University of Microchip
U of M
MASTERs 2007

11045 UB2

Full-Speed USB
Hands-On Training

Prerequisites

Working knowledge of:

- PIC18 family of microcontrollers
- MPLAB® IDE
- C Programming Language

This training assumes you are familiar with basic USB terminology and concepts

Objectives

Upon completion, participants will:

- Get a hands-on experience in creating USB applications

- Be aware of the factors important in designing USB applications

Agenda

Checking your work station…

Quick USB Review

Demonstration - Microchip USB Solutions

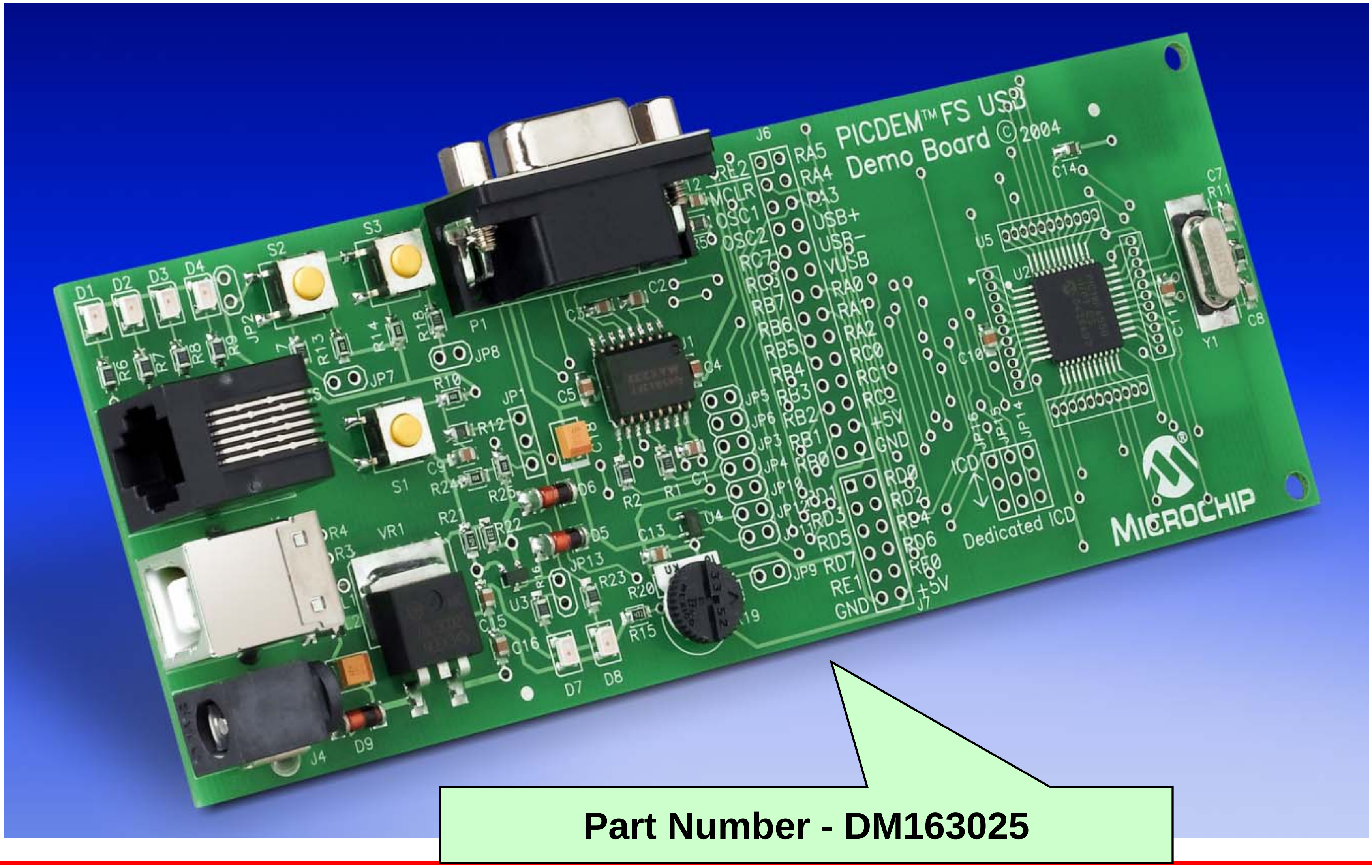
Exercise Group 1: Your turn to play!

Exercise Group 2: CDC RS-232 Emulation APIs

Exercise Group 3: Custom Class using Microchip General Purpose USB Windows Driver

Wrapping up and Q&A

PICDEM™ FS USB Demo Board



Part Number - DM163025

Literature Check…

Three documents:

- **PIC18F4550 family data sheet**
- **Application Note 956**
- **PICDEM™ FS USB Demo Board User’s Guide**

Please *DO NOT* remove these documents from the classroom. Your fellow attendees will need them in the next session!

Thank you for your cooperation!

Agenda

Checking your work station…

Quick USB Review

Demonstration - Microchip USB Solutions

Exercise Group 1: Your turn to play!

Exercise Group 2: CDC RS-232 Emulation APIs

Exercise Group 3: Custom Class using Microchip General Purpose USB Windows Driver

Wrapping up and Q&A

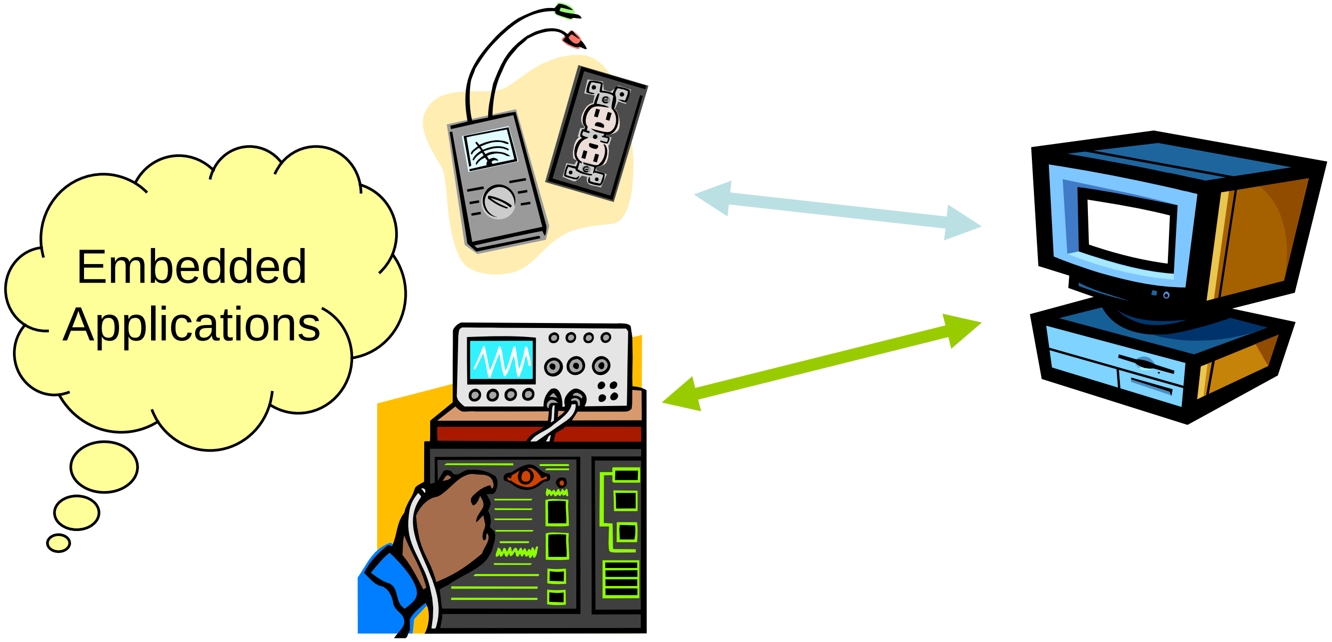
Quick USB Review

Understanding of the following concepts is important in making USB design choices

We will briefly cover materials on:

- USB in Embedded Applications
- Vendor ID (VID) & Product ID (PID)
- Device Classes

Universal Serial Bus



Embedded Applications

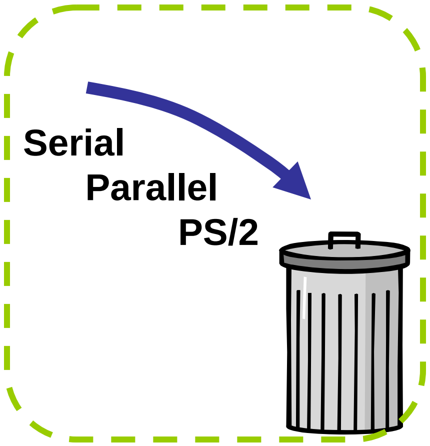
Computer is great

Processing power!

Data Analysis,
Data Logging,
Firmware Updates,
Diagnostic

- Auto detection & configuration (Plug&Play)
- Easy expansion using hubs
- Bus power

- Three speeds:
  Low- 1.5, Full- 12, High- 480 Megabits / second



Serial
Parallel
PS/2

Vendor ID & Product ID

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*

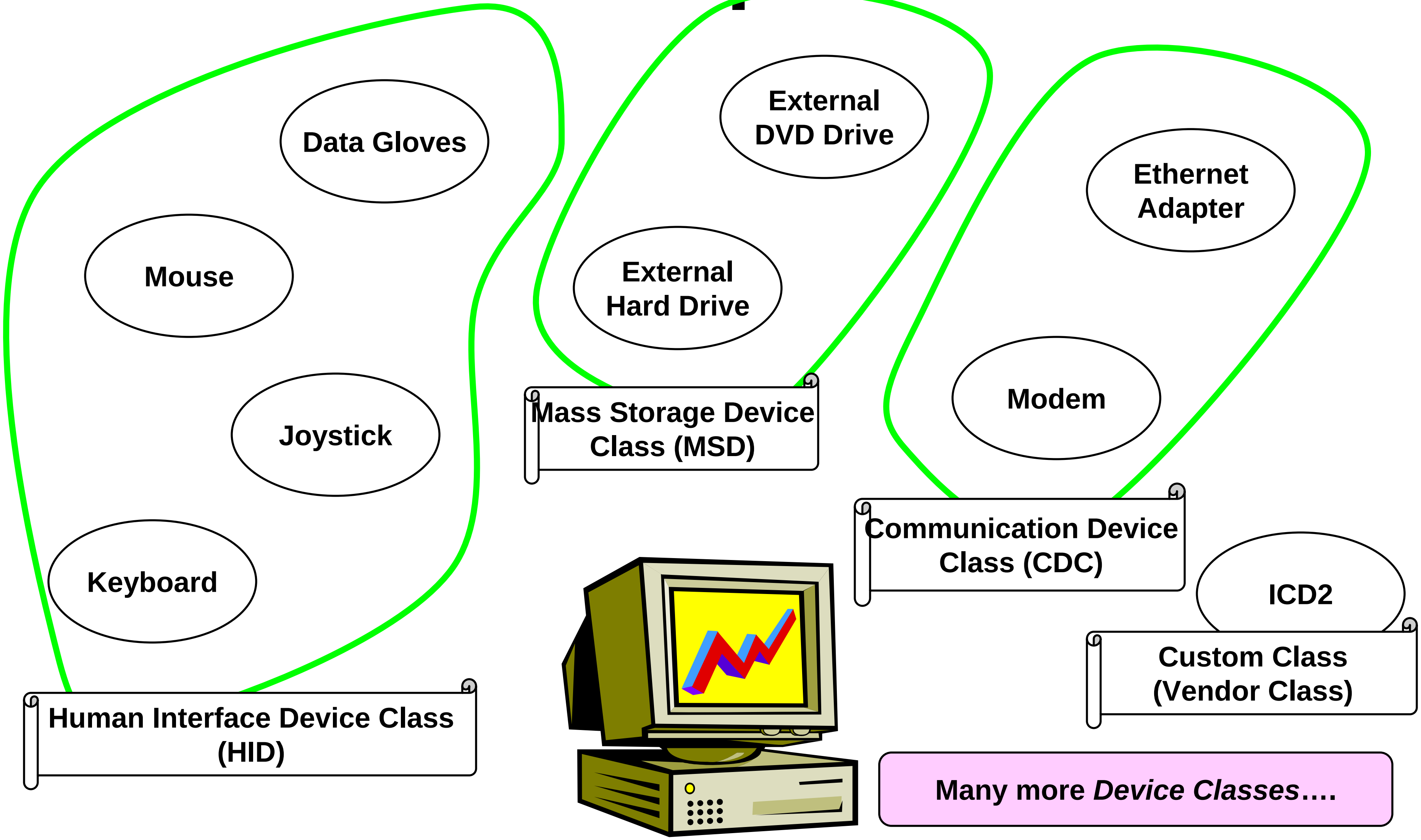
- Microchip's Sub-licensing Program

Every product is required to have a unique combination of VID and PID

Compliance Testing

- Must pass to use USB logo
- USD $1,500

Categorizing USB Peripherals



Data Gloves

Mouse

Joystick

Keyboard

Human Interface Device Class (HID)

External DVD Drive

External Hard Drive

Mass Storage Device Class (MSD)

Ethernet Adapter

Modem

Communication Device Class (CDC)

ICD2

Custom Class (Vendor Class)

Many more *Device Classes*….

Agenda

Checking your work station…

Quick USB Review

Demonstration - Microchip USB Solutions

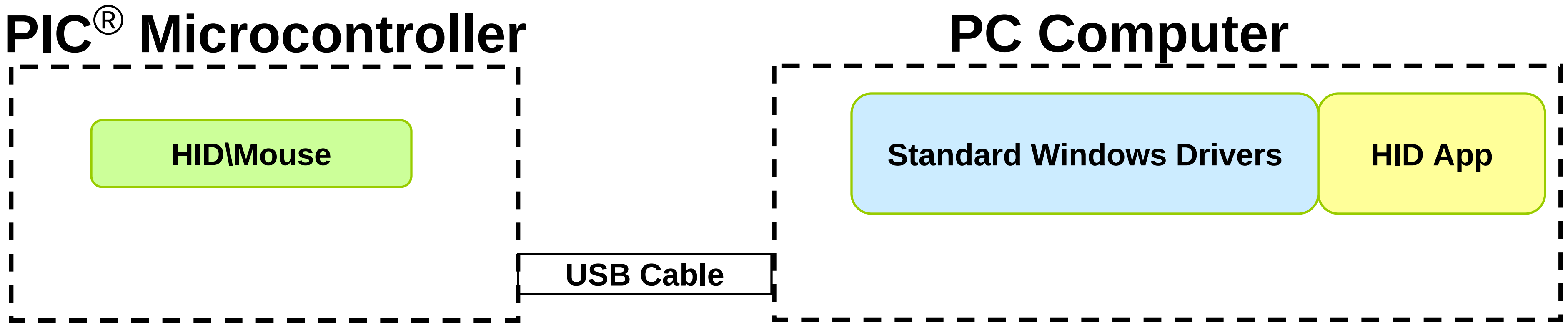
Exercise Group 1: Your turn to play!

Exercise Group 2: CDC RS-232 Emulation APIs

Exercise Group 3: Custom Class using Microchip General Purpose USB Windows Driver

Wrapping up and Q&A

Human Interface Device (HID)



PIC® Microcontroller

HID\Mouse

USB Cable

PC Computer

Standard Windows Drivers

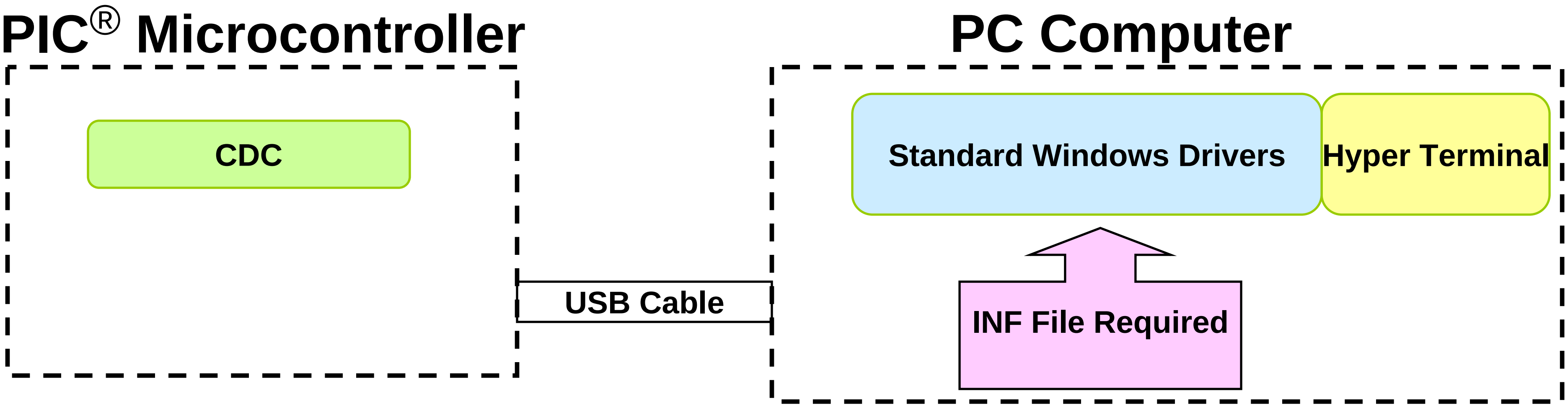
HID App

Design Considerations:

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

PICkit™ 2 Kit
Uses HID Class!

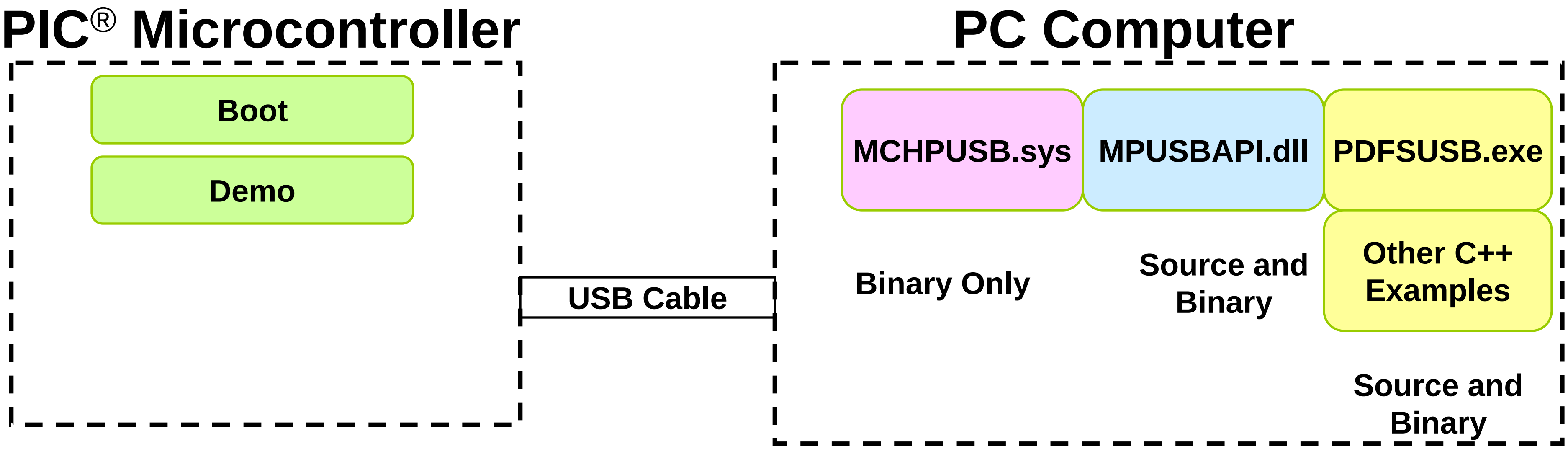
CDC - RS-232 Emulation



Design Considerations:

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

General Purpose Driver



PIC® Microcontroller

Boot

Demo

USB Cable

PC Computer

MCHPUSB.sys

MPUSBAPI.dll

PDFSUSB.exe

Other C++ Examples

Binary Only

Source and Binary

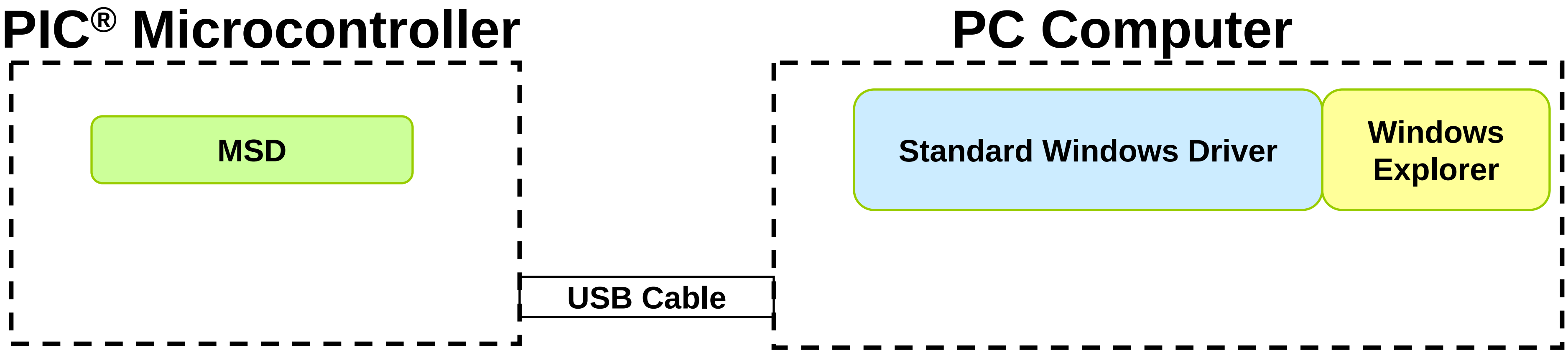
Source and Binary

Design Considerations:

- ~1,088 KB/s max
- Very flexible
- Not a standard Windows driver
- PC programming is required

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

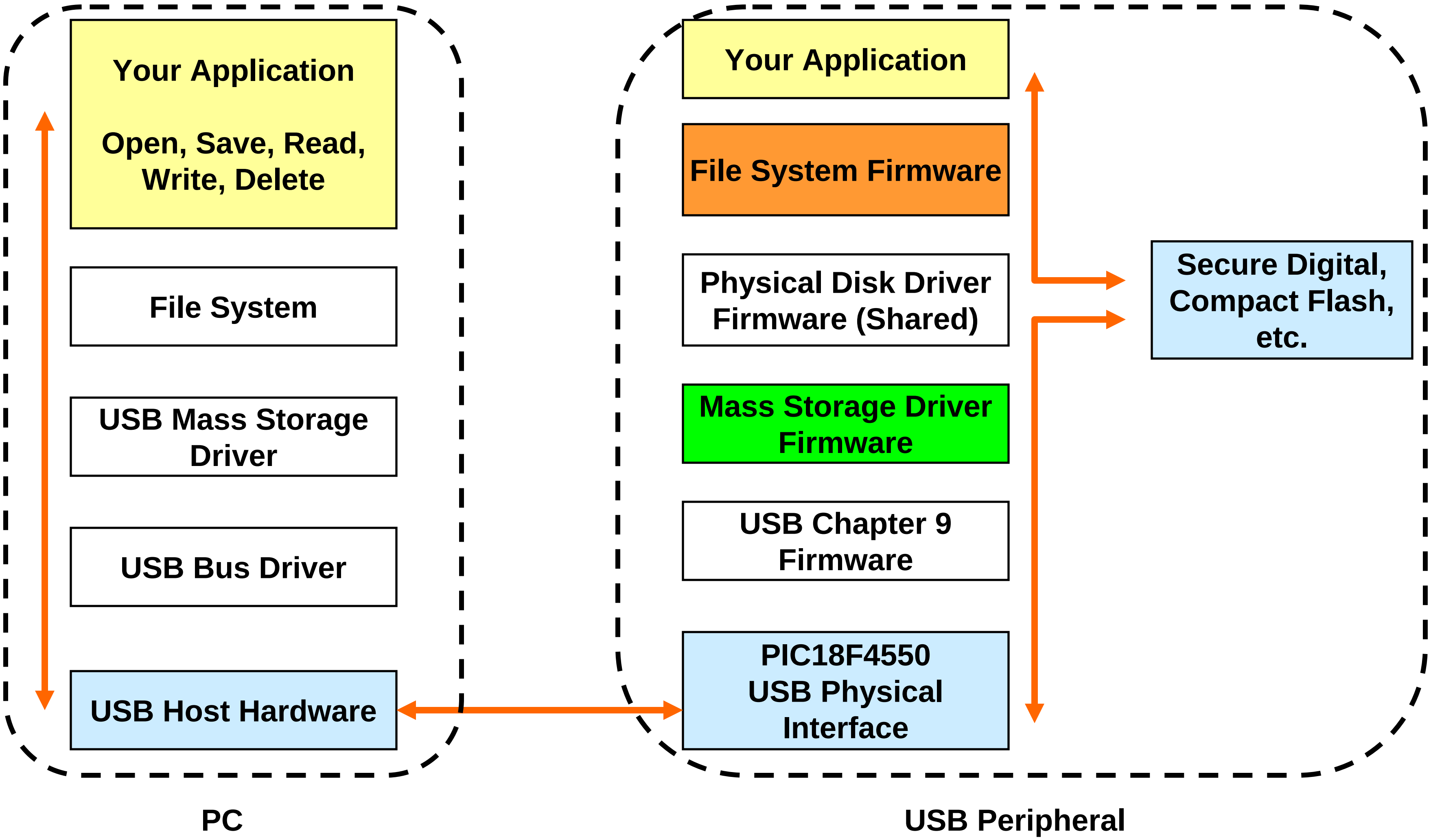
Mass Storage Device (MSD)



Design Considerations:

- Acts like a disk drive
- Fast data transfer over USB
- Speed is limited depending on the physical media interface
- For usefulness, a File System should also be implemented in firmware

Typical MSD Application

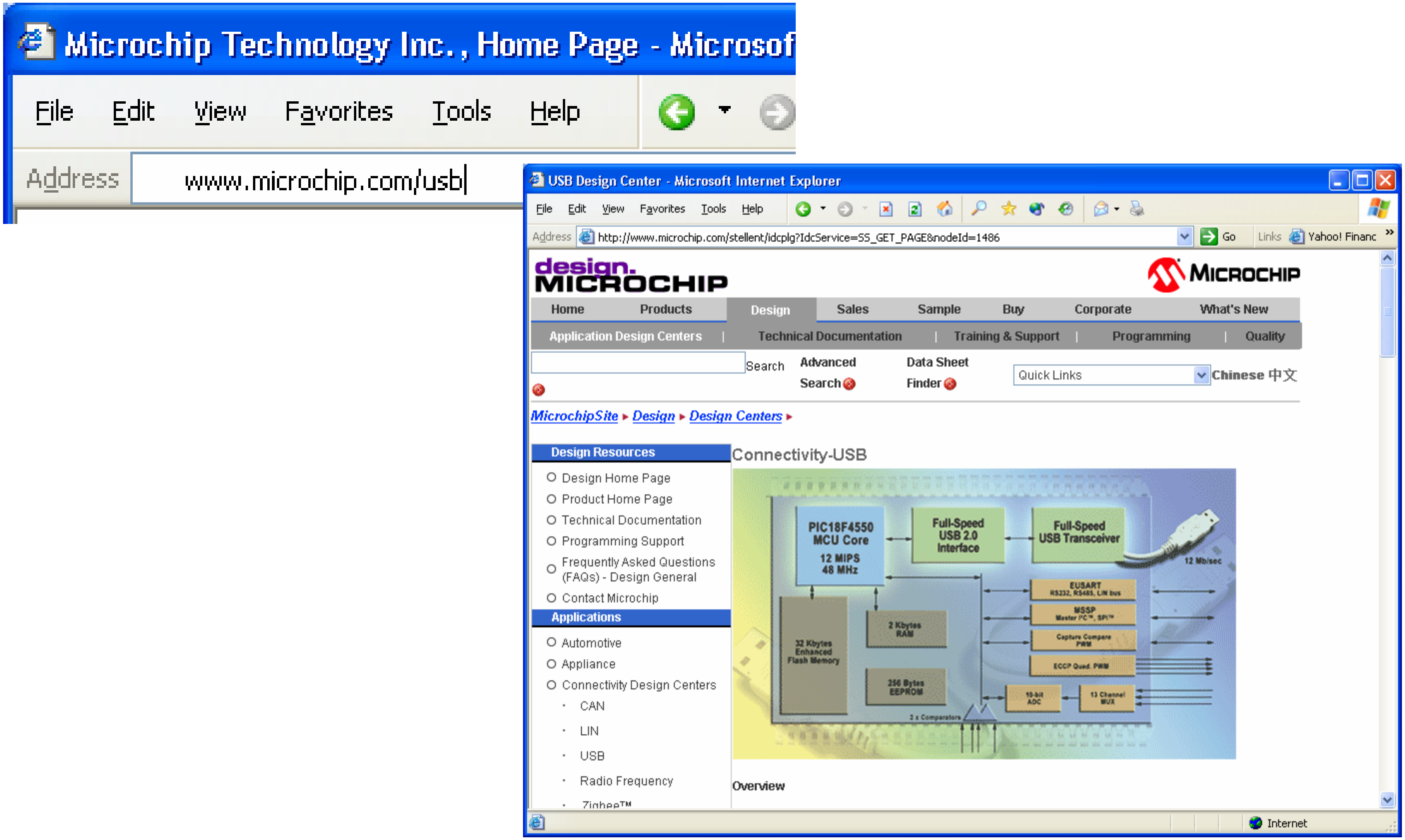

Your Application
Open, Save, Read, Write, Delete
File System
USB Mass Storage Driver
USB Bus Driver
USB Host Hardware
PC
Your Application
File System Firmware
Physical Disk Driver Firmware (Shared)
Mass Storage Driver Firmware
USB Chapter 9 Firmware
PIC18F4550 USB Physical Interface
Secure Digital, Compact Flash, etc.
USB Peripheral

What Does Your Application Need?

- HID
- CDC – RS-232 Emulation
- General Purpose
- MSD

- Others..?

Software Package



Microchip Technology Inc., Home Page - Microsof

Address www.microchip.com/usb

USB Design Center - Microsoft Internet Explorer

Address http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1486

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
  - Zigbee™

Connectivity-USB

Overview

University of Microchip UofM MASTERs 2007

Agenda

Checking your work station…

Quick USB Review

Demonstration - Microchip USB Solutions

Exercise Group 1: Your turn to play!

Exercise Group 2: CDC RS-232 Emulation APIs

Exercise Group 3: Custom Class using Microchip General Purpose USB Windows Driver

Wrapping up and Q&A

Exercise Group 1: Your turn to play!

At this point, 35-40 minutes should have lapsed

Exercise Group 1 should take approximately 30 minutes to complete

Exercise Group 1: Objectives

Familiarize you with the development environment

We will not be writing any code

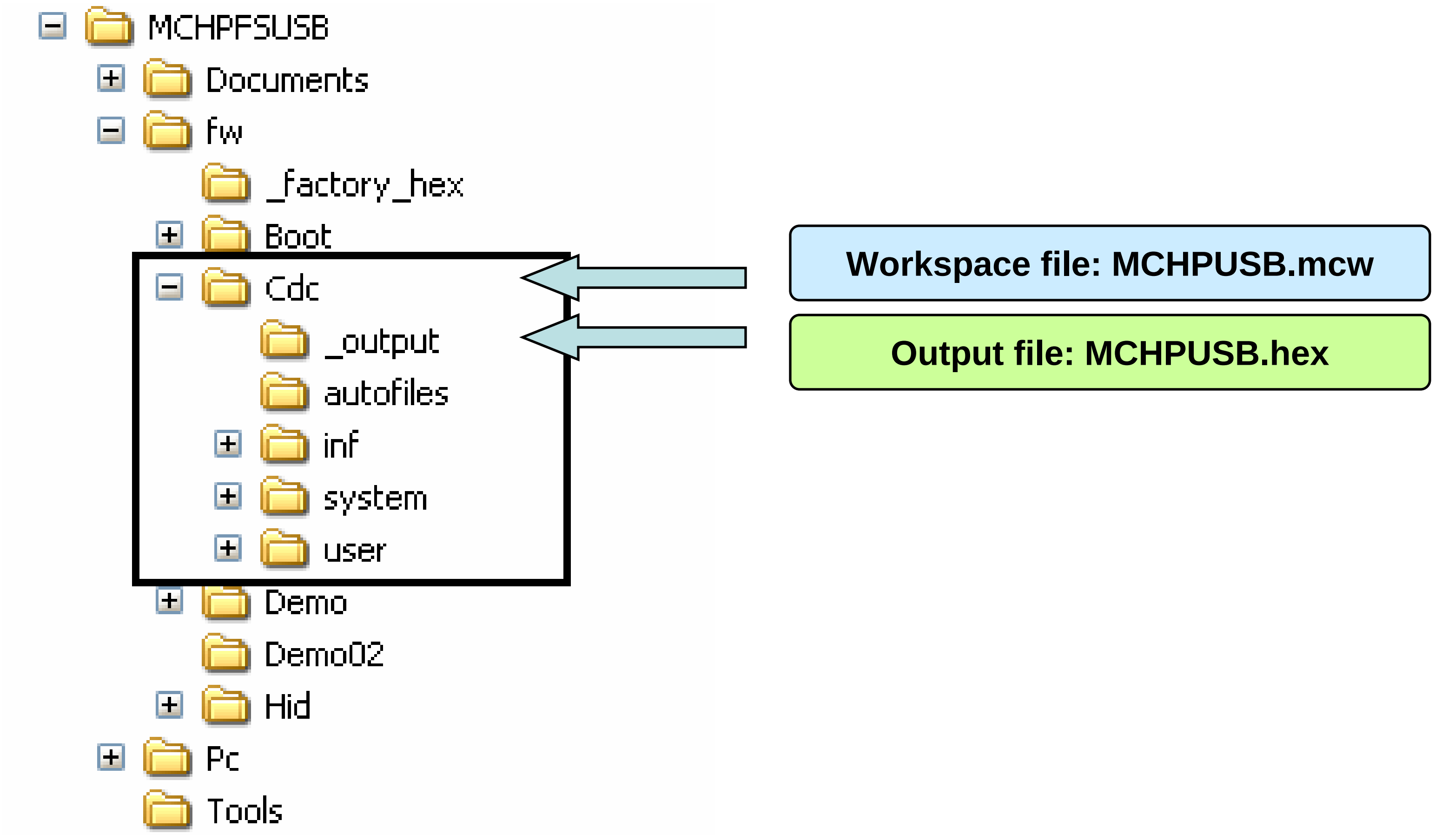
Just compiling, programming Flash memory, and playing with the default demonstration

Ex 1.1 General Purpose

Ex 1.2 Upload HID mouse example using the bootloader

Project Folder Structure

C:\MCHPFSUSB\



- MCHPFSUSB
  - Documents
  - fw
    - _factory_hex
    - Boot
    - Cdc
      - _output
      - autofiles
      - inf
      - system
      - user
    - Demo
    - Demo02
    - Hid
  - Pc
  - Tools

Workspace file: MCHPUSB.mcw

Output file: MCHPUSB.hex

Ex 1.1 – General Purpose Driver Demo

- Launch MPLAB® IDE
- Import
  - [illegible]
- Select MPLAB ICD2 as a programmer
- Program the device
- Remove the 9 inch modular cable and follow instructions in Chapter 3 of the PICDEM™ FS USB User's Guide on how to run the demo

End of Ex 1.1

Ex 1.2 – Download HID Mouse Demo via a Bootloader

Launch PDFSUSB.exe if you have not done so already

Enter bootload mode by holding down push button S2 and hit S1 to reset

Select Boot Mode Tab in PDFSUSB.exe

Follow instructions in Chapter 3 of the User’s Guide on how to download a hex file

Download:

[illegible]

[illegible]

End of Ex 1.2

Agenda

Checking your work station…

Quick USB Review

Demonstration - Microchip USB Solutions

Exercise Group 1: Your turn to play!

Exercise Group 2: CDC RS-232 Emulation APIs

Exercise Group 3: Custom Class using Microchip General Purpose USB Windows Driver

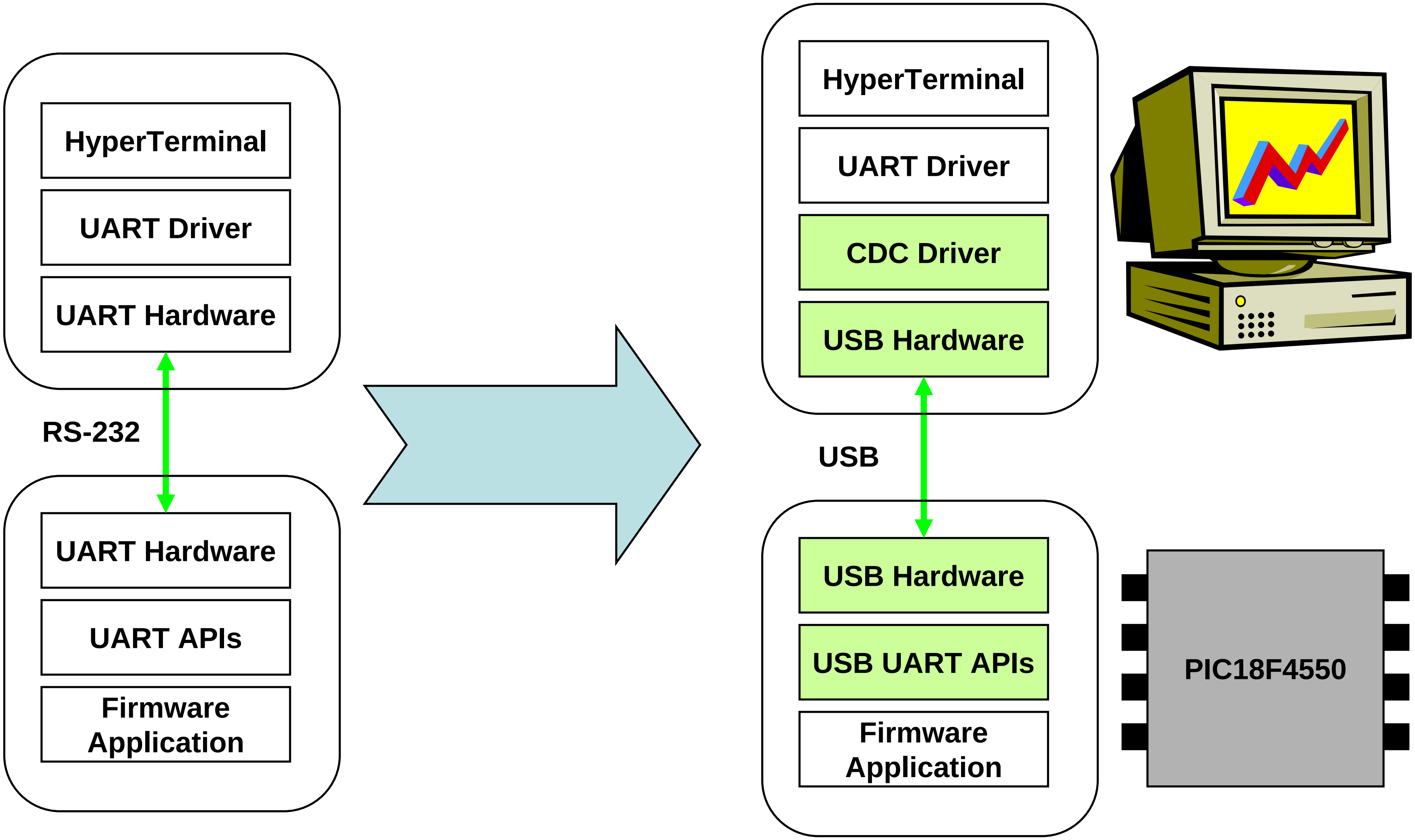
Wrapping up and Q&A

Exercise Group 2: Objectives

Learn how to use provided APIs to transfer data between a computer and the PIC18F4550

Be able to compile the project and download the hex file to program memory successfully

Migration Path



HyperTerminal

UART Driver

UART Hardware

RS-232

UART Hardware

UART APIs

Firmware Application

HyperTerminal

UART Driver

CDC Driver

USB Hardware

USB

USB Hardware

USB UART APIs

Firmware Application

PIC18F4550

Opening CDC Project Workspace

If using an ICD 2...

Attach 9” Modular cable to the demo board

Attach USB cable to the demo board

Select “File/Open Workspace…”

Open project “C:\MCHPFSUSB\fw\CDC\MCHPUSB.mcw”

Do not do anything else just yet

Configuration Bits

FS USB Clock Source: Clock src from 96 MHz PLL/2

CPU Sys Clk Postscaler: [OSC1/OSC2 Src: /1][96MHz PLL Src: /2]

96MHz PLL Prescaler: Div by 5 {20 MHz input}

Oscillator: HS: HS+PLL, USB-HS

USB Voltage Regulator: Enabled

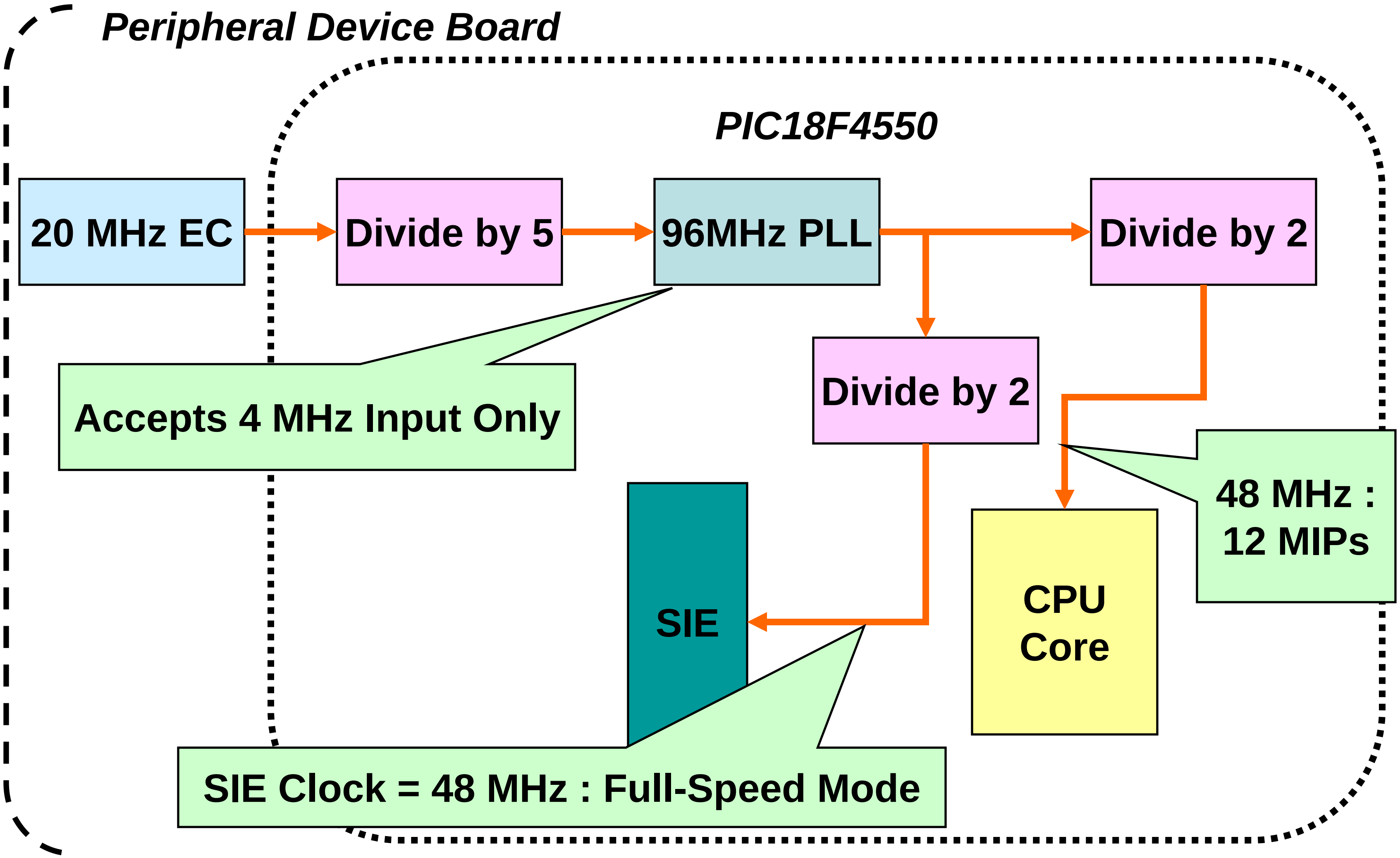
Watchdog Timer: Disabled

PortB A/D Enable: configured as digital I/O on RESET

Low Voltage Program: Disabled

The only file you will need to modify is user.c

Clock Configuration Example



Running the Default Code

Press F10 to compile project

Select “Programmer\Select Programmer” and make sure that MPLAB® ICD2 is selected

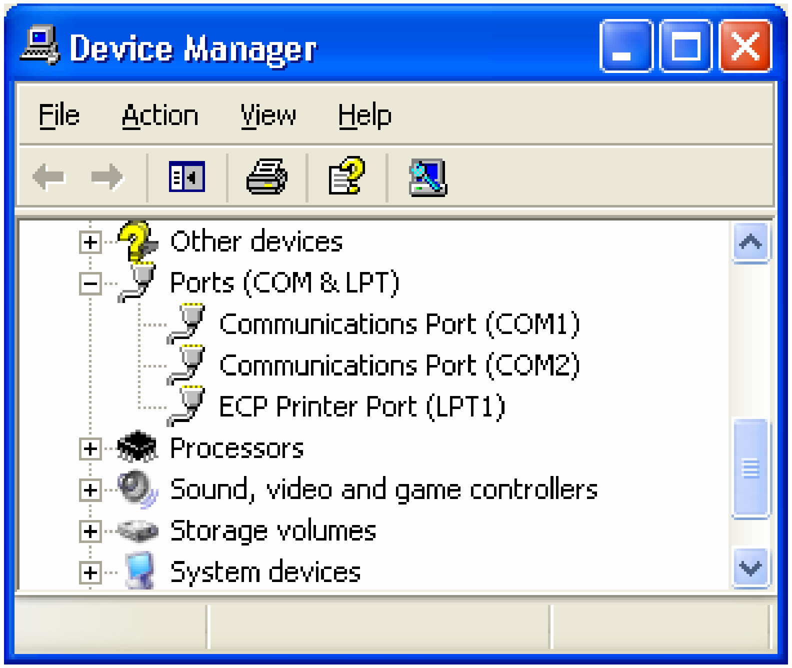
Select “Programmer\Program” to download the code to the microcontroller

Do not disconnect 9” Modular cable just yet

Device Manager

Open "Device Manager" and take a look at the "Ports (COM & LPT)" category

Observe how many COM ports are currently there on your computer



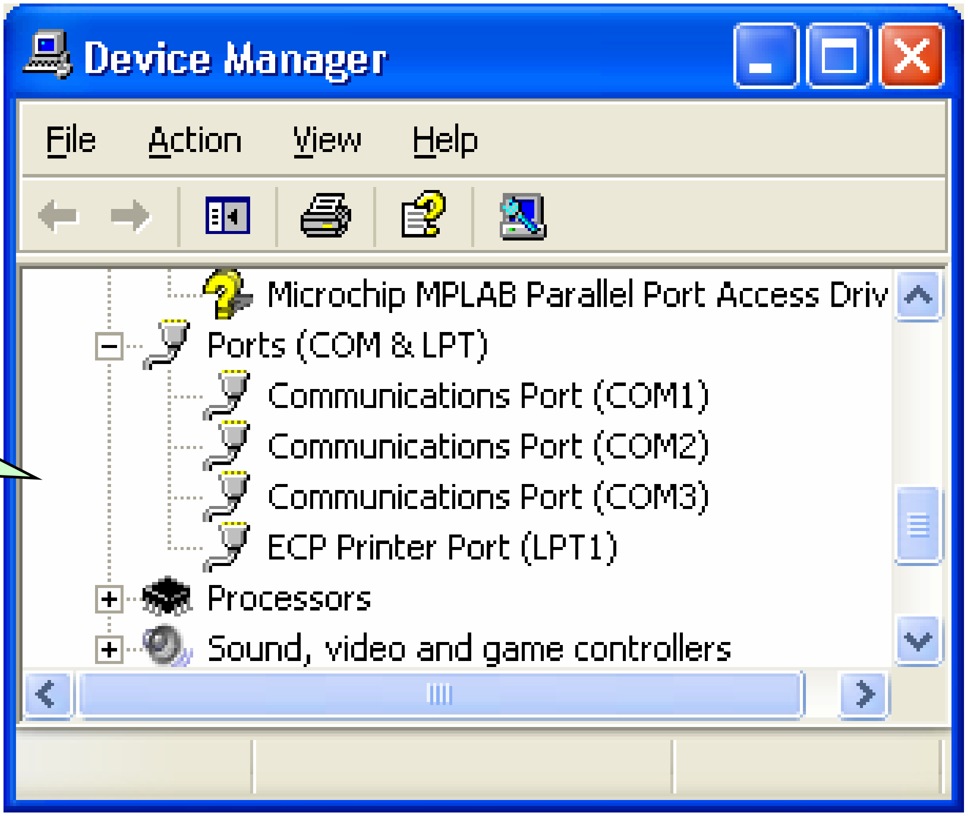
Other devices
Ports (COM & LPT)
Communications Port (COM1)
Communications Port (COM2)
ECP Printer Port (LPT1)
Processors
Sound, video and game controllers
Storage volumes
System devices

Device Manager

Now disconnect the 9" modular cable

The device should be detected by Windows

You should see an additional 'virtual' COM port. Remember the number of the COM port

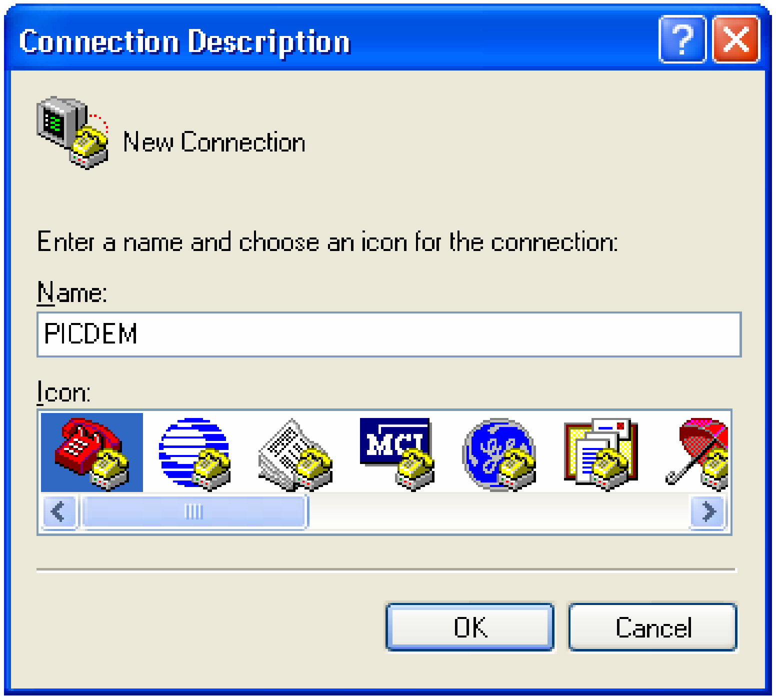


Microchip MPLAB Parallel Port Access Driv
Ports (COM & LPT)
Communications Port (COM1)
Communications Port (COM2)
Communications Port (COM3)
ECP Printer Port (LPT1)
Processors
Sound, video and game controllers

Using HyperTerminal

Launch HyperTerminal

Type in a name for a new connection, and hit “OK”



New Connection

Enter a name and choose an icon for the connection:

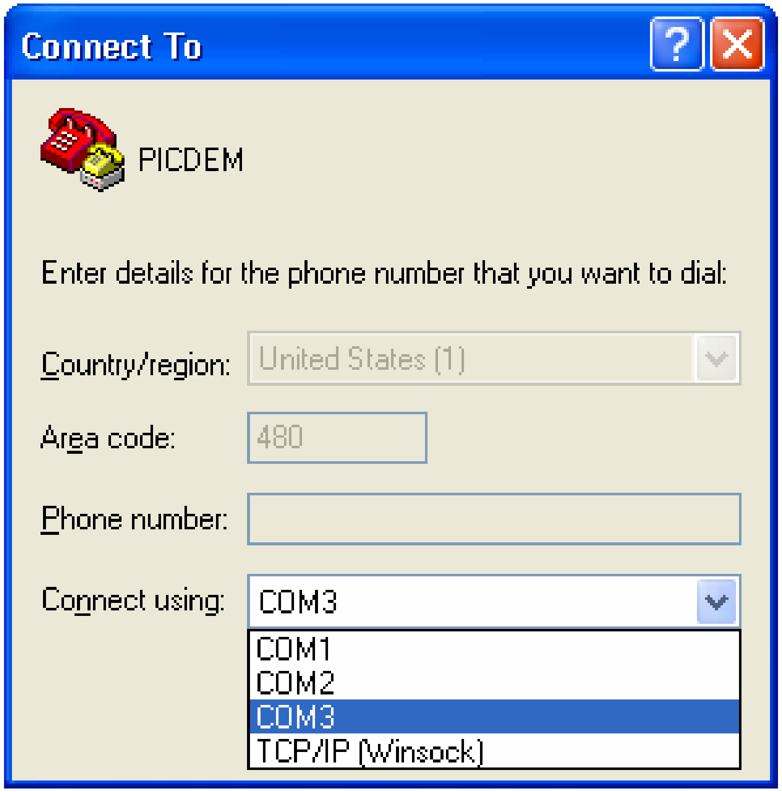
Name:

PICDEM

Icon:

HyperTerminal: Connect To

Select the new COM port that was added during USB CDC installation and hit “OK”



Connect To

PICDEM

Enter details for the phone number that you want to dial:

Country/region: United States (1)

Area code: 480

Phone number:

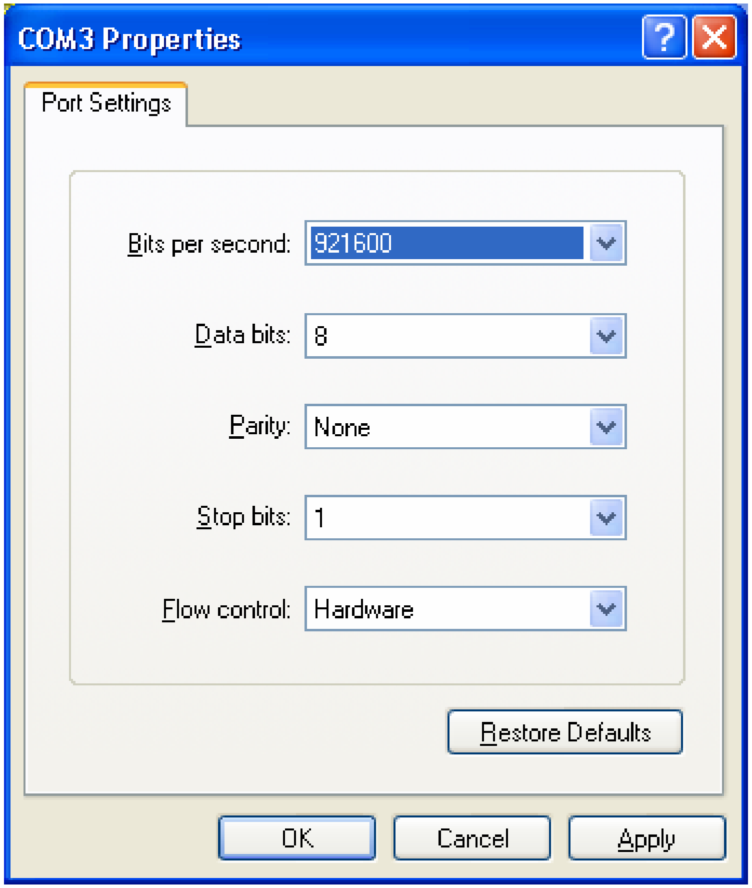
Connect using: COM3

COM1
COM2
COM3
TCP/IP (Winsock)

HyperTerminal: COM Properties

Hit “OK”

Terminal is now ready



COM3 Properties

Port Settings

Bits per second: 921600

Data bits: 8

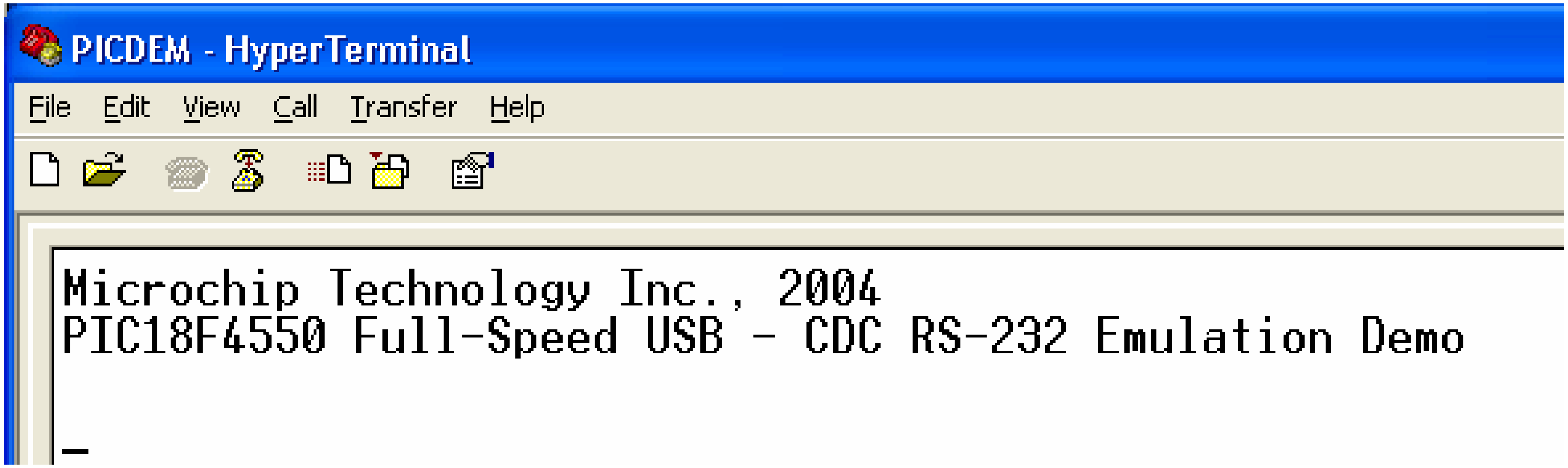
Parity: None

Stop bits: 1

Flow control: Hardware

HyperTerminal: First Use

Now, hit switch 2 on the demo board, you should see the message below…



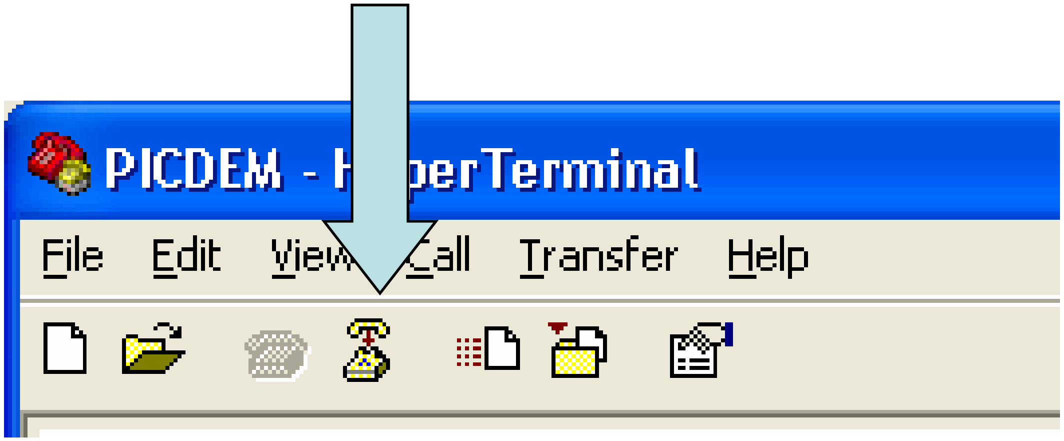
```
Microchip Technology Inc., 2004
PIC18F4550 Full-Speed USB - CDC RS-232 Emulation Demo


_
```

HyperTerminal: Important Note

Before physically disconnecting a USB device, you must hang up the connection first, otherwise, the program must be closed and opened again the next time a USB device is connected.

Let's take a look at the APIs



CDC APIs

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

Determinants of API

Do you know the length of the data you want to send?

Is the data stored in ROM or RAM?

Is the string of data null-terminated (end in 0x00)?

API for Null-terminated Literal & ROM Data

Use ‘putrs’ for literal and data stored in ROM

Usage Example:
putrsUSBUSART(“Hello”);
or
putrsUSBUSART(welcome);
where
rom char welcome[]=(“Bye”);

Importance of Checking State

cdc_trf_state

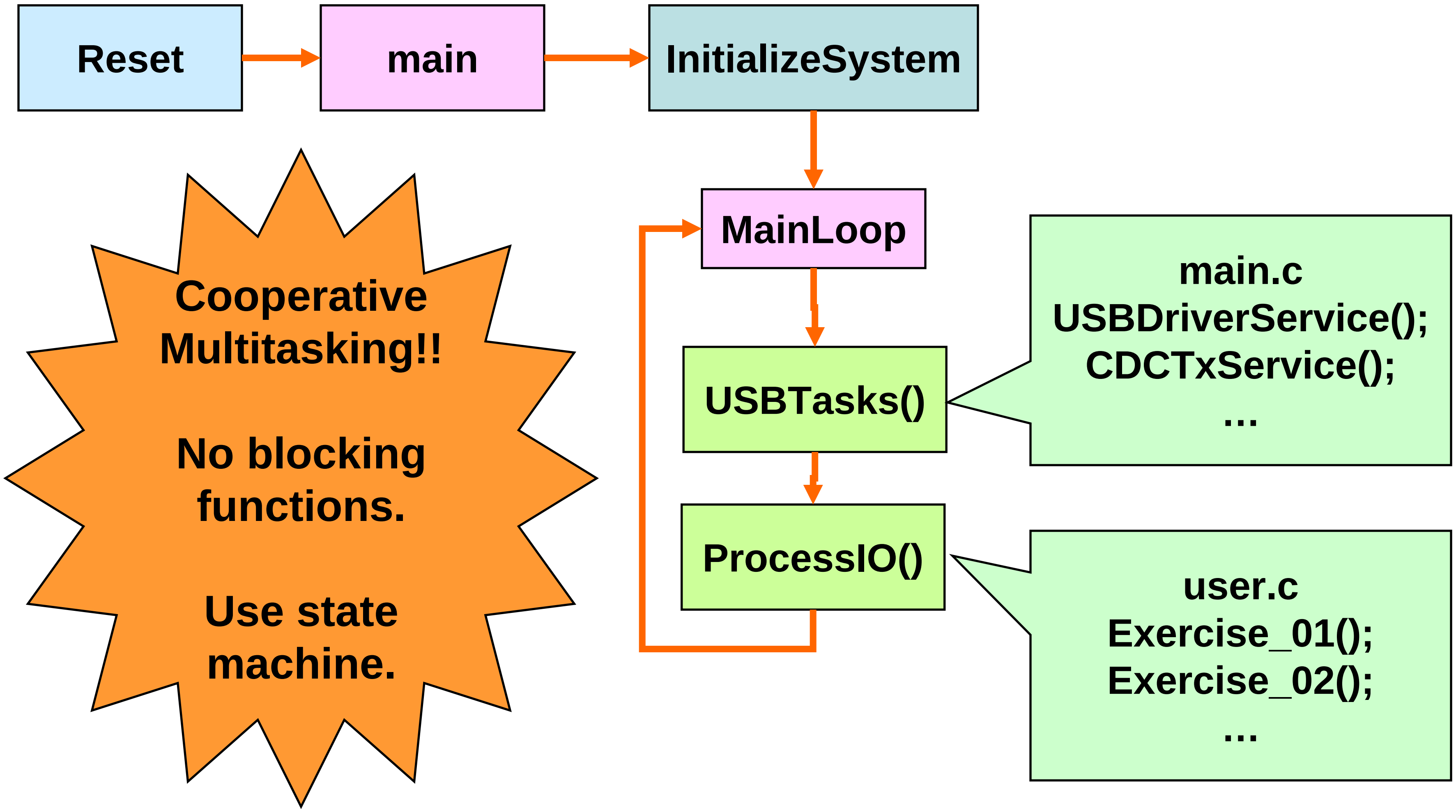
- CDC_TX_READY
- CDC_TX_BUSY
- CDC_TX_BUSY_ZLP
- CDC_TX_COMPLETING

When calling an API that sends data to the host, must check that:
cdc_trf_state == CDC_TX_READY

Or use macro:

- [illegible]

Program Flow



Reset

main

InitializeSystem

MainLoop

USBTasks()

ProcessIO()

main.c
USBDriverService();
CDCTxService();
…

user.c
Exercise_01();
Exercise_02();
…

Cooperative Multitasking!!

No blocking functions.

Use state machine.

Exercise 2.1: Sending Literal Data

Find function [illegible] in user.c

Write code in this function that sends a literal null-terminated string of text (“Hello World!\r\n”) to the PC when switch 2 is pressed

Take a look at examples in Exercise_Example();

Useful functions:

- [illegible]
    - **Returns 1 if switch 2 was pressed**
    - **Returns 0 if switch 2 was not pressed**
- [illegible]
- Use: [illegible].

End of Exercise 2.1

Exercise 2.2: Sending ROM Data

Find function [illegible] in user.c

Write code in this function that sends a null-terminated string of text stored in program memory pointed to by “ex02_string” to the PC when switch 3 is pressed

Useful functions:

- [illegible]
- [illegible]
- Use: [illegible].

End of Exercise 2.2

APIs for Receiving Data

len = expected number of input

buffer = pointer to user buffer of size equals to or larger than len

getsUSBUSART returns actual number of bytes copied to user buffer

Exercise 2.3: Receiving Data

Find function [illegible] in user.c

Write code in this function that reads data from USB bus and toggles LED D4 when the data read equals ASCII character ‘1’ (0x31)

Useful functions & variable:

- [illegible]
- [illegible]
- Use: [illegible].

End of Exercise 2.3

APIs for a String of Data with Known Length

Use mUSBUSARTTxRom for data stored in program memory

255-byte maximum

Use mUSBUSARTTxRam for data stored in data memory

Exercise 2.4: Sending Non Null-terminated Data

Before starting, comment out call to [illegible] first

Find function [illegible] in user.c

Write code in this function that sends the following 4 bytes of data: 0x30,0x31,0x32,0x33 when switch 2 is pressed

Useful function & variable:

- [illegible]
- [illegible]
- [illegible]
- Use: [illegible]

End of Exercise 2.4

API for Null-terminated RAM Data

- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]

Use ‘puts’ for data stored in RAM

Usage Example:
putsUSBUSART(output_buffer);
where
char output_buffer[]=(“Microchip”);

Exercise 2.5: Temperature Data

Find function [illegible] in user.c

The program currently outputs temperature data on the physical UART port. Change the program to output data through the USB port.

Follow instructions in [illegible]

[illegible]

[illegible]

End of Exercise 2.5

What is Wrong with This Code?

```
[illegible]
{
    [illegible]
}
```

Did not check:

[illegible]

What is Wrong with This Code?

```
[illegible]
{
    [illegible]

    [illegible]
}
```

Answer:
Blocking function!
Not good for cooperative multitasking.

CDCTxService() in USBTasks() will never get called, and cdc_trf_state will never be updated.

Program will just be stuck in a loop.

Remember, use a state machine!

Post Exercise Analysis

Speed
  - 640 Kbits/s = 80 Kbytes/s
  - Faster than RS-232 (UART)
  - This solution is not possible with a low-speed USB device because it does not have bulk endpoint

No hardware handshakes

Program Memory Usage: ~ 3 KB

Application Note
  - AN956: Migrating Applications to USB from RS-232 UART with Minimal Impact on PC Software

University of Microchip UofM MASTERs 2007

10 minute Break

Agenda

Checking your work station…

Quick USB Review

Demonstration - Microchip USB Solutions

Exercise Group 1: Your turn to play!

Exercise Group 2: CDC RS-232 Emulation APIs

Exercise Group 3: Custom Class using Microchip General Purpose USB Windows Driver

Wrapping up and Q&A

Exercise Group 3

Approximately 140 mins should have lapsed

Exercise Group 3 should take approximately 60 minutes

Please see handouts for instructions

Wrapping Up…

Many choices of device classes

- HID
- CDC – RS-232 Emulation
- General Purpose
- MSD
- Others…

In the end, all you really wanted is to move data from one location to another

Other Resources

Developers Discussion Forum

- http://www.usb.org/phpbb/
- http://forum.microchip.com/tt.aspx?forumid=102

Various application notes from:

- www.microchip.com/usb

USB Complete 3rd Edition
by Jan Axelson

University of Microchip
U of M
MASTERs 2007

Questions?

Thank you!

Addendum

Exercise 2.1: Solution

```
[illegible]
{
    [illegible]
    {
        [illegible]
        {
            [illegible]
        }
    }
}
```

Exercise 2.2: Solution

```
[illegible]
{
    [illegible]
    {
          [illegible]
          {
                    [illegible]
          }
    }
}
```

Exercise 2.3: Solution

```
[illegible]
{
    [illegible]
    {
        [illegible]
                [illegible]
    }
}
```

Exercise 2.4: Solution

```
[illegible]
{
    [illegible]
    {
        [illegible]
        {
                [illegible]
                [illegible]
                [illegible]
                [illegible]
                [illegible]
        }
    }
}
```

Exercise 2.5: Solution

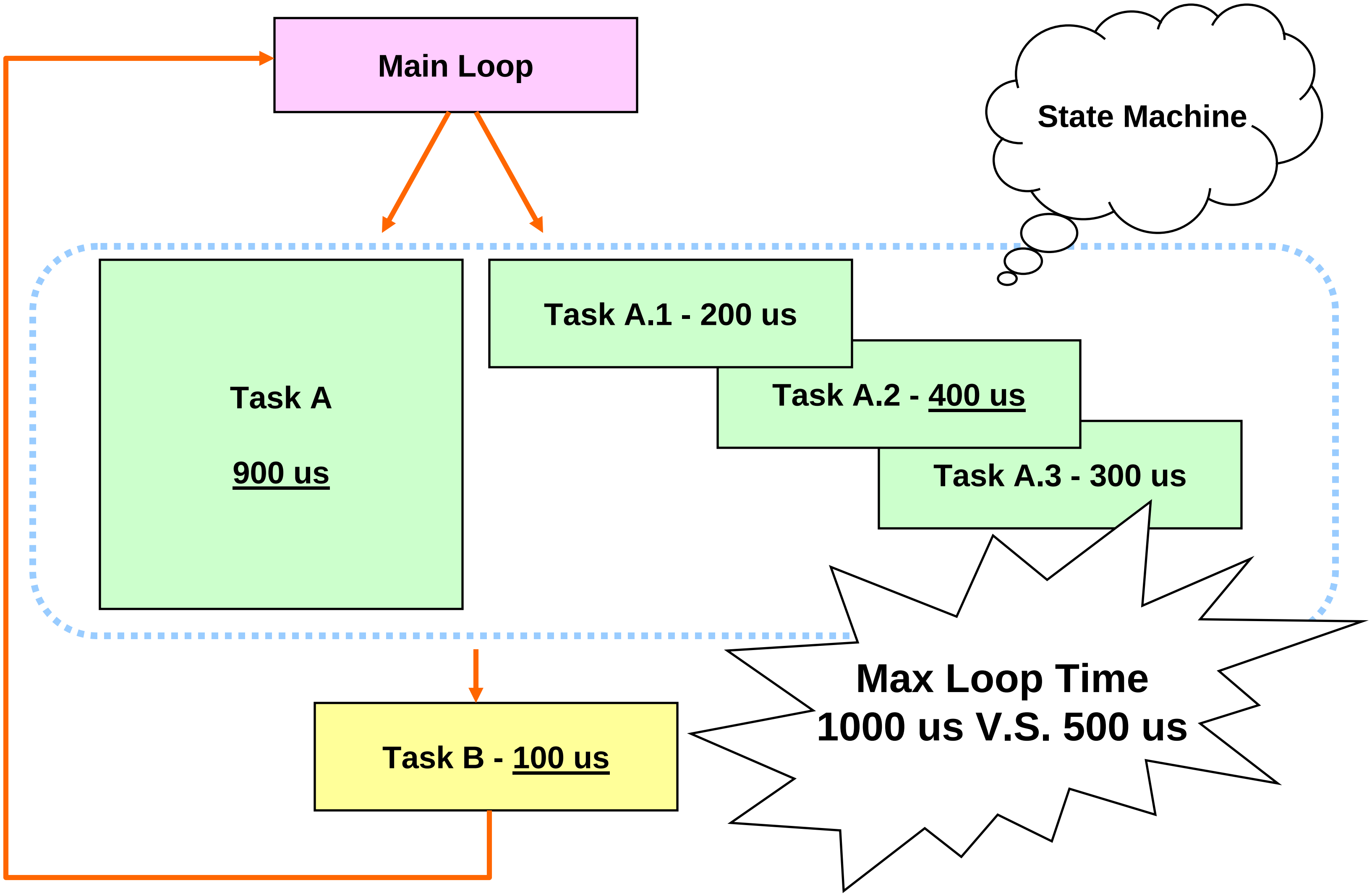
```
[illegible]
{
        [illegible]
[illegible]
{
[illegible]
[illegible]
[illegible]
[illegible]
  [illegible]

[illegible]

[illegible]
{
[illegible]
[illegible]
}

[illegible]
}
[illegible]
[illegible]
[illegible]
```

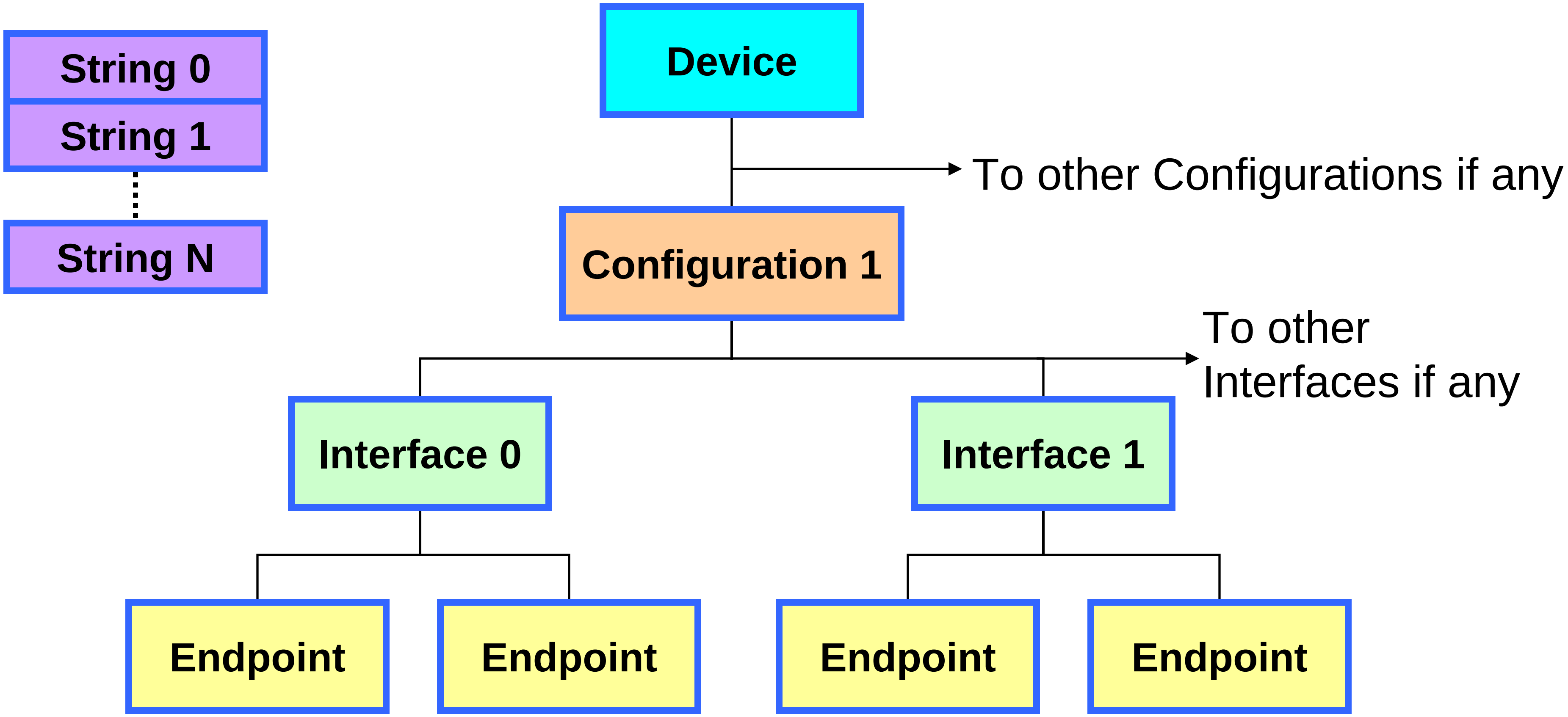
Cooperative Multitasking



PIC18F4550 & PIC16C7X5 USB Features Summary

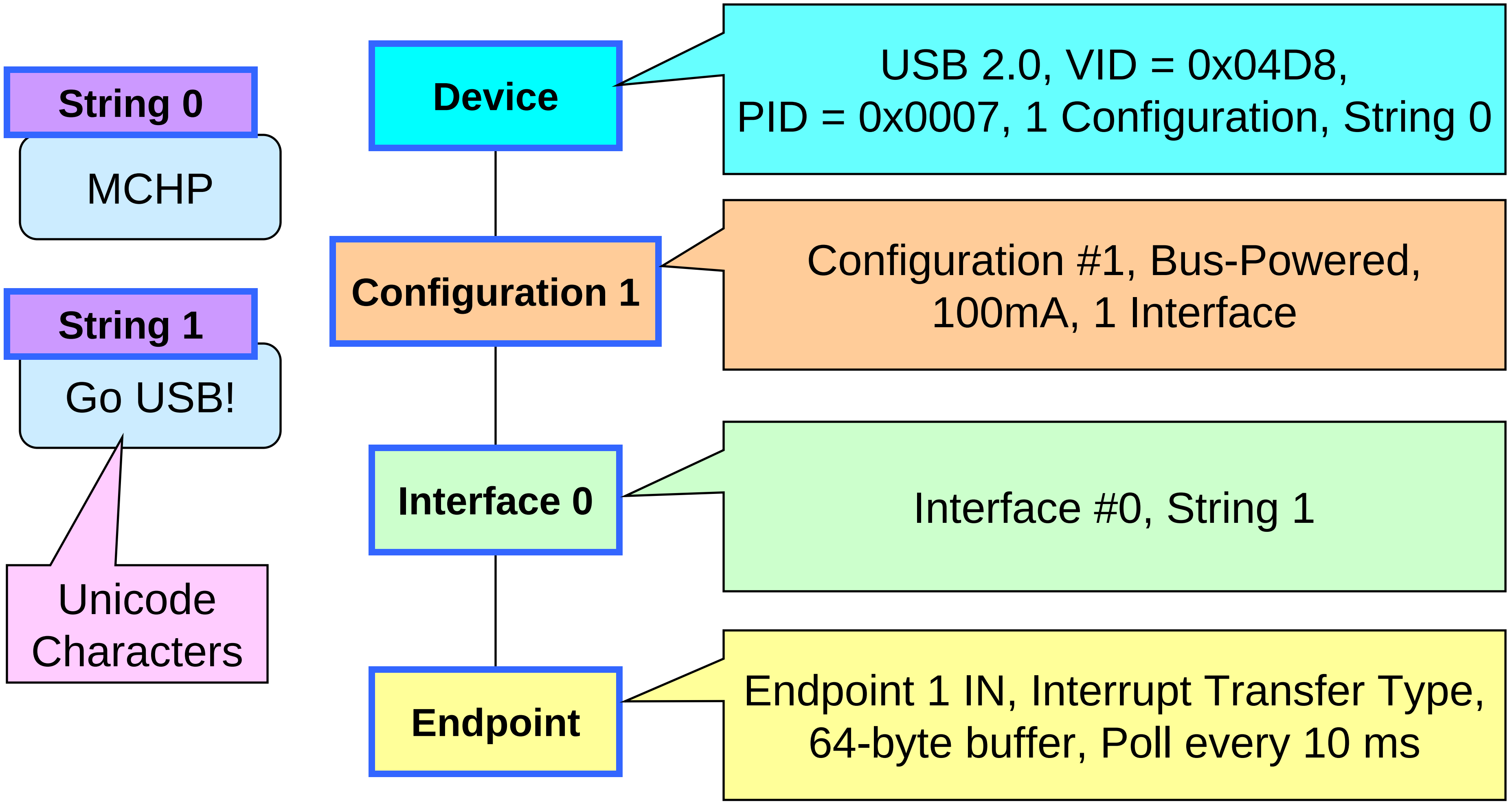
Features	*PIC18F4550*	*PIC16C7X5*
On-chip USB XCVR & Voltage Regulator	Yes	Yes
USB Dual-Port RAM	1-Kbyte	64-byte
USB Speed	Low & Full	Low
On-chip Pull-up Resistors	Yes	No
External XCVR Interface	Yes	No
Streaming Parallel Port	Yes	No
System Clock	Flexible	Fixed
Endpoints	32	6
Flash	Yes	No
USB Transfer Types	Control, Interrupt, Bulk, Isochronous	Control, Interrupt
Pins Compatible?	Yes	

Descriptors



Descriptors must be stored in a non-volatile memory.

Descriptors - Example



String 0

MCHP

String 1

Go USB!

Unicode Characters

Device

Configuration 1

Interface 0

Endpoint

USB 2.0, VID = 0x04D8,
PID = 0x0007, 1 Configuration, String 0

Configuration #1, Bus-Powered,
100mA, 1 Interface

Interface #0, String 1

Endpoint 1 IN, Interrupt Transfer Type,
64-byte buffer, Poll every 10 ms

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.