USB架构简介

通用串行总线（英语：Universal Serial Bus，缩写：USB）是连接计算机系统与外部设备的一种串口总线标准，也是一种输入输出接口的技术规范，被广泛地应用于个人电脑和移动设备等信息通讯产品。最初是由英特尔与微软公司倡导发起，其最大的特点是支持热插拔和即插即用。

**USB架构规范的历史版本**

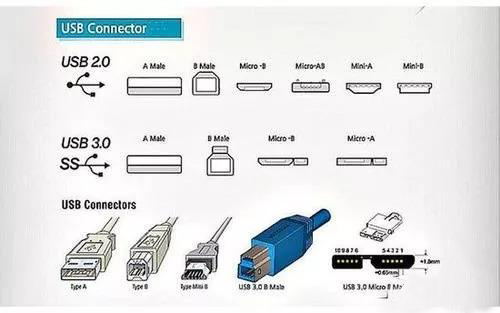
一、USB 1.0是在1996年出现的，速度只有1.5Mb/s(位每秒)；

二、USB1.1是较为普遍的USB规范，其高速方式的传输速率为12Mbps，低速方式的传输速率为1.5Mbps（b是Bit的意思），b/s 一般表示位传输速度，bps 表示位传输速率，数值上相等。B大部分MP3为此类接口类型。

三、USB2.0规范是由USB1.1规范演变而来的。它的传输速率达到了480Mbps，折算为MB为60MB/s，足以满足大多数外设的速率要求。USB 2.0中的“增强主机控制器接口”（EHCI）定义了一个与USB 1.1相兼容的架构。它可以用USB 2.0的驱动程序驱动USB 1.1设备。也就是说，所有支持USB 1.1的设备都可以直接在USB 2.0的接口上使用而不必担心兼容性问题，而且像USB 线、插头等等附件也都可以直接使用。

使用USB为打印机应用带来的变化则是速度的大幅度提升，USB接口提供了12Mbps的连接速度，相比并口速度提高达到10倍以上，在这个速度之下打印文件传输时间大大缩减。USB 2.0标准进一步将接口速度提高到480Mbps，是普通USB速度的20倍，更大幅度降低了打印文件的传输时间。

四、USB 3.0的理论速度为5.0Gb/s，其实只能达到理论值的5成，那也是接近于USB 2.0的10倍了。USB3.0的物理层采用8b/10b编码方式，这样算下来的理论速度也就4Gb/s，实际速度还要扣除协议开销，在4Gb/s基础上要再少点。可广泛用于PC外围设备和消费电子产品。



USB协议

USB系统可以分为两个系统：

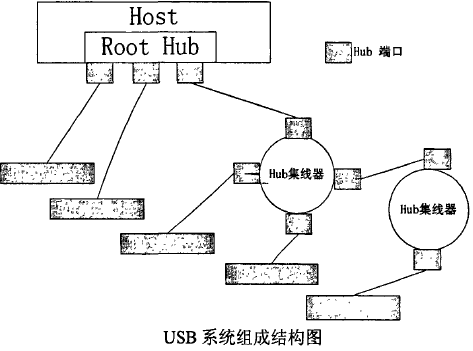
1. 主机系统HOST
2. 设备系统Device

USB主机系统

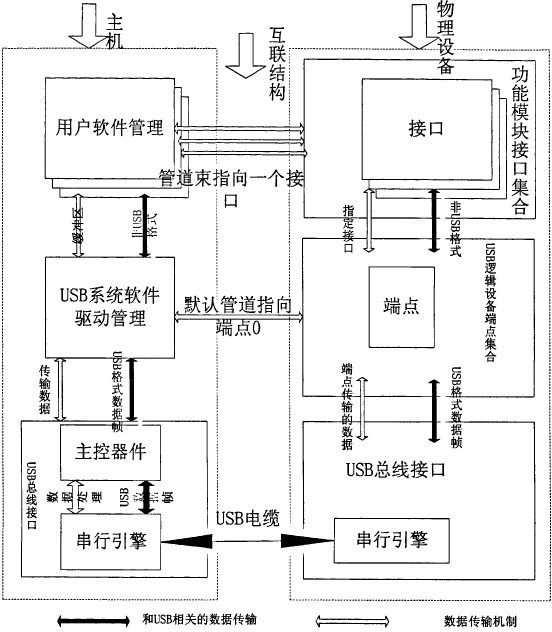
  在USB主机系统中，通过根集线器与外部USB从机设备相连的处理芯片，称为USB主机控制器。USB主机控制器包含硬件、软件和固件一部分。

USB设备系统

USB设备按功能可分为两部分：集线器Hub、功能部件。如下图所示，主机通过根集线器连接到各种外围设备（集线器、功能部件）。



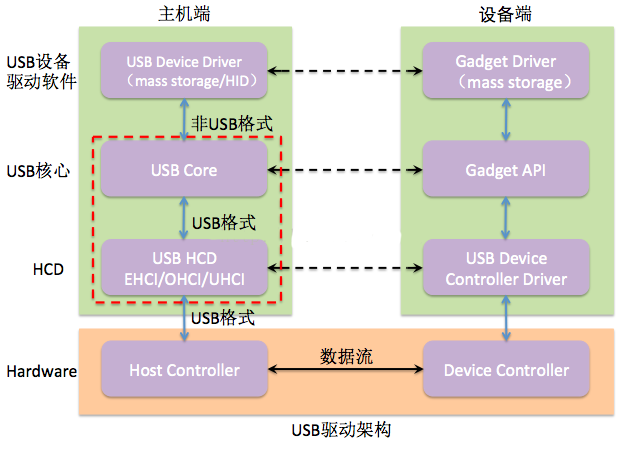
USB 通信模型



上图展示了USB主机和USB设备之间的数据传输过程。在设备端，USB设备将非USB格式的数据进行打包处理，转换成USB格式的数据包，然后传递到链路层，经过硬件处理、传递到物理层，由物理层通过PHY以数据流的形式传输到主机。

       在USB设备和USB主机之间发起的传输过程，称为事务。每次事务以2到3个数据包的形式进行USB总线传输。每个数据包包含2到3个步骤：  
       1) USB主机控制器向USB设备发出命令  
       2) USB控制器和USB设备之间传递读写请求，其方向取决于第一部分的命令是读还是写  
       3) 握手信号。  
            USB主机控制器向USB设备发送事务类型请求，通过分组标识符来进行识别。

下图展示了从主机驱动设备上的应用的USB驱动架构。



**设备端**

**应用程序**

USB core

USB Core包含Host Core Driver（主机核心驱动）、Hub Driver（总线驱动），简称USBD ，独立于硬件的协议栈，支持USB Host协议、USB Hub功能。

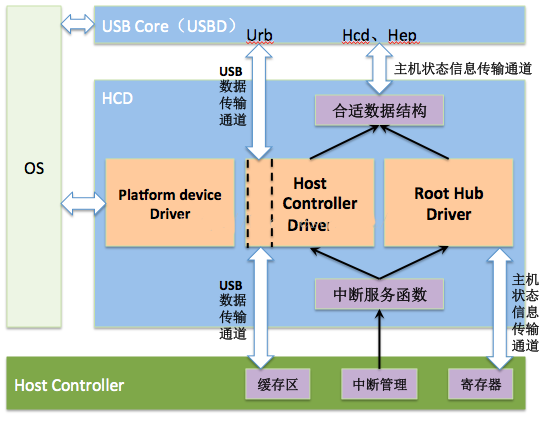
USB HCD（Host Controller Driver/主机控制器驱动程序）

硬件之上运行的是USB HCD，USB HCD包含三种USB接口规范：

（1）UHCI：通用主机控制接口，USB1.0/1.1；

（2）OHCI：开放主机控制接口，USB1.0/1.1；

（3）EHCI：增强主机控制接口，USB2.0；

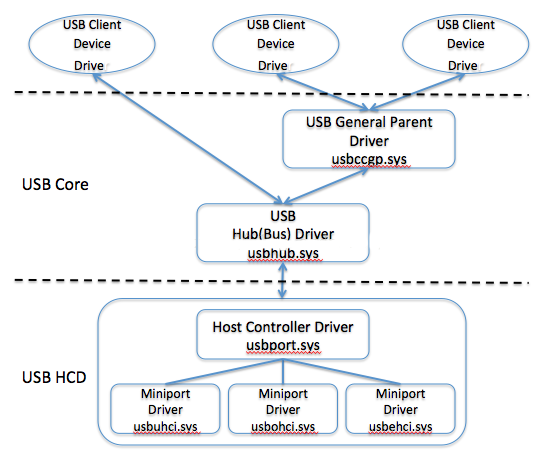


其中，USB HCD包含了Platform Device Driver（平台设备驱动程序）、Host Controller Driver、Root Hub Driver。

USB Core和HCD构成了USB驱动的核心部分，在上层就是用各种USB设备的厂商提供相应的驱动程序。

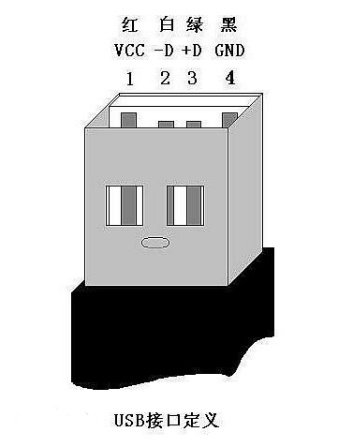
USB设备工作流程

当设备功能驱动希望向某个管道发出读写请求时，首先构造请求（USB Request Block，URB）发给USB总线驱动（USBD中的Hub Driver），USB总线驱动解释该请求，并转换请求发往USB主机控制器端口驱动（usbport.sys，属于USB HCD），进而通过小端口驱动（usbohci.sys等，属于HCD）与设备进行通信。具体如下图所示：



USB数据传输结构

 一条USB的传输线分别由地线、电源线、D+、D-四条线构成，D+和D-是差分输入线(抗干扰)，它使用的是3.3V的电压，而电源线和地线可向设备提供5V电压，最大电流为500MA。OTG 的做法就是增来一个 ID pin 来判断设备是接入设备的是主还是从。VCC 主要是供电， D+/D- 则是用来传输数据，就是我们前面所讲的主设备和从设备间唯一的一条铁路。



|  |  |  |
| --- | --- | --- |
|  | 信号线名称 | 颜色 |
| 1 | VCC | 红 |
| 2 | D- | 白 |
| 3 | D+ | 绿 |
| 4 | GND | 黑 |
| shell （金属壳） | 屏敝层 |  |

差分输入方式：差分输入的是将两个输入端的差值作为信号，其优点如下：

（1）差分信号在低电平应用时非常有效。因为如果一个信号的电平非常低，那么这个信号就比较容易受到其他信号的干扰，而差分信号使这个信号的电平加倍。  
　　（2）由于差分信号是电平相同而且反向的两根信号，不需要参考平面作为信号回路，这样就不需要地平面或电源平面的连续性和完整性。  
　　（3）如果两根信号都存在噪声干扰，就可以通过相减来抵消噪声，因此差分信号对信号干扰有着天然的免疫力。  
　　（4）相对于单点信号，差分信号的开关时序会比较精确，因为单点信号受到噪声干扰后时序或占空比通常会发生变化。

USB协议的四种传输类型

　　因为usb支持的设备实在是太多，而且不同的设备对于传输数据各有各的要求和这就导致了我们需要不同的传输方式。USB支持4种传输方式：控制传输；批量传输；中断传输；实(等)时传输。

**控制传输**：首先发送 Setup 传输事务，然后IN/OUT传输事务，最后是 STATUS transaction，向主机汇报前面SETUP 和 IN/OUT阶段的结果。控制传输主要用于向设备发送配置信息、获取设备信息、发送命令道设备，或者获取设备的状态报告。控制传输一般发送的数据量较小，当USB设备插入时，USB核心使用端点0对设备进行配置，另外，端口0与其他端点不一样，端点0可以双向传输。

**批量传输**：由OUT事务和IN事务构成，用于大容量数据传输，没有固定的传输速率，也不占用带宽，当总线忙时，USB会优先进行其他类型的数据传输，而暂时停止批量转输。批量传输通常用在数据量大、对数据实时性要求不高的场合，例如USB打印机、扫描仪、大容量存储设备、U盘等。

**中断传输**：由OUT事务和IN事务构成，中断传输就是中断端点以一个固定的速度来传输较少的数据，USB键盘和鼠标就是使用这个传输方式。这里说的中断和硬件上下文中的中断不一样，它不是设备主动发送一个中断请求，而是主机控制器在保证不大于某个时间间隔内安排一次传输。中断传输对时间要求比较严格，所以可以用中断传输来不断地检测某个设备，当条件满足后再使用批量传输大量的数据。

**等时传输**：由OUT事务和IN事务构成，有两个特殊地方，第一，在同步传输的IN和OUT事务中是没有握手阶段；第二，在数据包阶段所有的数据包都为DATA0 。等时传输同样可以传输大批量数据，但是对数据是否到达没有保证，它对实时性的要求很高，例如音频、视频等设备（USB摄像头，USB话筒）。