

15.3.1 HID 类设备简介

1. 设备范围

HID(human interface device)类主要由人类用来控制计算机系统操作的设备组成。HID 类设备的典型示例包括:

- (1) 键盘和指针设备:标准鼠标设备、跟踪球和操纵杆。
- (2) 面板控制:旋钮、开关、按钮和滑块。
- (3) 电话、VCR 遥控器、游戏或模拟设备等设备上的控件:数据手套、油门、方向盘和方向舵踏板。
- (4) 某些 HID 类设备可能不需要人工交互,但可以提供 HID 类似的格式数据的设备:条形码阅读器、温度计或电压表。

许多典型的 HID 类设备包括指示器、专用显示器、音频反馈以及力或触觉反馈。因此, HID 类定义包括对针对最终用户的各种类型的输出的支持。

2. HID 规范的意义

HID 规范的主要意义是:

- (1) 尽量节省设备的数据空间。
- (2) 允许软件应用程序跳过未知信息。
- (3) 可扩展且鲁棒。
- (4) 支持嵌套和集合。
- (5) 自我描述以允许通用软件应用程序。

15.3.2 主机请求

1. 标准请求

HID 类使用 USB 标准请求 GetDescriptor,如 USB 规范中所述。当发出 GetDescriptor(configuration)请求时,它返回每个接口的配置描述符、所有接口描述符、所有端点描述符和 HID 描述符。它不应返回字符串描述符、HID 报告描述符或其他可选的 HID 类描述符。HID 描述符应处于 HID 接口的接口和端点描述符之间。顺序应为:

- (1) 配置描述符
- (2) 接口描述符(指定 HID 类)
- (3) HID 描述符(与上述接口关联)
- (4) 端点描述符(用于 HID 中断端点)
- (5) 可选端点描述符(用于 HID 中断输出端点)

注: GetDescriptor 可用于检索标准、类别和供应商具体描述符,取决于描述符类型字段的设置。

表 15.43 定义了 Descriptor Type(Get_Descriptor 请求中 wValue 的高字节)。

表 15.43 Descriptor Type 说明

部分	说明
Descriptor Type	<p>指定 Descriptor Type 的特征位</p> <p>7 保留(应始终 0)</p> <p>6..5 类型</p> <p>0=标准</p> <p>1=类</p> <p>2=供应商</p> <p>3=保留</p> <p>4..0 描述符</p> <p>请参阅标准类或供应商描述符类型表。</p>

表 15.44 定义了类描述符的有效类型。

表 15.44 类描述符的有效类型

值	类描述符类型
0x21	HID
0x22	报告
0x23	物理描述符
0x24~0x2F	保留

(1) GetDescriptor 请求

GetDescriptor 请求返回设备的描述符,如表 15.45 所示。

表 15.45 GetDescriptor 数据结构的构成

字段	USB 标准描述符	HID 类描述符
bmRequestType	100 xxxxx	10000001
bRequest	GET_DESCRIPTOR(0x06)	GET_DESCRIPTOR(0x06)
wValue	描述符类型一节描述符索引	描述符类型一节描述符索引
wIndex	0 或者语言 ID	接口号
wLength	描述符长度	描述符长度
Data	描述符	描述符

注意:

- 于标准 USB 描述符,bmRequestType 的位 0~4 表示请求的描述符是否与设备、接口、端点或其他相关联。
- wValue 字段中高字节指定描述符类型,低字节指定描述符索引。
- 低字节是用于指定物理描述符集的描述符索引,对于其他 HID 类描述符,低字节将重置为零。

1) 如果正在请求 HID 类描述符,则 wIndex 字段指示 HID 接口的编号。如果请求标准描述符,则 wIndex 字段指定字符串描述符的语言 ID,并为其他标准描述符重置为零。

2) 请求物理描述符集 0 返回一个特殊的描述符,标识描述符集的数量及其大小。

3) 物理索引等于 1 的 GetDescriptor 请求将请求第一个物理描述符集。设备可能对其项目有其他用途。可以通过在递增描述符索引的同时发出后续 GetDescriptor 请求来枚举这些请求。设备将向索引大于 HID 描述符中定义的最后一个数字的请求返回最后一个描述符集

(2) SetDescriptor 请求

Set_Descriptor 请求允许主机更改设备中的描述符。对该请求的支持是可选的,如表 15.46 所示。

表 15.46 Set_Descriptor 数据结构的构成

字段	USB 标准描述符	HID 类描述符
bmRequestType	100 XXXXX	10000001
bRequest	SET_DESCRIPTOR(0x07)	SET_DESCRIPTOR(0x07)
wValue	描述符类型一节描述符索引	描述符类型一节描述符索引
wIndex	0 或者语言 ID	接口号
wLength	描述符长度	描述符长度
Data	描述符	描述符

2. 类特定请求

此部分请求特定于 HID 类设备,请求允许主机查询设备的功能和状态,并设置输出和功能项的状态。这些事务通过控制管道完成,因此遵循 USB 规范中定义的控制管道中标准请求格式。表 15.47 为特定类请求数据结构的一般格式。

表 15.47 特定类请求数据结构

字段	偏移(字节)	说明
bmRequestType	0/1	指定请求特征的位。有效值为 10100001 或 00100001 仅基于以下描述: 7 数据传输方向 0=从主机到设备 1=从设备到主机 6..5 类型 1=类 4..0 接收对象 1=接口
bRequest	1/1	详细的请求
wValue	2/2	指定字长字段的数值表达式(根据请求变化而变化)
wIndex	4/2	指定字长字段的索引或偏移量(根据请求变化而变化)
wLength	6/2	指定在数据阶段传输的字节数

表 15.48 定义了 bRequest 的有效值。

表 15.48 bRequest 有效值表

请求	bRequest 值	功能
GET_REPORT	0x01	请求允许主机通过控制管道接收报告
SET_REPORT	0x09	请求允许主机向设备发送报告,以设置输入、输出或功能控件的状态
GET_IDLE	0x02	请求读取特定输入报告的当前空闲率
SET_IDLE	0x0A	请求屏蔽中断输入管道上的特定报告,直到新事件发生或经过指定的时间间隔
GET_PROTOCOL	0x03	请求读取当前已激活的协议(例如引导协议或报告协议)
SET_PROTOCOL	0x0B	切换引导协议或报告协议

注意:以上请求所有设备需要强制支持,以上请求仅启动引导类设备需要。

1) Get_Report 请求

Get_Report 请求允许主机通过控制管道接收报告。

表 15.49 Get_Report 请求内容

bmRequestType	bRequest	wValue	wIndex	wLength	数据过程
0xA1	GET_REPORT	报告类型和报告 ID	接口号	报告大小	报告

wValue 字段在高字节中指定报告类型,在低字节中指定报告 ID。如果不使用报告 ID,需要将报告 ID 设置为 0(零)。报告类型指定如见表 15.50。

表 15.50 wValue 字段值表

值	报告类型
01	输入
02	输出
03	特征
04~FF	保留

此请求在初始化绝对项和确定特征项的状态时很有用。此请求不用于定期轮询设备状态。中断 IN 管道应该用于重复的输入报告。输入报告回复与来自中断管道的报告具有相同的格式。

可以选择将中断输出管道用于低延迟输出报告。如果未声明中断输出端点,则通过中断输出管道的输出报告具有与通过控制管道发送的输出报告相同的格式。

2) Set_Report 请求

Set_Report 请求允许主机向设备发送报告,以设置输入、输出或控件功能,如表 15.51 所示。

表 15.51 Set_Report 请求内容

bmRequestType	bRequest	wValue	wIndex	wLength	数据
0x21	SET_REPORT	报告类型和报告 ID	接口号	报告大小	报告

Set_Report 请求的请求字段的含义与 Get_Report 请求相同,只是数据方向相反,报告数据是从主机发送到设备的。

设备可能会选择忽略输入的 Set_Report 请求,因为它没有意义。或者,这些报告可用于重置控件的原点(即,当前位置应报告为零)。发送报告的效果还取决于收件者控件是绝对的还是相对的。

3) Get_Idle 请求

Get_Idle 请求读取特定输入报告的当前空闲率,如表 15.52 所示。

表 15.52 Get_Idle 请求内容

bmRequestType	bRequest	wValue	wIndex	wLength	数据
0x81	GET_IDLE	0 和报告 ID	接口号	1	空闲速率

4) Set_Idle 请求

Set_Idle 请求使中断输入管道上的特定报告被屏蔽,直到发生新事件或经过指定的时间量,如表 15.53 所示。

表 15.53 Set_Idle 请求内容

bmRequestType	bRequest	wValue	wIndex	wLength	数据
0x21	SET_IDLE	持续和报告 ID	接口号	0	无

备注:该请求用于限制端点中断的报告频率。具体来说,该请求会导致端点对端点中的中断进行任何轮询,而其当前报告保持不变。在没有更改的情况下,轮询将在给定的基于时间的持续时间内继续进行 NAK。该请求包含以下部分,如表 15.54 所示。

表 15.54 Set_Idle 请求特性

部分	说明
持续	<p>当 wValue 的高字节为 0(零)时,持续时间是未知的。端点将永远禁止报告,仅在报告数据中检测到更改时才报告。</p> <p>当 wValue 的高字节不为零时,则使用固定的持续时间。持续时间将与高字节的值线性相关,LSB 的权重为 4 毫秒。这提供了从 0.004 到 1.020 秒的值范围,分辨率为 4 毫秒。如果持续时间小于设备轮询率,则以轮询率生成报告。</p> <p>如果给定的持续时间过去了,报告数据没有变化,那么端点将生成单个报告,并且报告抑制将使用先前的持续时间重新开始。</p>
报告 ID	<p>如果 wValue 的低字节为零,则空闲率适用于设备生成的所有输入报告。当 wValue 的低字节为非零时,则空闲率仅适用于低字节的值指定的 Report ID。</p>
精度	此持续时间应具有 $\pm(10\%+2 \text{ 毫秒})$ 的精度

部分	说明
延迟	<p>如果在当前执行周期结束前至少 4 毫秒收到新请求,则新请求将被执行,就好像它是在上次报告之后立即发出的一样。如果在当前周期结束后的 4 毫秒内收到新请求,则新请求将在报告之后才生效。</p> <p>如果当前期间已超过新规定的持续时间,则将立即生成报告。</p>

如果端点的中断为多个报告提供服务,Set_Idle 请求可用于仅影响为指定报告 ID 生成重复报告的速率。例如,具有两个输入报告的设备可以为报告 1 指定 20ms 的空闲率,为报告 2 指定 500 ms 的空闲率。

推荐的默认空闲速率(设备初始化时的速率)对于键盘是 500 ms(在第一次重复速率之前的延迟),对于操纵杆和鼠标是无穷大。

5) Get_Protocol 请求

Get_Protocol 请求读取当前处于活动状态的协议(引导协议或报告协议),如表 15.55 所示。

表 15.55 Get_Protocol 请求内容

bmRequestType	bRequest	wValue	wIndex	wLength	数据
0x81	GET_PROTOCOL	0	接口号	1	0=boot 协议 1=报告协议

Boot 子类中的设备支持此请求。wValue 字段指示应该使用哪个协议。

6) Set_Protocol 请求

Set_Protocol 在引导协议和报告协议之间切换(反之亦然),如表 15.56 所示。

表 15.56 Set_Protocol 请求内容

bmRequestType	bRequest	wValue	wIndex	wLength	数据
0x21	SET_PROTOCOL	0=Boot 协议 1=报告协议	接口号	0	无

启动子类中的设备支持此请求。wValue 字段指示应该使用哪个协议。

初始化时,所有设备默认报告协议。然而,主机不应该对设备的状态做出假设,并且应该在初始化设备时设置所需的协议。

15.3.3 描述符

有关 USB 设备的信息存储在其 ROM 的段中。这些段称为描述符。描述符分为 HID 标准描述符和类特定描述符,而 HID 类设备的描述符结构可由图 15.29 描述。

1. 标准描述符

HID 类设备类使用以下标准 USB 描述符。

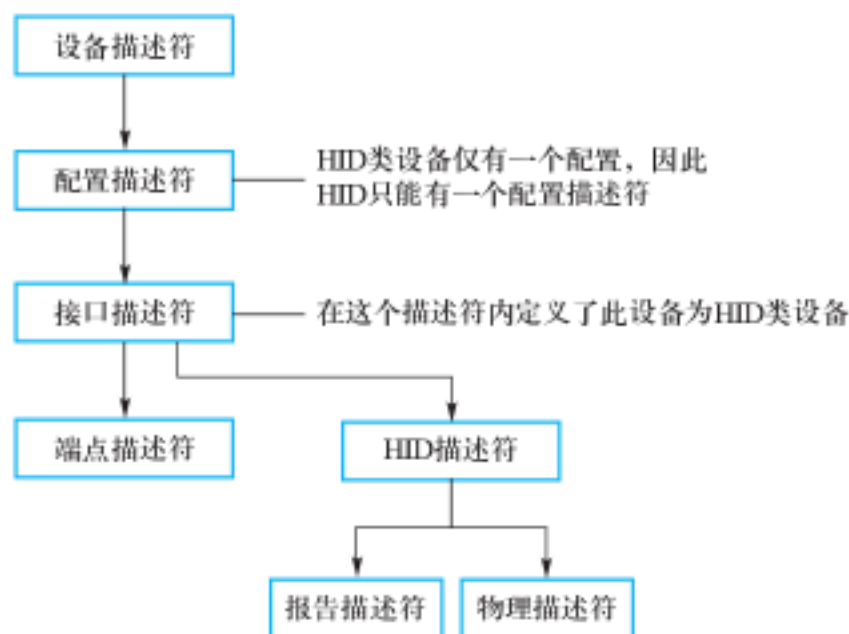


图 15.29 HID 类设备的描述符结构

1) 设备描述符

注意:设备描述符中的 bDeviceClass 和 bDeviceSubClass 字段不应用于将设备标识为属于 HID 类。标识设备是 HID 类是使用接口描述符中的 bInterfaceClass 和 bInterfaceSubClass 字段。

2) 配置描述符

3) 接口描述符

- bInterfaceClass 字段始终为 3。
- bInterfaceSubClass 字段若支持引导协议为 1,否则为 0
- bInterfaceProtocol 字段有在 bInterfaceSubClass 字段声明设备支持启动接口时才有意义,否则为 0。bInterfaceProtocol 字段值若为 0,则表示无设备,bInterfaceProtocol 字段值若为 2 表示该设备为鼠标设备,bInterfaceProtocol 字段值若为 1 表示该设备为键盘设备。其余值作保留

4) 端点描述符

5) 字符串描述符

有关标准描述符数据结构其他内容请参照 USB 标准描述符一节

2. 类特定描述符

每个设备类包含一类或多类特定的描述符。这些描述符区别于标准 USB 描述符。HID 类设备使用以下特定类描述符:

- HID 描述符(HID)
- 报告描述符(Report)
- 物理描述符(Physical)

1) HID 描述符

HID 描述符标识设备的从属描述符的长度和类型。

表 15.57 HID 描述符字段描述

字段	偏移位/大小(字节)	描述
<i>bLength</i>	0/1	表示 HID 描述符总大小的数值表示。

字段	偏移位/大小(字节)	描述
<i>bDescriptorType</i>	1/1	指定 HID 类型的常量名称描述符。
<i>bcdHID</i> (版本)	2/2	标识 HID 类规范版本的数值表示。
<i>bCountryCode</i>	4/1	识别本地化硬件的国家代码的数值表示。
<i>bNumDescriptors</i>	5/1	指定类描述符数量的数值表示总是至少一个,即报告描述符。)
<i>bDescriptorType</i>	6/1	标识类描述符类型的常量名称。
<i>wDescriptorLength</i>	7/2	表示报告描述符总大小的数值表式。
[<i>bDescriptorType</i>] ...	9/1	指定可选描述符类型的常量名称。
[<i>wDescriptorLength</i>] ...	10/2	表示可选描述符总大小的数值表式。

注意:

- 如果指定了可选描述符,则相应的长度项目也必须是指定的。
- 多个可选描述符和相关长度可以分别指定为偏移量 $(3 * n) + 6$ 和 $(3 * n) + 7$ 。
- 值 *bNumDescriptors* 标识附加类特定的数量存在的描述符。此数字必须至少为一(1),因为报告描述符始终存在。HID 描述符的其余部分具有每个附加类描述符的长度和类型。
- 值 *bCountryCode* 标识硬件的国家/地区从属。大多数硬件未本地化,因此该值为零(0)。但是,键盘可以使用该字段来指示键帽的语言。设备不需要在此字段中放置零以外的值,但某些操作环境可能需要此信息。表 15.58 指定了有效的国家代码。

表 15.58 有效国家代码

码值(十进制)	国家	码值(十进制)	国家
00	Not Supported	18	Netherlands/Dutch
01	Arabic	19	Norwegian
02	Belgian	20	Persian(Farsi)
03	Canadian-Bilingual	21	Poland
04	Canadian-French	22	Portuguese
05	Czech Republic	23	Russia
06	Danish	24	Slovakia
07	Finnish	25	Spanish
08	French	26	Swedish
09	German	27	Swiss/French
10	Greek	28	Swiss/German

码值(十进制)	国家	码值(十进制)	国家
11	Hebrew	29	Switzerland
12	Hungary	30	Taiwan
13	International(ISO)	31	Turkish-Q
14	Italian	32	UK
15	Japan(Katakana)	33	US
16	Korean	34	Yugoslavia
17	Latin American	35	Turkish-F

2) 报告描述符

报告描述符是 HID 设备最重要的一个描述符,报告描述符重要的作用是描述在中断传输中的报告的格式。对于初学者,仅仅需了解报告描述符的构成即可。要注意,报告描述符和报告是两个概念。

报告描述符与其他描述符不同,因为它不仅仅是一个值表。报告描述符的长度和内容因设备报告所需的一个或多个报告所需的数据字段数而异。报告描述符由提供有关设备的信息的**项目(item)**组成。项目的第一部分包含三个字段:**项目类型(item type)**、**项目标签(item tag)**和**项目大小(item size)**。这些字段共同标识项目提供的信息类型。

有三种项目类型:“主项目(main)”、“全局项目(global)”和“本地项目(local)”。主项目标签有五个,分别如下:

① **输入项标签(input item tag)**:指来自设备上一个或多个类似控件的数据。例如,可变数据(例如读取单个轴或一组杠杆的位置)或阵列数据(例如一个或多个按钮或开关)。

② **输出项标签(output item tag)**:指的是设备上一个或多个类似控件的数据,例如设置单个轴或一组杠杆的位置(可变数据)。或者,它可以将数据表示到一个或多个 LED(阵列数据)。

③ **特征项标签(feature item tag)**:描述不用于最终用户使用的设备输入和输出,例如,软件功能或控制面板切换。

④ **集合项标签(collection item tag)**:输入、输出和功能项的有意义的分组,例如鼠标、键盘、操纵杆和指针。

⑤ **结束集合项目标签(end collection tag)**:用于指定项目集合结束的终止项目。

报告描述符提供了设备中每个控件所提供的数据的描述。每个主项目标签(输入 Input、输出 Output 或特征 Feature)确定由特定控件返回的数据的大小,并确定数据是绝对的还是相对的,以及其他相关的信息。前面的局部和全局项目定义了最小和最大的数据值,以此类推。一个报告描述符是一个设备的所有项目的完整集合。仅仅通过查看报告描述符,应用程序就可以知道如何处理传入的数据,以及数据可以用来做什么。

控件的一个或多个数据字段由一个主项定义,并由前面的全局和本地项进一步描述。本地项只描述下一个主项所定义的数据字段。全局项是该描述符中所有后续数据字段的默认属性。

报告描述符的构成:一个报告描述符可能包含几个主项。一个报告描述符必须包含以下各项来描述控件的数据(所有其他项都是可选的):

- Input(Output or Feature) //输入或输出标志
- Usage
- Usage Page
- Logical Minimum
- Logical Maximum
- Report Size
- Report Count

以下是 3 键鼠标的项目的编码示例。在这种情况下,主项目前面是全局项目,例如使用 (Usage)、报告数量 (Report Count) 或报告大小 (Report Size) (每行都是一个新项目)。

代码清单 15-12 报告描述符示例

```
Usage Page (Generic Desktop),      ; Use the Generic Desktop Usage Page
Usage (Mouse),
    Collection (Application),      ; 鼠标集合起始
    Usage (Pointer),
    Collection (Physical),         ; 指针集合起始
        Usage Page (Buttons)
        Usage Minimum (1),
        Usage Maximum (3),
        Logical Minimum (0),
        Logical Maximum (1),      ; Fields return values from 0 to 1
        Report Count (3),
        Report Size (1),           ; Create three 1 bit fields (button 1, 2, & 3)
        Input (Data, Variable, Absolute),
                                   ; Add fields to the input report.
        Report Count (1),
        Report Size (5),           ; Create 5 bit constant field
        Input (Constant),          ; Add field to the input report
    Usage Page (Generic Desktop),
    Usage (X),
    Usage (Y),
    Logical Minimum (-127),
    Logical Maximum (127),        ; Fields return values from -127 to 127
    Report Size (8),
    Report Count (2),              ; Create two 8 bit fields (X & Y position)
    Input (Data, Variable, Relative),
                                   ; Add fields to the input report
    End Collection,                ; 指针集合结束
End Collection                     ; 鼠标集合结束
```

(1) 项目的类型和标签

所有项目都包含一个 1 字节的前缀,表示项目的基本类型。HID 类定义了项目的两种基本格式:

- 短项目:总长度 1-5 个字节;用于最常见的项目。一个短项通常包含 1 或 0 个字节的可选数据。
- 长项目:长度为 3-258 字节;用于需要较大数据的项目零件的结构。

注意:本文仅对短项目进行说明,长项目与短项目的概念与主项目、全局项目、本第项目的概念是不同的。

(2) 短项目

程序设计中使用的项目均为短项,故读者需要仔细阅读短项的内容。短项目格式将项目大小、类型和标签由第一个字节描述。第一个字节后面可能跟着 0、1、2 或 4 个可选数据字节,具体取决于数据的大小。

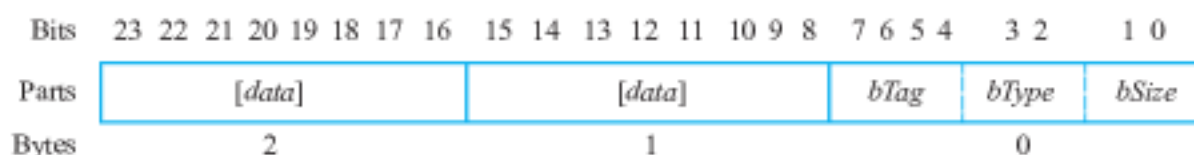


图 15.30 短项目位图

其中具体的字段的定义如表 15.59 所示。

表 15.59 字段含义

字段	描述
bSize	指明数据的大小 0=0 bytes, 1=1 byte, 2=2 bytes, 3=4 bytes
bType	指定项目类型 0=主项目, 1=全局项目, 2=本地项目, 3=保留
bTag	指定该项目的功能
[data]	可选的数据

注意

- 短项目标签没有明确的 bSize 值与之关联。相反,项目数据部分的值决定了项目的大小。也就是说,如果项目数据可以用一个字节表示,那么数据部分可以指定为 1 个字节。
- 如果预计有一个长的数据项,如果它的高位都是 0,此数据仍可以被缩写。例如,一个 32 位的数据,其中字节 1、2、3 都是 0,则该数据可以缩写为一个字节。
- 短项目分为三类:主项目、全局项目和本地项目。项目类型(bType)指定项目类别。

(3) 长项目

与短项格式一样,长项格式将项大小、类型和标签由第一个字节描述。长项格式使用一个特殊的项目标签值来指示它是一个长项。项目数据最多可包含 255 个字节的数据,如图 15.31 所示。



图 15.31 长项目位图

其中具体的字段的定义如表 15.60 所示。

表 15.60 字段含义

字段	描述
bSize	指定项目的总大小,其中大小为 10(2 字节);表示项目类型为长项目
bType	指定项目类型的数字表达式,其中 3=保留
bTag	指定该项目的功能,只能是 1111
[bDataSize]	长项目数据大小
[bLongItemTag]	长项目标签
[data]	可选的数据

重要提示:本文中并没有长项目标签相关的定义。这些标签被保留给将来使用。标签 xF0~xFF 由供应商定义。

(4) 主项目

主项目用于定义或分组报告描述符中的某些类型的数据字段,如表 15.61 所示。主项目有两种类型:数据类型和非数据类型。数据类型用于在报告中创建字段,包括输入项目、输出项目和特征项目。随后接着的字段为非数据类型,该类型不在报告中创建字段。

所有主项目默认数据值为 0。

表 15.61 主项目有效值表

主项的标签	一字节前缀 (nn 表示大小)	有效值	
Input(输入项)	1000 00 nn	Bit 0	{数据(0) 常数(1)}
		Bit 1	{数组(0) 变量(1)}
		Bit 2	{绝对(0) 相对(1)}
		Bit 3	{无回转(0) 回转(1)}
		Bit 4	{线性(0) 非线性(1)}
		Bit 5	{首选状态(0) 非首选状态(1)}
		Bit 6	{无空状态(0) 空状态(1)}
		Bit 7	保留(0) 字段(0) 缓冲作用字节(1)
		Bit 8	保留(0)
		Bit 32-9	保留(0)
Output(输出项)	1001 00 nn	Bit 0	{数据(0) 常数(1)}
		Bit 1	{数组(0) 变量(1)}
		Bit 2	{绝对(0) 相对(1)}
		Bit 3	{无回转(0) 回转(1)}
		Bit 4	{线性(0) 非线性(1)}
		Bit 5	{首选状态(0) 非首选状态(1)}
		Bit 6	{无空状态(0) 空状态(1)}
		Bit 7	{非易失(0) 易失的(1)}
		Bit 8	{字段(0) 缓冲作用字节(1)}
		Bit 32-9	保留(0)

主项的标签	一字节前缀 (nn 表示大小)	有效值	
Feature(特征项)	1011 00 nn	Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 Bit 7 Bit 8 Bit 32-9	{数据(0) 常数(1)} {数组(0) 变量(1)} {绝对(0) 相对(1)} {无回转(0) 回转(1)} {线性(0) 非线性(1)} {首选状态(0) 非首选状态(1)} {无空状态(0) 空状态(1)} {非易失的(0) 易失的(1)} {字段(0) 缓冲作用字节(1)} 保留(0)
Collection(集合项)	1010 00 nn	0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07-0x7F 0x80-0xFF	Physical(group of axes) Application(mouse, keyboard) Logical(interrelated data) Report Named Array Usage Switch Usage Modifier Reserved Vendor-defined
End Collection	1100 00 nn	无	关闭项目集合
Reserved	1101 00 nn 到 1111 00 nn	无	为以后项目作保留

(5) 输入项、输出项以及特征项

输入、输出和特征项用于在报告中创建数据字段。

输入项描述了有关由一个或多个物理控件提供的数据的信息。应用程序可以使用此信息来解释设备提供的数据。在单个项目中定义的所有数据字段共享相同的数据格式。

输出项用于定义报告中的输出数据字段。除了描述发送到设备的数据(例如键盘 LED 状态)外,该项目类似于输入项目。

特征项描述可以发送到设备的设备配置信息。

表 15.62 输入项、输出项以及特征项的详细描述

位	Part	值	描述
0	数据 常数	0 1	指示项目是数据还是常量值 数据:该项目正在定义包含可修改设备数据的报告 字段 常量:该项目是报告中的静态只读字段,不能由主 机修改(写入)

位	Part	值	描述
1	数组 变量	0 1	<p>指示项目是否在报告中创建变量或数组数组字段。</p> <p>变量:每个字段表示来自物理控件的数据。为每个字段保留的位数由前面的报告大小/报告计数项确定。例如,一组八个开/关开关可以由变量输入项声明的1个字节报告,其中每个位代表一个开关,开(1)或关(0)(报告大小=1,报告计数=8)。或者,可变输入项可以添加1个报告字节,用于表示四个三位按钮的状态,其中每个按钮的状态由两位表示(报告大小=2,报告计数=4)。或者来自变量输入项的1个字节可以表示操纵杆的x位置(报告大小=8,报告计数=1)。数组:提供了另一种方法来描述从一组按钮返回的数据。数组更有效,但不如变量灵活。数组不为组中的每个按钮返回一个位,而是在每个字段中返回一个与按下的按钮相对应的索引(如键盘扫描代码)。数组字段中的超出范围的值被视为未断言的控件。数组中同时按下的按钮或键需要在多个字段中报告。因此,数组输入项(报告计数)中的字段数决定了可以同时报告的最大数量。使用具有3个8位字段的数组(报告大小=8,报告计数=3),键盘最多可以同时报告三个键。逻辑最小值指定数组返回的最低索引值,逻辑最大值指定最大值。可以通过检查逻辑最小值和逻辑最大值之间的差异来推断数组中的元素数量(元素数量=逻辑最大值-逻辑最小值+1)</p>
2	绝对 相对	0 1	<p>指示数据是绝对的(基于固定的原点)还是相对的(指示上次报告中值的变化)</p> <p>鼠标设备通常提供相对数据</p> <p>平板电脑通常提供绝对数据</p>
3	无回转 回转	0 1	<p>指示数据在达到极高值或极低值时是否“翻转”。例如,可以自由旋转360度的表盘可能会输出0到10的值。如果指示回转,在递增方向经过10位置后报告的下一个值将是0</p>
4	线性 非线性	0 1	<p>指示来自设备的原始数据是否经过某种方式处理,不再代表测量的数据与报告的数据之间的线性关系。加速度曲线和操纵杆死区就是这类数据的例子。灵敏度设置会影响单元项目,但数据仍然是线性的</p>
5	首选状态 非首选状态	0 1	<p>指示控件是否具有当用户未与控件进行物理交互时它将返回到的首选状态。按钮(与切换按钮相反)和自定义操纵杆就是示例</p>

位	Part	值	描述
6	无空状态 空状态	0 1	指示控件是否处于未发送有意义数据的状态。空状态的一种可能用途是用于需要用户与控件进行物理交互以报告有用数据的控件。例如,一些操纵杆具有多向开关(帽子开关)。当帽子开关没有被按下时,它处于空状态。当处于空状态时,控件将报告指定逻辑最小值和逻辑最大值之外的值(最大负值,例如 8 位值的-128)
7	非易失的 易失的 Reserved	0 10	指示功能或输出控件的值是否应由主机更改。易失性输出可以在有或没有主机交互的情况下发生变化。为避免同步问题,易失性控制应尽可能是相对的。如果 volatile 输出是绝对的,则在发出 Set Report (Output)时,请求将您不想更改的任何控件的值设置为指定的逻辑最小值和逻辑最大值之外的值(最大负值,例如-128 为 8 位值)。设备会忽略对控件的无效输出 数据第 7 位没有定义输入项且其用作保留
8	字段 缓冲字节	0 1	指示控件发出固定大小的字节流。数据字段的内容由应用程序确定。缓冲区的内容不会被解释为单个数字量。由缓冲字节定义的报告数据必须在 8 位边界上对齐。来自条形码阅读器的数据就是一个例子
9-31	保留	0	保留

附注:

- 如果输入项是一个数组,则只有“数据/常数”、“变量/数组”还有“绝对/相对”属性适用。
- 可以通过检查“报告大小(Report Size)”和“报告计数(Report Count)”值来确定报告中的数据字段大小。例如,报告大小为 8,且报告计数为 3,则报告具有 3 个 8 位数据字段。
- 数组项返回的值是一个索引,因此建议:
 - 1) 当数组中没有控件被断言时,数组字段返回 0 值。
 - 2) 逻辑最小值(Logical Minimum)值等于 1。
 - 3) 逻辑最大值(Logical Maximum)等于数组中元素的数量。
- 输入项定义输入报告,可通过控制管道用 Get_Report(Input)请求访问。
- 输入报告也通过中断输入管道以轮询方式发送。
- 输出项的数据/常数、变量/数组,绝对/相对、非线性、包络和空状态数据与输入项的数据相同。
- 输出项使输出报告可以通过 Set_Report(output)命令通过控制管道访问。
- 输出类型的报告可以选择通过一个中断输出管道发送。虽然功能相似,但输出项和特征项在以下方面有所不同:

- 1) 特征项定义了设备的配置选项,通常由控制面板应用程序设置。因为它们会影响设备的行为(例如,按键重复速率、复位原点等),特征项通常对软件应用程序不可见。相反,输出项代表设备对用户的输出(例如,LED、音频、触觉反馈等等)。软件应用可能会设置设备的输出项。
- 2) 特征项目可以是其他项目的属性。例如,一个原点复位特性可能适用于一个或多个位置输入项目。与输出项一样,特征项可通过控制管道的 Get_Report 和 Set_Report 请求访问。

(6) 集合项和集合结束项

集合项标识两个或多个数据(输入(Input)项、输出(Output)项或特征(Feature)项)之间的关系。例如,鼠标可以描述为两到四个数据(x坐标、y坐标、按键1、按键2)的集合。当使用集合项(Collection item)打开一个数据集合时,必须使用集合结束项(End Collection item)关闭一个集合。表 15.63 详细描述了集合项第一字节代表的含义。

表 15.63 集合项详细

集合的类型	值	描述
物理(physical)	0X00	物理集合用于代表在一个几何点收集的数据点的一组数据项目。这对于可能需要将测量或传感的数据集与一个点联系起来的传感设备很有用。它并不表示一组数据值来自一个设备,如键盘。在报告多个传感器位置的设备的情况下,物理集合被用来显示哪些数据来自于每个独立的传感器
应用(Application)	0X01	一组应用程序可能熟悉的主要项目。它还可用于识别在单个设备中服务于不同目的的项目组。常见的例子是键盘或鼠标。带有集成定点设备的键盘可以定义为两个不同的应用程序集合。数据报告通常(但不一定)与应用程序集合相关联(每个应用程序至少有一个报告ID)
逻辑(Logical)	0X02	当一组数据项形成一个复合数据结构时,就会使用逻辑集合。这方面的一个例子是数据缓冲区和数据的字节计数之间的关联。该集合建立了计数和缓冲区之间的联系
报告(report)	0X03	定义一个包含报告中所有字段的逻辑集合。此集合中将包含一个唯一的报告ID。应用程序可以很容易地确定设备是否支持某个功能。请注意,可以为报告集合声明任意有效的报告ID值
命名数组(named array)	0X04	命名数组是一个逻辑集合,包含一个选择器用法的数组。对于给定的功能,类似设备使用的选择器集可能会有所不同。在记录硬件寄存器时,字段的命名是常见的做法。要确定设备是否支持特定功能(如状态),应用程序可能必须查询几个已知的状态选择器用法,然后才能确定设备是否支持状态。命名数组用法允许命名包含选择器的数组字段,因此应用程序只需查询状态用法即可确定设备支持状态信息

集合的类型	值	描述
用法切换 (Usage Switch)	0X05	Usage Switch 是一个逻辑集合,可以修改它所包含的用法的含义。此集合类型向应用程序指示在此集合中找到的用法必须是特殊情况。例如,不是在 LED 页面上为每个可能的功能声明用法,而是可以将指示符用法应用于 Usage Switch 集合,并且该集合中定义的标准用法现在可以标识为功能的指示符,而不是功能本身。请注意,此集合类型不用于标记 Ordinal 集合,而是使用逻辑集合类型
用法修饰 (Usage Modifier)	0X06	修改附加到包含集合的用法的含义。用法通常定义控件的单一操作模式。用法修饰允许扩展控件的操作模式。例如,LED 通常是打开或关闭的。对于特定状态,设备可能需要一种通用方法来闪烁或选择标准 LED 的颜色。将 LED 使用情况附加到 Usage Modifier 集合将向应用程序指示该使用情况支持新的操作模式
Reserved	0X07-0X7F 0X80-0XFF	保留 厂商定义

附注:

- 集合项和集合结束项之间的所有主项都包含在该集合中。一个集合可能包含其他集合中。
- 集合项不生成数据。但是,使用项标签必须与任何集合(例如鼠标或油门设备)相关联。集合项可以嵌套,且它们始终是可选的,但顶级应用程序集合除外。
- 如果遇到未知的供应商定义的集合类型,则应用程序必须忽略该集合中声明的所有主要项目。请注意,在该集合中声明的全局项目将影响状态表。
- 如果有未知用法附加到已知集合中,则应忽略该集合的内容。请注意,在该集合中声明的全局项目将依然影响状态表。
- 字符串和物理索引以及分隔符可能与集合相关联。

(7) 全局项

全局项描述而不是定义来自控件的数据。一个新的主项目承担着项目状态表的特征。全局项目可以改变状态表。因此,除非被另一个全局项覆盖,否则全局项标记适用于所有随后定义的项。表 15.64 一字节前缀指的是项目的第 0 字节。

表 15.64 全局项

全局项标签 (global item tag)	一字节前缀 (nn 表示大小值)	描述
Usage page	0000 01nn	指定当前使用页面的无符号整数。由于使用是 32 位值,因此可以使用使用页面项通过设置后续使用的高 16 位来节省报告描述符中的空间。定义为 16 位或更少的任何使用都被解释为使用 ID 并与使用页面连接形成一个 32 位的用法

全局项标签 (global item tag)	一字节前缀 (nn 表示大小值)	描述
Logical Minimum	0001 01nn	以逻辑单位表示的范围值。这是变量或数组项将报告的最小值。例如,报告 x 位置值从的 0 到 128 范围变化的鼠标,将具有 0 的逻辑最小值和 128 的逻辑最大值
Logical Maximum	0010 01nn	以逻辑单位表示的范围值。这是变量或数组项将报告的最大值
Physical Minimum	0011 01nn	变量项的物理范围的最小值。这表示应用了单位的逻辑最小值
Physical Maximum	0100 01nn	变量项的物理范围的最大值
Unit Exponent	0101 01nn	以 10 为底的单位指数值
Unit	0110 01nn	单位值
Report Size	0111 01nn	无符号整数,以位为单位指定报告字段的大小。这允许解析器构建一个项目映射供报告处理程序使用
Report ID	1000 01nn	<p>指定报告 ID 的无符号值。如果在报告描述符的任何地方使用了报告 ID 标签,则设备的所有数据报告都在一个单字节的 ID 字段之前。在第一个报告 ID 标签之后但在第二个报告 ID 标签之前的所有项目都包括在一个以 1 字节 ID 为前缀的报告中。在第二个报告 ID 标签之后但在第三个报告 ID 标签之前的所有项目都包括在第二个报告中,以第二个 ID 为前缀,依此类推</p> <p>这个报告 ID 值表示添加到一个特定报告的前缀。例如,一个报告描述符可以定义一个 3 字节的报告,报告 ID 为 01。这个设备将产生一个 4 字节的数据报告,其中第一个字节是 01。该设备还可以生成其他报告,每个报告都有一个独特的 ID。这允许主机区分不同类型的报告通过管道中的一个中断传送。并允许设备区分通过单个中断输出管道传送的不同类型的报告。报告 ID 0 保留,不能被使用</p>
Report Count	1001 01nn	无符号整数,指定项目的数据字段数;确定该特定项目的报告中包含多少字段(以及因此向报告中添加多少位)
Push	1010 01nn	将全局项目状态表的副本放在堆栈上
Pop	1011 01nn	将项目状态表替换为堆栈中的栈顶
Reserved	1100 01nn to 1111 01nn	保留

附注:

- 虽然逻辑最小值和逻辑最大值(范围)限制了设备返回的值,但物理最小值和物理最大值通过允许报告值偏移和缩放来赋予这些边界意义。例如,温度计的逻辑范围可能为 0 和 999,但物理范围为 32 和 212。分辨率可以通过以下算法确定。

```

if ((Physical Maximum == UNDEFINED)
|| (Physical Minimum == UNDEFINED)
|| ((Physical Maximum == 0) && (Physical Minimum == 0)))
{
    Physical Maximum = Logical Maximum;
    Physical Minimum = Logical Minimum;
}

If (Unit Exponent == UNDEFINED)
    Unit Exponent = 0;

Resolution = (Logical Maximum - Logical Minimum) / ((Physical Maximum - Physi-
cal Minimum) * (10 Unit Exponent))

```

当线性解析报告描述符时,单位指数、物理最小值和物理最大值的全局状态值被认为处于“未定义”状态,直到它们被声明。

例如,一个 400-dpi 的鼠标可能有表 15.65 中所示的项目。

表 15.65 鼠标项目

项目	值
Logical Minimum	-127
Logical Maximum	127
Physical Minimum	-3175
Physical Maximum	3175
Unit Exponent	-4
Unit	Inches

因此其计算分辨率的公式为:

$$\text{Resolution} = (127 - (-127)) / ((3175 - (-3175)) * 10^{-4}) = 400 \text{ 计数值/英尺}$$
 单位项目的限定值如表 15.66 所示。

表 15.66 单位项目的限定值

半字节	单位制度	0x0	0x1	0x2	0x3	0x4
	Exponent	0	1	2	3	4
0	System	None	SI Linear	SI Rotation	English Linear	English Rotation
1	Length	None	Centimeter	Radians	Inch	Degrees
2	Mass	None	Gram	Gram	Slug	Slug
3	Time	None	Seconds	Seconds	Seconds	Seconds
4	Temperature	None	Kelvin	Kelvin	Fahrenheit	Fahrenheit

半字节	单位制度	0x0	0x1	0x2	0x3	0x4
5	Current	None	Ampere	Ampere	Ampere	Ampere
6	Luminous intensity	None	Candela	Candela	Candela	Candela
7	Reserved	None	None	None	None	None

注意:对于系统部分,代码 0x5 至 0xE 是保留;代码 0xF 是供应商定义的

- 如果逻辑最小值和逻辑最大值都被定义为正值(0 或更大),那么报告字段可以被假定为无符号值。否则,所有整数值都是以 2 的补码格式表示的有符号值。
- 在报告描述符中声明物理最小值和物理最大值之前,HID 分析器会假定它们分别等于逻辑最小值和逻辑最大值。将它们声明为可以应用于(输入项、输出项或特征项)主项后,它们将继续影响所有后续的主项。如果物理最小值和物理最大值的范围都等于 0,那么它们将恢复到它们的默认释义。
- 之前的表格中没有显示的代码和指数,如表 15.67 所示。

表 15.67 指数项

码值	指数
0x5	5
0x6	6
0x7	7
0x8	-8
0x9	-7
0xA	-6
0xB	-5
0xC	-4
0xD	-3
0xE	-2
0xF	-1

- 大多数复杂的单位可以从长度、质量、时间、温度、电流和发光强度的基本单位中得出。例如,能量(焦耳)可以表示为:

```
joule = [mass(grams)] [length(centimeters) 2] [time(seconds) -2]
```

单位指数将是 7,因为一个焦耳是由公斤(1 公斤等于 10³ 克)和米组成,如表 15.68 所示。

表 15.68 示例

Nibble	Part	Value
3	Time	-2
2	Mass	1
1	Length	2
0	System	1

- 一些常见单位的部件如表 15.69 所示。

表 15.69 常见单位的部件

单位	半字节						
	5(i)	4(t)	3(t)	2(m)	1(l)	0(sys)	Code
Distance(cm)	0	0	0	0	1	1	x0011
Mass(g)	0	0	0	1	0	1	x0101
Time(s)	0	0	1	0	0	1	x1001
Velocity(cm/s)	0	0	-1	0	1	1	xF011
Momentum	0	0	-1	1	1	1	xF111
Acceleration	0	0	-2	0	1	1	xE011
Force	0	0	-2	1	1	1	xE111
Energy	0	0	-2	1	2	1	xE121
Angular Acceleration	0	0	-2	0	1	2	xE012
Voltage	-1	0	-3	1	2	1	x00F0D121

- 在数组的情况下,报告计数决定了可能包含在报告中的最大控件数量,因此也决定了可能同时按下的键或按钮的数量以及每个元素的大小。例如,一个支持最多同时按三个键的数组,每个字段是 1 个字节,看起来像这样:

```
...
Report Size (8),
Report Count (3),
...
```

- 在变量项目的情况下,报告计数指定了报告中包括多少个控件。例如,八个按钮可以是这样的:

```
...
Report Size (1),
Report Count (8),
...
```

- 如果使用报告 ID,则必须在报告描述符中的第一个输入、输出或功能主项声明之前声明一个报告 ID。
- 在报告描述符中可以多次遇到相同的报告 ID 值。随后声明的输入、输出或功能主要项目将在相应的 ID/类型(输入、输出或功能)报告中找到。

(8) 本地项

本地项目标签定义控件的特征。这些项目不会延续到下一个主项目。如果一个主项定义了一个以上的控件,那么它前面可能有几个类似的本地项标签。例如,一个输入项可能有几个与之相关的使用标签。表 15.70 所示一字节前缀指的是项目的第 0 字节。

标签位	一字节前缀 (nn 表示大小值)	描述
Usage	0000 10 nn	项目使用情况的使用索引;表示项目或集合的建议用法。在一个项目表示多个控件的情况下,Usage 标签可能会建议数组中每个变量或元素的用法
Usage Minimum	0001 10 nn	定义与数组或位图关联的起始用法
Usage Maximum	0010 10 nn	定义与数组或位图关联的结束用法。
Designator Index	0011 10 nn	确定用于控件的主体部分。索引指向物理描述符中的指示符
DesignatorMinimum	0100 10 nn	定义与数组或位图关联的起始指示符的索引
DesignatorMaximum	0101 10 nn	定义与数组或位图关联的结束指示符的索引
String Index	0111 10 nn	字符串描述符的字符串索引;允许字符串与特定项或控件相关联
String Minimum	1000 10 nn	指定将一组连续字符串分配给数组或位图中的控件时的第一个字符串索引
String Maximum	1001 10 nn	指定将一组连续字符串分配给数组或位图中的控件时的最后一个字符串索引
Delimiter(定界符)	1010 10 nn	定义一组局部项的开始或结束(1=打开集(open set), 0=关闭集(close set))
Reserved	1010 10 nn to 1111 10 nn	保留

附注:

- 虽然本地项目不会延续到下一个主要项目,但它们可能适用于单个项目中的多个控件。例如,如果定义五个控件的输入项前面有三个用法标记,则三个用法将按顺序分配给前三个控件,第三个用法也将分配给第四个和第五个控件。如果项目没有控件(报告计数=0),则本地项目标签适用于主项目(通常是集合项目)。
- 要为单个主项中的每个控件分配唯一的用法,只需按顺序指定每个 Usage 标签(或使用 Usage Minimum 或 Usage Maximum)。
- 所有本地项都是无符号整数。
注意:正确使用用法很重要。虽然存在非常具体的用途(起落架、自行车车轮等),但这些用途旨在识别具有非常具体应用的设备。带有通用按钮的操纵杆绝不应将特定于应用程序的用途分配给任何按钮。相反,它应该分配一个通用用法,例如“按钮”。但是,健身车或飞行模拟器的驾驶舱可能希望狭义地定义其每个数据源的功能。
- 同样重要的是要记住,使用项(Usage items)传达有关数据预期用途的信息,并且可能与实际测量的内容不对应。例如,操纵杆将具有与其轴数据关联的 X 和 Y(Y Usage)使用情况(而不是使用情况 Rx 和 Ry)。
- 因为按钮位图和数组可以用一个项目表示多个按钮或开关,所以将多个用途分配给一个主项目可能很有用。Usage Minimum 指定要与数组或位图中的第一个未关联控件关联的使用情况。使用最大值指定与项目元素关联的使用值范围的结束。以下示例

说明了如何将其用于 105 键键盘,如表 15.71 所示。

表 15.71 键盘报表描述符

Tag(标签)	Result(结果)
Report Count(1)	一个字段将添加到报告中
Report Size(8)	新增字段的大小为 1 字节(8 位)
Logical Minimum(0)	将 0 定义为可能的最低返回值
Logical Maximum(101)	将 101 定义为可能的最高返回值,并将范围设置为 0 到 101。将 101 定义为可能的最高返回值,并将范围设置为 0 到 101
Usage page(0X07)	选择键盘使用页面。(select keyboard usage page)
Usage Minimum(0X00)	分配 101 键用法第一个键
Usage Maximum(0X65)	分配 101 键用法最后一个键
Input; (Data, Array, Absolute)	在中断报告中创建并添加 1 字节数组

- 如果将使用最小值 (Usage Minimum) 声明为扩展使用,则关联的使用最大值 (Usage Maximum) 也必须声明为扩展使用。
- Usage、Usage Minimum 或 Usage Maximum 项目的解释根据项目的 bSize 字段而有所不同。如果 bSize 字段 = 3,则该项目被解释为 32 位无符号值,其中高 16 位定义使用页面 (Usage page),低 16 位定义使用 ID (Usage ID)。定义使用页面和使用 ID 的 32 位使用项通常称为“扩展”使用。

如果 bSize 字段 = 1 或 2,则 Usage 被解释为在当前定义的 Usage Page 上选择 Usage ID 的无符号值。当解析器遇到一个主要项目时,它会将最后声明的 Usage Page 与一个 Usage 连接起来,形成一个完整的使用值。扩展使用可用于覆盖当前定义的单个使用的使用页面。

- 通过简单地将它们与分隔符项 (delimiter item) 括起来,可以将两个或多个替代用法与控件相关联。分隔符允许为控件定义别名,以便应用程序可以通过多种方式访问它。形成分隔集的用法按优先顺序组织,其中声明的第一个用法是控件的最首选用法。HID 解析器必须处理分隔符 (delimiters),但是,对它们定义的替代用法的支持是可选的。系统软件可能无法访问除定义的第一个 (最首选) 用法之外的其他用法。

定义适用于应用程序集合 (Application collection) 或数组项 (Array item) 的用法时,不能使用分隔符。

(9) 填充

报告可以通过声明适当大小的主项和不声明主项的用法来填充到字节对齐的字段。

3) 物理描述符

该特定描述符不在之后的程序设计中涉及,故在此不作讲解,读者如对此感兴趣请阅读 HID 官方协议规。

15.3.4 报告

设备通过向主机发送报告,可以将诸如鼠标移动、点击等信息返回给主机。报告通过中断输入管道从设备发送到主机。报告也可以请求 (轮询) 报告并通过控制管道发送,或通过可选

的中断输出管道发送。主机需依据设备提供的报告描述符对设备上传的报告进行识别。在此重复提醒:报告与报告描述符是两个概念

如果在报告描述符中使用了报告 ID 标签,则所有报告都包含一个单字节 ID 前缀。如果未使用报告 ID 标签,则所有值都将在单个报告中返回,并且前缀 ID 不包含在该报告中。

1. 标准项报告格式

报告格式由 8 位(8-bit)报告标识符和属于该报告的数据组成,如图 15.32 所示。



图 15.32 位图

报告 ID

报告 ID 字段的长度为 8 位。如果报告描述符中没有使用报告 ID 标签,则只有一份报告,并且报告 ID 字段被省略。

报告数据

数据字段是报告可变长度字段。

2. 数组项的报告格式

数组中的每个按钮都报告一个分配的编号,称为数组索引。这可以通过查找数组元素“用法”页和“用法”转换为键码。当任何按钮在打开和关闭之间转换时,数组中当前关闭的按钮的整个索引列表将传输到主机。

由于每个数组字段中只能报告一个数组元素,因此修饰键应报告为位图数据(一组 1 位变量字段)。例如,CTRL、SHIFT、ALT 和 GUI 键等键构成了标准键盘报告中的 8 位修饰符字节。尽管这些用法代码在用法表中定义为 E0-E7,但用法不会作为数组数据发送。修饰符字节定义如表 15.72 所示。

表 15.72 修饰按键字节位结构

偏移	位长	说明
0	1 bit	左 ctrl
1	1 bit	左 shift
2	1 bit	左 alt
3	1 bit	左 GUI(如 windows 键)
4	1 bit	右 ctrl
5	1 bit	右 shift
6	1 bit	右 alt
7	1 bit	右 GUI(如 windows 键)

下面的示例演示由键入 ALT+CTRL+DEL 的用户生成的报告,如表 15.73 所示。

表 15.73 修饰按键字节位结构

按键变化	修饰字节	数组字节
左 alt 按下	0000 0100	00
右 ctrl 按下	0001 0100	00
del 按下	0001 0100	4C
del 弹起	0001 0100	00
右 ctrl 弹起	0000 0100	00
左 alt 弹起	0000 0000	00

如果此设备有多个报告,则每个报告前面都会有其唯一的报告 ID,如图 15.33 所示。

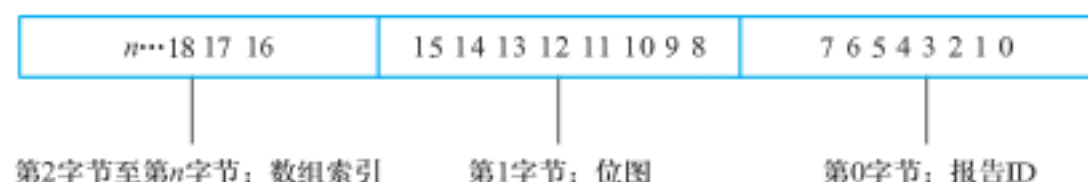


图 15.33

如果一组键或按钮不能互斥,则它们必须表示为位图或多个数组。例如,101 键键盘上的功能键有时用作修饰键,例如 F1、A。在这种情况下,应在一个数组项中报告至少两个数组字段,即 Report Count(2)。

3. 报告约束

以下约束适用于报告和报告处理程序:

- 报告中的项目字段不能超过 4 个字节。例如,一个 32 位项目必须从字节边界开始以满足此条件。
- 一次 USB 传输中只允许一份报告。
- 一份报告可能跨越一个或多个 USB 事务。例如,具有 10 字节报告的应用程序将跨越低速设备中的至少两个 USB 事务。
- 除了最长的超过端点的 wMaxPacketSize 的报告之外的所有报告都必须以一个短数据包终止。最长的报告不需要短数据包终止。
- 每个顶级集合必须是一个应用程序集合,并且报告不得跨越多个顶级集合。
- 如果顶级集合中有多个报告,则所有报告(最长的报告除外)都必须以短数据包结束。
- 报告始终是字节对齐的。如果需要,报告用位(0)填充,直到到达下一个字节边界。

4. 报告示例

代码清单 15-14 中的内容为某个设备的报告描述符

代码清单 15-14 报告描述符示例

```
Usage Page (Generic Desktop),
Usage (Mouse),
Collection (Application),
    Usage (Pointer),
    Collection (Physical),
```

```

Report ID (0A),           ;Make changes to report 0A Usage (X), Usage (Y),
Logical Minimum (-127), ;Report data values range from -127
Logical Maximum (127),  ;to 127
Report Size (8), Report Count (2),
Input (Data, Variable, Relative),
                               ;Add 2 bytes of position data (X & Y) to report 0A
Logical Minimum (0),     ;Report data values range from -127
Logical Maximum (1),     ;to 127
Report Count (3), Report Size (1),
Usage Page (Button Page),
Usage Minimum (1),
Usage Maximum (3),
Input (Data, Variable, Absolute),
                               ;Add 2 bits (Button 1,2& 3) to report 0A

Report Size (5),
Input (Constant),        ;Add 5 bits padding to byte align the report 0A
End Collection,
End Collection

```

上述设备生成的报告结构如图 15.34 所示。

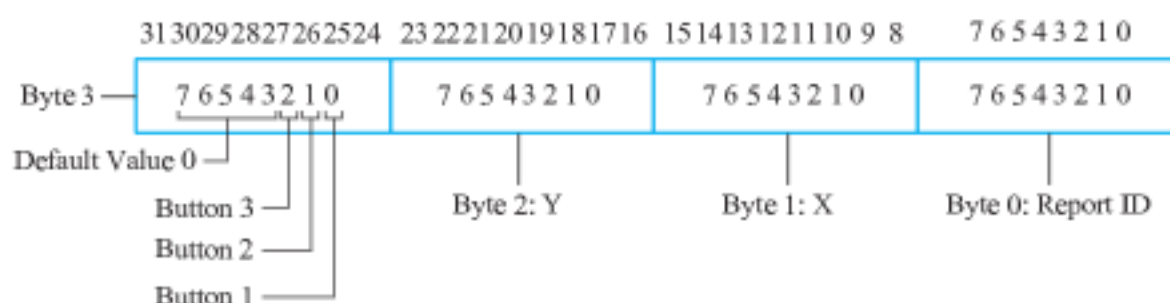


图 15.34 示例报告描述符设备生成的报告结构

表 15.74 使用带有集成鼠标功能的键盘来演示如何为只有一个接口的设备使用两个报告。

表 15.74 生成的报告

条目	用法	报告 ID
Collection(Application)	键盘	
Report ID(00)		
Input(变量, 绝对)	修饰键	00
Output(变量, 绝对)	LEDs	00
Input(数组, 绝对)	主键	00
End Collection		
Collection(Application)	鼠标	
Report ID(01)		
Collection(Physical)	指针	
Input(变量, 相对)	X, Y 坐标	01
Input(变量, 绝对)	Button	01
End Collection		
End Collection		

注意:仅存在输入、输出和特征项(非集合项)报告中的数据。此示例演示了多个报告,但是此接口不能用于引导设备(对键盘和鼠标设备要使用单独的接口)。

15.3.5 HID 协议小结

USB HID 设备主要基于两种协议:报告协议(report protocol)和引导协议(boot protocol)。报告是从设备发送到主机的数据结构,也可以从主机发送到设备。当设备向主机发送报告时,通常包含状态变化信息,例如按键、鼠标移动等。当主机向设备发送报告时,通常包含用于配置设备的命令(set_report 请求),例如在键盘上设置 LED。这个协议依赖于标准的 USB 框架。USB HID 设备使用中断传输进行通信,因为设备并不是一直在传输数据,但当设备传输数据时,主机需要软件有快速的响应。中断传输时传输的数据量通常很小。报告通常有两种类型,具体取决于协议类型。HID 报告协议基于“项目(item)”的概念,其数据结构在报告描述符中定义。引导协议要简单得多,并且遵循鼠标和键盘的标准结构。依据本次实验内容,此处将仅讨论引导协议。

1. 主机检测 HID 设备

HID 设备在其设备描述符中的类(blInterfaceClass)/子类(blInterfaceSubclass)值都为零,其接口描述符中的类(blInterfaceClass)与子类(blInterfaceSubclass)的值才有效。注意:不能手动请求接口描述符,并且必须与配置和端点描述符一起获取。接口描述符中标识 HID 设备的类(blInterfaceClass)值为 3。接口描述符中的子类(blInterfaceSubclass)值可以是 1,表示设备支持引导协议,也可以是 0,表示设备仅支持报告协议。接口描述符中的协议(blInterfaceProtocol)字段也确定它是鼠标还是键盘。具体来说,1 表示 HID 设备是键盘,2 表示 HID 设备是鼠标。

2. “GET_REPORT” 请求

该软件可以使用控制端点和常规 SETUP 数据包从 USB 设备请求报告。SETUP 数据包的请求类型将包含 0xA1,“GET_REPORT”的请求代码为 1,SETUP 数据包的“wValue”字段将包含 0x0100,以请求 ID 为 0 的输入数据包,并且该字长将会是数据阶段所期望接收的字长。SETUP 数据包应发送到设备端点 0(控制端点)。对于键盘,中断传输阶段通常为 8 个字节,而对于鼠标,中断传输阶段具有标准格式的前 3 个字节,而其余的可能由设备特定的功能使用。建议仅以这种方式接收报告以测试设备初始化是否成功完成等,并且不推荐使用“GET_REPORT”请求来轮询 HID 设备以进行更改,因为 SETUP 和 STATUS stage 浪费了太多时间。相反,软件应使用中断传输,使用中断 IN 端点轮询 HID。

3. 中断端点

通常建议 HID 使用中断传输向软件报告,并且软件通常应避免上述“GET_REPORT”请求。驱动程序软件应请求 HID 设备的配置描述符。HID 设备必须支持至少一种配置。然后软件应扫描端点描述符,搜索指示“中断输入”类型的描述符,该端点是使用中断传输将设备数据发送到主机的端点。软件应保存端点的 4 位 ID,以及端点的 8 位间隔。间隔值以毫秒(ms)为单位对时间进行编码,软件应在该时间空间内轮询一次报告包。例如,如果间隔值为 8,则软件应每 8 毫秒向设备请求一次报告。如果软件过早请求报告,例如在 6 ms 之后,设备可能会发送与之前相同的数据包,或者它可能不发送任何内容,而是返回 NAK。如果软件在

时间跨度之后请求报告,例如 9 毫秒,则设备将发送新数据包。软件使用此描述的方法不断轮询 USB HID 设备。

4. USB 键盘

USB 键盘使用报告与软件通信,就像其他 HID 设备一样。通过接口描述符中的类字段 (bInterfaceClass)3 和协议字段 (bInterfaceProtocol)1 来检测 USB 键盘。

1) 报告格式

该报告必须由软件每隔固定时间使用一次中断传输请求,并且此间隔应在 USB 键盘的端点 IN 触发的中断中定义。USB 键盘报告的大小可能高达 8 个字节,但这些字节不一定都被使用,如果只使用前三个或四个字节来实现正确的键码传输是可行的。此处描述键盘的完整报告机制。表 15.75 定义的报告数据结构仅适用于引导协议。

表 15.75 USB 键盘报告结构

偏移	大小	说明
0	1 字节	修饰按键键状态
1	1 字节	保留
2	1 字节	普通按键按下#1
3	1 字节	普通按键按下#2
4	1 字节	普通按键按下#3
5	1 字节	普通按键按下#4
6	1 字节	普通按键按下#5
7	1 字节	普通按键按下#6

修饰按键状态:该字节是一个位域,其中每个位对应于一个特定的修饰键。当某位置 1 时,则表示对应修饰键被按下。该字节的位结构见表 15.76。

表 15.76 修饰按键字节位结构

位	位长	说明
0	1 bit	左 ctrl
1	1 bit	左 shift
2	1 bit	左 alt
3	1 bit	左 GUI(如 windows 键)
4	1 bit	右 ctrl
5	1 bit	右 shift
6	1 bit	右 alt
7	1 bit	右 GUI(如 windows 键)

按键字段:一个键盘报告数据结构最多可表示 6 个普通按键按下状态。按键值是无符号的 8 位二进制编码,该编码为扫描码(HID 码),不是 ASCII 码。

2) 按键机制

USB 键盘在按下或释放键时发送中断。当用户按下一个键时,中断会在其中一个按键字段中带有扫描码值。当一个键被释放时,对应的按键字段在下一个数据包中返回零。为了更清楚地说明这一点并说明为什么有多个普通按键扫描码字段,让我们看以下示例。假设用户按下“A”键,即扫描码 0x04。返回的中断数据包为:

```
00 00 04 00 00 00 00 00
```

注意修饰键为零,因为用户没有按下修饰按键键。按照 USB HID 规范的建议,保留字段也为零。第一个普通按键字段包含 0x04,对应于“A”键。现在,让我们假设用户松开“A”键。发送的数据包如下所示:

```
00 00 00 00 00 00 00 00
```

现在,假设用户按下“A”键,然后按下“B”键(扫描码 0x05)而不松开“A”键,发送的数据包如下:

```
00 00 04 05 00 00 00 00
```

注意一个中断数据包如何能够同时传输两个按键。现在让我们假设用户按下“C”键(扫描码 0x06)而不放开“A”或“B”键,发送的数据包如下:

```
00 00 04 05 06 00 00 00
```

显然,USB 键盘按照首先按下的顺序返回扫描码。因此,如果第一个普通按键字段为零,则没有按键被按下。如果它不为零,主机软件可以检查下一个字段,以查看是否也按下了另一个普通按键。

修饰键的概念十分简单,我们假设用户使用“X”键(扫描码 0x1B)按下左 shift 键。发送的数据包如下:

```
02 00 1B 00 00 00 00 00
```

注意,修饰符字段的第 1 位(值 0x02)已设置,表示正在按下左 shift 键。

还有一个“幻象状态”,可以将其视为溢出。一个 USB 键盘数据包在一次传输中最多可以指示 6 个按键,但是让我们想象一下有人一次按下了 6 个以上的按键。键盘将进入幻象状态,其中所有报告的键都是无效的扫描码 0x01。但是,仍会报告修改键。例如:有 8 个键(或任何大于 6 的随机数)被按下,同时右 shift 键也被按下。发送的数据包如下所示:

```
20 00 01 01 01 01 01 01
```

修饰键字段数据正常,但普通按键字段都返回了幻像状态。除了幻象条件外,还有其他特殊的扫描码:0x00 表示没有扫描码并且没有按键被按下,0x01 表示我们刚刚解释的幻象状态,0x02 表示键盘的自检失败,0x03 表示发生了未定义的错误。从 0x04 开始,扫描码有效并对应

于“真实”按键。

3) LED 灯的设置

LED 灯也是在软件中处理的,根据硬件,NumLock、CapsLock 和 ScrollLock 是发送正常扫描码的正常键。当按下这些键之一时,驱动程序负责操作 LED 灯。

为了设置 LED 灯,驱动程序使用标准 USB 设置事务向设备发送一个 SET_REPORT 请求,其中包含一个字节的的数据阶段。设置包的请求类型应为 0x21,SET_REPORT 的请求代码为 0x09。SETUP 数据包的 value 字段在低字节中包含报告 ID,其值为 0。高字节包含报告类型,其值为 0x02 表示输出报告,或从软件发送到硬件的报告。wIndexL 字段应包含 USB 键盘的接口编号,即接口描述符中的编号,表明该设备是 USB 键盘。如果硬件支持中断 OUT 端点,只需将 1 字节数据阶段传输到中断 OUT 端点,而无需额外的 SETUP 阶段开销。如果硬件支持中断 OUT 端点,则应尽可能避免使用控制端点,因为轮询中断 OUT 端点更快。

主机向 OUT 端点发送的设置 LED 灯的 1 个字节数据结构如表 15.77 所示。

表 15.77 1 个字节数据的数据结构

位	位长	说明
0	1	Num Lock 指示灯
1	1	Caps Lock 指示灯
2	1	Scroll Lock 指示灯
3	1	Compose
4	1	Kana
5	3	保留

5. USB 鼠标

USB 鼠标与任何其他 HID 设备一样,使用报告与软件通信,报告通过中断端点发送,或者可以通过“GET_REPORT”请求手动请求。USB 鼠标在接口描述符中的协议字段(bInterfaceProtocol)为 2。

1) 报告格式

主机必须每隔固定时间使用一次中断传输请求此报告。USB 鼠标设备中,报告仅有 3 个字节,其依次从低到高进行排列,表 15.78 给出了支持引导协议的 USB 鼠标的报告数据结构。

表 15.78 USB 鼠标的报告数据结构

偏移	大小/字节	描述
0	1	按键状态
1	1	X 方向上的位移(相对值)
2	1	Y 方向上的位移(相对值)

按键状态:该字段的字节位结构如表 15.79 所示。

表 15.79 按键状态字节位结构

偏移位(第几个比特位)	比特长度/bit	描述
0	1	当其值为 1 的时候表示左键按下
1	1	当其值为 1 的时候表示右键按下
2	1	当其值为 1 的时候表示中键按下

对于第二个和第三个字节则表示 X 和 Y 方向上的位移,该表示方法遵循右手坐标系,右移为正,下移为正。其 256 个数据的分配为:值小于 127 为正,大于 127 视为负值(-1 到-127 的补码表示)。