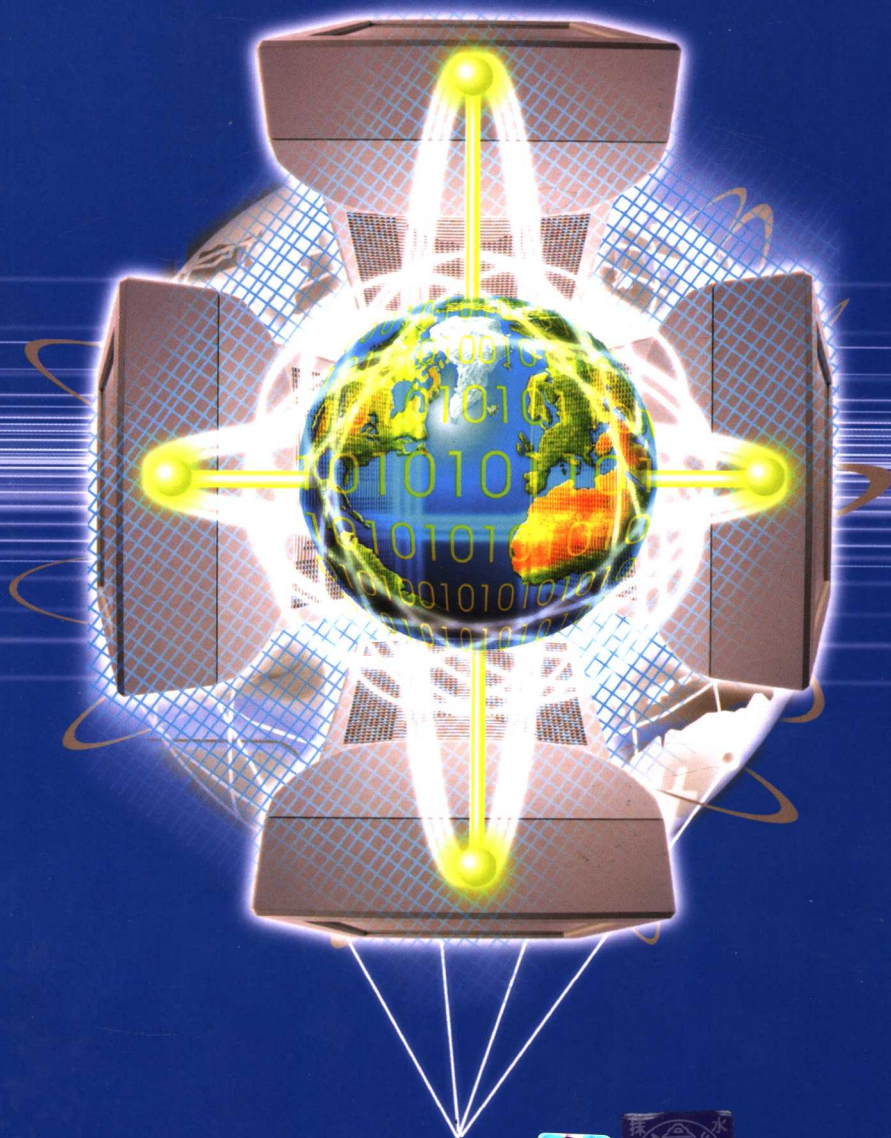


USB 2.0

硬件设计(第2版)

萧世文 宋延清 编著



清华大学出版社

USB 2.0 硬件设计

(第 2 版)

萧世文 宋延清 编著

清华大学出版社
北 京

内 容 简 介

USB 是外围设备与计算机进行连接的快速且灵活的新式接口。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 15 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、HID 设备开发实例以及 USB 参考资料等。

本书适合从事 USB 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

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

USB 2.0 硬件设计(第 2 版)/萧世文,宋延清编著. —北京:清华大学出版社,2006.10
ISBN 7-302-13979-2

I. U… II. ①萧… ②宋… III. 微型计算机—接口 IV. TP368.147

中国版本图书馆 CIP 数据核字(2006)第 118736 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社 总 机:010-62770175 客户服务:010-62776969

责任编辑:桑任松

封面设计:陈刘源

排版人员:房书萍

印 装 者:清华大学印刷厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:28 字数:663 千字

版 次:2006 年 10 月第 2 版 2006 年 10 月第 1 次印刷

书 号:ISBN 7-302-13979-2/TP·8416

印 数:1~4000

定 价:38.00 元

前 言

技术的发展永无止境。

当技术人员还沉湎于 RS-232 及打印机并口接口设计时，一种新的接口标准已经迅速发展成熟，这就是 USB。随着 USB 的出现，一场老式接口的淘汰运动已经拉开了序幕。

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置硬件与软件的选项，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 操作系统包含了标准外围设备的 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 种传输类型和 3 种传输速度。USB 的传输类型包括控制、中断、批量与等时。USB 的传输速度包括低速、中速和高速。低速最快每秒可以传输 1.5Mbits 的数据，中速最快每秒可以传输 12Mbits 的数据，高速最快每秒可以传输 480Mbits 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 芯片与工具有助于简化设计工作。在本书中除了介绍 USB 的基本原理外，也介绍 USB 芯片与工具，并且以实际的范例程序代码，来帮助读者快速进入 USB 设计领域。

本书共分成 15 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，以及各种传输类型的结构、数据格式、传输速度和适用的设备种类。

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 传送各种请求给设备，设备则是依据请求的种类来响应。

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，控制传输可以让 PC 设置 USB 设备的配置，或是用来交换数据。控制传输包含 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。该章还介绍 USB 的标准请求、设备类别的请求以及厂商自定义的请求。

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。通过几个较受欢迎的 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，并介绍

辅助设计用的硬件与软件工具。

第 7 章“驱动程序”介绍 Windows 操作系统的驱动程序模型,撰写客户化的驱动程序所需的注意事项和辅助工具软件。Windows 操作系统使用.inf 文件来建立 USB 设备与驱动程序的信息,该章介绍.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)使用电源的方式。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。该章还详细介绍了 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。该章还介绍集线器的描述符与请求。

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

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

第 15 章“HID 设备开发实例”带领技术人员进入开发实景。通过对电路连接、主机应用程序、固件汇编程序、固件 C 语言程序的全方位展示,进行实际开发前的预演。与前面各章的内容相比,此章的软件内容比较丰富,对于正确理解前面的各种基础概念具有实际的辅助作用(此章为第 2 版新增内容)。

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

本书第 1 版承蒙读者厚爱获得了多次重印的机会。在收到的许多读者反馈中,包含了大量中肯的意见,第 2 版针对这些意见,对若干术语进行了订正,并对第 1 章、第 2 章的部分内容以及其他各章进行了重新整理和编辑,此外,还增加了第 15 章,以纠正第 1 版中实例不足的缺憾。

目 录

第 1 章 USB 简介.....1	1.9.1 所需的链路元件..... 20
1.1 USB 的发展史.....2	1.9.2 开发工具..... 20
1.1.1 规范的版本.....3	1.9.3 开发步骤..... 21
1.1.2 USB 2.0.....4	第 2 章 USB 传输..... 23
1.1.3 IEEE-1394.....4	2.1 传输的基础..... 23
1.2 USB 的优点.....4	2.1.1 配置通信..... 23
1.2.1 容易使用.....4	2.1.2 应用程序通信..... 23
1.2.2 传输速度快.....5	2.1.3 管理总线的数据..... 24
1.2.3 低价位.....6	2.1.4 主机速度和总线速度..... 25
1.2.4 低功耗.....6	2.2 传输要素..... 26
1.2.5 可靠性.....6	2.2.1 设备的端点..... 26
1.2.6 操作系统的支持.....7	2.2.2 管道..... 27
1.2.7 外围设备的支持.....7	2.2.3 传输类型..... 27
1.2.8 灵活性.....7	2.2.4 消息管道与流管道..... 29
1.2.9 USB 实施者论坛.....8	2.2.5 传输的初始化..... 29
1.3 USB 的缺点.....8	2.3 事务..... 30
1.3.1 缺乏对旧硬件的支持.....8	2.3.1 事务的步骤..... 31
1.3.2 缺乏点对点的通信能力.....9	2.3.2 信息包的顺序..... 33
1.3.3 速度的限制.....9	2.3.3 定时约束和保障..... 34
1.3.4 距离的限制.....9	2.3.4 分割事务..... 34
1.3.5 硬件的错误或故障.....9	2.4 联络信号..... 36
1.3.6 协议的复杂性.....9	2.4.1 ACK..... 37
1.3.7 费用.....9	2.4.2 NAK..... 37
1.4 与其他接口的比较.....10	2.4.3 STALL..... 37
1.5 PC 的软硬件要求.....11	2.4.4 NYET..... 38
1.5.1 主机控制器.....11	2.4.5 ERR..... 38
1.5.2 操作系统.....11	2.4.6 无响应..... 38
1.5.3 组件.....12	2.4.7 控制传输的状态..... 39
1.5.4 总线拓扑.....13	2.5 错误的检验..... 39
1.5.5 连接端口.....15	2.5.1 错误校验位..... 39
1.6 主机的责任.....15	2.5.2 数据交替位..... 40
1.7 外围设备的责任.....16	2.6 控制传输..... 41
1.8 传输速度.....19	2.6.1 结构..... 41
1.9 开发流程.....20	2.6.2 信息包的大小..... 44

2.6.3 传输速度	44	第4章 控制传输	75
2.6.4 检测错误	45	4.1 设置阶段	75
2.7 批量传输	46	4.1.1 令牌信息包	76
2.7.1 结构	46	4.1.2 数据信息包	77
2.7.2 信息包大小	47	4.1.3 联络信息包	79
2.7.3 传输速度	47	4.2 数据阶段	80
2.7.4 检测错误	47	4.2.1 令牌信息包	81
2.8 中断传输	47	4.2.2 数据信息包	81
2.8.1 结构	48	4.2.3 联络信息包	81
2.8.2 信息包大小	48	4.3 状态阶段	81
2.8.3 传输速度	48	4.3.1 令牌信息包	82
2.8.4 检测错误	49	4.3.2 数据信息包	82
2.9 等时传输	49	4.3.3 联络信息包	82
2.9.1 结构	49	4.4 检测错误	83
2.9.2 信息包大小	51	4.5 请求	83
2.9.3 传输速度	51	4.5.1 Clear_Feature	84
2.9.4 检测错误	51	4.5.2 Get_Configuration	85
2.10 时间关键的传输	51	4.5.3 Get_Descriptor	85
2.10.1 总线的带宽	51	4.5.4 Get_Interface	86
2.10.2 设备的能力	52	4.5.5 Get_Status	87
2.10.3 主机软件的能力	52	4.5.6 Set_Address	88
2.10.4 Windows 的延迟	52	4.5.7 Set_Configuration	88
第3章 检测设备	54	4.5.8 Set_Descriptor	89
3.1 检测设备的过程	54	4.5.9 Set_Feature	89
3.1.1 检测的步骤	55	4.5.10 Set_Interface	90
3.1.2 检测集线器	58	4.5.11 Synch_Frame	91
3.1.3 移除设备	58	4.5.12 类别特定的请求	91
3.2 描述符	59	4.5.13 厂商特定的请求	92
3.2.1 类型	59	第5章 USB 控制器芯片	93
3.2.2 设备描述符	60	5.1 USB 控制器的组成	93
3.2.3 Device_Qualifier 描述符	62	5.1.1 USB 连接端口	93
3.2.4 配置描述符	64	5.1.2 数据缓冲区	94
3.2.5 Other_Speed_Configuration 描述符	65	5.1.3 CPU	94
3.2.6 接口描述符	66	5.1.4 程序内存	94
3.2.7 端点描述符	68	5.1.5 数据存储器	95
3.2.8 字符串描述符	70	5.1.6 寄存器	95
3.3 USB 2.0 兼容设备的描述符	73	5.1.7 其他接口	95
		5.1.8 其他特性	96

5.2 USB 控制器芯片结构	96	6.3.2 C 编译器的用法	123
5.2.1 专为 USB 设计的芯片	96	6.4 芯片的结构	123
5.2.2 与现有的芯片兼容	98	6.4.1 特性和限制	123
5.2.3 需要外部微处理器接口 的芯片	99	6.4.2 芯片内部	123
5.2.4 芯片的文件	102	6.4.3 内存	124
5.2.5 固件示例程序	102	6.4.4 程序堆栈指针	127
5.2.6 驱动程序	103	6.4.5 数据堆栈指针	127
5.2.7 调试工具	103	6.5 USB 通信	127
5.2.8 开发计划的考虑	104	6.5.1 设备地址	129
5.3 Cypress enCoRe	105	6.5.2 模式	129
5.3.1 CPU 结构	105	6.5.3 端点状态及控制	131
5.3.2 USB 控制器	106	6.5.4 USB 状态及控制	131
5.4 Cypress EZ-USB	106	6.6 其他 I/O	133
5.4.1 CPU 结构	107	6.6.1 一般用途的 I/O	133
5.4.2 USB 控制器	109	6.6.2 内部电路	133
5.5 Microchip PIC 16C7x5	109	6.6.3 中断	135
5.5.1 结构	109	6.6.4 SPI 连接端口	136
5.5.2 USB 控制器	111	6.6.5 PS/2 接口	137
5.6 Netchip NET2888	111	6.7 定时器的功能	138
5.6.1 结构	111	6.7.1 执行定时工作	138
5.6.2 USB 控制器	112	6.7.2 测量时距	139
5.7 National Semiconductor USBN9603	113	6.8 中断的处理	140
5.7.1 结构	113	6.8.1 中断服务例程	140
5.7.2 USB 控制器	114	6.8.2 GPIO 中断	141
5.8 Philips Semiconductor PDIUSB11/12	114	6.8.3 USB 端点中断	142
5.8.1 结构	115	6.8.4 定时器中断	142
5.8.2 USB 控制器	115	6.8.5 中断状态	143
第 6 章 Cypress enCoRe 控制器芯片	116	6.9 CPU 的状态、控制及脉冲	144
6.1 选择适合的芯片	116	6.9.1 暂停 CPU	144
6.2 汇编语言编译器	116	6.9.2 重置	144
6.2.1 汇编程序的基础	118	6.9.3 选择和控制脉冲	146
6.2.2 汇编语言码	119	6.9.4 电源的管理	146
6.2.3 汇编语言编译器的用法	120	6.10 硬件的工作	146
6.3 C 编译器	122	6.11 固件的工作	147
6.3.1 C 语言的优点	122	6.11.1 端点 0 中断	147
		6.11.2 传送数据给主机	149
		6.11.3 从主机接收数据	151
		6.11.4 处理中断传输	153
		6.11.5 其他的响应	153

6.12 硬件开发工具.....	154	7.10 .inf 文件简介.....	176
6.12.1 平台板.....	154	7.10.1 语法.....	178
6.12.2 监视/调试器.....	155	7.10.2 Version 区段.....	179
第7章 驱动程序.....	157	7.10.3 ClassInstall 区段.....	179
7.1 设备驱动程序.....	157	7.10.4 Manufacturer 区段.....	180
7.2 设备驱动程序的来源.....	158	7.10.5 DestinationDirs 区段.....	180
7.2.1 标准设备类型.....	158	7.10.6 Strings 区段.....	182
7.2.2 客户化设备.....	159	7.10.7 HID 的通用.inf 文件.....	182
7.3 设备驱动程序的执行模式.....	159	7.10.8 建立.inf 文件的工具.....	182
7.4 Win32 驱动程序模型.....	159	7.10.9 使用.inf 文件的技巧.....	183
7.5 分层的驱动程序.....	161	第8章 设备类别.....	185
7.5.1 USB 驱动程序层.....	161	8.1 类别.....	185
7.5.2 函数驱动程序.....	161	8.1.1 类别规范.....	185
7.5.3 总线驱动程序.....	162	8.1.2 类别的定义.....	186
7.6 通信的流程.....	163	8.2 标准类型的设备.....	188
7.6.1 用户的角色.....	163	8.2.1 键盘、鼠标和游戏杆.....	188
7.6.2 应用程序的角色.....	163	8.2.2 大容量存储设备.....	188
7.6.3 设备驱动程序的角色.....	165	8.2.3 打印机.....	192
7.6.4 集线器驱动程序的角色.....	166	8.2.4 数码相机和扫描仪.....	193
7.6.5 总线类别驱动程序的角色.....	166	8.2.5 音频.....	195
7.6.6 主机控制器驱动程序 的角色.....	167	8.2.6 调制解调器.....	196
7.6.7 设备的角色.....	167	8.3 非标准类型的设备.....	197
7.6.8 响应.....	167	8.3.1 以中等速度传输数据 的设备.....	197
7.6.9 通信的终结.....	168	8.3.2 更新 RS-232 设备.....	197
7.7 驱动程序的类型.....	168	8.3.3 POS 设备.....	199
7.7.1 Windows 提供的驱动程序.....	168	8.3.4 取代非标准并行端口设备.....	199
7.7.2 厂商提供的驱动程序.....	168	8.3.5 PC 到 PC 的连接.....	200
7.8 编写客户化的驱动程序.....	168	8.3.6 无线连接.....	200
7.8.1 工具软件.....	169	第9章 设备测试.....	202
7.8.2 WinRT for USB.....	169	9.1 USB 测试工具.....	202
7.8.3 WinDriver USB.....	171	9.1.1 测试.....	203
7.8.4 提供函数库的工具软件.....	172	9.1.2 HIDView.....	207
7.9 Windows 如何选择驱动程序.....	173	9.2 测试仪器.....	208
7.9.1 .inf 文件.....	173	9.2.1 协议分析器.....	208
7.9.2 注册表.....	174	9.2.2 USB Expert 的硬件.....	209
7.9.3 检测特定的设备.....	175	9.2.3 USB Expert 的软件.....	209
7.9.4 检测通用的设备.....	176	9.2.4 CATC Traffic Generator.....	211

9.2.5 Root 1 测试适配器.....	212	10.5.2 高速 Data J 和 Data K 状态.....	228
9.3 测试和标记.....	212	10.5.3 Chirp J 和 Chirp K.....	228
9.3.1 USB 实施者论坛的 合格计划.....	213	10.5.4 高速 Squelch.....	228
9.3.2 USB 标志.....	214	10.5.5 高速闲置.....	228
9.3.3 Windows 硬件品质 实验室测试.....	214	10.5.6 高速信息包开始.....	228
9.3.4 驱动程序签名.....	214	10.5.7 高速信息包结束.....	228
第 10 章 电源管理和信号编码.....	216	10.5.8 高速未连接状态.....	229
10.1 电源的管理.....	216	10.6 数据编码.....	229
10.1.1 电压.....	216	10.6.1 维持同步.....	229
10.1.2 可以使用总线电源 的设备.....	216	10.6.2 位填充.....	230
10.1.3 电源要求.....	217	10.6.3 SYNC 字段.....	230
10.1.4 通知主机.....	218	10.6.4 信息包结尾.....	231
10.2 集线器电源.....	218	10.6.5 时序正确性.....	231
10.2.1 电的来源.....	219	10.7 信息包格式.....	232
10.2.2 过流保护.....	221	10.7.1 SYNC 字段.....	232
10.2.3 电源开关.....	222	10.7.2 PID 字段.....	232
10.3 节省电能.....	222	10.7.3 地址字段.....	233
10.3.1 全局挂起和选择挂起.....	222	10.7.4 端点字段.....	233
10.3.2 挂起设备的限制.....	222	10.7.5 帧号码字段.....	233
10.3.3 恢复通信.....	223	10.7.6 数据字段.....	233
10.4 低速和中速的总线状态.....	223	10.7.7 CRC 字段.....	234
10.4.1 Differential 1 和 Differential 0 状态.....	224	10.7.8 信息包间的延迟.....	234
10.4.2 Single-ended 0 状态.....	225	10.8 测试模式.....	234
10.4.3 Single-ended 1 状态.....	225	第 11 章 电路接口.....	237
10.4.4 Data J 和 Data K 状态.....	225	11.1 接收器与信号.....	237
10.4.5 闲置状态.....	225	11.2 低速和中速接收器.....	238
10.4.6 恢复状态.....	225	11.3 高速接收器.....	240
10.4.7 信息包开始.....	225	11.3.1 电路.....	240
10.4.8 信息包结束.....	225	11.3.2 转换速度.....	241
10.4.9 未连接状态.....	226	11.3.3 检测高速设备的移除.....	242
10.4.10 连接状态.....	226	11.3.4 高速的挂起和恢复.....	242
10.4.11 重置状态.....	226	11.4 信号电压.....	243
10.5 高速的总线状态.....	226	11.4.1 低速和中速.....	243
10.5.1 高速 Differential 1 和 Differential 0 状态.....	227	11.4.2 高速.....	243
		11.5 电缆.....	246
		11.5.1 连接器.....	248
		11.5.2 可拆分的与捕获性 的电缆.....	249

11.5.3 电缆长度	250	13.4.1 发送数据给主机	282
11.6 信号的品质	250	13.4.2 从主机接收数据	283
11.6.1 噪声	250	13.5 报表的结构	285
11.6.2 平衡线	250	13.5.1 HID Descriptor Tool	286
11.6.3 双绞线	251	13.5.2 短项目	287
11.6.4 边缘速率	251	13.5.3 长项目	287
第 12 章 集线器	252	13.6 Main 项目类型	288
12.1 集线器的基础	252	13.6.1 Input、Output 和 Feature 项目	288
12.1.1 避免设备进入挂起状态	253	13.6.2 Collection 与 End Collection 标记	291
12.1.2 可串联的集线器数目	253	13.7 Global 项目类型	291
12.2 集线器中继器	254	13.7.1 识别报表	292
12.2.1 低速和中速的中继器	254	13.7.2 数据的使用	293
12.2.2 高速的中继器	254	13.7.3 转换原始数据	294
12.3 事务转换器	255	13.7.4 数据的大小与格式	296
12.4 集线器控制器	256	13.7.5 Global 项目的储存 与恢复	296
12.5 速度	257	13.8 Local 项目类型	296
12.6 集线器类别	257	13.9 填充	298
12.6.1 标准描述符	258	13.10 实体描述符	298
12.6.2 集线器描述符	260	第 14 章 HID 程序设计	300
12.6.3 标准请求	261	14.1 主机的通信	300
12.6.4 集线器类别请求	262	14.1.1 说明文件	301
12.6.5 连接端口指示器	265	14.1.2 HID 函数	301
第 13 章 HID 设备	266	14.1.3 DirectX	302
13.1 HID 简介	266	14.2 调用 API 函数	303
13.1.1 硬件要求	267	14.2.1 使用 Visual C++	303
13.1.2 固件要求	268	14.2.2 使用 Visual Basic	304
13.2 HID 的描述符	269	14.3 取得 HID 类别的 GUID	305
13.2.1 描述符的内容	272	14.3.1 使用 Visual C++	306
13.2.2 HID 类别描述符	272	14.3.2 使用 Visual Basic	306
13.2.3 报表描述符	274	14.4 取得 HID 信息的结构数组	307
13.3 HID 特定请求	278	14.4.1 使用 Visual C++	307
13.3.1 Get_Report	279	14.4.2 使用 Visual Basic	307
13.3.2 Set_Report	279	14.4.3 详解	307
13.3.3 Get_Idle	279	14.5 识别 HID 接口	308
13.3.4 Set_Idle	280	14.5.1 使用 Visual C++	308
13.3.5 Get_Protocol	281		
13.3.6 Set_Protocol	281		
13.4 传输数据	281		

14.5.2 使用 Visual Basic.....	308	14.13 从设备读取输入报表.....	319
14.5.3 详解.....	309	14.13.1 使用 Visual C++.....	320
14.6 取得设备的路径.....	309	14.13.2 使用 Visual Basic.....	320
14.6.1 使用 Visual C++.....	310	14.13.3 详解.....	321
14.6.2 使用 Visual Basic.....	311	14.14 使用重叠 ReadFile 读取报表.....	321
14.7 取得设备的代号.....	312	14.14.1 使用 Visual C++.....	322
14.7.1 使用 Visual C++.....	312	14.14.2 使用 Visual Basic.....	323
14.7.2 使用 Visual Basic.....	313	14.15 使用多线程的 ReadFile 读取报表.....	324
14.8 取得厂商与产品 ID	314	14.15.1 使用 Visual C++.....	324
14.8.1 使用 Visual C++.....	314	14.15.2 使用 Visual Basic.....	326
14.8.2 使用 Visual Basic.....	314	14.16 发送特征报表给设备.....	326
14.8.3 详解.....	315	14.16.1 使用 Visual C++.....	326
14.9 取得包含设备能力的缓 冲区指针	315	14.16.2 使用 Visual Basic.....	326
14.9.1 使用 Visual C++.....	315	14.17 从设备读取特征报表.....	327
14.9.2 使用 Visual Basic.....	315	14.17.1 使用 Visual C++.....	327
14.9.3 详解.....	316	14.17.2 使用 Visual Basic.....	327
14.10 取得设备的能力.....	316	14.18 关闭通信.....	327
14.10.1 使用 Visual C++.....	316	14.18.1 使用 Visual C++.....	328
14.10.2 使用 Visual Basic.....	317	14.18.2 使用 Visual Basic.....	328
14.10.3 详解.....	317	第 15 章 HID 设备开发实例.....	330
14.11 取得数值的能力.....	317	15.1 HID 设备开发综述.....	330
14.12 发送输出报表给设备.....	318	15.2 HID 温度计开发示例.....	333
14.12.1 使用 Visual C++.....	318	15.3 用 C 语言来设计固件	400
14.12.2 使用 Visual Basic.....	318	附录 USB 参考资料.....	430
14.12.3 详解.....	319		

第 1 章 USB 简介

不管是计算机还是所连接的外围设备，在新开发产品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。在设计一个外围设备的接口时，应该具备下列特性：

- 稳定——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。
- 灵活性——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、设置和使用。
- 被操作系统支持——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的外围设备接口。

USB 不仅形式简单，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、外接式硬盘和打印机等，都可以使用 USB 接口。图 1.1 是 USB 的应用范畴示意图。

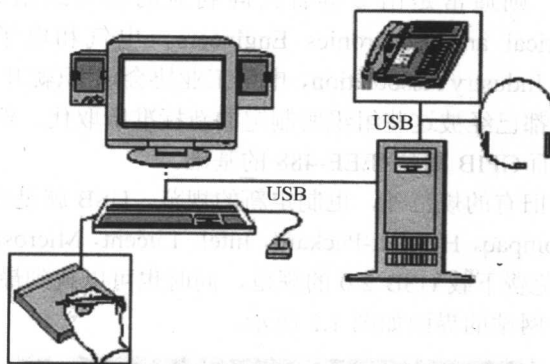


图 1.1 USB 的应用范畴示意图

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 连接端口和外围设备。

尽管 USB 在使用时比较容易和灵活，但是 USB 接口的开发比起以前的接口开发来得更复杂。由于 USB 接口须设计成让各种类型的硬件都可以使用，所以对设计 USB 外围设备的人员来说，开发 USB 接口实在是一项挑战。

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得到了解决。因为越来越多的控制芯片、开发工具以及操作系统都支持 USB 接口，事实上，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，及其优点和缺点。

1.1 USB 的发展史

USB 是一种新型接口。新接口并不会经常出现,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和成本,而且用户也比较能够接受。

在 IBM PC 最早的设计中,就是使用已经存在了 20 年的并行端口(Centronics parallel port)及 RS-232 串行端口(serial port)。这不仅加速了原始 PC 设计的流程,并且允许将已经在市场上出现的打印机和调制解调器立即连接到 PC 上。

不过随着计算机和外围设备的发展,并行端口和 RS-232 串行端口已逐渐成为通信的瓶颈,并且越来越严重。USB 能够突破旧接口的限制,不仅具备快速通信的能力,而且其灵活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竞争公司或者对该公司可能修改接口而不做另行通知的考虑,也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,则通常是由一群有共同利益的公司或组织联合开发的。例如 IEEE(Institute of Electrical and Electronics Engineers, 电气和电子工程师学会)组织和 TIA(Telecommunication Industry Association, 电信工业协会)组织就开发并公布了许多规范。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。

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

您可以从 Internet 免费下载 USB 2.0 的规范,同时也可以找到最新的 USB 信息,网址为 <http://www.usb.org>, 网站的界面如图 1.2 所示。

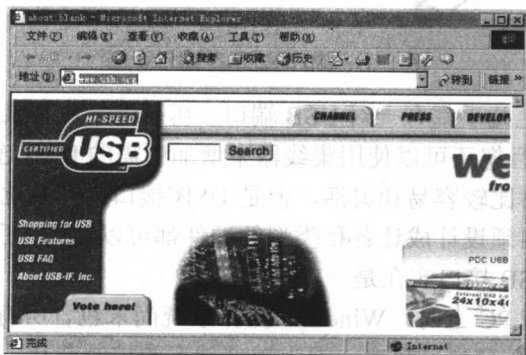


图 1.2 USB 网站

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)所制定并成为公开的标准。ACCESS.bus 是由 I2C synchronics serial bus 发展而来的。

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 电源线和接地线。

ACCESS.bus 定义了键盘、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 使用集电极开路式(open-collector)的驱动器, 包含一条数据线(data wire)和一条时钟线(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, 并且增加了一个新的传输类型(中断输出)。在本书中, USB 1.x 表示 USB 1.0 和 1.1。USB 2.0 于 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 和 2.5。这些并不提供给消费者, 而是直接卖给 PC 零售商。基于这些版本的 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 和第二版更增强了对 USB 的支持。Windows 98 的最初版本叫做 Windows 98 Gold, 第二版则叫做 Windows 98 SE(Second Edition)。

本书主要介绍 Windows 98 及其后来版本所支持的 PC。Windows NT 4 并没有提供 USB

的支持,但 Windows 2000 提供了 USB 的支持。在 Windows 2000 上的 USB 比较稳定,主要使用在商业上。最新的 Windows XP 以 Windows 2000 为基础,既能满足商业用途,也适合于家庭使用。

1.1.2 USB 2.0

USB 2.0 是版本更新的一大跃进,它加入了许多高速传输的特性支持。原来的预测是希望增进 20 倍的传输速度,实际上则达到了 40 倍的传输速度(480Mbps/s)。对打印机、扫描仪、硬盘以及影像设备等,无疑是一大福音。

USB 2.0 与 USB 1.1 兼容,它们使用相同的连接器和电缆。不过如果要使用最新的高速传输,则必须使用与 USB 2.0 兼容的主机和集线器。与 USB 2.0 兼容的主机和集线器同时也能够与 USB 1.1 的外围设备沟通。如果您将较慢的外围设备连接到 USB 2.0 兼容的集线器上,集线器会自动转换外围设备的速度和 USB 2.0 的速度。

1.1.3 IEEE-1394

除了 USB 接口之外,另外一个新的外围设备接口是 IEEE-1394。在 Apple 的计算机上,IEEE-1394 接口称作“火线(Firewire)”。

IEEE-1394 比 USB 快速而且更有灵活性,不过其价钱则较贵。USB 适合使用在键盘、扫描仪、硬盘以及打印机等中低速度与中低价位的应用上。而 IEEE-1394 则适合使用在需要高速的影像或连接设备上,及没有 PC 主机的场合。

对许多设备来说,IEEE-1394 与 USB 都适用。在使用 USB 的场合,一台主机可以控制与许多外围设备的通信。所有的复杂性都由主机来控制,外围设备的电路则相对简单,而且很便宜。

IEEE-1394 则是使用点对点(peer-to-peer)的模式,外围设备可以彼此沟通。单一的传输可以同时传给多个接收设备。所以 IEEE-1394 比 USB 更有灵活性,不过相对地,外围设备的电路则较复杂且昂贵。

USB 1.x 的传输速度是 12Mbps/s,而 USB 2.0 的传输速度可以到 480Mbps/s。IEEE-1394 的传输速度是 400Mbps/s,而 IEEE-1394.b 的传输速度则可以达到 3.2Gbps/s,比 USB 2.0 的速度快了 6 倍。

1.2 USB 的优点

将外部设备连接到计算机上时,USB 接口是优先的选择。不管是使用外围设备的用户,还是开发 USB 软硬件的设计者,USB 都有让双方满意的特性。

从用户的观点来看,USB 的优点是容易使用,具有快速与稳定的传输,价位低,省电以及具有灵活性。

1.2.1 容易使用

容易使用是 USB 的主要设计目标,而 USB 的设计具有下列特性:

- 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。此后在使用阶段，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。
- 无用户设置：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，只需要将一个 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 和 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，以及是否可以进行读取操作。
- 不需要外部电源：USB 接口包含+5V 的电源线和接地线，您可以从计算机或是集线器提供电源。在不大于 500mA 的情况下，USB 外围设备可以直接使用总线提供的电源。
- 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和一个 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，而且常常需要占用一个扩充卡插槽。

1.2.2 传输速度快

在传输速度方面，USB 支持 3 种总线速度：低速(low speed)的 1.5Mbps/s，中速(full speed)的 12Mbps/s，以及高速(high speed)的 480Mbps/s。

这些速度是指总线(Bus)所支持的信号速度或位速率。具体到某个 USB 设备可期望的数据传输速率会低一点。因为除了数据之外，总线还必须携带状态、控制和错误检查信号。另外，由于多个 USB 设备可能在分享总线，所以理论上单一传输的最大速率在高速时可超过 53Mbytes/s，在中速时大约为 1.2Mbytes/s，在低速时则为 800 bytes/s。

之所以要采取 3 种速度，是因为如下的原因。

对于低速而言，首先，低速设备通常造价低廉，此外对鼠标和其他需要柔性电缆的设备来说，低速的电缆由于不需要多少屏蔽，因此会更加柔软。

中速与现有的串行端口和并行端口的速度具有可比性，或性能更好。因此可以替代这些端口。

在 USB 1.0 发布之后，发现更快的接口显然会很有用；当时通过调查表明，速度可以提高 40 倍，而且还能让接口与低速和中速的设备向后兼容。于是在后来发布的 USB 2.0 之中，高速就成为了一个选项。

具备 USB 功能的 PC 都支持低速与中速，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。