

USB

开发大全

【美】Jan Axelson 著

李鸿鹏 郑瑞霞 陈香凝 张铮 译

(第4版)

涵盖 USB 硬件的全部技术细节

USB 接口的工作原理和组织结构

USB 编程技术及 HID 设备驱动开发

USB 无线通信技术

USB 3.0 强大动力和超高速总线技术



人民邮电出版社
POSTS & TELECOM PRESS

USB

开发大全 (第4版)

【美】Jan Axelson 著

李鸿鹏 郑瑞霞 陈香凝 张铮 译



人民邮电出版社
北京

图书在版编目 (CIP) 数据

USB开发大全 : 第4版 / (美) 阿克塞尔森
(Axelson, J.) 著 ; 李鸿鹏等译. — 北京 : 人民邮电出版社, 2011.10
ISBN 978-7-115-25955-4

I. ①U… II. ①阿… ②李… III. ①USB总线—串行接口 IV. ①TP334.7

中国版本图书馆CIP数据核字 (2011) 第158215号

版 权 声 明

USB Complete: The Developer's Guide, Fourth Edition

Copyright © 1999-2009 by Jan Axelson

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, by photocopying, recording or otherwise, without the prior permission in writing from Lakeview Research LLC. CHINESE SIMPLIFIED language edition published by POSTS & TELECOMMUNICATION PRESS, Copyright©2010.

本书中文简体版由美国芝加哥评论独立出版集团 (IPG) 授权人民邮电出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

内 容 提 要

本书内容大致可分为4个方面。第一, 阐述USB协议, 包括基本常识、设备枚举、传输过程、设备类、主机通信方式以及总线状态等。读者通过学习, 可迅速掌握USB接口的工作原理和组织结构。第二, 详细地介绍了USB硬件, 从控制器芯片的选择、电源供给、集线器原理, 到线缆的电气和机械特性, 几乎涵盖USB硬件的全部技术细节。第三, 讨论了USB的程序编写, 涉及设备固件和驱动, 重点阐述了HID设备类驱动程序的开发, 并给出了在厂商专属设备上应用WinUSB的方法。第四, 作者也关注了几种开发工具的使用。

本书是USB软硬件的开发指南, 一本无可替代的工具书。新版本与时俱进地增加了USB 3.0协议及其超高速总线的內容。无论初学者还是资深开发人员, 读到此书, 都将大受裨益。

USB 开发大全 (第4版)

- ◆ 著 [美] Jan Axelson
译 李鸿鹏 郑瑞霞 陈香凝 张 铮
责任编辑 张 涛
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
大厂聚鑫印刷有限责任公司印刷
- ◆ 开本: 787×1092 1/16
印张: 20.25
字数: 483千字 2011年10月第1版
印数: 1-3000册 2011年10月河北第1次印刷
著作权合同登记号 图字: 01-2010-7134

ISBN 978-7-115-25955-4

定价: 55.00元

读者服务热线: (010)67132692 印装质量热线: (010)67129223

反盗版热线: (010)67171154

前言

本书适用于那些参与对通用串行总线（Universal Serial Bus, USB）接口设备进行设计和编程的开发人员。如果您是一位硬件设计师，负责编写驻留在 USB 设备中的固件程序，或者是编写用于与设备通信的应用程序，这本书就是为您准备的。

USB 十分强大，足以服务于大量的设备功能；大家熟悉的 USB 周边设备包括鼠标、键盘、驱动器、打印机、话筒和相机等。USB 还适用于数据采集单元、控制系统及其他特定功能的设备，包括一次有效（one-of-a-kind）的设计。正确选择设备硬件、软件驱动程序、开发工具和各项技术，可使旨在获得执行功能且不带错误或无用户抱怨的产品的的设计过程变得轻松。本书将一路指引您朝着这个目标前进。

书中都有什么

USB 规范是关于 USB 接口的最终依据。但对于实际设计来说，USB 规范省略了用于特定设备硬件、软件、其他工具和产品上的建议、代码例程及更多信息。而本书则填补了 USB 规范与实际设计间的这个鸿沟，它将为您节省时间、避免开发和使用软件来访问设备时所遇到的麻烦。下面是本书中回答了的一些问题。

- USB 设备如何通信？我不会赘述 USB 规范了。相反，我会侧重于那些为开发高效、可靠的通信设备所需要知道的事情。
- 怎样判断设备是否该使用 USB 接口呢？找出您的设备是否应当使用 USB 接口或其他接口。若选择 USB，您将学会如何在 USB 的 4 种速度（还包括 USB 3.0 的超高速）中作出选择，以及判断 4 种 USB 传输类型中哪个更适合您的应用程序。
- 我的设备该使用什么样的控制器芯片呢？所有的 USB 设备都含有一个智能控制器，用于管理 USB 通信。大量的芯片厂商提供了具有不同架构和能力的控制器芯片。本书将基于您的工程需要、预算及对芯片架构、编程语言和工具的偏好，帮您挑选到一个合适的控制器。
- 应用程序是如何与我的设备通信的呢？在计算机上，应用程序通过与操作系统事先指定给设备的驱动程序进行通信，实现对某个 USB 设备的访问。您将了解到设备是否可使用由主机操作系统类提供的某个驱动程序。对于操作系统所不支持的类的设备，您可以在微软公司的 WinUSB 驱动程序、其他通用驱动程序和定制的驱动程序中进行尝试。代码示例给出了如何实现检测，以及使用 Visual Basic、Visual C# 应用程序与设备通信的方法。
- 为支持 USB 通信，我的设备需要什么样的固件呢？您将在书中找到编写固件的方法，固件可使设备能够响应 USB 请求、事件和任何目的数据交换。
- 我的设备需要自己的电源吗？USB 接口可向设备提供电源，包括为电池供电的设备充

电。您将了解到如何判定设计可否从总线上获得它所需的全部电源，如何满足 USB 的节电要求，以及如何从总线上为电池供电的设备充电等。

- 如何实现无线通信呢？无线 USB 规范定义了一种可在无线缆的情况下，实现 USB 设备通信的方式。其他工业标准和厂商技术还提供了额外的选择。您可了解到哪种技术适合您的设备。
- 设备如何访问其他 USB 设备呢？您将找到为嵌入式系统或一个 USB On-The-Go（USB OTG，无主机参与的情况下外设直接互连通信）设备开发主机的方法，其中 USB OTG 设备既是 USB 设备，还可作为一个具备有限能力、可访问其他 USB 设备的主机。
- 如何保证通信的稳定性呢？所有设备必须能够响应 USB 端口上的请求和其他事件。而主机计算机则必须能完成如检测设备接入、分配恰当驱动程序及与设备进行数据交换的任务。本书提供了有助于完成这些任务的小提示、代码示例和调试软、硬件的信息。

新加的内容

自 1996 年 USB 1.0 发布以来，规范的核心内容基本保持不变，而接口却已被大大扩展了。现在的接口支持更快的总线速度、改进了的电源管理、更多的设备类、无线通信功能、双重角色设备（设备和主机）以及更多功能。而且新改进的芯片和开发工具，使设备开发及通过软件访问设备的任务变得轻松。

此版经过了全面的修订和升级。第 4 版新增的内容是对 USB 3.0 和超高速总线的介绍。同样，您还将了解到如何使用微软 WinUSB 驱动程序对厂商专属功能的设备进行访问。所升级的主题包括：设备控制器芯片、无线 USB 通信技术、节电协议及 USB 设备类。

我提供了以 Visual Basic 和 Visual C#两种语言编写的应用程序的代码实例。而对于设备固件，则在讨论中使用了 Micro Engineering Labs 的 PICBASIC PRO™以及 Microchip Technology 的 MPLAB® C 编译器。

本书的升级和更多内容

想要找到更多关于开发 USB 设备及其通信软件的信息，请访问我的 USB 主页 www.Lvr.com。您将找到代码示例、文章、产品、工具和关于 USB 设备开发的网站链接。

本书的修改和升级也可从 www.Lvr.com 上获得。若您发现了本书中的错误，也请通知我。

代码示例

在每个代码示例的开始，将由副标题给出所使用的编程语言。

侧 标	编成语言	供 应 商
VB	Visual Basic .NET	微软
VC#	Visual C# .NET	微软
PBP	PICBASIC PRO	Micro Engineering Labs 有限公司
C18	用于 PIC18 CPU 的 MPLAB C 编译器	Microchip Technology 有限公司

所提供的.NET 代码与.NET Framework Version 2.0 及其后续版本兼容。应用程序示例可免费从网站 www.Lvr.com 下载获得。

缩写

这本书使用如下的缩写和符号来表示数量和单位。

乘数单位		
符 号	描 述	乘 数
p	皮	10^{-12}
n	纳	10^{-9}
μ	微	10^{-6}
m	毫	10^{-3}
k	千	10^3
K	千	2^{10} (1024)
M	兆	10^6 或 2^{20} , 取决于数据内容
G	吉	10^9 或 2^{30} , 取决于数据内容

电气单位		
符 号	描 述	
A	安培	
F	法拉	
Ω	欧姆	
V	伏特	

时间单位		
符 号	描 述	
s	秒	
Hz	赫兹 (每秒的周期数)	

距离单位		
符 号	描 述	
in.	英寸 (1 英寸 $\approx$ 25.4mm)	
ft	英尺 (1 英尺 $\approx$ 0.3048m)	
m	米	

数据单位		
符 号	描 述	
bit (b)	比特	
B	字节	
bit/s	比特每秒	

数制系统

二进制数会有一个尾随的下标 “b”。例如：10100011_b。一个例外是，当您能从数的内容中明确它为二进制时，“b” 将被省略。例如：将比特位 6 到 5 设置为 01。

十六进制数值将以一个 “h” 结尾。例如：A3h。

其他所有数值都是十进制的。例如：163。

致谢

USB 是个很大的主题，若没有其他人的帮助，我根本无法完成此书。所以我要感谢许多人。

我的质检员给了我许多反馈，帮我使得这本书尽可能详尽和精确。但是，书中所有错误的责任将归咎于我（且只归咎于我）。非常感谢 Paul E.Berg、Greg Burk、Robert Dunstan、John Garney、Bill Jacobus、Kosta Koeman 和 Matt Leptich。

其他我要对其支持表示感谢的人是 J. Gordon Electronic design 的 Phyllis Brown、DeVaSys Embedded Systems 的 Michael DeVault、USB-IF 的 Traci Donnell、Microchip Technology 有限公司的 David Flowers、Ellisys 的 Laurent Guinnard、CWAV 有限公司的 Tim Harvey、Bitwise Systems 的 Blake Henry、usb-by-example.com 的 John Hyde、英特尔公司的 Rahman Ismail 和 Jeff Ravencraft、Trace Systems 有限公司的 Bob Miller 博士，以及 microEngineering Labs 有限公司的 Jeff Schmoyer。

在由下列朋友帮助获得的早先版本的基础上，才有了本书这个新版本。感谢 Joshua Buerger、Gary Crowell、Fred Dart、Wendy Dee、Lucio DiJasio、Keith Dingwall、Dave Dowler、Mike Fahrion、David Goll、John M. Goodman、Lane Hauck、David James、Christer Johansson、Geert Knapen、Alan Lowne、Jon Lueker、Brad Markisohn、Rich Moran、Bob Nathan、Walter Oney、Amar Rajan、Marc Reinig、Rawin Rojvanit、Glenn M. Roberts、Robert Severson、Craig R. Smith 和 Dave Wright。

希望您觉得本书有用，并欢迎向信箱 jan@Lvr.com 发送您的评论。

目 录

第 1 章 USB 基础..... 1

1.1 USB 的用途和局限.....1	
1.1.1 给用户带来的好处.....1	
1.1.2 给开发者带来的好处.....3	
1.1.3 USB 不能做什么.....5	
1.1.4 USB 与 Ethernet.....7	
1.1.5 USB 与 IEEE-1394.....7	
1.2 USB 接口的发展演变.....7	
1.2.1 USB 1.0.....7	
1.2.2 USB 1.1.....8	
1.2.3 USB 2.0.....8	
1.2.4 USB 3.0.....9	
1.2.5 USB On-The-Go.....9	
1.2.6 无线 USB.....9	
1.3 总线组成部分.....10	
1.3.1 拓扑结构.....10	
1.3.2 关于总线速率的考虑.....11	
1.3.3 术语.....12	
1.4 分割工作.....13	
1.4.1 主机的责任.....13	
1.4.2 设备的职责.....14	
1.4.3 总线速率和数据吞吐量.....15	
1.5 开发设备.....16	
1.5.1 组件.....16	
1.5.2 开发工具.....16	
1.5.3 工程开发的步骤.....16	
1.6 USB 3.0 常见问题.....17	
1.6.1 功能.....17	
1.6.2 兼容性.....18	
1.6.3 线缆.....19	
1.6.4 电源和功耗.....20	

第 2 章 深入 USB 传输..... 21

2.1 传输的基础.....21	
2.1.1 传输要素.....21	
2.1.2 通信的目的.....22	
2.1.3 管理总线上的数据.....22	
2.2 传输要素.....23	
2.2.1 端点：数据的来源和数据接收端.....23	
2.2.2 事务类型.....23	
2.2.3 管道：将端点连接到主机.....24	
2.2.4 传输类型.....24	
2.2.5 流管道和消息管道.....25	
2.2.6 发起传输.....25	
2.3 USB 2.0 事务.....26	
2.3.1 事务步骤.....27	
2.3.2 信息包的顺序.....29	

2.3.3 时间约束和保证.....29	
2.3.4 分割事务.....29	
2.4 保证传输成功.....30	
2.4.1 状态与控制.....30	
2.4.2 报告控制传输的状态.....32	
2.4.3 错误检查.....32	
2.5 超高速事务.....34	
2.5.1 信息包类型.....34	
2.5.2 数据传输.....36	
2.5.3 连接管理信息包.....38	

第 3 章 通用的传输类型..... 39

3.1 控制传输.....39	
3.1.1 可用性.....39	
3.1.2 结构.....39	
3.1.3 数据大小.....44	
3.1.4 速度.....44	
3.1.5 错误的检测与处理.....45	
3.1.6 设备的职责.....45	
3.2 批量传输.....45	
3.2.1 可用性.....46	
3.2.2 结构.....46	
3.2.3 数据大小.....48	
3.2.4 速率.....48	
3.2.5 错误的检测和处理.....48	
3.2.6 设备的职责.....48	
3.3 中断传输.....49	
3.3.1 可用性.....49	
3.3.2 结构.....49	
3.3.3 数据大小.....50	
3.3.4 速率.....50	
3.3.5 错误的检测和处理.....51	
3.3.6 设备的职责.....51	
3.4 等时传输.....51	
3.4.1 可用性.....52	
3.4.2 结构.....52	
3.4.3 数据大小.....54	
3.4.4 速度.....54	
3.4.5 错误的检测和处理.....55	
3.4.6 设备的职责.....55	
3.5 更多关于时间关键传输的内容.....55	
3.5.1 总线带宽.....55	
3.5.2 设备性能.....56	
3.5.3 主机性能.....56	
3.5.4 主机等待时间.....56	

第 4 章 枚举：主机如何了解设备..... 58

4.1 枚举过程.....58	
-----------------	--

4.1.1 枚举步骤	59	6.3.1 Microchip 的 PIC18F4550	96
4.1.2 设备移除	62	6.3.2 Cypress EZ-USB	99
4.1.3 枚举成功的秘诀	62	6.3.3 ARM	101
4.2 描述字	63	6.4 连接到 CPU 的控制器	101
4.2.1 类型描述字	63	6.4.1 ST-NXP Wireless 公司的 ISP1582 型控制器	101
4.2.2 设备描述字	64	6.4.2 PLX Technology 的 NET2272 型控制器	102
4.2.3 Device_Qualifier 描述字	67	6.4.3 FTDI 公司的 USB UART 和 USB FIFO	103
4.2.4 配置描述字	67	第 7 章 设备类	105
4.2.5 Other_Speed_Configuration 描述字	68	7.1 元素与用途	105
4.2.6 接口联合描述字	68	7.1.1 已被公认的规范	105
4.2.7 接口描述字	69	7.1.2 类规范的组成	106
4.2.8 端点描述字	71	7.2 已定义的类	107
4.2.9 超高速端点伙伴描述字	73	7.2.1 音频类	107
4.2.10 字符串描述字	73	7.2.2 通信类	109
4.2.11 二进制目标存储及设备性能描述字	74	7.2.3 内容安全类	113
4.2.12 其他标准描述字	75	7.2.4 设备固件升级类	114
4.2.13 微软 OS 描述字	75	7.2.5 人机接口类	116
4.2.14 更新描述字到 USB 2.0	76	7.2.6 IrDA 桥设备类	118
第 5 章 控制传输：用于关键数据的 结构化请求	77	7.2.7 大容量存储设备类	119
5.1 控制传输的组成	77	7.2.8 个人健康设备类	122
5.1.1 设置阶段	77	7.2.9 打印机设备类	122
5.1.2 数据阶段	78	7.2.10 智能卡设备类	124
5.1.3 状态阶段	79	7.2.11 静态图像获取设备类	126
5.1.4 错误处理	80	7.2.12 测试与测量设备类	127
5.1.5 设备固件	80	7.2.13 视频设备类	128
5.2 标准请求	81	7.3 非标准功能的实现	131
5.2.1 Get Status (获得状态) 请求	82	7.3.1 选择驱动程序	131
5.2.2 Clear Feature (清除特征) 请求	83	7.3.2 使用通用驱动程序	131
5.2.3 Set Feature (设置特征) 请求	83	7.3.3 从 RS-232 转换到 USB	131
5.2.4 Set Address (设置地址) 请求	84	7.3.4 从并行端口转换到 USB	132
5.2.5 Get Descriptor (获得描述字) 请求	85	7.3.5 PC 与 PC 间的通信	133
5.2.6 Set Descriptor (设定描述字) 请求	85	第 8 章 主机的通信方式	134
5.2.7 Get Configuration (获得配置) 请求	85	8.1 设备驱动程序	134
5.2.8 Set Configuration (设定配置) 请求	86	8.1.1 分层的驱动程序模型	134
5.2.9 Get Interface (获得接口) 请求	86	8.1.2 用户及内核模式	134
5.2.10 Set Interface (设置接口) 请求	87	8.2 分层结构内部	136
5.2.11 Synch Frame (帧同步) 请求	87	8.2.1 应用程序	136
5.2.12 Set SEL (设定选择) 请求	87	8.2.2 用户模式下的用户驱动程序	137
5.2.13 Set Isochronous Delay (设定等时延时) 请求	88	8.2.3 内核模式下的用户驱动程序	137
5.3 其他请求	88	8.2.4 总线和主机控制器驱动程序	138
5.3.1 类专属请求	88	8.3 编写驱动程序	140
5.3.2 厂商定义的请求	88	8.3.1 内核模式的驱动程序	140
第 6 章 芯片选择	89	8.3.2 用户模式的驱动程序	140
6.1 USB 设备的组成部分	89	8.3.3 测试工具	141
6.1.1 USB2.0 控制器内部结构	89	8.4 GUID 的使用	141
6.1.2 其他的设备组件	90	8.4.1 设备设置 GUID	141
6.2 简化设备开发	92	8.4.2 设备接口 GUID	142
6.2.1 设备要求	92	第 9 章 为设备匹配驱动程序	143
6.2.2 芯片文档	93	9.1 使用设备管理器	143
6.2.3 驱动程序的选择	93	9.1.1 查看设备	143
6.2.4 调试工具	93	9.1.2 属性页	144
6.3 USB 微控制器	95	9.2 注册表中的设备信息	145

9.2.1 Hardware 键值	145	11.4.2 工具	190
9.2.2 Class 键值	146	第 12 章 人机接口设备：报告	191
9.2.3 Driver 键值	147	12.1 HID 报告的数据结构	191
9.2.4 Service 键值	147	12.1.1 使用 HID 描述符编辑工具	191
9.3 深入 INF 文件	148	12.1.2 控制项及数据项目的值	192
9.3.1 结构及语法	150	12.1.3 项目格式	192
9.3.2 设备专属取值	151	12.2 主项目	193
9.4 使用设备标识字符串	151	12.2.1 输入项目，输出项目及特征项目	193
9.4.1 找到匹配	153	12.2.2 Collection 项目	196
9.4.2 何时提供 INF 文件	153	12.3 全局项目类型	196
9.5 工具及诊断协助	154	12.3.1 识别报告	196
9.5.1 使用 INF 文件的技巧	154	12.3.2 描述数据用途	197
9.5.2 用户可见到什么	155	12.3.3 单位转换	198
第 10 章 设备检测	156	12.3.4 转换原始数据	199
10.1 关于 Windows API 函数调用的 简要指导	156	12.3.5 描述数据大小及数据格式	201
10.1.1 托管代码与非托管代码	156	12.3.6 储存和恢复全局项目	201
10.1.2 管理数据	159	12.4 局部项目类型	201
10.2 找到您的设备	161	12.4.1 物理描述符	203
10.2.1 获取设备接口 GUID	163	12.4.2 填充	203
10.2.2 请求指向设备信息集的指针	163	第 13 章 人机接口设备：主机 应用程序	204
10.2.3 标识设备接口	164	13.1 HID API 函数	204
10.2.4 请求带有设备类路径名的结构体	166	13.1.1 请求 HID 信息	204
10.2.5 提取设备路径名	168	13.1.2 报告发送与接收	205
10.2.6 结束通信	168	13.1.3 报告数据提供及使用	206
10.3 获取句柄	169	13.1.4 HID 通信管理	207
10.3.1 请求通信句柄	169	13.2 设备识别	207
10.3.2 关闭句柄	171	13.2.1 读取 Vendor ID 及 Product ID	207
10.4 检测设备连接和移除	171	13.2.2 获取设备功能指针	209
10.4.1 关于设备通知	171	13.2.3 获取设备功能	210
10.4.2 请求设备通知	171	13.2.4 获取按钮及数值的功能	211
10.4.3 记录设备变化信息	174	13.3 报告发送与接收	212
10.4.4 读取设备变化消息	175	13.3.1 向设备发送输出报告	212
10.4.5 收回消息中的设备路径名	176	13.3.2 由设备读取输入报告	214
10.4.6 停止设备通知	178	13.3.3 向设备写入特征报告	220
第 11 章 人机接口设备：使用控制和 中断传输	180	13.3.4 从设备读取特征报告	221
11.1 什么是 HID	180	13.3.5 关闭通信	222
11.1.1 硬件要求	181	第 14 章 利用 WinUSB 的厂商 定义函数	224
11.1.2 固件要求	182	14.1 功能与局限	224
11.2 描述字	182	14.1.1 设备要求	224
11.2.1 HID 接口	185	14.1.2 主机要求	224
11.2.2 HID 类描述字	185	14.1.3 设备固件	224
11.2.3 报告描述字	186	14.1.4 分配 WinUSB 驱动程序	226
11.3 HID 专属请求	187	14.2 访问设备	227
11.3.1 Get Report 请求	187	14.2.1 获取 WinUSB 句柄	228
11.3.2 Get Idle 请求	188	14.2.2 请求接口描述符	229
11.3.3 Get Protocol 请求	188	14.2.3 端点识别	230
11.3.4 Set Report 请求	188	14.2.4 配置管道策略	233
11.3.5 Set Idle 请求	189	14.2.5 利用批量传输及中断传输写入数据	237
11.3.6 Set Protocol 请求	189	14.2.6 利用批量传输及中断传输读取数据	238
11.4 数据传输	189	14.2.7 使用厂商定义的控制传输	239
11.4.1 编写固件	189		

14.2.8 关闭通信	242	18.1.6 包的格式	281
第 15 章 集线器	243	18.1.7 包间延迟	282
15.1 USB 2.0	243	18.1.8 测试模型	282
15.1.1 集线器转发器	244	18.2 超速通信	283
15.1.2 事务转换器	245	18.2.1 数据加扰	283
15.1.3 集线器控制器	249	18.2.2 编码	283
15.1.4 传输速率	249	18.2.3 链路层	283
15.1.5 维持已激活链接	250	18.2.4 复位	284
15.2 USB 3.0	250	第 19 章 电气与机械接口	285
15.2.1 总线速率	251	19.1 USB 2.0 收发器	285
15.2.2 USB 3.0 的构成	251	19.1.1 线缆段	285
15.2.3 通信管理	251	19.1.2 低速和全速收发器	286
15.3 集线器类	252	19.1.3 高速收发器	288
15.3.1 集线器描述符	252	19.1.4 信号电压	291
15.3.2 集线器类请求	252	19.2 USB 2.0 线缆	292
15.3.3 端口指示器	252	19.2.1 导线	292
第 16 章 电源管理	253	19.2.2 连接器	293
16.1 电源选项	253	19.2.3 可拆卸及不可分离线缆	294
16.1.1 电压	253	19.2.4 线缆长度	294
16.1.2 使用总线电源	254	19.2.5 总线长度	295
16.1.3 电源需求	255	19.2.6 芯片间连接	295
16.1.4 通知主机	255	19.3 USB 3.0	296
16.1.5 电池充电	256	19.3.1 发送器和接收器	296
16.2 集线器电源	257	19.3.2 线缆	296
16.2.1 电源	257	19.4 确保信号质量	299
16.2.2 电流超载保护	258	19.4.1 噪声来源	299
16.2.3 功率切换	259	19.4.2 对称传输线	300
16.3 省电模式	259	19.4.3 双绞线	300
16.3.1 USB 2.0 链路电源管理	259	19.4.4 屏蔽	301
16.3.2 挂起状态	259	19.4.5 边沿变化率	301
16.3.3 休眠状态	260	19.4.6 被隔离的接口	301
16.3.4 超速通信电源管理	261	19.5 无线方式实现 USB	301
16.3.5 Windows 操作系统下的电源管理	264	19.5.1 无线 USB 认证	302
第 17 章 测试与调试	266	19.5.2 Cypress 无线 USB	302
17.1 工具	266	19.5.3 其他选择	303
17.1.1 硬件协议分析器	267	第 20 章 嵌入式系统主机	305
17.1.2 软件协议分析器	268	20.1 USB OTG	305
17.1.3 流量发生器	269	20.1.1 能力与局限	305
17.2 测试	269	20.1.2 OTG 连接器	306
17.2.1 合规性	270	20.1.3 A 型设备及 B 型设备	306
17.2.2 Windows 徽标	273	20.1.4 OTG 设备要求	306
第 18 章 总线上的包	276	20.1.5 OTG 描述符	310
18.1 USB 2.0 通信	276	20.1.6 HNP 特征代码	310
18.1.1 低速、全速总线状态	276	20.2 其他主机选项	311
18.1.2 高速总线状态	277	20.2.1 嵌入式主机要求	311
18.1.3 数据编码	279	20.2.2 设备端口	312
18.1.4 保持同步	279	20.3 控制器芯片	312
18.1.5 精确计时	280	20.3.1 微控制器	312
		20.3.2 接口芯片	313

第 1 章

USB 基础

凭借每年逾 20 亿新单元被安装的速度，USB 已成为最成功的个人计算机接口。如今，每台计算机都有 USB 端口，可与键盘、鼠标、游戏手柄、扫描仪、照相机、打印机、驱动器以及更多设备相连。USB 可靠、高速、用途广泛、省电、价格便宜并且得到了主要的操作系统的支持。USB 3.0 还新增了超高速总线，这意味着 USB 将继续作为（不断增多的）周边设备接口的主要选择。

本章介绍 USB 的优点、局限、接口历史和近期的功能增强，此外还讲到了在用 USB 接口对设备进行设计和编程时将涉及的事情。

1.1 USB 的用途和局限

任何时候，若您想用计算机与外部设备通信，USB 都可成为一种解决方案。像指纹读取器这样的内部设备就可以使用 USB 接口。USB 接口既适合于大规模生产的消费类电子产品，也可用于专门设计的小批量产品，甚至那些独一无二的工程。

一种接口若想成功，须使两类受众满意：设备用户和设计硬件、编写设备通信相关代码的开发者。USB 接口具有使他们都满意的特性。

1.1.1 给用户带来的好处

从用户的角度来说，USB 拥有使用方便、数据传输快速可靠、成本低和省电等优点。表 1-1 给出了 USB 与其他接口的比较。

表 1-1 USB 比其他接口更为灵活，其他接口常只针对特殊应用

接 口	类 型	最大设备连接 数目（包括 PC）	最大距离（ft）	最高速率（bit/s）	典型应用
USB 3.0	双单工串行	127（每条总线）	9（典型）（通过 5 个 集线器可扩展为 49）	5G	大容量存储、视频
USB 2.0	半双工串行	127（每条总线）	16（通过 5 个集线器 可扩展为 98）	1.5M, 12M, 480M	键盘、鼠标、驱动器、 话筒、打印机、 照相机

续表

接 口	类 型	最大设备连接 数目（包括 PC）	最大距离（ft）	最高速率（bit/s）	典型应用
eSATA	串行	2（端口倍增支持为 16）	6	3G	驱动器
Ethernet	串行	1024	1600	10G	通用网络通信
IEEE-1394(FireWire 800)	串行	64	300	3.2G	视频、大容量存储
IEEE-488(GBIB)	并行	15	60	8M	仪表
I ² C	同步串行	40	18	3.4M	微控制器通信
Microwave	同步串行	8	10	2M	微控制器通信
MIDI	串行电流环	2（在流通过模式下更多）	50	31.5k	音乐、播放控制
并行打印机端口	并行	2（在菊花链的支持下为 8）	10~30	8M	打印机、扫描仪、磁盘驱动器
RS-232(EIA/TIA-232)	异步串行	2	50~100	20k（在硬件的帮助下可达 115k）	调制解调器、鼠标、乐器
RS-485(TIA/EIA-485)	异步串行	32 个单位负载（有些芯片支持多达 256 个设备）	4000	10M	数据获取及控制系统
SPI	同步串行	8	10	2.1M	微控制器通信

1. 使用方便

使用方便是 USB 的主要设计目标，这使 USB 适合于如下接口用途。

一个接口用于多种设备连接。对于标准计算机周边设备的功能，USB 的适用性足够强。换句话说，单个 USB 接口可服务多种设备，用户无需针对不同周边设备的功能采用不同类型的连接器和线缆。

自动配置。只要用户把 USB 设备连接到计算机，操作系统就会自动检测设备，并装载合适的软件驱动程序。设备首次连接时，操作系统可能会提示用户插入含有驱动程序软件的磁盘。但这之后设备使用时，驱动程序都将自行安装。且设备工作之前，用户无须重启计算机。

连接简单。典型的计算机拥有多个 USB 端口和集线器，这使计算机连接端口变得很简单，而且此过程不需打开计算机。

线缆方便。USB 连接器与其他接口（诸如 RS-232）的连接器相比，小而紧凑。为保证操作可靠，USB 规范定义了线缆的电器要求。随总线速度不同，线缆最长可达 5m。通过使用集线器（同样也受总线速度的影响），可将设备与主机间的最远距离提高到 30m。

无线连接（可选）。USB 最初是一种有线接口，但当前的技术水平已允许与 USB 设备进行无线通信。

支持热插拔。用户可在任何时候（无论系统和设备是否通电），连接和断开 USB 设备，且不会损坏计算机和设备。设备连接时，操作系统将对其进行检测，并使设备就绪、等待工作。

无须用户设置。USB 设备没有类似端口地址编写和中断请求（IRQ）线的用户可选设置，故用户无须通过使用跳线设定和配置用途来让设备运行。

多数情况下不需要电源。USB 接口中包含了电源线和地线，可从计算机或集线器上获取标

称值为+5V 的电源。电能需求在 500mA 以下（对于 USB2.0）或 900mA 以下（对于 USB3.0）的设备均可由总线供电，无需专门的电源。相反，使用其他接口设备则不得不对其提供内部或外部电源。

2. 多种传输速率

USB 支持 4 种总线速率：超高速（5Gbit/s）、高速（480 Mbit/s）、全速（12 Mbit/s）和低速（1.5 Mbit/s）。超高速状态要求主机内含有 USB 3.0 控制器。而 USB 2.0 主机控制器则支持低速、全速和高速三种模式。

总线速率（bus speed）描述的是总线上信息流动的速率。除应用程序数据外，总线还能够携带状态、控制及检错信息。而且，多个设备还可能共享一条总线（这使每个设备所获得的速率更小）。因此，单个设备的数据速率要小于总线速率。USB 协议所支持的（单个设备的）数据传输，在超高速模式下约为 400Mbit/s、高速模式下约为 53 Mbit/s、全速模式下约为 1.2 Mbit/s、低速模式为 800 bit/s。来自硬件和软件的限制还可能导致更低的实际速度。

USB 1.0 规范只定义了低速和全速模式。全速模式是为大部分原来使用 RS-232（串行）和并行端口的周边设备而准备的。全速数据传输的速率与早期接口的速率相当。鼠标就趋向于使用低速传输模式，其相对宽松的线缆要求允许更为灵活的线缆选择。低速设备制造成本低的部分原因，就是其更加便宜的线缆。USB 2.0 发布后，高速传输模式成为新的选择，而之后的 USB 3.0 又定义了超高速传输模式。

3. 可靠性高

USB 的可靠性缘于其硬件和协议。USB 对驱动程序、接收端和线缆的硬件规范保证了一个平静的接口电器环境，避免了导致数据错误的大部分噪声干扰。USB 协议使设备和主机能够在接收数据时进行错误检测，并通知发送端以便重新发送。硬件在执行检测、确认（通知）和重传的过程中，无需软件或用户支持。

4. 价格低廉

由于主机提供了大部分接口控制信息，USB 设备所使用的部件都不贵。一个带有 USB 接口的设备也许只有与使用其他接口的等效设备相等甚至可能更少的花费。

5. 省电

USB 的省电电路和省电协议降低了设备功耗，同时保证了设备能在需要时完成通信。减少功耗即节约了成本，这对环境有益，且利于电池供电设备在两次充电之间工作得更持久。

1.1.2 给开发者带来的好处

许多带给用户的好处也同时方便了开发人员。例如，USB 的线缆标准和错误检查，这意味着开发者不必再为确定线缆特征或在软件上提供错误检查手段而费心。

USB 的其他优点还对那些挑选元件、设计设备电路的硬件工程师以及编写嵌入设备中固件和设备通信软件的编程人员有很大的帮助。

USB 的这些好处，皆缘自于那些已被构建到 USB 协议中的灵活性、控制器芯片和操作系统内的支持，以及从 USB 实践者论坛（USB-IF）上所获得的支持。

1. 用途多样

USB 的 4 种传输类型和 4 种速度，使其适用于多种周边设备。USB 的多种传输使其在有

或没有时间限制的条件下，都可完成大块和小块数据的交换。对于不容许延迟的数据，USB 可保证其带宽。这些能力特别受 Windows 系统的欢迎，因为对周边设备完成实时访问经常是个挑战。尽管操作系统、设备驱动程序以及应用程序软件不可避免地会引入延迟，USB 仍可使完成近于实时传输的任务变得相当轻松，甚至在台式机系统上也是如此。

与其他接口不同，USB 没有给信号线指定特定功能，也没有做出其他有关系统如何使用接口的假设。如计算机并行端口上的状态和控制线，就被定义用于与行式打印机通信；而 USB 没有做这类假设，因而适合于几乎任何类型的周边设备。

为与诸如打印机、键盘和驱动器等周边设备通信，USB 类规定了设备要求和协议。开发者可通过编程，使设备符合某个类规范，而不必从零开始，对所有的代码都重新开发。

2. 操作系统支持

本书集中于讨论个人计算机 Windows 操作系统上的编程，其他计算机和操作系统（包括 Linux 系统和苹果计算机的 Macintosh 系统）同样有 USB 支持。某些实时内核也支持 USB。支持 USB 的操作系统起码要做到以下三点。

- 设备连接或移除系统时，对其进行检测。
- 与新连接的设备通信，了解如何与其交换数据。
- 提供一种机制，使得软件驱动程序能够在 USB 硬件与想要访问 USB 周边设备的应用程序之间传递通信。

在更高的层次上，操作系统支持还意味着含有类驱动程序，它能使应用程序访问特定类型的设备。如果操作系统不含某个设备专门的驱动程序，则必须由设备厂商提供。

微软公司一直在改进和增加 Windows 中所包含的类驱动程序。所支持的设备类型包括人机接口设备（键盘、鼠标、游戏手柄等），话筒和其他声音设备、调制解调器、驱动器、静态图像和视频摄像机、扫描仪、指纹识别器和智能卡读取器。过滤器驱动程序可支持设备专属特征及类内能力。应用程序使用“应用程序编程接口”（API）功能或其他软件组件，经由设备驱动程序访问设备。

拥有厂商专属功能的设备，有时也可使用 Windows 所支持的设备类，如通信设备类或人机接口设备类。厂商专属功能设备的其他选择包括：微软 WinUSB 驱动程序和其他来源的通用驱动程序（generic driver）。部分芯片公司还向开发者提供了可在其芯片上使用的驱动程序。

Windows 系统下，USB 设备驱动程序的编写者可使用微软的 Windows 驱动程序构建（WDF）模型。WDF 提供了程序的框架，简化了编写驱动程序的复杂度。

3. 设备支持

在设备方面，硬件必须含有一个能够处理 USB 通信的控制器芯片。它负责对确认和配置设备的请求进行响应，也负责对总线上其他数据的读写。有些控制器会完全依靠硬件来实现某些功能。

许多 USB 控制器都是基于如 Intel 公司的 8051 或 Microchip Technology 的 PIC 的通用微控制器架构而开发的，并且添加了支持 USB 通信的硬件。其他控制器则不含 CPU，而是一个与外部微控制器相连的串行或并行接口。如果您已经熟悉了某种芯片架构，且这个芯片拥有支持 USB 功能的衍生产品，您无须再去学习另一种新架构。大部分芯片公司都会提供有助于工程开发的代码示例。

4. USB 实践者论坛

USB 实践者论坛 (USB Implementers Forum) 或称 USB-IF (www.usb.org) 是由开发 USB 规范的公司建立的非营利性组织。

USB-IF 的使命是支持 USB 技术的进步和应用。为此 USB-IF 提供了 USB 的相关信息、工具和测试支持。信息包括规范文件、白皮书、FAQ 和网上论坛。工具是对开发和测试产品有帮助的软件和硬件。测试支持包括确认操作满足 USB 规范的符合性测试, 以及获准开发者参加专题研讨会 (测试和认证产品, 并准予使用 USB 图标)。

1.1.3 USB 不能做什么

USB 的诸多优势意味着它是实现许多设备接口的不错的候选, 但仅 USB 接口还不能完成所有任务。

1. USB 接口的局限

USB 的限制包括距离有限、不支持类间 (主机和主机间、设备与设备间) 通信或广播以及缺乏对老式硬件和操作系统的支持。

距离。USB 被定位为一种台式机扩展总线, 即设备处于距离相对较近的周边范围内。其他接口, 包括 RS-232、RS-485、IEEE-1394b 和 Ethernet, 都允许使用更长的线缆。为延长设备与主机间的距离, 可使 USB 接到一个具有“桥梁”功能的周边设备, 作为连接终端电路的接口。

点对点通信。每个 USB 通信都发生在主机计算机和设备之间 (USB 3.0 提供的一种备选方案是个例外)。这里, 主机指个人电脑或其他含有主机控制器硬件的计算机。设备则应含有设备控制器硬件。主机之间、设备之间都不能直接相互通话。有些接口, 如 IEEE-1394, 则允许设备到设备的直接通信。

USB 提供了一种部分解决方案, 即使用 USB On-The-Go 选项。On-The-Go 设备可以既作为设备, 又作为有限能力的主机 (可与其他设备通信)。

两个 USB 主机可经由共享缓冲区的两个 USB 设备 (组成“线缆桥”) 进行相互通信。USB 3.0 定义了一条新线缆, 用于超高速传输。配合相关驱动程序的支持, 这条新增线缆可支持主机到主机的通信。

广播。USB 不支持同时向多个设备传送数据 (USB 3.0 的时间戳数据包除外)。主机必须分别向每个设备发送数据。如果您需要广播功能, 可使用 IEEE-1394 或以太接口。

传统硬件。老式的“传统”计算机和周边设备都不含 USB 端口。支持老式设备的话题已经不再热门了, 因为老式系统已基本被淘汰。如果您需要将传统周边设备连接到 USB 端口上, 解决方法是使用智能适配器, 在 USB 和老式接口之间作转换。从几个来源可获得用于带有 RS-232、RS-485 及并行端口的周边设备的适配器。适配器只对使用了其所支持协议的设备有效。例如, 大部分并行端口适配器只支持同打印机的通信, 同其他采用并行端口的周边设备则无法通信。RS-232 适配器则只对 RS-232 设备都有效。

若想在支持 USB 的计算机上使用 USB 设备, 解决办法是在此计算机上添加 USB 性能。为此, 您需添加 USB 主机控制器硬件, 并使用支持 USB 的操作系统。硬件则通过插在 PCI 插槽或替代性主板上的扩展卡变为 USB 可用硬件。对于 Windows 系统, 只有 Windows 98 或后续的版本才支持 USB。

如果通过升级计算机来支持 USB 的办法不可行，则可考虑使用一个能够将周边设备的 USB 接口转换到计算机的 RS-232、并行接口或其他接口的适配器。若您的计算机有老式接口，很少使用适配器来解决此问题，因为含有主机控制器硬件和代码的适配器市场有限，设计、制造起来特别昂贵。

即使在新硬件系统上，用户也很少使用较旧的操作系统（诸如 DOS）来运行应用程序。主要是因为缺少驱动程序，操作系统无法访问 USB 设备。虽然也可为 DOS 编写 USB 驱动程序，但没有几家设备厂商真正提供。鼠标和键盘则是个例外，系统 BIOS 一般都支持这两个设备，以保证它们在任何情况下（无论是从 DOS 环境启动还是在启动计算机时所看到的 BIOS 屏幕）都可用。

2. 开发者面临的挑战

USB 开发者所面临的挑战包括复杂的协议，操作系统不支持某些应用程序，以及（对小规模产品开发者）需要获取厂商 ID。

协议复杂度。USB 设备是一种必须对总线上的请求和其他事件进行响应的智能设备。完成 USB 通信所需的固件支持有多有少，不同的控制器芯片之间存在差异。多数情况下，为 USB 设备编程，您需要熟悉 USB 总线交换数据的协议。在主机端，设备驱动程序使应用程序的编写人员不必知道太多有关协议和硬件接口的低层次细节。但设备驱动程序的编写者则需要熟悉 USB 协议。

相反，有些旧式接口则能与由软件直接寻址的简单电路相连。例如，计算机最初的并行打印机端口就是一种数字输入输出接口；您可连接继电器、开关和模数转换器之类的输入输出电路，无须使设备端拥有计算机的智能。通过驱动程序（负责对端口的访问），应用程序可监视和控制每个端口上的数据位。

USB 是一种带有确定协议的共享总线，操作系统阻止了应用程序直接访问硬件。为访问某个 USB 设备，应用程序必须先与某个类或设备驱动程序通信，以此完成与控制总线通信的更低层次 USB 驱动程序的通信。设备必须支持那些使计算机能够对设备进行检测、确认和通信的协议。

操作系统支持的不演化。Windows 内的类驱动程序使应用程序能够与多种设备通信。通常，您可在设计产品时使用某个 Windows 所提供的驱动程序。或者也可使用或改编由芯片公司或其他来源提供的驱动程序。若需自行编写驱动程序，那么开发过程中，第三方驱动程序工具包也许会对您有帮助。

费用。USB-IF 网站提供了 USB 规范、相关文件、符合性测试所需软件和更多的免费信息。任何人都可开发 USB 软件而不必支付许可费用。所有的 USB 设备都含有厂商 ID 和产品 ID，以便使操作系统能够确认设备。写此书时，USB-IF 会对每个厂商 ID 的使用权索取 2000 美元的管理费。厂商 ID 持有者可指定产品 ID。加入 USB-IF（每年 4000 美元会费），您便获得一个厂商 ID，同时还有其他权利，如参加符合性论坛。

没经过验证性测试、不显示 USB-IF 标志的设备有低成本的实现途径。有些芯片公司，包括 Future Technology Devices 国际有限公司（FTDI）和 Microchip Technology 公司，会为用户指定一系列的产品 ID（在其产品中使用），并采取此公司的厂商 ID，一般不收取费用。所有 USB 通信都以硬件方式实现的芯片可使用嵌入在硬件里厂商的 ID 和产品 ID，例如 FTDI 的 USB