

PDIUSB12 的接口应用设计

USB 作为一种新型的接口技术,以其简单易用,速度快等特点而备受青睐.本文简单介绍 USB 接口的特点和 PHILIPS 公司的 USB 接口芯片 PDIUSB12,并详细说明 USB 软硬件开发过程中应注意的问题。

引言

USB 是近年来应用在 PC 领域的新型接口技术,是一些 PC 大厂商,如 Microsoft、Intel 等为了解决日益增加的 PC 外设与有限的主板插槽和端口之间的矛盾而制定的一种串行通信的标准,自 1995 年在 Comdex 上亮相以来,至今已广泛地为各 PC 厂家所支持现在生产的 PC 几乎都配备了 USB 接口。Microsoft 的 Windows98、NT 以及 MacOS、Linux、FreeBSD 等流行操作系统都增加了对 USB 的支持。

USB 的主要优点:

1. 使用方便。连接外设不必再打开机箱允许外设热插拔而不必关闭主机电源;
2. 速度快。USB 接口的最高传输率可达 12 Mb/s;
3. 提供低速方式,速率为 1.5 Mb/s 扣除用于总线状态控制和错误检测等数据传输,最大理论速度也能达到 1.2 Mb/s 和 9.6 Mb/s。
4. 连接灵活。一个 USB 口理论上可以连接 127 个 USB 设备,连接的方式也十分灵活,既可以使用串行连接,也可以使用集线器 Hub,把多个设备连接在一起,再同 PC 机的 USB 口相接。
5. 独立供电。USB 接口提供了内置电源。

现在的 USB 生产厂商很多,几乎所有的硬件厂商都有 USB 的产品。USB 控制器一般有两种类型:一种是 MCU 集成在芯片里面的如 Intel 的 8X930AX、CYPRESS 的 EZ-USB、SIEMENS 的 C541U 以及 MOTOLORA、National Semiconductors 等公司的产品。另一种就是纯粹的 USB 接口芯片仅处理 USB 通信如 PHILIPS 的 PDIUSB11 (I2C 接口)、PDIUSB11A、PDIUSB12 (并行接口)、National Semiconductor 的 USBN9602、USBN9603、USBN9604 等、前一种由于开发时需要单独的开发系统,因此开发成本较高;而后一种只是一个芯片与 MCU 接口实现 USB 通信功能,因此成本较低,而且可靠性高。本文主要介绍 PHILIPS 公司的 PDIUSB12 器件。

1 PDIUSB12 芯片特点和内部结构

PDIUSB12 是一个性能优化的 USB 器件。通常用于基于微控制器的系统并与微控制器通过高速通用并行接口进行通信,也支持本地 DMA。传输该器件采用模块化的方法实现一

个 USB 接口，允许在众多可用的微控制器中选择最合适的作为系统微控制器，允许使用现存的体系结构并使固件投资减到最小。这种灵活性减少了开发时间、风险和成本。是开发低成本且高效的 USB 外围设备解决方案的一种最快途径。PDIUSBD12 完全符合 USB1.1 规范，也能适应大多数设备类规范的设计如成像类、大容量存储类、通信类、打印类和人工输入设备等。因此，PDIUSBD12 非常适合做很多外围设备，如打印机、扫描仪、外部大容量存储器（Zip 驱动器）和数码相机等现在用 SCSI 实现的很多设备如果用 USB 来实现可以直接降低成本。

PDIUSBD12 挂起时的低功耗以及 LazyClock 输出符合 ACPI、OnNOW 和 USB 电源管理设备的要求。低功耗工作允许实现总线供电的外围设备。

PDIUSBD12 还集成了像 SoftConnect、GoodLink、可编程时钟输出、低频晶振和终端电阻等特性。所有这些特性都能在系统实现时节省成本，同时在外围设备上很容易实现更高级的 USB 功能。

1.1 主要特性

- 符合 USB 1.1 协议规范；
- 集成了 SIE FIFO 存储器收发器和电压调整器的高性能 USB 接口芯片；
- 适应大多数设备类规范的设计；
- 与任何微控制器/微处理器有高速 2MB/s 的并行接口；
- 全自动 DMA 操作；
- 集成了 320 B 的多配置 FIFO 存储器；
- 主端点有双缓存配置，增加吞吐量，容易实现实时数据传输；
- 在块传输模式下有 1MB/s 的数据传输率，在同步传输模式下有 1Mb/s 的数据传输率；
- 具有总线供电能力，有非常好的 EMI 性能；
- 在挂起时有可控制的 LazyClock 输出；
- 可通过软件控制 USB 总线连接 SoftConnect；
- 在 USB 传输时有闪亮的 USB 连接指示灯 GoodLink；
- 时钟频率输出可编程；
- 符合 ACPI、OnNOW 和 USB 电源管理要求；
- 具有内部上电复位和低电压复位电路；
- 有 SO18 和 TSSOP28 封装；
- 能在 -40~ +85℃工业级工作；
- 片内 8kV 静电保护；
- 双电压工作(3.3±0.3 V)或扩大的 5 V 电压范围（3.6~5.5 V）；
- 多中断模式，方便块传输和同步传输。

1.2 内部结构

PDIUSB12 的内部框图如图 1 所示。

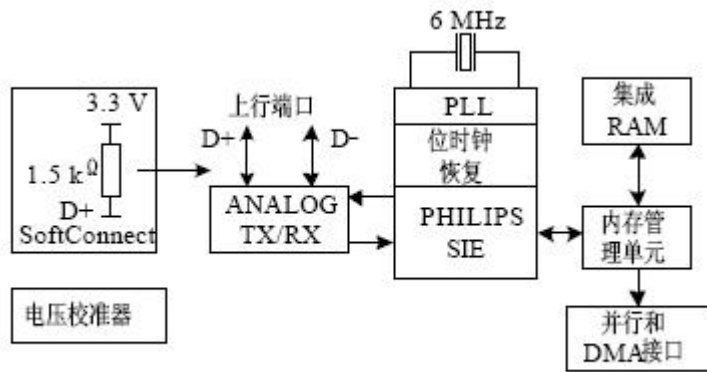


图 1 PDIUSB12 内部结构框图

1. 模拟收发器。集成的收发器直接通过终端电阻与 USB 电缆接口。
2. 电压调整器。片上集成的 1 个 3.3 V 电压调整器为模拟收发器供电，也提供连接到外部 1.5 k 上拉电阻的输出电压。作为选择，PDIUSB12 提供集成 1.5 k 上拉电阻的 SoftConnect 技术。
3. PLL。片上集成 1 个 6~48 MHz 的倍频 PLL（锁相环）允许使用 6 MHz 的晶振，EMI 也由于使用低频晶振而减小。PLL 的工作不需要外部器件。
4. 位时钟恢复。位时钟恢复电路用 4 倍过采样原理从输入的 USB 数据流中恢复时钟，能跟踪 USB 规范中指出的信号抖动和频率漂移。
5. PHILIPS 串行接口引擎 PSIE、PHILIPS 的 SIE 完全实现 USB 协议层。考虑到速度，它是全硬件的，不需要固件微程序介入。这个模块的功能包括：同步模式识别、并/串转换、位填充/不填充、CRC 校验、PID 确认、地址识别以及握手鉴定。
6. SoftConnect 高速设备与 USB 的连接是靠把 D+ 通过 1 个 1.5 k 的上拉电阻接到高电平来建立的。在 PDIUSB12 中这个上拉电阻是集成在芯片内的，缺省是没有连接到 VDD，这个连接是靠外部 MCU 发一个命令来建立的。这使得系统微处理器可以在决定建立 USB 连接之前完成初始化。重新初始化 USB 总线连接也可以不用拔掉电缆来完成。
7. GoodLink。GoodLink 是靠一个引脚接发光二极管实现的。在 USB 设备枚举时 LED 指示灯将立即闪亮；当 PDIUSB12 被成功枚举并配置时，LED 指示灯将会始终亮；经过 PDIUSB12 的 USB 数据传输过程中，LED 将一闪一闪；传输成功后，LED 熄灭；在挂起期间 LED 熄灭；这种特性可以使我们知道 PDIUSB12 的状态方便电路调试。
8. 存储器管理单元 MMU 和集成 RAM。MMU 和集成 RAM 能缓冲 USB（工作在 12Mb/s）数据传输和微控制器之间并行接口之间的速度差异，这允许微控制器以自己的速度读写 USB 包。

9. 并行和 DMA 接口。并行接口容易使用，速度快并且能直接与主微控制器接口。对于微控制器，PDIUSB12 可以看成是一个有 8 位数据总线和 1 位地址线的存储设备，PDIUSB12 支持多路复用和非多路复用的地址和数据总线。在主端点（端点 2）和局部共享存储器之间也可使用 DMA（直接存储器存取）传输。它支持单周期模式和块传送模式两种 DMA 传输。

2 PDIUSB12 的引脚说明及典型连接

2.1 PDIUSB12 引脚说明

PDIUSB12 引脚如图 2 所示，引脚如表 1 所列。

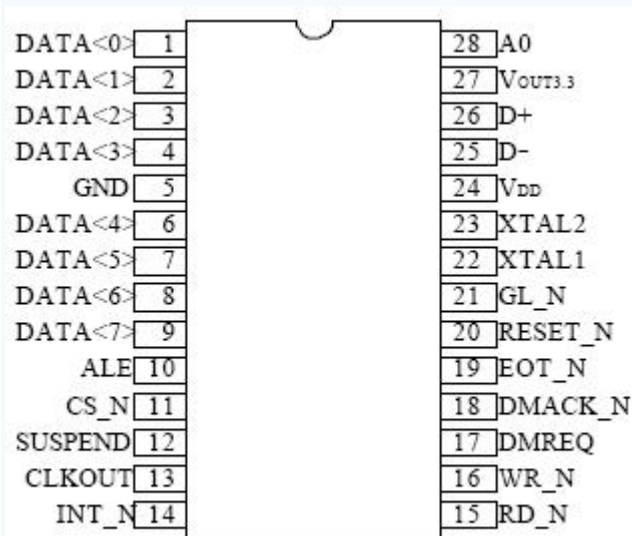


图 2 PDIUSB12 的引脚图

表 1 PDIUSB12 引脚说明

引脚号	符号	说 明
1~4、 6~9	DATA<0>~ DATA<7>	8 位双向数据
5	GND	地
10	ALE	地址锁存允许。在多路复用地址/数据总线时，ALE 下降沿用于锁存地址信息；独立地址/数据总线时将 ALE 永久接地
11	CS_N	片选（低电平有效）
12	SUSPEND	芯片进入挂起状态
13	CLKOUT	可编程时钟输出
14	INT_N	中断输出（低电平有效）
15	RD_N	读选通（低电平有效）
16	WR_N	写选通（低电平有效）
17	DMREQ	DMA 请求
18	DMACK_N	DMA 响应（低电平有效）
19	EOT_N	DMA 传输结束（低电平有效）。另一个功能是 VBUS 感知器
20	RESET_N	复位（低电平有效、异步）。有片内上电复位电路，该引脚可以接高
21	GL_N	GoodLink 发光二极管指示器（低电平有效）
22	XTAL1	晶振连接 1（6 MHz）
23	XTAL2	晶振连接 2（6 MHz）
24	V _{DD}	正电源（4.0~5.5 V）。让芯片工作在 3.3 V，将 3.3 V 电压加到 V _{DD} 和 V _{OUT3.3} 两个引脚上
25	D ⁻	USB D ⁻ 数据线
26	D ⁺	USB D ⁺ 数据线
27	V _{OUT3.3}	3.3 V 输出
28	A0	地址位。A0=1 选择命令，A0=0 选择数据。在多路复用地址和数据总线配置时，这一位将不考虑，应接高电平

2.2 PDIUSB12 的典型连接

PDIUSB1 与 80C51 的连接电路如图 3 所示。在这个例子中，ALE 始终接低电平，说明采用单独地址和数据总线配置。A0 脚接 80C51 的任何 I/O 引脚，控制是命令还是数据输入到 PDIUSB12。80C51 的 P0 口直接与 PDIUSB12 的数据总线相连接。CLKOUT 时钟输

出为 80C51 提供时钟输入。

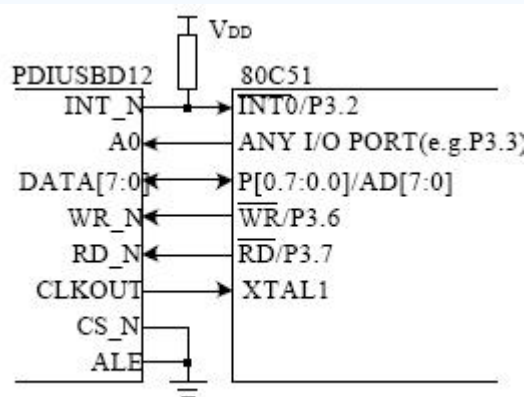


图 3 PDIUSBD12 与 80C51 的连接电路图

3 软件设计

3.1 单片机方面软件设计

对于单片机控制程序，目前没有任何厂商提供自动生成固件 **firmware** 的工具,因此所有程序都要由自己手工编制。USB 单片机控制程序通常由三部分组成：第一、初始化单片机和所有的外围电路（包括 PDIUSBD12）；第二、主循环部分。其任务是可以中断的；第三、中断服务程序，其任务是对时间敏感的，必须马上执行。根据 USB 协议，任何传输都是由主机 **host** 开始的，这样，单片机作它的前台工作，等待中断，主机首先要发令牌包给 USB 设备（这里是 PDIUSBD12），PDIUSBD12 接收到令牌包后就给单片机发中断，单片机进入中断服务程序，首先读 PDIUSBD12 的中断寄存器，判断 USB 令牌包的类型，然后执行相应的操作。因此，USB 单片机程序主要就是中断服务程序的编写，在 USB 单片机程序中要完成对各种令牌包的响应，其中比较难处理的是 **SETUP** 包，主要是端口 0 的编程。

单片机与 PDIUSBD12 的通信主要是靠单片机给 PDIUSBD12 发命令和数据来实现的。PDIUSBD12 的命令字分为三种：初始化命令字、数据流命令字和通用命令字。PDIUSBD12 给出了各种命令的代码和地址。单片机先给 PDIUSBD12 的命令地址发命令，根据不同命令的要求再发送或读出不同的数据。因此可以将每种命令做成函数，用函数实现各个命令，以后直接调用函数即可。

在编写 USB 的单片机程序时，需要注意：

单片机的中断应设置为电平触发：中断后一定要读上次传输状态寄存器（命令 40-45H），以清除中断寄存器中的中断标志，这样，PDIUSBD12 的中断输出才能变回高电平，这一点非常重要。

1. 在接收到 **Setup** 包后，一定要调用 **ACK setup** 命令重新使能端口 0。
2. 在向 **IN** 端点写完数据后，一定调用 **Validate Buffer**（命令 **FAH**）指明缓冲区中的数据有效，可以发送到主机。
3. 当读完数据后，一定调用 **Clear Buffer**（命令 **F2H**）以保证可以接收新的包。

4. 可以通过调用 Read Chip ID（命令 FDH）检查 PDIUSB12 是否工作。该命令要读两个字节数据

USB 初始化过程为

- Set Address Enable;
- Set Endpoint Enable（此时 LED 亮）;
- Disconnect;
- delay（1~2 s）;
- Connect（即用 43h 参数调用 Set Mode，此时 LED 灭）;
- Read Interrupt Register。

完成初始化工作后，就可作其它的前台工作了，并在前台判断是否有 Setup 包（通过一个变量，当中断服务程序检测到有 Setup 包时，设置该变量）。然后执行响应的控制传输。在调试 USB 单片机程序时，还要特别注意 Windows 对 USB 设备的枚举顺序。

1. GetDeviceDescriptor。主机主要对 Length 域感兴趣，发送内容一定要正确，特别是第 2 字节 type 一定为 0x01，即 Device；否则，主机将不响应，或者再重复 2 次后放弃，可检查一下对 EP0 的 RX、TX 的设置次序。
2. SetAddress。一般为 02 或 03；
3. 连续 3 次 GetDeviceDescriptor。读取全部设备描述符，一般为 18 B，分为多次传输，如果不正确主机将不响应或重复 2 次后放弃。
4. GetConfigDescriptor。注意第 2 字节一定为 0x02，即 config。
5. GetStringDescriptor（可能没有）。根据在设备描述符中是否有 String 索引而定，一般先读取 LanguageID，再读取 product string。
6. 读取全部 ConfigDescriptor。次数根据描述符的大小决定（端点个数不同，描述符大小不同），如果不正确，主机将不响应或再重复 2 次后放弃。

如果以上步骤都正确，主机将找到新设备，提示安装驱动程序；否则找到未知设备，不可用。安装驱动程序后，以后的每次 PlugIn，枚举次序与以上步骤略有不同，之后会有 SetConfiguration、GetConfiguration 和 GetInterface 等调用。

3.2 主机方面软件设计

Windows98 提供了多种 USB 设备的驱动程序，但好像还没有一种是专门针对数据采集系统的，所以必须针对特定的设备来编制驱动程序。尽管系统已经提供了很多标准接口函数，但编制驱动程序仍然是 USB 开发中最困难的一件事情，通常采用 Windows DDK 来实现。目前有许多第三方软件厂商提供了各种各样的生成工具，像 Compuware 的 driverworks，Blue Waters 的 Driver Wizard 等，它们能够很容易地在几分钟之内生成高质量的 USB 的驱

动程序。作为 WIN98 和 WIN2K 推荐的一项新技术来说, USB 的驱动程序和以往的直接跟硬件打交道的 WIN95 的 VXD 方式的驱动程序不同, 它是 WDM 类型的。

在调试 USB 设备时, 可使用 UsbView 程序检测设备是否能被 Windows 枚举并配置, 如果成功, 还可在该程序中查看设备描述符、配置描述符和端点描述符是否正确。之后可以使用 Driver Wizard 生成一个通用驱动程序, 在 Windows 提示安装驱动程序时, 选择 Driver Wizard 生成的驱动程序。其实 Driver Wizard 生成的仅是一个 Windows 控制台的应用程序, 它会调用安装 Driver Wizard 时安装在系统中的通用 USB 驱动程序。使用该程序就可测试设备是否能够正确传输数据以及传输速度, 该程序也可作为最终产品 USB 传输部分的框架; 如果不能满足要求, 也可用 WDM 重新编制驱动程序, 用调试好的 USB 设备来开发、调试主机软件。

4 应用实例

本文介绍一个高速数据采集系统, 以AD公司的ADuC812 为系统控制器。该单片机本身就是高度集成的高精度 12 位数据采集系统, 在其片内不仅组合了可重新编程非易失性闪速/电擦除程序存储器的高性能 8 位(与 8051 兼容MCU)还包含了高性能的自校准多通道(8 个输入通道) 12 位ADC 和两个 12 位DAC, 且内核与 8051 指令集兼容。PDIUSB12 作为 ADuC812 的存储器外设, 接口比较简单。需要注意的地方是DMACK引脚必须接高电平, 否则将不能接收任何命令和数据; EOT_N必须通过电阻接到USB的+5 V, 以正确检测到USB 连接; INT_N引脚加 1 个上拉电阻, +5 V 接到VDD 引脚; 在VOUT3.3 引脚加 1uF 电解电容和 0.1uF 两个退耦电容。

结束语

PDIUSB12 是一个性能优化的 USB 器件, 它的 SoftConnect 和 GoodLink 技术使开发和调试 USB 设备时非常方便, 在性能、速度、方便性以及成本上都具有很大的优势。因此使用 PHILIPS 公司的 PDIUSB12 可以快速开发出高性能的 USB 设备。