WebHID Call Control

HID描述符解析

1. **前言**

   本节内容参考 https://blog.csdn.net/Britripe/article/details/111662665 ，通过本节可以了解HID协议作用以及如何解析该协议。

   HID它的作用十分广泛，鼠标、键盘、VR、医疗设备、教育设备都是使用该报表（也称报告描述符，英文report descriptor）实现，这个报表是USB最灵活也是最复杂的报表。

   在开始阅读前，请先自行回忆一下十六进制与二进制如何相互转换的知识，字节与bit之间的关系。

2. **报告描述符组成**

   报告描述符由每个项目（item）组成，下图中每一行即为一个项目

```
HID_Report_desc_table:
 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=+127
 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=+127
 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
```

下图是经过usb lyzer解析后的描述符，可读性更高（注意：下图和上图为两份不同的描述符）

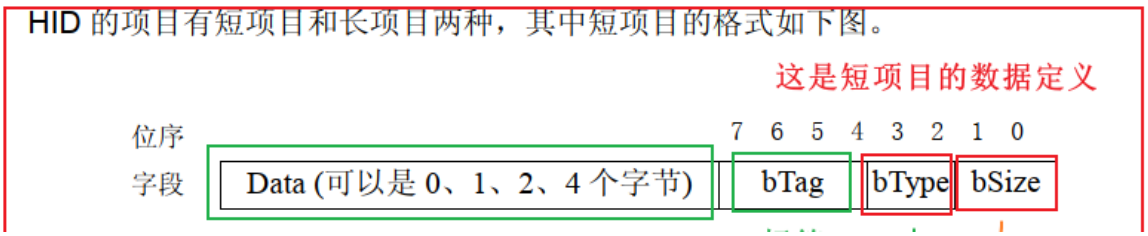
Interface 1 HID Report Descriptor Consumer Control

Item Tag (Value)	Raw Data
Usage Page (Consumer Devices)	05 0C
Usage (Consumer Control)	09 01
Collection (Application)	A1 01
Report ID (1)	85 01
Usage (Volume)	09 E0
Logical Minimum (-24)	15 E8
Logical Maximum (24)	25 18
Report Size (7)	75 07
Report Count (1)	95 01
Input (Data,Var,Rel,NWrp,Lin,Pref,NNul,Bit)	81 06
Logical Minimum (0)	15 00
Logical Maximum (1)	25 01
Report Size (1)	75 01
Usage (Mute)	09 E2
Input (Data,Var,Rel,NWrp,Lin,Pref,NNul,Bit)	81 06
End Collection	C0

3. **项目（item）**

   项目可以分为长项目和短项目，长项目用的比较少本文不做过多介绍。

   项目第一个字节 由3部分组成：项目类型（item type）、项目标签（item tag）、项目长度（item size）



   如input 类型的项目描述符原始数据为81 06，我们解析第一个字节81，将其转为二进制并分为三个部分 1000 00 01

   bSize 占用两位，定义data部分的长度，如上01 data长度为1个字节

   bType 占用两位，为0、1、2时分别为 Main、Global、Local类型

   bTag 占用四位，项目标签，如上的input标签

4. **项目分类**

   报表项目有Main、Global和Local三大类，每个类都有多个不同项目，实现不同描述

- Main

  Main类项目用于定义描述符中的数据项，也可以组合其中若干个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。带数据的Main用于生成报表中的数据项，包括input、output、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。

- Global

  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类项目对后续的所有项目有效，除非遇到新的Global类项目。

- Local

  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，所以在每一个Main项目前可能有多个Local项目。Local项目用于描述后面的input、output和feature项目。

下面列出全部项目和简要说明

项目类型	项目标志（Tag）	项目前缀，nn为数据长度	功能说明
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居存储空间。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 – 1111 01 nn	保留
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

实例解析：

Interface 1 HID Report Descriptor Consumer Control

Item Tag (Value)	Raw Data
Usage Page (Consumer Devices)	05 0C
Usage (Consumer Control)	09 01
Collection (Application)	A1 01
Report ID (1)	85 01
Usage (Volume)	09 E0
Logical Minimum (-24)	15 E8
Logical Maximum (24)	25 18
Report Size (7)	75 07
Report Count (1)	95 01
Input (Data,Var,Rel,NWrp,Lin,Pref,NNul,Bit)	81 06
Logical Minimum (0)	15 00
Logical Maximum (1)	25 01
Report Size (1)	75 01
Usage (Mute)	09 E2
Input (Data,Var,Rel,NWrp,Lin,Pref,NNul,Bit)	81 06
End Collection	C0

注意观察Usage Page项raw data为05 0C，解析第一个字节，根据上方表格Usage Page前缀 0000 01nn，将05转为二进制0000 0101，因此可得data长度为1个字节；

下面再举一个两字节的例子：

```
db 06h, A0h, FFh   ; Usage Page(Vendor defined)  定义设备功能
```

raw data：06 a0 ff。将06转为0000 0110 可得data为两字节，a0 ff 为data。

5. **collection 和end collection**

- 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。如下图表示application类型集合：

- 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。在设备报告多个传感器的位置的时候，使用 PhysicalCollection 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。

6. **Input、output、feature**
    - input、output、feature这三个都是用来定义报表中的数据字段
    - input，设备向主机发送消息，使用中断传输的方式输入报表
    - output，主机向设备发送消息，如果有中断传输管道，HID1.1兼容主机使用中断传输来发送报表，否则使用Set_Report控制请求。
    - feature 主机向设备发送，或主机向设备读取信息，主机使用Set_Report和Get_Report请求传输与接收。
    - 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。下图是关于这三个项目的数据定义：

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：报告全部控制的状态。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：报告作用中的控制。如在键盘报表中只报告按下的键的编号，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。游戏杆通常是报告绝对数据（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。鼠标通常是报告相对数据（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。

- 实例解析

Interface 3 HID Report Descriptor Headset

Item Tag (Value)	Raw Data	
Usage Page (Telephony Devices)	05 0B	
Usage (Headset)	09 05	
Collection (Application)	A1 01	
Report ID (2)	85 02	
Usage Page (Telephony Devices)	05 0B	
Logical Minimum (0)	15 00	
Logical Maximum (1)	25 01	
Usage (Hook Switch)	09 20	
Usage (Line BusyTone)	09 97	
Report Size (1)	75 01	
Report Count (2)	95 02	
Input (Data,Var,Abs,NWrp,Lin,NPrf,NNul,Bit)	81 22	← i1
Usage (Phone Mute)	09 2F	
Usage (Flash)	09 21	
Usage (Redial)	09 24	
Usage (Speed Dial)	09 50	
Report Size (1)	75 01	
Report Count (4)	95 04	
Input (Data,Var,Rel,NWrp,Lin,Pref,NNul,Bit)	81 06	← i2
Usage (Programmable Button)	09 07	
Usage Page (Button)	05 09	
Report Size (1)	75 01	
Report Count (1)	95 01	
Input (Data,Var,Abs,NWrp,Lin,Pref,NNul,Bit)	81 02	← i3
Usage Page (Telephony Devices)	05 0B	
Usage (Telephony Key Pad)	09 06	
Collection (Logical)	A1 02	
Usage Minimum (Phone Key 0)	19 B0	
Usage Maximum (Phone Key Pound)	29 BB	
Logical Minimum (0)	15 00	
Logical Maximum (11)	25 0B	
Report Size (4)	75 04	
Report Count (1)	95 01	
Input (Data,Ary,Abs)	81 00	
End Collection	C0	
Report Size (1)	75 01	
Report Count (5)	95 05	
Input (Cnst,Ary,Abs)	81 01	
Usage Page (LEDs)	05 08	
Logical Minimum (0)	15 00	
Logical Maximum (1)	25 01	
Usage (Off-Hook)	09 17	
Usage (Mute)	09 09	
Usage (Ring)	09 18	
Usage (Hold)	09 20	
Usage (Microphone)	09 21	
Report Size (1)	75 01	
Report Count (5)	95 05	
Output (Data,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Bit)	91 02	← o1
Report Size (1)	75 01	
Report Count (3)	95 03	
Output (Cnst,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Bit)	91 03	← o2
End Collection	C0	
Usage Page (Vendor-Defined 49)	06 30 FF	
Usage (Vendor-Defined 5)	09 05	
Collection (Application)	A1 01	

input类型的i1的raw data为81 22，第一个字节81表示的是项目类型、项目标签和数据长度，第二个字节为22表示数据，此处我们解析22，22转为二进制 0010 0010，再根据上方表格查出每位含义（注意第input类型七位不使用）：data，variable，

absolute，no wrap，linear，preferred，no null position，bit field。也由此可见usb lyzer已经为我们转译好每个位的含义。

再解析一下output类型的o1,data为91 02:，02（这里应该是省略了一个00字节，因为有九位）转为二进制0 0000 0010对应的就是：data，variable，no wrap，linear，preferred，no null position，non-volatile，bit field。

7. **Usage Page 和Usage**

- Usage Page项目的数据分为1~2字节，目前定义的都是一个字节。Usage Page 定义了常用的设备功能，关于Usage Page更多可以查阅 https://www.usb.org/sites/default/files/hut1_22.pdf
- 下面是HID Usage Tables对Usage Page的定义：

Page ID	Page Name	Section or Document
00	*Undefined*	
01	Generic Desktop Page (0x01)	4
02	Simulation Controls Page (0x02)	5
03	VR Controls Page (0x03)	6
04	Sport Controls Page (0x04)	7
05	Game Controls Page (0x05)	8
06	Generic Device Controls Page (0x06)	9
07	Keyboard/Keypad Page (0x07)	10
08	LED Page (0x08)	11
09	Button Page (0x09)	12
0A	Ordinal Page (0x0A)	13
0B	Telephony Device Page (0x0B)	14
0C	Consumer Page (0x0C)	15
0D	Digitizers Page (0x0D)	16
0E	Haptics Page (0x0E)	17
0F	PID Page	USB Physical Interface Device definitions for force feedback and related devices.
10	Unicode Page (0x10)	18
11-11	*Reserved*	
12	Eye and Head Trackers Page (0x12)	19
13-13	*Reserved*	
14	Auxiliary Display Page (0x14)	20
15-1F	*Reserved*	
20	Sensors Page (0x20)	21
21-3F	*Reserved*	
40	Medical Instrument Page (0x40)	22
41	Braille Display Page (0x41)	23
42-58	*Reserved*	
59	Lighting And Illumination Page (0x59)	24
5A-7F	*Reserved*	
80-83	Monitor Pages	USB Device Class Definition for Monitor Devices
84-87	Power Pages	USB Device Class Definition for Power Devices
88-8B	*Reserved*	
8C	Bar Code Scanner page	USB Device Class Definition for Point of Sale Devices

8D	Scale page	USB Device Class Definition for Point of Sale Devices
8E	Magnetic Stripe Reading (MSR) Devices	USB Device Class Definition for Point of Sale Devices
8F	Reserved Point of Sale pages	USB Device Class Definition for Point of Sale Devices
90	Camera Control Page (0x90)	25
91	Arcade Page	OAAF Definitions for arcade and coinop related Devices
92	Gaming Device Page (0x92)	26
93-F1CF	*Reserved*	
F1D0	FIDO Alliance Page (0xF1D0)	27
F1D1-FEFF	*Reserved*	
FF00-FFFF	Vendor-defined	

- Usage Page子级定义了更为详细的功能，如01 Generic Desktop Page

Generic Desktop Page (0x01)

Usage ID	Usage Name	Usage Types	Section
00	*Undefined*		
01	**Pointer**	CP	4.1
02	**Mouse**	CA	4.1
03-03	*Reserved*		
04	**Joystick**	CA	4.1
05	**Gamepad**	CA	4.1
06	**Keyboard**	CA	4.1
07	**Keypad**	CA	4.1
08	**Multi-axis Controller**	CA	4.1
09	**Tablet PC System Controls**	CA	4.1
0A	**Water Cooling Device** [6]	CA	4.1
0B	**Computer Chassis Device** [6]	CA	4.1
0C	**Wireless Radio Controls** [13]	CA	4.1
0D	**Portable Device Control** [23]	CA	4.1
0E	**System Multi-Axis Controller** [33]	CA	4.1
0F	**Spatial Controller** [39]	CA	4.1
10	**Assistive Control** [49]	CA	4.1
11	**Device Dock** [57]	CA	4.15
12	**Dockable Device** [57]	CA	4.15
13-2F	*Reserved*		
30	X	DV	4.2
31	Y	DV	4.2
32	Z	DV	4.2
33	Rx	DV	4.2
34	Ry	DV	4.2
35	Rz	DV	4.2
36	Slider	DV	4.3
37	Dial	DV	4.3
38	Wheel	DV	4.3
39	Hat Switch	DV	4.3
3A	**Counted Buffer**	CL	4.6
3B	Byte Count	DV	4.6
3C	Motion Wakeup	OSC/DF	4.3
3D	Start	OOC	4.3
3E	Select	OOC	4.3
3F-3F	*Reserved*		
40	Vx	DV	4.4
41	Vy	DV	4.4

这里只截取了一小部分，更多内容需要查询文档。可以看出这个page定义了鼠标键盘之类的功能，点击section字段蓝色链接可以看到详细描述，如下图：

4.1 Application Usages

Usage Name	Usage Type	Description
Pointer	CP	A collection of axes that generates a value to direct, indicate, or point user intentions to an application.
Mouse	CA	A hand-held, button-activated input device that when rolled along a flat surface, directs an indicator to move correspondingly about a computer screen, allowing the operator to move the indicator freely in select operations or to manipulate text or graphics. A mouse typically consists of two axes (X and Y) and one, two, or three buttons.
Joystick	CA	A manual control or cursor device. A joystick minimally consists of two variable axes (X and Y) and two buttons. A joystick is typically a rotational motion sensor. However, for legacy reasons, it is defined using linear axes. Traditionally, a joystick driver applies its own scaling to values returned from a joystick. That is, the driver simply linearizes and translates the range of values generated by the stick into normalized values between 0 and 64K, where 32K is centered. The application (game) then interprets the normalized values as necessary. Because of this, joysticks normally do not declare **Units** or **Physical Minimum** and **Physical Maximum** values for their axes. Depending on the driver, these items may be ignored if they are declared.
Gamepad	CA	A manual control or cursor device. A game pad minimally consists of a thumb-activated rocker switch that controls two axes (X and Y) and has four buttons. The rocker switch consists of four contact closures for up, down, right, and left.
Keyboard	CA	The primary computer input device. A **Keyboard** minimally consists of 103 buttons as defined by the **Boot Keyboard** definition.
Keypad	CA	Any keyboard configuration that does not meet the minimum requirements of the **Boot Keyboard**. **Keypad** often refers to a supplementary calculator-style keyboard.
Multi-axis Controller	CA	An input device used to orient eyepoints and or objects in 3 dimensional space. A Multi-axis Controller typically consists of six, variable axes (X, Y, Z, Rx, Ry and Rz) and is used by CAD/digital content creation applications for model manipulation and visualization in 3D space. The device may incorporate zero or more buttons.
Tablet PC System Controls	CA	System controls on Tablet PCs. This collection is not intended to contain display or audio data nor touchscreen input. Appropriate controls might be buttons, wheels, or simple indicators. This collection is intended to be opened by the operating system in exclusive mode and is not meant for application developers to open directly.
Water Cooling Device	CA	A collection of sensors and controls that represents a device using liquid to provide cooling of a thermal environment. A water cooling device contains at least one thermal reporting control.
Computer Chassis Device	CA	A collection of usages that represent data about the condition, state, and controls of sensors and devices attached to a chassis containing the motherboard and associated components (e.g., processor, graphics controller, hard drives) of a computing device.
Wireless Radio Controls	CA	A collection of buttons or switches that enable all-wireless radio communication to be turned on/off.
Portable Device Control	CA	A collection of controls on the portable devices, for example, volume controls, rotation lock, power, camera controls, home button, etc.

注意这里的Usage Type，可以在文档的3.4章节查询，如图：

如Usage Type为CA，到3.4.3.2查询详细描述：

3.4.3.2 Collection Application (CA)

The Collection Application Usage type identifies Usages that are used only in application-level collections. An application collection identifies a HID device or a functional subset of a complex device. An operating system uses the Usage associated with this collection to link the device to its controlling application or driver. Common examples are a keyboard or mouse. A keyboard with an integrated pointing device could contain two different application collections.

*Note: **Data** reports cannot span application collections.*

- 实例解析

Interface 3 HID Report Descriptor Headset

Item Tag (Value)	Raw Data
Usage Page (Telephony Devices)	05 0B
Usage (Headset)	09 05
Collection (Application)	A1 01
Report ID (2)	85 02
Usage Page (Telephony Devices)	05 0B

如图05 0B，usb lyzer已经帮我们解析好了Usage Page是Telephony Devices。

如果是自己查询可以按如下步骤：第一个字节表示item类型标签和长度，第二字节0B表示的是哪个Page查询文档

再根据Usage项查询具体某个功能，例子中的Usage为09 05：

14 Telephony Device Page (0x0B)

This usage page defines the keytop and control usages for telephony devices. Note that in many cases usage definitions are intentionally vague, this is because it is assumed that the controls are interpreted by the telephone software application (PBX). For instance, one software implementation may allow the Park usage to hold the line open while waiting for the target number to go on-hook, while another implementation will allow the user to hang up and then ring the user back when the target number is available. Often recommendations are made so that users of USB telephones see consistent interfaces across multiple vendors, minimizing learning curves and frustration when dealing with new or multiple systems.

Indicators on a phone are handled by wrapping them in LED: **Usage In Use Indicator** and LED: **Usage Selected Indicator** usages. For example, a message-indicator LED would be identified by a Telephony: Message usage declared as a **Feature** or **Output** in a LED: **Usage In Use Indicator** collection.

Usage ID	Usage Name	Usage Types	Section
00	*Undefined*		
01	**Phone**	CA	14.1
02	**Answering Machine**	CA	14.1
03	**Message Controls**	CL	14.1
04	**Handset**	CL	14.1
05	**Headset**	CL	14.1
06	**Telephony Key Pad**	NAry	14.2
07	**Programmable Button**	NAry	14.2
08-1F	*Reserved*		
20	Hook Switch	OOC	14.3
21	Flash	MC	14.3
22	Feature	OSC	14.3
23	Hold	OOC	14.3
24	Redial	OSC	14.3
25	Transfer	OSC	14.3
26	Drop	OSC	14.3
27	Park	OOC	14.3
28	Forward Calls	OOC	14.3
29	Alternate Function	MC	14.3
2A	**Line**	OSC/NAry	14.3
2B	Speaker Phone	OOC	14.3
2C	Conference	OOC	14.3
2D	Ring Enable	OOC	14.3
2E	Ring Select	OSC	14.3
2F	Phone Mute	OOC	14.3

8. **Report ID**

- 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（以我的理解额 报告ID为0 就不要定义，定义报告ID的话就不要为0）。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。

9. **Report Size 和Report Count**
   - 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Size 和 Report Count 项目。
10. **Usage、Usage Minimun 和 Usage Maximum**
    - 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

      ```
      例如下面报表描述符的一个局部，报表含有两个输入字节，第一个的用法是x，第二个是y。
      Report Size (8)
      Report Count (2)
      Usage (x)
      Usage (y)
      Input (data,Variable,Absolute)
      ```

    - 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。

      ```
      例如下面报表包含16个字节输入数据，第一个字节用法x，
      第二个为y，剩余的14个字节为厂商定义的用法。
      ```

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

- 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。

```
例如在一个键盘描述符中定义的标准键盘左、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、alt和GUI键，从HID UsageTables文档中的第10节可以查询到关于键
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。以下是报告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)
```

再次声明：

本节转载于https://blog.csdn.net/Britripe/article/details/111662665，

标题：**HID报表描述符（目前最全的解析，也是USB最复杂的描述符）**

作者：_WindChimes

WebHID

1. **前言**

   WebHID功能于谷歌89版本正式启用，部署于navigator.hid。借此浏览器可以通过该API访问hid设备。本节将介绍如何使用以及webHID是如何将HID报告描述符进行转译的。

   webHID文档：https://wicg.github.io/webhid/，该标准目前还处于草案阶段，仅谷歌内核浏览器支持。

   demo实例详解：https://web.dev/hid/

获取各个厂商的VendorId以及productId：http://www.linux-usb.org/usb.ids

2. **API使用**

   a. 协议支持

      该api需要在https协议下才能访问，如果是本地开发可以使用localhost或file协议访问。

   b. API方法

      requestDevice()请求访问设备，会弹出窗口让用户选择一个设备，如下图：

      用户选择设备后，才可以使用getDevice方法获取设备，从而进行数据通信。
      requestDevice方法可传入过滤参数过滤设备，如下：

```
const { hid } = navigator
if(hid){
    const filters = [
        { vendorId: 0x0b0e },// 捷波朗
        {
           vendorId: 0x6993,
           usage: 0x0005,
           usagePage: 0x000B,
```

```
        }, // yealink
        { vendorId: 0x1377 } // 森海塞尔
     ];
    //请求设备
    const devices = await navigator.hid.requestDevice({
        filters: filter
    });
}
```

getDevices()获取设备，如果没有获得用户许可将得到一个空数组，如果此前用户许可过将会返回设备列表数组，代码示例：

```
const { hid } = navigator
if (hid) {
  (async () => {
      const filters = [
          { vendorId: 0x0b0e },// 捷波朗
          {
                vendorId: 0x6993,
                usage: 0x0005,
                usagePage: 0x000B,
           }, // yealink
           { vendorId: 0x1377 } // 森海塞尔
      ];
      // 获取设备
      const devices = await navigator.hid.getDevices();
      // 过滤设备
      const fDevices = devices.filter(item => item.vendorId === filters.vendorId)
      if (fDevices.lenght > 0) {
           console.log(fDevices);
      } else {
            // 当前没有设备，请求设备获得用户许可
            // ...
      }
  })();
}
```

获得设备后可以访问的方法：

open() 设置设备状态为打开，用于监听设备发来的report

close() 设置设备状态为关闭，不再监听设备发来的数据

sendReport() 发送outputReport给设备

sendFeatureReport() 发送FeatureReport给设备

receiveFeatureReport() 接收FeatureReport给设备

事件：

inputreport 监听报告输入

connect 与设备建立连接

disconnect 与设备断开连接

完整代码示例：

```
const { hid } = navigator
if (hid) {
    (async () => {
        // 过滤条件vendorId为产商id，这个id可以在 http://www.linux-usb.org/usb.ids 查到
        // 除了 vendorId 还可以使用 usage usagePage productId 作为过滤条件
        const filters = [
            { vendorId: 0x0b0e },// 捷波朗
            {
                vendorId: 0x6993,
                usage: 0x0005,
                usagePage: 0x000B,
                // productId:0xb009   // 还可以传产品id过滤某系产品型号
            }, // yealink
            { vendorId: 0x1377 } // 森海塞尔
        ];
        const devices = await navigator.hid.getDevices();
        const fDevices = devices.filter(item => item.vendorId === filters.vendorId)
        if (fDevices.lenght > 0) {
            console.log('fDevices', fDevices[0]);
        } else {
            initQreust(filters)
        }
    })();
}
// 请求设备
function initQreust(filter) {
    // 按钮，用户点击按钮后调用navigator.hid.requestDevice请求设备方法
    const ele = document.getElementById('button')
    let devices;
    ele.addEventListener('click', async () => {
        try {
            // 请求设备
            devices = await navigator.hid.requestDevice({
                filters: filter
            });
            const device = devices[0]
            // 获取到设备后，打开设备
            await device.open();
            // 监听设备发来的数据
            device.addEventListener("inputreport", (e) => {
                console.log('收到inputreport');
                console.table([
                    {
                        '型号': e.device.productName,
                        '报告id（reportId）': e.reportId,
                        '报告内容（十进制）': e.data.getUint8(0)
                    }
```

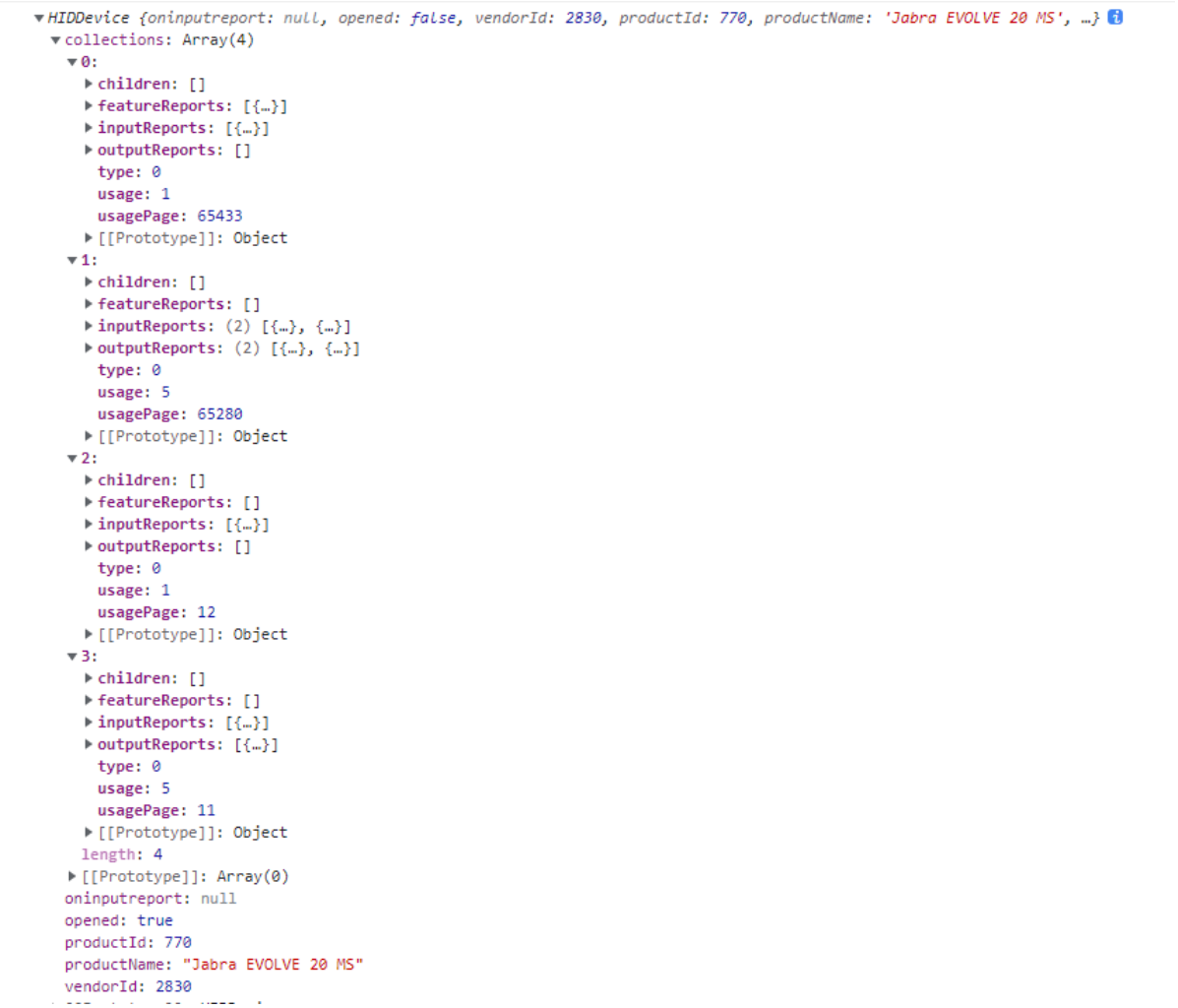
```
                ])
                console.log(new Uint8Array(e));
            });
            // 接收featureReport数据
            devices.receiveFeatureReport(0x02).then(data => {
                console.log('receiveFeatureReport data', data);

            })
            // 页面关闭时关闭设备
            window.addEventListener("onbeforeunload", async () => {
                await device.close();
            });
        } catch (error) {
            console.log("No device was selected.");
        }
    });
    // 页面发送按钮，点击后发送数据给设备
    const sendEle = document.getElementById('send')
    sendEle.addEventListener('click', async () => {
        const val = document.getElementById('opt').value
        const device = devices[0]
        const rumbleData = [val & 0xff]
        // 调用hid发送outputReport数据方法，数据为无符号整型数组
        await device.sendReport(0x02, new Uint8Array(rumbleData));
        console.log(`Send Report "${rumbleData}" To "${device.productName}" HID device`);
    })
}
```

3. **解析webHID**

   HID报告描述符会被浏览器转译，导致与原报文结构有些不同，下面介绍如何转译HID描述符。

   这是浏览器解析后的描述符，如图：



▼ 原报文，经过usb lyzer解析后的结果,设备同为Jabra EVOLVE 20 MS：

```
Interface 3 HID Report Descriptor Vendor-Defined 1
Item Tag (Value) Raw Data
Usage Page (Vendor-Defined 154) 06 99 FF
Usage (Vendor-Defined 1) 09 01
Collection (Application) A1 01
    Usage (Vendor-Defined 3) 09 03
    Report ID (154) 85 9A
    Logical Minimum (0) 15 00
    Logical Maximum (255) 26 FF 00
    Report Size (8) 75 08
    Report Count (63) 95 3F
    Feature (Data,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Buf) B2 02 01
    Usage (Vendor-Defined 4) 09 04
    Report ID (155) 85 9B
    Logical Minimum (0) 15 00
    Logical Maximum (1) 25 01
    Report Size (1) 75 01
    Report Count (1) 95 01
```

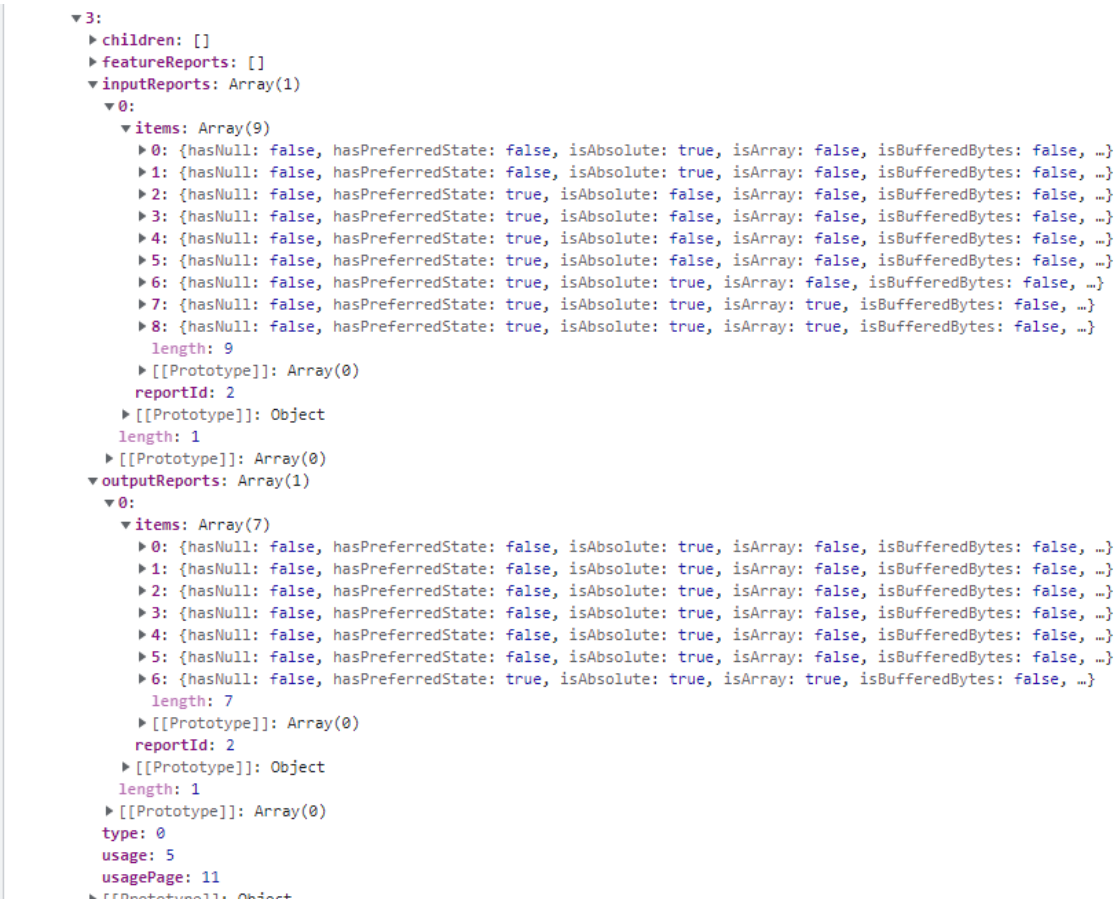
```
    Input (Data,Var,Rel,NWrp,Lin,Pref,NNul,Bit) 81 06
    Report Size (1) 75 01
    Report Count (15) 95 0F
    Input (Cnst,Ary,Abs) 81 01
End Collection C0
Usage Page (Telephony Devices) 05 0B
Usage (Headset) 09 05
Collection (Application) A1 01
    Report ID (2) 85 02
    Usage Page (Telephony Devices) 05 0B
    Logical Minimum (0) 15 00
    Logical Maximum (1) 25 01
    Usage (Hook Switch) 09 20
    Usage (Line BusyTone) 09 97
    Report Size (1) 75 01
    Report Count (2) 95 02
    Input (Data,Var,Abs,NWrp,Lin,NPrf,NNul,Bit) 81 22
    Usage (Phone Mute) 09 2F
    Usage (Flash) 09 21
    Usage (Redial) 09 24
    Usage (Speed Dial) 09 50
    Report Size (1) 75 01
    Report Count (4) 95 04
    Input (Data,Var,Rel,NWrp,Lin,Pref,NNul,Bit) 81 06
    Usage (Programmable Button) 09 07
    Usage Page (Button) 05 09
    Report Size (1) 75 01
    Report Count (1) 95 01
    Input (Data,Var,Abs,NWrp,Lin,Pref,NNul,Bit) 81 02
    Usage Page (Telephony Devices) 05 0B
    Usage (Telephony Key Pad) 09 06
    Collection (Logical) A1 02
        Usage Minimum (Phone Key 0) 19 B0
        Usage Maximum (Phone Key Pound) 29 BB
        Logical Minimum (0) 15 00
        Logical Maximum (11) 25 0B
        Report Size (4) 75 04
        Report Count (1) 95 01
        Input (Data,Ary,Abs) 81 00
    End Collection C0
    Report Size (1) 75 01
    Report Count (5) 95 05
    Input (Cnst,Ary,Abs) 81 01
    Usage Page (LEDs) 05 08
    Logical Minimum (0) 15 00
    Logical Maximum (1) 25 01
    Usage (Off-Hook) 09 17
    Usage (Mute) 09 09
    Usage (Ring) 09 18
    Usage (Hold) 09 20
    Usage (Microphone) 09 21
    Report Size (1) 75 01
    Report Count (5) 95 05
    Output (Data,Var,Abs,NWrp,Lin,NPrf,NNul,NVol,Bit) 91 22
    Usage Page (Telephony Devices) 05 0B
    Logical Minimum (0) 15 00
    Logical Maximum (1) 25 01
```

```
    Usage (Ringer) 09 9E
    Report Size (1) 75 01
    Report Count (1) 95 01
    Output (Data,Var,Abs,NWrp,Lin,NPrf,NNul,NVol,Bit) 91 22
    Report Size (1) 75 01
    Report Count (10) 95 0A
    Output (Cnst,Ary,Abs,NWrp,Lin,Pref,NNul,NVol,Bit) 91 01
End Collection C0
Usage Page (Vendor-Defined 1) 06 00 FF
Usage (Vendor-Defined 5) 09 05
Collection (Application) A1 01
    Report ID (4) 85 04
    Usage Page (Vendor-Defined 49) 06 30 FF
    Logical Minimum (0) 15 00
    Logical Maximum (1) 25 01
    Usage (Vendor-Defined 32) 09 20
    Usage (Vendor-Defined 151) 09 97
    Report Size (1) 75 01
    Report Count (2) 95 02
    Input (Cnst,Var,Abs,NWrp,Lin,NPrf,NNul,Bit) 81 23
    Usage (Vendor-Defined 47) 09 2F
    Usage (Vendor-Defined 33) 09 21
    Usage (Vendor-Defined 36) 09 24
    Usage (Vendor-Defined 65533) 0A FD FF
    Usage (Vendor-Defined 80) 09 50
    Report Size (1) 75 01
    Report Count (5) 95 05
    Input (Cnst,Var,Rel,NWrp,Lin,Pref,NNul,Bit) 81 07
    Usage (Vendor-Defined 6) 09 06
    Collection (Logical) A1 02
        Usage Minimum (Vendor-Defined 176) 19 B0
        Usage Maximum (Vendor-Defined 187) 29 BB
        Logical Minimum (0) 15 00
        Logical Maximum (12) 25 0C
        Report Size (4) 75 04
        Report Count (1) 95 01
        Input (Data,Ary,Abs) 81 00
    End Collection C0
    Report Size (1) 75 01
    Report Count (5) 95 05
    Input (Cnst,Ary,Abs) 81 01
    Usage Page (Vendor-Defined 65) 06 40 FF
    Logical Minimum (0) 15 00
    Logical Maximum (1) 25 01
    Usage (Vendor-Defined 23) 09 17
    Usage (Vendor-Defined 9) 09 09
    Usage (Vendor-Defined 24) 09 18
    Usage (Vendor-Defined 32) 09 20
    Usage (Vendor-Defined 33) 09 21
    Report Size (1) 75 01
    Report Count (5) 95 05
    Output (Data,Var,Abs,NWrp,Lin,NPrf,NNul,NVol,Bit) 91 22
    Usage Page (Vendor-Defined 49) 06 30 FF
    Logical Minimum (0) 15 00
    Logical Maximum (1) 25 01
    Usage (Vendor-Defined 158) 09 9E
    Report Size (1) 75 01
```

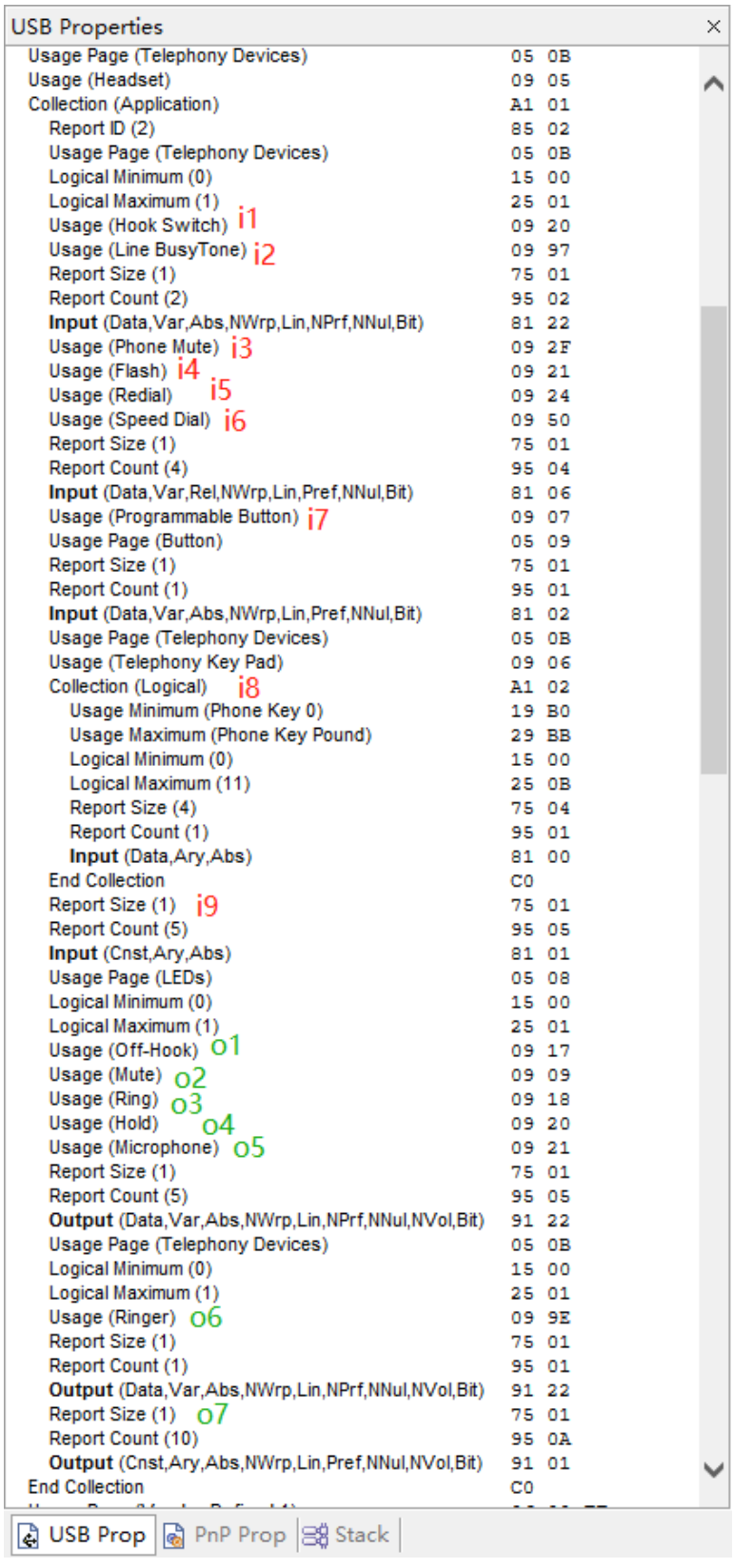
```
    Report Count (1) 95 01
    Output (Data,Var,Abs,NWrp,Lin,NPrf,NNul,NVol,Bit) 91 22
    Report Size (1) 75 01
    Report Count (10) 95 0A
    Output (Cnst,Ary,Abs,NWrp,Lin,Pref,NNul,NVol,Bit) 91 01
    Report ID (5) 85 05
    Usage Page (Vendor-Defined 1) 06 00 FF
    Usage (Vendor-Defined 1) 09 01
    Logical Minimum (0) 15 00
    Logical Maximum (255) 26 FF 00
    Report Size (8) 75 08
    Report Count (32) 95 20
    Output (Data,Var,Abs,NWrp,Lin,Pref,NNul,NVol,Buf) 92 02 01
    Usage (Vendor-Defined 1) 09 01
    Logical Minimum (0) 15 00
    Logical Maximum (255) 26 FF 00
    Report Size (8) 75 08
    Report Count (32) 95 20
    Input (Data,Var,Abs,NWrp,Lin,Pref,NNul,Buf) 82 02 01
End Collection C0
Usage Page (Consumer Devices) 05 0C
Usage (Consumer Control) 09 01
Collection (Application) A1 01
    Report ID (1) 85 01
    Usage Page (Consumer Devices) 05 0C
    Logical Minimum (0) 15 00
    Logical Maximum (1) 25 01
    Usage (Volume Decrement) 09 EA
    Usage (Volume Increment) 09 E9
    Report Size (1) 75 01
    Report Count (2) 95 02
    Input (Data,Var,Abs,NWrp,Lin,Pref,NNul,Bit) 81 02
    Report Size (1) 75 01
    Report Count (14) 95 0E
    Input (Cnst,Ary,Abs) 81 01
End Collection C0
```

经过比对两者都有四个集合，根据usage 和 usagePage属性（浏览器将描述符数据转为了十进制，查询HID Usage Table时需要转为十六进制）可以查出四个集合的Usage Page含义，分别为：Vendor-Defined，Vendor-Defined ，Consumer Devices，Telephony Devices。也由此可以看出浏览器将集合首先转换为数组。

本文重点是call control，所以将重点关注Telephony Devices 的Usage Page，打开集合最后一个元素，如图：



可以看到inputReports有9个元素，outputReports有7个元素，这些元素每一个都对应了一个usage，结合usb lyzer报文：



注意i9和o7没有usage，浏览器会将其usage置空，如图：

接下来解析数据的每一位含义：

打开第一个inputReport查询usages值，将720928转为十六进制为b 0020，第一个字节为usage Page后面16位为Usage ID，查表可得hook switch功能

```
▼0:
    hasNull: false
    hasPreferredState: false
    isAbsolute: true
    isArray: false
    isBufferedBytes: false
    isConstant: false
    isLinear: true
    isRange: false
    isVolatile: false
    logicalMaximum: 0
    logicalMinimum: 0
    physicalMaximum: 0
    physicalMinimum: 0
    reportCount: 1
    reportSize: 1
    unitExponent: 0
    unitFactorCurrentExponent: 0
    unitFactorLengthExponent: 0
    unitFactorLuminousIntensityExponent: 0
    unitFactorMassExponent: 0
    unitFactorTemperatureExponent: 0
    unitFactorTimeExponent: 0
    unitSystem: "none"
  ▶ usages: [720928]
    wrap: false
  ▶ [[Prototype]]: Object
```


14 Telephony Device Page (0x0B)

This usage page defines the keytop and control usages for telephony devices. Note that in many cases usage definitions are intentionally vague, this is because it is assumed that the controls are interpreted by the telephone software application (PBX). For instance, one software implementation may allow the Park usage to hold the line open while waiting for the target number to go on-hook, while another implementation will allow the user to hang up and then ring the user back when the target number is available. Often recommendations are made so that users of USB telephones see consistent interfaces across multiple vendors, minimizing learning curves and frustration when dealing with new or multiple systems.

Indicators on a phone are handled by wrapping them in LED: **Usage In Use Indicator** and LED: **Usage Selected Indicator** usages. For example, a message-indicator LED would be identified by a Telephony: Message usage declared as a **Feature** or **Output** in a LED: **Usage In Use Indicator** collection.

Usage ID	Usage Name	Usage Types	Section
00	*Undefined*		
01	**Phone**	CA	14.1
02	**Answering Machine**	CA	14.1
03	**Message Controls**	CL	14.1
04	**Handset**	CL	14.1
05	**Headset**	CL	14.1
06	**Telephony Key Pad**	NAry	14.2
07	**Programmable Button**	NAry	14.2
08-1F	*Reserved*		
20	Hook Switch	OOC	14.3
21	Flash	MC	14.3
22	Feature	OSC	14.3
23	Hold	OOC	14.3
24	Redial	OSC	14.3

再根据reportSize为1，ReportCount为1，hook switch功能只占用一个bit，在inputReport发来的数据中的第一位表示，而hook switch的Usage Type为OOC也就是0代表关，1代表开。

后面几位也如法炮制，即可解析出inputReport传来数据的含义。

如web收到消息如下：

（索引）	型号	报告id（reportId）	报告内容（十进制）
0	'Yealink UH36'	2	2

收到02，结合描述符可知设备line BusyTone开启。

（索引）	型号	报告id（reportId）	报告内容（十进制）
0	'Yealink UH36'	2	3

收到03，设备Hook Switch与Line BusyTone开启。

（索引）	型号	报告id（reportId）	报告内容（十进制）
0	'Yealink UH36'	2	5

收到05，设备Mute和Hook Switch开启。

同理outputReport也是如此解析，发送4（0100）为响铃，发送1（0001）offHook...

Call Control

1. **概念**

call control指通话线控，基本功能有：on-hook（挂断）/off-hook(接听)、mute/unmute、ring/unring、hold/flash等，越高级的耳机功能越多，如上图最左边那款带显示屏的还能支持拨号功能。

本文只实现on-hook（挂断）/off-hook(接听)、mute/unmute、ring/unring、hold/flash这四个功能。这四个功能几乎所有耳机都支持。

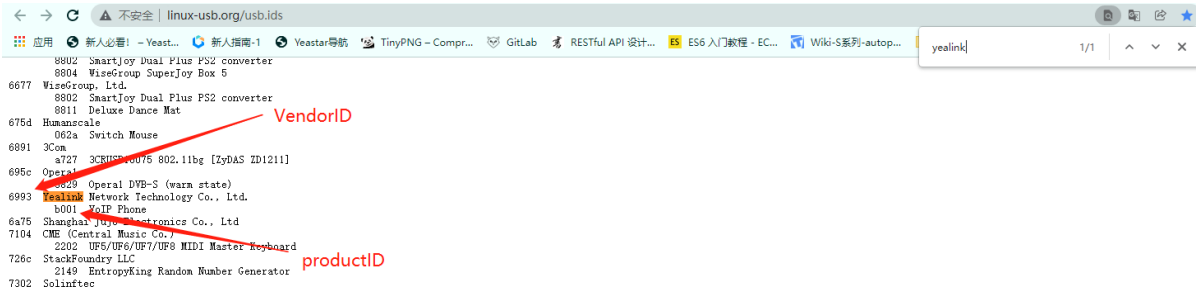
2. **VendorID限制设备厂商**

   为了精准过滤非耳机设备，我们需要给requestDevice方法设置过滤条件，即通过VendorId限制只显示某些耳机厂商的设备。

   下图为不限制VendorID，会访问到USB Keyboard设备：

   下图是限制了VendorID

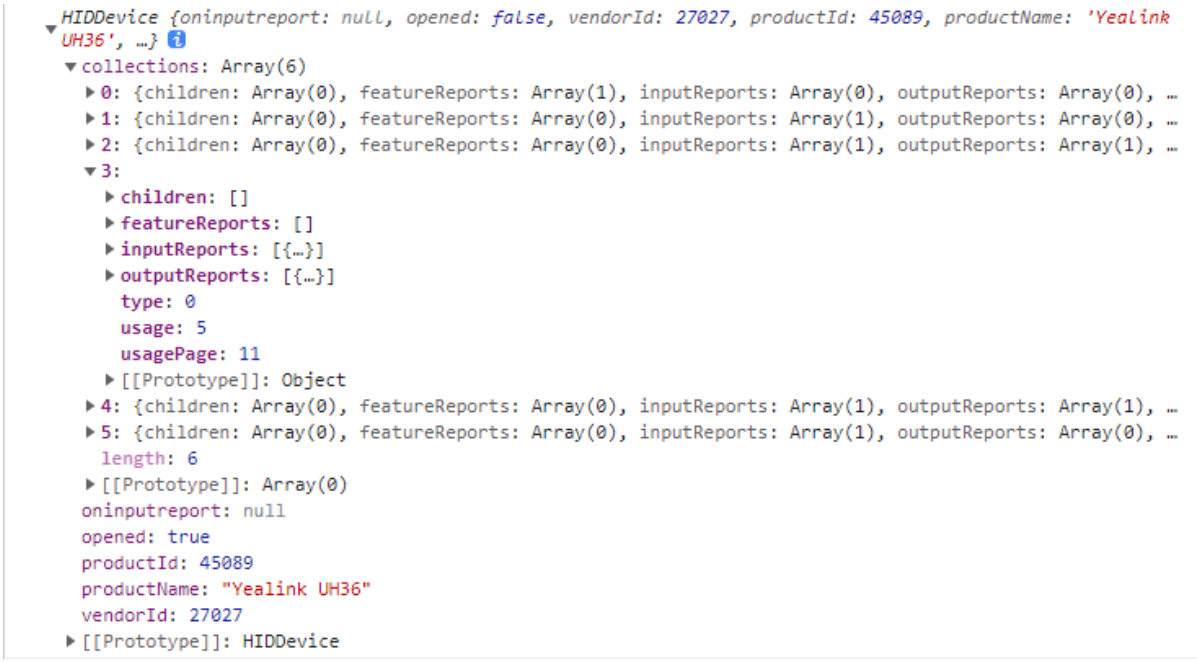
VendorID通过http://www.linux-usb.org/usb.ids可以获得，如查询亿联VendorID



此外还可以在过滤条件增加productId属性限制某些型号，以及UsagePage和Usage限制带有哪些功能的设备才可以被访问。

3. **获取Telephony Device Page集合**

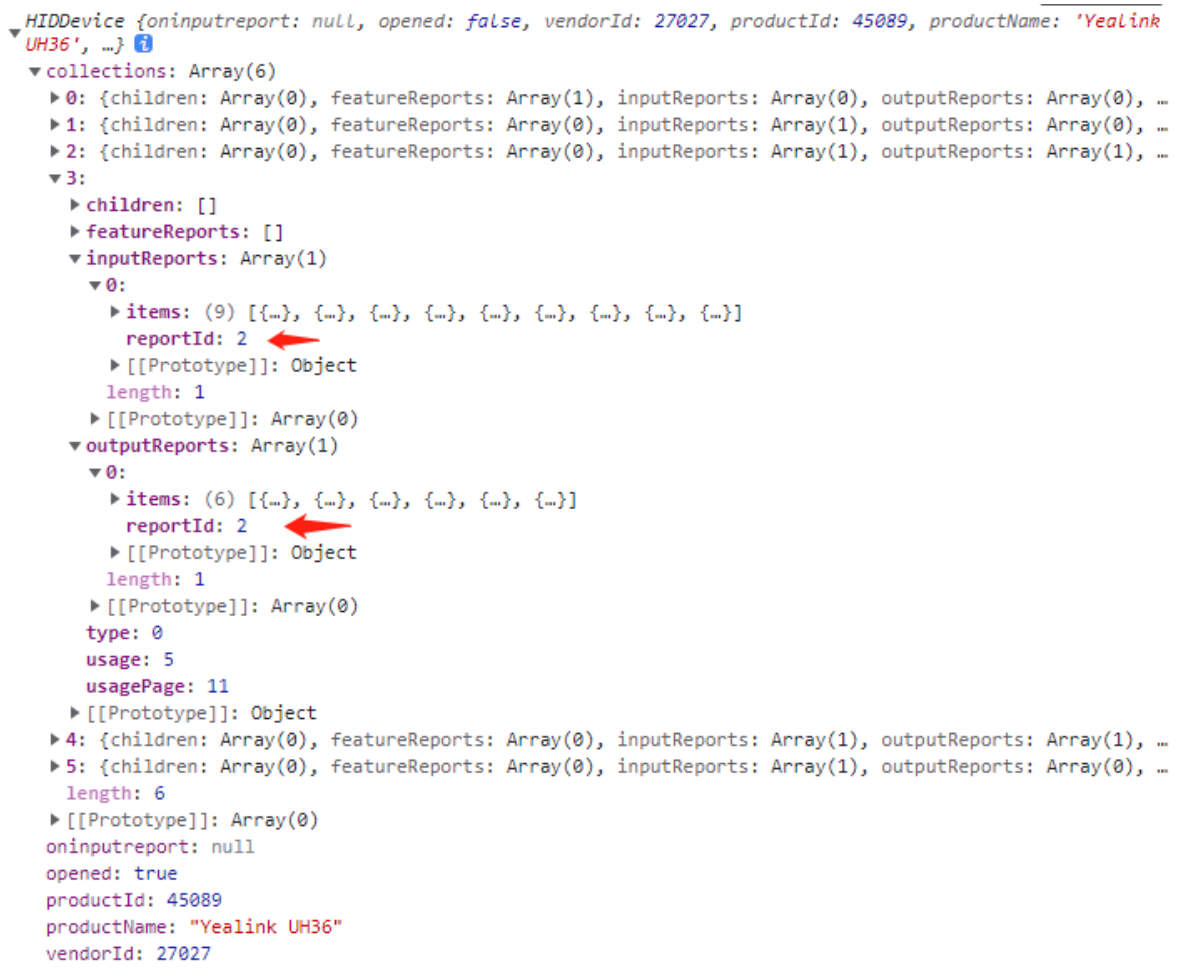
   浏览器获得设备后，可以获取设备描述符，过滤出UsagePage为11的元素，11为0xb即Telephony Device Page



通过该UsagePage集合定义了耳机onHook/offHook、mute/unmute、ring/unring、hold/flash

4. **获取ReportID**

   在获取Telephony Device Page后即可在inputReports和outputReports属性下取得reportId

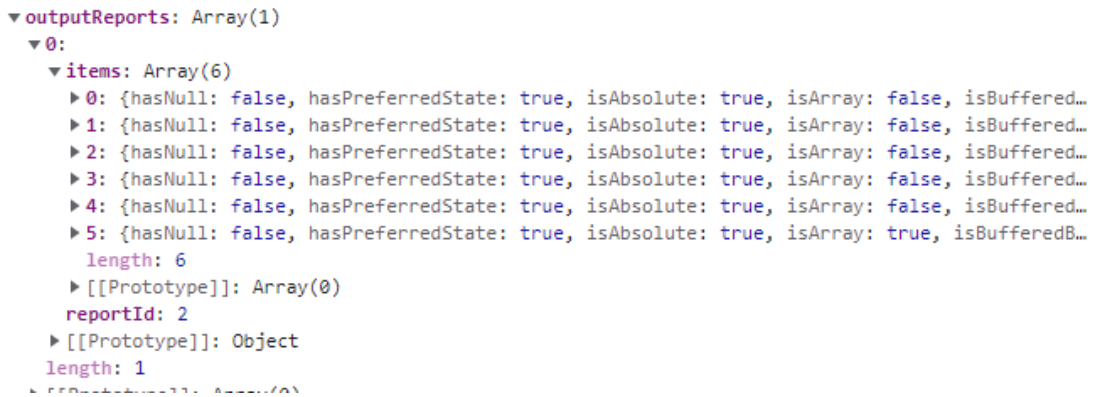


通过这个reportId才能修改对应的usage状态

```
// 第一个参数为reportId，不同厂商的reportId不一样，所以需要动态获取
device.sendReport(0x02, new Uint8Array(rumbleData));
```

5. **解析描述符inputReport和outputReport每个bit功能**

- outputReport指令集成



设置Usages常量，使用枚举类型存储，如下：

```
enum OutputUsagesEnum {
  OFF_HOOK = 524311, // 接听按键LED灯状态，主机向耳机发送接听指令其实是点亮LED灯
  MUTE = 524297,
  RING = 524312,
  HOLD = 524320
}
```

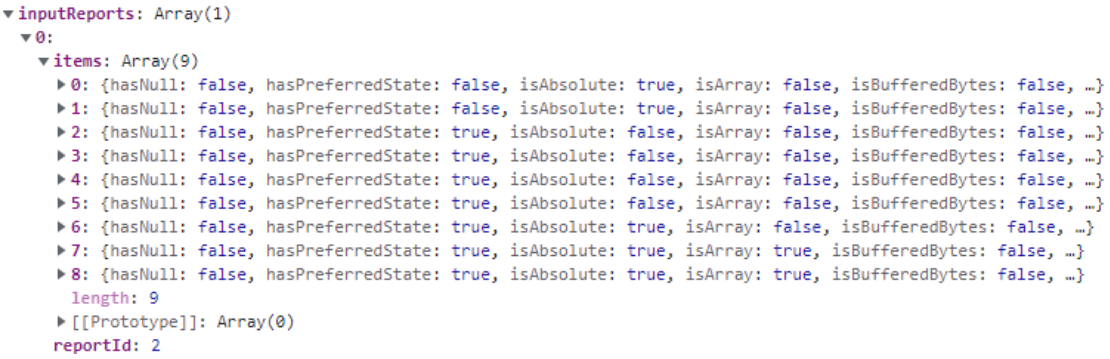
循环遍历items数组，获取outputReport bit个数及usage，如下：

```
// 假设获取到的outputReports数组,为了简便忽略了其他属性
const items = [
  {reportCount:1,reportSize:1,usages:[524311]},
  {reportCount:1,reportSize:1,usages:[524297]},
  {reportCount:1,reportSize:1,usages:[524312]},
  {reportCount:1,reportSize:1,usages:[524320]},
  {reportCount:1,reportSize:1,usages:[524321]},
]
// 记录outputReport所有bit，元素为0不使用该元素
const outputBits = [];
items.forEach((item,i)=>{
  const {reportCount=0,reportSize=0,usages = []} = item;
  const len = reportCount * reportSize;
  const tempArr = new Array(len);
  switch(usages[0]){
    case OutputUsagesEnum.OFF_HOOK:
    case OutputUsagesEnum.MUTE:
    case OutputUsagesEnum.RING:
    case OutputUsagesEnum.HOLD:
      tempArr.fill(usages[0])
      break
    default:
      tempArr.fill(0)
      break
  }
  outputBits.push(...tempArr)
```

```
});
// 提取出outputBits后就可以根据该数组进行进制转换及指令切割等工作，本文不详细展开
```

- inputReport 数据解析

  inputReport解析同outputReport



  设置InputReport的Usage常量，同样使用枚举：

```
enum InputUsagesEnum {
  HOOK_SWITCH = 720928,
  LINE_BUSY_TONE = 721047,
  PHONE_MUTE = 720943,
  FLASH = 720929,
}
```

  循环遍历items数组，获取inputReport bit个数及usage，如下：

```
// 假设获取到的inputReports数组,为了简便忽略了其他属性
const items = [
  {reportCount:1,reportSize:1,usages:[720928]},
  {reportCount:1,reportSize:1,usages:[721047]},
  {reportCount:1,reportSize:1,usages:[720943]},
  {reportCount:1,reportSize:1,usages:[720929]},
  {reportCount:1,reportSize:1,usages:[720932]},
]
// 记录outputReport所有bit，元素为0不使用该元素
const inputBits = [];
items.forEach((item,i)=>{
  const {reportCount=0,reportSize=0,usages = []} = item;
  const len = reportCount * reportSize;
  const tempArr = new Array(len);
  switch(usages[0]){
    case InputUsagesEnum.HOOK_SWITCH:
    case InputUsagesEnum.LINE_BUSY_TONE:
```

```
    case InputUsagesEnum.PHONE_MUTE:
    case InputUsagesEnum.FLASH:
      tempArr.fill(usages[0])
      break
    default:
      tempArr.fill(0)
      break
  }
  inputBits.push(...tempArr)
});

// 通过inputBits数组解析设备发来的数据
// 假设设备监听inputReport方法如下
device.addEventListener('inputreport',(e)=>{
  // 获得十进制数据
  const data = e.data.getUint8(0);
  // 将数据转为二进制并剪切成数组
  const bits = parseInt(data).toString(2).split('');
  bits.forEach((item,i)=>{
    // 结合inputBits 获取当前bit的功能和状态
    switch(inputBits[i]){
        case InputUsagesEnum.HOOK_SWITCH:
          item ? 'offHook状态':'onHook状态';
          break
        case InputUsagesEnum.LINE_BUSY_TONE:
          item ? '占线状态':'空闲状态';
          break
        case InputUsagesEnum.PHONE_MUTE:
          item ? 'mute状态':'unmute状态';
          break
        case InputUsagesEnum.FLASH:
          item ? 'flash状态':'非flash状态';
          break
        default:
          break
    }
  })
})
```

6. **log记录**

   call control 额外要处理的功能，方便问题排查。

   需要记录一份完整的描述符。

   outputReport发送记录，包含reportId和data。

   inputReport 收到的数据，这里只记录usagePage集合下的reportId的数据，因为设备有可能会不断发一些没用的状态数据，所以做一层过滤。