

USB(Universal Serial Bus)

USB(通用串行总线)是用于将适用 USB 的外围设备连接到主机的外部总线结构,其主要是用在中速和低速的外设。USB 是通过 PCI 总线和 PC 的内部系统数据线连接,实现数据的传输。USB 同时又是一种通信协议,他支持主系统(host)和 USB 的外围设备(device)之间的数据传输。

USB 的动机

1. 连接 PC 与电话。PC 具有很强的运算能力,而电话提供最为广泛的通信互连。运算与通信成为计算机应用的基础,而计算机与通信是两个相对独立发展的产业,USB 旨在提供可以广泛应用于 PC 到电话的互连的普遍性的连接。
2. **Plug-and-Play**。从用户端来看,PC 的串行口、并行口和键盘鼠标端口都不能 Plug-and-Play。USB 则提供真正的 Plug-and-Play。
3. 端口扩展。PC 的已有的串行/并行口等端口只适用于一两种的外设,并且不易扩展。USB 提供双向、低成本、低速到中速(USB 2.0 可达 480Mb/s)的通用外设总线,适用于连接各种各样的外设,并且易于扩展。

USB 的拓扑结构

USB的拓扑结构如图 1:

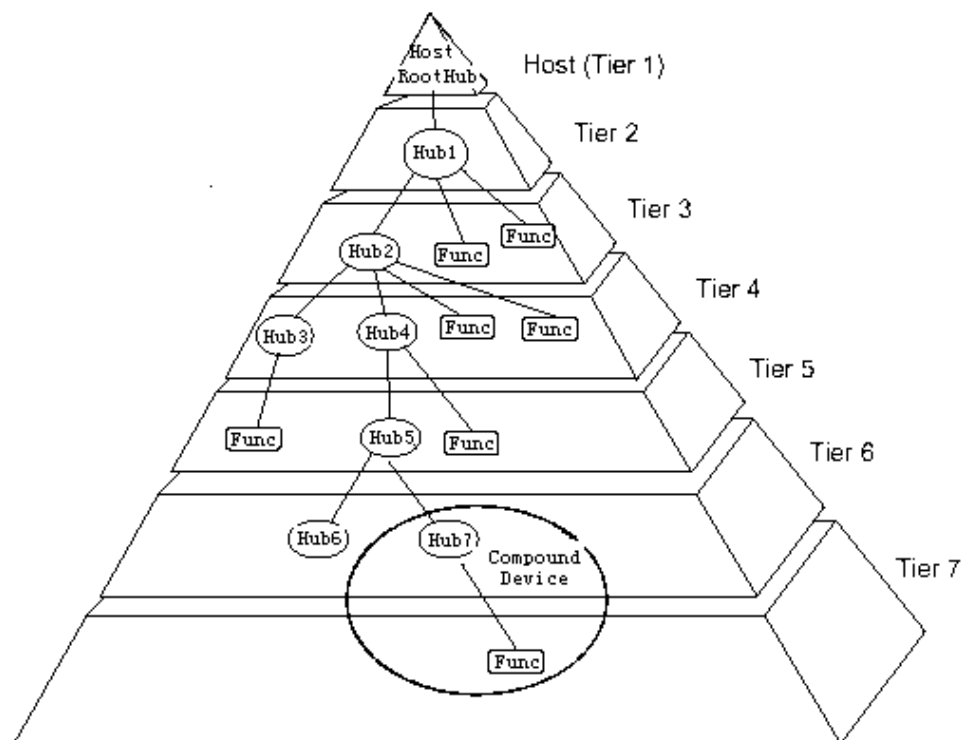


图 1: USB Toplogy

在 USB 的网络协议中，每个 USB 的系统有且只有一个 host，它负责管理整个 USB 系统，包括 USB Device 的连接与删除、Host 与 USB Device 的通信、总路线的控制等等。Host 端有一个 Root Hub，可提供一个或多个 USB 下行端口。每个端口可以连接一个 USB Hub 或一个 USB Device。USB Hub 是用于 USB 端口扩展的，即 USB Hub 可以将一个 USB 端口扩展为多个端口。图 1 中的每个 Func(Function)就是一个 USB Device，如 USB 键盘、USB 鼠标、USB MODEM、USB 硬盘等等。Compound Device 是指带一个 Hub 和一个或多个不可删除的 USB Device 的复合设备。一个 USB 系统可连接多达 127 个 Function。

USB 的系统有且只有一个 host，而 PC 端的 USB 都是 Host，所以将两台 PC 的 USB 口通过 A-A USB 电缆连接起来，是不能实现通信。如果将两个 host 连起来通信，这样一来的一个 USB 的系统有了两个的 host，与它的网络协议冲突。Anchorchip、Prolific 等厂家有 USB 到 USB 的设备控制器，可以实现两个 USB Host 的相连。

USB 设备

USB 的设备可以接在 PC 上的任意的 USB 接口上。而使用 Hub 还可以扩展使更多的 USB 设备连接到系统中，USB 的 Hub 有一个上行的端口(到 host)，有多个的下行端口(连接其它的设备)，从而可以使整个的系统可以扩展的连接 127 个外设，其中 Hub 也算外设。对于 USB 系统来说，USB 的 host 永远在 PC 边，所有的其他连接到 host 都称为设备，在设备与设备之间是无法实现直线通信的，只有通过 host 的管理与调节才能够实现数据的互相传送。在系统中，通常会有一个根 Hub，这个 Hub 一般有两个下行的端口。

USB 的设备类型(device class)

虽然 USB 设备都会表现 USB 的一些基本的特征。但是 USB 的设备还是可以分成多个不同类型，同类型的设备可以拥有一些共同的行为特征和工作协议，从而使设备的驱动程序的书写变得简单一些。下表中就给出一些基本的 USB 的设备类型分类。

设备类型(device class)	设备举例	类型常量(Class constant)
音频(audio)	扬声器	USB_DEVICE_CLASS_AUDIO
通信	MODEM	USB_DEVICE_CLASS_COMMUNICATIONS
HID	键盘、鼠标	USB_DEVICE_CLASS_HUMAN_INTERFACE
图像	摄相机、扫描仪	USB_DEVICE_CLASS_IMAGE
显示	监视器	USB_DEVICE_CLASS_MONITOR
物理回应设备	动力回馈式游戏操纵杆	USB_DEVICE_CLASS_PHYSICAL_INTERFACE
电源	不间断电源供应	USB_DEVICE_CLASS_POWER
打印机		USB_DEVICE_CLASS_PRINTER
Bulk 存储器	硬盘	USB_DEVICE_CLASS_STORAGE
HUB		USB_DEVICE_CLASS_HUB

USB 的优点有以下几条：

1. USB 为所有的 USB 外设提供了单一的、易于使用的标准的连接类型。这样一来就简化了 USB 外设的设计，同时也简化了用户在判断哪个插头对应哪个插槽时的任务,实现了单一的数据通用接口。
2. 整个的 USB 的系统只有一个端口和一个中断，节省了系统资源。
3. USB 支持热插拔(hot plug)和 PNP(Plug-and-Play)，也就是说在不关闭 PC 的情况下可以安全的插上和断开 USB 设备，计算机系统动态地检测外设的插拔，并且动态地加载驱动程序。其他普通的外围连接标准，如 SCSI 设备等必须在关掉主机的情况下才能插拔外围设备。
4. USB在设备供电方面提供了灵活性。USB直接连接到 Hub 或者是连接到 Host 的设备可以通过 USB 电缆供电，也可以通过电池或者其它的电力设备来供电，或使用两种供电方式的组合.并且支持节约能源的挂机和唤醒模式。
5. USB 提供全速 12Mbps 的速率和低速 1.5Mbps 的速率来适应各种不同类型的外设。USB2.0 还支持 480Mbps 的高速传输速率。
6. 为了适应各种不同类型外围设备的要求，USB 提供了四种不同的数据传输类型：控制传输、Bulk 数据传输、中断数据传输和同步数据传输。同步数据传输可为音频和视频等实时设备的实时数据传输提供固定带宽。
7. USB 的端口具有很灵活的扩展性，一个 USB 端口串接上一个 USB Hub 就可以扩展为多个 USB 端口。

传输方式

USB 有四种的传输方式：控制(Control)、同步(isochronous)、中断(interrupt)、大量(bulk)。如果你是从硬件开始来设计整个的系统，你还要正确选择传输的方式，而作为一个驱动程序的书写者，就只需要弄清楚他是采用的什么工作方式就行了。通常所有的传输方式下的主动权都在 PC 边,也就是 host 边。

1. 控制(Control)方式传输：控制传输是双向传输，数据量通常较小。USB 系统软件用来主要进行查询、配置和给 USB 设备发送通用的命令。控制传输方式可以包括 8、16、32 和 64 字节的数据，这依赖于设备和传输速度。控制传输典型地用在主计算机和 USB 外设之间的端点(Endpoint)0 之间的传输，但是指定供应商的控制传输可能用到其它的端点。
2. 同步(isochronous)方式传输：同步传输提供了确定的带宽和间隔时间 (latency)。它被用于时间严格并具有较强容错性的流数据传输，或者用于要求恒定的数据传输率的即时应用中。例如执行即时通话的网络电话应用时，使用同步传输模式是很好的选择。同步数据要求确定的带宽值和确定的最大传输次数。对于同步传输来说，即时的数据传递比完美的精度和数据的完整性更重要一些。
3. 中断(interrupt)方式传输：中断方式传输主要用于定时查询设备是否有中断数据要传输。设备的端点模式器的结构决定了它的查询频率，从 1 到 255ms 之间。这种传输方式典型的应用在少量的分散的、不可预测数据的传输。键盘、操纵杆和鼠标就属于这一类型。中断方式传输是单向的并且对于 host 来说只有输入的方式。
4. 大量(bulk)传输：主要应用在数据大量传输传输和接受数据上，同时又没有带宽和间隔

时间要求的情况下，要求保证传输。打印机和扫描仪属于这种类型。这种类型的设备适合于传输非常慢和大量被延迟的传输，可以等到所有其它类型的数据的传输完成之后再传输和接收数据。

USB 将其有效的带宽分成各个不同的帧(frame)，每帧通常是 1ms 时间长。每个设备每帧只能传输一个同步的传输包。在完成了系统的配置信息和连接之后，USB 的 host 就会对不同的传输点和传输方式做一个统筹安排，用来适应整个的 USB 的带宽。通常情况下，同步方式和中断方式的传输会占据整个带宽的 90%，剩下的就安排给控制方式传输数据。

数据交换 (Transactions)

一个 transaction 是在 host 和设备 (device)之间的不连续相互数据交换，通常由 host 开始交换，交换的开始是由 Token 的包开始的，接下来是双方向上的数据包，在数据包传送完之后，就会由设备 (device)返回一个握手 (handshake)包。USB 系统通过 IN,OUT,和 SETUP 的包来指定 USB 地址和 endpoint (最多是 128 个，0 通常被用来用做缺省的传送配置信息的)，并且这些被指定的设备必须通过上面形式的包来回应这种形式的指定。每个 SETUP 的包包含 8 个 byte 的数据，数据用来指示传送的数据类型。对于 DATA 数据包来说，设置两种类型的数据包是为了能够在传送数据的时候做到更加的精确。ACK handshake 的包用来指示数据传送的正确性，而 STALL handshake 则表示数据包在传送的过程中出了故障，并且请示 host 重新发数据或者清除这次传送。PRE 格式的包主要是用在在一个 USB 的系统中如果存在不同速率的设备的时候，将不同于总线速度的设备中就会回应一个 PRE 的包从而会忽略该设备。各种不同类型的包的大小是不同的，DATA 的数据包最大是 1023bytes。

USB 设备的开发

1. 从<http://www.usb.org/developer>处得到 USB 的 Specification。阅读相关内容，同时可参考 *The USB Handbook* by Jaff Kosar、*Developing USB Peripherals* by Wooi Ming Tan 等参考书。
2. 需要一台支持 USB 的 PC。如果要开发 WDM 驱动程序则需要一台有以下配置的 PC：

Windows NT 4.0

Windows NT DDK

USB DDK

Win 32 DDK

Visual C++

先装 VC，然后 DDK，然后 Nemega 公司的 DriverStudio 或是 WinDriver(开发工具)。

3. 编写 USB Device 的规格书。
4. 无论如何，不要做无谓的重复，可以购买相应的评估板和评估软件，并找到尽可能多的实例源程序代码。然后需要有你所选的 MCU 的程序编译器。
5. 申请成为 USB 论坛(USB forum)的成员(每年大约需要支付\$2500)，同时就可以获得一个 Vendor ID。也可以向他们购买 Vendor ID，每个 Vendor ID 的零售价格是\$200。不过 USB 论坛的成员可以在 USB 相关的支持方面可以得到许多的好处。
6. 购买其它在开发中要用到的设备，如总线分析仪、USB Hubs、竞争者的相应 USB 设备以及 USB 鼠标 USB 键盘等。

- 7. 编写或改写 USB 驱动程序、MCU 固件程序和客户程序代码。
- 8. 参加 USB 遵从试验，在不同的系统与配置验证你的 USB 设备。

驱动程序编写

USB 的驱动程序和以往的直接跟硬件打交道的 WIN95 的 VXD 的方式的驱动程序不同，它应该是 WDM 类型的。Windows 下的 USB 软件总体框图如图 2。

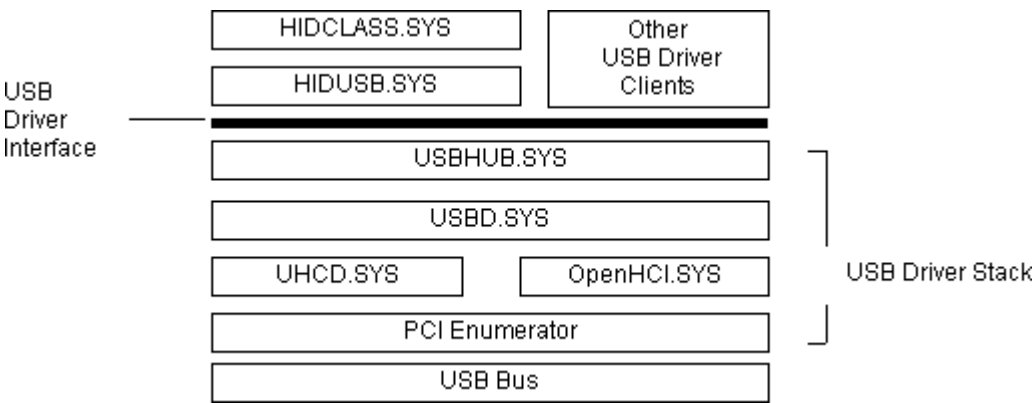


图 2：USB 软件总体框图