



USB 设备驱动开发

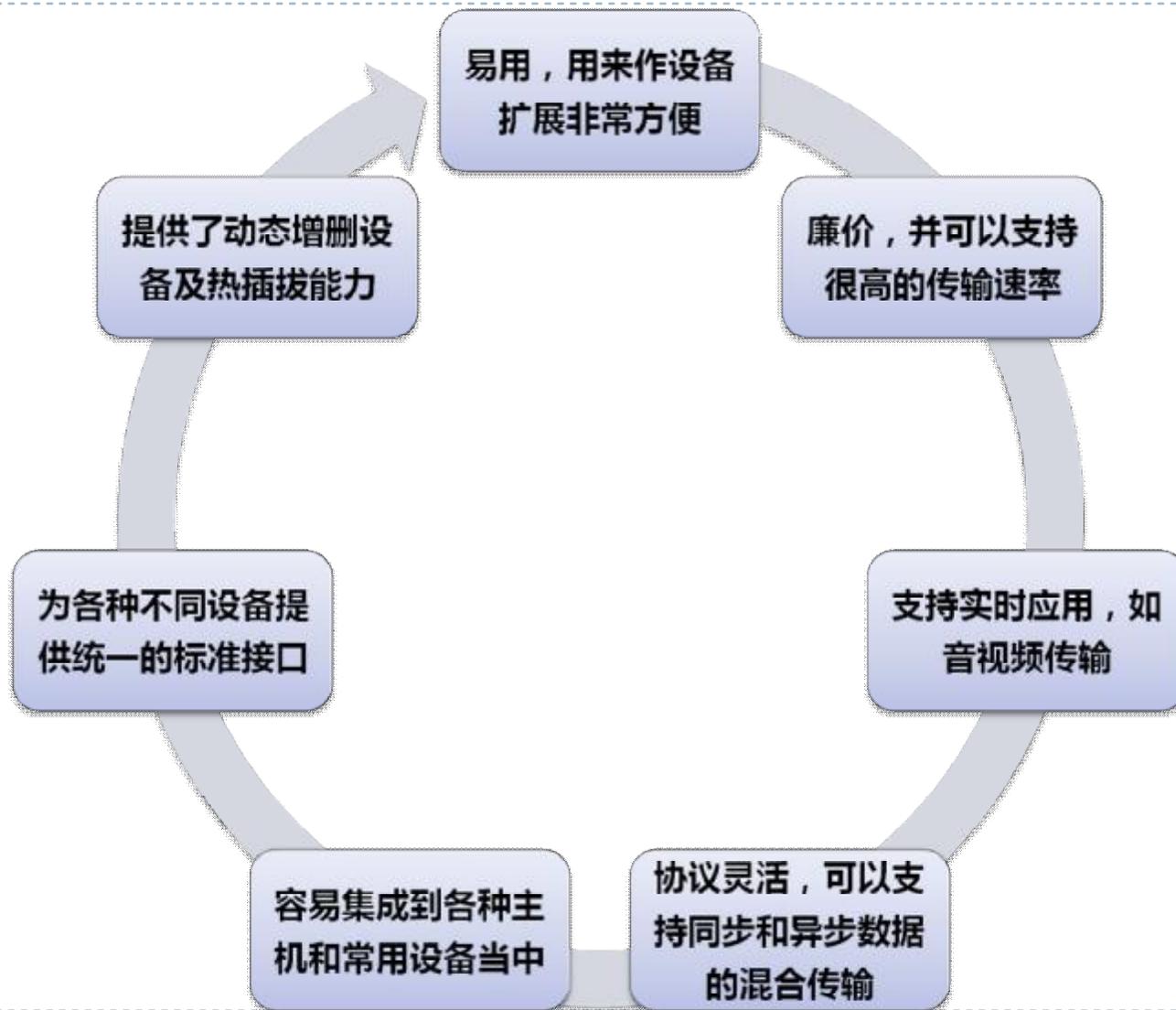
知识点

- } **USB总线介绍**
- } USB拓扑结构
- } USB设备探测
- } USB描述符
- } USB请求格式
- } USB编程

USB总线诞生背景

- } 传统的硬件设配器每次插拔时都不得不重复执行关闭机器、插入板卡、启动机器、安装驱动等繁杂步骤
- } 有没有一种简单易行的接口既能最大限度的节省计算机的软硬件资源，又能方便使用呢？答案是肯定的，那就是今天我们要讨论的**USB**接口

USB总线特点



USB总线局限性

} 带宽的限制

- } USB最高通讯速率为480Mbit/s，但在处理实时数据时还是显得力不从心。实际带宽<300M，而传输普通的640*480*24 (bit) *30 (Fram/s) 时需要的带宽约为220Mbit/s。而IEEE-1394.b可以达到3.2Gbit/s。

} 点对点通讯

- } USB设备之间不能传输数据，而IEEE-1394允许直接通讯

} 距离的限制

- } 小于5m

USB总线规范

- } 1994年，一个由四个行业伙伴(Compaq、Intel、Microsoft 和 NEC)组成的联盟开始制定 USB 协议。该协议最初的目的是将 PC 与电话相连并提供容易扩展和重新配置的 I/O 接口。
- } 1996年 1月，发表了 USB 规范的第一个版本，1998年 9月发表了后续版本(版本 1.1)。这个规范允许 127台设备同时连接到一起，总的通信带宽限制为 12 Mbps。
- } 后来，又有三个成员(Hewlett-Packard、Lucent 和 Philips)加入了这个联盟。
- } 2000年 4月，发表了 USB 规范的 2.0版本，它支持高达 480 Mbps 的传输率。

USB规范的版本

PERFORMANCE

APPLICATIONS

ATTRIBUTES

LOW-SPEED

- Interactive Devices
- 10 – 100 kb/s

Keyboard, Mouse
 Stylus
 Game Peripherals
 Virtual Reality Peripherals

Lowest Cost
 Ease-of-Use
 Dynamic Attach-Detach
 Multiple Peripherals

FULL-SPEED

- Phone, Audio, Compressed Video
- 500 kb/s – 10 Mb/s

POTS
 Broadband
 Audio
 Microphone

Lower Cost
 Ease-of-Use
 Dynamic Attach-Detach
 Multiple Peripherals
 Guaranteed Bandwidth
 Guaranteed Latency

HIGH-SPEED

- Video, Storage
- 25 – 400 Mb/s

Video
 Storage
 Imaging
 Broadband

Low Cost
 Ease-of-Use
 Dynamic Attach-Detach
 Multiple Peripherals
 Guaranteed Bandwidth
 Guaranteed Latency
 High Bandwidth

USB1.1规范

- } USB1.1规范支持低速（1.5Mb/s）和全速（12Mb/s）两种不同速率的数据传输和4种不同类型的数据传输方式：
 - } 控制传输（CONTROL TRANSFER）
 - } 中断传输（INTERRUPT TRANSFER）
 - } 批量传输（BULK TRANSFER）
 - } 等时传输（ISOCRONOUS TRANSFER）

USB2.0规范

- } USB2.0在兼容USB1.1低速（1.5Mb/s）和全速（12Mb/s）数据传输基础上，支持高速（480Mb/s）数据传输。
- } 对于USB2.0规范，同样支持控制传输、中断传输、批量传输和等时传输4种类型的数据传输方式。
- } 在物理结构和拓扑结构上，USB2.0与USB1.1也是完全相同的。

USB OTG规范

- } USB OTG规范是作为对USB2.0规范的补充而出现的，其目的是为了**满足便携式设备对USB接口性能的需求**。
- } 根据USB OTG规范，一个USB接口可同时具有**USB主机和USB设备两种功能**，根据与其连接的其他设备属性，**USB OTG接口会自动转换成为适合USB总线需求的接口类型**。

USB设备类（CLASS）规范



- } USB-IF（USB应用厂商论坛）将可能的物理设备划分成不同的类型，每种类型的设备具有相类似的通信方式。
- } 针对不同类型的设备，USB-IF发布了相关的USB设备类规范，比如：
 - } USB人机交互类（HID CLASS）规范
 - } USB通信类（CDC CLASS）规范
 - } USB大容量存储类（MASS STORAGE CLASS）规范

USB HOST规范

- } 关于USB HOST接口，在符合USB规范的基础上，不同的厂商开发的USB HOST器件可能有着不同的结构特性。当前流行的USB HOST规范有：
 - } OHCI (OPEN HOST CONTROL INTERFACE)
 - } UHCI (UNIVERSAL HOST CONTROL INTERFACE)
 - } EHCI (ENHANCED HOST CONTROL INTERFACE)

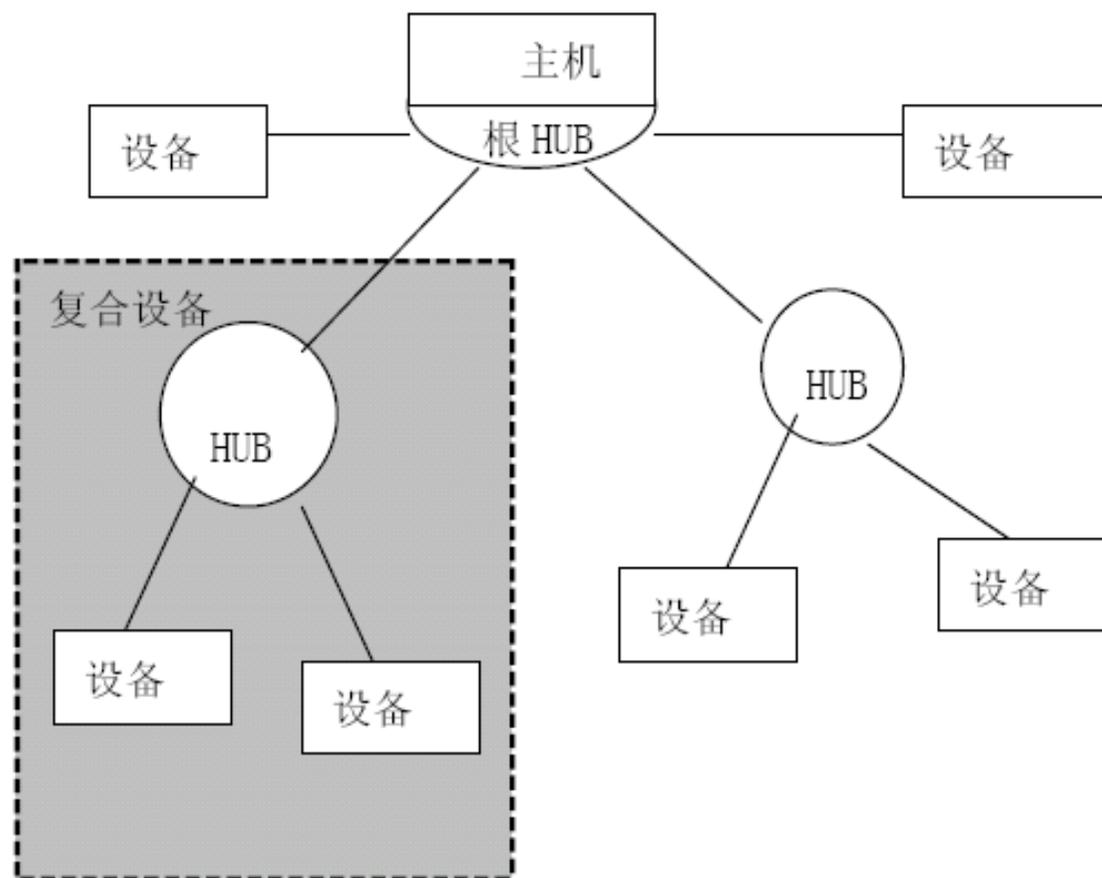
EHCI与OHCI

- } ehci-hcd模块支持的是USB2.0控制器的高速模式，它本身并不支持全速或低速模式
- } ohci-hcd或uhci-hcd模块提供对USB1.1设备的支持
- } 如果我们只配置了EHCI，就没有办法使用usb的鼠标键盘

知识点

- } USB总线介绍
- } **USB拓扑结构**
- } USB 设备探测
- } USB描述符
- } USB请求格式
- } USB编程

USB拓扑结构



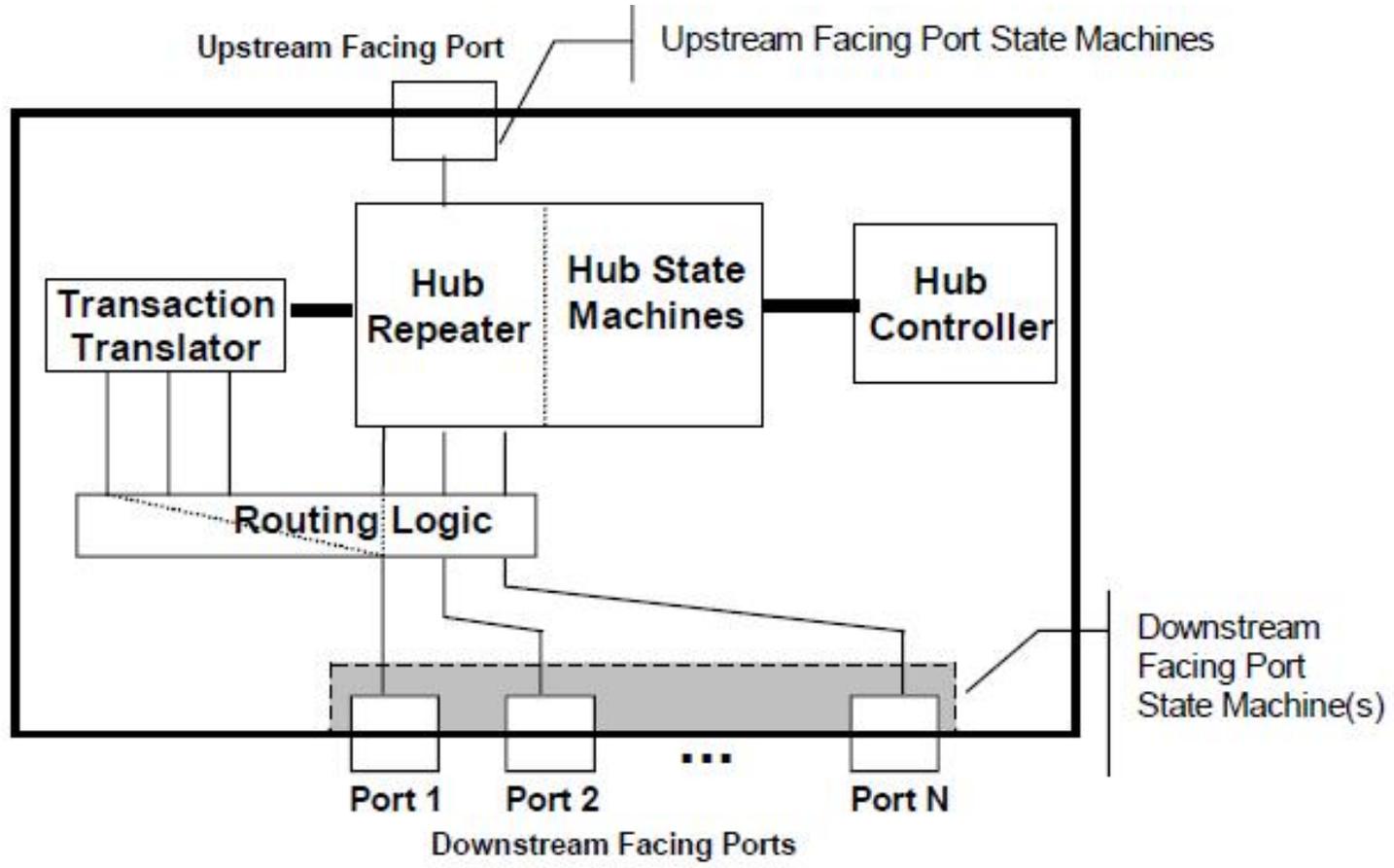
主机控制器介绍

- } 在和外围设备通讯时主机控制器始终占主动地位，其主要作用有：
 - } 检测设备
 - } 管理数据流
 - } 错误检查
 - } 提供电源

HUB功能介绍

- } USB HUB主要功能有：
 - } 管理主机与设备的连接
 - } 电源管理
 - } 设备接入与断开检测
 - } 总线错误的检测与恢复
 - } 对高速、全速、低速设备的支持

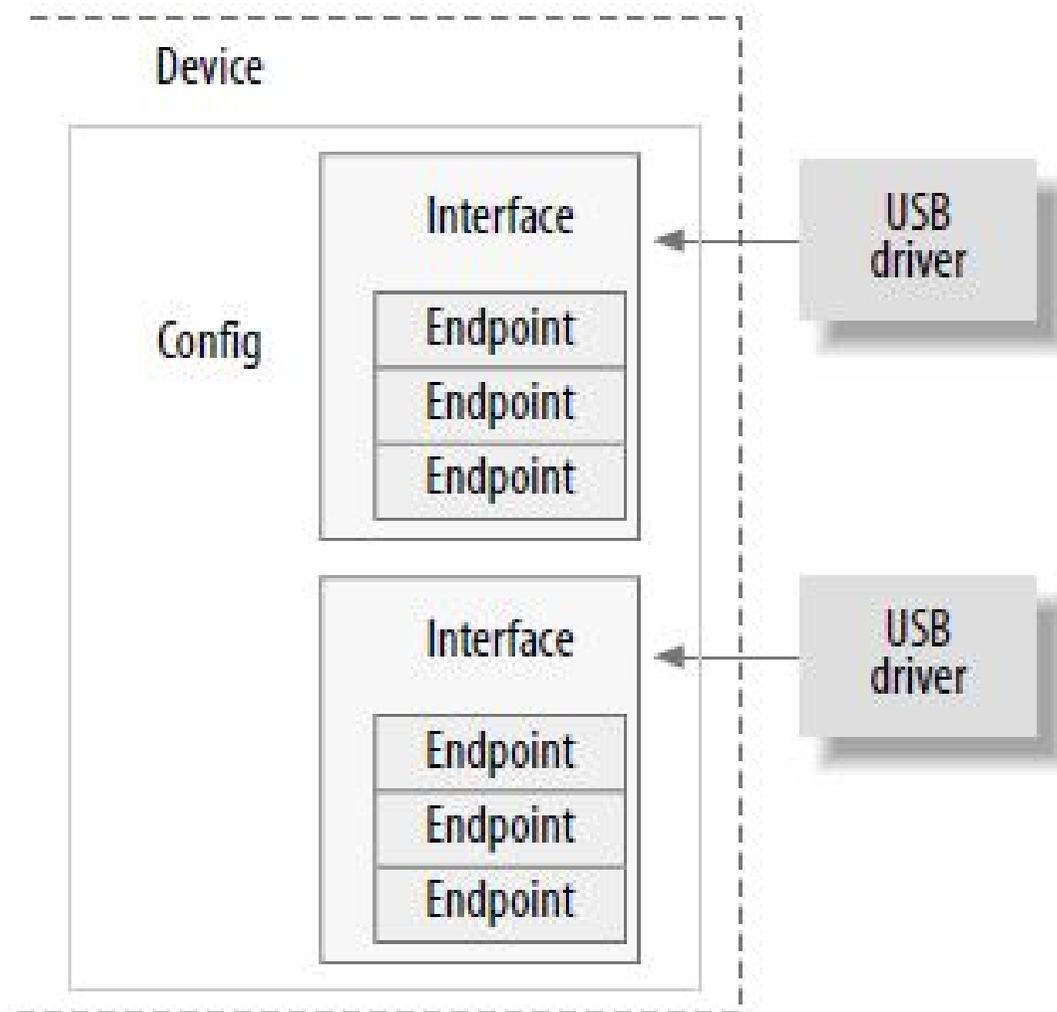
USB hub结构



USB设备基本概念

- } USB规范中规定的标准概念由Linux USB core来实现处理
- } 端点endpoints
 - } 端点是USB总线传输最基本的概念，一个端点可以单方向传输数据。可以把端点看作是一个单方向的管道。端点有4种类型
- } 接口interfaces
 - } 若干端点可以捆绑起来，成为一个端口。端口可以作为完整的逻辑设备连接，例如鼠标设备、键盘设备。
 - } 需要指出的是，一个硬件设备可能包含多个逻辑设备
 - } 接口可以有多个预设值，用来指定不同的参数
- } 配置configurations
 - } 接口捆绑起来成为配置。一个USB设备可以在不同的配置之间切换，一次只能激活一个配置

USB设备基本概念示意图



USB端点分类

- } **USB 总线中的通信可以使用下面四种数据传输类型中的任意一种：**
 - } **控制传输：**这些是一些短的数据包，用于设备控制和配置，特别是在设备附加到主机上时。
 - } **批量传输：**这些是数量相对大的数据包。像扫描仪或者SCSI适配器这样的设备使用这种传输类型。
 - } **中断传输：**这些是定期轮询的数据包。主控制器会以特定的间隔自动发出一个中断。
 - } **等时传输：**这些是实时的数据流，它们对带宽的要求高于可靠性要求。音频和视频设备一般使用这种传输类型。

知识点

- } USB总线介绍
- } USB拓扑结构
- } **USB 设备探测**
- } USB描述符
- } USB请求格式
- } USB编程

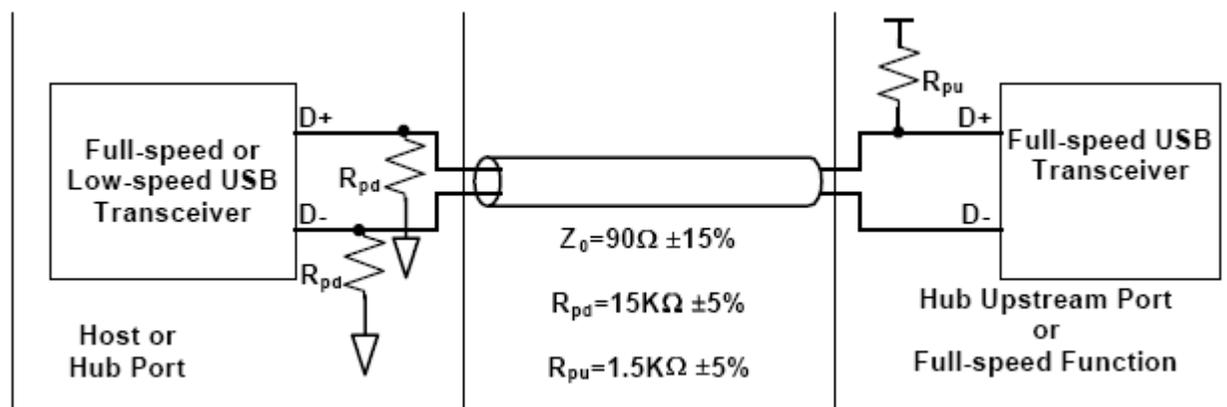
USB Hub对端口监测

} USB协议规定

} 低速外设在D-端并联一个1.5K的接地电阻

} 全速外设在D+端并联一个1.5K的接地电阻

} 当USB设备插入某一个HUB端口时，硬件检测D+和D-信号线的电平就可以很容易判断器件的种类



USB 设备状态

连接	加电	缺省	编址	配置	挂起	说 明
不	— —	— —	— —	— —	— —	设备尚未连接至接口. 其他特性无关
是	不	— —	— —	— —	— —	设备已连接至接口, 但未加电. 其他特性无关.
是	是	不	— —	— —	— —	设备已连接至接口, 并且已加电. 但尚未被复位.
是	是	是	不	— —	— —	设备已连接至接口, 已加电. 并被复位. 但尚未分配地址. 设备在缺省地址处可寻址.
是	是	是	是	不	— —	设备已连接至接口, 已加电. 并被复位. 且分配了唯一地址. 尚未被配置.
是	是	是	是	是	不	设备已连接至接口, 已加电. 并被复位. 且分配了唯一地址, 并被配置. 设备功能可被使用.
是	是	— —	— —	— —	是	设备在至少 3 毫秒以内探测不到总线活动, 自动进入挂起. 设备功能不可用.

知识点

- } USB总线介绍
- } USB拓扑结构
- } USB 设备探测
- } **USB描述符**
- } USB请求格式
- } USB编程

USB描述符介绍



✓ **USB**描述符记录了关于设备的各种信息，**USB**设备使用这些描述符来向主机报告它的属性。描述符具有一定的数据结构。在主机使用控制传输请求时，设备就会将这些信息以标准格式发送给主机，从而使主机控制器获取诸如设备及供应商**ID**，设备通讯能力等重要信息。

USB描述符类型



描述符类型	数值
设备 (<i>Device</i>)	0x01
配置 (<i>Configuration</i>)	0x02
字符串 (<i>String</i>)	0x03
接口 (<i>Interface</i>)	0x04
端点 (<i>Endpoint</i>)	0x05
类(Class)描述符形态	数值
人机接口设备 (<i>HID</i>)	0x21
报告 (<i>Report</i>)	0x22
实体 (<i>Physical</i>)	0x23

知识点

- } USB总线介绍
- } USB拓扑结构
- } USB 设备探测
- } USB描述符
- } **USB请求格式**
- } USB编程

USB设备请求的概念

- } 在USB通讯中，主机有绝对的主控权，主机和设备之间的通讯遵循某种特定的命令格式，以达到通讯的目的，而这个命令的格式就是“设备请求”。设备请求存在于控制传输的setup阶段。它是一个8个字节的数据封装包

USB的标准设备请求

Offset	Field	Size	Value	Description
0	<i>bmRequestType</i>	1	Bitmap	Characteristics of request: D7: Data transfer direction 0 = Host-to-device 1 = Device-to-host D6...5: Type 0 = Standard 1 = Class 2 = Vendor 3 = Reserved D4...0: Recipient 0 = Device 1 = Interface 2 = Endpoint 3 = Other 4...31 = Reserved
1	<i>bRequest</i>	1	Value	Specific request (refer to Table 9-3)
2	<i>wValue</i>	2	Value	Word-sized field that varies according to request
4	<i>wIndex</i>	2	Index or Offset	Word-sized field that varies according to request; typically used to pass an index or offset
6	<i>wLength</i>	2	Count	Number of bytes to transfer if there is a Data stage

部分标准设备请求



要求类型 BmRequestType	要求 Request	数值 Wvalue	指数 Windex	长度 Wlength
00000000B	Get_Configuration (08H)	0	0	1
00000000B	Get_Descriptor (06H)	描述符类型	0	描述符长度
00000000B	Get_Interface (0AH)	0	接口	0
00000000B	Set_Address (05H)	新的设备地址	0	2
00000000B	Set_Configuration (09H)	设定值	0	0
00000000B	Set_Descriptor (07H)	描述符类型	0	描述符长度
00000000B	Set_Interface (0BH)	切换的设定	接口	0

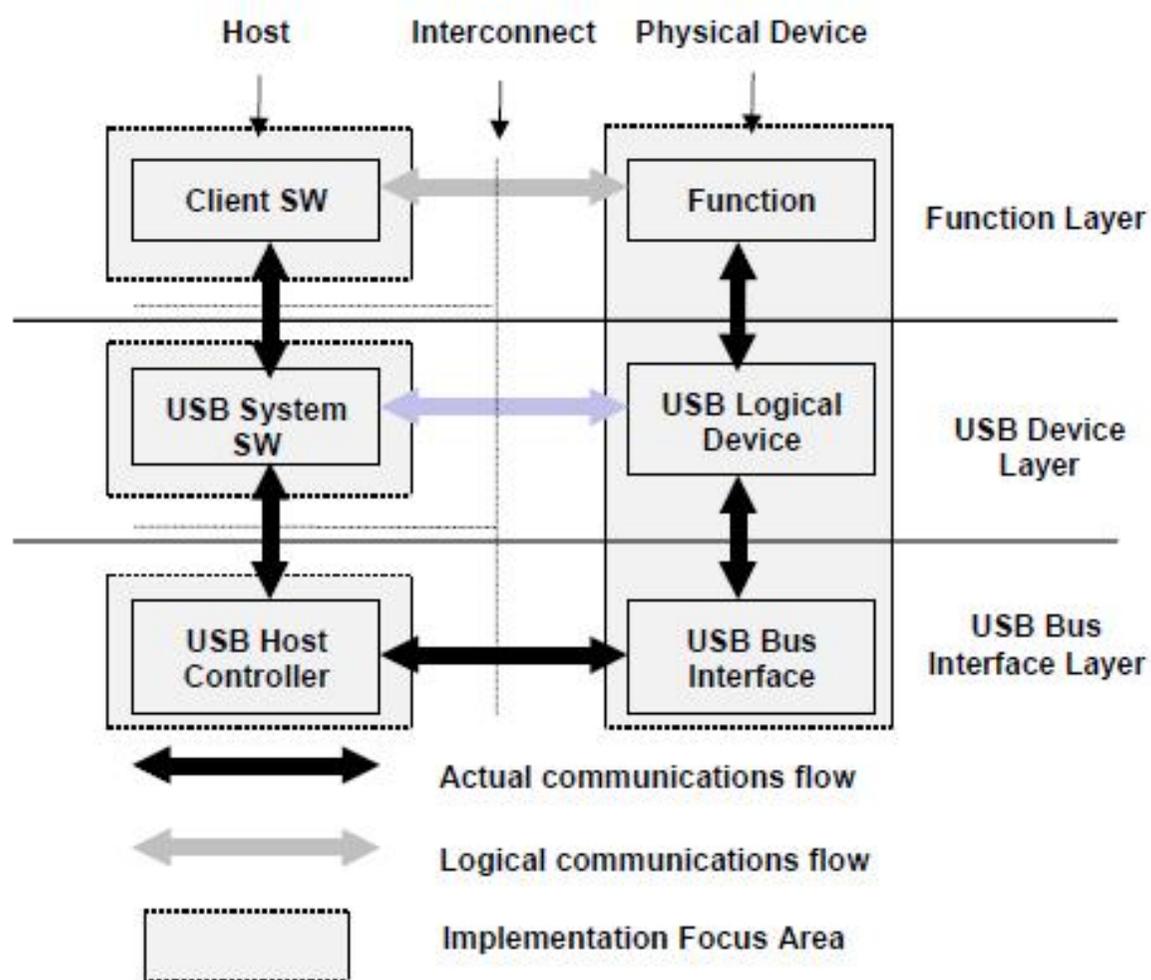
设备请求举例

- } 使用Get_Descriptor请求“设备描述符”，请求码？
 - } 第1个字节： BmRequestType=80
 - } 第2个字节： Request=06
 - } 第3、4个字节： 00、01（设备描述符）
 - } 第5、6个字节： 00、00
 - } 第7、8个字节： 00、12（18个字节）

知识点

- } USB总线介绍
- } USB拓扑结构
- } USB 设备探测
- } USB描述符
- } USB请求格式
- } **USB编程**

USB层次化连接示意图



USB编程开发要点

- } 主机控制器驱动
- } HUB驱动
- } **设备控制器编程**
- } **主机端设备驱动**
- } **设备端固件程序编写**

USB驱动高级研修班介绍（1）

} 1. LINUX下USB驱动开发基础

1.1 USB规范介绍

1.2 USB主机控制器

1.3 USB HUB

1.4 USB设备状态

1.5 USB描述符

1.6 USB请求

1.7 USB通讯数据格式

1.8 S3C2410 USB device控制器

1.9 编写usbtransfer固件程序

USB驱动高级研修班介绍（2）

} 2. LINUX下USB驱动高级开发

2.1 Linux USB驱动层次

2.2 Linux USB主机控制器驱动介绍

2.3 Linux USB HUB驱动介绍

2.4 Linux USB 设备驱动程序

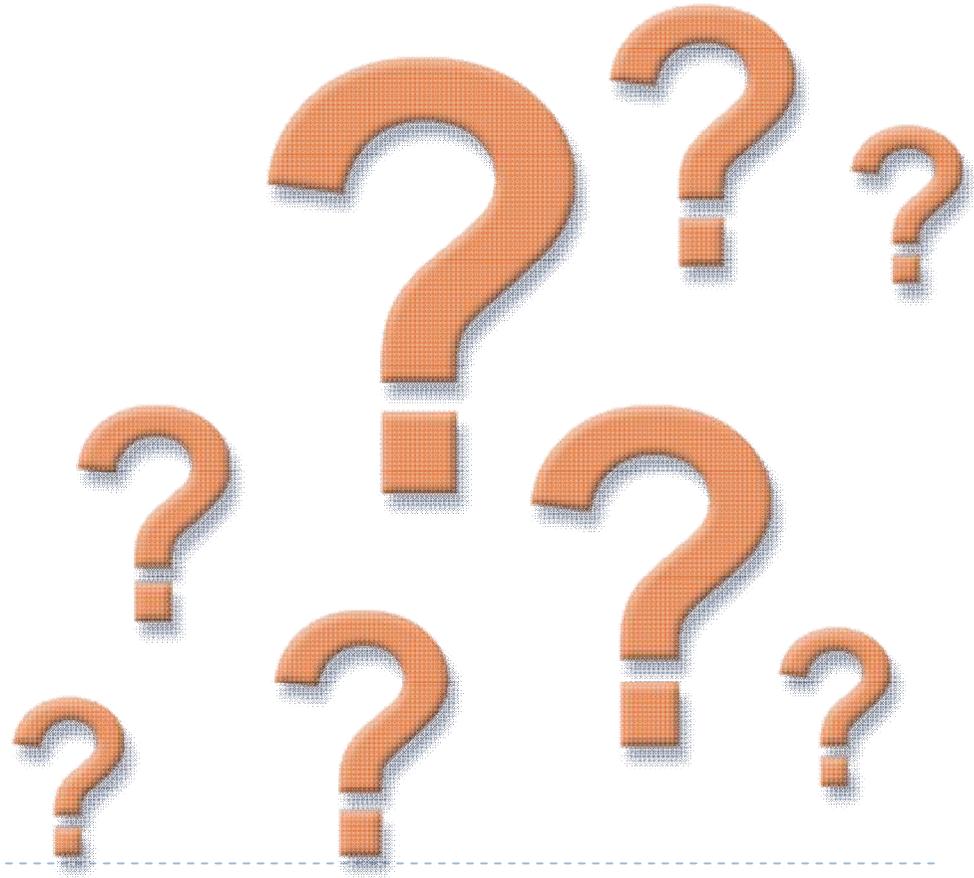
2.5编写usbtransfer设备驱动程序

2.6编写usbtransfer设备应用程序

有什么问题吗？



Q&A



谢谢!

