[Linux USB 3.0驱动分析（一）—— USB设备基础概念](https://www.cnblogs.com/wen123456/p/14212640.html)

**一.基础概念**

在终端用户看来，USB设备为主机提供了多种多样的附加功能，如文件传输，声音播放等，但对USB主机来说，它与所有USB设备的接口都是一致的。一个USB设备由3个功能模块组成：**USB总线接口**、**USB逻辑设备**和**功能单元：**

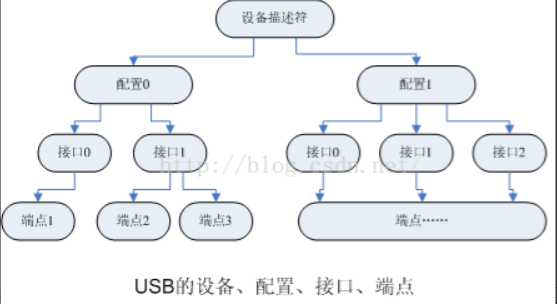
a -- 这里的USB总线接口指的是USB设备中的串行接口引擎（SIE）；

b -- USB逻辑设备被USB系统软件看作是一个**端点的集合**；

c -- **功能单元**被客户软件看作是一个**接口的集合**。SIE、端点和接口都是USB设备的组成单元；

        为了更好地描述USB设备的特征，USB提出了**设备架构**的概念。从这个角度来看，可以认为USB设备是由一些**配置**、**接口**和**端点**组成，即一个USB设备可以含有一个或多个配置，在每个配置中可含有一个或多个接口，在每个接口中可含有若干个端点。其中，**配置和接口是对USB设备功能的抽象**，实际的数据传输由**端点**来完成。在使用USB设备前，必须指明其采用的**配置和接口**。这个步骤一般是在设备接入主机时设备进行枚举时完成的

这些单元之间的关系如下：



设备通常有一个或多个配置；

配置通常有一个或多个接口；

接口通常有一个或多个设置；

接口有零或多个端点。

       这样的概念太抽象了，可以这样看：有一个设备，如支持视频和音频的一个播放器。那么，对于上面提到的4个描述符，对它们设置的时候，它们分别对于哪一个描述符呢？

      从我现在的理解来看，这样**一个设备对应一个设备描述符**，支持视频的功能对应一个**接口描述符**，支持音频功能的对应一个接口描述符。为了支持视频，在下层有多个端口同时工作为提供视频数据传输的支持，所以有**多个端点描述符**。  
**二.USB描述符**

      USB设备使用各种描述符来说明其设备架构，包括设备描述符、配置描述符、接口描述符、端点描述符和字符串描述符，他们**通常被保存在USB设备的固件程序中**。

**路径：include/uapi/linux/usb/ch9.h**

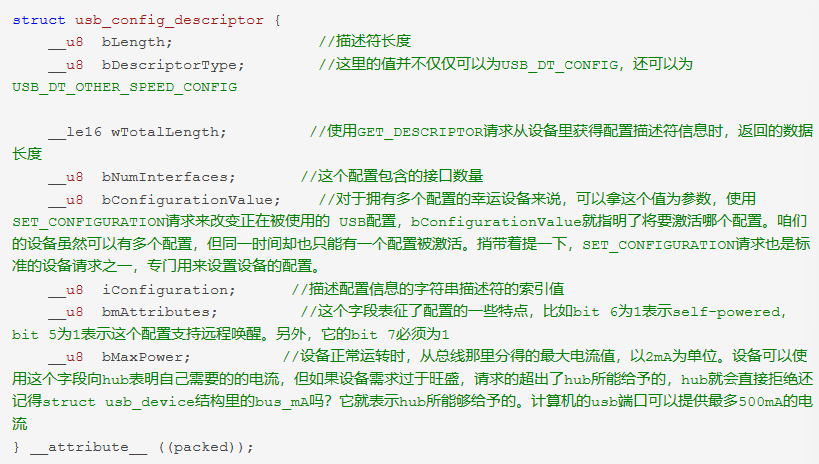
**1、设备描述符**

      设备代表一个USB设备，它由一个或多个配置组成。设备描述符用于说明设备的总体信息，并指明其所含的配置的个数。**一个USB设备只能有一个设备描述符**。



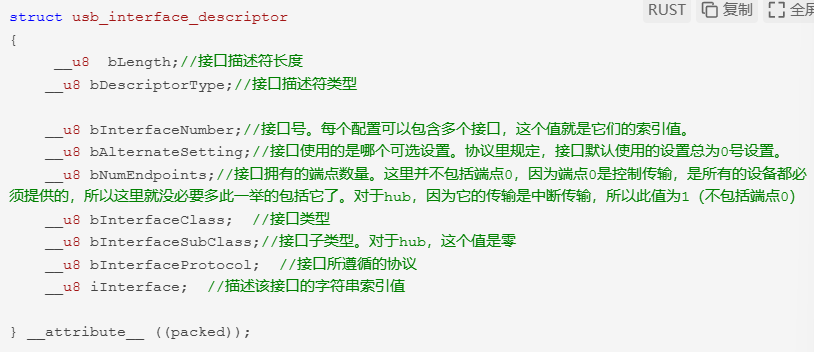
**2、配置描述符**

       一个USB设备可以包含一个或多个配置，如USB设备的低功耗模式和高功耗模式可分别对应一个配置。在使用USB设备前，必须为其选择一个合适的配置。配置描述符用于说明USB设备中各个配置的特性，如配置所含接口的个数等。USB设备的每一个配置都必须有一个配置描述符。



**3、接口描述符**

      一个配置可以包含一个或多个接口，例如对一个光驱来说，当用于文件传输时，使用其大容量存储接口；而当用于播放CD时，使用其音频接口。接口是端点的集合，可以包含一个或多个可替换设置，用户能够在USB处于配置状态时改变当前接口所含的个数和特性。接口描述符用于说明设备中各个接口的特性，如接口所属的设备类及其子类等。USB设备的每个接口都必须有一个接口描述符



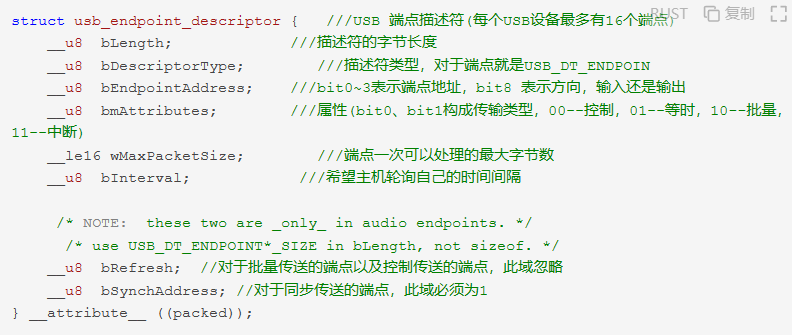
**4、端点描述符**

      端点是USB设备中的实际物理单元，**USB数据传输就是在主机和USB设备各个端点之间进行的**。端点一般由USB接口芯片提供，例如Freescale公司的MC68HC908JB8和MC9S12UF32。USB设备中的每一个端点都有唯一的端点号，每个端点所支持的数据传输方向一般而言也是确定的：或是输入（IN），或是输出（OUT）。也有些芯片提供的端点的数据方向是可以配置的，例如MC68HC908JB8包含有两个用于数据收发的端点：端点1和端点2。其中端点1只能用于数据发送，即支持输入（IN）操作；端点2既能用于数据发送，也可用于数据接收，即支持输入（IN）和输出（OUT）操作。而MC9S12UF32具有6个端点。

     利用设备地址、端点号和传输方向就可以指定一个端点，并与它进行通信。端点的传输特性还决定了其与主机通信是所采用的传输类型，例如控制端点只能使用控制传输。根据端点的不同用途，可将端点分为两类：0号端点和非0号端点。

**0号端点比较特殊，它有数据输入IN和数据输出OUT两个物理单元，且只能支持控制传输**。所有的USB设备都必须含有一个0号端点，用作默认**控制管道**。USB系统软件就是使用该管道与USB逻辑设备进行配置通信的。0号端点在USB设备上的以后就可以使用，而非0号端点必须要在配置以后才可以使用。

     根据具体应用的需要，USB设备还可以含有多个除0号端点以外的其他端点。对于低速设备，其附加的端点数最多为2个；对于全速/高速设备，其附加的端点数最多为15个。



**5、字符串描述符**

      在USB设备中通常还含有字符串描述符，以说明一些专用信息，如制造商的名称、设备的序列号等。它的内容以UNICODE的形式给出，且可以被客户软件所读取。对USB设备来说，字符串描述符是可选的。



**6、管道**

      在USB系统结构中，可以认为数据传输时在USB主机软件与USB设备的各个端点之间直接进行的，它们之间的连接称为管道。管道是在USB设备的配置过程中建立的。管道是对USB主机与USB设备间通信流的抽象，表示USB主机的数据缓冲区与USB设备的端点之间存在着逻辑数据传输，而实际的数据传输是由USB总线接口层来完成的。

     管道与USB设备中的端点一一对应。一个USB设备含有多少个端点，其与USB主机进行通信时就可以使用多少条管道，且端点的类型决定了管道中数据的传输类型，例如中断端点对应中断管道，且该管道只能进行中断传输。不论存在着多少条管道，在各个管道中进行的数据传输都是相互独立的。

**7、USB端点分类**

      USB 通讯的最基本形式是通过端点。一个USB端点只能向一个方向传输数据（从主机到设备(称为输出端点)或者从设备到主机(称为输入端点)）。端点可被看作一个单向的管道。

     USB 端点有 4 种不同类型, 分别具有不同的数据传送方式：

**1) 控制CONTROL**

    控制端点被用来控制对USB设备的不同部分访问. 通常用作配置设备、获取设备信息、发送命令到设备或获取设备状态报告。这些端点通常较小。每个 USB 设备都有一个控制端点称为"端点 0", 被 USB 核心用来在插入时配置设备。USB协议保证总有足够的带宽留给控制端点传送数据到设备.

**2) 中断INTERRUPT**

     每当 USB 主机向设备请求数据时，中断端点以固定的速率传送小量的数据。此为USB 键盘和鼠标的主要的数据传送方法。它还用以传送数据到USB设备来控制设备。通常不用来传送大量数据。USB协议保证总有足够的带宽留给中断端点传送数据到设备.

**3) 批量BULK**

    批量端点用以传送大量数据。这些端点通常比中断端点大得多. 它们普遍用于不能有任何数据丢失的情况。USB 协议不保证传输在特定时间范围内完成。如果总线上没有足够的空间来发送整个BULK包，它被分为多个包进行传输。这些端点普遍用于打印机、USB Mass Storage和USB网络设备上。

**4) 等时ISOCHRONOUS**

    等时端点也批量传送大量数据, 但是这个数据不被保证能送达。这些端点用在可以处理数据丢失的设备中，并且更多依赖于保持持续的数据流。如音频和视频设备等等。

    控制和批量端点用于异步数据传送，而中断和等时端点是周期性的。这意味着这些端点被设置来在固定的时间连续传送数据，USB 核心为它们保留了相应的带宽。