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

Start Of Frame

SOF

Int EP1

Iso EP1

Iso EP1

SOF

Int EP1

Iso EP1

Iso EP1

SOF

Int EP1

Iso EP1

Bulk EP2

Bulk EP2

1 ms

1 ms

1 ms
Buses & Speeds Comparison

- **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**
- 500 Kb/s

**Parallel Port**
- 1 Mb/s
- 1.5 Mb/s
- 12 Mb/s
- 100 Mb/s 480 Mb/s 1 Gb

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

Host (Tier 1)

USB Host Controller & Root Hub

Hub: Max Chaining = 5

PIC18 USB devices are designed to be peripherals!

Up to 126 peripherals...

Tier 2
- Keyboard
- Speaker

Tier 3
- Logic Analyzer

Tier 4

Tier 5

Tier 6

Tier 7
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**
The “Logical” USB Device

- **Device** (ex. USB Mouse)
- **Configuration** (Active)
- **Interface** (Mouse)
  - Endpoint 0 (Control - OUT)
  - Endpoint 1 (Interrupt - IN)
  - Endpoint 1 (Interrupt - OUT)

**USB Device Driver (HID)**

**USB System Software**

**HID TX/RX Functions**

**Real-World**
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

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

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.
USB Transaction - IN

IN Token Packet

Data Packet
USB Transaction - IN

IN Token Packet

Data Packet

Handshake Packet

Acknowledge - ACK
IN Transaction ACK

**USB Traffic**

- **IN Token Packet** (to EP1 IN)
- **Data Packet**
- **Handshake Packet**

**Serial Interface Engine**

- **SFR EP1 OUT BD** (UOWN, Length, Pointer)
- **SFR EP1 IN BD** (UOWN, Length, Pointer)

**USB PIC® MCU**

- RAM Endpoint 1 OUT
- RAM Endpoint 1 IN

**Keystrokes**

- Sets UOWN = 0
- Update BD

- If UOWN = 1
- SIE Sends Data

© 2007 Microchip Technology Incorporated. All Rights Reserved.
IN Transaction NAK

USB Traffic

IN Token Packet (to EP1 IN)

Handshake Packet

USB PIC® MCU

Serial Interface Engine

IF UOWN = 0

SFR EP1 OUT BD (UOWN, Length, Pointer)

RAM Endpoint 1 OUT

RAM Endpoint 1 IN

SFR EP1 IN BD (UOWN, Length, Pointer)

SIE Sends NAK
OUT/SETUP Transaction ACK

USB Traffic

OUT/SETUP Token Packet
(to EP1 OUT)

Data Packet

Handshake Packet

USB PIC® MCU

Serial Interface Engine

SFR EP1 OUT BD
(UOWN, Length, Pointer)

SFR EP1 IN BD
(UOWN, Length, Pointer)

RAM Endpoint 1 OUT

RAM Endpoint 1 IN

If UOWN = 1

Sets Length, UOWN = 0

SIE Sends ACK

SIE Deposits Data
OUT/SETUP Transaction NAK

USB Traffic

OUT/SETUP Token Packet (to EP1 OUT)

Data Packet

Handshake Packet

SIE Sends NAK

SIE Discards Data

USB PIC® MCU

Serial Interface Engine

If UOWN = 0

RAM Endpoint 1 OUT

RAM Endpoint 1 IN

SFR EP1 OUT BD (UOWN, Length, Pointer)

SFR EP1 IN BD (UOWN, Length, Pointer)
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.

Transaction 1
- 64 Byte Payload

Transaction 2
- 64 Byte Payload

Transaction 3
- 1 Byte Payload
# Data Transfer Types

<table>
<thead>
<tr>
<th>Transfer/Endpoint Type</th>
<th>Polling Interval</th>
<th>Priority</th>
<th>Guarantees</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interrupt</strong></td>
<td>Fixed, Periodic</td>
<td>High</td>
<td>64 Bytes/Period, Data Integrity</td>
</tr>
<tr>
<td><strong>Isochronous</strong></td>
<td>Fixed, Periodic</td>
<td>High</td>
<td>1023 Bytes/Period</td>
</tr>
<tr>
<td><strong>Bulk</strong></td>
<td>Variable, Uses Free Bandwidth</td>
<td>Low</td>
<td>Data Integrity</td>
</tr>
<tr>
<td><strong>Control</strong></td>
<td>Variable</td>
<td>Medium</td>
<td>Some Bandwidth, Data Integrity</td>
</tr>
</tbody>
</table>
Interrupt Transfer Example

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

Traffic to other Endpoints or Idle

Transactions

Frame = 1ms
Bulk Transfer Example

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

Transactions only occur if no higher priority traffic

Frame = 1 ms
Theoretical Maximum Transfer Rate Per Endpoint

<table>
<thead>
<tr>
<th>Transfer Type</th>
<th>Low</th>
<th>Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>24</td>
<td>832</td>
</tr>
<tr>
<td>Interrupt</td>
<td>0.8</td>
<td>64</td>
</tr>
<tr>
<td>Bulk</td>
<td>200</td>
<td>1216</td>
</tr>
<tr>
<td>Iso</td>
<td>1023</td>
<td></td>
</tr>
</tbody>
</table>

KByte/s
Transfer Types - Examples

Mouse
  Control + Interrupt

Printer
  Control + Bulk

Speaker
  Control + Isochronous
USB Device Classes

- Data Glove
- Mouse
- Joystick
- PICkit™ 2 Starter Kit
- Keyboard
- Human Interface Device Class (HID)
- Floppy Drive
- External Hard Drive
- Mass Storage Device Class (MSD)
- Ethernet Adapter
- Modem
- Communication Device Class (CDC)
- Custom Class (Vendor Class)
- ICD2
- 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

Peripheral Device

V_{USB} 3.3 V

1.5 kΩ±5%

+5V
D+
D-
GND
USB Connector

PIC18 USB Microcontroller

Full Speed Identification
D+ line pull-up

Transceiver
Auto-Detection: Low-Speed

Peripheral Device

PIC18 USB Microcontroller

Low Speed Identification D-line pull-up

V_{USB} 3.3 V

1.5 kΩ±5%

+5V
D+
D-
GND

USB Connector

Transceiver
On-chip Pull-up Resistors

Peripheral Device

PIC18 USB Microcontroller

On-chip pull-up resistors!
Controlled by UCFG<UPUEN> & UCFG<FSEN>

V_{USB} 3.3 V

USB Connector

+5V
D+
D-
GND
Endpoint 0 and Enumeration

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

Diagram:
- Control Transfers
- PIC18 USB Microcontroller
- Dual Port/Access RAM
- Endpoint 0 OUT (Control Data)
- Endpoint 0 IN (Control Data)
- Other Endpoints
- Bus Reset
- Set Address
- Get Descriptors
- Set Configuration
- Device Ready
Power Planning

- Max USB suspend current is 0.5/2.5mA

Don’t:
- Power from USB Cable

![Diagram showing 22µF capacitor and V_DD pin with an X mark]
Power Planning

- Max USB suspend current is 0.5/2.5mA
- Do:
  - Power from USB Cable
  - ≤10μF
  - [Diagram of a circuit with a capacitor connected to the USB cable and a checkmark indicating correctness]
If the device is self-powered, you **MUST** 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

Peripheral Device Board

3.3 V Regulator

Bus Pull-Up Resistors

Transceiver

Data RAM

USB RAM

Control Registers

Program Memory

D+/D-

SIE

PIC18F4550
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

PIC18 USB MCU

Transceiver

Data RAM

USB RAM

Control Registers

Program Memory

© 2007 Microchip Technology Incorporated. All Rights Reserved.
390_USB v1.0
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

HPC Explorer – DM183022

Part Number – AC164110
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
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**

- **PIC24FJ256GB1**
  - **On the Go (Master & Slave)**
  - **Low Speed 1.2 Mbps**
  - **Device/Slave/Node**
  - **Full Speed 12 Mbps**

**Ideal for Embedded Market – Serial Port Replacement**

- **Mini-Host (Thumb Drives)**
  - **IN DEVELOPMENT**
- **NOW**
- **IN DEVELOPMENT**

**Other USB PIC microcontrollers**

**Not Supported**

- **Hub**
- **High Speed 480 Mbps**
- **Not Targets for Embedded Market**
USB Device Classes

- Joystick
- Mouse
- External Hard Drive
- Thumb Drive
- Floppy Drive
- Ethernet Adapter
- Modem
- Mass Storage Device Class (MSD)
- Communication Device Class (CDC)
- ICD2
- Custom Class (Vendor Class)
- Many more USB classes....

Beta Support

Human Interface Device Class (HID)

Data Glove

Keyboard

PIckit™ 2 Starter Kit

Joystick

Thumb Drive

Ethernet Adapter

Modem

Many more USB classes....
Mass Storage Application Software Architecture for Thumb Drive Demo

- Application (“PIC-DOS”)
- FAT16
- SCSI Interface
- USB Mass Storage Class (Host)
- USB Host
Thank you !