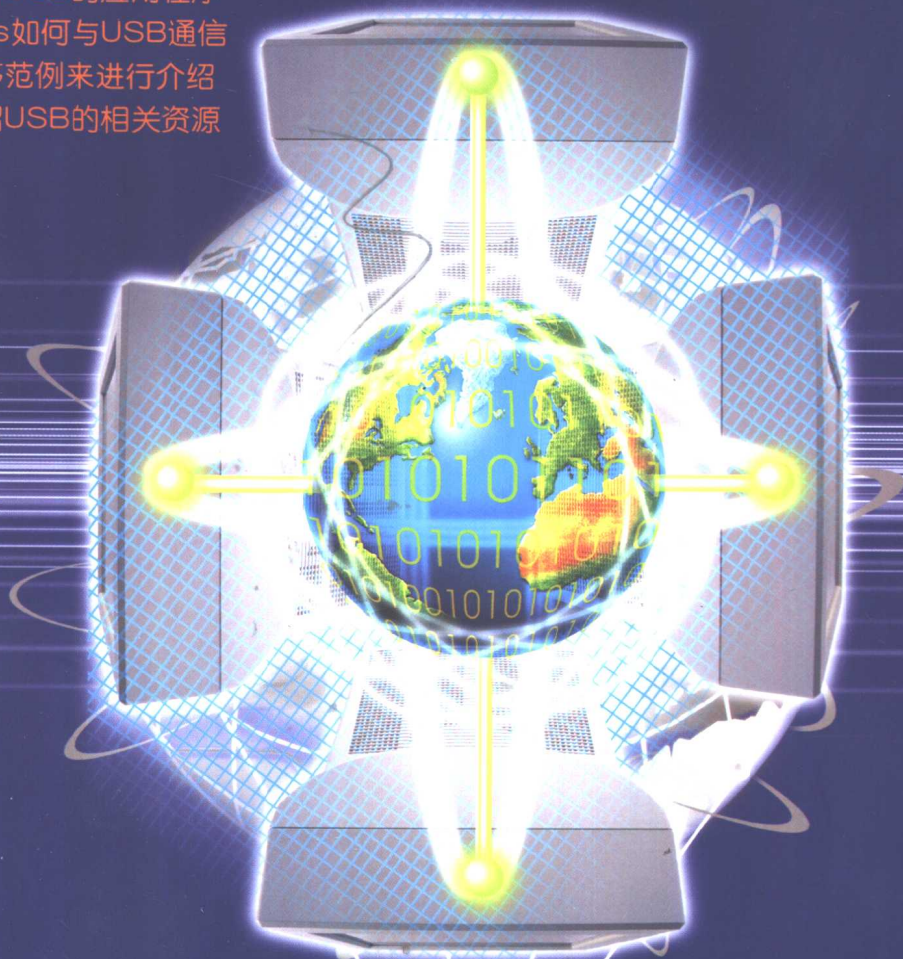


# USB 2.0

## 硬件设计

介绍最新的USB 2.0规范  
如何设计USB的设备  
如何设计USB的应用程序  
Windows如何与USB通信  
使用程序范例来进行介绍  
详尽介绍USB的相关资源

萧世文 编著



清华大学出版社

TP360.6

8

# USB 2.0 硬件设计

萧世文 编著

清华大学出版社

## 内 容 简 介

USB 是外围设备与计算机进行连接快速且有弹性的新式接口。本书介绍最新的 USB 2.0 标准, 从 USB 的基础开始介绍, 包括实际的硬件与软件设计, 并且使用汇编语言、Visual C++、Visual Basic 等来示范程序代码的编写。另外还有各种 USB 芯片介绍, 帮助读者快速设计自己的 USB 设备。

本书共 14 章和 1 个附录, 内容包括: USB 简介、USB 传输、检测设备、控制传输、USB 控制器芯片、Cypress enCore 控制器芯片、驱动程序、设备类别、设备测试、电源管理与信号编码、电路接口、集线器、HID 设备、HID 程序设计以及 USB 参考资料等。

本书适合从事 USB 的硬件、软件设计的程序员和其他相关技术人员使用或参考。

本书繁体字版书名为《USB 2.0 硬件设计》, 由文魁资讯股份有限公司出版, 版权属萧世文所有。本书简体字中文版由文魁资讯股份有限公司授权清华大学出版社独家出版。未经本书原版出版者和本书出版者书面许可, 任何单位和个人均不得以任何形式或任何手段复制或传播本书的部分或全部内容。

北京市版权局著作权合同登记号: 图字 01-2002-2551 号

**版权所有, 翻印必究。**

**本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。**

图书在版编目(CIP)数据

USB 2.0 硬件设计/萧世文编著.—北京: 清华大学出版社, 2002

ISBN 7-302-05937-3

I.U... II.萧... III.电子计算机—接口—外部设备—设计 IV.TP334

中国版本图书馆 CIP 数据核字(2002)第 074929

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 桑任松

印 刷 者: 北京市清华园胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 21.75 字数: 525 千字

版 次: 2002 年 10 月第 1 版 2003 年 1 月第 2 次印刷

书 号: ISBN 7-302-05937-3/TP·3528

印 数: 4001~6500

定 价: 35.00 元

# 前 言

USB(Universal Serial Bus, 通用串行总线)是用来连接外围设备与计算机的新式接口,每一台新式的 PC 都至少有两个 USB 连接端口。标准的外围设备(例如鼠标,键盘,调制解调器,外部磁盘驱动器,以及游戏控制设备等)都已经改用 USB 接口来连接到 PC。USB 的接头与连接端口不但体积小,而且可以利用集线器(hub)让多个 USB 设备同时连接到一个连接端口。

安装 USB 设备不需要用户自己设定硬件与软件的选项,您可以在计算机运行时任意连接或是移除设备,操作系统会自动侦测有无 USB 设备的连接。Windows 包含有标准外围设备的 USB 驱动程序,并且提供 API 函数让用户与设备通信。

USB 设备可以使用 4 种传输类型与 3 种传输速度。USB 的传输类型包括控制、中断、批量与实时。USB 的传输速度包括低速、全速与高速。低速最快可以每秒传输 1.5 Mb 的数据,全速最快可以每秒传输 12 Mb 的数据,高速最快可以每秒传输 480 Mb 的数据。

要设计 USB 设备的硬件与软件,需要了解 USB 的工作原理,以及 PC 机的操作系统如何与 USB 设备通信。除此之外,选择适合的 USB 芯片与工具会帮助用户简化设计工作。在本书中除了介绍 USB 的基本原理外,另外也有 USB 芯片与工具的介绍,并且以实际的范例程序代码,来帮助您快速地进入 USB 设计的领域。

本书共分成 14 章:

第 1 章“USB 简介”介绍 USB 的发展史,优点与缺点,以及与其他接口(例如 RS-232, IEEE-1394 等)的比较。另外,还介绍建立 USB 接口所需的 PC 与外围设备的软硬件规格,如何下载 USB 的规范与工具软件等。

第 2 章“USB 传输”介绍 USB 的 4 种传输类型、各种传输类型的结构、数据格式、传输速度以及适用的设备种类。USB 的传输分成一笔笔的事务,每一笔事务由信息包组成。USB 的错误检验,则是使用 CRC 值与数据交替位来确保传输的数据没有错误。

第 3 章“检测设备”介绍 Windows 操作系统在 USB 设备连接时,如何自动检测设备以了解设备的功能。Windows 传送各种要求给设备,设备则是依据要求的种类来反应。

第 4 章“控制传输”介绍 USB 最基本的传输类型。每一个 USB 设备都必须支持控制传输,控制传输让 PC 设置 USB 设备的配置,或是用来交换数据。控制传输包含有 3 个阶段:设置、数据与状态。每个阶段会传送 3 种信息包:令牌、数据与联络。本章还介绍 USB 的标准要求,设备类别的要求,以及厂商自定的要求。

第 5 章“USB 控制器芯片”介绍 USB 控制器的内部组成组件及其结构。书中以几个较受欢迎的 USB 芯片来做实际的解说。

第 6 章“Cypress enCoRe 控制器芯片”以 Cypress 公司所出品的 enCoRe 系列芯片为范例,介绍此芯片的内部结构与电路。书中使用汇编语言来示范程序设计的实例,以及介绍辅助设计用的硬件与软件工具。

第 7 章“驱动程序”介绍 Windows 的驱动程序模型。撰写客户化的驱动程序所需的注意事项,以及辅助工具软件。Windows 使用.inf 文件来建立 USB 设备与驱动程序的信息,

本章介绍.inf文件的格式与使用技巧。

第8章“设备类别”介绍设备类别的定义。USB设备可以归类成各种设备类别，例如键盘、鼠标与游戏杆等属于人机接口设备(Human Interface Device, HID)类别。非标准类型的设备，也可以改用USB接口来连接到PC。

第9章“设备测试”介绍USB设备的测试工具，测试仪器与测试的规范。当USB设备通过测试后，就可以挂上USB所认可的标志。

第10章“电源管理与信号编码”介绍USB设备(包括集线器)如何使用电源。USB设备可以通过总线来取得电源，也可以自己提供电源。在数据传输的过程中，总线会改变成各种状态。当设备不用时，会自动进入节能状态。本章还详细介绍了USB的数据传输使用的编码格式。

第11章“电路接口”介绍USB设备的电路接口，以及USB电缆与连接器的规格。USB缆线使用双绞线来减少噪声的干扰。

第12章“集线器”介绍集线器的内部构造及其功能。集线器可以让多个USB设备，同时连接到PC机上。本章还介绍集线器的描述符与要求。

第13章“HID设备”介绍HID设备的软硬件要求、描述符以及特定要求。HID设备使用报表来传输数据。本章详尽介绍报表的格式与内容。

第14章“HID程序设计”介绍Windows用来与HID设备通信的API函数。书中同时以Visual C++与Visual Basic程序代码做范例，来显示API函数的实际应用。

附录A“USB参考资料”介绍USB的规范与工具软件，USB芯片相关厂商等的网址资料。

# 目 录

第 1 章 USB 简介.....	1	1.9.2 开发工具.....	20
1.1 USB 的发展史.....	2	1.10 开发计划的步骤.....	21
1.1.1 规范的版本.....	3	1.10.1 初步决定.....	21
1.1.2 USB 2.0.....	4	1.10.2 检测设备.....	21
1.1.3 IEEE-1394.....	4	1.10.3 交换数据.....	21
1.2 USB 的优点.....	4	第 2 章 USB 传输.....	22
1.2.1 容易使用.....	4	2.1 传输的基础.....	22
1.2.2 传输速度快.....	5	2.1.1 配置.....	22
1.2.3 低价位.....	6	2.1.2 应用程序.....	22
1.2.4 低能耗.....	6	2.1.3 数据的管理.....	23
1.2.5 稳定性.....	6	2.1.4 总线速度.....	24
1.2.6 操作系统的支持.....	7	2.2 传输的要件.....	24
1.2.7 外围设备的支持.....	7	2.2.1 设备的端点.....	25
1.2.8 有弹性.....	7	2.2.2 管道.....	26
1.2.9 USB 实施者论坛.....	8	2.2.3 传输类型.....	26
1.3 USB 的缺点.....	8	2.2.4 消息管道与流管道.....	27
1.3.1 缺乏对旧硬件的支持.....	8	2.2.5 传输的初始化.....	28
1.3.2 点对点的通信.....	9	2.3 事务.....	28
1.3.3 速度的限制.....	9	2.3.1 事务的相位.....	30
1.3.4 距离的限制.....	9	2.3.2 信息包的串行化.....	32
1.3.5 硬件的错误或故障.....	9	2.3.3 时间的限制.....	32
1.3.6 协议的复杂性.....	9	2.3.4 分割事务.....	32
1.3.7 版权费.....	9	2.4 联络信号.....	35
1.4 与其他接口的比较.....	10	2.4.1 ACK.....	35
1.5 PC 的软硬件要求.....	11	2.4.2 NAK.....	36
1.5.1 主机控制器.....	11	2.4.3 STALL.....	36
1.5.2 操作系统.....	11	2.4.4 NYET.....	36
1.5.3 组件.....	12	2.4.5 ERR.....	37
1.5.4 信道.....	14	2.4.6 没有响应.....	37
1.5.5 连接端口.....	15	2.4.7 控制传输的状态.....	37
1.6 主机的功能.....	16	2.5 错误的检验.....	38
1.7 外围设备的功能.....	17	2.5.1 错误校验位.....	38
1.8 传输速度.....	19	2.5.2 数据交替位.....	38
1.9 开发的流程.....	20	2.6 控制传输.....	39
1.9.1 所需组件.....	20		

2.6.1	结构	40	3.2.8	字符串描述符	72
2.6.2	信息包大小	42	3.3	USB 2.0 兼容设备的描述符	75
2.6.3	传输速度	43	<b>第 4 章</b>	<b>控制传输</b>	<b>77</b>
2.6.4	检测错误	44	4.1	设置阶段	77
2.7	批量传输	45	4.1.1	令牌信息包	78
2.7.1	结构	45	4.1.2	数据信息包	79
2.7.2	信息包大小	46	4.1.3	联络信息包	81
2.7.3	传输速度	46	4.2	数据阶段	82
2.7.4	检测错误	48	4.2.1	令牌信息包	83
2.8	中断传输	48	4.2.2	数据信息包	83
2.8.1	结构	48	4.2.3	联络信息包	83
2.8.2	信息包大小	48	4.3	状态阶段	83
2.8.3	传输速度	48	4.3.1	令牌信息包	84
2.8.4	检测错误	50	4.3.2	数据信息包	84
2.9	实时传输	51	4.3.3	联络信息包	84
2.9.1	结构	51	4.4	检测错误	85
2.9.2	信息包大小	52	4.5	要求	85
2.9.3	传输速度	52	4.5.1	Clear_Feature	86
2.9.4	检测错误	54	4.5.2	Get_Configuration	87
2.10	时间关键的传输	54	4.5.3	Get_Descriptor	87
2.10.1	总线的带宽	54	4.5.4	Get_Interface	88
2.10.2	设备的能力	54	4.5.5	Get_Status	89
2.10.3	主机软件的能力	55	4.5.6	Set_Address	90
2.10.4	Windows 的延迟	55	4.5.7	Set_Configuration	90
<b>第 3 章</b>	<b>检测设备</b>	<b>56</b>	4.5.8	Set_Descriptor	91
3.1	检测设备的过程	56	4.5.9	Set_Feature	91
3.1.1	检测的步骤	57	4.5.10	Set_Interface	92
3.1.2	检测集线器	60	4.5.11	Synch_Frame	93
3.1.3	移除设备	60	4.5.12	类别特定的要求	93
3.2	描述符	61	4.5.12	厂商特定的要求	94
3.2.1	类型	61	<b>第 5 章</b>	<b>USB 控制器芯片</b>	<b>95</b>
3.2.2	设备描述符	62	5.1	USB 控制器的组件	95
3.2.3	Device_Qualifier 描述符	64	5.1.1	USB 连接端口	95
3.2.4	配置描述符	66	5.1.2	数据缓冲区	96
3.2.5	Other_Speed_Configuration 描述符	67	5.1.3	CPU	96
3.2.6	接口描述符	68	5.1.4	程序内存	96
3.2.7	端点描述符	70	5.1.5	数据存储器	97
			5.1.6	寄存器	97

5.1.7 其他接口 .....	97	6.3.1 C 语言的优点.....	124
5.1.8 其他特性 .....	98	6.3.2 C 编译器的用法.....	125
5.2 芯片的结构.....	98	6.4 芯片的结构.....	125
5.2.1 专为 USB 设计的芯片 .....	98	6.4.1 特性与限制 .....	125
5.2.2 与现有的芯片兼容 .....	100	6.4.2 芯片内部 .....	125
5.2.3 需要外部微处理器接口的 芯片 .....	101	6.4.3 内存 .....	126
5.2.4 芯片的文件 .....	103	6.4.4 程序堆栈指针 .....	129
5.2.5 固件示例程序 .....	104	6.4.5 数据堆栈指针 .....	129
5.2.6 驱动程序 .....	104	6.5 USB 通信.....	129
5.2.7 调试工具 .....	104	6.5.1 设备地址 .....	131
5.2.8 开发计划的考虑 .....	106	6.5.2 模式 .....	131
5.3 Cypress enCoRe.....	107	6.5.3 端点状态与控制 .....	133
5.3.1 CPU 结构 .....	107	6.5.4 USB 状态与控制 .....	133
5.3.2 USB 控制器 .....	107	6.6 其他 I/O.....	135
5.4 Cypress EZ-USB.....	108	6.6.1 一般用途的 I/O.....	135
5.4.1 CPU 结构 .....	108	6.6.2 内部电路 .....	135
5.4.2 USB 控制器 .....	110	6.6.3 中断 .....	137
5.5 Microchip PIC 16C7x5.....	111	6.6.4 SPI 连接端口 .....	138
5.5.1 结构 .....	111	6.6.5 PS/2 接口.....	139
5.5.2 USB 控制器 .....	112	6.7 定时器的功能.....	140
5.6 Netchip NET2888.....	113	6.7.1 执行定时工作 .....	140
5.6.1 结构 .....	113	6.7.2 测量时距 .....	141
5.6.2 USB 控制器 .....	114	6.8 中断的处理.....	142
5.7 National Semiconductor USBN9603... ..	114	6.8.1 中断服务例程 .....	142
5.7.1 结构 .....	115	6.8.2 GPIO 中断.....	143
5.7.2 USB 控制器 .....	116	6.8.3 USB 端点中断 .....	144
5.8 Philips Semiconductor		6.8.4 计时器中断 .....	144
PDIUSBD11/12.....	116	6.8.5 中断状态 .....	145
5.8.1 结构 .....	116	6.9 CPU 的状态、控制与脉冲.....	146
5.8.2 USB 控制器 .....	117	6.9.1 暂停 CPU .....	146
6.3 C 编译器.....	124	6.9.2 重置 .....	146
6.3.1 C 语言的优点.....	124	6.9.3 选择与控制脉冲 .....	148
6.3.2 C 编译器的用法.....	125	6.9.4 电源的管理 .....	148
6.4 芯片的结构.....	125	6.10 硬件的工作.....	148
6.4.1 特性与限制 .....	125	6.11 固件的工作.....	149
6.4.2 芯片内部 .....	125	6.11.1 端点 0 中断 .....	149
6.4.3 内存 .....	126	6.11.2 传送数据给主机 .....	151
6.4.4 程序堆栈指针 .....	129	6.11.3 从主机接收数据 .....	153
6.4.5 数据堆栈指针 .....	129	6.11.4 处理中断传输 .....	155
6.5 USB 通信.....	129		
6.5.1 设备地址 .....	131		
6.5.2 模式 .....	131		
6.5.3 端点状态与控制 .....	133		
6.5.4 USB 状态与控制 .....	133		
6.6 其他 I/O.....	135		
6.6.1 一般用途的 I/O.....	135		
6.6.2 内部电路 .....	135		
6.6.3 中断 .....	137		
6.6.4 SPI 连接端口 .....	138		
6.6.5 PS/2 接口.....	139		
6.7 定时器的功能.....	140		
6.7.1 执行定时工作 .....	140		
6.7.2 测量时距 .....	141		
6.8 中断的处理.....	142		
6.8.1 中断服务例程 .....	142		
6.8.2 GPIO 中断.....	143		
6.8.3 USB 端点中断 .....	144		
6.8.4 计时器中断 .....	144		
6.8.5 中断状态 .....	145		
6.9 CPU 的状态、控制与脉冲.....	146		
6.9.1 暂停 CPU .....	146		
6.9.2 重置 .....	146		
6.9.3 选择与控制脉冲 .....	148		
6.9.4 电源的管理 .....	148		
6.10 硬件的工作.....	148		
6.11 固件的工作.....	149		
6.11.1 端点 0 中断 .....	149		
6.11.2 传送数据给主机 .....	151		
6.11.3 从主机接收数据 .....	153		
6.11.4 处理中断传输 .....	155		

## 第 6 章 Cypress enCoRe 控制器芯片 ... 118

6.1 选择适合的芯片.....	118
6.2 汇编语言编译器.....	118
6.2.1 汇编程序的基础 .....	120
6.2.2 汇编语言码 .....	121
6.2.3 汇编语言编译器的用法 .....	122
6.3 C 编译器.....	124



6.11.5 其他的响应.....	155	7.9.3 检测特定的设备.....	176
6.12 硬件开发工具.....	156	7.9.4 检测通用的设备.....	177
6.12.1 平台板.....	156	7.10 .inf 文件简介.....	177
6.12.2 监视/调试器.....	157	7.10.1 语法.....	179
<b>第 7 章 驱动程序.....</b>	<b>159</b>	7.10.2 Version 区段.....	180
7.1 设备驱动程序.....	159	7.10.3 ClassInstall 区段.....	180
7.2 设备驱动程序的来源.....	160	7.10.4 Manufacturer 区段.....	181
7.2.1 标准设备类型.....	160	7.10.5 DestinationDirs 区段.....	181
7.2.2 客户化设备.....	160	7.10.6 Strings 区段.....	183
7.3 设备驱动程序的执行模式.....	160	7.10.7 HID 的通用.inf 文件.....	183
7.4 Win32 驱动程序模型.....	161	7.10.8 建立.inf 文件的工具.....	183
7.4.1 程序语言.....	162	7.10.9 使用.inf 文件的技巧.....	184
7.5 分层的驱动程序.....	162	<b>第 8 章 设备类别.....</b>	<b>186</b>
7.5.1 USB 驱动程序层.....	162	8.1 类别.....	186
7.5.2 函数驱动程序.....	162	8.1.1 类别规范.....	186
7.5.3 总线驱动程序.....	164	8.1.2 类别的定义.....	187
7.6 通信的流程.....	164	8.2 标准类型的设备.....	188
7.6.1 用户的角色.....	165	8.2.1 键盘、鼠标与游戏杆.....	189
7.6.2 应用程序的角色.....	165	8.2.2 大容量存储设备.....	189
7.6.3 设备驱动程序的角色.....	166	8.2.3 打印机.....	193
7.6.4 集线器驱动程序的角色.....	168	8.2.4 数码相机与扫描仪.....	194
7.6.5 总线类别驱动程序的角色.....	168	8.2.5 音频.....	196
7.6.6 主机控制器驱动程序的角色.....	168	8.2.6 调制解调器.....	197
7.6.7 设备的角色.....	169	8.3 非标准类型的设备.....	198
7.6.8 响应.....	169	8.3.1 以中等速度传输数据的设备.....	198
7.6.9 通信的终结.....	169	8.3.2 更新 RS-232 设备.....	198
7.7 驱动程序的类型.....	169	8.3.3 POS 设备.....	200
7.7.1 Windows 提供的驱动程序.....	169	8.3.4 取代非标准并行端口设备.....	200
7.7.2 厂商提供的驱动程序.....	169	8.3.5 PC 到 PC 的连接.....	201
7.8 编写客户化的驱动程序.....	170	8.3.6 无线连接.....	201
7.8.1 工具软件.....	170	<b>第 9 章 设备测试.....</b>	<b>203</b>
7.8.2 WinRT for USB.....	171	9.1 USB 测试工具.....	203
7.8.3 WinDriver USB.....	172	9.1.1 测试.....	204
7.8.4 提供函数库的工具软件.....	173	9.1.2 HIDView.....	208
7.9 Windows 如何选择驱动程序.....	174	9.2 测试仪器.....	209
7.9.1 .inf 文件.....	174	9.2.1 协议分析器.....	209
7.9.2 注册表.....	175	9.2.2 USB Expert 的硬件.....	210

9.2.3	USB Expert 的软件.....	210	10.4.11	重置状态.....	227
9.2.4	CATC Traffic Generator.....	212	10.5	高速的总线状态.....	227
9.2.5	Root 1 Test Adapter.....	213	10.5.1	高速 Differential 1 与 Differential 0 状态.....	228
9.3	测试与标志.....	213	10.5.2	高速 Data J 与 K 状态.....	229
9.3.1	USB 实施者论坛合格计划.....	214	10.5.3	Chirp J 与 Chirp K.....	229
9.3.2	校验表.....	214	10.5.4	高速 Squelch.....	229
9.3.3	合格测试.....	214	10.5.5	高速闲置.....	229
9.3.4	USB 标志.....	215	10.5.6	高速信息包开始.....	229
9.3.5	Windows 硬件品质实验室 测试.....	215	10.5.7	高速信息包结束.....	229
9.3.6	驱动程序签名.....	216	10.5.8	高速未连接状态.....	230
<b>第 10 章</b>	<b>电源管理与信号编码.....</b>	<b>217</b>	10.6	数据编码.....	230
10.1	电源的管理.....	217	10.6.1	维持同步.....	230
10.1.1	电压.....	217	10.6.2	位填充.....	231
10.1.2	可以使用总线电源的 设备.....	217	10.6.3	SYNC 字段.....	231
10.1.3	电源要求.....	218	10.6.4	信息包结尾.....	232
10.1.4	通知主机.....	219	10.6.5	时序正确性.....	232
10.2	集线器电源.....	219	10.7	信息包格式.....	233
10.2.1	电的来源.....	220	10.7.1	SYNC 字段.....	233
10.2.2	过流保护.....	222	10.7.2	PID 字段.....	233
10.2.3	电源开关.....	223	10.7.3	地址字段.....	234
10.3	节省电能.....	223	10.7.4	端点字段.....	234
10.3.1	全域与选择挂起.....	223	10.7.5	帧号码字段.....	234
10.3.2	挂起设备的限制.....	223	10.7.6	数据字段.....	235
10.3.3	恢复通信.....	224	10.7.7	CRC 字段.....	235
10.4	低速与全速的总线状态.....	224	10.7.8	信息包间的延迟.....	235
10.4.1	Differential 1 与 Differential 0 状态.....	225	10.8	测试模式.....	235
10.4.2	Single-ended 0 状态.....	226	<b>第 11 章</b>	<b>电路接口.....</b>	<b>238</b>
10.4.3	Single-ended 1 状态.....	226	11.1	接收器与信号.....	238
10.4.4	Data J 与 K 状态.....	226	11.2	低速与全速接收器.....	239
10.4.5	闲置状态.....	226	11.3	高速接收器.....	241
10.4.6	恢复状态.....	226	11.3.1	电路.....	241
10.4.7	信息包开始.....	226	11.3.2	转换速度.....	242
10.4.8	信息包结束.....	226	11.3.3	检测高速设备的移除.....	243
10.4.9	未连接状态.....	227	11.3.4	高速的挂起与恢复.....	243
10.4.10	连接状态.....	227	11.4	信号电压.....	243
			11.4.1	低速与全速.....	243
			11.4.2	高速.....	244

11.5	电缆.....	246	13.3.4	Set_Idle .....	280
11.5.1	连接器.....	248	13.3.5	Get_Protocol .....	280
11.5.2	可拆分的与捕获性的 电缆.....	249	13.3.6	Set_Protocol.....	281
11.5.3	电缆长度.....	250	13.4	传输数据.....	281
11.6	信号的品质.....	250	13.4.1	传送数据给主机 .....	281
11.6.1	噪声.....	250	13.4.2	从主机接收数据 .....	282
11.6.2	平衡线.....	250	13.5	报表的结构.....	285
11.6.3	双绞线.....	251	13.5.1	HID Descriptor Tool .....	285
11.6.4	边缘速率.....	251	13.5.2	短项目 .....	286
<b>第 12 章</b>	<b>集线器</b> .....	<b>252</b>	13.5.3	长项目 .....	287
12.1	集线器的基础.....	252	13.6	Main 项目类型 .....	288
12.1.1	避免设备进入挂起状态.....	253	13.6.1	Input、Output 与 Feature 项目 .....	288
12.1.2	可串联的集线器数目.....	253	13.6.2	Collection 与 End Collection 项目.....	290
12.2	集线器中继器.....	254	13.7	Global 项目类型.....	291
12.2.1	低速与全速的中继器.....	254	13.7.1	识别报表 .....	292
12.2.2	高速的中继器.....	254	13.7.2	数据的使用 .....	292
12.3	事务转换器.....	255	13.7.3	转换原始数据 .....	294
12.4	集线器控制器.....	256	13.7.4	数据的大小与格式 .....	295
12.5	速度.....	257	13.7.5	Global 项目的储存与 恢复 .....	296
12.6	集线器类别.....	257	13.8	Local 项目类型 .....	296
12.6.1	标准描述符.....	258	13.9	填充.....	298
12.6.2	集线器描述符.....	260	13.10	实体描述符.....	298
12.6.3	标准要求.....	261	<b>第 14 章</b>	<b>HID 程序设计</b> .....	<b>300</b>
12.6.4	集线器类别要求.....	262	14.1	主机的通信.....	300
12.6.5	连接端口指示器.....	265	14.1.1	说明文件 .....	301
<b>第 13 章</b>	<b>HID 设备</b> .....	<b>266</b>	14.1.2	HID 函数 .....	301
13.1	HID 简介 .....	266	14.1.3	DirectX.....	302
13.1.1	硬件要求.....	267	14.2	调用 API 函数.....	303
13.1.2	固件要求.....	268	14.2.1	使用 Visual C++ .....	303
13.2	HID 的描述符 .....	268	14.2.2	使用 Visual Basic .....	304
13.2.1	描述符的内容.....	271	14.3	取得 HID 类别的 GUID .....	305
13.2.2	HID 类别描述符 .....	272	14.3.1	使用 Visual C++.....	306
13.2.3	报表描述符.....	273	14.3.2	使用 Visual Basic .....	306
13.3	HID 特定要求 .....	278	14.4	取得 HID 信息的结构数组 .....	307
13.3.1	Get_Report.....	278	14.4.1	使用 Visual C++.....	307
13.3.2	Set_Report.....	279			
13.3.3	Get_Idle.....	279			

14.4.2	使用 Visual Basic .....	307	14.12.1	使用 Visual C++.....	318
14.4.3	详解.....	308	14.12.2	使用 Visual Basic.....	319
14.5	识别 HID 接口 .....	308	14.12.3	详解 .....	320
14.5.1	使用 Visual C++.....	308	14.13	从设备读取输入报表.....	320
14.5.2	使用 Visual Basic .....	309	14.13.1	使用 Visual C++.....	320
14.5.3	详解.....	309	14.13.2	使用 Visual Basic.....	320
14.6	取得设备的路径.....	310	14.13.3	详解 .....	321
14.6.1	使用 Visual C++.....	310	14.14	使用重叠 ReadFile 读取报表.....	322
14.6.2	使用 Visual Basic .....	311	14.14.1	使用 Visual C++.....	322
14.7	取得设备的代号.....	312	14.14.2	使用 Visual Basic.....	323
14.7.1	使用 Visual C++.....	313	14.15	使用多线程的 ReadFile 读取 报表.....	325
14.7.2	使用 Visual Basic .....	313	14.15.1	使用 Visual C++.....	325
14.8	取得厂商与产品 ID .....	314	14.15.2	使用 Visual Basic.....	326
14.8.1	使用 Visual C++.....	314	14.16	传送特征报表给设备.....	326
14.8.2	使用 Visual Basic .....	314	14.16.1	使用 Visual C++.....	327
14.8.3	详解.....	315	14.16.2	使用 Visual Basic.....	327
14.9	取得包含设备能力的缓冲区指针.....	315	14.17	从设备读取特征报表.....	327
14.9.1	使用 Visual C++.....	315	14.17.1	使用 Visual C++.....	327
14.9.2	使用 Visual Basic .....	316	14.17.2	使用 Visual Basic.....	328
14.9.3	详解.....	316	14.18	关闭通信.....	328
14.10	取得设备的能力.....	316	14.18.1	使用 Visual C++.....	328
14.10.1	使用 Visual C++.....	316	14.18.2	使用 Visual Basic.....	329
14.10.2	使用 Visual Basic .....	317	附录 A	USB 参考资料.....	330
14.10.3	详解 .....	318			
14.11	取得数值的能力.....	318			
14.12	传送输出报表给设备.....	318			

# 第 1 章 USB 简介

不管是计算机还是连接的外围设备，在新品开发时保持兼容性是必要的考虑。即使是革命性的新外围设备，也必须使用到所连接计算机提供的接口。当设计一个外围设备的接口时，下列是该外围设备应该具备的特性：

- 稳定性：具备自动查错与除错的功能，使错误的发生率几乎为零。
- 便宜：让大部分的用户都有能力购买。
- 省电：在便携式计算机上节省电池的消耗。
- 有弹性：让许多不同种类的外围设备，都可以使用这个接口。
- 快速：此接口不可以成为传输的瓶颈。
- 容易使用：用户容易安装、设置与使用。
- 操作系统的支持：如果操作系统支持此接口，开发者就不必自行开发底层的驱动程序来使用此接口。

USB(Universal Serial Bus, 通用串行总线)就是一个符合上述所有条件，所开发出来的最新外围设备接口。USB 不仅设计简单，而且使用起来非常有效，许多不同种类的外围设备，例如鼠标、键盘、扫描仪、外接式硬盘、打印机等，都可以使用 USB 接口。图 1.1 是一个使用 USB 的结构范例。

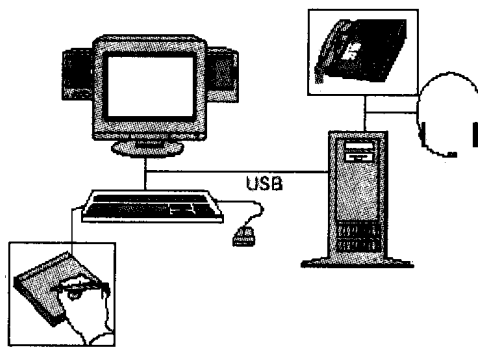


图 1.1 传统的 USB 结构

现在，新式的 PC 上几乎都有一对 USB 端口，可以连接到标准的外围设备或是用户自行设计的外围设备。而且您还可以使用集线器，来增加所需的 USB 连接端口与外围设备。

由于 USB 容易使用，而且使用时比较灵活，使得 USB 接口比起以前的接口开发来得复杂。由于 USB 接口是设计成让各种类的硬件都可以使用，所以对设计 USB 外围设备的人来说，开发 USB 接口是一项挑战。

当 USB 最早出现在 PC 上时，Windows 操作系统尚未包含所有外围设备的驱动程序。不过这些问题，现在都已经获得了解决。因为越来越多的控制芯片、开发工具，以及操作系统支持 USB 接口，USB 已成为外围设备的最新标准接口。

本章将介绍 USB 的发展史，以及其优点与缺点。

## 1.1 USB 的发展史

USB 是一个新接口。新接口并不会常常出现，因为人们已经习惯于旧有的接口，对于新接口往往不会立即全盘接受。如果新接口是建立在旧接口的基础上，则不仅可节省开发的时间以及成本，而且用户也比较能够接受。

IBM PC 最早的设计，就是使用已经存在 20 年的并行端口(Centronics parallel port)与 RS-232 串行端口(serial port)。这不仅加速了原始 PC 设计的流程，并且允许已经在市场上出现的打印机与调制解调器可以立即连接到 PC 上。

不过随着计算机与外围设备的发展，并行端口与 RS-232 串行端口反而成为通信的瓶颈，并且越来越严重。USB 突破旧接口的限制，不仅具备快速的通信速度，而且其弹性令它可以取代各种外围设备所使用的接口。

在过去，一个新接口的开发通常都是单一公司的努力。例如 HP(Hewlett Packard)开发了 HPIB(HP Interface Bus, HP 接口总线)，也称作 GPIB(General Purpose Interface Bus, 通用接口总线)，来连接到实验室的仪器。现在打印机所使用的并行端口接口，则由 Centronics 数据计算机公司(Centronics Data Computer Corporation)开发，因此仍然称作 Centronics 接口。

由单一公司所开发的接口，常会禁止其他公司来使用，或者是要求付费。即使此接口可以免费使用，基于与原开发公司是敌对的公司，或者对该公司可能修改接口而不做另行通知的考虑，使得许多公司不愿使用单一公司所开发的接口。

现在开发的接口，则通常由一群有共同利益的公司或组织所联合开发。例如 IEEE(Institute of Electrical and Electronics Engineers, 电气和电子工程师学会)组织与 TIA(Telecommunication Industry Association, 电信工业协会)组织就开发与出版了许多规范。许多旧的制造业标准，都已经被这些组织所制定的新标准所取代。例如 Centronics 接口被 IEEE-1284 标准取代，而 GPIB 则是 IEEE-488 的基础。

这些组织除了将更改旧有的规范外，也制定新的规范。USB 就是由 7 家公司所共同制定的标准。这 7 家公司为 Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, 以及 Philips。

您可以在下列网址免费下载 USB 2.0 的规范，同时也可以找到最新的 USB 信息。网址 <http://www.usb.org> 如图 1.2 所示。

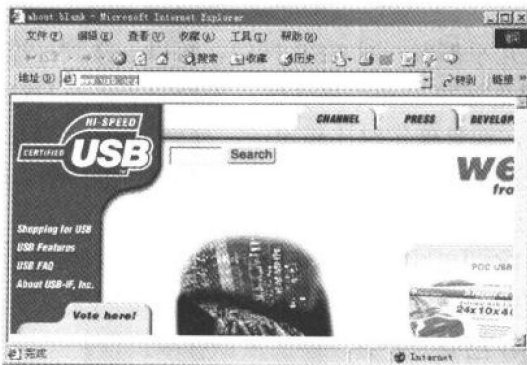


图 1.2 USB 网站

### 1.1.1 规范的版本

USB 规范的前身是 ACCESS.bus 规范，由 Philips 与数字设备公司(Digital Equipment Corporation)所制定而成公开的标准。ACCESS.bus 则是由 I2C synchronics serial bus 发展而来。

ACCESS.bus 的设计是让键盘、指示设备以及其他设备在每秒 100 千位(kilobits)的速度下工作。ACCESS.bus 可以支持 125 个设备，以及 10 米长的电缆。这些设备是插入式的，电缆则包括+5V 电源线与接地线。

ACCESS.bus 定义键盘、指示设备、监视器、显示控制以及文字设备的类别。与 USB 不同的是，ACCESS.bus 使用开放收集器(open-collector)的驱动程序，包含一条数据线(data wire)与一条脉冲线(clock wire)。ACCESS.bus 从未在 PC 上使用，不过仍然使用在智能式电池控制上。

USB 规范的版本变动，如表 1.1 所示。

表 1.1 USB 规范的版本

版本	发表日期	说明
0.7	1994 年 11 月 11 日	覆盖 0.6e
0.8	1994 年 12 月 30 日	修改第 3~8, 10, 11 章, 新增附录
0.9	1995 年 4 月 13 日	修改所有章节
0.99	1995 年 8 月 25 日	修改所有章节
1.0 FDR	1995 年 11 月 13 日	修改第 1, 2, 5~11 章
1.0	1996 年 1 月 15 日	修改第 5~11 章
1.1	1998 年 9 月 23 日	修改所有章节
2.0 (draft 0.79)	1999 年 10 月 5 日	修改第 5, 7, 8, 9, 11 章来增加高速
2.0 (draft 0.9)	1999 年 12 月 21 日	修改所有章节来增加高速
2.0	2000 年 4 月 27 日	高速模式的版本

经过多次版本更新后，USB 1.0 发表于 1996 年 1 月。USB 1.1 则修订了 1.0 版本的问题，并且新增一个新的传输类型(中断输出)。在本书中，USB 1.x 表示 USB 1.0 与 1.1。USB 2.0 在 2000 年 4 月发表，新增了高速模式。2000 年 12 月的 ECN(Engineering Change Notice)则提供了修正，定义了一个新的 mini-B 连接器。

在 PC 上最早提供 USB 支持的是 Windows 95 的 OEM Service Release 2, 之后还有两个修订版: OSR 2.1 与 2.5。这些并不提供给消费者，而是直接卖给 PC 零售商。在这些版本上的 USB，不但功能有限，而且有许多错误或故障，能够连接的外围设备也不多，限制了 USB 的应用。

从 1998 年 7 月的 Windows 98 开始，USB 外围设备开始陆续出现，同时 USB 也成为最受欢迎的接口。Windows 98 的 service Pack 与第二版，更增强对 USB 的支持。Windows 98 的最初版本叫做 Windows 98 Gold，第二版则是叫做 Windows 98 SE(Second Edition)。

本书主要介绍使用 Windows 98，以及其后版本支持的 PC。Windows NT 4 并没有提供

原书缺页



- **自动设置:** 当用户将 USB 设备连接到计算机上时, Windows 操作系统会自动检测该设备, 并且加载适合的驱动程序。在第一次安装时, Windows 会提醒用户放入包含 USB 设备驱动程序的磁盘。其后的安装过程, Windows 会自动帮助用户完成。您不必安装程序, 也不需要重新开机就可以直接使用该 USB 外围设备。
- **没有用户设置:** USB 外围设备没有用户设置的选项, 例如指定通信端口地址或是中断号码(IRQ)等。在安装 USB 设备时, PC 会自动检测。
- **容易连接:** USB 外围设备是属于外接设备, 您不必打开计算机机箱来安插扩充卡。新式的 PC 至少会有两个 USB 连接端口, 如果您要连接更多的 USB 设备, 只要将一个 USB 集线器连接到 PC 上的一个 USB 端口, 然后将其他的 USB 设备连接到该集线器即可。  
您可以在 PC 与 USB 外围设备开机的状态下, 插入或是拔掉 USB 连接头, 而不会造成对 PC 或是 USB 外围设备的损害。操作系统会自动检测 USB 外围设备是否连接, 以及是否可以读取操作。
- **不需要电源:** USB 接口包含+5V 的电源线与接地线, 您可以从计算机或是集线器提供电源。在 500 mA 下, USB 外围设备可以直接使用信道提供的电源。
- **共同接口:** 不同种类的 USB 外围设备可以使用相同的接口, 因此不必设计另外的连接类型。使用 USB 外围设备可以释放多余的 IRQ 中断, PC 会指定多个通信端口地址与一个 IRQ 中断给 USB 接口。如果使用的是非 USB 的外围设备, 不但各需一个 IRQ 中断, 而且常常需要占掉一个扩充卡插槽。

## 1.2.2 传输速度快

在传输速度方面, USB 支持三种信道速度: 低速(low speed)的 1.5 Mb/s, 全速(full speed)的 12 Mb/s, 以及高速(high speed)的 480 Mb/s。具备 USB 功能的 PC 都支持低速与全速, 高速则需要支持 USB 2.0 的主机板或扩充卡。表 1.2 是 USB 的 3 种传输速度及其应用领域的比较。

表 1.2 USB 的传输速度与其应用领域

性能	应用	属性
低速: 交互设备 10 Kb/s ~100 Kb/s	键盘、鼠标、游戏机、虚拟环境	低价 容易使用 动态连接或移除 可接多个外围设备
全速: 电话, 音频, 影像压缩 500 Kb/s ~ 10 Mb/s	POTS、广播、音频、麦克风	低价 容易使用 动态连接或移除 可接多个外围设备 保证带宽