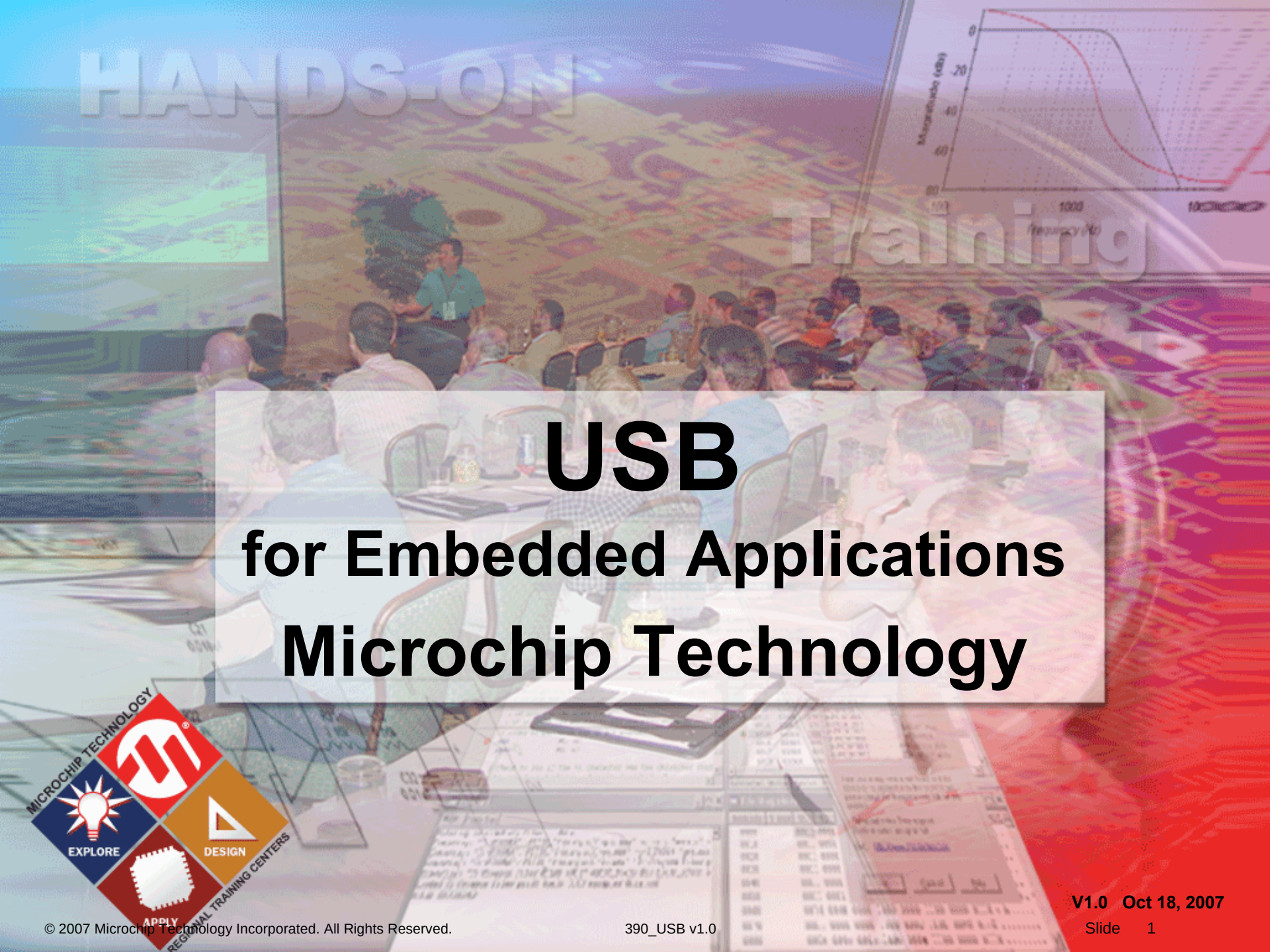
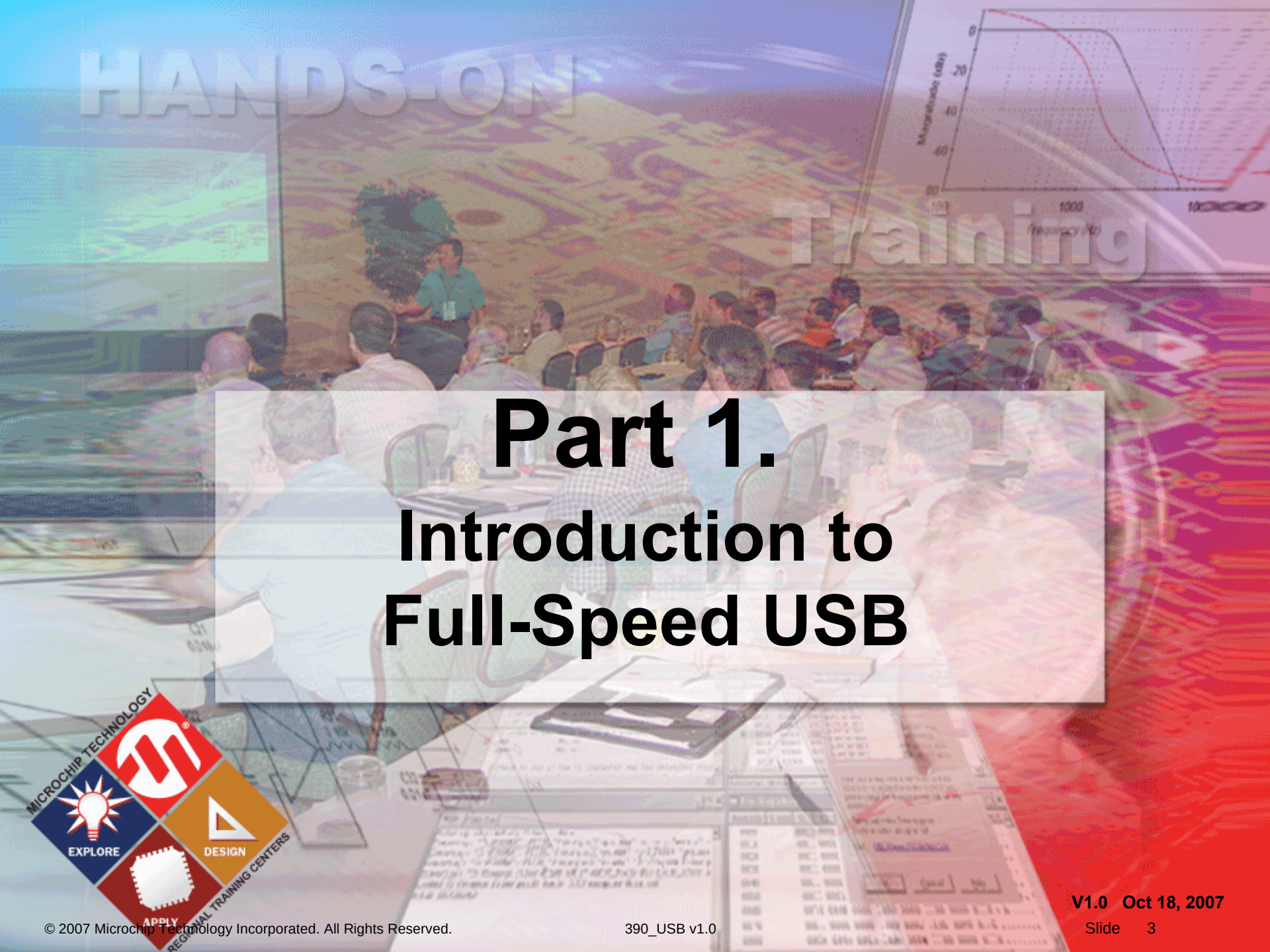


USB
for Embedded Applications
Microchip Technology

Objective

- Understand the basics of USB, and how it can be used in an embedded application
- Be familiar with Microchip's MCUs, development boards, and USB software framework.
- Be able to create a simple PC application that exchanges data with a USB device



Part 1.
Introduction to Full-Speed USB

Objectives

- Understand how USB can be useful in an embedded system
- Learn about fundamental USB architecture, protocol and programmer's model
- Be aware of the factors important in designing a USB application
- Identify key USB capabilities in PIC18 USB MCUs

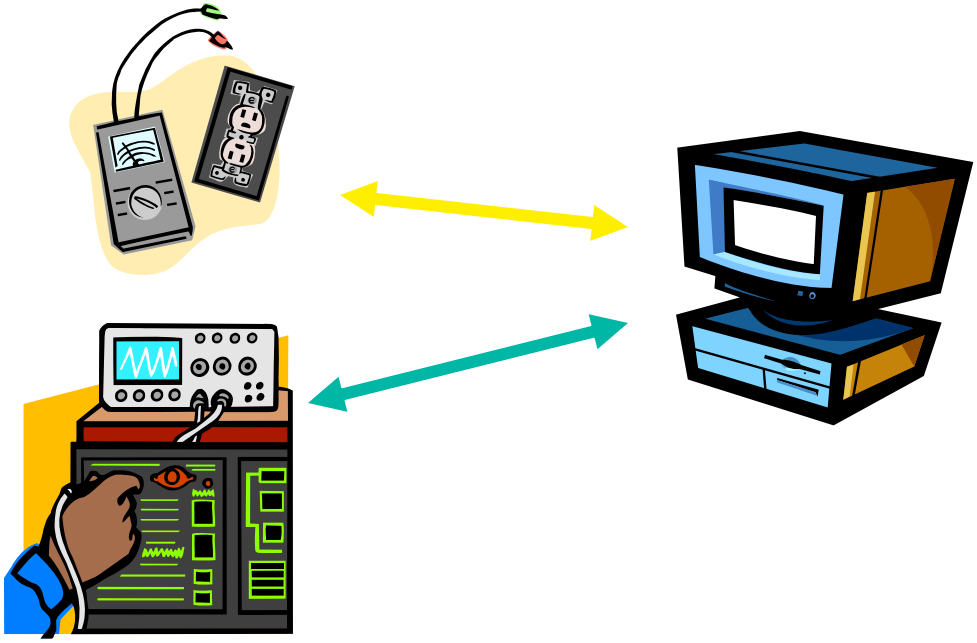
Agenda

- **Brief history of USB & USB-IF**
- **USB Fundamentals - The serious & important stuff**
  - **Basics/Speeds**
  - **Architecture/Programmer's Model**
  - **Physical Connection**
  - USB Transactions
  - USB Transfers
  - Device Classes
  - Enumeration
  - Descriptors
  - Power Planning
  - VID/PID & USB Compliance
- **PIC18F USB Microcontrollers**
- **Microchip Demo/Development Solutions**

A little history...

- **USB was co-developed by a group of companies….**
  - Compaq
  - Intel
  - Microsoft
  - NEC
  - …who wanted to make it much easier to add/remove peripheral devices from PCs
- **1998 - USB 1.1**
- **2000 - USB 2.0**

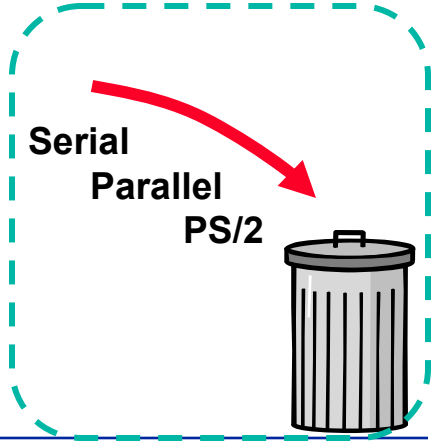
Universal Serial Bus



Extend the functionality of your computer!

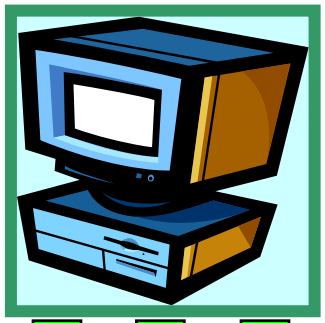
**Data Analysis,
Data Logging,
Firmware Updates,
Diagnostics,
Embedded Applications!**

- Auto detection & configuration (Plug&Play)
- Easy expansion using hubs
- Bus power
- Data CRC protected, bad packets resent.
- Three speeds:
  Low- 1.5, Full- 12, High- 480 Megabits / second

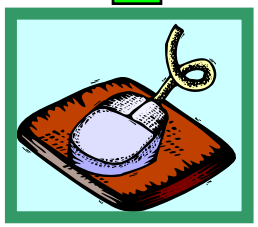


USB Basics

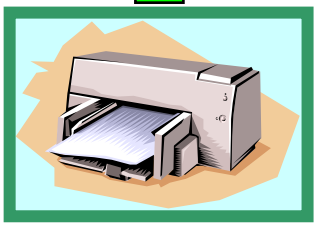
- **USB is a "Single Master + Multiple Slaves" polled bus**



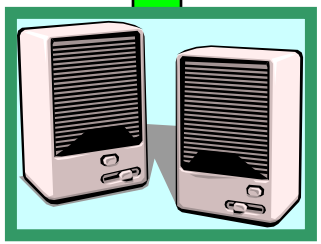
USB Host Controller (Master) and Root Hub



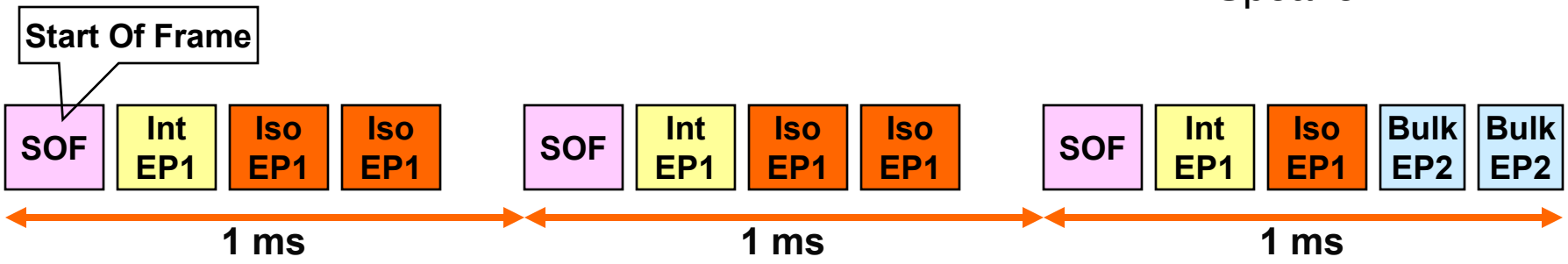
Mouse



Printer



Speaker



Buses & Speeds Comparison

Does not have to support High-Speed to be USB 2.0 Compliant

1394-Fire Wire

Ethernet

WiFi (b/g)

USB 2.0

LS-USB 1.5 Mb/s

FS-USB 12 Mb/s

HS-USB 480 Mb/s

USB 1.1

CAN

Serial Port

Parallel Port

500 Kb/s 1 Mb/s 1.5 Mb/s 12 Mb/s 100 Mb/s 480 Mb/s 1 Gb

Biggest Myth

- **MYTH:** A Low-Speed USB peripheral can transfer application data up to 187.5 KB/s (1.5 Mbps)

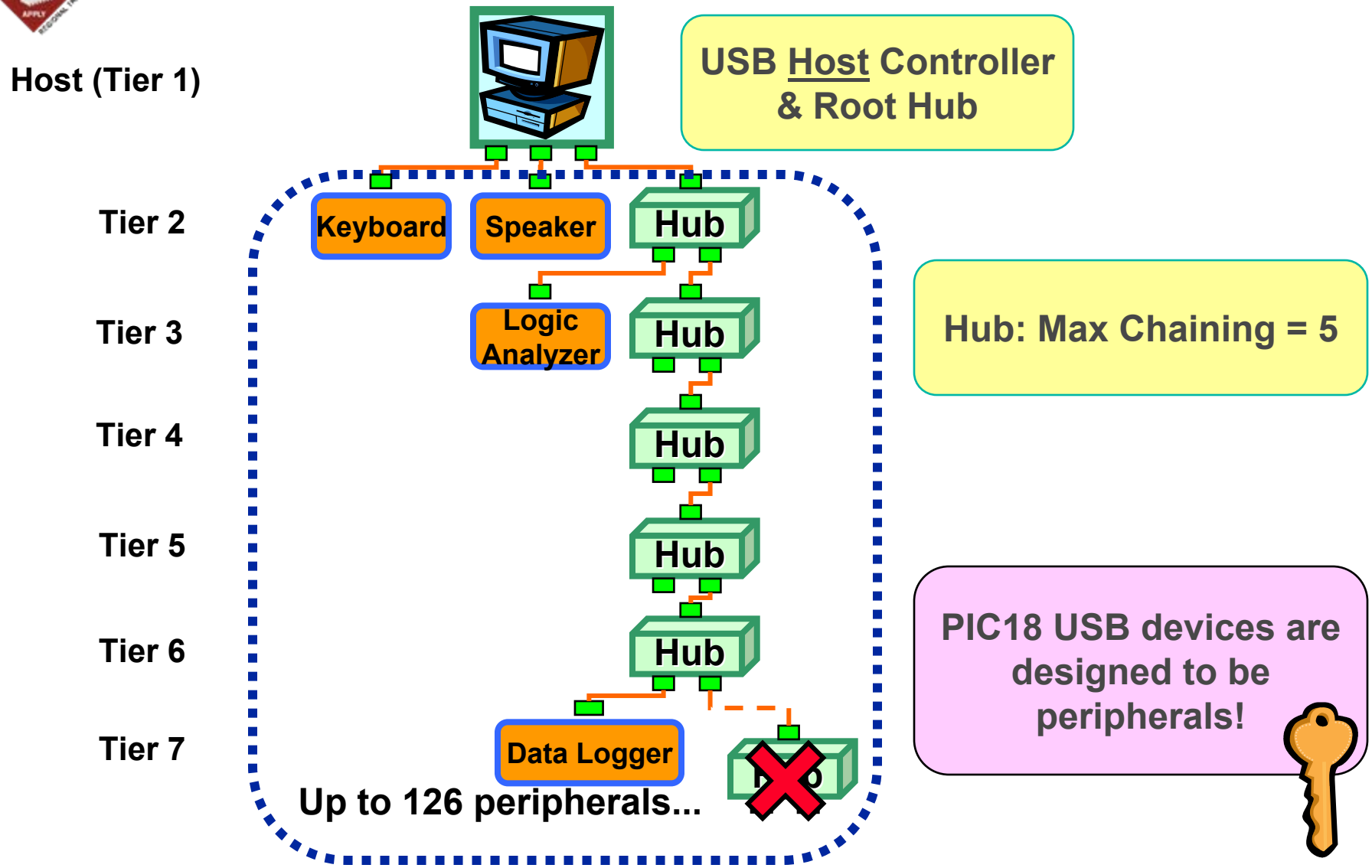
- **FACT:** Impossible, because of a USB specification restriction:
  - 8 byte data transfer every 10 ms
  - = 800 Bytes/second only

Next Biggest Myth

- **MYTH:** A Full-Speed USB peripheral can transfer data up to 1.5 MB/s (12 Mb/s)

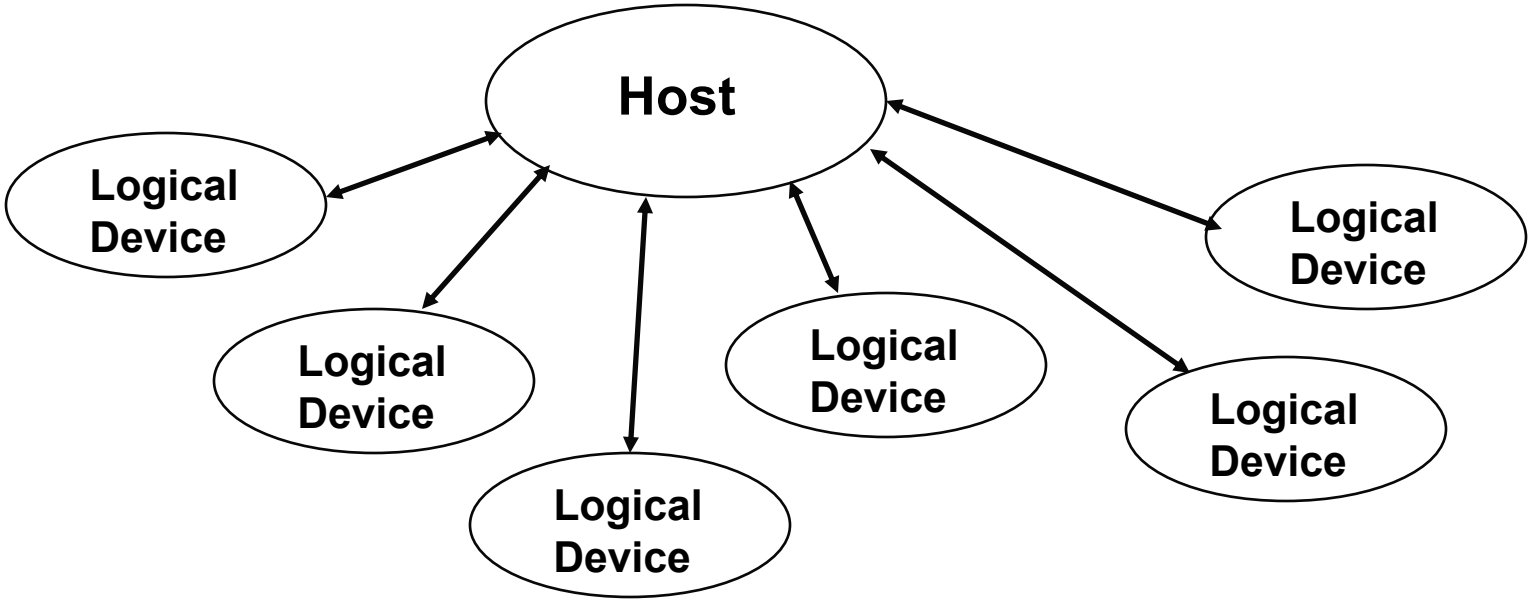
- **FACT:** Impossible, 1.5 MB/s is the total bus bandwidth
  - **Must be shared among peripherals**
  - **Protocol overhead**
  - **Protocol restrictions**
  - **Realistic raw data throughput to a single peripheral is ~1.0 MB/s**
  - **Only 64KB/s in some cases**

Physical Bus Topology



Logical Bus Topology

- Not a tiered-star!
- Host software communicates to each “logical” device as if it were directly connected to the root hub

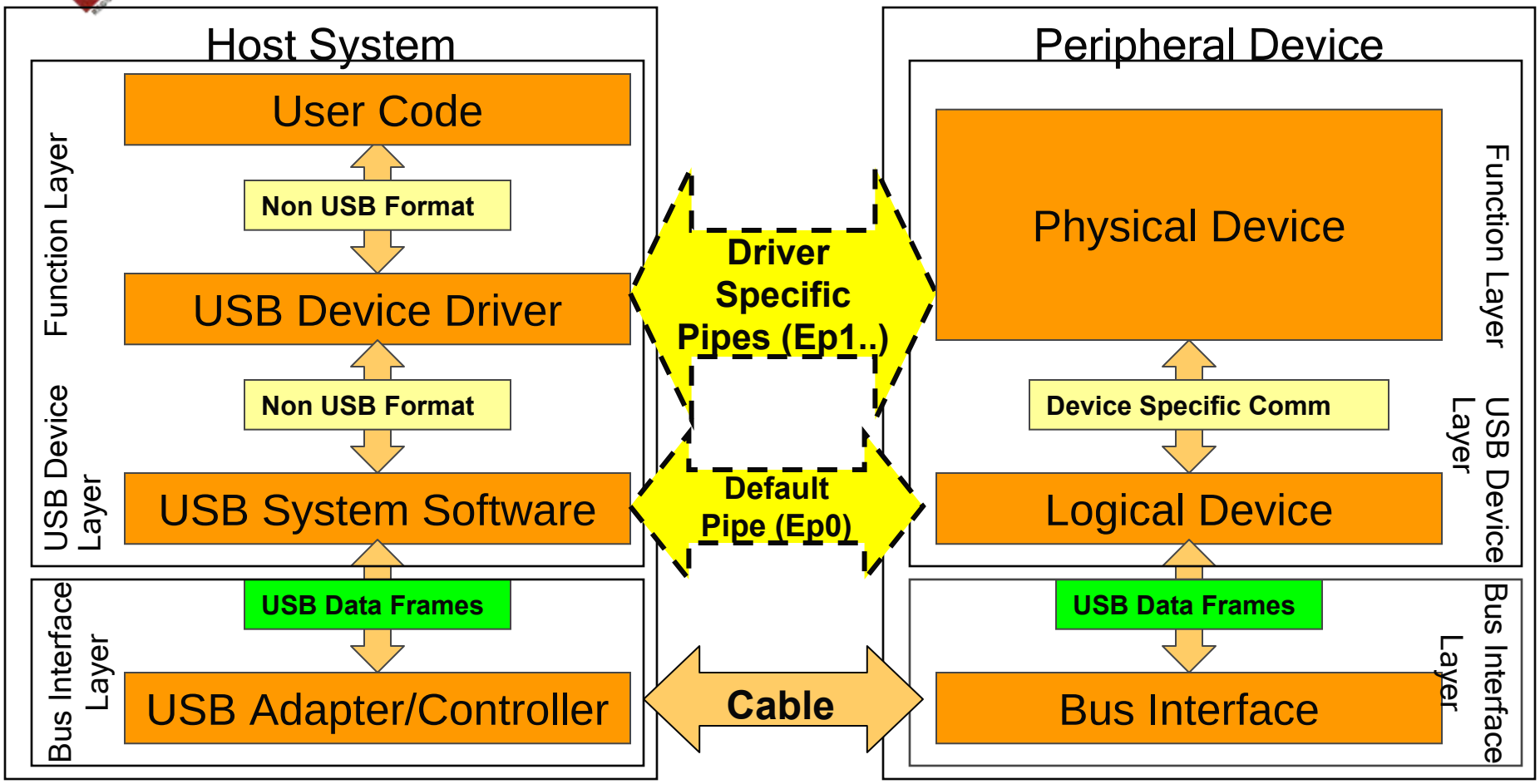


Accessing PC Peripherals

- **Old Way**
- **PC Peripherals:**
  - Memory-mapped into the x86 I/O address space
  - Assigned a specific IRQ line
  - Assigned a specific DMA channel
- **Accessed directly (ISA, PCI, PCMCIA)**

- **USB Way**
- **PC Peripherals:**
  - Mapped into a virtual 127 device address space
  - Does not use any PC I/O, IRQ or DMA resources
- **Accessed indirectly using the programming interface provided by the device driver**

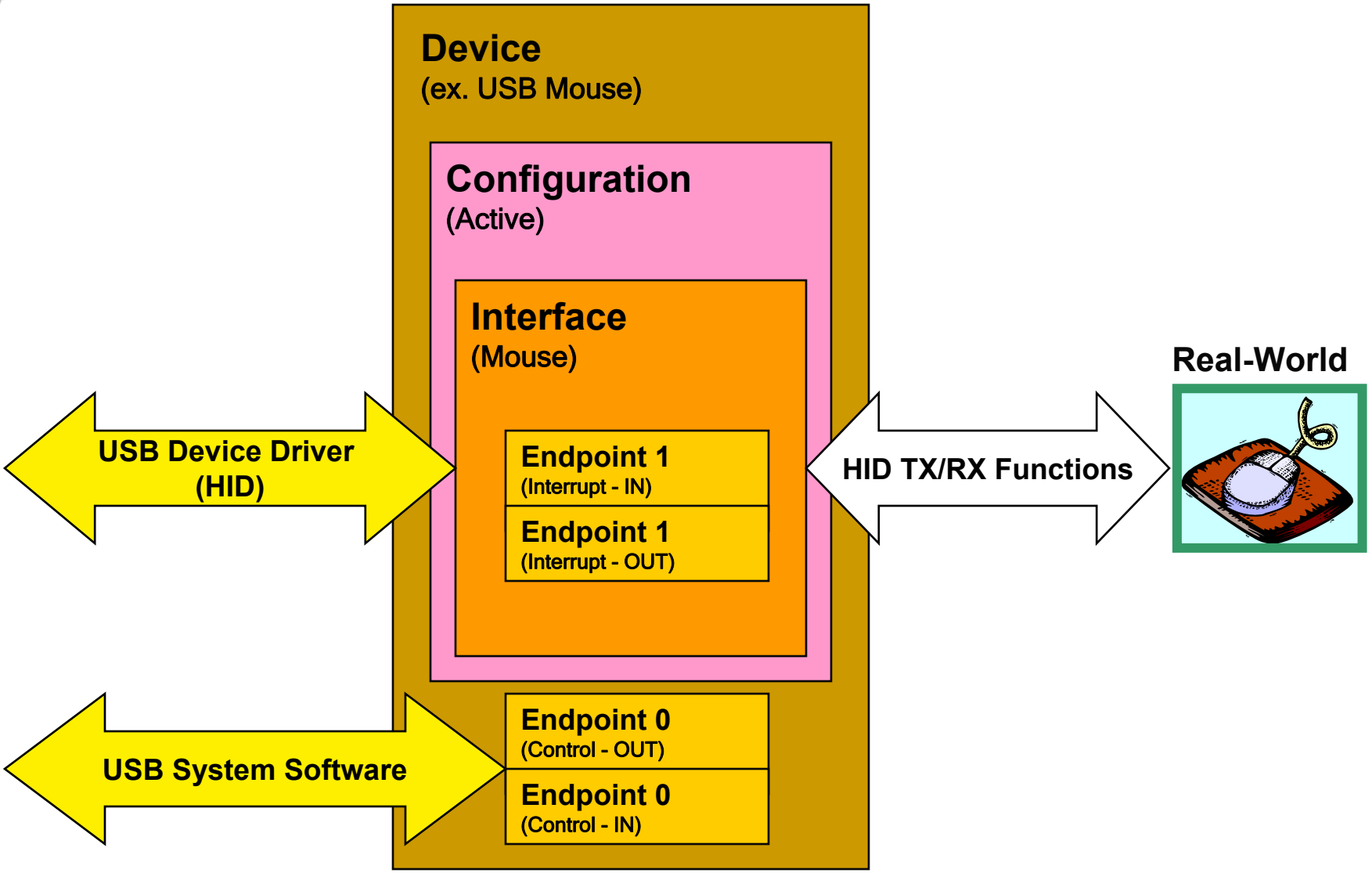
USB Device Framework
- Software View of Hardware -



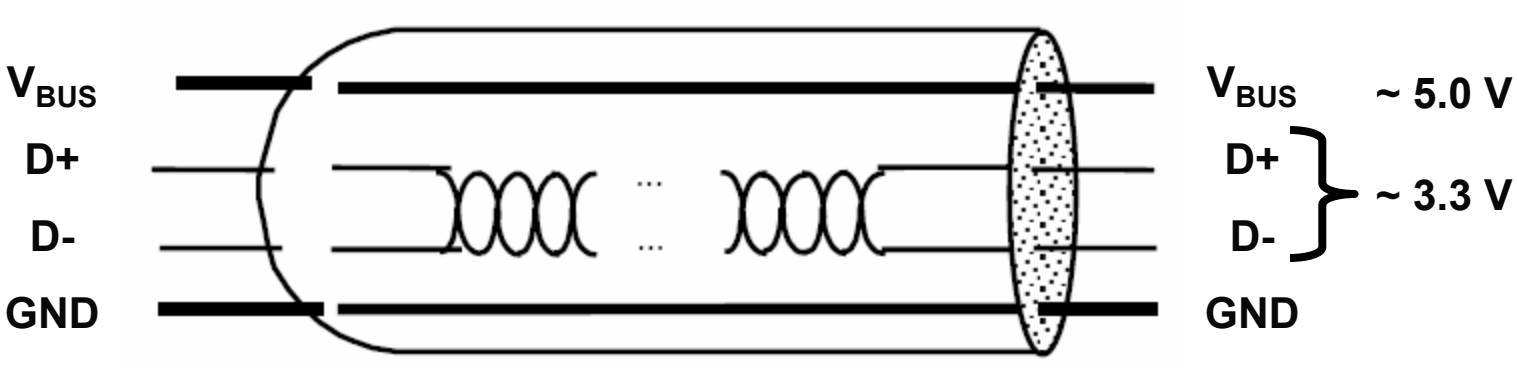
Physical Communication Path

Logical Communication Path ("Pipe")

The “Logical” USB Device



Physical Interface



- **Half Duplex with NRZI Data Encoding**
- **Bus Power to each device:**
  - 4.40 - 5.25 V
  - Guaranteed 100 mA
  - 500 mA maximum through negotiation

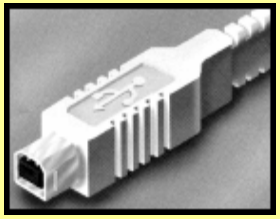
Must use external power if more is required

USB Connectors

“A” Plug

USB Host

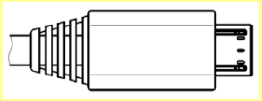
“B”
FS, HS
Peripheral

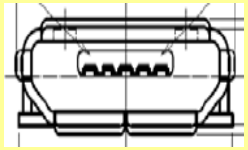


“mini-B”
FS, HS
Peripheral

“micro-B”

Peripheral



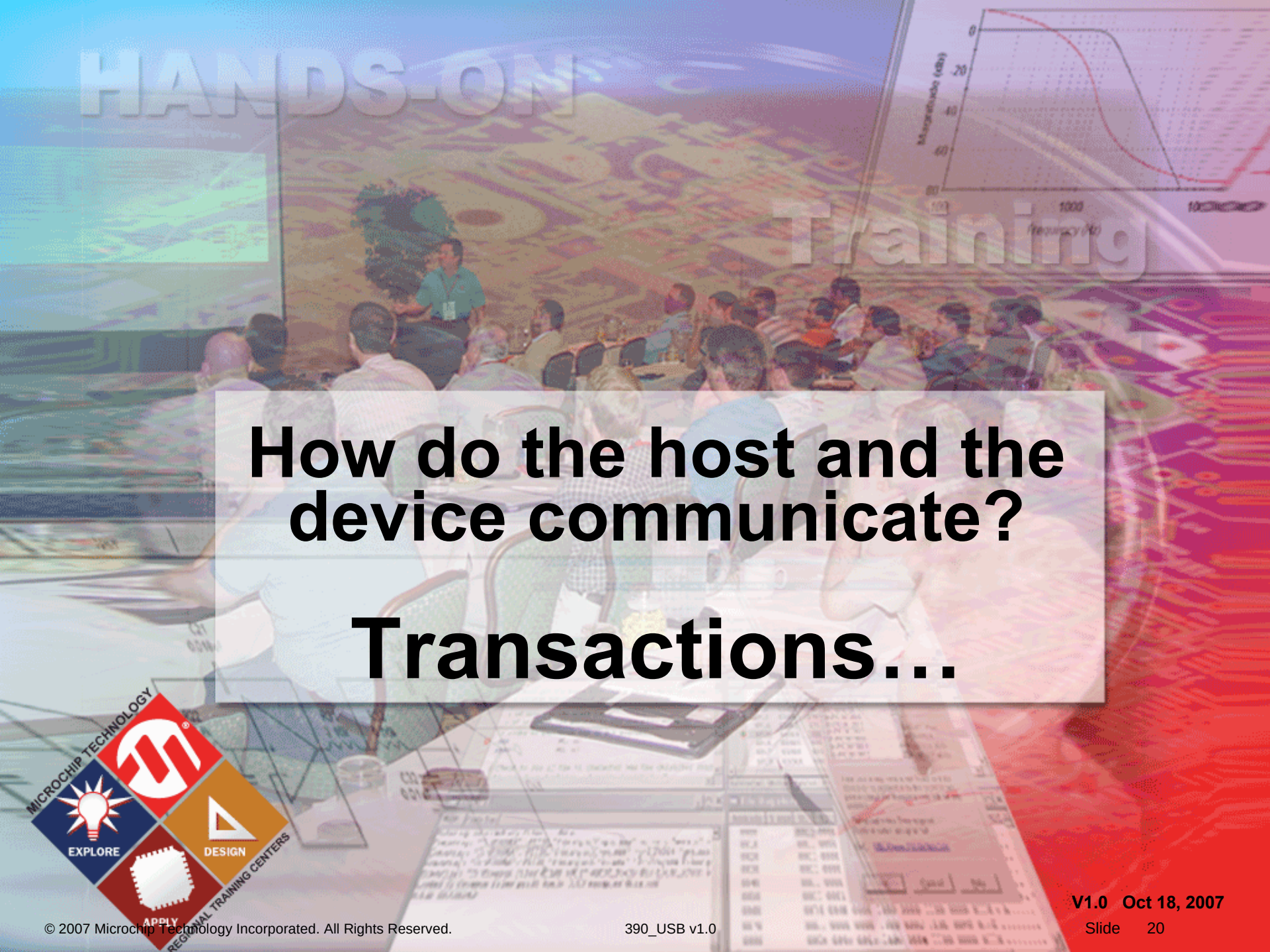


LS, FS, HS
Peripheral

Hardwired
or
Custom
Detachable

Agenda

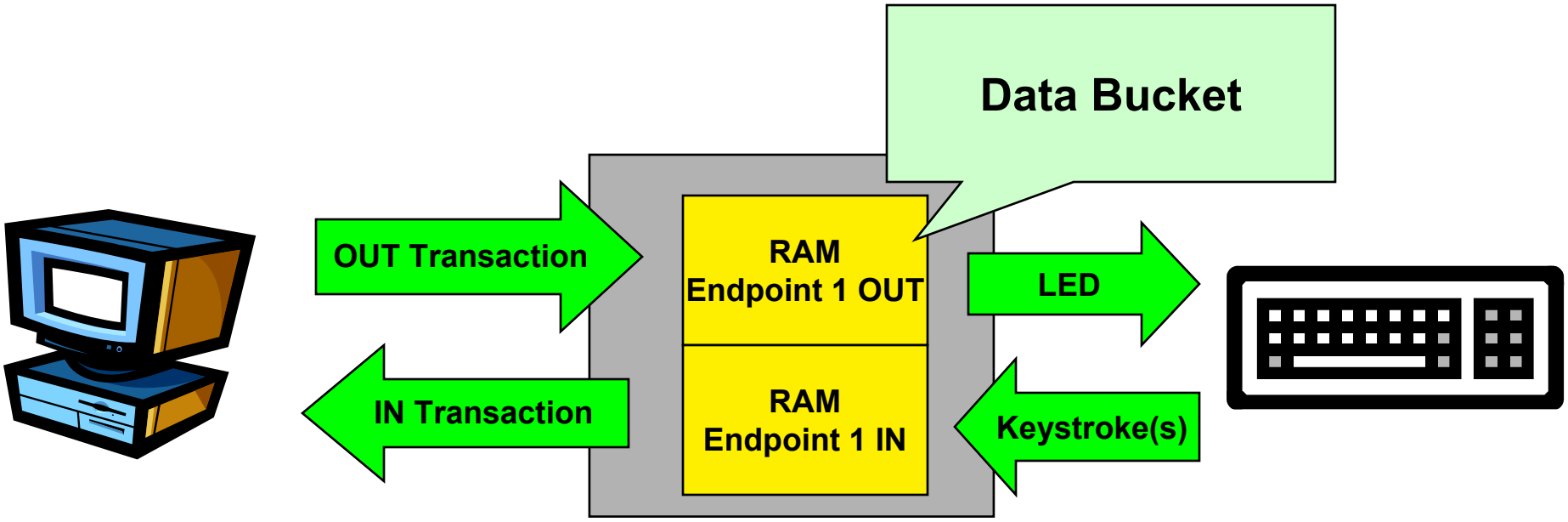
- **Brief history of USB & USB-IF**
- **USB Fundamentals - The serious & important stuff**
  - Basics/Speeds
  - Architecture/Programmer's Model
  - Physical Connection
  - **USB Transactions**
  - **USB Transfers**
  - **Device Classes**
  - Enumeration
  - Descriptors
  - Power Planning
  - VID/PID & USB Compliance
- **PIC18F USB Microcontrollers**
- **Microchip Demo/Development Solutions**



How do the host and the device communicate?

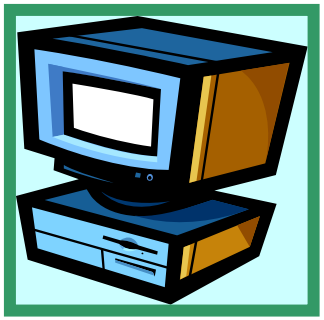
Transactions…

An Endpoint is a Buffer

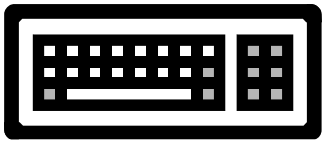


- **Maximum number of endpoints per device specified by USB specification:**
  - 16 OUT endpoints + 16 IN endpoints = 32 endpoints
  - PIC18F87J50, PIC18F4550 supports up to 32 endpoints
- **EP0 = Default Communication Pipe**

USB Transaction



Token Packet

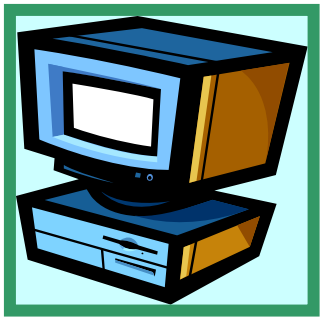


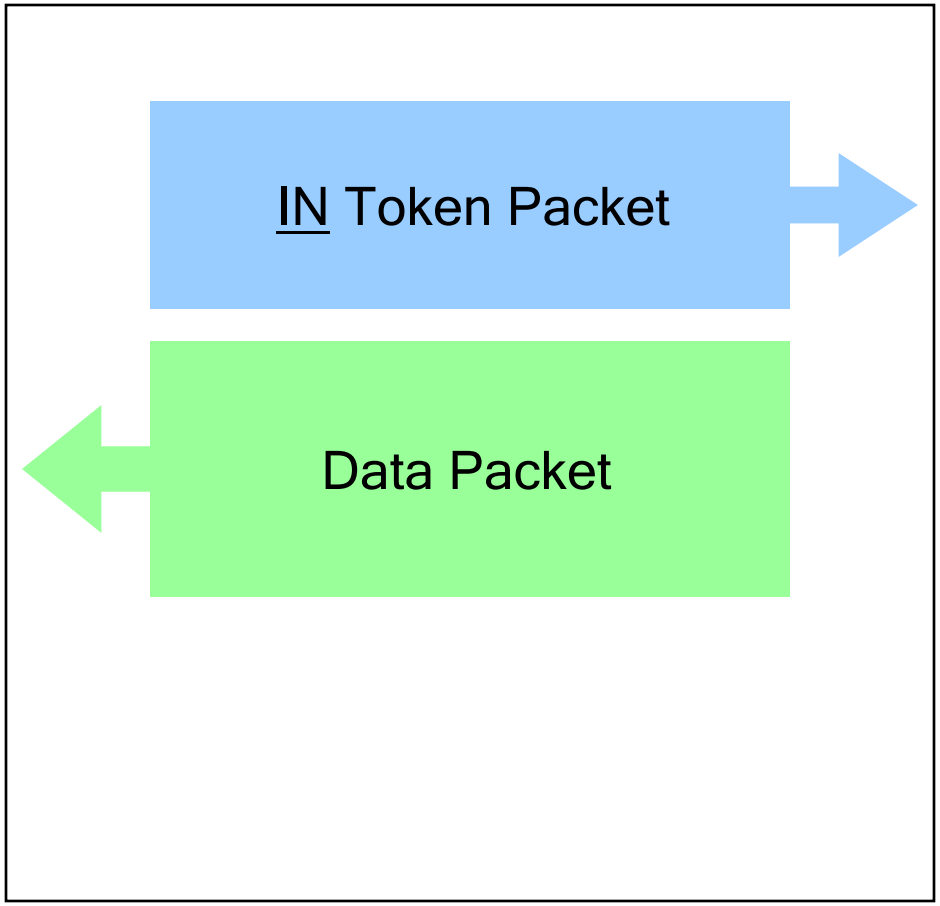
SETUP and OUT token types inform the target device that the host wants to send data.

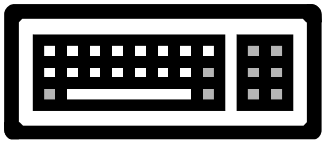
IN token type informs the target device that the host wants to fetch data.

Specifies:
- **Target device address**
- **Endpoint number**
- **Direction of the data transfer**

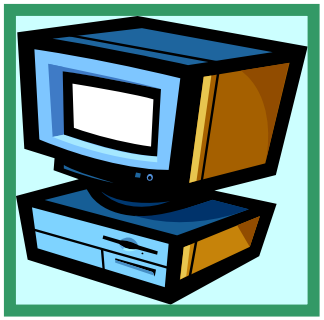
USB Transaction - IN







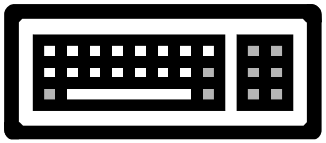
USB Transaction - IN



IN Token Packet

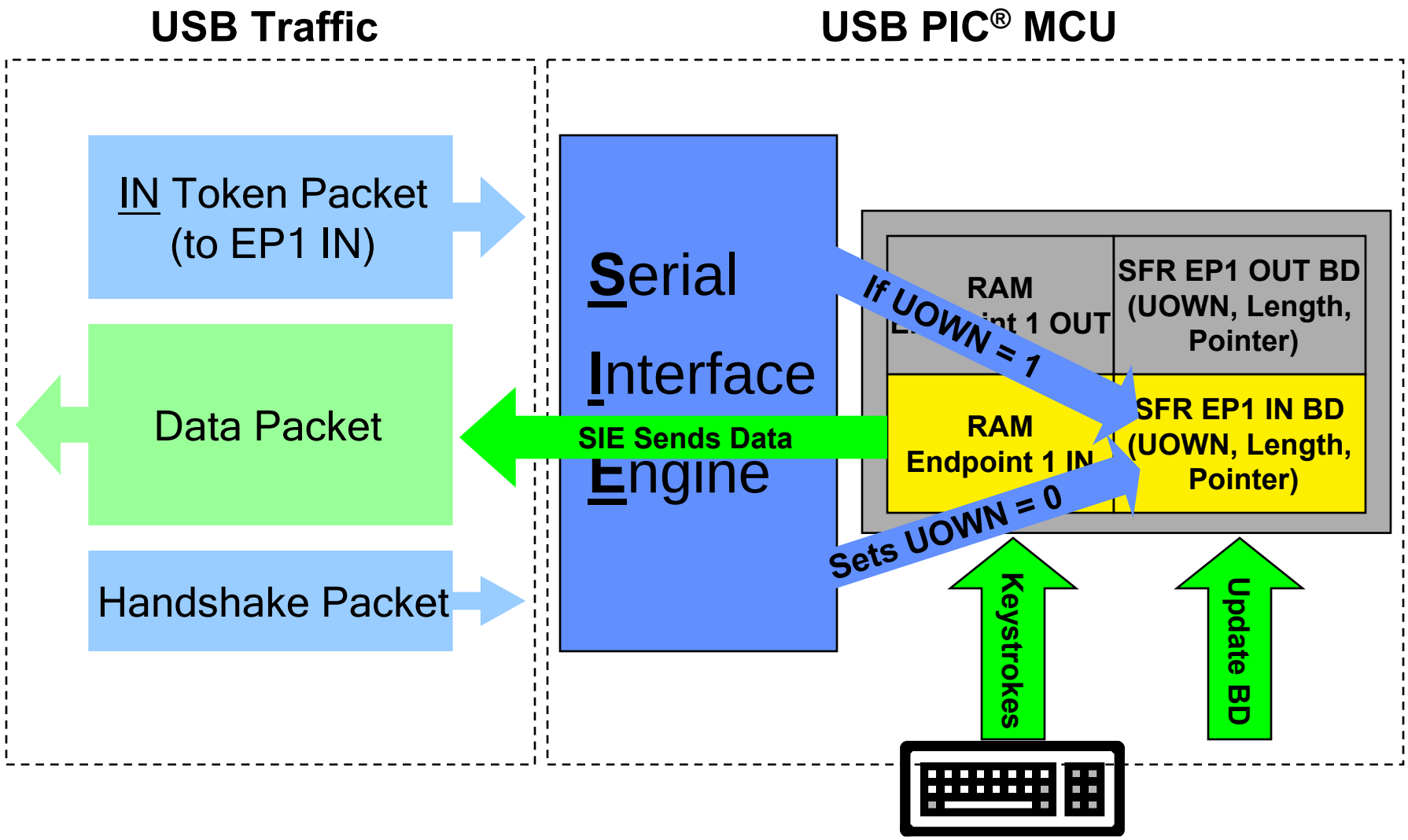
Data Packet

Handshake Packet

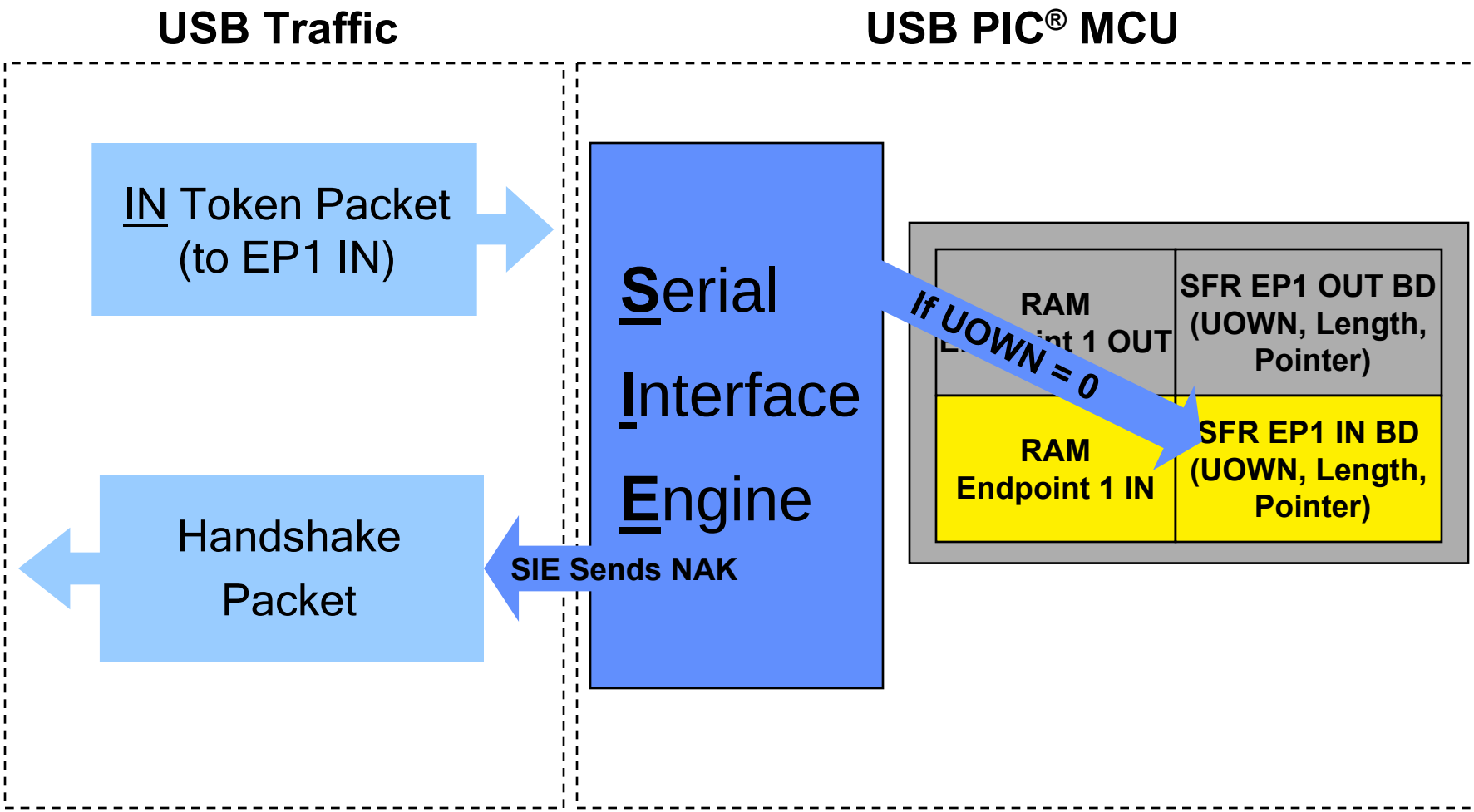


Acknowledge - ACK

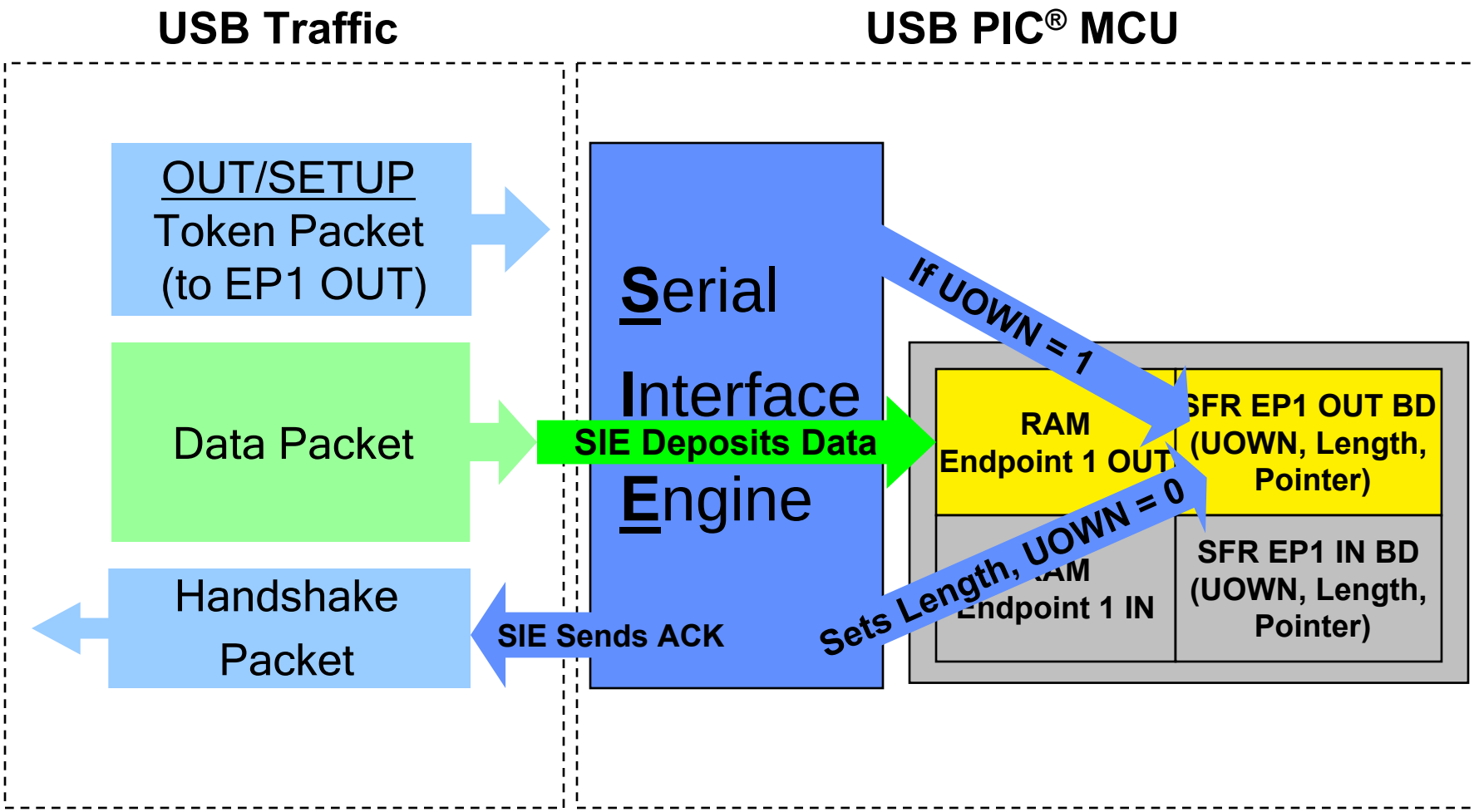
IN Transaction ACK



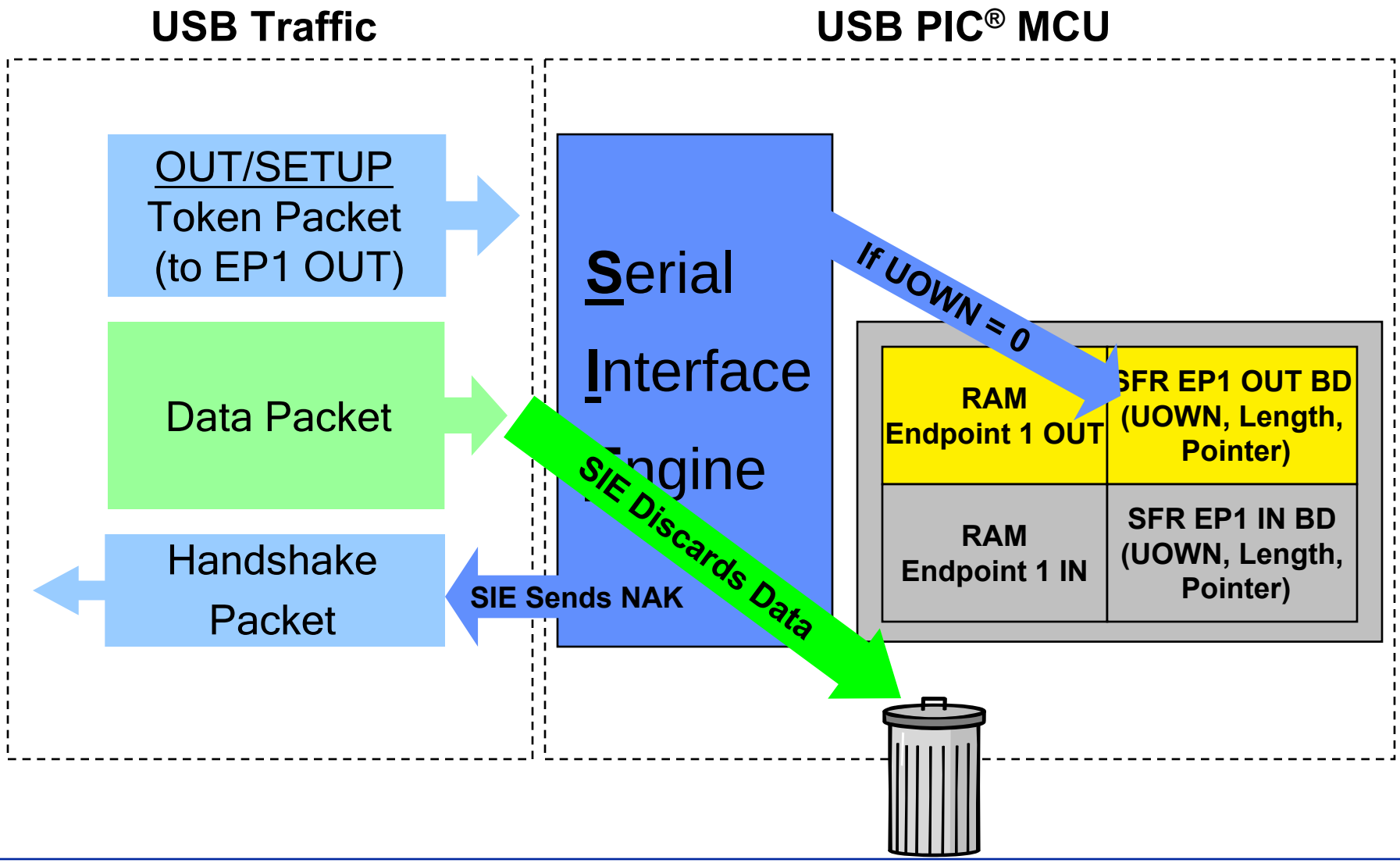
IN Transaction NAK



OUT/SETUP Transaction ACK



OUT/SETUP Transaction NAK

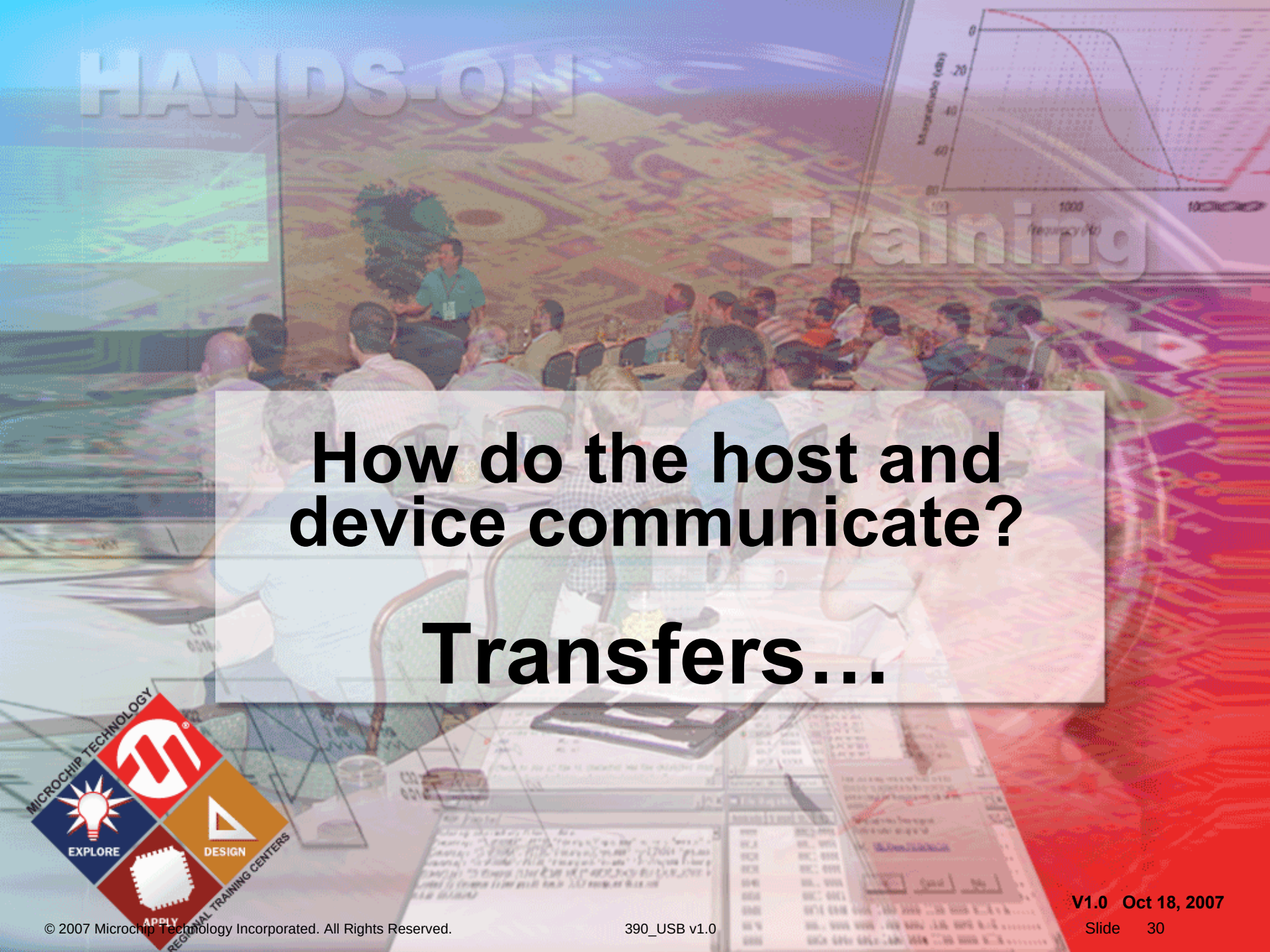


Key: Token Types

SETUP

OUT

IN

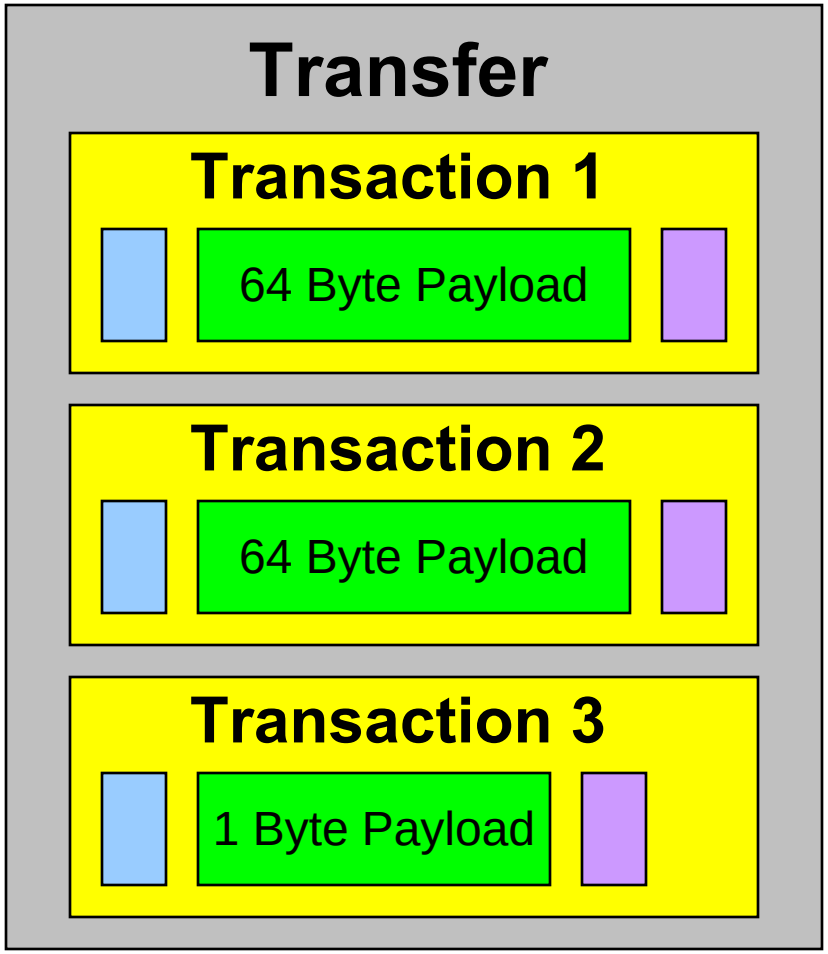


How do the host and device communicate?

Transfers…

Transfers Vs. Transactions

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



Key:

- OUT Token Packet
- Data Packet
- ACK Handshake Packet

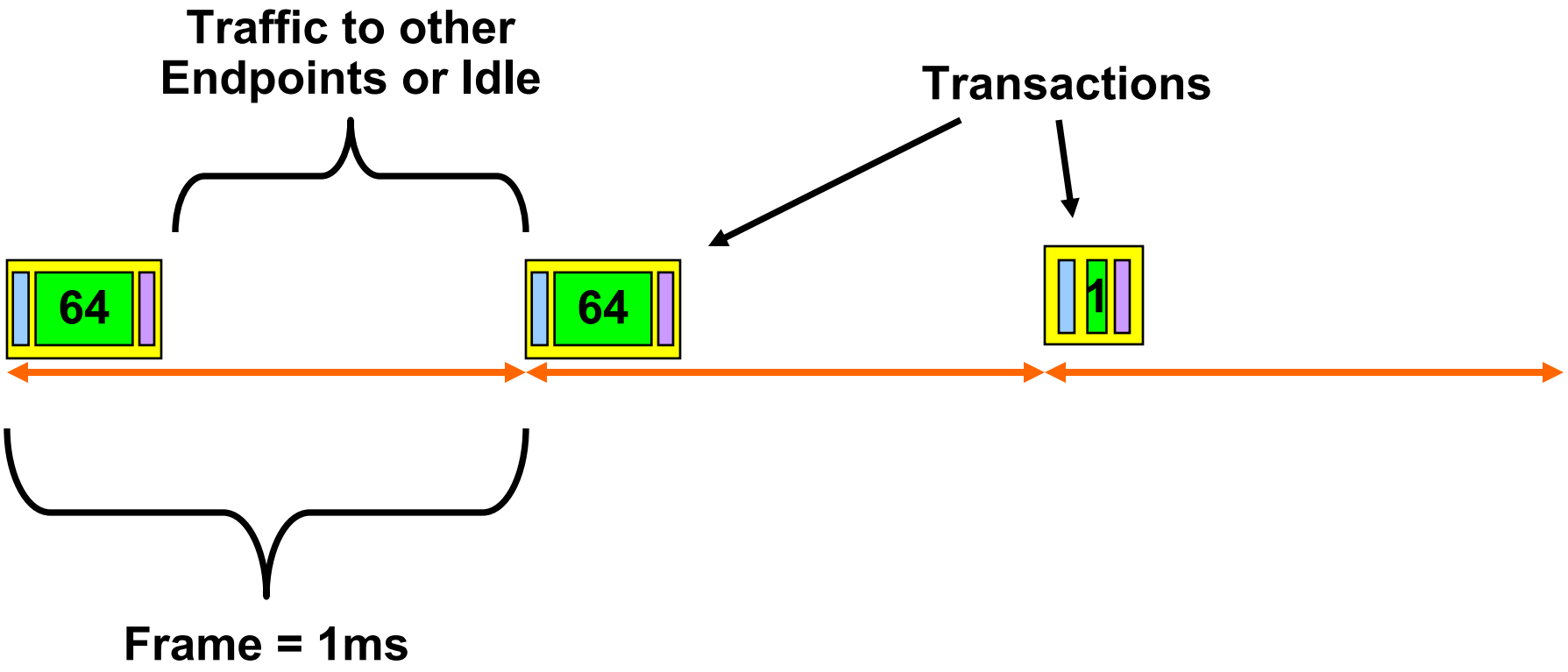
Transfer: Group of related transactions.

Data Transfer Types

Transfer/ Endpoint Type	Polling Interval	Priority	Guarantees
Interrupt	Fixed, Periodic	High	64 Bytes/Period, Data Integrity
Isochronous	Fixed, Periodic	High	1023 Bytes/Period
Bulk	Variable, Uses Free Bandwidth	Low	Data Integrity
Control	Variable	Medium	Some Bandwidth, Data Integrity

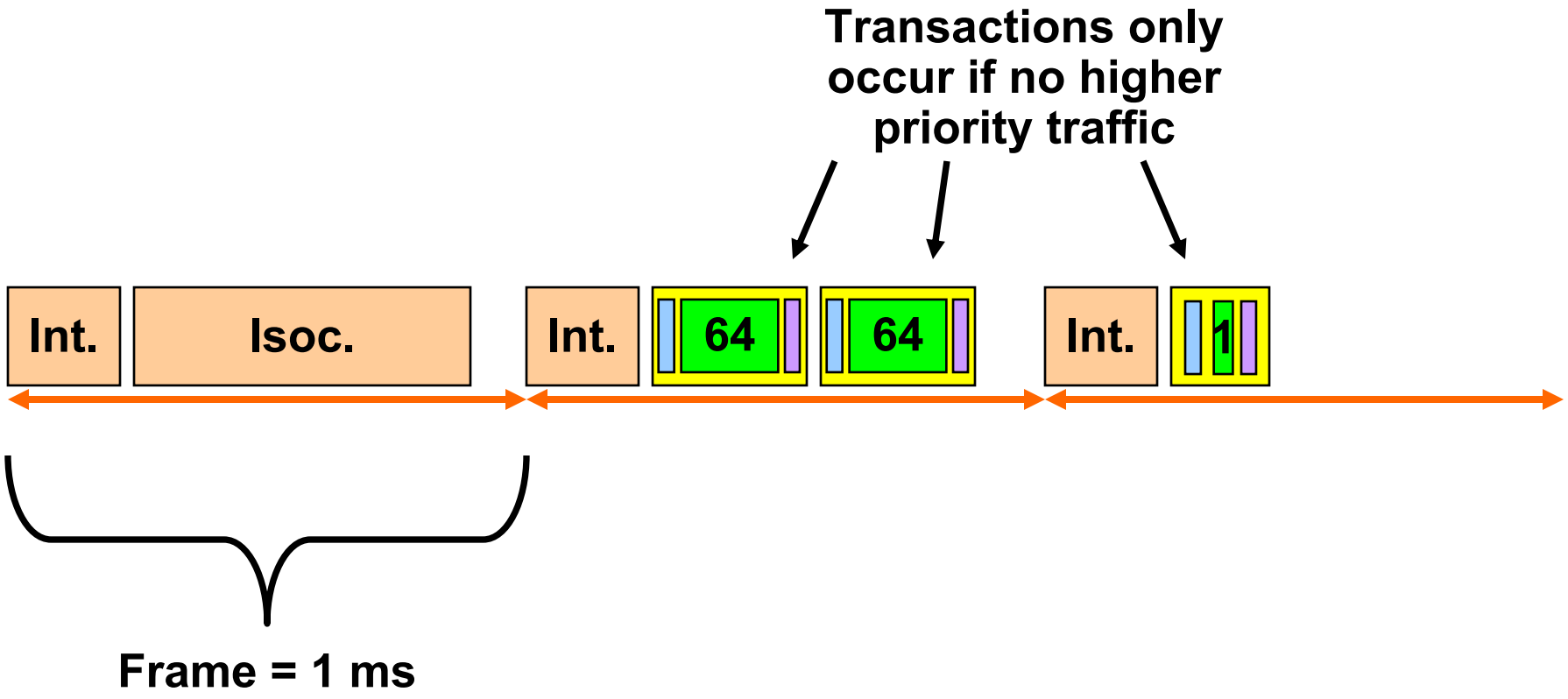
Interrupt Transfer Example

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

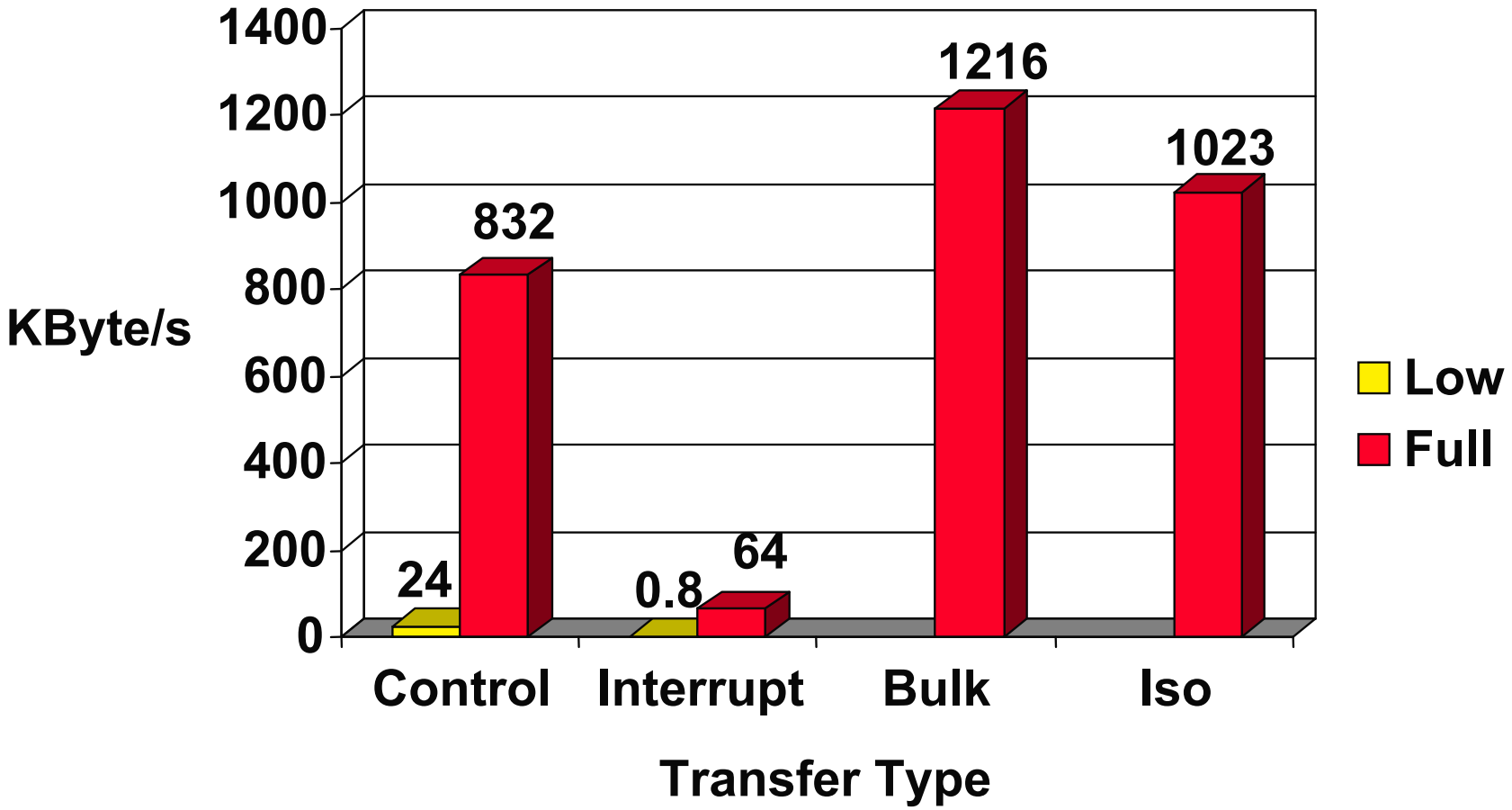


Bulk Transfer Example

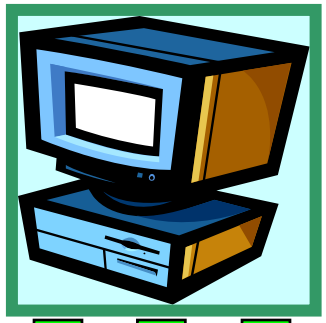
MPUSBWrite(EP7, Pointer, Size = 129, Timeout)

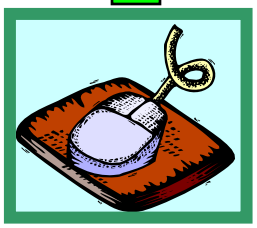


Theoretical Maximum Transfer Rate Per Endpoint



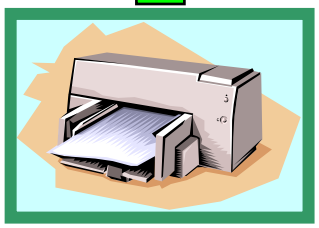
Transfer Types - Examples





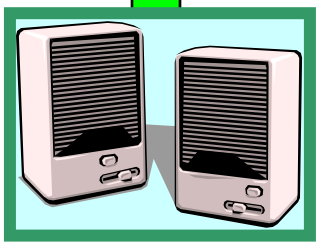
Mouse

Control
+
Interrupt



Printer

Control
+
Bulk



Speaker

Control
+
Isochronous

USB Device Classes

Data Glove

Mouse

PICkit™ 2 Starter Kit

Joystick

Keyboard

Human Interface Device Class (HID)

Floppy Drive

External Hard Drive

Mass Storage Device Class (MSD)

Ethernet Adapter

Modem

Communication Device Class (CDC)

ICD2

Custom Class (Vendor Class)

Many more classes….

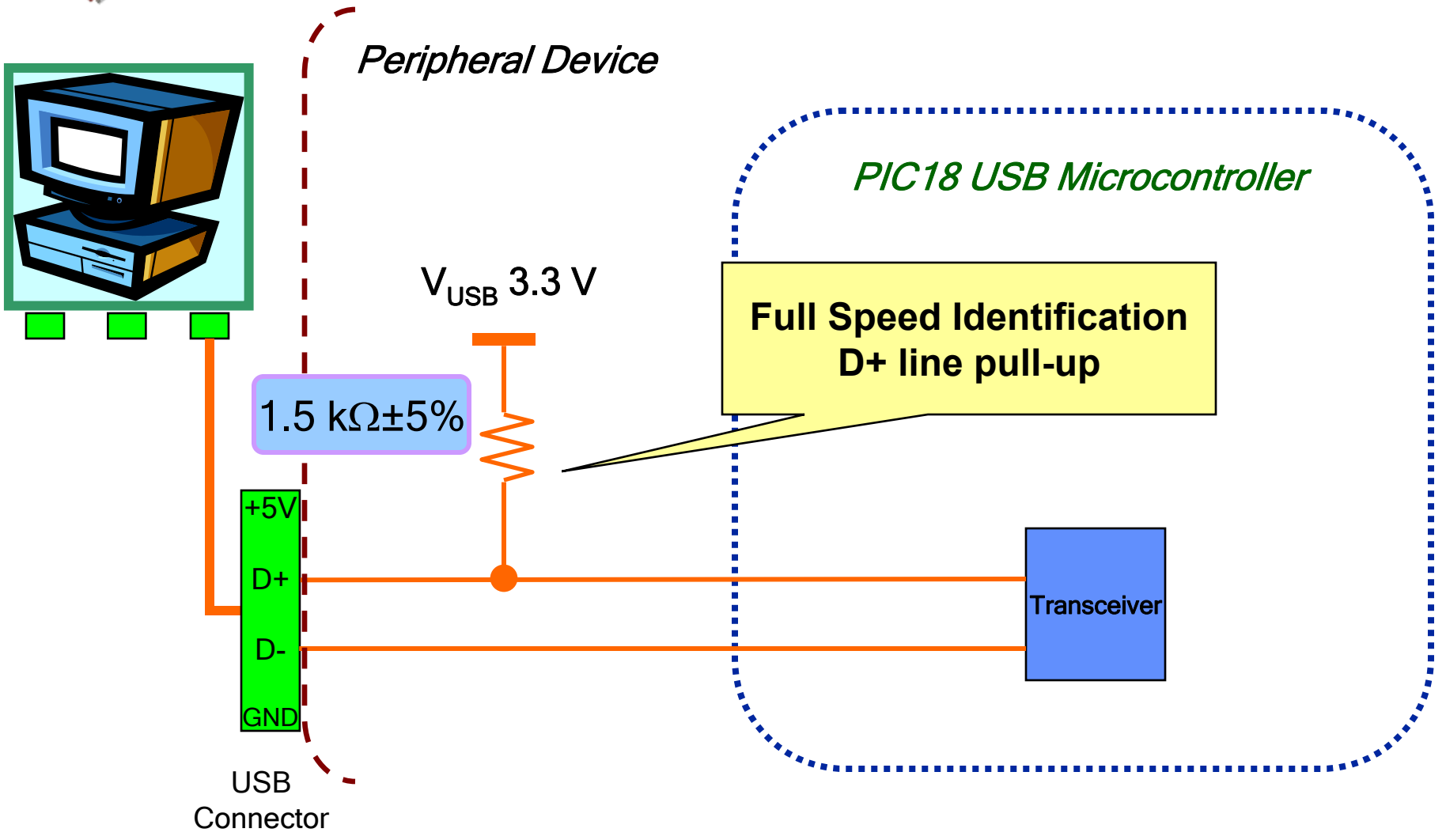
Agenda

- **Brief history of USB & USB-IF**
- **USB Fundamentals - The serious & important stuff**
  - Basics/Speeds
  - Architecture/Programmer's Model
  - Physical Connection
  - USB Transactions
  - USB Transfers
  - Device Classes
  - **Enumeration**
  - **Descriptors**
  - **Power Planning**
  - **VID/PID & USB Compliance**
- **PIC18F USB Microcontrollers**
- **Microchip Demo/Development Solutions**

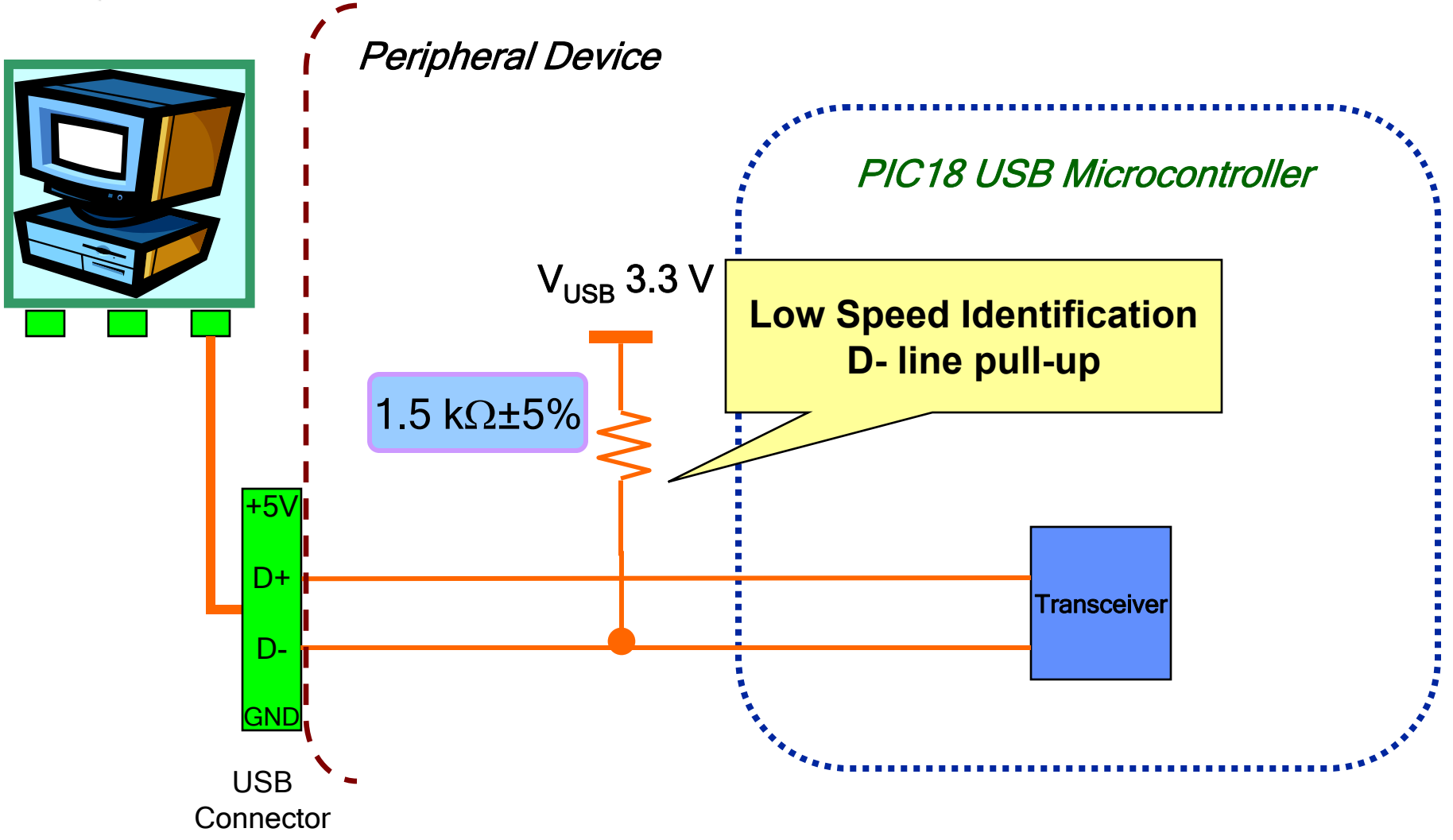
Enumeration and the magic behind "Plug&Play"

Auto-Detection & Auto-Configuration

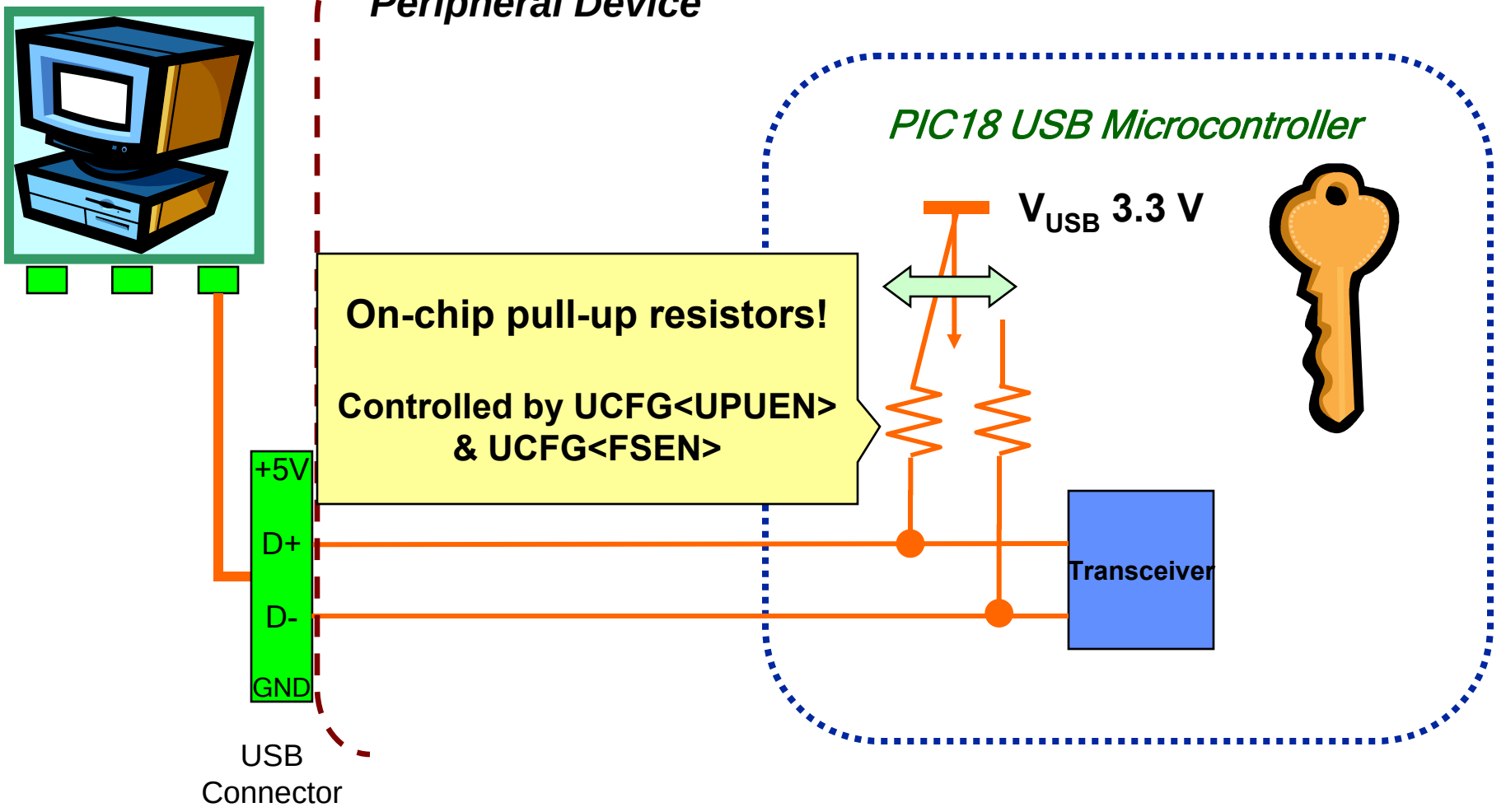
Auto-Detection: Full-Speed



Auto-Detection: Low-Speed

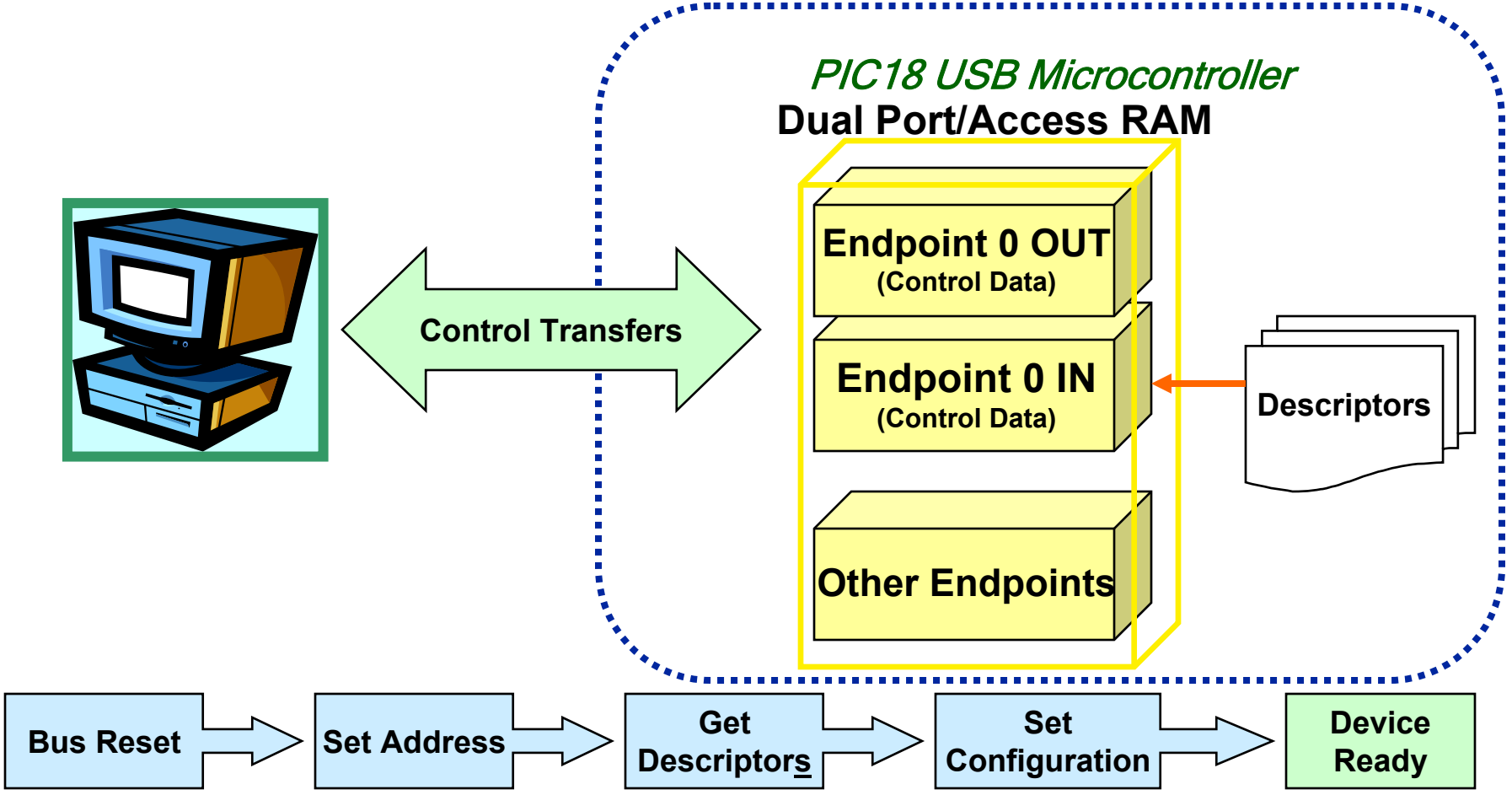


On-chip Pull-up Resistors



Endpoint 0 and Enumeration

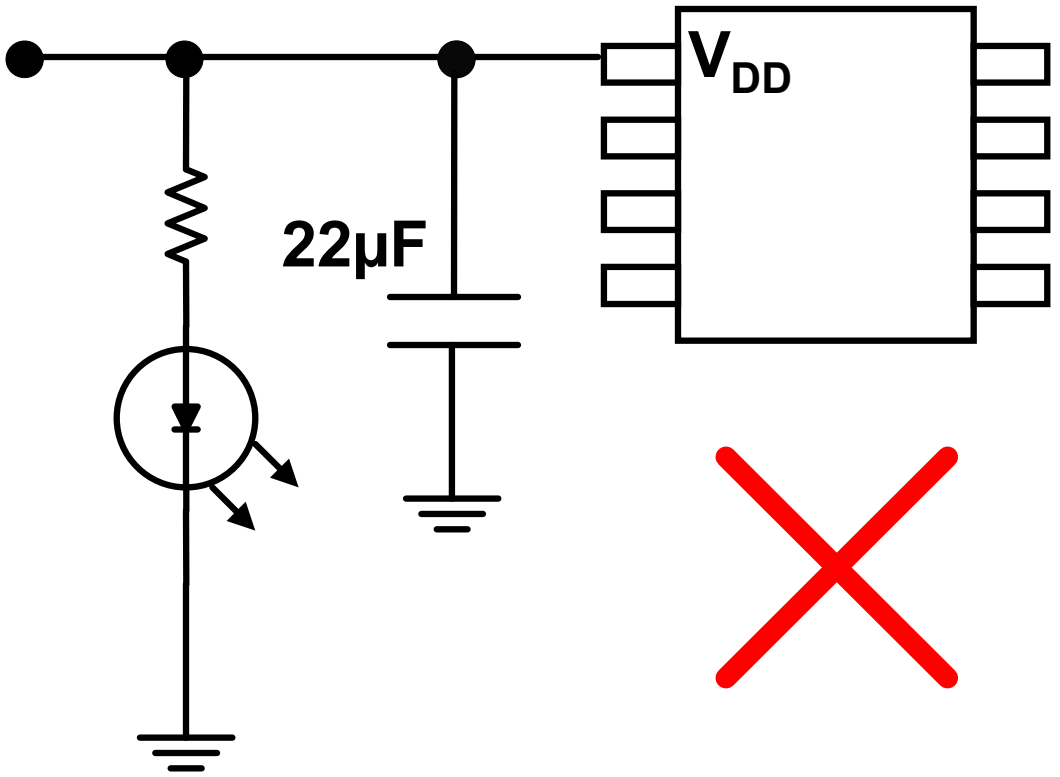
- See Chapter 9 in USB 2.0 Spec for more info.



Power Planning

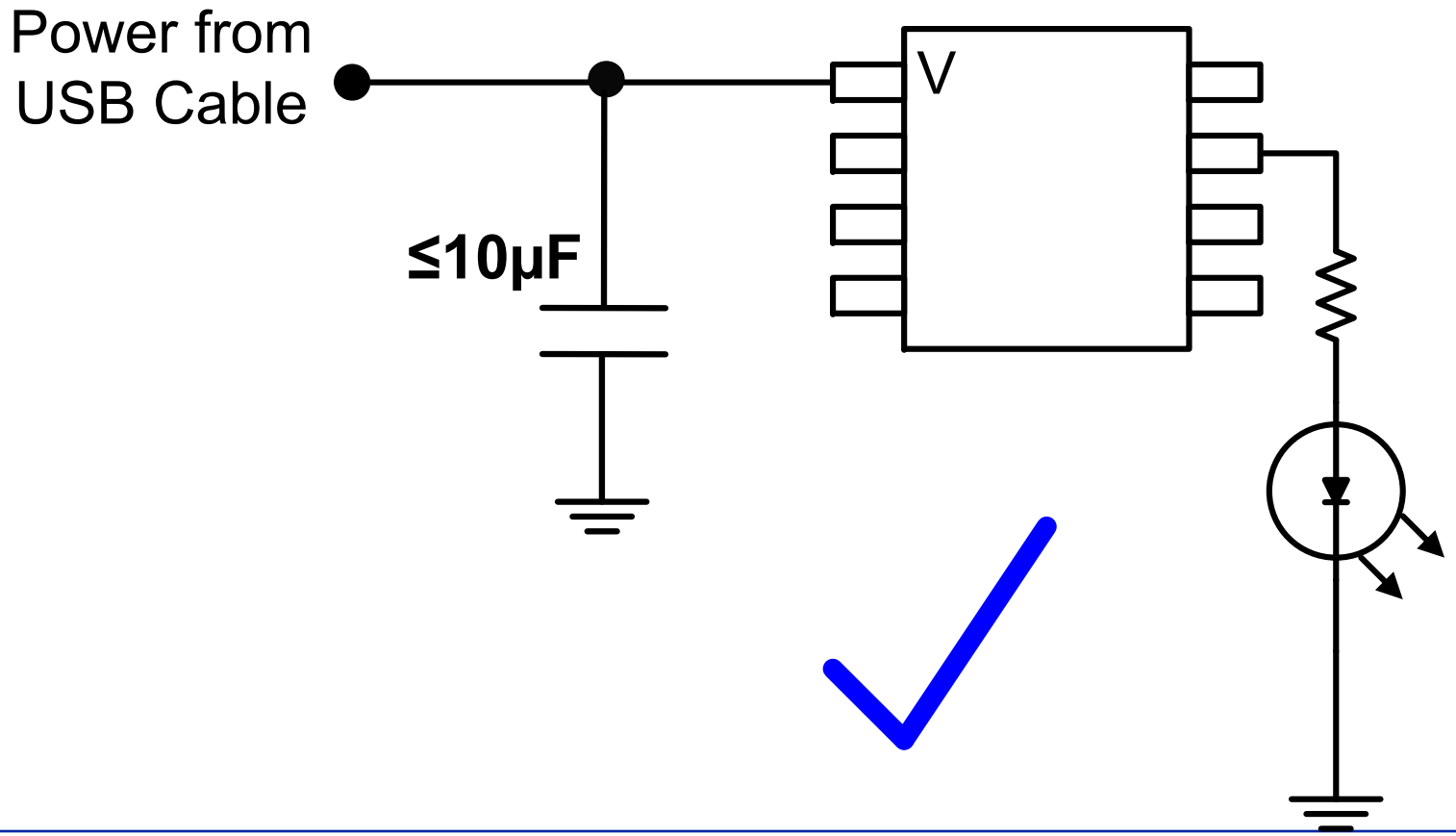
- **Max USB suspend current is 0.5/2.5mA**
- **Don't:**

Power from USB Cable

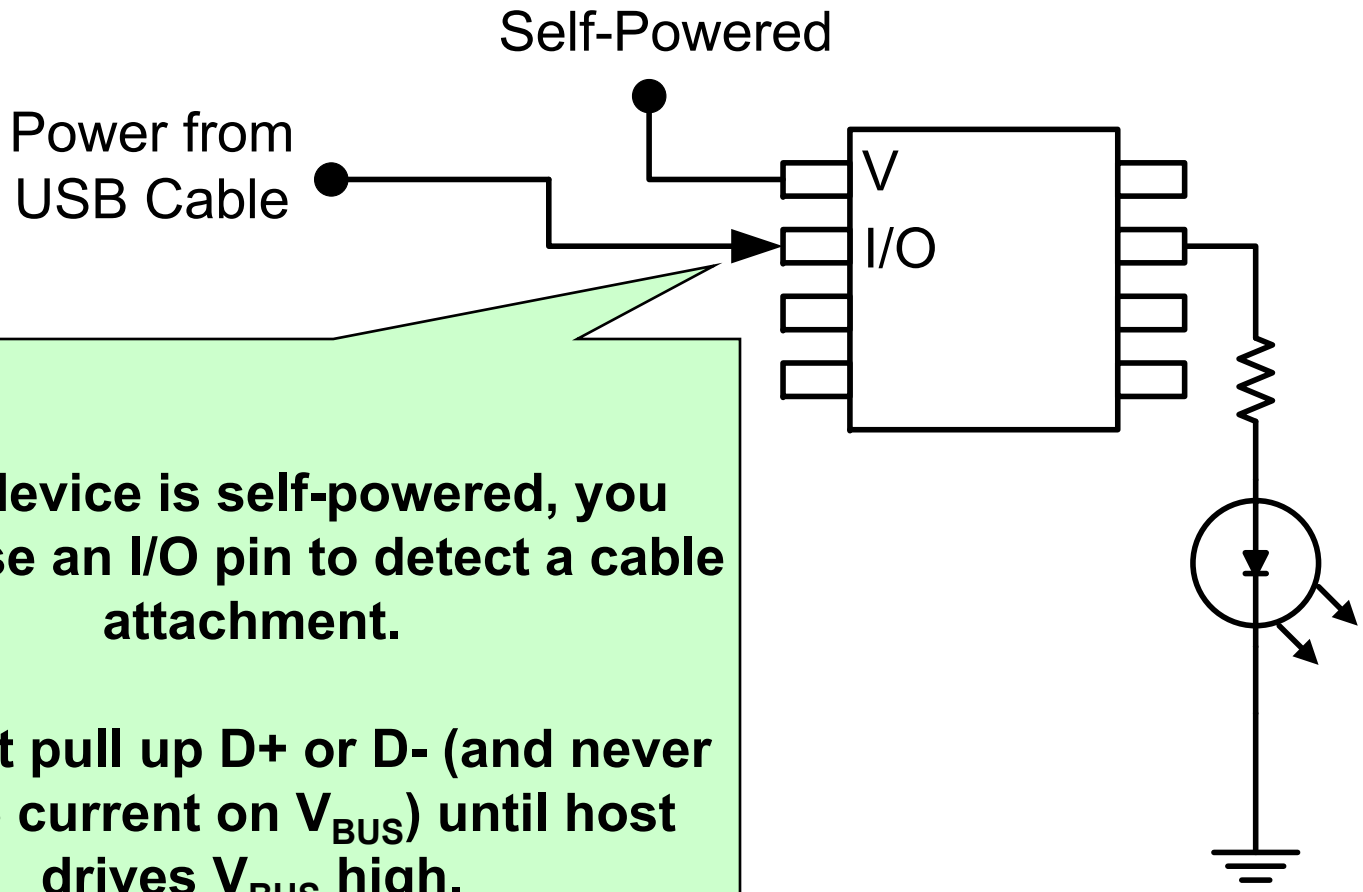


Power Planning

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



PIC® MCU: Detecting a USB Attachment



If the device is self-powered, you <u>MUST</u> use an I/O pin to detect a cable attachment.

Must not pull up D+ or D- (and never source current on V_{BUS}) until host drives V_{BUS} high.

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 your own 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” 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"

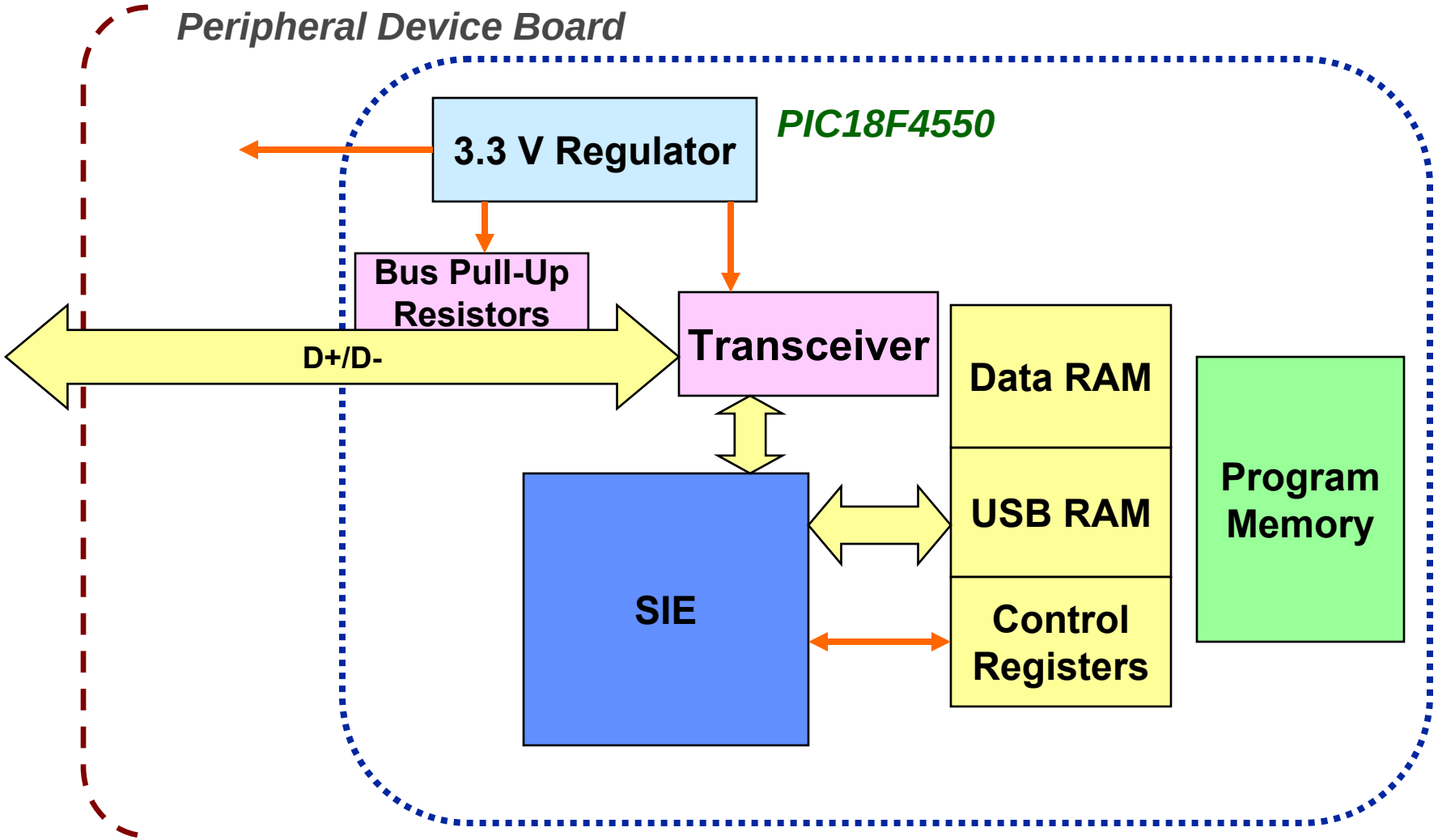
  - TID: Test ID
  - Use certified USB receptacle and cable for testing
  - Know the TID of your components
  - All USB PIC® MCUs have a TID number. Get it at www.microchip.com/usb

- **Probably a good idea to take a look at the checklist even before starting your design!**

Agenda

- **Brief history of USB & USB-IF**
- **USB Fundamentals - The serious & important stuff**
  - Basics/Speeds
  - Architecture/Programmer's Model
  - Physical Connection
  - USB Transactions
  - USB Transfers
  - Device Classes
  - Enumeration
  - Descriptors
  - Power Planning
  - VID/PID & USB Compliance
- **PIC18F USB Microcontrollers**
- **Microchip Demo/Development Solutions**

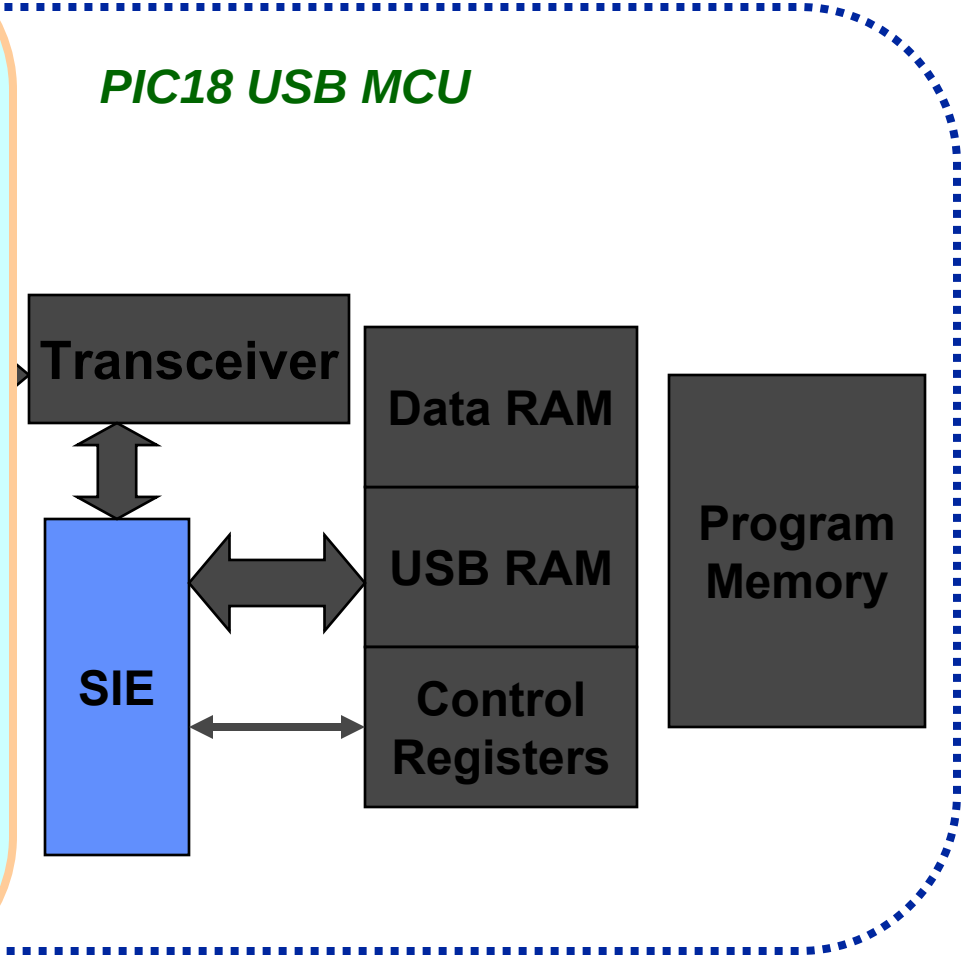
USB Module



Serial Interface Engine

SIE ...

- Serializes and deserializes USB data
- Encodes and decodes NRZI data
- Handles bit stuffing
- Checks CRC to validate data packet
- Detects bus signaling events and notifies the CPU through interrupts
- Handles USB transactions
- Handles handshaking protocol



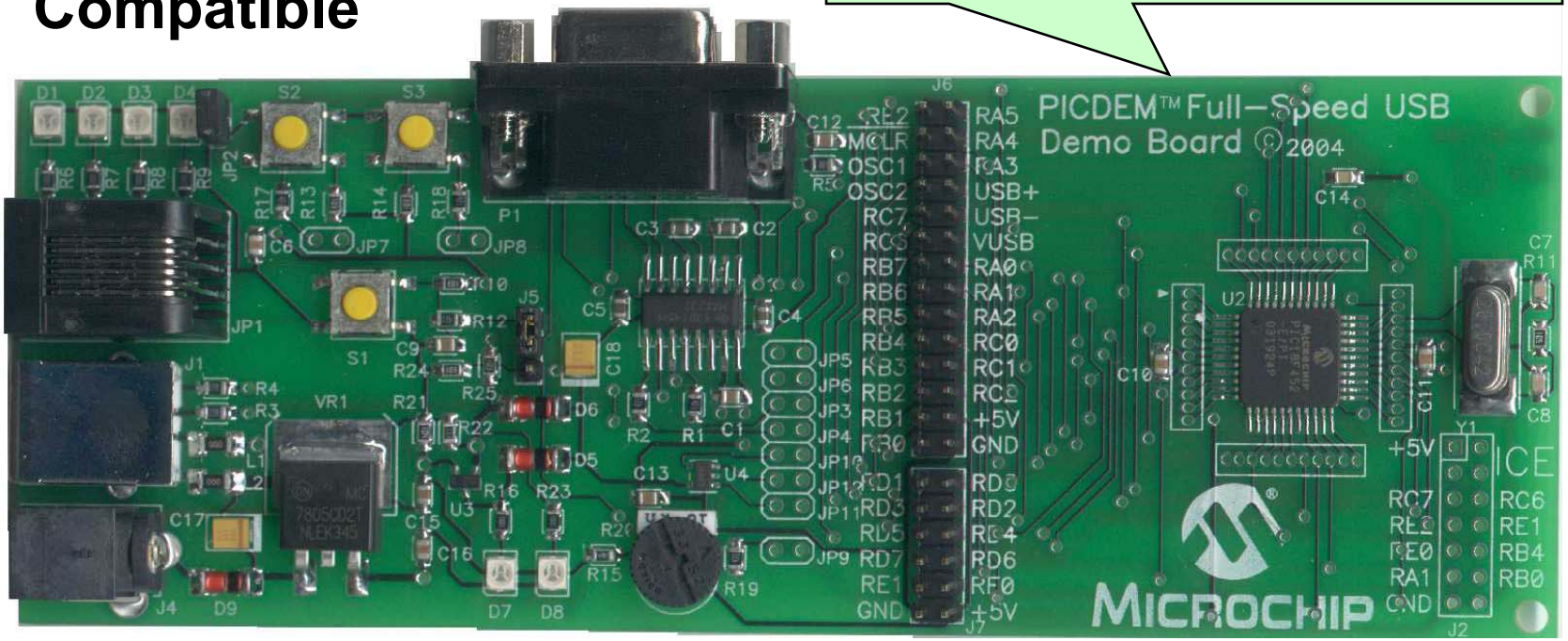
Agenda

- **Brief history of USB & USB-IF**
- **USB Fundamentals - The serious & important stuff**
  - Basics/Speeds
  - Architecture/Programmer's Model
  - Physical Connection
  - USB Transactions
  - USB Transfers
  - Device Classes
  - Enumeration
  - Descriptors
  - Power Planning
  - VID/PID & USB Compliance
- **PIC18F USB Microcontrollers**
- **Microchip Demo/Development Solutions**

PICDEM™ Full-Speed USB

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

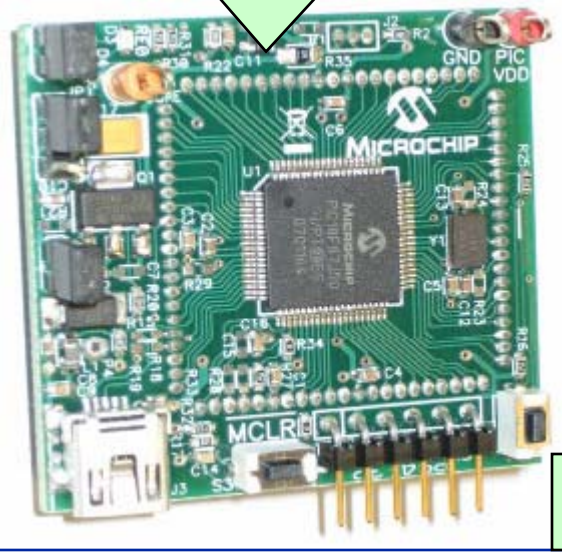
Part Number - DM163025



PIC18F87J50 FS USB Plug-In Module (PIM)

- PIC18F87J50, 12 MHz xtal
- USB Port, mini-B
- Linear Regulator
- 2 LEDs, 2 Pushbuttons
- Stand alone operation
- Operation with: HPC Explorer Demo Board
- ICSP™ Technology: 6-pin header

Part Number – MA180021



Part Number – AC164110

HPC Explorer – DM183022

PICtail™ Daughter Board for SD and MMC

- Secure Digital Card Interface - using SPI
- AN1003: USB Mass Storage

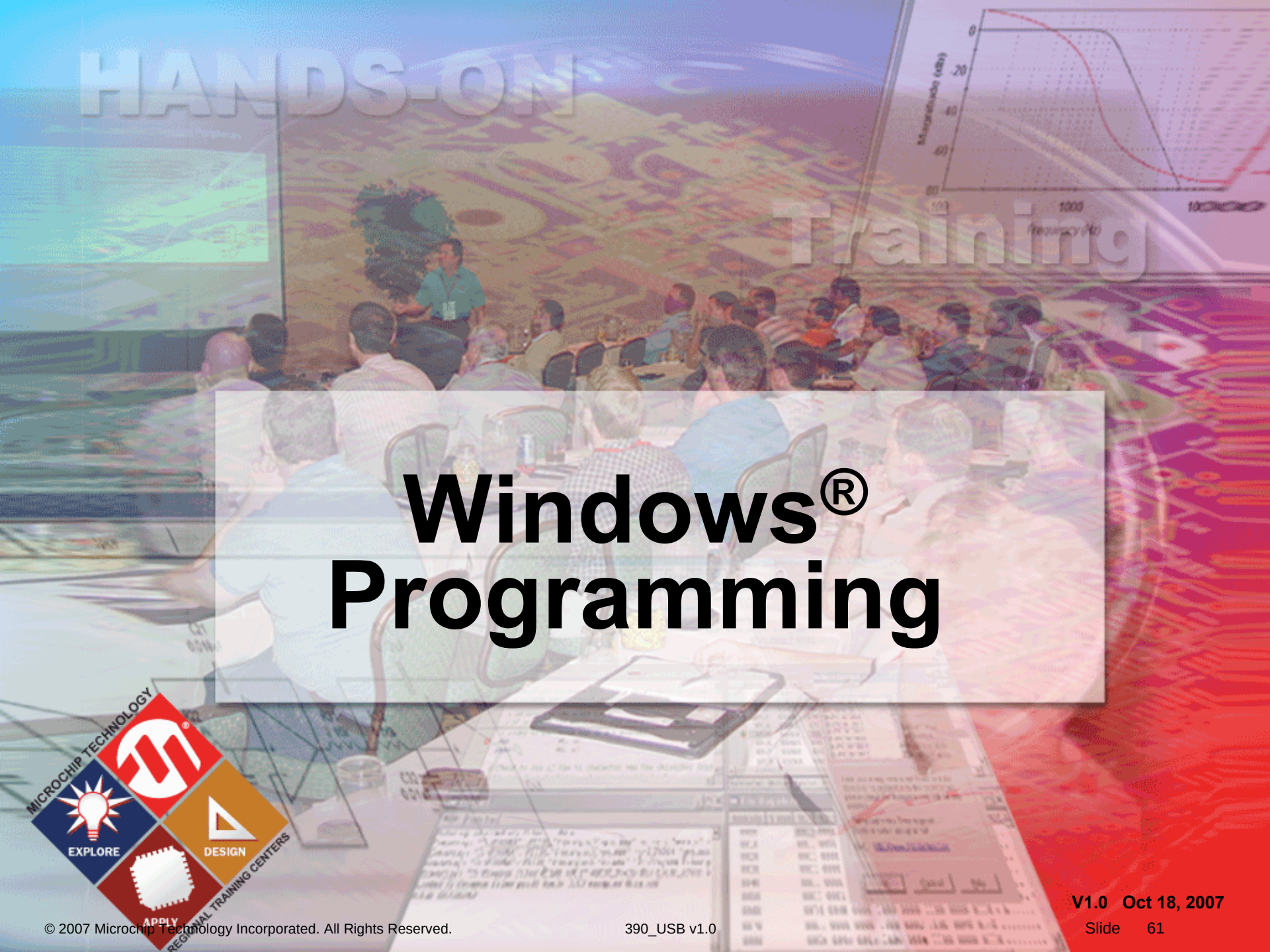
Part Number – AC164122

Microchip USB Firmware Characteristics

- **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

Summary

- **Basics/Architecture**
  - Up to 126 devices sharing bandwidth
  - Host is master
- **Host/Device Communication**
  - Transactions
  - Transfers
- **Enumeration/Chapter 9**
  - Descriptors
- **Microchip Offerings: MCUs, Demo Boards, Firmware, Custom Driver**



Windows® Programming

Windows® Programming

- **With Visual C++ 2005 you have three basic ways of creating a interactive, graphical application:**
  - Using the **Windows API (Win32)**
    - Most programming intensive
  - Using the **Microsoft Foundation Classes (MFC)**,
    - Encapsulate the Win32 API - easier
  - Using **Windows Forms (.NET Framework)**
    - Least programming intensive

- **Visual C++ 2005 also allows you to create two types of console application (Win32 and CLR (.NET))**
  - We will be developing a “Win32 Console” C application
  - Why? Simplicity & Desire to focus on the DLL functions

USB OTG

“On The Go”

Types of USB

Supported

Not Supported

PIC24FJ256GB1

Mini-Host (Thumb Drives)

IN DEVELOPMENT

On the Go *(Master & Slave)*

IN DEVELOPMENT

Hub

Not Targets for Embedded Market

NOW

Low Speed 1.2 Mbps

Device/Slave/Node

NOW

High Speed 480 Mbps

Ideal for Embedded Market – Serial Port Replacement

Full Speed 12 Mbps

Other USB PIC microcontrollers

USB Device Classes

Beta Support

Data Glove

Mouse

PICkit™ 2 Starter Kit

Joystick

Keyboard

Human Interface Device Class (HID)

Floppy Drive

Thumb Drive

External Hard Drive

Mass Storage Device Class (MSD)

Ethernet Adapter

Modem

Communication Device Class (CDC)

ICD2

Custom Class (Vendor Class)

Many more USB classes….

Mass Storage Application Software Architecture for Thumb Drive Demo