



# 教你玩USB

【博客藏经阁丛书】

刘荣 编著  
[网名 电脑圈]



 北京航空航天大学出版社



TP334  
31

TP334  
31

【博客藏经阁丛书】

附光盘

# 圈圈教你玩 USB

刘 荣 [网名 电脑圈圈] 编著

北京航空航天大学出版社

2009年1月第1版

北京航空航天大学出版社

## 内 容 简 介

【计算机应用案例】

通过 U 盘、USB 鼠标、USB 键盘、USB MIDI 键盘、USB 转串口、自定义的 USB HID 设备和自定义的 USB 设备等几个具体的 USB 例子，一步步讲解 USB 设备及驱动程序和应用程序开发的详细过程和步骤。最后两章介绍 USB WDM 驱动开发，并给出一个简单的 USB 驱动和 USB 上层过滤驱动的实例。

本书附带的光盘中有 USB 学习板的电路图以及所有实验的完整源代码包(C/C++语言)。

本书的读者对象主要是 USB 设备与驱动设计的初学者和提高者，以及所有对圈圈的支持者。

### 图书在版编目(CIP)数据

圈圈教你玩 USB/刘荣编著. —北京:北京航空航天大学出版社, 2009. 1

ISBN 978-7-81124-600-1

I. 圈… II. 刘… III. 电子计算机—接口 IV. TP334

中国版本图书馆 CIP 数据核字(2008)第 159685 号

© 2009, 北京航空航天大学出版社, 版权所有。

未经本书出版者书面许可, 任何单位和个人不得以任何形式或手段复制本书及其所附光盘内容。侵权必究。

### 圈圈教你玩 USB

刘 荣[网名 电脑圈圈] 编著

责任编辑 张冀青

\*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100191) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail:emsbook@gmail.com

涿州市新华印刷有限公司印装 各地书店经销

\*

开本:787 mm×960 mm 1/16 印张:20.5 字数:459 千字

2009 年 1 月第 1 版 2009 年 1 月第 1 次印刷 印数:5 000 册

ISBN 978-7-81124-600-1 定价:39.00 元(含光盘+PCB 板)

# 前言

USB 作为一种计算机总线技术,在如今的个人计算机上已经是必不可少的了。看看我们身边的计算机外部设备,有多少是通过 USB 口与计算机连接的?鼠标、键盘、游戏手柄、打印机、扫描仪、MP3、数码相机、U 盘、移动硬盘及移动光驱等,另外,还有一些我们不太常见的 USB 设备,例如一些具有 USB 口的仪器仪表、开发用的调试器、烧录机、USB 网卡、USB 耳机、USB 话筒及 USB 电话,甚至一些移动电话(手机)也具备 USB 口。总之,只要是与计算机通信的外部设备,似乎都可以用 USB 来连接,这足见 USB 之强大。

USB 之所以使用得如此广泛,是因为它具有连接简单、速度快、可扩展性强、支持热插拔操作和标准统一等特点。由于 USB 协议详细地规定了各种参数以及数据结构、格式,因而使得各厂生产出来的设备都能够很好地相互兼容。不过,这却给 USB 设备开发者带来了一些麻烦。USB 设备开发者首先要很清楚 USB 协议才能开发出符合协议的 USB 设备,然而 USB 协议本身是一个比较复杂、庞大的系统,再加上众多的子类协议,使得很多设计者望而生畏,或者时间上不允许。所以有一些小公司将 USB 部分外包或者使用别人现成的 USB 模块来加快开发进度。如果你会开发 USB 相关设备的话,可能会获得不少机会哦😊。不过,虽然圈圈(就是笔者我啦,读者一定要牢记这点,不然把你弄晕了我可不负责)会一点 USB,但目前还是穷光蛋一个😞。当然,你也可以学圈圈这样,写本书出来忽悠忽悠。

虽然 USB 这么好用,但是教你如何设计 USB 设备的书在市面上却是少见,大部分都是对 USB 协议翻译,只有少量的内容是实际开发的内容和代码。圈圈从 2003 年年底(大二)开始学习 USB,花了约一年的时间(主要是利用课余时间)阅读了一些 USB 书籍和文档后,才真正开始动手做自己的第一个 USB 设备:一个基于 AT89C52+PDIUSB12 的假 U 盘。圈圈自认自己资质不差,却让一个小小的 USB 困扰了这么久,我想除了自己的原因之外,还跟 USB 协议本身的复杂性和没有一些好的书籍有很大关系。在整个 USB 学习过程中主要都是靠自己慢慢摸索,走了一些弯路。现在回过头再来看看,如果当初能有一个整体的认识,按照合理的步骤来操作,就不会走这么多弯路了。因此圈圈意识到,迫切需要写一本能让 USB 初学者少走弯路、快速入门和上手的书籍。圈圈写这本书,是真心地想要更多的人能够学会 USB,会开发 USB 设备。本书并不是对 USB 协议简单地进行翻译,而是尽量用圈圈自己的语言来进行描述。所以,本书的一大特点就是语言有些口语化,逻辑不是太严密,有些语句也可能存在

着错误。但我想这样读起来也许会更轻松些,太严谨、太严密的描述,可能会显得有些乏味。另外,有些地方可能会存在一些重复性描述,主要是想方便理解和加深印象。

本书通过 U 盘、USB 鼠标、USB 键盘、USB MIDI 键盘、USB 转串口、自定义的 USB HID 设备和自定义的 USB 设备等几个具体的 USB 例子,来介绍 USB 设备设计的具体流程。提到自定义的 USB 设备,就不可避免地要提到自己开发 USB 的驱动程序。本书最后两章简单地介绍了 USB WDM 驱动的开发,并给出一个简单的 USB 驱动和一个 USB 下层过滤驱动的实例。驱动程序开发更深层次的研究不属本书范畴,留给读者自行深究。另外,本书也包括了一些圈圈在 USB 的学习和实际设计过程中总结的一些电路设计和程序设计的经验及方法,希望能够帮助大家更好地学习和设计 USB 系统。

本书主要面向的读者是刚接触 USB 开发的电子设计人员,需要有一定的电子技术、计算机技术和 C 语言基础。对于已经熟悉 USB 的读者意义不大,应以官方的数据手册和文档为主。本书为了让读者快速入门,可能会对一些模型做简化处理,也可能有一些地方是圈圈本身理解偏差或者错误的,当本书与官方的协议和文档不一致或者冲突时,以官方文档为准。

书中的实例,都是基于圈圈设计的 USB 学习板之上的。本书附带的光盘中该学习板的电路图(pdf 格式)以及所有实验的完整源代码包,电路图和实验的代码包也可以到圈圈的 USB 小组或者博客中下载。

作为一个人或一本书,出错之处在所难免,如果大家发现有错误的地方,请告诉圈圈一声。你可以在圈圈的 USB 专区里给圈圈留言,也可以去个人博客里留言。圈圈的官方博客地址: <http://blog.ednchina.com/computer00/>或 <http://computer00.2lic.org>。注意是电脑圈圈,是两个数字 0,而不是字母 O。如果你记不住这些地址也没关系,直接去网上搜索“电脑圈圈的 USB 专区”或者“电脑圈圈”,也可找到它们。

在此感谢北京航空航天大学出版社嵌入式系统事业部主任胡晓柏先生对出版本书的关心和支持;感谢同学以及网友对本书的建议和支持;感谢家人对我的支持和理解,尤其是要感谢老婆,为了写书,少了很多陪她的时间,谢谢她的支持、理解和爱;同时也要感谢购买本书的读者——你。

2009 年 1 月  
于广州

电脑圈圈

# 目 录

.....	.....	1.10.1
.....	.....	1.10.2
.....	.....	1.10.3
.....	.....	1.10.4
.....	.....	1.10.5
.....	.....	1.10.6
<b>第 1 章 USB 概述及协议基础</b>		
1.1	USB 是什么	1
1.2	USB 的特点	1
1.3	USB 的拓扑结构	2
1.4	USB 的电气特性	5
1.5	USB 的线缆、插头及插座	6
1.6	USB 的插入检测机制	8
1.7	USB 的描述符及其之间的关系	10
1.8	USB 设备的枚举过程	11
1.9	USB 的包结构及传输过程	13
1.9.1	USB 的包结构及包的分类	13
1.9.2	令牌包	15
1.9.3	数据包	16
1.9.4	握手包	17
1.9.5	特殊包	17
1.9.6	如何处理数据包	18
1.10	USB 的四种传输类型	19
1.10.1	USB 事务	19
1.10.2	批量传输	19
1.10.3	中断传输	21
1.10.4	等时传输	22
1.10.5	控制传输	22
1.10.6	端点类型与传输类型的关系	24

# 目 录

1.10.7 传输类型与端点支持的最大包长 .....	24
1.11 本章小结 .....	24

## 第 2 章 硬件系统设计

2.1 方案及芯片的选定 .....	25
2.2 D12 引脚功能说明 .....	26
2.3 D12 与 AT89S52 的连接 .....	30
2.4 串口部分电路 .....	32
2.5 按键部分 .....	33
2.6 指示灯部分 .....	34
2.7 IDE 接口部分 .....	34
2.8 单片机部分 .....	35
2.9 元件安装 .....	35
2.10 电路调试 .....	38
2.11 测试程序的编写和调试 .....	39
2.11.1 建立一个工程 .....	40
2.11.2 为工程添加源文件 .....	41
2.11.3 KEIL 工具栏及仿真介绍 .....	43
2.11.4 按键驱动的编写 .....	46
2.11.5 串口驱动的编写 .....	52
2.11.6 PDIUSB12 读写函数及读 ID 的实现 .....	57
2.12 本章小结 .....	62

## 第 3 章 USB 鼠标的实现

3.1 USB 鼠标工程的建立 .....	63
3.2 USB 的断开与连接 .....	63
3.3 USB 中断的处理 .....	67
3.4 读取从主机发送到端点 0 的数据 .....	68
3.5 USB 标准请求 .....	75
3.5.1 USB 标准设备请求的结构 .....	75
3.5.2 GET_DESCRIPTOR 请求 .....	77
3.5.3 SET_ADDRESS 请求 .....	78
3.5.4 SET_CONFIGURATION 请求 .....	78
3.6 设备描述符的实现 .....	79

3.7	设备描述符的返回	83
3.8	设置地址请求的处理	93
3.9	配置描述符集合的结构	95
3.9.1	配置描述符的结构	95
3.9.2	接口描述符的结构	96
3.9.3	端点描述符的结构	97
3.9.4	HID 描述符的结构	98
3.10	配置描述符集合的实现以及返回	99
3.11	字符串及语言 ID 请求的实现	103
3.12	设置配置请求的实现	109
3.13	报告描述符的结构及实现	112
3.14	报告的返回	118
3.15	Bus Hound 工具的简介	122
3.16	本章小结	124
<b>第 4 章</b>	<b>USB 键盘的实现</b>	
4.1	USB 键盘工程的建立	125
4.2	设备描述符的实现	125
4.3	配置描述符集合的实现	126
4.3.1	配置描述符	126
4.3.2	接口描述符	127
4.3.3	HID 描述符	127
4.3.4	端点描述符	127
4.4	字符串描述符	130
4.5	报告描述符	130
4.6	输入和输出报告的实现	133
4.7	USB 键盘实例的测试	136
4.8	再谈 USB HID 的报告描述符	138
4.9	带鼠标功能的 USB 键盘(方法一)	140
4.10	带鼠标功能的 USB 键盘(方法二)	146
4.11	多媒体 USB 键盘	154
4.12	本章小结	160



## 第 5 章 用户自定义的 USB HID 设备

5.1	MyUsbHid 工程的建立	161
5.2	描述符的修改	161
5.3	报告的实现	163
5.4	对用户自定义的 USB HID 设备的访问	165
5.5	访问 HID 设备时所用到的相关函数	166
5.5.1	获取 HID 设备的接口类 GUID 的函数	166
5.5.2	获取指定类的所有设备信息集合的函数	166
5.5.3	从设备信息集合中获取一个设备接口信息的函数	167
5.5.4	获取指定设备接口详细信息的函数	168
5.5.5	打开设备的函数	169
5.5.6	获取 HID 设备属性的函数	170
5.5.7	从设备读取数据的函数	170
5.5.8	往设备写数据的函数	171
5.5.9	通过控制端点 0 读取报告的函数	171
5.5.10	通过控制端点 0 发送报告的函数	171
5.5.11	关闭句柄的函数	172
5.5.12	需要包含的库文件	172
5.6	访问 USB HID 设备的上位机软件的实现	172
5.6.1	上位机程序编写的思路	172
5.6.2	查找及打开 HID 设备的代码	173
5.6.3	读输入报告线程的代码	178
5.6.4	写输出报告的代码(发送 LED 的状态)	181
5.6.5	写输出报告线程的代码	183
5.6.6	线程的创建以及设备插拔事件的注册	184
5.6.7	对设备状态改变事件的处理	186
5.7	软件界面以及使用方法	187
5.8	本章小结	188

## 第 6 章 USB 转串口

6.1	串口家族历史	189
6.2	串口接头的引脚分布及功能	189
6.3	USB 转串口的实现方法	190
6.4	设备描述符	191
6.5	字符串描述符	192

6.6	配置描述符集合	192
6.6.1	配置描述符	193
6.6.2	CDC 接口描述符	193
6.6.3	类特殊接口描述符——功能描述符	193
6.6.4	接口 0(CDC 接口)的端点描述符	196
6.6.5	数据类接口的接口描述符	196
6.6.6	接口 1(数据类接口)的端点描述符	197
6.6.7	修改好描述符后的测试	198
6.7	类请求的实现	200
6.7.1	GET_LINE_CODING 请求	200
6.7.2	SERIAL_STATE 通知	201
6.7.3	SET_CONTROL_LINE_STATE 请求	202
6.7.4	SET_LINE_CODING 请求	202
6.7.5	实现类请求后的测试	204
6.8	对串口数据的处理	205
6.9	安装驱动用的 inf 文件	211
6.10	本章小结	214
<b>第 7 章</b>	<b>USB MIDI 键盘</b>	
7.1	MIDI 简介	216
7.2	MIDI 的工作原理	217
7.3	USB MIDI 设备的数据流模型	217
7.4	设备描述符	218
7.5	配置描述符集合	218
7.5.1	配置描述符	218
7.5.2	音频控制接口描述符	218
7.5.3	类特殊音频控制接口描述符	219
7.5.4	MIDI 流接口描述符	220
7.5.5	类特殊 MIDI 流接口描述符	220
7.5.6	端点描述符和类特殊端点描述符	225
7.5.7	字符串描述符	227
7.6	修改好描述符后的测试	227
7.7	USB MIDI 键盘的数据返回	228
7.8	USB MIDI 键盘的使用	230
7.9	单片机自动弹奏的实现	232
7.10	本章小结	233

## 第 8 章 U 盘

8.1	USB 大容量存储设备	234
8.2	设备描述符	234
8.3	字符串描述符	235
8.4	配置描述符集合	235
8.4.1	配置描述符	235
8.4.2	接口描述符	235
8.4.3	端点描述符	236
8.5	测试	236
8.6	类特殊请求	237
8.6.1	Get Max LUN 请求	237
8.6.2	Bulk-Only Mass Storage Reset 请求	238
8.7	仅批量传输协议的数据流模型	239
8.7.1	命令块封包 CBW 的结构	239
8.7.2	命令状态封包 CSW 的结构	240
8.7.3	对批量数据的处理	240
8.8	SCSI 命令集和 UFI 命令集	241
8.8.1	查询命令 INQUIRY	241
8.8.2	读格式化容量命令 READ FORMAT CAPACITIES	243
8.8.3	读容量命令 READ CAPACITY	244
8.8.4	READ(10)命令	245
8.8.5	WRITE(10)命令	246
8.8.6	REQUEST SENSE 命令	247
8.8.7	TEST UNIT READY 命令	248
8.9	FAT 文件系统	248
8.9.1	关于 DBR	249
8.9.2	关于 FAT 表	251
8.9.3	关于目录项	252
8.10	模拟一个 FAT16 文件系统	253
8.11	实验结果	254
8.12	IDE 转 USB 的实现	256
8.13	本章小结	257

## 第 9 章 自定义 USB 设备及驱动开发

9.1	用户自定义 USB 设备	258
-----	--------------	-----

9.1.1	设备描述符 .....	258
9.1.2	配置描述符集合 .....	258
9.1.3	字符串描述符 .....	259
9.1.4	数据的处理 .....	259
9.2	驱动程序开发简介 .....	259
9.3	WDM 驱动开发编程环境的建立 .....	259
9.4	创建一个 USB WDM 驱动程序 .....	262
9.5	对工程的编译 .....	270
9.6	关于 inf 文件 .....	272
9.7	驱动程序的修改 .....	272
9.7.1	Read(KIrp I)函数 .....	273
9.7.2	Write(KIrp I)函数 .....	276
9.7.3	EP1_READ_Handler(KIrp I)函数 .....	277
9.7.4	EP1_WRITE_Handler(KIrp I)函数 .....	279
9.7.5	EP2_READ_Handler(KIrp I)函数 .....	280
9.7.6	EP2_WRITE_Handler(KIrp I)函数 .....	280
9.8	驱动的安装及安装后的信息 .....	280
9.9	应用程序对驱动器的访问 .....	284
9.10	测试软件的使用 .....	286
9.11	本章小结 .....	287
<b>第 10 章 USB 过滤驱动开发</b>		
10.1	过滤驱动简介 .....	289
10.2	使用 DS 创建一个下层过滤驱动 .....	290
10.3	过滤驱动代码的修改 .....	294
10.4	过滤驱动的安装 .....	300
10.5	过滤驱动的卸载 .....	302
10.6	驱动程序测试 .....	303
10.7	本章小结 .....	306
附 录 第 3 章实例的完整调试信息 .....		307
参考文献 .....		314
后 记 .....		315



# 第 1 章

## USB 概述及协议基础

本章首先概要地介绍一下 USB 系统,包括 USB 的出现、特点、结构及一些基本概念等,让 USB 设计者对 USB 系统有一个总体上的认识。

### 1.1 USB 是什么

USB 是什么呢?一种说法 USB 是 You SB 的意思。另一种说法是 USB 其实是美国的弟弟,因为美国叫 USA,USB 当然是他的弟弟了(由此看出,美国叫 USA 也不是随便叫的,而是经过一番深思熟虑的,不管是 USB、USC 还是 US×,都得做他的小弟了)。

那么 USB 到底是什么呢?其实 USB 是通用串行总线(Universal Serial Bus)的缩写,它已经有 10 多年的历史了。它的出现主要是为了简化个人计算机与外围设备的连接,增加易用性。在比较老的个人计算机中,基本上是要关掉计算机(往往还要打开机箱)之后才能连接或者更换设备,连接好之后还要在硬件上分配资源,然后重新启动计算机设备才能正常工作。这个过程是非常不方便的,如果设备能够在计算机运行过程中随意地接入,并且立刻就能正常工作,那么这样的特性叫做即插即用 PnP(Plug and Play)。这一特性将使计算机变得更易用、更大众化。USB 正是为解决这个问题而诞生的,它支持热插拔,并且是即插即用的;另外,它还具有很强的可扩展性,速度也很快。现在,USB 设备已经十分普及,只要跟计算机打过交道的人,几乎都知道 USB。

USB 协议出现过的版本有 USB1.0、USB1.1、USB2.0 等。由于 USB 是主从模式的结构,设备与设备之间、主机与主机之间不能互连。为了解决这个问题,扩大 USB 的应用范围,又出现了 USB OTG(On The Go)。USB OTG 的做法是:同一个设备,在不同的场合下可以在主机与从机之间切换。

### 1.2 USB 的特点

在 USB1.0 和 USB1.1 版本中,只支持 1.5 Mb/s 的低速(low-speed)模式和 12 Mb/s 的全速(full-speed)模式。在 USB2.0 中,又加入了速度更快(480 Mb/s)的高速(high-speed)模

## 第1章 USB 概述及协议基础

式。目前,USB3.0 协议正在制订当中,据说速度会比 USB2.0 的高速模式还要快 10 倍,即达到 5 Gb/s 左右。

值得注意的是,USB2.0 并不是高速设备的代名词,因为 USB2.0 协议对设备的高速模式并不是强制的,而是可选的。例如 PDIUSB12,它是符合 USB2.0 协议的,但是不支持高速模式,只支持 12 Mb/s 的全速模式。因此,在选择 USB 芯片时要注意,不要光看到支持 USB2.0 协议就认为一定具有高速模式。作为一个专业的 USB 设计人员,一定要搞清楚这个关系。

USB 具有很多优点,例如即插即用 PnP,容易使用,方便携带,传输速度快,可扩展性强,标准统一,价格便宜等。目前流行的 USB 设备主要有移动硬盘、数码相机、MP3、U 盘、USB 鼠标、普通键盘、游戏杆、USB MIDI 键盘、USB 摄像头、USB 打印机、USB 扫描仪、USB 声卡、USB 话筒、USB 网卡、USB 显示器、USB 电话,以及具有 USB 口的各种仪器仪表等。只要是连到计算机上的外设,就基本上可以通过 USB 来实现,足见 USB 系统的强大。

然而 USB 也有一些缺点,例如传输距离短,开发、调试难度大等。当然,它还有一个更大的缺点,那就是找出它的缺点是件非常令人头疼的事情。圈圈为了列出它的几个缺点而头疼了数月之久……

要开发 USB,一个网站是开发者必须要知道的,那就是 USB 开发者论坛,网址是 [Http://www.usb.org/](http://www.usb.org/)。该网站是公布 USB 的相关协议和标准的官方网站,大家需要相关协议和文档时,可以去该网站下载。另外,还有一个网站 <USB 专区> 小组 <http://group.ednchina.com/93/>,大家有什么问题也可以去那里交流。那里有很多搞 USB 的朋友,也有很多资源和代码下载。

现在 USB 技术已经很流行了,就像以前的串口一样。以前的电子工程师不会搞串口通信就有点落伍了,而现在的电子工程师如果不会搞 USB 通信就有点落伍了。



图 1.2.1 USB 的 Logo

图 1.2.1 所示是 USB 的 Logo(标志),看到它你就应该想起 USB。

### 1.3 USB 的拓扑结构

USB 是一种主从结构的系统。主机叫做 Host,从机叫做 Device(也叫做设备)。

通常所说的主机具有一个或者多个 USB 主控制器(host controller)和根集线器(root hub)。主控制器主要负责数据处理,而根集线器则提供一个连接主控制器与设备之间的接口和通路。另外,还有一类特殊的 USB 设备——USB 集线器(USB hub),它可以对原有的 USB 口在数量上进行扩展,就可以获得更多的 USB 口。

**注意:**集线器只能扩展出更多接口的 USB 口,而不能扩展出更多的带宽。带宽是共享一个 USB 主控制器的。

通常,PC上有多个USB控制器和多个USB口。每个主控制器下有一个根集线器,根集线器通常具有一个或者几个USB口。当你有多个不同的USB设备都需要较大的数据带宽时,可以考虑将它们分别接到不同的主控制器的根集线器上,以避免带宽不足。

在Windows的设备管理器里,可以看到USB控制器和USB根集线器。右击“我的电脑”,在弹出的菜单中选择“属性”,在“属性”对话框中选择“硬件”选项卡,再单击“设备管理器”按钮,打开“设备管理器”对话框(以后经常要到设备管理器中查看一个设备的属性,例如驱动程序是否安装好,查看驱动程序信息、设备的PID和VID等,请记住查看设备管理器的方法)。在“设备管理器”对话框中找到“通用串行总线控制器”并展开之,即可看到USB控制器和根集线器。随便选择一个根集线器,双击之,在弹出的窗口中选择“电源”标签页,就可以看到该根集线器下的端口信息了。

图1.3.1和图1.3.2是圈圈的计算机中设备管理器以及根集线器的情况。从图中可以看到,总共有7个USB控制器和7个USB根集线器。所选择查看的那个根集线器具有两个USB口,一个正在使用中,需要电流为100mA(它就是圈圈正在使用的鼠标),另一个是空闲的。

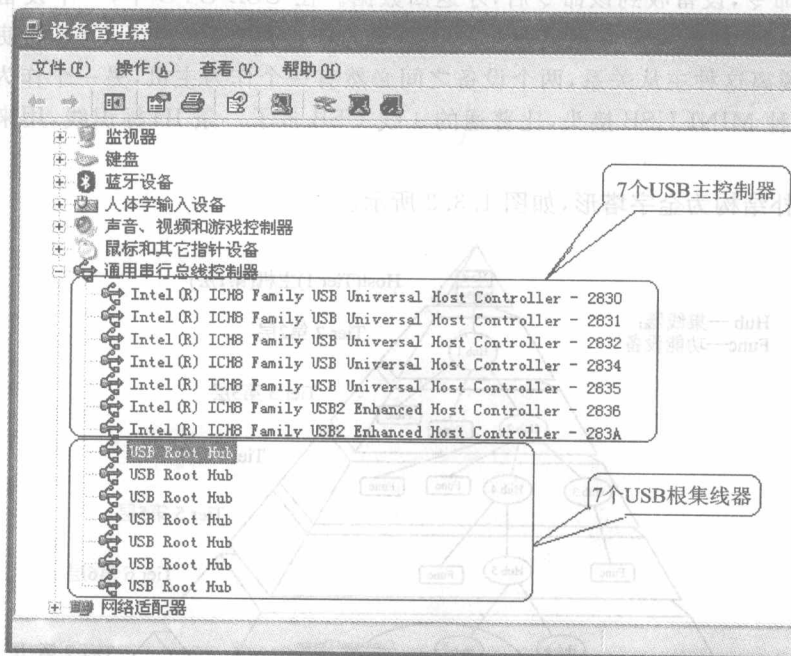


图 1.3.1 设备管理器中的控制器和根集线器

USB的数据交换只能发生在主机与设备之间,主机与主机、设备与设备之间不能直接互连和交换数据。为了在物理上区分主机和设备,使用了不同的插头和插座,具体在1.5节中会讲到。所有的数据传输都由主机主动发起,而设备只是被动地负责应答。例如,在读数据时,

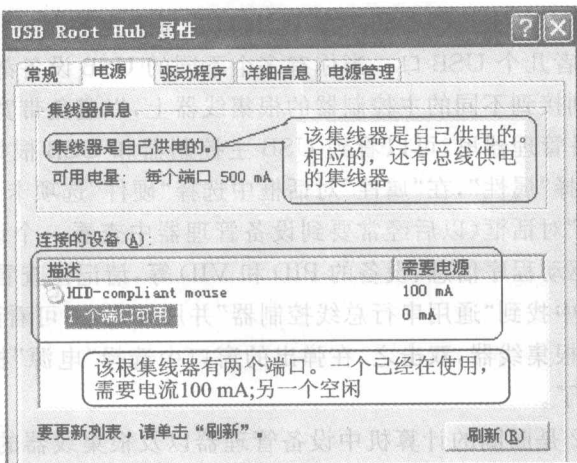


图 1.3.2 一个根集线器的属性

USB 先发出读命令,设备收到该命令后,才返回数据。在 USB OTG 中,一个设备可以在从机与主机之间切换,这样就可以实现设备与设备之间的连接,大大增加了 USB 的使用范围。但这时依然没有脱离这种主从关系,两个设备之间必然有一个作为主机,另一个作为从机。USB OTG 增加了一种 MINI USB 接头,比普通的 4 线 USB 多了一条 ID 标识线,用来表明它是主机还是设备。

USB 的拓扑结构为金字塔形,如图 1.3.3 所示。

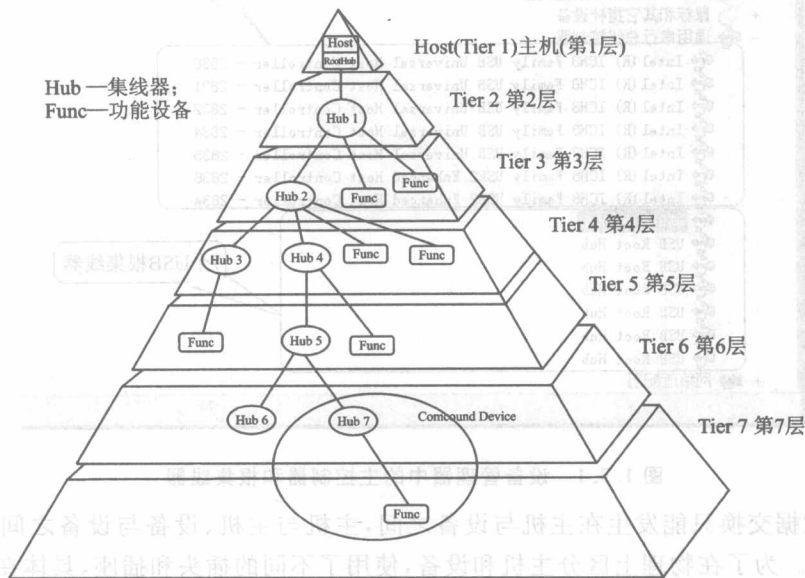


图 1.3.3 USB 的金字塔形拓扑结构



塔顶为 USB 控制器和根集线器,下面接 USB 集线器,USB 集线器将一个 USB 口扩展为多个 USB 口,多个 USB 口又可以通过集线器扩展出更多的接口。但 USB 协议中对集线器的层数是有限制的,USB1.1 规定最多为 4 层,USB2.0 规定最多为 6 层。理论上,一个 USB 主控制器最多可接 127 个设备,这是因为协议规定每个 USB 设备具有一个 7 bit 的地址(取值范围为 0~127,而地址 0 是保留给未初始化的设备使用的)。实际上,通常是不会连接 127 个设备的。所说的一个 USB 主控制器可以连接多个 USB 设备,并不是直接简单地将多个设备并联或者串联,而是要由集线器负责端口扩展,才能连接更多的设备。在 PC 上,也有一个或者多个(视主板上的 USB 主控制器的个数而定)集线器,它就是前面提到的根集线器,直接连在 USB 主控制器上。

一个完整的 USB 数据传输过程如下:首先由 USB 主控制器发出命令和数据,通过根集线器,再通过下面的集线器(如果有)发给 USB 设备;设备对接收到的数据进行处理后,返回一些信息或者数据,它首先到达其上一层的集线器,上层的集线器再交给更上层的集线器,一直到 USB 主控制器为止;最终,USB 主控制器将数据交给计算机的 CPU 处理。在标准的 PC 上,USB 主控制器是挂接在 PCI 总线上的。在 Windows 中,由各种 USB 功能驱动程序负责产生和管理 USB 功能设备(FDO)。这就是我们最终所看到的实际设备。我们的应用程序可以通过 Windows 提供的一些 API 函数来访问 USB 设备,例如 CreateFile(), ReadFile(), DeviceIOControl()等。

## 1.4 USB 的电气特性

标准的 USB 连接线使用 4 芯电缆:5 V 电源线( $V_{BUS}$ )、差分数据线负(D<sup>-</sup>)、差分数据线正(D<sup>+</sup>)及地(GND)。在 USB OTG 中,又增加了一种 MINI USB 接头,使用的是 5 条线,比标准的 USB 多了一条身份识别(ID)线。USB 使用的是差分传输模式,因而有 2 条数据线,分别是 D<sup>+</sup>和 D<sup>-</sup>。在 USB 的低速和全速模式中,采用的是电压传输模式;而在高速模式下,则是电流传输模式。关于具体的各种电气参数,请参看 USB 协议。

USB2.0 支持 3 种传输速度:低速模式(1.5 Mb/s)、全速模式(12 Mb/s)以及高速模式(480 Mb/s)。传输速度是指总线上每秒传输的位数,实际的数据速率要比这个速度低一些,因为有很多协议开销,例如同步、令牌、校验、位填充和包间隙等。

USB 使用的是 NRZI 编码方式:当数据为 0 时,电平翻转;数据为 1 时,电平不翻转,如图 1.4.1 所示。为了防止出现长时间电平不变化(这样不利于时钟信号的提取),在发送数据前要经过位填充(bit stuffing)处理。位填充处理的过程是这样的:当遇到连续 6 个数据 1 时,就强制插入一个数据 0。经过位填充后的数据,由串行接口引擎(SIE)将数据串行化和 NRZI