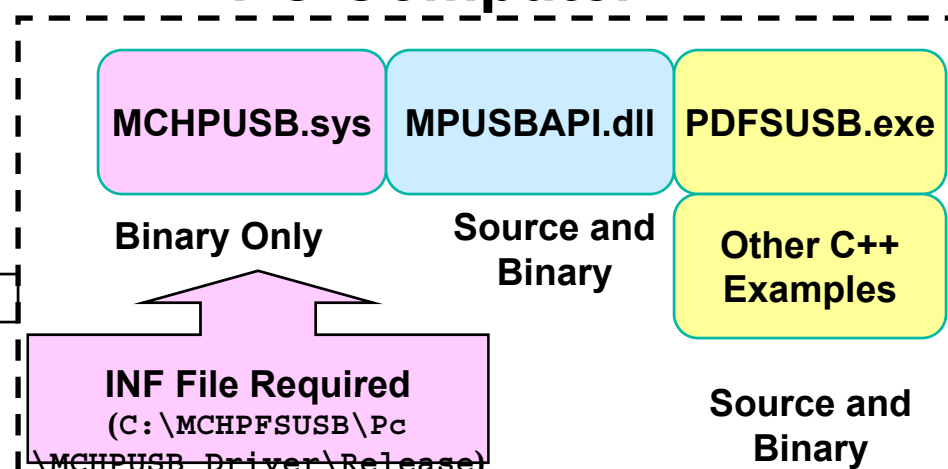
- ~80 KB/s max
- Bulk Transfers
- PC applications can access the device as though it is connected to a serial COM port

Personalizar Clase de Driver

PIC® Microcontroller



PC Computer



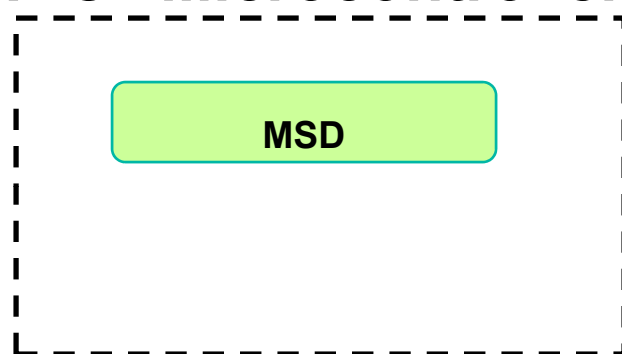
Design Considerations:

- ~1,088 KB/s max
- Very flexible (Control, Bulk, Int, Iso Transfers are possible)
- Not a standard Windows driver
- PC programming is required

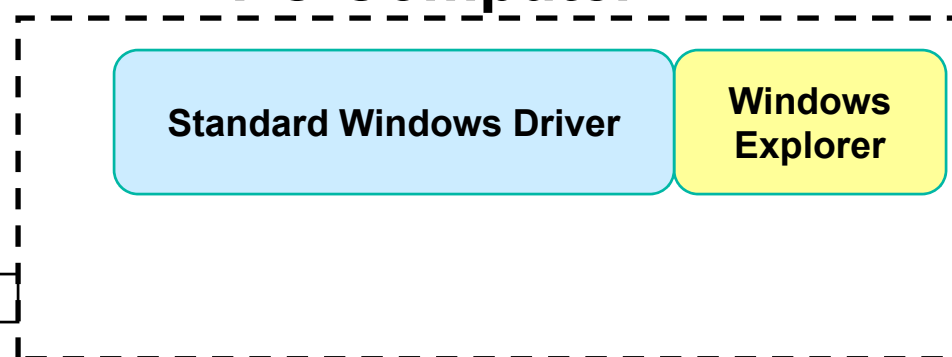
Dispositivo de almacenamiento masivo (MSD)

See AN1003: USB Mass Storage

PIC[®] Microcontroller



PC Computer

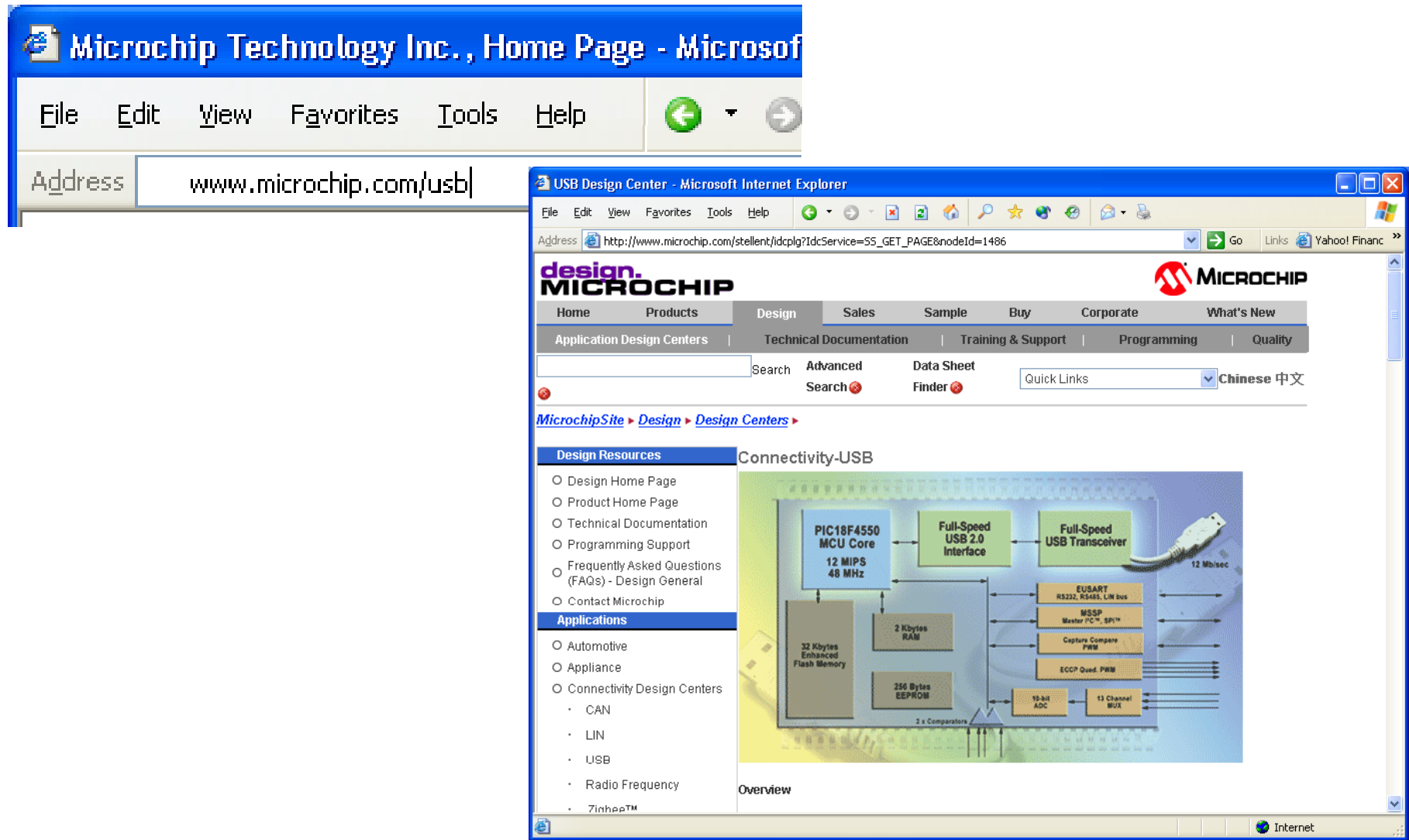


USB Cable

Design Considerations:

- Acts like a disk drive
- Rápida Transferencia de datos sobre USB
- Speed is limited depending on the physical media interface
- For usefulness, a File System should also be implemented in firmware

Descargando el Microchip USB Framework



Microchip Technology Inc., Home Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address

USB Design Center - Microsoft Internet Explorer

Address

design. MICROCHIP

Home Products Design Sales Sample Buy Corporate What's New

Application Design Centers Technical Documentation Training & Support Programming Quality

Search **Advanced Search** **Data Sheet Finder** Quick Links **Chinese 中文**

[MicrochipSite](#) > [Design](#) > [Design Centers](#) >

Design Resources

- Design Home Page
- Product Home Page
- Technical Documentation
- Programming Support
- Frequently Asked Questions (FAQs) - Design General
- Contact Microchip

Applications

- Automotive
- Appliance
- Connectivity Design Centers
 - CAN
 - LIN
 - USB
 - Radio Frequency
 - Zinhop™

Connectivity-USB

PIC18F4550 MCU Core
12 MIPS
48 MHz

Full-Speed USB 2.0 Interface

Full-Speed USB Transceiver

12 Mb/sec

32 Kbytes Enhanced Flash Memory

2 Kbytes RAM

256 Bytes EEPROM

2 x Comparators

EUSART RS232, RS485, LIN bus

MSSP Master I²C™, SPI™

Capture Compare PWM

ECCP Quad. PWM

10-bit ADC

13 Channel BUX

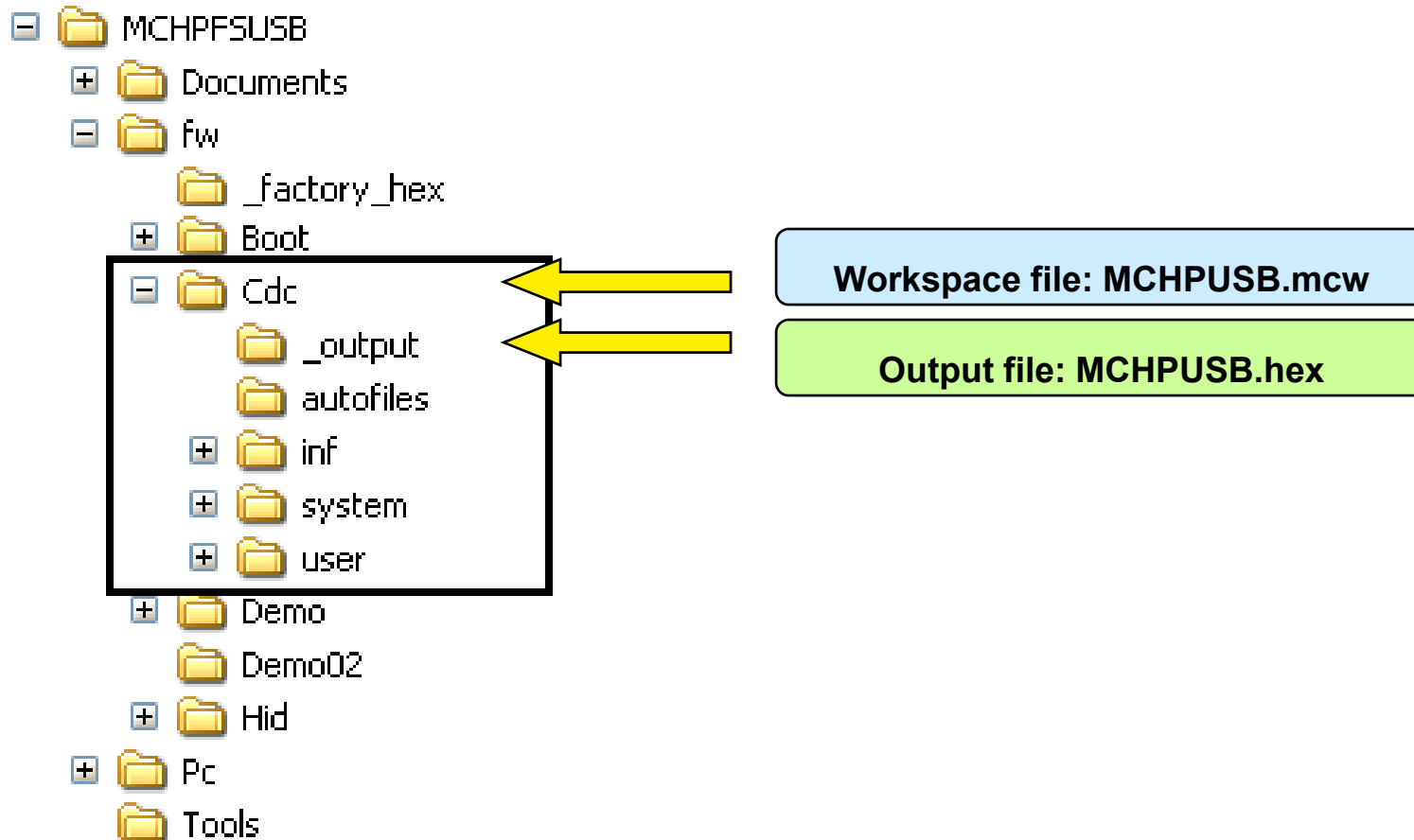
Overview

Internet

MCHPFSUSB Firmware

- Project Folder Structure -

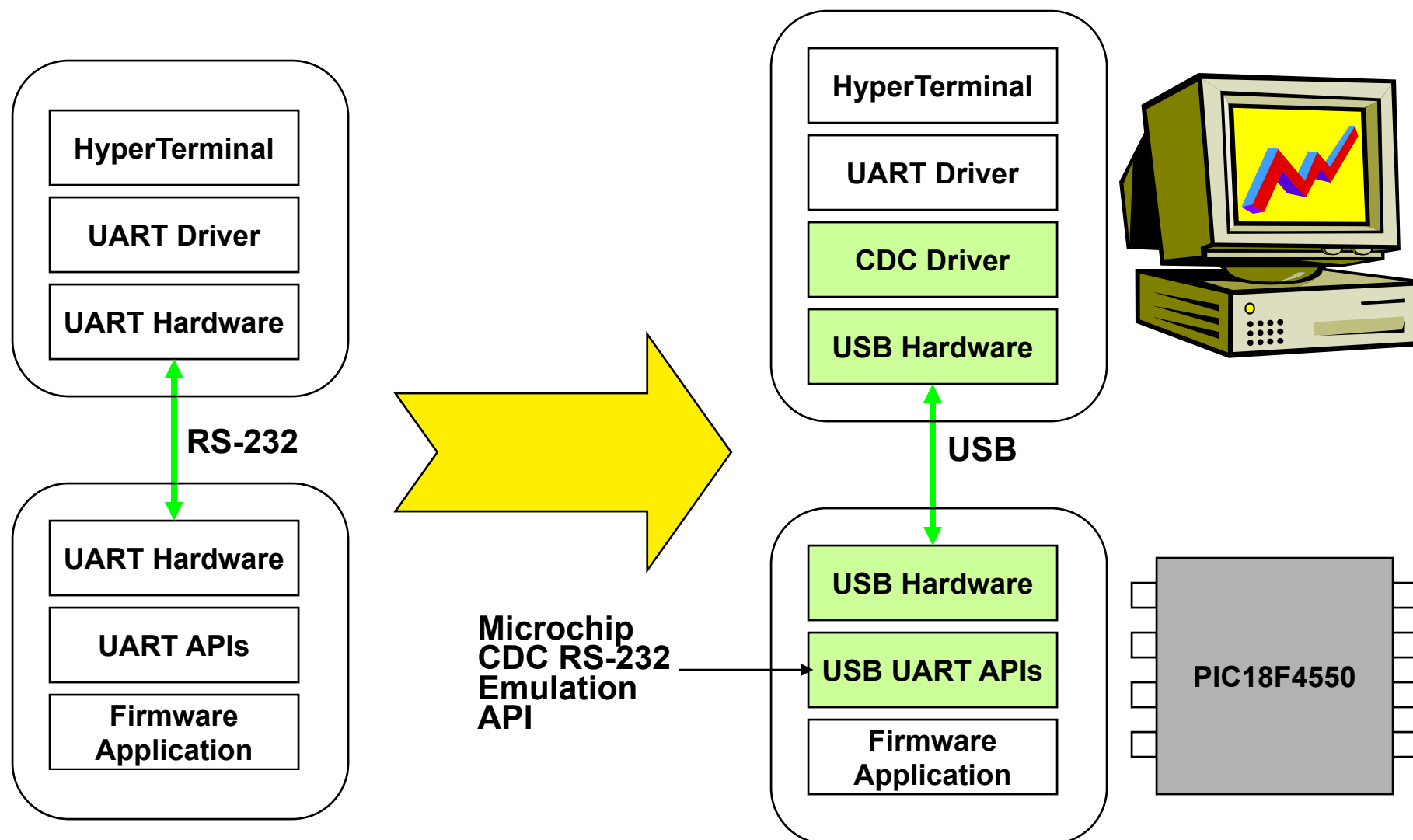
- C:\MCHPFSUSB



Demo 1 & 2

- **Demo 1:** General Purpose Driver Demo
- **Demo 2:** Upload HID mouse example using the bootloader

Aplicación de un PORT COM“Virtual”



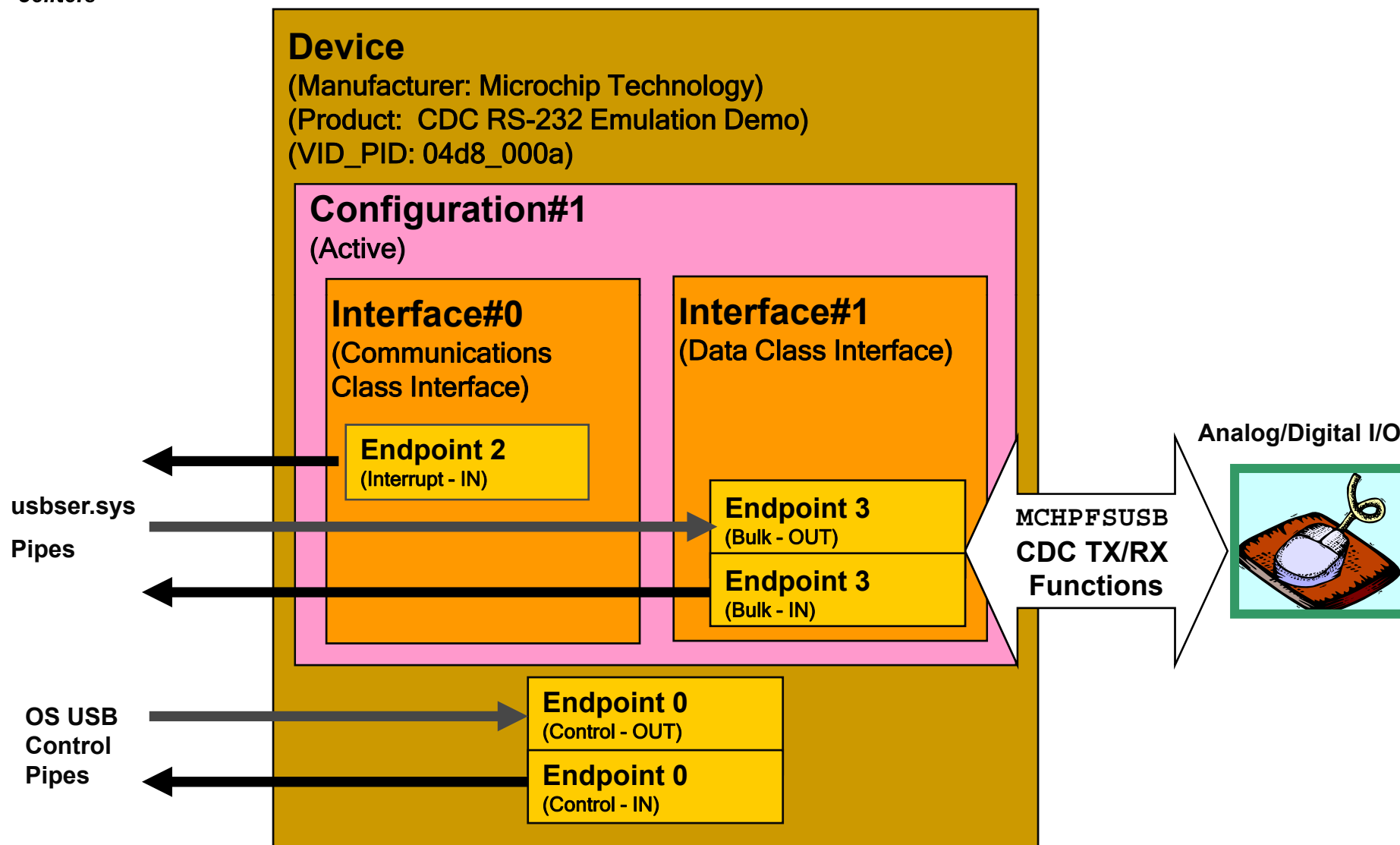
Las Clases de dispositivos de comunicaciones (CDC)

- **Abarca una variedad de dispositivos**
 - Dispositivos “POTs (Plain Old Telephone Service)”
 - **Teléfonos analógicos y modems, adaptador a terminal ISDN , Teléfonos celulares, “Virtual” COM ports**
 - Dispositivos de RED de “Velocidad Media”
 - **ADSL Modems, Cable Modems, Ethernet Adaptors**
- **La interfaz USB en estos dispositivos acarrea datos genéricos como un pozo como aplicaciones específicas encapsuladas protocolos de comandos , por ejemplo Terminal V25 para modems (pone comandos AT).**

Modelo Abstracto de Control (ACM) Emulación Serie (3.6.2.1)

- **ACM implementació cubierta de ports COM vituales**
- **Son requeridos 2 interfaces USB USB**
 - Communications Class Interface
 - **1 - Int IN Ep**
 - **Used to notify host of the current RS-232 connection status from the USB RS-232 emulated device**
 - Data Class Interface
 - **1 – Bulk IN Ep & 1 – Bulk OUT Ep**
 - **Used to transfer raw data bytes that would normally be transferred over RS-232**
 - **Also may be used to send encapsulated call control (AT) commands**

The “Logical” CDC-ACM USB Device



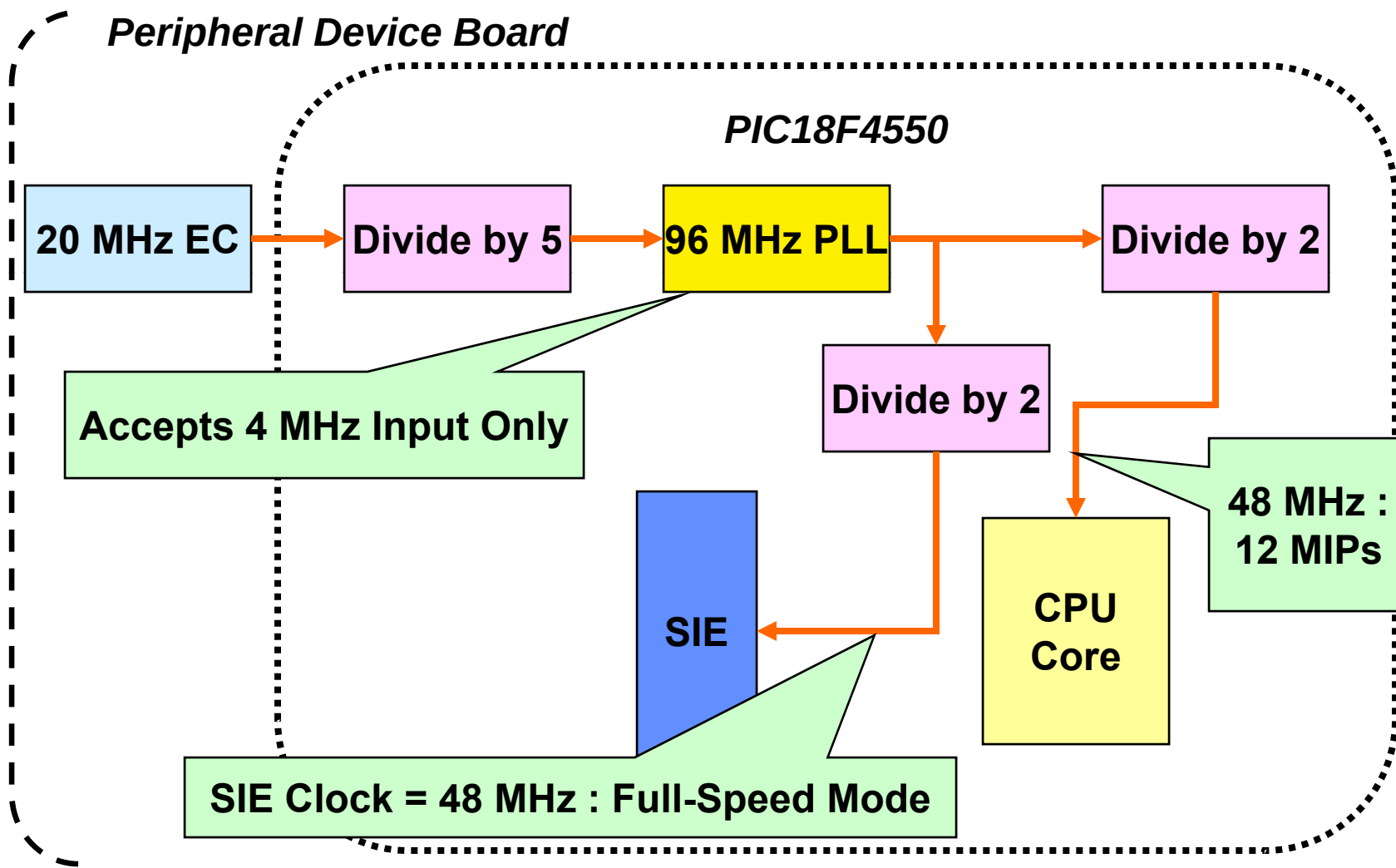
Proyecto CDC

- Clase ACM-Requerimientos específicos -

- **Resumen de clase-especificar requisitos y notificaciones implementadas (ver 6.2 & 6.3 en CDC especificaciones para descripción):**
 - las implementaciones CDC firmware tratadas para
 - **SEND_ENCAPSULATED_COMMAND**
 - **GET_ENCAPSULATED_RESPONSE**
 - **SET_LINE_CODING**
 - **GET_LINE_CODING**
 - **SET_CONTROL_LINE_STATE**
 - Sin embargo, el CDC firmware corrientemente retorna una **NAK** para decir al host que no hay respuesta disponible
 - **Usted no necesita preocuparse por atender estos requerimientos**
 - **Solo envíe/reciva su dato crudo vía el CDC RS-232 funciones API**

Proyecto CDC

- Clock Configuración -



CDC APIs

```
void putrsUSBUSART(const rom char *data);
```

```
void putsUSBUSART(char *data);
```

```
void mUSBUSARTTxRom(rom byte *pData, byte len);
```

```
void mUSBUSARTTxRam(byte *pData, byte len);
```

```
byte getsUSBUSART(char *buffer, byte len);
```

```
byte mCDCGetRxLength(void);
```

Determinantes del API

- **Usted es quien intercambia la nueva linea ASCII strings terminada (modo canónico) o bytes crudos (modo no-canónico) con la aplicación host ?**
- **El dato es almacenado en ROM o RAM?**

API para Null-terminated Literal & ROM Data

```
void putrsUSBUSART(const rom char *data);
```

```
void putsUSBUSART(char *data);
```

```
void mUSBUSART(rom byte *pData, byte len);
```

```
void mUSBUSART(rom byte *pData, byte len);
```

```
byte mUSBUSART(rom byte *pData, byte len);
```

```
byte mUSBUSART(rom byte *pData, byte len);
```

Use 'putrs' para datos literales almacenados en ROM

Ejemplos de Uso:

putrsUSBUSART("Hello");

or

putrsUSBUSART(welcome);

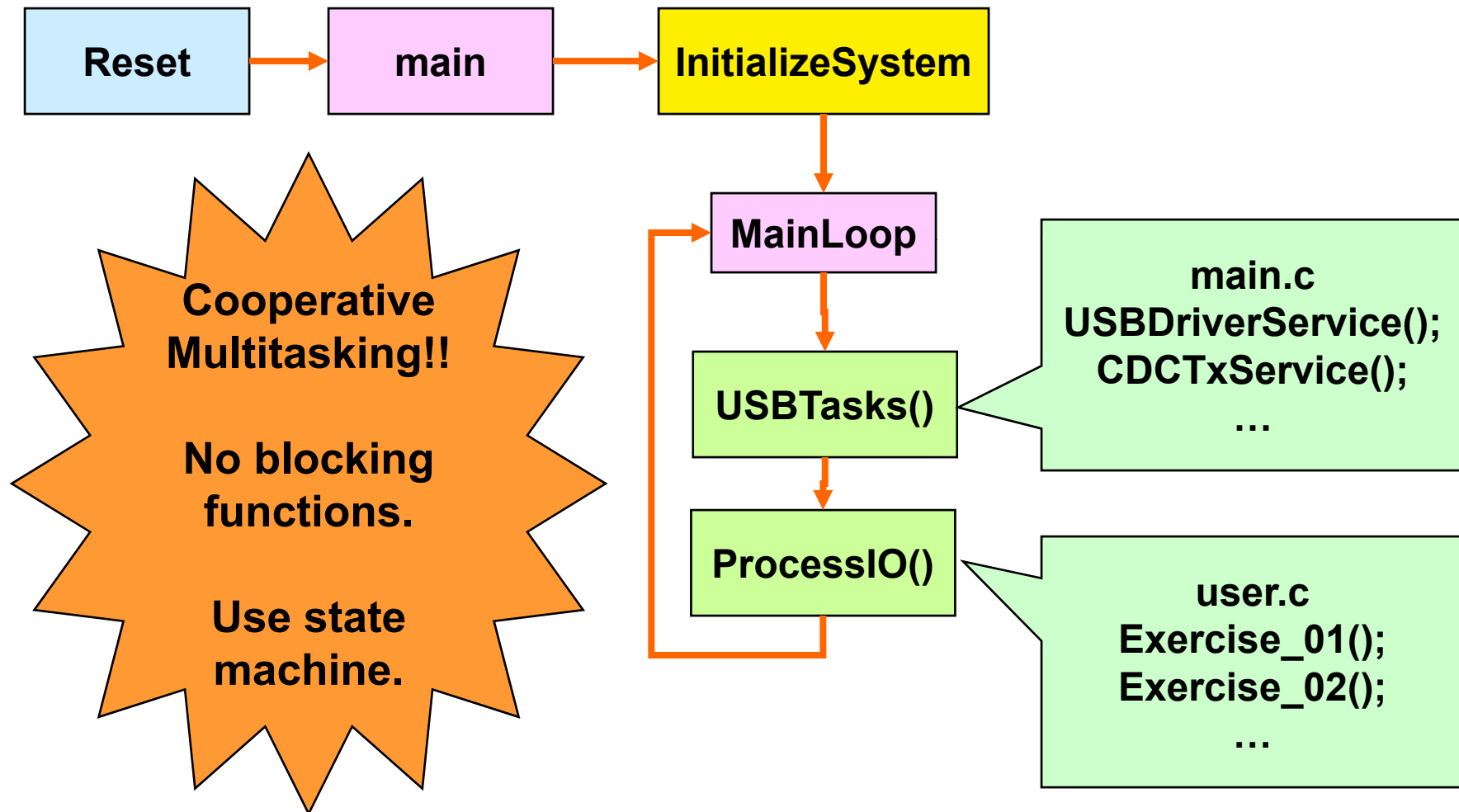
where

rom char welcome[]=("Bye");

Importancia del estado Checking

- **Estado cdc_trf_**
 - CDC_TX_READY
 - CDC_TX_BUSY
 - CDC_TX_BUSY_ZLP
 - CDC_TX_COMPLETING
- **Cuando llamando un API que envía datos para el host, necesita chequear que:**
cdc_trf_state == CDC_TX_READY
- **O use macro:**
 - **BOOL mUSBUSARTIsTxTrfReady(void);**
Retorna '1' cuando el estado es CDC_TX_READY

Diagrama de Flujo del Programa



Post Demo Análisis

- Speed
 - **640 Kbits/s = 80 Kbytes/s**
 - **Faster than RS-232 (UART)**
 - **Esta solución no es posible con un dispositivo USB low-speed porque este no tiene bulto endpoint**
- No hardware handshakes
- Memoria de Programa Usada: ~ 4 KB
- Nota de Aplicación
 - **AN956: Migrating Applications to USB from RS-232 UART with Minimal Impact on PC Software**
- **Que hacer si necesita un rango de transferencia de datos mas rápido?**
 - Considere usar nuestras class de driver windows personalizadas y dispositivos framework...

Agenda

- **The Connectivity Challenge for Embedded Application Designers**
- **USB Overview**
- **Principles of USB Operation**
- **PIC® Microcontroller solutions for USB from MCHP**
- **Using the Microchip USB Framework**
- **USB PC Drivers**
- **Microchip USB Development Tools**
- **Some USB Solutions**
- **Summary & Resources**

mchpusb.sys vs. HID vs. CDC

Features	mchpusb.sys (v1.0.0.6)	HID	CDC
Driver support built into Windows	No	Yes	Yes (need .inf)
64-bit PC Support	Yes	Yes	Yes
XP Ready	Yes	Yes	Yes
Vista Ready	Yes	Yes	Yes
Transfer Types (not incl. Ep0)			
Control	Yes	Yes	No
Iso	Yes	No	No
Interrupt	Yes	Yes	Yes
Bulk	Yes	No	Yes
Max Speed	~1.0 MB/s	64KB/s	~80kByte/s

Otras Clases de Drivers personalizados...

- **LibUSB**
 - Open Source
 - Supports Windows and Linux
 - Supports Interrupt, Bulk, and Isochronous transfers
 - <http://libusb-win32.sourceforge.net/>
- **WinUSB**
 - Supported in Windows XP & Vista only
 - Supports Interrupt and Bulk
 - <http://msdn2.microsoft.com/en-us/library/aa476426.aspx>

MPUSBAPI

- Introducción -



- **Código de la envoltura Win32 API sistema llama funciones (comuniquese en ultima instancia con `mchpusb.sys`)**
- **Trabaja con todos los dispositivos Microchip USB**
- **Trabajah Visual C++ .NET, C++ Builder, Visual Basic .NET, y mas...**
- **Transferencia de datos Facil y Flexible**
- **Suporta transferencias Bulk, Interrupt, Control y Isochronous**
- **Interfaz Funcional simplifica el desarrollo de aplicaciones**
- **Libre**

Microchip Driver de dispositivo USB de Propósito General

- mchpusb.inf – Este archivo contiene la información del driver del dispositivo USB Microchip
- mchpusb.sys - Microchip USB Device Driver
- Estos archivos pueden encontrarse en la siguiente ruta de acceso
 - C:\MCHPFSUSB\Pc\MCHPUSB Driver\Release

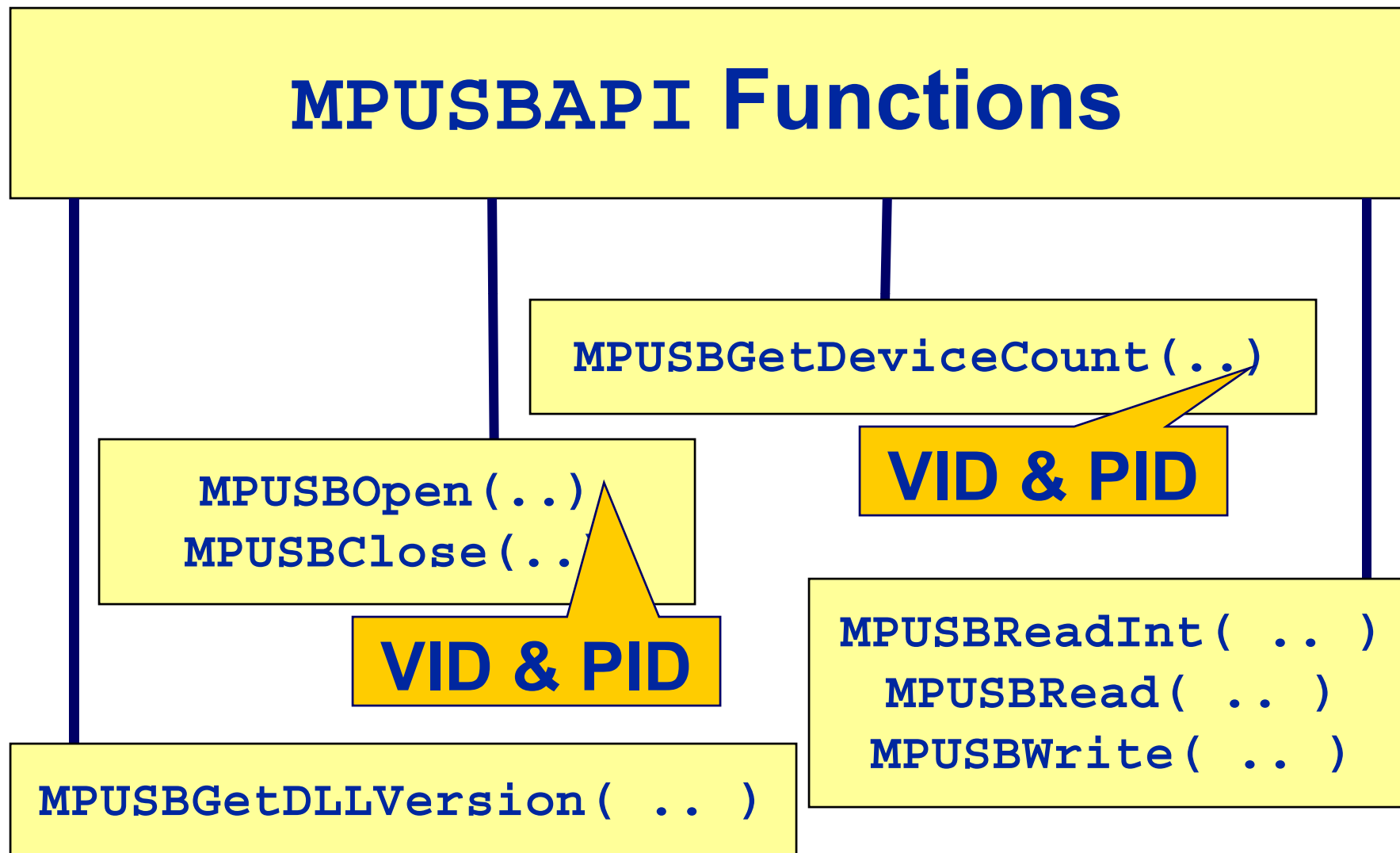
mchpusb.inf

- **.inf file tells windows how to configure and use a device**
- **Device Manager reads the VID and PID from the target device**
- **Windows uses .inf file to:**
 - Associates driver to use with VID & PID
 - Specifies device identification strings
 - Specifies source and destination of device files
 - Sets registry keys

.inf File Modifications

- **Modifiable sections in mchpusb.inf**
 - [DeviceList]
 - %DESCRIPTION%=DriverInstall, USB\VID_04D8&PID_000B,
USB\VID_04D8&PID_000C
 - [Strings]
 - ONEYSOFT="Walter Oney Software"
MFGNAME="Microchip Technology, Inc."
INSTDISK="Microchip Technology, Inc.
Installation Disc"
DESCRIPTION="PIC18F4550 Family Device"

Functions in MPUSBAPI.DLL



Summary - DLL Function Transfer Type Usage

<u>Transfer/ Endpoint Type</u>	<u>Functions</u>	<u>Time-Out Applicable?</u>
Interrupt-IN	MPUSBRead, MPUSBReadInt	Yes
Interrupt-OUT	MPUSBWrite	Yes
Bulk-IN	MPUSBRead	Yes
Bulk-OUT	MPUSBWrite	Yes
Isochronous-IN	MPUSBRead	No
Isochronous-OUT	MPUSBWrite	No

Custom Class Firmware API

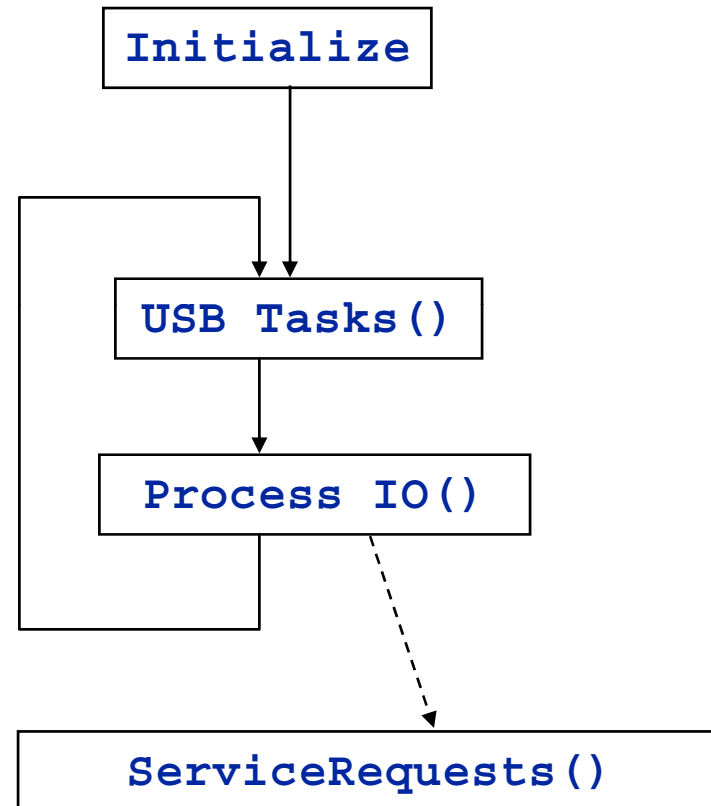
- See `usbgen.c` & `usbgen.h` -

- **`void USBGenInitEP(void);`**
 - Called after a SET CONFIGURATION request
 - Initializes endpoints, state machine and variables
- **`void USBGenWrite(byte *buffer, byte len);`**
 - Copies data from user's buffer to the Ep buffer
- **`byte USBGenRead(byte *buffer, byte len);`**
 - Copies data from the Ep buffer to the user's buffer
- **`(bit) mUSBGenRxIsBusy(void);`**
 - Macro checks if the OUT Ep is busy (owned by the SIE or not)
- **`(bit) mUSBGenTxIsBusy(void);`**
 - Macro checks if the IN Ep is busy (owned by the SIE or not)
- **`byte mUSBGenGetRxLength(void);`**
 - Retrieves the number of bytes copied into the user's buffer after the most recent call to `USBGenRead()`

Custom Class (“Demo”) Firmware Structure

Synchronous model

- Accepts commands from host
- Returns results
- Waits for next command



Inicializando los Periféricos PIC18F

- In `user.c`, a function has been defined for you to place application-specific peripheral initialization code:

```
void UserInit(void)
{
    mInitAllLEDs(); //macros defined in "io_cfg.h"

    mInitPOT();

} //end UserInit
```

- `UserInit()` is called in `main()` during system initialization.

Descriptor Tabla Seteos

```
File usbdsc.c
/* Device Descriptor */
rom USB_DEV_DSC device_dsc=
{
    sizeof(USB_DEV_DSC),    // Size of this descriptor in bytes
    DSC_DEV,                // DEVICE descriptor type
    0x0200,                 // USB Spec Release Number in BCD format
    0x00,                   // Class Code
    0x00,                   // Subclass code
    0x00,                   // Protocol code
    EP0_BUFF_SIZE,         // Max packet size for EP0, see usbcfg.h
    0x04D8,                 // Vendor ID
    0x000C,                 // Product ID: PICDEM FS USB (DEMO Mode)
    0x0000,                 // Device release number in BCD format
    0x01,                   // Manufacturer string index
    0x02,                   // Product string index
    0x00,                   // Device serial number string index
    0x01                     // Number of possible configurations
};
```

Endpoint Descriptors

File usbdsc.c

```
/* EndPoint Descriptor */
```

```
sizeof(USB_EP_DSC), DSC_EP, _EP01_OUT, _BULK, USBGEN_EP_SIZE, 1,  
sizeof(USB_EP_DSC), DSC_EP, _EP01_IN, _BULK, USBGEN_EP_SIZE, 1
```

```
_BULK or _INT (Interrupt)
```

```
USB_EP_SIZE defined in USBCFG.H
```

```
#define USBGEN_EP_SIZE 64
```

String Descriptors

```
rom struct{byte bLength;byte bDscType;word string[1];}sd000={  
sizeof(sd000),DSC_STR,0x0409};
```

```
rom struct{byte bLength;byte bDscType;word string[25];}sd001={  
sizeof(sd001),DSC_STR,  
'M','i','c','r','o','c','h','i','p',' ',  
'T','e','c','h','n','o','l','o','g','y',' ','I','n','c','.'};
```

```
rom struct{byte bLength;byte bDscType;word string[33];}sd002={  
sizeof(sd002),DSC_STR,  
'P','I','C','D','E','M',' ','F','S',' ','U','S','B',' ',  
'D','e','m','o',' ','B','o','a','r','d',' ','(','C',')',  
' ','2','0','0','7'};
```

Resumario

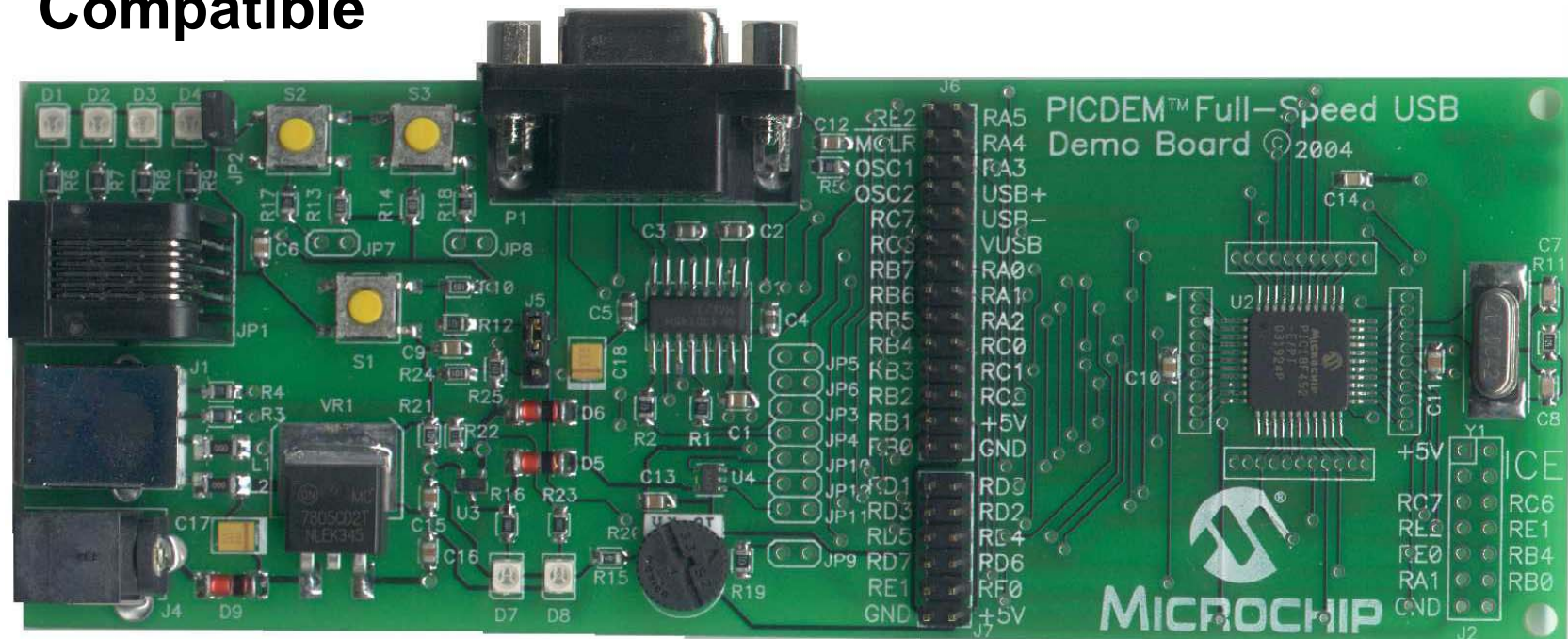
- **Drivers built into Windows for most USB devices**
- **A Custom Class Device needs a Vendor specific Device Driver**
- **mchpusb.sys file is a Vendor specific Device Driver**
- **/MCHPFSUSB/fw/demo contains the custom class device framework**
- **.inf Files**
- **MPUSBAPI.DLL simplifies your development Activities**

Agenda

- The Connectivity Challenge for Embedded Application Designers
- USB Overview
- Principles of USB Operation
- PIC® Microcontroller solutions for USB from MCHP
- Using the Microchip USB Framework
- USB PC Drivers
- **Microchip USB Development Tools**
- Some USB Solutions
- Summary & Resources

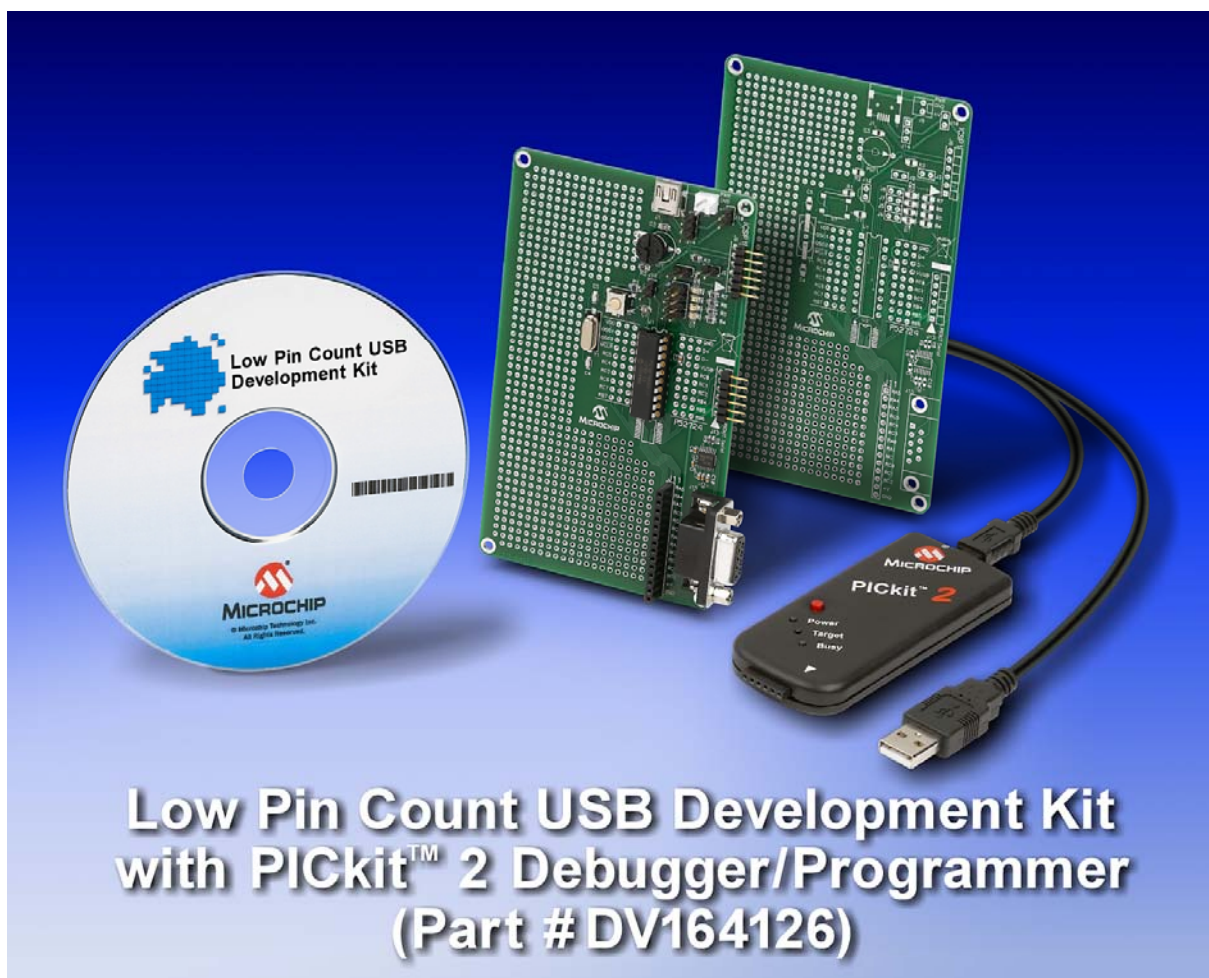
PICDEM™ Full-Speed USB

- PIC18F4550, 20 MHz xtal
- USB Port
- Serial Port
- PICtail™ Daughter Board Compatible
- Power LED Indicators
- Potentiometer
- Temperature Sensor



Low Pin Count USB Development Kit

- Demo board for PIC18F14K50
- Connects directly to PICKit 2
- USB 2.0 Port
- RS232 Port
- LEDs
- Potentiometer
- Breadboard area
- Software for USB peripheral devices

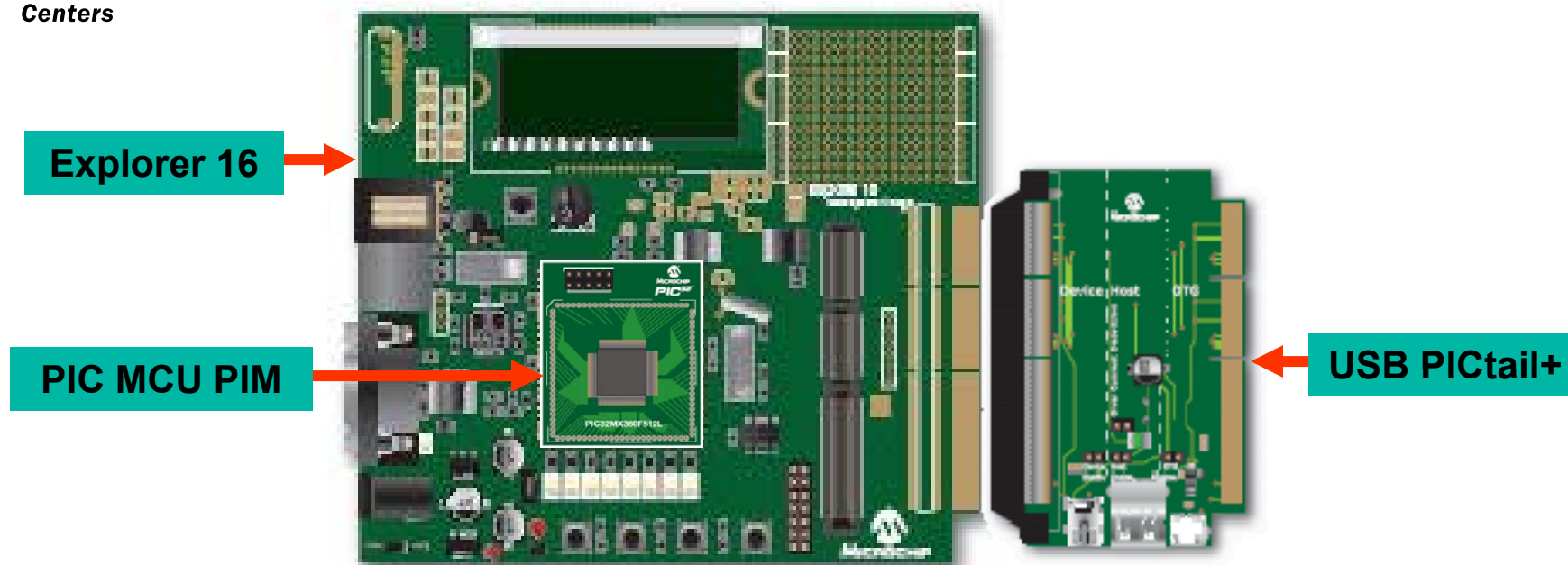


PIC18F87J50 FS USB Plug-In Module (PIM)

- PIC18F87J50, 12 MHz xtal
- USB Full Speed Port
- LEDs,
- Pushbuttons
- Plugs as PIM on other boards

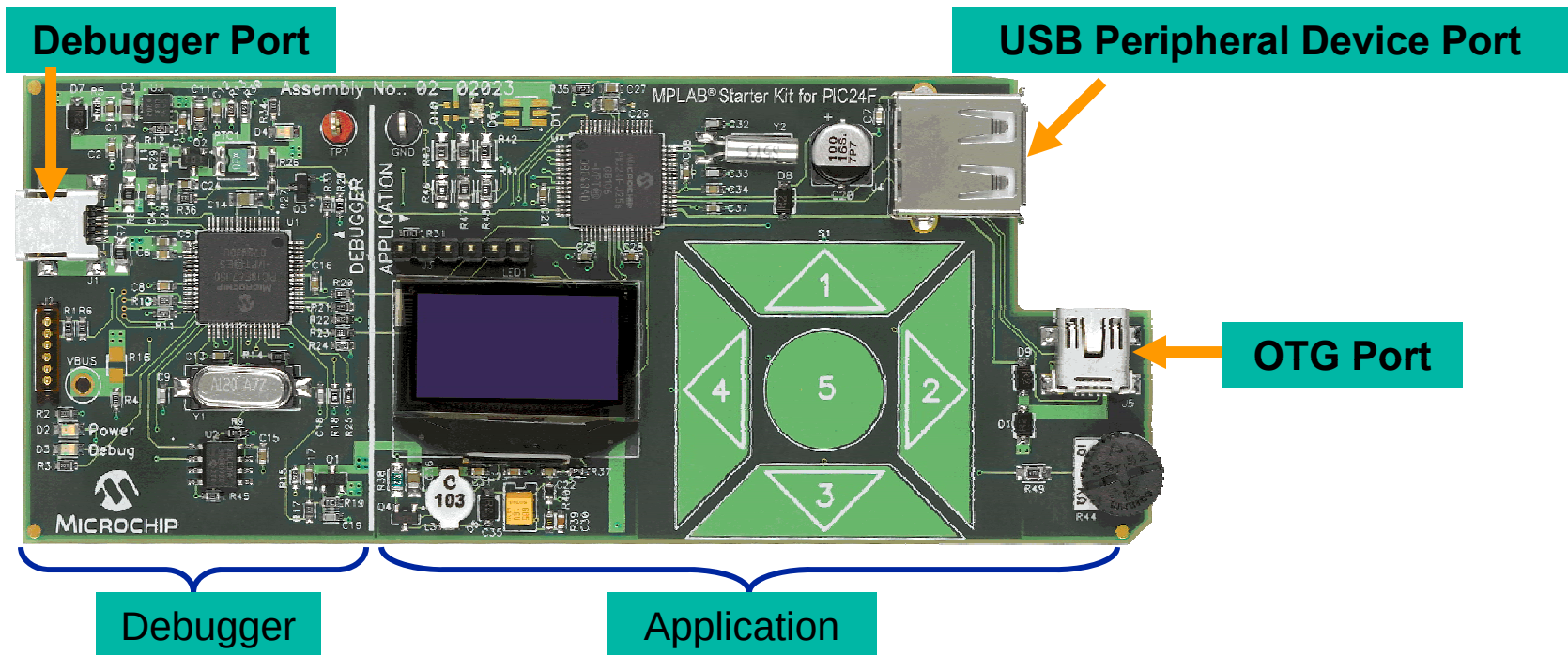


Explorer 16 USB Demo boards



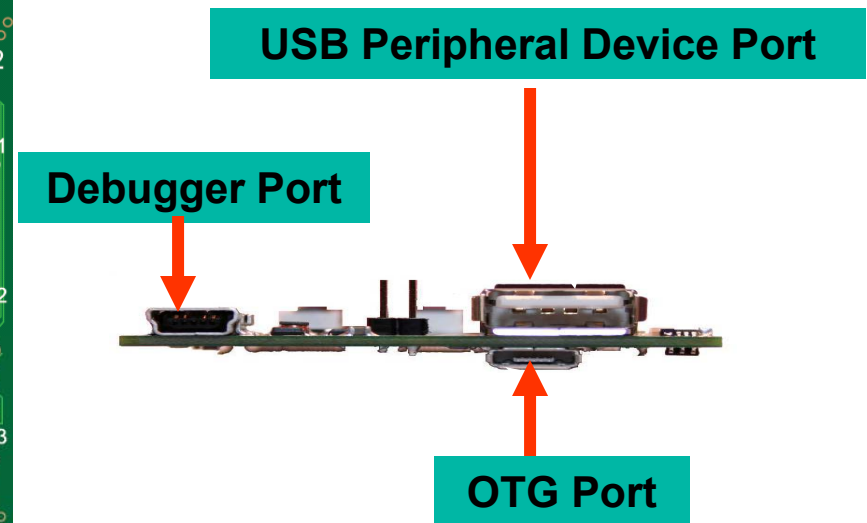
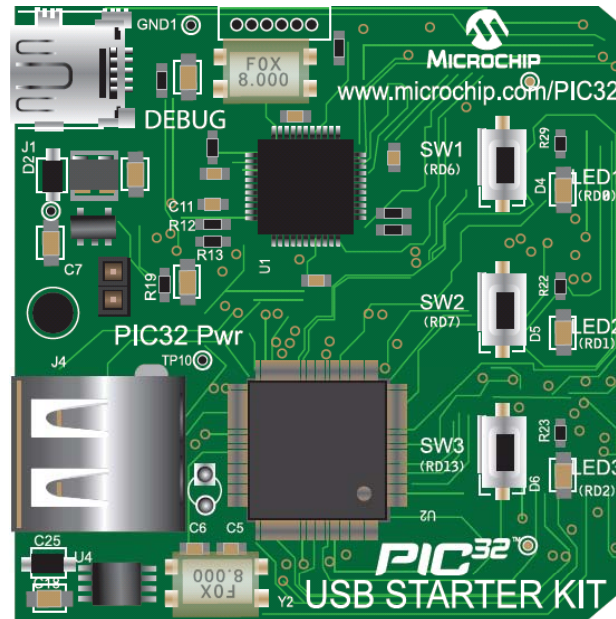
- Demo board for all 16 and 32bit PIC MCUs
- Requires 16 or 32 bit USB PIC MCU PIM
- Requires USB PICtail+ Daughter Board
- Supports OTG, Embedded Host and Device
- USB OTG stacks for Device and host
- Class drivers for Generic, HID, and Mass Storage

PIC24F Starter Kit



- The PIC24F Starter Kit is based on the PIC24 USB MCU
- Highlights the PIC24 as a Embedded Host or OTG
- USB Flash Drive Interface
- Interactive Display and Touch keypad interface
- Built-in MPLAB compatible Debugger
- Board powered by USB port to PC

PIC32 USB Starter Board



- The PIC32 USB Starter Board is based on the PIC32MX460F512 USB MCU
- Highlights the PIC32 as an Embedded Host or OTG
- USB Peripheral Device Interface
- Built-in MPLAB compatible Debugger
- Board powered by USB port to PC

USB Development Support

- **Microchip USB Design Center**



- **Buy USB Development Boards and Kits from microchipDirect (www.microchipdirect.com)**
- **Detailed Product Information at:**
 - www.microchip.com/pic32
 - www.microchip.com/pic24
 - www.microchip.com/pic18

Agenda

- **The Connectivity Challenge for Embedded Application Designers**
- **USB Overview**
- **Principles of USB Operation**
- **PIC® Microcontroller solutions for USB from MCHP**
- **Using the Microchip USB Framework**
- **USB PC Drivers**
- **Microchip USB Development Tools**
- **Some USB Solutions**
- **Summary & Resources**

Microchip's USB PIC MCU Solutions

- **USB Device, Embedded Host or OTG Applications:**
 - Real world interface
 - **Collect data on Temperature, Pressure, weight, etc.**
 - **Control heaters, motors, displays etc.**
 - Interface to a PC, Embedded Host or USB Device
 - **For data analysis, display, storage, etc.**

USB Solutions from Microchip

Complexity and Speed



Full-Speed On-The-Go (12 Mbps)
Full Speed USB Embedded Host or Device
"Negotiates" with other OTG Products
Copy Box, PDA, Portable Instrument



Full-Speed Embedded Host (12 Mbps)
Full Speed USB Embedded Host
Security System, Download Files, Upload Software



Full-Speed Device (12 Mbps)
Full Speed USB Device
Biometrics, Data loggers

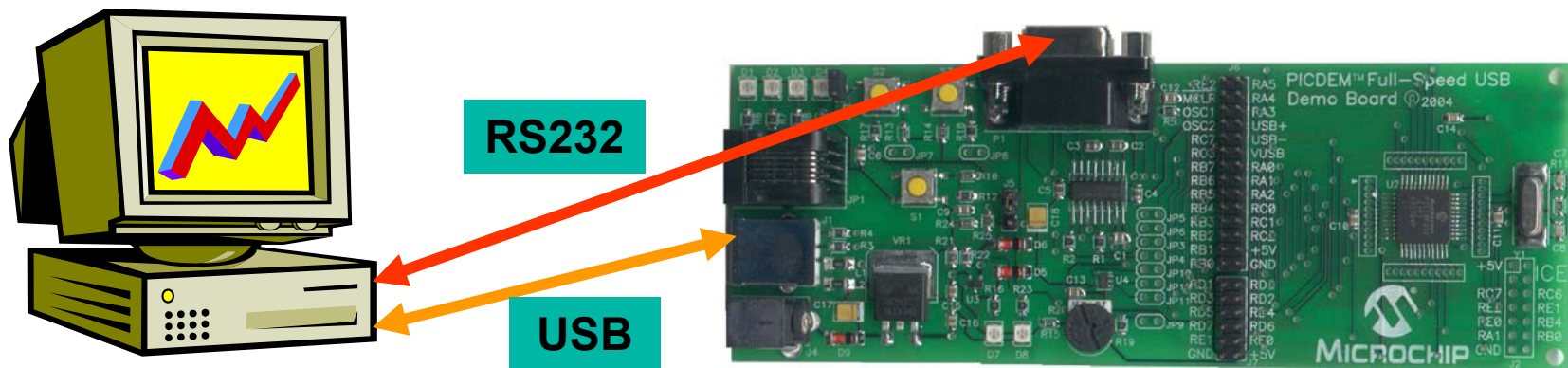


Low-Speed Device (1.5 Mbps)
Low Speed USB Device
Mouse, Keyboard, Speaker, Cell Charger

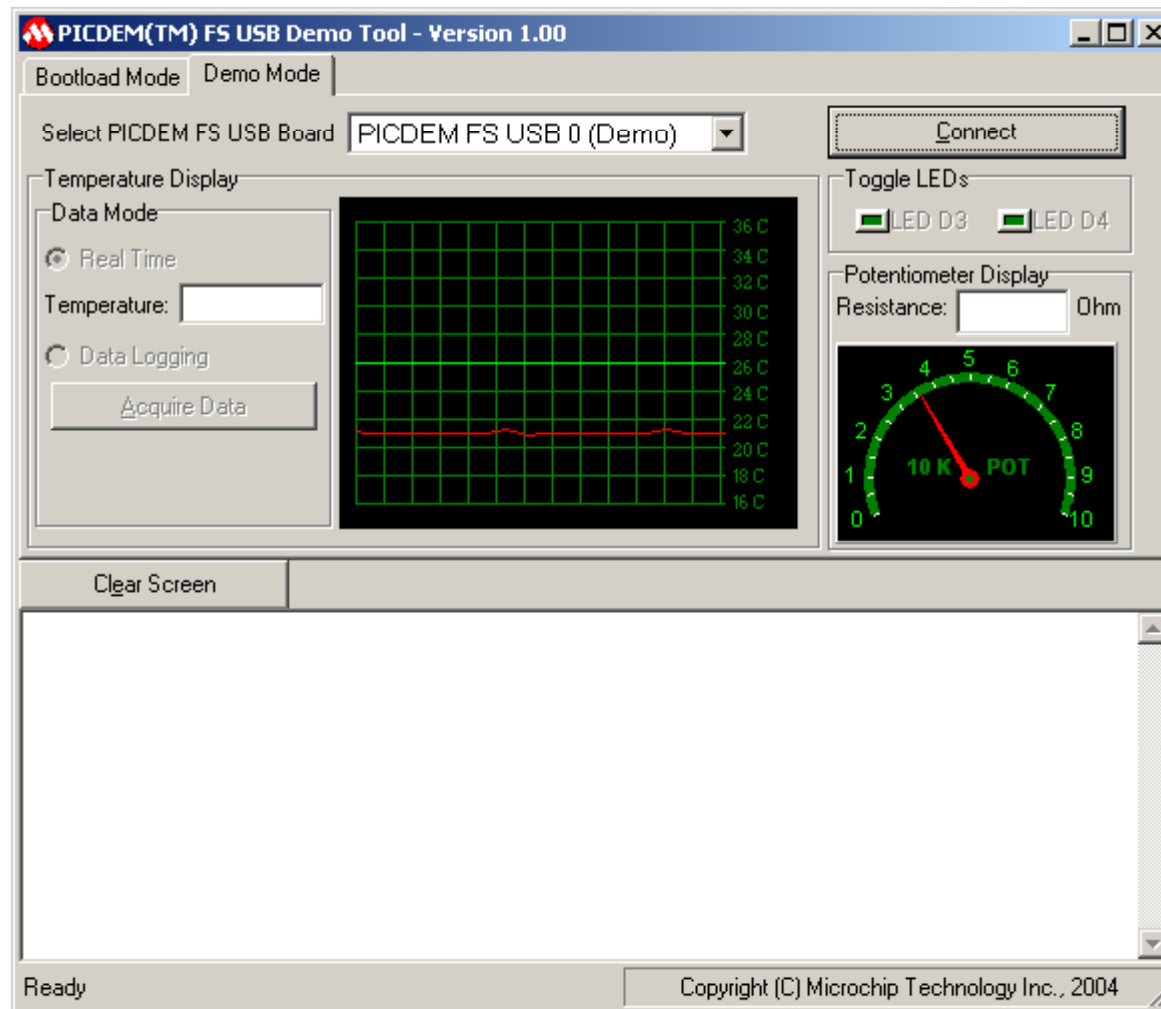


Solution From Microchip

- **AN956: Migrating Applications to USB from RS232**
 - PIC18 based applications
 - Implementation done on Standard Full Speed Demo Board
 - Communication Device Class (CDC) Implementation
 - C Code easily migrates to other USB PIC MCUs
 - 4KB of firmware to implement USB interface
 - Supports all PIC MCUs with Full Speed USB
 - Includes windows based drivers for the Host PC

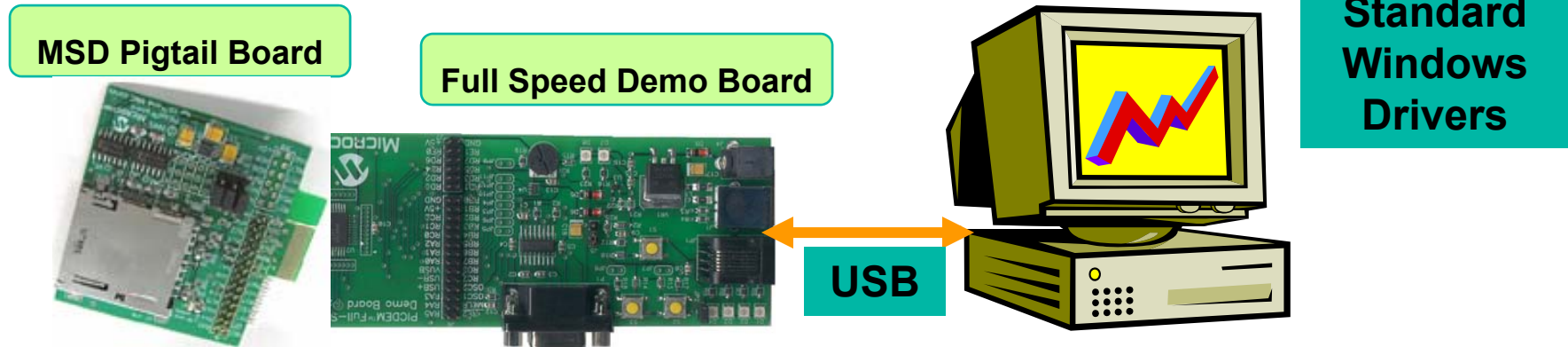


Demo 1 Peripheral Device PIC18



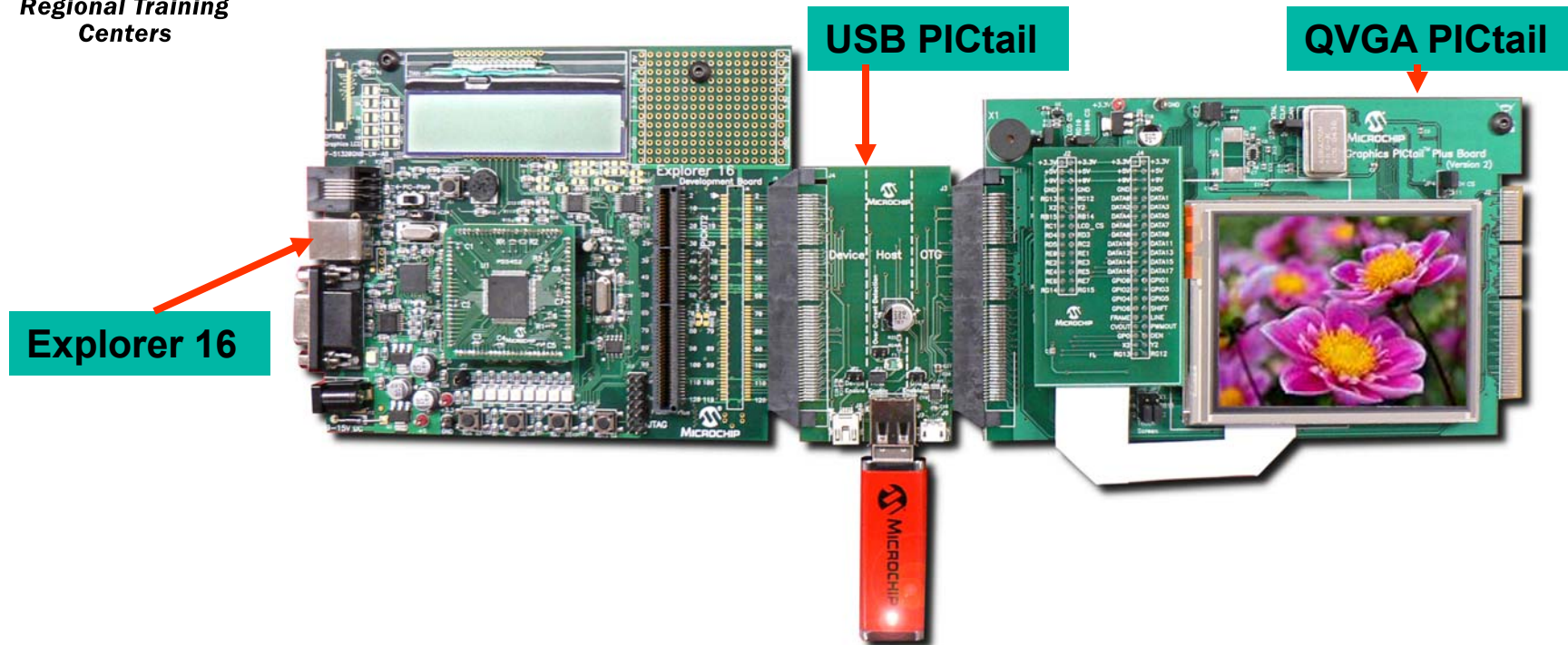
Solution From Microchip

AN1003: USB Mass Storage



- **USB Interface to a MSD Class device**
- **Implemented using Secure Digital/Multimedia Card**
- **PIC MCU implementation on standard Full Speed Demo board and MSD Pigtail Board**
- **Uses standard Windows based drivers: usbstor.sys**

Solution From Microchip



- **Embedded Host Using the PIC24FJ256GB110 16-bit PIC MCU**
- **Host Reads jpeg or bit mapped pictures from USB Flash Drive**
- **Host Displays pictures on QVGA color display**

VID & PID

- **Vendor ID (VID)** *16-bit number*
 - Required to market your product
 - <http://www.usb.org/developers/vendor/>
 - USD \$2,000
 - Technical & Legal trouble if not using an approved VID
- **Product ID (PID)** *16-bit number*
 - Microchip's Sub-licensing Program
- **Every product line is required to have a unique combination of VID and PID**

USB Compliance

- **Compliance Testing**
 - Must pass to use USB logo
 - USD ~\$1,500
- **Ch9 and other USB Firmware**
 - USB Protocol Analyzer
 - “USBCV” USB Command Verifier
 - Tests device for conformance to Ch9, Hub, HID , MSD and Video Class specifications
 - www.usb.org/developers/tools/
- **Electrical Signal Quality**
- **Power Management**



Compliance Testing

- **For USB Compliance: Independent Test Labs**
- **For Device ‘Sanity Check’: USB “Plugfest”**
- **For USB Compliance Testing:**
 - Must submit a compliance checklist
 - www.usb.org/developers/compliance/peripheral_low/
 - Download “Peripheral Checklist”
 - TID: Test ID
 - Use certified USB receptacle and cable for testing
 - Know the TID of your components
 - All USB PIC® MCUs have a TID number. Get it at www.microchip.com/usb
- **Probably a good idea to take a look at the checklist even before starting your design!**

Agenda

- **The Connectivity Challenge for Embedded Application Designers**
- **USB Overview**
- **Principles of USB Operation**
- **PIC® Microcontroller solutions for USB from MCHP**
- **Using the Microchip USB Framework**
- **USB PC Drivers**
- **Microchip USB Development Tools**
- **Some USB Solutions**
- **Summary & Resources**

Summary

- **Today, we**
 - Covered the basics of USB, and saw how it could be used in an embedded application
 - Became familiar with Microchip's MCUs, development boards, and USB software framework.
 - Reviewed Microchip's custom class PC Host USB drivers and device framework

Summary

- **Microchip Offers a Complete Range of USB Solutions:**
 - **Wide Range of PIC MCU**
 - **with built-in USB and USB OTG Transceivers**
 - **8-bit, 16-bit and 32-bit MCU solutions**
 - **USB Device solutions for HID, CDC, MSD and Custom Class**
 - **Embedded Host and OTG solutions for MSD, HID and Generic Class**
 - **PC side Drivers for Windows interface via USB**
 - **Standard Demo boards for quick prototype development**
 - **Application Notes for Step-by-Step guidance**

Resources

- **Microchip USB Design Center: Firmware, Custom Driver, App. Notes, other USB specific resources**
 - <http://www.microchip.com/usb>
- **USB 2.0 Specifications and Device Class Specifications**
 - <http://www.usb.org>
- **Developers Discussion Forum**
 - <http://www.usb.org/phpbb/>
 - <http://forum.microchip.com/tt.aspx?forumid=102>
- **“USB Complete, 3/E”
& “Serial Port Complete, 2/E” by Jan Axelson**

Thank You!

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, KeeLoq logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICTail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.