USB HID 设备应用（进阶篇）

目录

HID（Human Interface Device，人机接口设备）是 USB 设备中常用的设备类型，是直接与人交互的 USB 设备，例如键盘、鼠标与游戏杆等。在 USB 设备中，HID 设备的成本也较低。

HID 类是 Windows 完全支持的第一批 USB 设备类型中的一种。在 Windows 98 以及后来的版本中内置有 HID 设备的驱动程序，应用程序可以直接使用这些驱动程序来与设备通信。由于这个原因，符合 HID 类 USB 设备才就可很容易的设置运行。

本文详细介绍了 HID 设备固件基础、HID 设备描述符、HID 的特定请求，最后通过 USB 协议分析仪捕获 USB 总线数据包，分析 HID 设备枚举过程中，HID 设备与 HOST 的通信数据。

1. HID 设备固件基础

1.1 什么是 HID 设备？

在你知道能否使用 Windows 的 HID 驱动和设备通信之前，你需要了解设备是否符合 HID 类。

人机接口这个词表明设备直接和人交互。当人按下键或移动鼠标或游戏杆时，设备可以检测到，或者主机发送游戏杆的影响来给用户去体验。HID 的典型例子是键盘、鼠标和游戏杆。其他的 HID 包括把柄、开关、按钮和滑杆的前端面板、遥控、电话按键板，游戏控制端比如数据手套和转向轮等。

但是 HID 不是必须要有一个人机接口的。它只是需要能在类规范的限制内起到一定作用。下面是 HID 类的主要功能和限制：

- 交换的数据储存在称为报表（Report）的结构内，设备的固件必须支持 HID 报表的格式。主机通过控制和中断传输中的传送和请求报表来传送和接收数据。报表的格式非常灵活。
- 每一笔事务可以携带小量或中量的数据。低速设备每一笔事务最大是 8B，全速设备每一笔事务最大是 64B，高速设备每一笔事务最大是 1024B。一个报表可以使用多笔事务。
- 设备可以在未预期的时间传送信息给主机，例如键盘的按键或是鼠标的移动。所以主机会定时轮询设备，以取得最新的数据。
- HID 设备的最大传输速度有限制。主机可以保证低速的中断端点每 10ms 内最多 1 笔事务，即每秒最多是 800B。保证全速端点每 1ms 内最多一笔事务，即每秒最多是 64000B。保证高速端点每 125 us 三笔事务，即每秒最多是 24.576MB。
- HID 设备没有保证的传输速率。如果设备是设置在 10ms 的间隔，则事务之间的时间可能等于或小于此值。配置成每 1ms 一个事务的设备是例外。因为这是最快可能的轮询速率，设备可以保证这个速率。

HID 设备除了传送数据给主机外，它也会从主机接收数据。只要能够符合 HID 类别规范的设备都可以是 HID 设备。

设备除了 HID 接口之外，它可能同时还包含有其他的 USB 接口。例如影像显示设备可能使用 HID 接口来做亮度、对比度的软件控制，而使用传统的影像接口来传送要显示的数据。USB 扩音器可以使用实时传输来播放语音，同时使用 HID 接口来控制音量、低音等。

HID 类别设备的规范文件主要是以下两份：

- Device Class Definition for Human interface Devices（人机接口设备的设备类定义）；
- HID Usage Tables（HID 使用表）

其中前者是 HID 的基本规范文件，它定义 HID 类，后者定义了有助于主机理解和使用 HID 数据的值得表，为开发人员提供实际的控制类型的描述。这两份文件是由 USB Device Working Group 制定的，可以在网上下载。

1.2 HID 设备硬件要求

HID 接口必须符合 Device Class Definition for Human interface Devices 规范内所定义的 HID 类别的需求。在此文件内描述了所需的描述符、传输的频率以及传输的类型等。为了符合规范，HID 接口的端点与描述符都必须符合数个要求。

所有的 HID 传输都是使用默认控制管道或是一个中断管道，HID 设备必须有一个中断输入端点来传送数据到主机，中断输出端点则不是必需的。

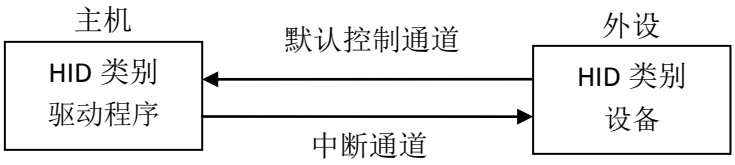


图 1 HID 传输的传输的类型

表 1 HID 设备的传输类型

传输类型	数据来源	数据类型	是否需要通道
控制	设备（输入）	没有严格时间限制的数据	是
	主机（输出）	没有严格时间限制的数据或是没有中断输出管道时的任何数据	
中断	设备（输入）	定时或低延迟的数据	是
	主机（输出）	定时或低延迟的数据	是

主机与设备之间所交换的数据，可以分成两种类型：

- 低延迟的数据，必须尽快地到达目的；
- 配置或其他的数据，没有严格时间限制的需求。

中断管道是控制管道之外的另一种数据交换的方式，特别适合使用在接收端需定时或是尽可能及时收到数据的时候。中断输入管道携带数据到主机，中断输出管道则是携带数据到设备。在总线忙的时候，控制管道可能会被延迟，而中断管道保证会有可得到的带宽。HID 不需要一定有中断输出管道。如果没有中断输出管道，主机会在控制管道上使用 HID 设备特有的 Set_Report 请求来传送所有的报表。

1.3 HID 固件的请求

主机的驱动程序要与 HID 设备通信，设备的固件必须符合下列需求：

- 设备的描述符必须识别该设备包含有 HID 接口。
- 除了默认控制管道外，固件必须另外支持一个中断输入管道。
- 固件必须包含一个报表描述符来定义要传送与接收的设备数据。

如果要传送数据，固件必须支持 Get_Report 控制传输与中断输入传输。如果要接收数据，固件必须支持 Set_Report 控制传输与选择性的中断输出传输。

所有的 HID 数据都必须使用定义过的报表格式来定义报表中数据的大小与内容。设备可以支持一个或多个报表。在固件中的报表描述符用来描述了此报表，以及如何使用报表数据的信息。

在每一个报表中的一个数值，定义报表是一个输入（Input）、输出（Output）或是特征（Feature）报表。主机在输入报表中接收数据，在输出报表中传送数据，特征报表可以在任何方向传递。

Windows 98 以及后来版本的 HID 驱动程序使用中断传输来传递输入报表。输出报表的传输类型要根据设备支持的端点与 Windows 的版本而定。Windows 98 Gold 只符合 HID 1.0 规范，它的 HID 驱动程序使用控制传输来传递输出报表。Windows 98 SE、Windows 2000 符合 HID 1.1 规范，HID 驱动程序在有中断输出端点时使用中断传输，否则使用控制传输来传递输出报表。特征报表都是使用控制传输。

1.4 怎样识别一个设备为 HID？

对于任何 USB 设备，HID 的描述符告诉了主机为了和设备通信，它需要知道什么。当主机发送 Get_Descriptor 来请求包含 HID 的接口配置时，主机就知道了 HID 接口。配置的接口描述符设别设备为 HID。HID 类描述符指定了接口支持的报表描述符的数量。在枚举过程中，HID 驱动得到了 HID 类和报表描述符。

2. HID 设备相关描述符分类

HID 设备连接到 USB 主机后，主机通过发送 Get_Descriptor 请求读取 HID 设备的描述符，了解描述符对了解 USB 设备是至关重要的。

2.1 HID 设备的描述符

HID 设备除了支持 USB 设备的 5 种标准描述符之外，还支持 HID 设备特有的 3 种描述符。这些描述符是：

- USB 标准描述符：设备、配置、接口、端点和字符串描述符。
- HID 特有的描述符：HID、报表（Report）和实体（Physical）描述符。

从描述符的关联关系看，HID 描述符是关联于接口。所以如果一个 HID 设备有 2 个端点，设备不需要每个端点有一个 HID 描述符。

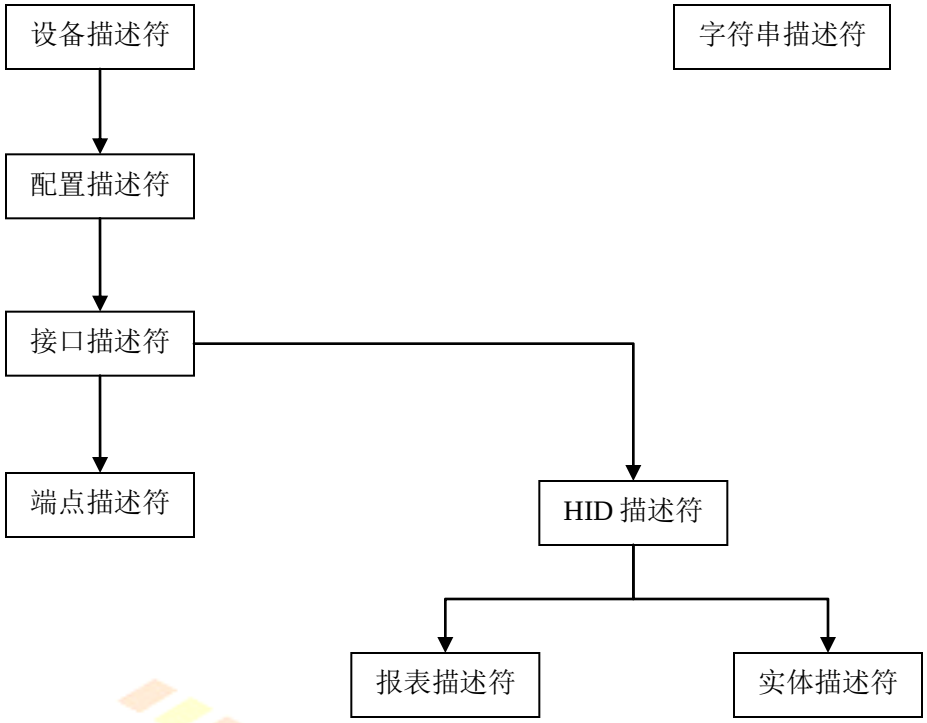


图 2 HID 描述符的关联关系

从前面的 USB 描述符可以看出一个规律，描述符的第一、二字节分别是描述符的长度和类型，描述符的类型字段（bDescriptorType）表明描述符的种类，下表列出了不同描述符的类型字段数值。

表 2 HID 的描述符

类型	描述符	应用	数值
标准	设备 Device	所有设备必须有，只能一个	01
	配置 Configuration	所有设备必须有，至少一个	02
	字符串 String	可选择	03
	接口 Interface	每一个接口一个	04
	端点 Endpoint	除端点 0 之外的每个端点一个	05
	设备限定 Device_Qualiffier	同时支持全速与高速的设备必须有一个	06
	Other_Speed_Configuration		07
	Interface_power		08
类别	HID	HID 设备必须有	21
	Hub		29
	报表 Report（HID 相关）	HID 设备必须有	22
	实体 Physical（HID 相关）	可选择的	23

对于一个 HID 设备，设备描述符与配置描述符没有 HID 特定的信息。其设备描述符的 bDeviceClass 和 bDeviceSubClass 字段的值为 0，接口描述符的 bInterfaceClass 字段值为 03，表示设备的该接口是 HID 类别。在接口描述符中其他包含 HID 特定信息的字段还有子类别码（bInterfaceSubClass）与协议码（bInterfaceProtocol 字段）。

在接口描述符中子类别码字段等于 1 表示此设备支持启动接口（Boot Interface）。如果设备有启动接口，即便主机的 HID 没有加载驱动程序，设备也可以使用。这种情形可能发生在计算机是由 DOS 直接启动，在启动时观看系统设置画面或使用 Wndows 的安全模式时。

含有启动接口的键盘或鼠标可以使用 BIOS 或许多主机支持的默认简单协议。HID 规范定义了键盘与鼠标的启动接口协议。

如果设备没有启动接口，并且接口描述符中协议码字段是 1，表示设备支持键盘接口，协议码字段是 2，表示支持鼠标接口。接口描述符中协议码字段是 0，表示设备不支持启动协议。

在 HID Usage Tables 规范中定义了键盘与鼠标的启动描述符（Boot Descriptor）。BIOS 不需要从设备中读取描述符，因为它知道启动协议，并且假设设备支持启动协议。所以要启动的设备不需要在固件内包含启动接口描述符，它只要在主机尚未要求在报表描述符中的定义协议时支持启动协议即可。在操作系统加载 HID 驱动程序后会使用 Set_Protocol 请求，将设备由启动协议转换成报表协议。

2.2 HID 描述符

HID 描述符的主要目的是识别 HID 通信中使用的其他描述符。类描述符可以有 7 个或更多个字段，这取决于其他描述符的数量。表 3 显示了这些字段。

表 3 HID 描述符结构

偏移量	字段	字节数	数值类型	说明
0	bLength	1	Numeric	描述符字节数
1	bDescriptorType	1	Constant	0x21 = HID 描述符
2	bcdHID	2	Numeric	HID 规范版本号（BCD）
4	bCountryCode	1	Numeric	硬件设备所在国家的国家代码
5	bNumDescriptors	1	Numeric	类别描述符数目（至少有一个报表描述符）
6	bDescriptorType	1	Constant	类别描述符的类型
7	wDescriptorLength	2	Numeric	报表描述符的总长度
9	[bDescriptorType]...	1	Constant	附加的描述符的类型，可选的
10	[wDescriptorLength]...	2	Numeric	附加的描述符的总长度，可选的

bcdHID：设备与其描述符所遵循的 HID 规范的版本号码，此数值是 4 个 16 进位的 BCD 格式字符。例如版本 1.1 的 bcdHID 是 0110h。

bCountryCode：硬件目的国家的识别码。如果不说明，该字段为 0。

bDescriptorType：HID 描述符附属的描述符的类型（报表或实体）。每一个 HID 都必须至少支持一个报表描述符。一个接口可以支持多个报表描述符，以及一个或多个实体描述符。

HID 描述符的偏移量为 9 和 10 的 bDescriptorType 和 wDescriptorLength 可以重复存在多个。

2.3 报表描述符

报表描述符定义了执行设备功能的数据格式和使用方法。如果设备是鼠标，则数据报表鼠标的运动按钮点击。如果设备是延迟控制器，则数据包含了指定开关延迟的代码。

报表描述符需要足够的灵活，来处理设备不同的功能。格式应该简单，这样当传输数据时不会浪费设备的存储空间或总线时间。HID 报表描述符通过付出了比较原始数据更复杂的格式的代价来实现上述目的。这样格式不限制报表里数据类型，但是报表描述符必须提前描述报表的大小和内容。报表描述符的内容和长度随设备而不同，可以短而简单，长复杂，或者介于二者之间，报表描述符的结构及内容细节介绍，将在后面开辟独立章节介绍。

报表描述符是一种类描述符。主机通过发送 wValue 高位字节为 22H 的 Get_Descriptor 请求来获得报表描述符。默认的报表数量为 00h。

获取报表描述符包含的内容和它的结构的一个途径是查看一个报表。程序表单表 4 是一个空骨架报表描述符，它描述了一个发送两个字节数据到主机的输入报表和一个发送两个字节数据到设备的输出报表。其他报表描述符建立在这种基本格式的基础上，因此通常从像这样的一个短描述符开始了解描述符是一个很好的方法。

表 4 报表描述符

```
db  06h, A0h, FFh   ;   Usage Page(Vendor defined)    定义设备功能
db  09h, A5h        ;   Usage(Vendor Defined)         定义用法
db  A1h, 01h        ;   Collection(Application)       开一个集合
db  09H, A6h        ;   Usage(Vendor defined)         定义用法


                    ;   输入报表
db  09h, A7h        ;   Usgae(Vendor defined)       定义用法
db  15h, 80h        ;   Logical Minimum             定义输入最小值=-128
db  25h, 7Fh        ;   Logical Maximum             定义输入最大值=+27
db  75h, 08h        ;   Report Size                 定义报表数据项大小=8
db  95h, 02h        ;   Report Count                定义报表数据向个数=2
db  81h, 02h        ;   Input(Data,Variable,Absolute)   输入项目


                    ;   输出报表
db  09h, A9h        ;   Usgae(Vendor defined)       定义用法
db  15h, 80h        ;   Logical Minimum             定义输入最小值=-128
db  25h, 7Fh        ;   Logical Maximum             定义输入最大值=+27
db  75h, 08h        ;   Report Size                 定义报表数据项大小=8
db  95h, 02h        ;   Report Count                定义报表数据向个数=2
db  91h, 02h        ;   Output(Data,Variable,Absolute)    输出项目


db  C0h             ;   End Collection                    关闭集合
```

在范例描述符的行是所有描述符都要求的。一些行应用于整个描述符，而其他的只单独用于输入和输出数据。更多复杂的报表描述符可以使用这些相同行的其他情况，同时还有其他的可选行。

在范例报表里每一行都是由包括表征这一行的 1 个字节和一个或多个包含行数据的字节组成的。下面是范例描述符里每一行所表示的内容。

用法页：行是由值 06h 表示的，它指出了这个设备的总的功能，例如一般的桌面控制、游戏控制或数字显示。你可以认为用法页是 HID 类的一个子集。在范例描述符里，供应商定义用法页的值是 FFA0hh。HID 规范列出了不同用法页的值和供应商定义的用法页的保留值。

用法行：由值 09h 表征，它指定了单个报表的功能。就像用法页是类的一个子集一样，用法行也是用法页的一个子集。举例来说，用法是提供给一般的桌面控制包括鼠标、游戏杆和键盘。因为范例的用法页是供应商定义的，所以用法页里的所有用法也是由供应商定义的。例子中的用法页是 A5h。

集合（应用程序）行开始了一个行的组，这个组一起执行一个类似应用程序的功能，例如鼠标或键盘。每个报表描述符必须有一个应用程序集合以使得 Windows 能正确列举它。集合行后面的用法行命名了集合的功能。在例子中，它是供应商定义的值 A6。

逻辑最小和最大值是 15h 和 25h，并指定了报表可以包含范围的值的范围。负值可以被表示为 2 的补数。在例子里，值 80h 和 7Fh 表示范围为-128~+127。

报表大小行值为 75h，表明每个报表数据行里有多少个位。例子里每个数据行是 8 位。

报表计数行值为 95h，表明报表包含多少数据行。例子里每个报表包含两个数据行。

最后一行指出了报表是从主机传输数据到设备（91h），还是从设备到主机（81h），还有这个数据的其他信息。

终端集合：行结束了应用程序集合。

3. HID 的特定请求

除了 USB 设备的 11 个标准请求外，HID 规范另外还定义了 6 个 HID 特定控制请求，表 5 列举出了这些请求，下面的内容中更详细的描述了每个请求。

所有的 HID 设备都必须支持 Get_Report 请求，同时支持启动的设备必须支持 Get_Protocol 请求和 Set_Protocol 请求，其他的请求是可选择的。如果设备没有中断输出端点，此设备需要支持 Get_Report 请求来从主机读取数据。

表 5 HID 特定请求

bmRequestType	bRequest	wValue	wIndex	wLength	数据阶段
1 01 00001b	Get_Report (1)	报表类型，报表 ID	接口	报表长度	报表
0 01 00001b	Set_Report (9)	报表类型，报表 ID	接口	报表长度	报表
1 01 00001b	Get_Idle (2)	0，报表 ID	接口	1	闲置时间
0 01 00001b	Set_Idle (10)	闲置时间，报表 ID	接口	0	无
1 01 00001b	Get_Protocol (3)	0	接口	1	0: 启动协议 1: 报表协议

0 01 00001b	Set_Protocol (11)	0: 启动协议 1: 报表协议	接口	0	无

3.1 Get_Report 请求

Get_Report 的作用是启用主机使用控制传输，来从设备读取数据。

在使用时 wValue 字段的高字节是报表类型，1 表示 Input 报表，2 表示 Output 报表，3 表示 Feature 报表。wValue 的低字节是报表的 Report ID，如果没有定义 Report ID，该字节为设 0。

在携带请求的控制传输的数据阶段，HID 设备回传指定的报表内容。

HID 规范不建议使用该请求获得未经定时的数据，这样的数据建议使用中断输入管道获得。

该请求用来取得在主机初始化设备时的特征项目状态和其他信息。使用开机协议的主机可以使用此请求来获得按键或鼠标数据。

3.2 Set_Report 请求

Set_Report 请求的参数含义和 Get_Report 一样，但 Set_Report 请求的数据方向与 Get_Report 相反，在后面的数据阶段，主机传送报表到 HID 设备，这样的输出报表可以用于复位设备的控制，复位产生的效果取决于对应的控制的类型是相对（Reletive）的还是绝对（Absolute）的。

3.3 Set_Idle 请求

Set_Idle 请求的作用是静默一个在中断输入管道的特定的报表，直到一个发生一个相关的事件或过去了规定的时间，当数据从上一个报表后没有改变时，可以通过限制中断输入端点的报表频率来节省传输带宽。HID 设备不是必需支持此请求。

wValue 字段的高字节是设置的闲置时间，是报表之间的最大间隔时间。该字节为 0 表示闲置时间为无限长，在这种情况下，设备只有在报表数据有改变时才传送报表，否则设备传回一个 NAK。

wValue 的低字节是报表的 Report ID。如果低字节是 0，此请求应用到设备的所有输入报表。

闲置时间以 4ms 为单位，范围在 4ms～1020ms 之间。如果报表的数据自从上一次报表后有改变，或是接收到一个请求，设备会传送一个报表。

如果报表的数据没有改变，而且从上一次报表后过去的时间自尚未达到规定的闲置时间，设备会传回一个 NAK。如果报表的数据没有改变，而且持续时间已经达到的闲置时间，设备会传送一个报表。

闲置时间设置为 0 表示无限长的闲置时间，设备只有在报表的数据有改变时才会传送一个报表，对于其他的中断输入请求则是传回 NAK。

在检测 HID 设备时，Windows 的 HID 驱动程序会试图将闲置时间设置成 0。如果 HID 设备不支持此请求，主机会收到传回的 Stall。

3.4 Get_Idle 请求

Get_Idle 请求的作用是过的设备的当前闲置时间，在数据阶段，HID 设备回传一个字节的闲置时间值。

3.5 Get_Protocol 请求

Get_Protocol 请求的作用是主机获取设备目前作用的是启动协议还是报表协议。

在数据阶段中设备回传的 1 个字节信息包中的数据值为 0 表示启动协议，为 1 表示报表协议。

启动设备必需支持该请求。

3.6 Set_Protocol 请求

Set_Protocol 的作用是主机指定设备使用启动协议或报表协议。

在数据阶段中主机传送的 1 个字节信息包中的数据值为 0 表示指定启动协议，为 1 表示指定报表协议。

启动设备必需支持该请求。

4. HID 报表描述符

上面内容已介绍了报表描述符的基本概念，本节就报表描述符的结构与内容作更加具体的介绍，以方便大家理解。

报表描述符定义了执行设备功能的数据格式和使用方法。报表描述符和 USB 的其他描述符是不一样的，它不是一个简单的表格，报表描述符是 USB 所有描述符中最复杂的。报表描述符非常复杂而有弹性，因为它需要处理各种用途的设备。报表的数据必须以简洁的格式来储存，这样才不会浪费设备内的储存空间以及数据传输时的总线时间。

实际上可以这样理解，报表内容的简洁，是通过报表描述符全面的、复杂的数据描述实现的。

报表描述符必须先描述数据的大小与内容。报表描述符的内容与大小因设备的不同而不同，在进行报表传输之前，主机必须先请求设备的报表描述符，只有得到了报表描述符才可正确解析报表的数据。

报表描述符是报表描述项目（Item）的集合，每一个描述项目都有相对统一的数据结构，项目很多，通过编码实现。

4.1 项目（Item）结构

报表描述符由描述 HID 设备的数据项目（Item）组成，项目的第一个字节（项目前缀）由三部分构成，即项目类型（item type）、项目标签（item tag）和项目长度（item size）。其中项目类型说明项目的数据类型，项目标签说明项目的功能，项目长度说明项目的数据部分的长度。

HID 的项目有短项目和长项目两种，其中短项目的格式如图 3 所示。

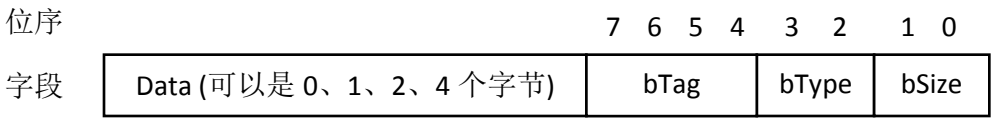


图 3 HID 报表短项目格式

短项目的数据字节数由 bSize 的值定义，bSize 为 0、1、2、3 时 Data 部分的字节数分别为 0、1、2、4 个字节。短项目的项目类型由 bType 定义，bType 为 0、1、2 时分别为 Main、Global 和 Local 类型。

长项目可以携带较多的数据，其格式如图 4 所示。

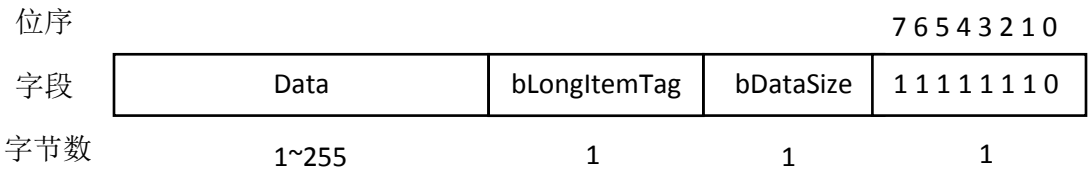


图 4 HID 报表长项目格式

项目中的第一个字节为上图中的特定值时表明该项目是一个长项目。长项目中的 bDataSize 说明 Data 部分的字节数，bLongItemTag 在 HID 规范中没有定义。

4.2 项目（Item）分类

报表的项目有 Main、Global 和 Local 三大类，每一类都有多个不同的项目，实现不同的描述。

Main 类项目用于定义报表描述符中的数据项。也可以组合其中的若干数据项成为一个集合。Main 项目可以分为带数据的 Main 项目和不带数据的 Main 项目。带数据项的 Main 用于生成报表中的数据项，包括 Input、Output 和 Feature 项目。不带数据的 Main 项目不生成报表中的数据项，包括 Collection 和 End Collection 项目。

Global 类项目实现对数据的描述，用来识别报表并且描述报表内的数据，包括数据的功能、最大与最小允许值以及数据项的大小与数目等。改变由 Main 类项目生成的项目状态表。Global 类项目描述对后续的所有项目有效，除非遇到有新的 Global 类项目。

Local 类项目定义控制的特征，这一类项目的作用域不超过下一个 Main 项目，所以在每一 Main 项目之前可能有多个 Local 项目。Local 项目用于描述后面的 Input、Output 和 Feature 项目。

表 6 列出的是全部的项目的前缀字和简要功能说明。

表 6 HID 项目列表

项目类型	项目标志（Tag）	项目前缀，nn 为数据长度	项目说明
Main 类项目	Input	1000 00 nn	定义输入报表，主机利用该信息解析设备提供的数据。主机向控制端口发送

			Get_Report实现输入
	Output	1001 00 nn	创建输出报表，通过向设备发送 Set_Report 实现输出
	Feature	1011 00 nn	定义送往设备的设置信息
	Collection	1010 00 nn	定义 2 个以上数据（Input、Output 和 Feature）的关系为集合，Collection 开始一个集合，之后的 End Collection 结束集合。Collection 项目的数据部分说明 Collection 的类型
	End Collection	1100 00 nn	
Global 类项目	Usage Page	0000 01 nn	指定设备的功能 另外由于Usage项目有32位数据值，Usage Page项目用于为Usage项目在报表描述符中占据存储空间。用于存放后续的Usage项目的高16位。
	Logical Minimum	0001 01 nn	定义变量或数组项目的逻辑最小值和最大值
	Logical Maximum	0010 01 nn	
	Physical Minimum	0011 01 nn	定义变量或数组项目的物理最小值和最大值，分别和 Logical Minimum、Logical Maximum 对应
	Physical Maximum	0100 01 nn	
	Unit Exponent	0101 01 nn	定义数值是基于 10 的指数
	Unit	0110 01 nn	单位
	Report Size	0111 01 nn	指定报表数据区域所包含的位数
	Report ID	1000 01 nn	报表 ID，该项目在报表中插入一个字节的报表 ID
	Report Count	1001 01 nn	报表中数据域的数目
	Push	1010 01 nn	将 Global 项目状态表送入堆栈
	Pop	1011 01 nn	从堆栈恢复 Global 项目状态表
		1100 01 nn –	保留

		1111 01 nn	
Local 类项目	Usage	0000 10 nn	用法索引值，表示对项目或集合建议的用法，用于当一个项目描述多个控制，对每一个变量和数组元素都有建议的用法
	Usage Minimum	0001 10 nn	定义阵列或位图中控制操作的第一个和最后一个用法
	Usage Maximum	0010 10 nn	
	Designator Index	0011 10 nn	确定用于控制的实体，指向物理描述符中的目标
	Designator Minimum	0100 10 nn	定义阵列或位图目标的起始和终止索引值
	Designator Maximum	0101 10 nn	
	String Index	0111 10 nn	确定字符串描述符中的索引值
	String Minimum	1000 10 nn	定义用于阵列或位图控制中字符串序列索引值的最小值和最大值
	String Maximum	1001 10 nn	
	Delimiter	1010 10 nn	定义一组 Local 项目的开始和结束，1=开始，0=结束
		1010 10 nn – 1111 10 nn	保留

在这些项目中，Usage Page 用来指定设备的功能，而 Usage 项目用来指定个别报表的功能。Usage Page 项目相当于是 HID 的子集合，Usage 相当于是 Usage Page 的子集合。

4.2.1 Input、Outpot 和 Feature 项目

这 3 个项目用来定义报表中的数据字段。

Input 项目可以应用到任何控制、计数器读数或其他设备传给主机的信息。一个输入报表包含一个或多个 Input 项目，主机使用中断输入传输来请求输入报表。

Ouput 项目用来定义主机传送给设备的信息。一个输出报表包含一个或多个 Outpot 项目。输出报表包含控制状态的数据。如果有中断输出管道，HID1.1 兼容主机使用中断输出传输来传送输出报表，否则使用 Set_Report 控制请求。

Feature 项目应用到主机传送给设备的信息，或是主机从设备读取 Feature 项目。一个特征报表包含一个或多个 Feature 项目，Feature 项目通常是包含影响设备与其组件整体行为的配置。特征报表通常是控制可以使用实际的控制面板调整的设置，例如主机可以使用虚拟控制面板来让用户选择控制特征。主机使用 Set_Report 与 Get_Report 请求来传送与接收特征报表。

在每一个 Input、Output 和 Feature 项目的前缀字之后是 32 位描述数据，目前最多定义了 9 个位，余的位则是保留。位 0~8 的定义中只有位 7 不能应用于 Input 项目，除此之外其他的位定义都适应于 Input、Output 和 Feature 项目。

表 7 Input、Output 和 Feature 项目的数据项说明

数据字段			含义说明
位	值	名称	
0	0	Data	**数据：表示项目的内容是可更改的（读/写）。**
	1	Constant	**常数：表示项目的内容是不可更改的（只读）。**
1	0	Array	**数组：报表全部控制的状态。如在键盘报表中每一个键在报表中占一位，报表传输全部键的状态，可以同时按下任意多个键。**
	1	Variable	**变量：报表作用中的控制。如在键盘报表中只报表按下的键的编号，可以同时按下的键的数目等于报表的计数（Global 类项目 Report Count）**
2	0	Absolute	**绝对：表示数值以一个固定值为基准。游戏杆通常是报表绝对数据（游戏杆目前的位置）。**
	1	Relative	**相对：表示数据的改变以上一个读数为基准。鼠标通常是报表相对数据（鼠标的移动位置）。**
3 ①	0	No Wrap	**如果设置为 1 表示回转，当数值超过最小值到最大值的范围时将回转，如果最小值是 0 而最大值是 10，超过最大值的下一个数值是 0。**
	1	Wrap	
4 ①	0	Linear	**线形：表示测量的数据与报表的数据有线性的关系。**
	1	Non-Linear	**非线性：表示测量的数据与报表的数据没有线性的关系。**
5 ①	0	Preferred	**优选状态：表示控制在没有用户交互时会回到一个特定的状态。如按钮就有优选状态，在无操作时保持未按下的状态。**
	1	Non-Preferred	**非优选状态：它维持在上一个用户选择的状态。如交替的开关就没有优选状态。**
6 ①	0	No Null Position	**无空状态位置：表示控制永远在传送有效的数据。**
	1	Null State	**空状态：表示控制支持一个没有传送有效数据的状态。如操纵杆可能具有一个多方向的按钮开关，在没有按下时在空状态，这时控制将传送一个在 Logical Minimum 与 Logical Maximum 范围之外的数值来表示它在空状态。**
7 ②	0	Non-Volatile	**不可变的：表示设备只有在主机请求时才改变数值。当主机传送一个报表并且不要改变不可变项目时，如果该项目是定义成相对**

			（Relative）的，数值 0 表示不改变数据，如果不可变项目是定义成绝对（Absolute）的，超出范围外的数值则表示不改变数据。
	1	Volatile	**可变的：表示设备可以自己改变数值，并不是必须主机传送报表要求给设备来改变数值。例如设备控制面板可以由主机软件传送一个报表给设备，也可以由用户自己按设备上的实际按钮。**
8 ①	0	Bit Field	**位字段：表示每一个位或是一个字节内的一组位可以代表一份数据。**
	1	Buffered Bytes	**缓冲字节：表示信息包含一个或多个字节，缓冲字节的报表大小必须是 8。**
9~31 位			**保留**

注：①：该位不能应用到数组。

②：只应用于 Output 和 Feature 项目，对于 Input 项目该位保留。

4.2.2 Collection 和 End Collection 项目

所有的报表类型都可以使用 Collection 与 End Collection 项目来将相关的 Main 类型项目组成群组。这两个项目分别用于打开和关闭集合。所有在 Collection 与 End Collection 项目之间的 Main 类型项目都是 Collection 的一部分。

Collection 有 3 种类型：Application、Physical 与 Logical，其项目的数据项的值分别为 1、0 和 2。厂商也可以自己定义 Collection 类型，数据项的值为 80h~FFh 保留给厂商定义。End Collection 项目无数据项。

Application Collection 包含有共同用途的项目或执行单一功能的项目。例如键盘的开机描述符将键盘的按键与 LED 指示灯数据集合成一个 Application Collection。所有的报表必须在一个 Application Collection 内。

Physical Collection 包含在一个单一几何点上的数据项目，可以将每个位置的数据集合成一个 Physical Collection。在设备报表多个传感器的位置的时候，使用 Physical Collection 指明不同的数据来自不同的传感器。

Logical Collection 形成一个数据结构，包含由 Collection 所连结的不同类型的项目。例如数据缓冲区的内容以及缓冲区内字节数目的计数。

4.2.3 Usage Page 和 Usage 项目

Usage page 项目的数据部分为 1~2 个字节，目前的定义全部都是一个字节。Usage Page 定义了常用的设备功能，关于 Usage Page（以及其他项目）的具体定义内容，可以查阅 HID Usage tables，表 8 是来自 HID Usage tables 的 Usage Page 定义。

表 8 Usage Page 定义

Page ID	Page Name
00	Undefined
01	Generic Desktop Controls
02	Simulation Controls
03	VR Controls
04	Sport Controls
05	Game Controls

06	Generic Device Controls
07	Keyboard/Keypad
08	LEDs
09	Button
0A	Ordinal
0B	Telephony
0C	Consumer
0D	Digitizer
0E	Reserved
0F	PID Page
10	Unicode
11-13	Reserved
14	Alphanumeric Display
15-3f	Reserved
40	Medical Instruments
41-7F	Reserved
80-83	Monitor pages
84-87	Power pages
88-8B	Reserved
8C	Bar Code Scanner page
8D	Scale page
8E	Magnetic Stripe Reading (MSR) Devices
8F	Reserved Point of Sale pages
90	Camera Control Page
91	Arcade Page
92-FEFF	Reserved
FF00-FFFF	Vendor-defined

关于Usage Page的每一个有效定义项，都有一个相应的下一级定义，如Usage Page的数据项数值为1，则设备定义为Generic Desktop Controls，关于该类设备的具体功能可以在HID Usage Tables中查到具体的定义。表9是HID Usage Tables中对Generic Desktop Controls设备的功能定义。

表 9 Generic Desktop Controls 用法定义

Usage ID	Usage Name	Usage Type	参阅 HID Usage Tables 中的相关章节
00	Undefined		
01	Pointer	CP	4.1
02	Mouse	CA	
03	Reserved		
04	Joystick	CA	4.1
05	Game Pad	CA	
06	Keyboard	CA	
07	Keypad	CA	
08	Multi-axis Controller	CA	
09	Tablet PC System Controls	CA	

0A-2F	Reserved		
30	X	DV	4.2
31	Y	DV	
32	Z	DV	
33	Rx	DV	
34	Ry	DV	
35	Rz	DV	
36	Slider	DV	4.3
37	Dial	DV	
38	Wheel	DV	
39	Hat switch	DV	
3A	Counted Buffer	CL	4.6
3B	Byte Count	DV	
3C	Motion Wakeup	OSC	4.3
3D	Start	OOC	
3E	Select	OOC	
3F	Reserved		
40	Vx	DV	4.3.1
41	Vy	DV	
42	Vz	DV	
43	Vbrx	DV	
44	Vbry	DV	
45	Vbrz	DV	
46	Vno	DV	
47	Feature Notification	DV,DF	4.8
48	Resolution Multiplier	DV	
49-7F	Reserved		
80	System Control	CA	4.5
81	System Power Down	OSC	
82	System Sleep	OSC	4.5.1
83	System Wake Up	OSC	
84	System Context Menu	OSC	4.5
85	System Main Menu	OSC	
86	System App Menu	OSC	
87	System Menu Help	OSC	
88	System Menu Exit	OSC	
89	System Menu Select	OSC	
8A	System Menu Right	RTC	
8B	System Menu Left	RTC	
8C	System Menu Up	RTC	
8D	System Menu Down	RTC	
8E	System Cold Restart	OSC	4.5.1
8F	System Warm Restart OSC		
90	D-pad Up	OOC	4.7
91	D-pad Down	OOC	
92	D-pad Right	OOC	
93	D-pad Left	OOC	
94-9F	Reserved		
A0	System Dock	OSC	4.5.1

A1	System Undock	OSC	
A2	System Setup	OSC	
A3	System Break	OSC	4.9
A4	System Debugger Break	OSC	
A5	Application Break	OSC	
A6	Application Debugger Break	OSC	
A7	System Speaker Mute	OSC	4.5.1
A8	System Hibernate	OSC	
A9-AF	Reserved		
B0	System Display Invert	OSC	4.10
B1	System Display Internal	OSC	
B2	System Display External	OSC	
B3	System Display Both	OSC	
B4	System Display Dual	OSC	
B5	System Display Toggle Int/Ext	OSC	
B6	System Display Swap Primary/Secondary	OSC	
B7	System Display LCD Autoscale	OSC	
B8-FFFF	**Reserved**		

用法（Usage）定义了各种各样设备特性，对于Usage Page的每一项都定义了常用的各种用法。

用法说明了3种信息，即控制、集合和数据。控制说明设备的状态，如on/off、Enable/Disable等。集合说明控制和数据的组合关系。

上表中的用法类型（Usage Type）描述了应用程序如何处理由Main类型项目生成的数据，具体的定义和详细说明请参阅HID Usage Tables。

4.2.4 Report ID 项目

Report ID放在信息包中报表数据之前，设备可以支持多个相同类型的报表，每一个报表包含不同的数据与其特有的ID。

在报表描述符中，Report ID项目作用于其后续所有的项目，直到遇到下一个Report ID为止。如果报表描述符中没有Report ID项目，默认的ID值是0，描述符不能定义一个为0的Report ID，输入报表、输出报表与特征报表可以分享同一个Report ID。

在Set_Report和Get_Report请求传输中，主机在设置事务的 wValue字段的低字节中指定一个Report ID。在中断传输中如果接口支持一个以上的 Report ID，Report ID必须是传送报表中的第一个字节。如果接口只支持数值为0的默认Report ID，此Report ID不应该在中断传输中随着报表一起传送。

4.2.5 Logical Minimum 和 Logical Maximum 项目

Logical Minimum与 Logical Maximum项目定义报表的变量（Variable）或阵列（Array）数据的限制范围，此限制范围以逻辑单位来表示。例如设备报表的一个电流值读数是500mA，而一个单位是2mA，则 Logical Maximum值等于250。

负数值以2的补码来表示。如果Logical Minimum与Logical Maximum都是正数，就不需要有正负号位。不管 Logical Minimum与Logical Maximum是以有正负号或是无正负号的数值来表示，设备都可以正确地传输数据。数据的接收者必须知道数据是否可以是负值。

4.2.6 Physical Minimum 和 Physical Maximum 项目

Physical Minimum和Physical Maximum项目定义数值的限制范围，该限制范围使用Unit项目定义的单位来表示。上例中设备报表的一个电流值读数是500mA，单位是2mA， Logical Maximum值等于250，而Physical Maximum值是500。

Logical Minimum与 Logical Maximum值说明了设备返回数值的边界，可以根据Physical Minimum和Physical Maximum值对数据进行偏移和比例变换。

4.2.7 Unit Exponent 项目

Unit Exponent项目定义了在使用逻辑范围和实际范围将设备的返回数值转换成实际数值时，使用10的多少次方对数值进行定标。Unit Exponent的值的编码为4位补码，代表其数范围是-8~+7。

表 10 Unit Exponent 数值表

代码	00h	01h	02h	03h	04h	05h	06h	07h	08h	09h	0Ah	0Bh	0Ch	0Dh	0Eh	0FH
数值	0	1	2	3	4	5	6	7	-8	-7	-6	-5	-4	-3	-2	-1

根据以上 5 个项目的值可以换算出报表传送数据（逻辑数据）与物理数据的转换关系。

物理数据值 = 逻辑数据值÷分辨率

分辨率 = （LogicalMaximum - LogicalMinimum）÷

（（PhysicalMaximum - PhysicalMinimum）×$10^{UnitExponent}$）

4.2.8 Unit 项目

Unit项目指定报表数据在使用Physical与Unit Exponent项目转换后使用什么度量单位，以及单位的幂指数值。Unit的数值部分可以长达4字节，按照4位为一段分段，可以分为8个半字节段，由高到低分别为半字节7、半字节6、…、半字节0。每一个半字节对应不同的基本单位，其数值表示单位的指数值，采用4位2的补码表示，取值范围是-8~+7之间。

从半字节0~6由下表给出了具体的定义，其中半字节0表示测量系统，半字节7保留。例如在半字节0数值为1（表示采用线性公制测量系统）的条件下，半字节1表示长度（单位为厘米），如果其数值为1表示厘米，数值为2表示（厘米）^{2}，成为面积单位。半字节3表示时间（单位为秒），如果其数值为-2，表示（秒）$^{-2}$。

表 11 Unit 单位的定义

半字节序号	测量项目	数值含义				
		0	1	2	3	4
0	测量系	无	线性、公	角度、公制	线性、英制	角度、英制

	统		制			
1	长度		厘米	半径	英寸	度
2	质量		克		石拉（slug）	
3	时间		秒			
4	温度		开式度（Kelvin）		华式度	
5	电流		安培			
6	亮度		烛光			
7	保留					

虽然表中只是定义了有限的基本单位，但可以通过这些基本单位的组合派生出大多数其它的常用单位。

例如报表使用一个字节传送一个从-20到110华氏度温度值，可以定义以下报表描述项目：

Logical Minimum = -128
Logical Maximum = 127
Physical Minimum = -20
Physical Maximum = 110
Unit Exponent = 0
Unit = 30003h

Unit的半字节0=3选择英制线性测量系统，半字节4=3选择华氏温度单位。

130（110+20）华氏度的数值范围线性分布到了256和有效数值区域，每一位相当于0.51华氏度，这样就提高了分辨率。

4.2.9 Report Size 和 Report Count 项目

Report Size项目指定Input、Output与Feature项目字段的大小，以位为单位。

Report Count项目指定Input、Output与Feature项目包含的字段数目。

例如两个8位的字段，Report Size等于8，而Report Count等于2。8个1位的字段，Report Size等于1，而 Report Count等于8。

Input、Output与Feature项目报表可以有多个项目，每一个项目可以有自己的Report Size和Report Count项目。

4.2.10 Push 和 Pop 项目

Push项目将一个Global项目状态表格的副本压入CPU的堆栈内。Global项目状态表格包含所有之前定义的Global项目的目前设置。

Pop项目恢复之前压入堆栈的Global项目状态的储存状态。

4.2.11 Usage、Usage Minimum 和 Usage Maximum 项目

Usage项目和Global类型的Usage Page项目协同描述项目或集合的功能。

一个报表可以指定一个Usage给许多个控制，或是指定不同的Usage给每一个控制。如果一个报表项目之前有一个Usage，此Usage应用到该项目的所有控制。如果一个报表项目之前有一个以上的Usage，每一个Usage应用到一个控制，Usage与控制是按顺序结合的。

报表也可指定一个Usage给多个控制，或者它也可以给每个控制指定不同的Usage。如果一个报表行的开始是一个单一用法，那么此Usage适用于行的所有控制。如果一个报表行的开始处有一个以上的Usage，并且控制的数量等于Usage的数量，那么一个Usage适用于一个控

制，Usage和控制按顺序结成对。在下面的例子里，报表包含两个字节，第一个字节的用法是X，第二个字节的用法是Y。

```
Report Size (8)
Report Count (2)
Usage (x)
Usage (y)
Input (Data, Variable, Absolute)
```

如果一个报表行的开始处有多个用法，且控制的数量大于 Usage 的数量，则每个 Usage 和一个控制结对，最后一个 Usage 适用于剩下的所有控制。在下面的例子里，报表是 16 个字节的。用法 X 适用于第 1 个字节，用法 Y 适用于低 2 字节，而一个供应商定义的 Usage 适用于第 3 到第 16 个字节。

```
Usage (x)
Usage (y)
Usage (Vendor defined)
Report Size (8)
Report Count (16)
Input (Data, Variable, Absolute)
```

Usage Minimum 和 Usage Maximum 可以指定一个 Usage 给多个控制或是数组项目。将从 Usage Minimum 到 Usage Maximum 定义的用法顺序对应到多个控制中。

例如在一个键盘描述符中定义的标准键盘的左、右修饰键的输入项目中，使用一个字节的 8 位分别输入键盘的左、右 Ctrl 键、Shift 键、Alt 键和 GUI 键，从 HID Usage tables 文档中可以查到关于键盘用法的定义，其中上述 8 个修饰键的用法定义值为 224 到 231。以下是报表描述符的修饰键部分描述。

```
Usage Page (1)                          ; 1 = Generic Desktop Controls
Usage (6)                               ; 6 = Keyboard
Collection (1)                          ; 1 = Application
Usage Page (7)                          ; 7 = Keyboard/Keypad
Usage Minimum (224)
Usage Maximum (231)
Logical Minimum (0)
Logical Maximum (1)
Report Size (1)
Report Count (8)
Input (Data, Variable, Absolute)

……
```

5. USB 接口的鼠标描述符

下面介绍在 KL25 的 HID 例程中，一个 USB 接口的鼠标的全部描述符。该鼠标固件部分由 MCU 实现全部控制功能，下面列出的代码为 MCU 汇编实现描述符定义。

5.1 设备描述符

```
;=========================================
; Device descriptor 设备描述符
;=========================================
DEVICE_DESC_DATA:
DB 0x12               ; bLength = 18，该描述符长度为18字节
DB 0x01               ; bDescriptorType = 01，表明是设备描述符
DB 0x00, 0x02        ; bcdUSB，      USB规范为2.0,BDC码
DB 0x00              ; DeviceClass，设备类码，HID设备为0，类别在接口描述符中定义
DB 0X00              ; DeviceSubClass，设备子类码，DeviceClass为0时该字段必须为0
DB 0X00              ; bDevicePortocol，协议码，DeviceClass为0时该字段必须为0
DB 0x10              ; bMaxPacketSize0，端点0的最大包尺寸
DB 0XA2, 0x15        ; bVendor，厂商ID，由USB实现者论坛确定的
DB 0x00, 0x01        ; bProduct，产品ID
DB 0x02, 0x00        ; bcdDevice，设备版本号，BCD码
DB 0x01              ; iManufacturer，厂商字符串的索引值，见字符串描述符
DB 0x02              ; iProduct，产品字符串的索引值，见字符串描述符
DB 0x00              ; iSerialNumber，产品序列号字符串的索引值，见字符串描述符
DB 0X01              ; bNumConfigurations，配置数目只有1个
```

5.2 配置描述符

```
;=========================================
; Configuration descriptor    配置描述符
;=========================================
CONFIG_DESC_DATA:
DB 0x09       ; bLength=9，该描述符长度为9字节
DB 0x02       ; bDescriptorType = 02，表明是配置描述符
DB 0x22, 0x00 ; wTotalLength = 34，配置、接口、端点和HID描述符的总和字节数
DB 0x01       ; bNumInterfaces = 1，本配置支持的接口数目为2个
DB 0x01       ; bConfigurationValue = 1，本配置描述符的标识符
DB 0x00       ; iCongfiguration = 0，配置描述符说明字符串的索引值，0表示无
DB 0XE0       ; bmAttributes，电源及唤醒设置，非自供电，支持远程唤醒
DB 0X32       ; MaxPower = 50，本设备最大耗电为50X2mA=100mA
```

5.3 接口描述符

```
;=========================================
; Interface descriptor    接口描述符
;=========================================
InterfaceDescriptor0:
DB 0x09              ; bLength = 9，该描述符长度
DB 0x04              ; bDescriptorType = 4，表明是接口描述符
DB 0x00              ; bInterfaceNumber = 0，此接口的识别标识符
DB 0x00              ; bAlternateSetting = 0，表示此接口无替代设置值
DB 0x01              ; bNumEndpoints = 1，本接口的端点数目，HID设备使用端点1
DB 0x03              ; bInterfaceClass = 3，表示该设备是HID类别
DB 0x01              ; bInterfaceSubClass = 1，表示支持启动接口
DB 0x02              ; bInterfaceProtocol = 2，表示支持鼠标协议
DB 0x00              ; iInterface = 0，接口描述符说明字符串的索引值，0表示无字符串
```

5.4 字符串描述符

```
;=========================================
```

```
; 语言ID的定义
;=========================================
DB 0x04                     ; bLength = 4，字符串描述符0的长度为4
DB 0x03                     ; bDescriptorType = 3，表明是字符串描述符
DB 0x09, 0x04               ; wLANGID = 0409h，表明是美式英语
```

```
;=========================================
; 产品的字符串描述符
;=========================================
DB 0x1E                     ; bLength = 0x1E，字符串描述符0的长度为30
DB 0x03                     ; bDescriptorType = 3，表明是字符串描述符
DB 0x20,0x00,0x20,0x00,0x4D,0x00,0x4B,0x00,0x20,0x00,0x48,0x00,0x49,
   0x00,0x44,0x00,0x20,0x00,0x44,0x00,0x45,0x00,0x4D,0x00,0x4F,0x00,
   0x20,0x00                ; bString=”  MK  HID DEMO”,Unicode编码的字符串
```

5.4 HID 描述符

```
;=========================================
; HID descriptor    HID描述符
;=========================================
HIDDescriptor0:
DB 0x09             ; bLength = 9，该描述符长度
DB 0x21             ; bDescriptorType = 21h，表明是HID描述符
DB 0x00, 0x01       ; bcdHID = 0100，HID规范版本为1.00
DB 0x00             ; bCountryCode = 0，硬件设备所在国家的国家代码，0表示未指明
DB 0x01             ; nNumDescriptors = 1，表示支持的描述符有1个，即一个报表描述符
DB 0x22             ; bDescriptorType = 22h，描述符类别，表示支持的描述符是报表描述符
DB 0x34, 0x00       ; wDescriptorLength = 52，表示支持的报表描述符的长度
```

5.5 端点描述符

```
;=========================================
; EndPoint descriptor    端点描述符
;=========================================
EndpointDescriptor0:
DB 0x07                     ; bLength = 7，该描述符长度
DB 0x05                     ; bDescriptorType = 5，表明是端点描述符
DB 0x81                     ; bEndpointAddress = 1000 0001b，表示1号输入端点
DB 0X03                     ; bmAttributes = 00000011b，表示中断类型端点
DB 0x08, 0x00               ; wMaxPacketSize = 8，端点发送和接收的最大包尺寸为8
DB 0x0A                     ; bInterval = 10，表示中断端点轮询时间间隔为10ms
```

5.6 报表描述表

```
;=========================================
;HID Reports Descriptor  报表描述符
;=========================================
DB 0x05, 1            ; Usage Page (1: Generic Desktop)
DB 0x09, 2            ; Usage (2: Mouse) 表示报表定义的是HID鼠标
 DB 0xA1, 1           ; Collection (1: Application) ====================集合开始
  DB 0x09, 1          ; Usage (1: Pointer)
  DB 0Xa1, 0          ; Collection (0: Physical)
    DB 0x05, 9        ; USAGE_PAGE (Button)
    DB 0x19, 1        ; Usage Minimum (1)
    DB 0x29, 3        ; Usage Maximum (3)
    DB 0x15, 0        ; Logical Minimum (0)
    DB 0x25, 1        ; Logical Maximum (1)
    DB 0x95, 3        ; Report Count (3)
    DB 0x75, 1        ; Report Size (1)
    DB 0x81, 2        ; Input (Data, Variable, Absolute)
                        ;
    DB 0x95, 1        ; Report Count (1)
    DB 0x75, 5        ; Report Size (5),
    DB 0x81, 1         ; Input (Constant)
                         ;
    DB 0x05, 1         ; USAGE_PAGE (Generic Desktop)
    DB 0x09, 30        ; USAGE (X)
    DB 0x09, 31        ; USAGE (Y)
    DB 0x09, 38        ; USAGE (Wheel)
    DB 0x15, 81        ; LOGICAL_MINIMUM (-127)
    DB 0x25, 7f        ; LOGICAL_MAXIMUM (127)
    DB 0x75, 8         ; REPORT_SIZE (8)
    DB 0x95, 3         ; REPORT_COUNT (3)
    DB 0x81, 6         ; INPUT (Data, Variable, Relative)

  DB 0xC0
DB 0xC0               ; End_Collection ================================= 集合结束
```

6. USB 接口的鼠标枚举过程介绍

6.1 USB 设备枚举过程分析

USB 主机在检测到 USB 设备插入后，就要对设备进行枚举了。为什么要枚举呢？枚举就是从设备读取各种描述符信息，这样主机就可以根据这些信息来加载合适的驱动程序，从而知道设备是什么样的设备，如何进行通信等。

在介绍 USB 接口的鼠标枚举过程之前，先请大家查阅资料，了解 USB 基本通信：包的结构、分类和明白 USB 通信中的事务处理概念。同时还需掌握 USB 的控制传输模式，该传输模式在 USB 中非常重要的，它要保证数据的正确性，在设备的枚举过程中都是使用控制传输。控制传输分为三个过程：建立过程、可选的数据过程及状态过程。

枚举过程分析如下：

1. USB 主机检测到 USB 设备插入后，就会先对设备复位。USB 设备在总线复位后其地址为 0，这样主机就会发起第一个控制传输（获取设备描述）与刚刚输入的 USB 设备通信：

- 主机 SETUP 包（发往地址 0 端点 0）、主机数据包（请求设备描述符）、设备握手包 ACK。

- 数据过程，主机先发一个 IN 令牌包、设备发一个数据包（这个数据已经准备好，SIE 收到 IN 令牌后，直接送到总线上，用户此时不干预）、主机发 ACK 包。
- 状态过程：主机发 OUT 包（通知设备要输出）、主机发 0 字节状态数据包（这个是 0 字节，表明自己收到设备描述符）、设备发握手 ACK 包。

2. 主机对设备又一次复位。这时就进入到设置地址阶段。

- 主机 SETUP 包（发往地址 0 端点 0）、主机数据包（请求设置地址）、设备握手包 ACK。所以 SETUP 包后面都会跟一个表明主机 SETUP 目的的数据包，要么 GET，要么 SET。
- 数据过程，本次传输没有数据。
- 状态过程：主机发 IN 包（通知设备要返回数据）、设备发 0 字节状态数据包（表明地址设置已经成功）、主机发握手 ACK 包（地址设置已经生效）。

3. 主机使用新地址获取完整的设备描述符。

   主机采用新地址发起第一个控制传输：

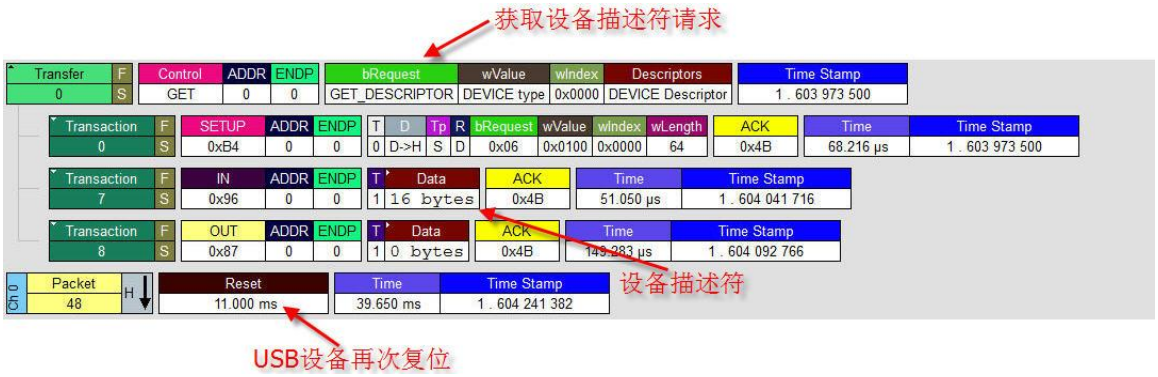
- 主机 SETUP 包（发往新的地址端点 0）、主机数据包（请求设备描述符）、设备握手包 ACK。
- 数据过程，主机先发一个 IN 令牌包、设备发一个数据包（这个数据已经准备好，SIE 收到 IN 令牌后，直接送到总线上，用户此时不干预）、主机发 ACK 包。
- 状态过程：主机发 OUT 包（通知设备要输出）、主机发 0 字节状态数据包（表明自己收到设备描述符）、设备发握手 ACK 包。

4. 接着逐次获取配置描述符、接口描述符、类特殊描述符（如果有）、端点描述符等。接口描述符、类特殊描述符、端点描述符是不能单独获取的，必须跟随配置描述符以一个集合的方式一并返回。

   注意：USB 设备如有字符串描述符，还要获取字符串描述符。另外，像 HID 设备还有报告描述等，它们是单独获取的。

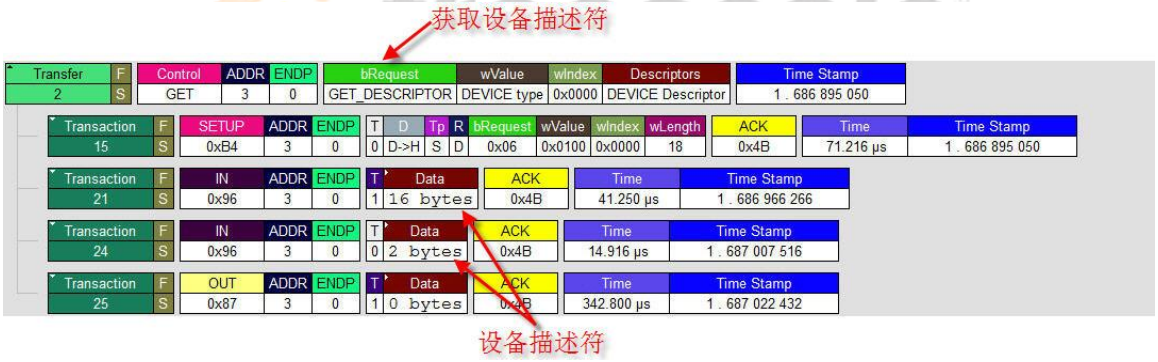
6.2 枚举过程数据包分析

- 阶段 1—获取设备描述符，接着再次复位 USBS 设备

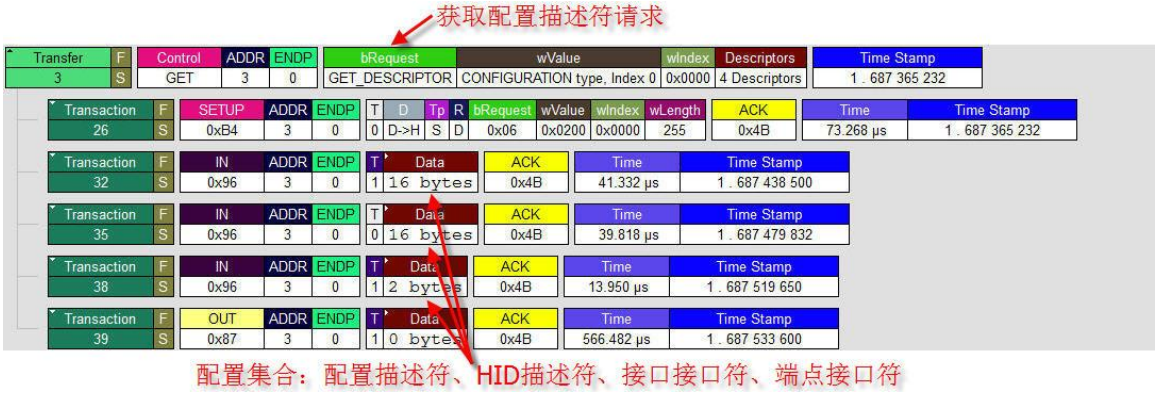


- 阶段 2—发送 SetAddress 请求

- 阶段 3— 重新获取设备描述符

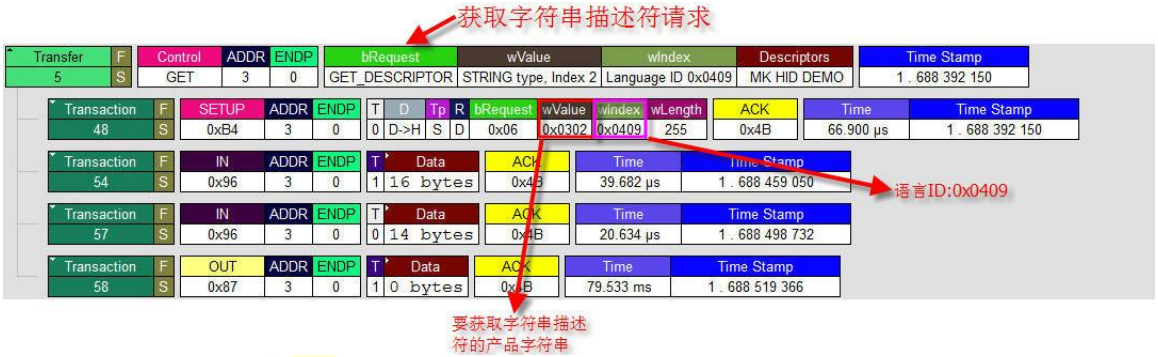


- 阶段 4—获取 USB 设备配置集合

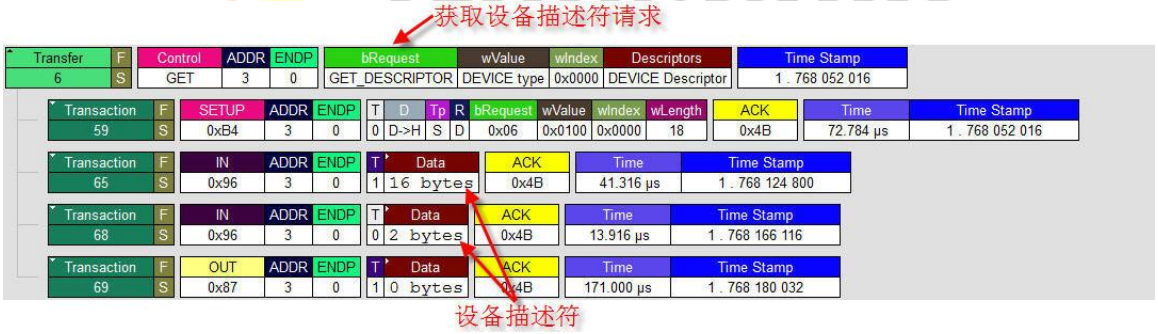


- 阶段 5—获取字符描述符的语言 ID

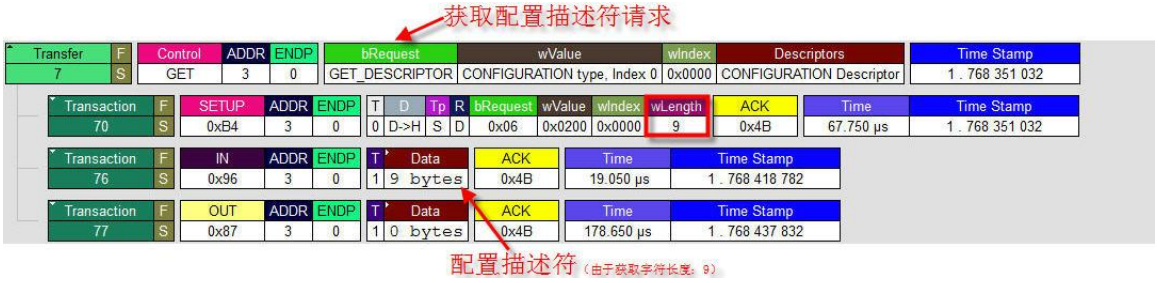
- 阶段 6—获取字符描述符的产品字符串



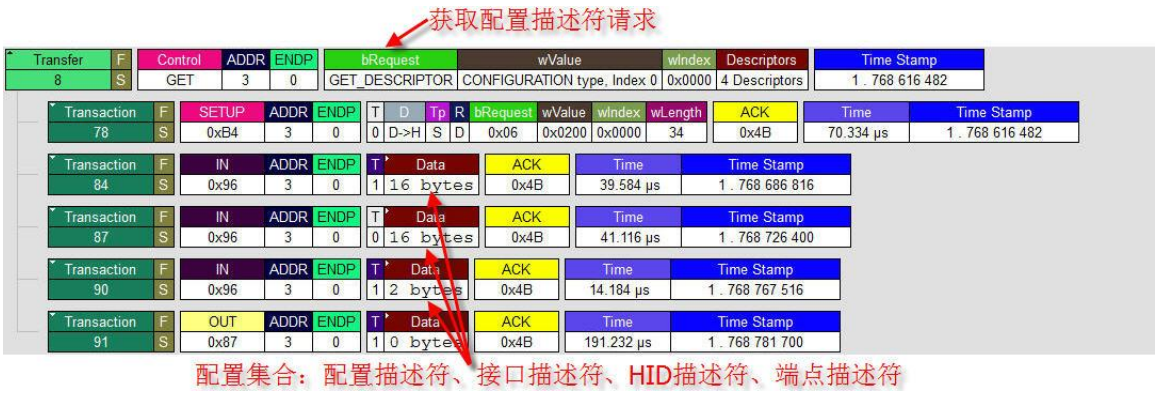
- 阶段 7—第三次获取设备描述符



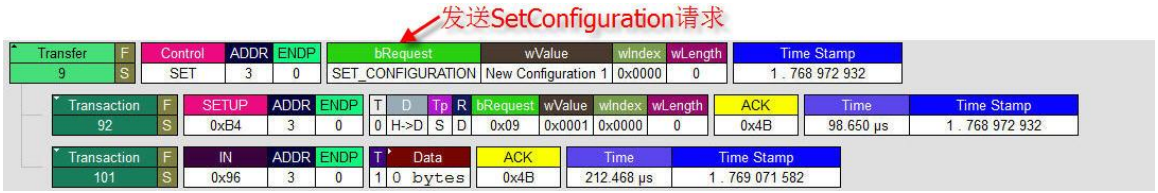
- 阶段 8—获取配置描述符



- 阶段 9—再次获取 USB 设备配置集合



- 阶段 10—发送 SetConfiguration 请求



- 阶段 11—发送 Set_Idle 请求

- 阶段 12—获取 HID 报告描述符