STM32 USB初级培训.协议介绍

2016 Shanghai

Lilian YAO

life.augmented

Universal Serial Bus Specification

Compaq

Hewlett-Packard

Intel

Lucent

Microsoft

NEC

Philips

Revision 2.0

April 27, 2000

通用串行总线 USB 2.0

life.augmented

www.usb.org

- SuperSpeed USB
- USB Power Delivery
- USB Type-C™
- Hi-Speed USB
- USB On-The-Go and Embedded Host
- Tools
- USB-IF eStore
- Documents
  - USB 2.0 Documents
  - Wireless USB Documents
  - USB Device Class Specifications
  - Technical White Papers
  - Presentations / Other Documents
  - Archived Documents
- USB-IF Compliance Program
- USB FAQ
- Events
- Join USB-IF, Inc.
- Resources
- ExpressCard

Home | About USB-IF | Channel | Press | Developers | Members | Products

Home > Developers > **Documents**

USB 3.1 Specification

Universal Serial Bus Revision 3.1 Specification (.zip file format, size 56.6 MB) provides the technical details to understand USB 3.1 requirements and design USB 3.1 compatible products. Modifications to the USB 3.1 specification are made through Engineering Change Notices (ECNs). Enclosed in this zip file are the following documents:

Home | About USB-IF | Channel | Press | Developers | Members | Products

Home > Developers > Documents > **USB 2.0 Documents**

USB 2.0 Documents Index

USB 2.0 Specification
USB 2.0 Adopters Agreement
USB 2.0 On-The-Go Specification Supplement Adopters Agreement
Low Voltage Inter-Chip Adopters Agreement Amendment
High-Speed Inter-Chip Adopters Agreement Amendment
Intellectual Property Rights Notice
USB 2.0 Icon Design Guideline

Home | About USB-IF | Channel | Press | Developers | Members | Products

Home > Developers > Documents > **USB Device Class Specifications**

USB-IF Device Class Documents

The USB Device Working Group ("DWG") enables and promotes increased interoperability and reliability between USB products through documents that augment, enhance and extend USB specifications. USB-IF members may participate in the development efforts of a Working Group only after having executed the IP Agreement corresponding to the specification under development by that Group. If you are interested in working on a Device Class document, you will find the various IP Agreements and can submit your request to join in the members-only section of this site. The current set of Defined 1.0 Class Codes is available.

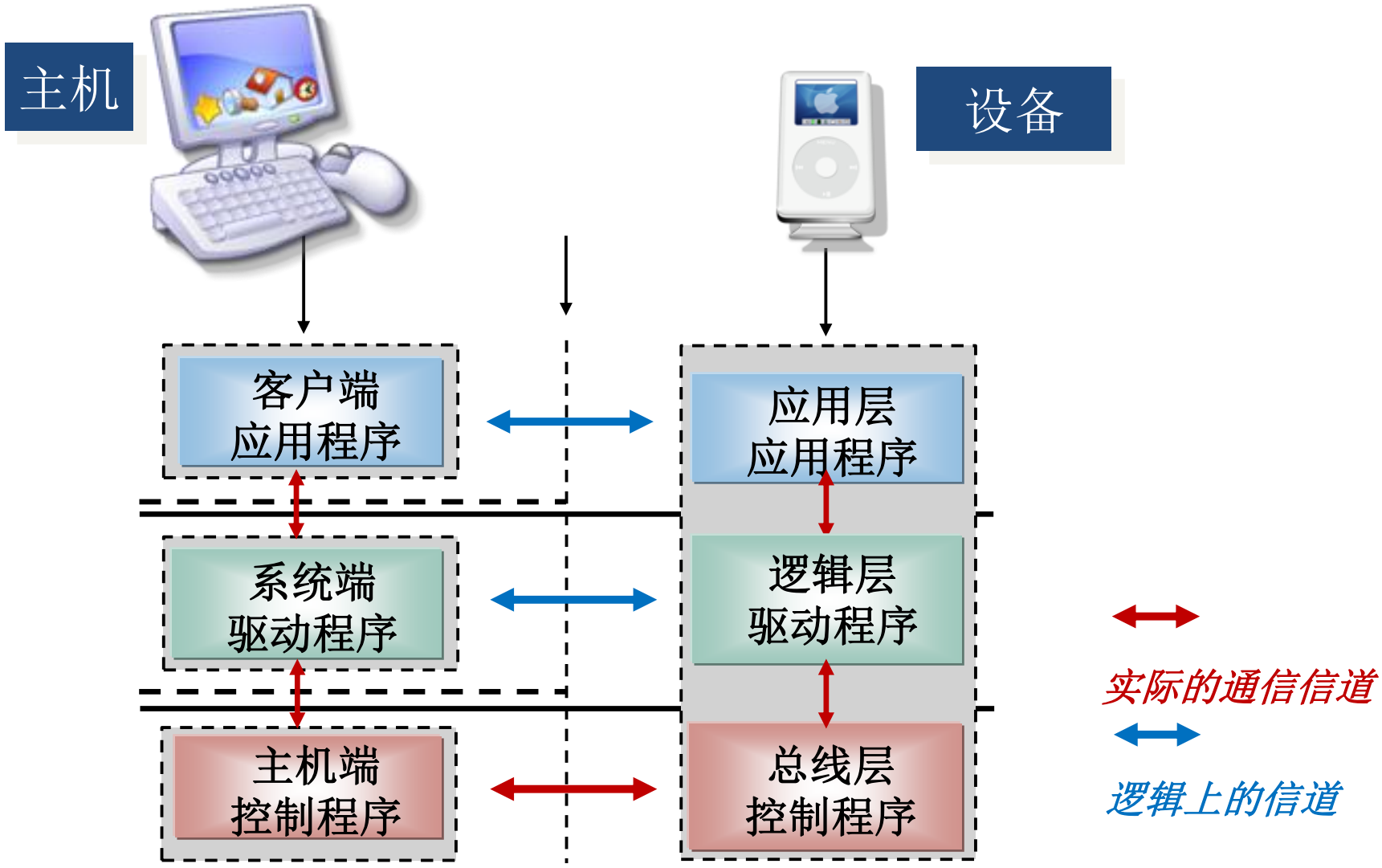
These specifications recommend design targets for classes of devices. For HID related information, please go to the HID Web page.

Approved Class Specification Documents

USB 应用范围

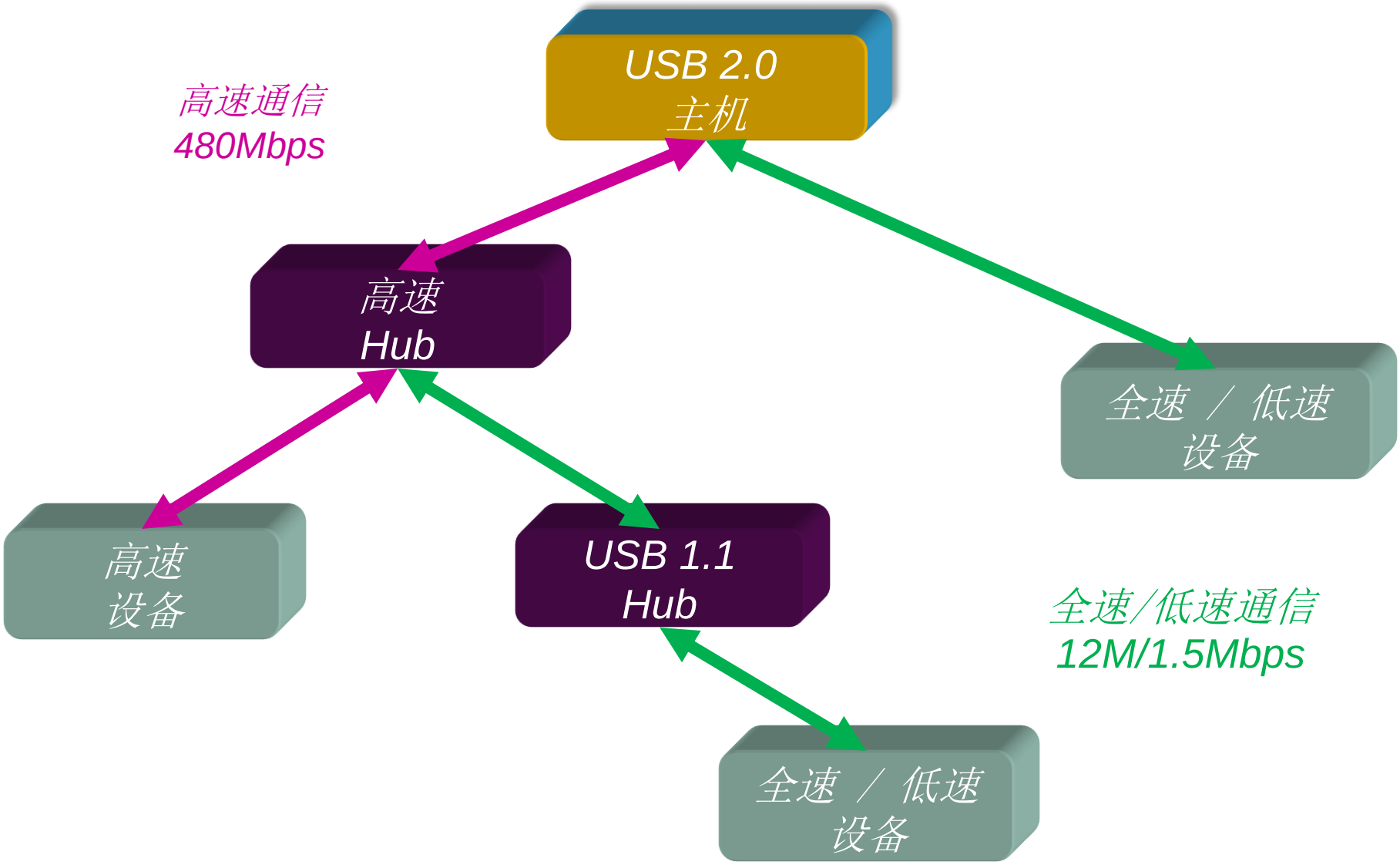
性能	应用	特性
低速(1.5Mbps): ✓交互式设备 ✓10-100kbps	➢键盘，鼠标 ➢手写笔 ➢游戏手柄 ➢虚拟设备 ➢外设	•极低的成本 •易于使用 •热插拔 •同时使用多个外设
全速(12Mbps): ✓电话，音频类 ✓压缩的视频类 ✓500kbps – 10Mbps	➢话音 ➢宽带 ➢音频 ➢麦克风	•较低的成本 •易于使用 •热插拔 •同时使用多个外设 •可保证的带宽 •可保证的延迟
高速(480Mbps): ✓视频，大容量存储 ✓25 – 400Mbps	➢视频 ➢大容量存储 ➢图像 ➢宽带	•低成本 •易于使用 •热插拔 •同时使用多个设备 •可保证的带宽 •可保证的延迟 •高带宽

USB 系统结构



life.augmented

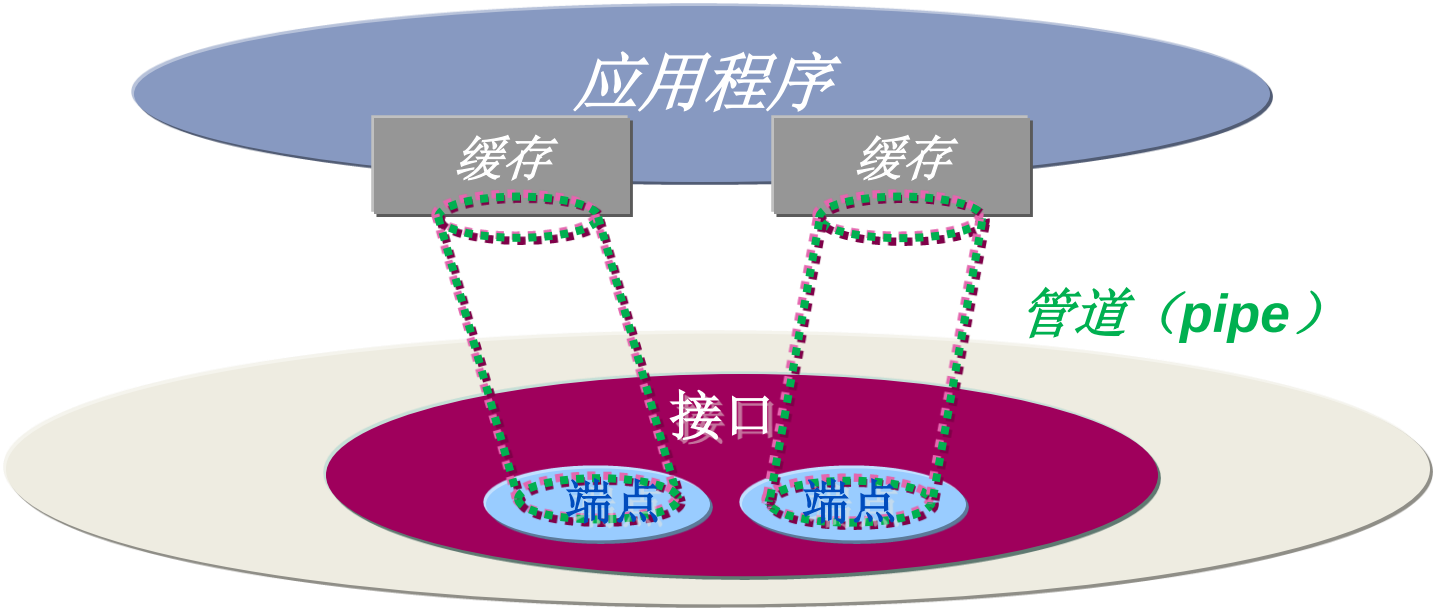
USB 拓扑结构示例



life.augmented

USB 逻辑部件

主机



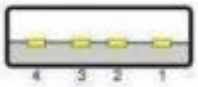
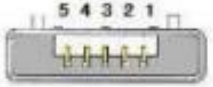
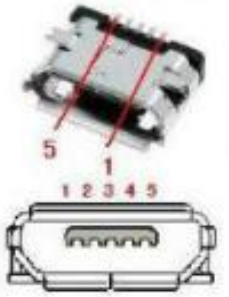
设备

接口(一个或多个)：用于描述特定功能，包含若干端点

端点（一个或多个）：传输的最终对象
>> 端点号、传输类型、
>> 传输方向、最大数据包长度…

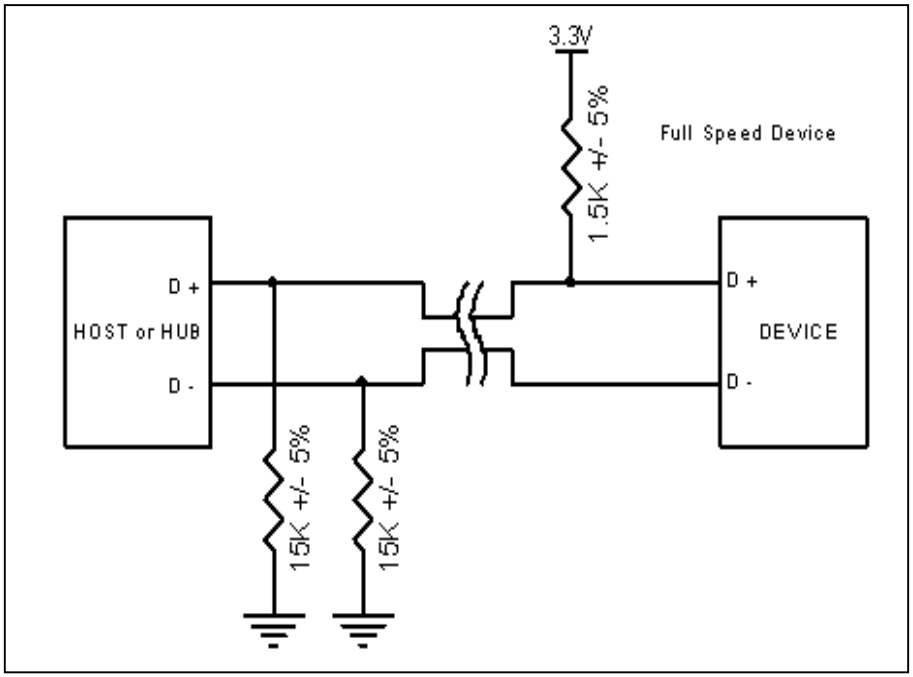
life.augmented

USB 接口图

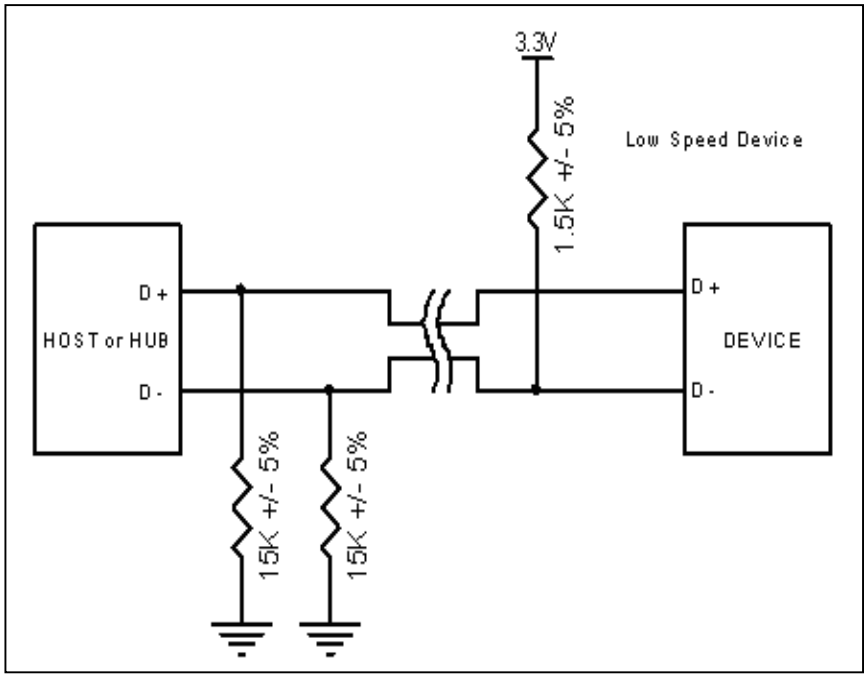
Type A	Type B	Mini-A	Mini-B	Micro-A	Micro-B
					
A型公口	B型公口	mini A型公口	mini B型公口	micro A型公口	micro B型公口
					
A型母口	B型母口	mini 母口		micro A型母口	micro B型母口
					

life.augmented

USB 设备的插入检测和速度检测



全速、高速设备：在D+上集成上拉电阻



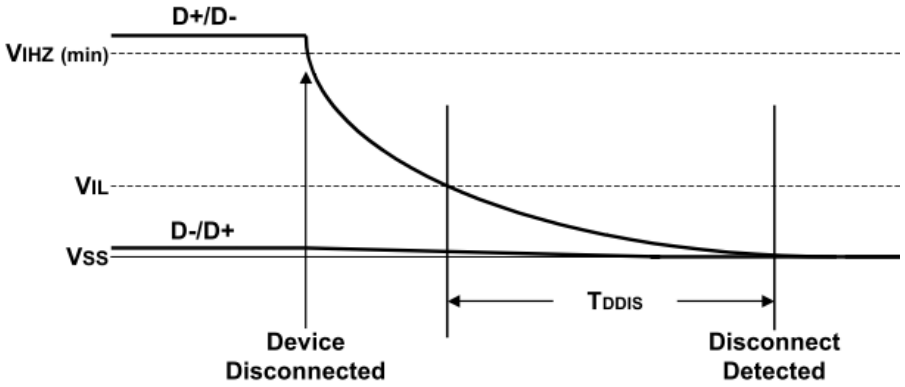
低速设备：在D-上集成上拉电阻

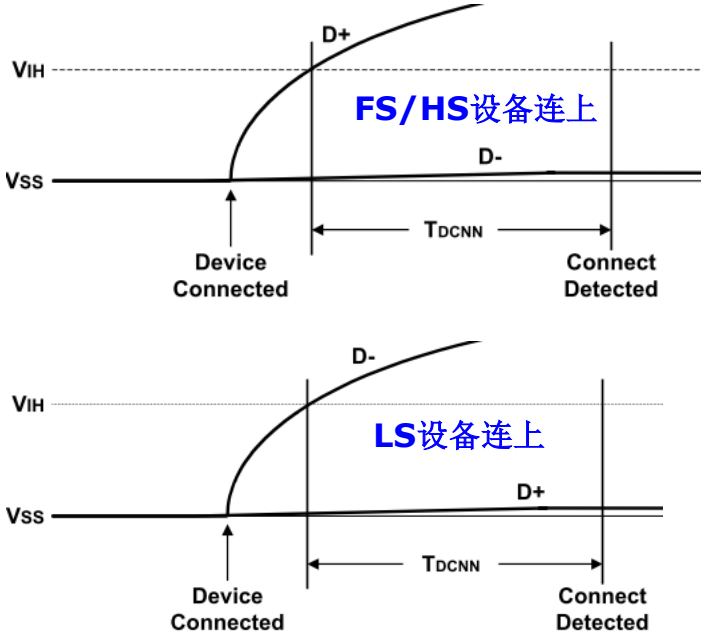
- 主机通过设备在D+或D-上的1.5K上拉来检测设备的连接和断开事件，并由此判别设备的速度
- 主机先把高速设备检测为全速设备，然后再通过“Chirp序列”的总线握手机制来识别高速和全速设备

life.augmented

连接和断开连接

- 没有设备连上主机时
  - D+和D-数据线上的下拉电阻起作用，使得二者都在低电平；主机端看来就是个SE0状态；同样地，当数据线上的SE0状态持续一段时间了，就被主机认为是断开状态
- 设备连上主机时
  - 当主机检测到某一个数据线电平拉高并保持了一段时间，就认为有设备连上来了
  - 主机必需在驱动SE0状态以复位设备之前，立刻采样总线状态来判断设备的速度



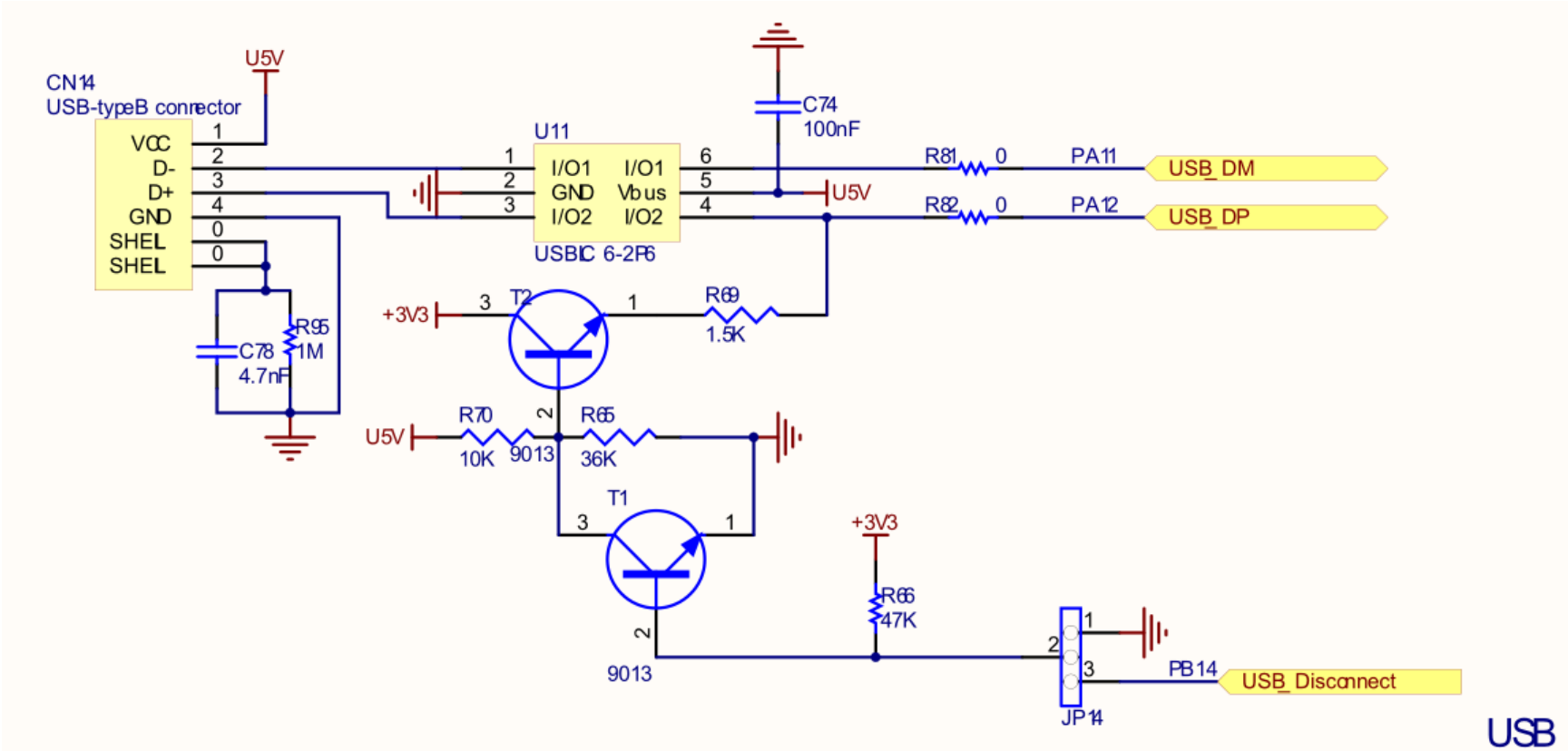


life.augmented

上下拉、寄存器设置、原理图

Bit 15 **DPPU:** DP pull-up control

This bit is set by software to enable the embedded pull-up on the DP line. Clearing it to '0' can be used to signalize disconnect to the host when needed by the user software.



当设备连上以后……

- 主机(下称H):你是什么设备?
- 设备(下称D):12 **01** 0100……**Device Descriptor**
- H:你有几种功能?
- D:09 **02** 09……**Configuration Descriptor**
- H:每个功能有几个接口?
- D:09 **04** 00……**Interface Descriptor**
- H:每个接口使用哪几个端点?
- D:06 **05** 82……**Endpoint Descriptor**
- H:我知道你是谁了,开始传输数据吧!
- D:OK, Read Go ☺

描述子的长度
描述子的类型编号

Descriptor Types	Value
DEVICE	1
CONFIGURATION	2
STRING	3
INTERFACE	4
ENDPOINT	5
DEVICE_QUALIFIER	6
OTHER_SPEED_CONFIGURATION	7
INTERFACE_POWER	8
OTG	9
DEBUG	10
INTERFACE_ASSOCIATION	11

life.augmented

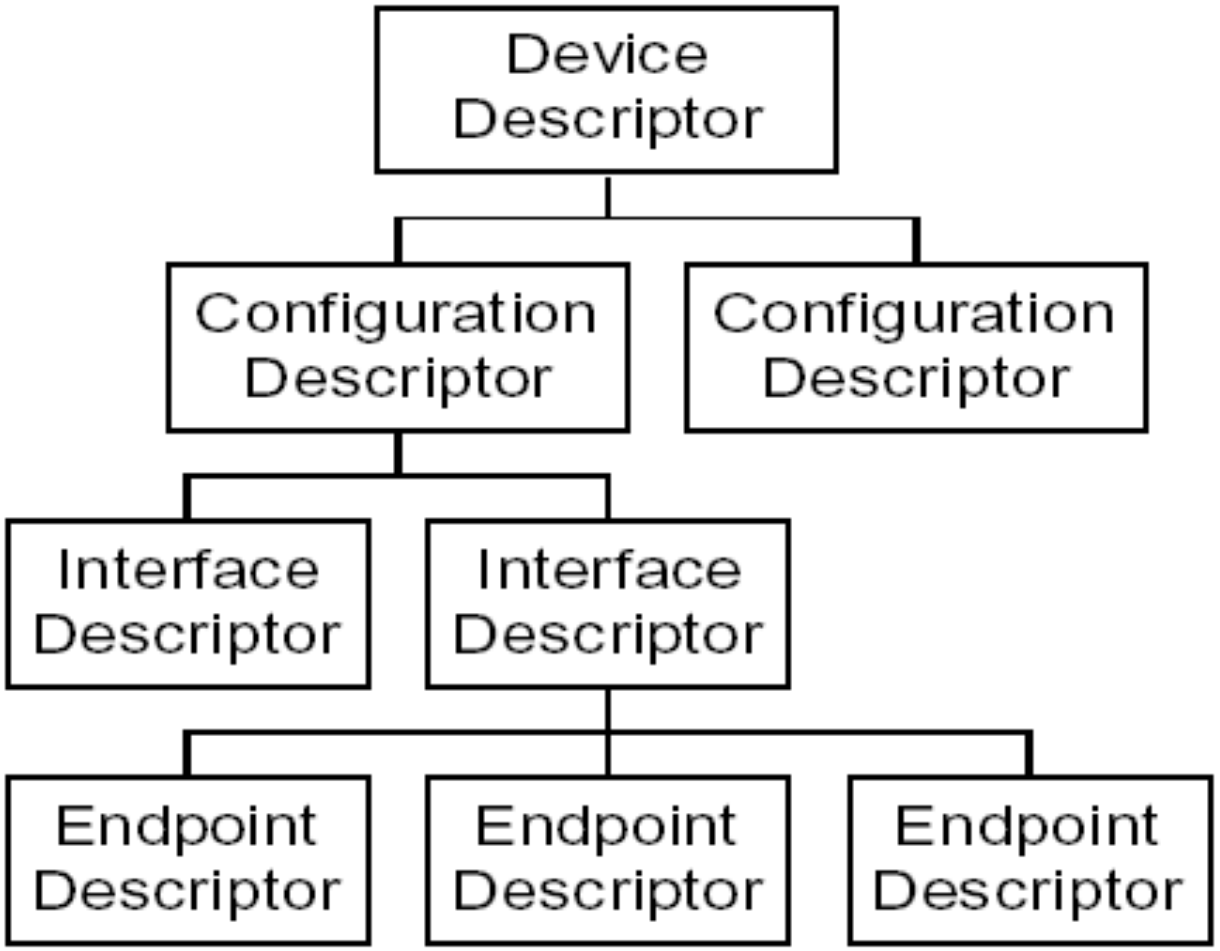
USB 设备枚举

当一个USB设备插入主机后，会有以下**枚举**活动:

- 供电
- 复位
- 获取Device Descriptor前8个字节信息
- 复位 (可选)
- 分配地址
- 获取Device Descriptor
- 获取Configuration Descriptor
- 获取String Descriptor (可选)
- 配置

即插即用的实现：描述符(Descriptor)

比如，一个USB接口的CD－ROM，作为一个设备，它具有两种功能，读取光盘和播放CD，所以有**2个Configuration描述符**。一个功能的实现要涉及许多接口，比如当CD播放机使用时，需要音频接口，同时还需要控制CD机的接口。而控制命令接口又由许多端点组成，比如有的端点负责向CD机发送命令，有的接口负责接收来自CD机的反馈



USB控制传输上的命令格式

Offset	Field	Size	Value	Description
0	*bmRequestType*	1	Bitmap	Characteristics of request: D7: Data transfer direction 0 = Host-to-device 1 = Device-to-host D6...5: Type 0 = Standard 1 = Class 2 = Vendor 3 = Reserved D4...0: Recipient 0 = Device 1 = Interface 2 = Endpoint 3 = Other 4...31 = Reserved
1	*bRequest*	1	Value	Specific request (refer to Table 9-3)
2	*wValue*	2	Value	Word-sized field that varies according to request
4	*wIndex*	2	Index or Offset	Word-sized field that varies according to request; typically used to pass an index or offset
6	*wLength*	2	Count	Number of bytes to transfer if there is a Data stage

>> 标准枚举命令

>> 类相关命令

>> 厂商自定义命令

USB 标准枚举命令

bmRequestType
b6:b5=00

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00000000B 00000001B 00000010B	CLEAR_FEATURE	Feature Selector	Zero Interface Endpoint	Zero	None
10000000B	GET_CONFIGURATION	Zero	Zero	One	Configuration Value
10000000B	GET_DESCRIPTOR	Descriptor Type and Descriptor Index	Zero or Language ID	Descriptor Length	Descriptor
10000001B	GET_INTERFACE	Zero	Interface	One	Alternate Interface
10000000B 10000001B 10000010B	GET_STATUS	Zero	Zero Interface Endpoint	Two	Device, Interface, or Endpoint Status
00000000B	SET_ADDRESS	Device Address	Zero	Zero	None
00000000B	SET_CONFIGURATION	Configuration Value	Zero	Zero	None
00000000B	SET_DESCRIPTOR	Descriptor Type and Descriptor Index	Zero or Language ID	Descriptor Length	Descriptor
00000000B 00000001B 00000010B	SET_FEATURE	Feature Selector	Zero Interface Endpoint	Zero	None
00000001B	SET_INTERFACE	Alternate Setting	Interface	Zero	None
10000010B	SYNCH_FRAME	Zero	Endpoint	Two	Frame Number

bRequest	Value
GET_STATUS	0
CLEAR_FEATURE	1
~~Reserved for future use~~	~~2~~
SET_FEATURE	3
~~Reserved for future use~~	~~4~~
SET_ADDRESS	5
GET_DESCRIPTOR	6
SET_DESCRIPTOR	7
GET_CONFIGURATION	8
SET_CONFIGURATION	9
GET_INTERFACE	10
SET_INTERFACE	11
SYNCH_FRAME	12

设备描述符（Device Descriptor）

Offset	Field	Size	Value	描述	备注
0	bLength	1	数字	该描述子包含的字节数目	一般都是18
1	bDescripotorType	1	常数	【设备描述符的类型常数】	1
2	bcdUSB	2	BCD	该设备符合遵循的USB协议版本号	
4	bDeviceClass	1	Class	由USB-IF分配的	一般都是0
5	bDeviceSubClass	1	SubClass		
6	bDeviceProtocol	1	Protocol		
7	bMaxPacketSize0	1	数字	EP0的最大包长	HS设备必须是64
8	idVendor	2	ID	USB-IF分配的	0x0483
10	iProduct	2	ID	厂家分配的	0x5710
12	bcdDevice	2	BCD	该设备自身的版本号	
14	iManufacturer	1	Index	各自在众多字符串描述符中的序号	
15	iProduct	1	Index		
16	iSerialNumber	1	index		
17	bNumConfigurations	1	数字	该设备在当前速度下支持多少种Configuration	

life.augmented

以Joystick为例

Joystick_DeviceDescriptor[ ] =

```
{
 0x12,              /* 整个Descriptor的长度：18字节 */
 0x01,              /* Descriptor的类别：Device Descriptor(0x01) */
 0x00, 0x02,        /* 设备所遵循的USB协议的版本号：2.00 */
 0x00,              /* 设备所实现的类：由每个接口描述符描述所实现的类*/
 0x00,              /* 设备所实现的子类：由每个接口描述符描述 */
 0x00,              /* 设备所遵循的协议类别：由每个接口描述符描述*/
 0x40,              /* 端点0的最大数据包长度：64字节*/
 0x83, 0x04,        /* IDVendor:  0x0483 (for ST) */
 0x10, 0x57,        /* IDProduct: 0x5710 */
 0x00, 0x02,        /* bcdDevice: 2.00*/
 1,                 /* 用于描述生产厂商的字符描述符的索引号 */
 2,                 /* 用于描述产品的字符描述符的索引号*/
 3,                 /* 用于描述产品系列号的字符描述符的索引号*/
 0x01               /* 设备所支持的配置数目：1*/
}
```

配置描述符（Configuration Descriptor）

Offset	Field	Size	Value	描述	备注
0	bLength	1	数字	该描述子包含的字节数目	一般都是9
1	bDescripotorType	1	常数	【配置描述符的类型常数】	2
2	wTotalLength	2	数字	包含了所有描述子的长度（配置、接口、端点、类相关、厂商相关）	X：当主机要求配置描述符时，所有相关的接口、端点描述符都要返回
4	bNumInterfaces	1	数字	该配置包含几个接口	
5	bConfigurationValue	1	数字	选择该配置时，传递给SetConfiguration()的参数	
6	iConfiguration	1	index	在众多字符串描述符中的序号	
7	bmAttributes	1	位图	D7：保留为1 D6：是否由总线供电 D5：是否支持远程唤醒 D4~D0：保留为0	鼠标通常是0xE0
8	bMaxPower	1	mA	该设备在该configuration下全速工作时需要从总线获取的电流（以2mA为单位）	

life.augmented

以Joystick为例

```
Joystick_ConfigurationDescriptor[ ] =
    0x09,                          /* 描述符的长度：9字节 */
    0x02,                          /* 描述符的类型: 0x02 配置描述符 */
    JOYSTICK_SIZ_CONFIG_DESC, 0x00,        /* 完整的描述符包括接口描述符、
                                           端点描述符和类描述符的长度 */
    0x01,                          /* 配置所支持的接口数目： 1*/
    0x01,                          /* 用SetConfiguration选择此配置时所指定的配置号*/
    0x00,                          /* 用于描述此配置的字符描述符的索引号： 0 */
    0xE0,                          /* 供电配置： B7(1 保留), B6(自供电), B5(远程唤醒), B4-
                                   B0(0 保留) */
    0x32,                          /* 最大功耗，以2mA为单位计算：0x32表示 100mA */
```

life.augmented

接口描述符（Interface Descriptor）

Offset	Field	Size	Value	描述	备注
0	bLength	1	数字	该描述子包含的字节数目	一般都是9
1	bDescripotorType	1	常数	【接口描述符的类型常数】	4
2	bInterfaceNumber	1	数字	该接口的序号：在该配置所支持的众多接口中的序号	从0开始
3	bAlternateSetting	1	数字	用来选择该接口的某个setting	
4	bNumEndpoints	1	数字	该接口支持多少个非0的EP	如果此项为0则该接口使用默认的控制通道
5	bInterfaceClass	1	Class	由USB-IF分配	*
6	bInterfaceSubClass	1	SubClass	由USB-IF分配	**
7	bInterfaceProtocol	1	Protocol	由USB-IF分配	**
8	iInterface	1	index	在众多字符串描述符中的序号	

* 常用设备类Class分配：AUDIO=1；COMMUNICATION=2；HID=3；MSC=8

** 取决于Class，有各自的定义：
Class = 3 (HID)的情况下，SubClass = 1 (BOOT) = 0 (no-BOOT) Protocol = 1 (KeyBoard) = 0 (Mouse)
1表示是个启动设备，一般对PC才有意义，表示bios启动时能识别并使用该设备；否则只能在OS启动后才能用

以Joystick为例

```
Joystick_InterfaceDescriptor[ ] =
    0x09,          /* 描述符的长度：9字节 */
    4,             /* 描述符的类型：0x04接口描述符(Interface) */
    0x00,          /* 选择此接口的索引号，从0开始计算：0 */
    0x00,          /* 用于选择此设置的索引号：0 */
    0x01,          /* 实现此接口需要使用的端点数目：1 */
    0x03,          /* 此接口所遵循的类：HID Class */
    0x01,          /* 此接口所遵循的子类：1=BOOT, 0=no boot */
    0x02,          /* 此接口所支持的协议：0自定义、1键盘、2鼠标 */
    0,             /* 用于描述此接口的字符描述符的索引号 */
```

端点描述符（Endpoint Descriptor）

Offset	Field	Size	Value	描述	备注
0	bLength	1	数字	该描述子包含的字节数目	一般都是7
1	bDescripotorType	1	常数	【端点描述符的类型常数】	5
2	bEndpointAddress	1	端点	Bit7：方向（0-OUT；1-IN） Bit6~4：保留为0 Bit3~0：端点号	
3	bmAttributes	1	位图	Bit1~0：传输类型（CTRL、ISO、BULK、INTR） Bit3~2：同步类型（） Bit5~4：使用类型（数据EP、反馈EP、非显式的反馈数据EP、…）	
4	wMaxPacketSize	2	数字	该端点的最大包长	
6	bInterval	1	数字	主机查询EP的间隔，以帧或微帧为单位 FS/HS的ISO EP：取值1~16，间隔是$2^{(bInterval-1)}$ FS/LS的INTR EP：取值1~255 HS的INTR EP：取值在1~16，间隔是$2^{(bInterval-1)}$ HS的BULK/CTRL OUT EP：设定了设备的最大NAK率（即每个微帧最多回几个NAK）；取值0~255	

life.augmented

以Joystick为例

```
Joystick_InterfaceDescriptor[ ] =
    0x07,           /* 描述符长度：7字节 */
    5,              /* 描述符类型：端点描述符*/
    0x81,           /* 端点的特性：B3-B0(端点号), B6-B4(0), B7(1=IN, 0=OUT)
                     0x81：Endpoint1/ IN */
    0x03,           /* 端点的类型：B1-B0(00=控制 01=同步 10=大容量 11=中断)
                    0x03：中断端点 */
    0x04, 0x00      /* 此端点的最大有效数据长度：4 字节 */
    0x20,           /* 主机查询此端点数据的间隔时间：(1ms或125us单位):
                    0x20：32 ms */
```

字符串描述符（String Descriptor）

Get String Descriptor (value= type3, index = 0)

Offset	Field	Size	Value	描述	备注
0	bLength	1	**N+2**	该描述子包含的字节数目	
1	bDescripotorType	1	常数	【字符串描述符的类型常数】	**3**
2	wLANGID[0]	2	数字	语言编号编码0	
…	…	..	**…**	…	
N	wLANGID[x]	2	数字	语言编号编码x	

Get String Descriptor (value= type3, index = 刚才得到的LANGID)

Offset	Field	Size	Value	描述	备注
0	bLength	1	数字	该描述子包含的字节数目	
1	bDescripotorType	1	常数	【字符串描述符的类型常数】	
2	bString	N	数字	字符串的UNICODE编码	

life.augmented

以Joystick为例

```
Joystick_StringVendor[ ] =
 {
    JOYSTICK_SIZ_STRING_VENDOR,          /* 描述符的长度 */
    USB_STRING_DESCRIPTOR_TYPE,          /* 描述符的类型：字符描述符 */
   'S', 0, 'T', 0, 'M', 0, 'i', 0, 'c', 0, 'r', 0, 'o', 0, 'e', 0,
   'l', 0, 'e', 0, 'c', 0, 't', 0, 'r', 0, 'o', 0, 'n', 0, 'i', 0,
   'c', 0, 's', 0                       /* 描述符所描述的内容："STMicroelectronics" */
 }
```

life.augmented

以Joystick为例

Joystick_ConfigDescriptor[ ] =

{

配置描述符：Configuration Descriptor

+

接口描述符：Interface Descriptor

+

类描述符：Class Descriptor

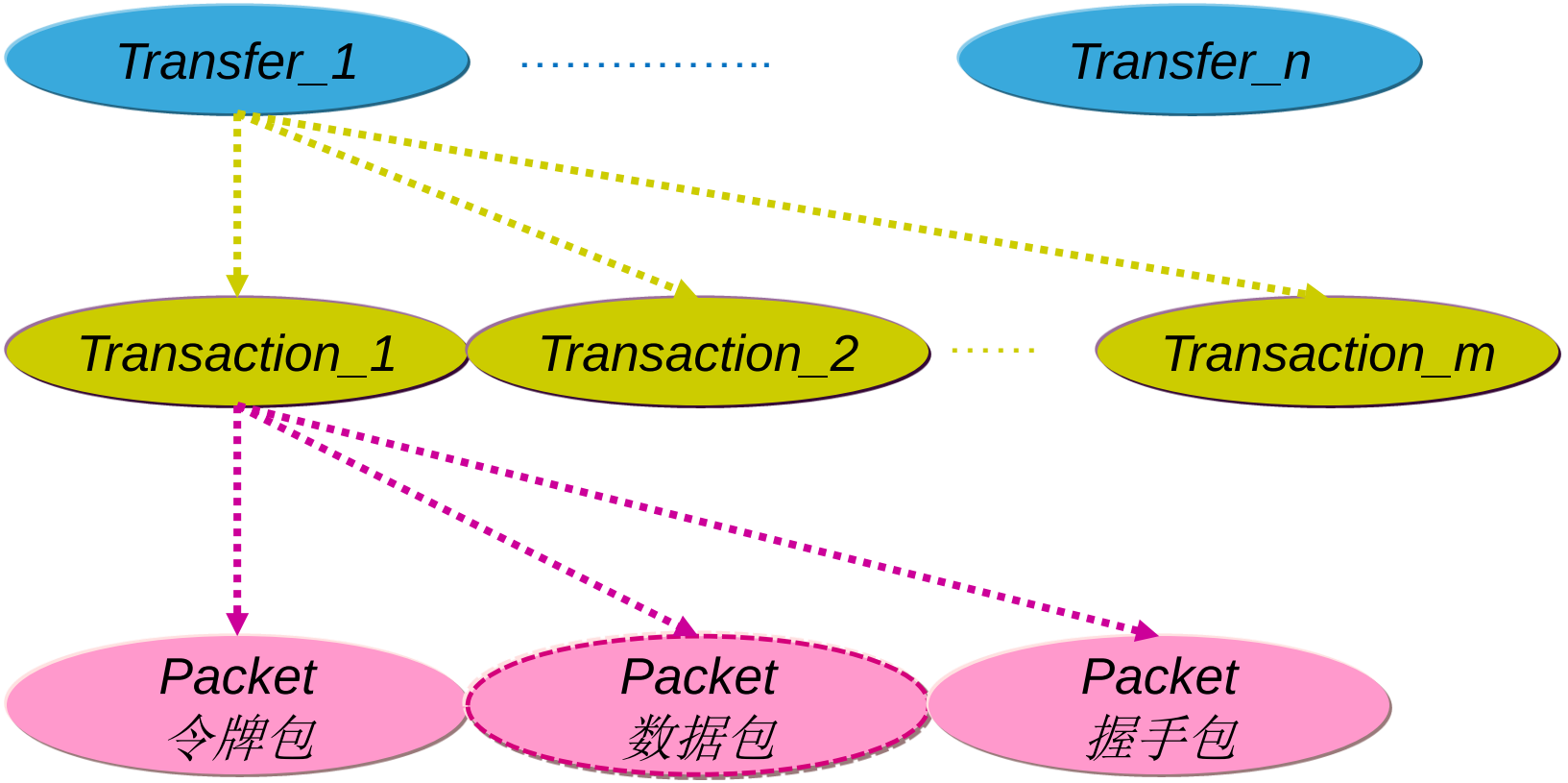
+

端点描述符：Endpoint Descriptor

}

life.augmented

USB通信：Packet、Transaction、Transfer



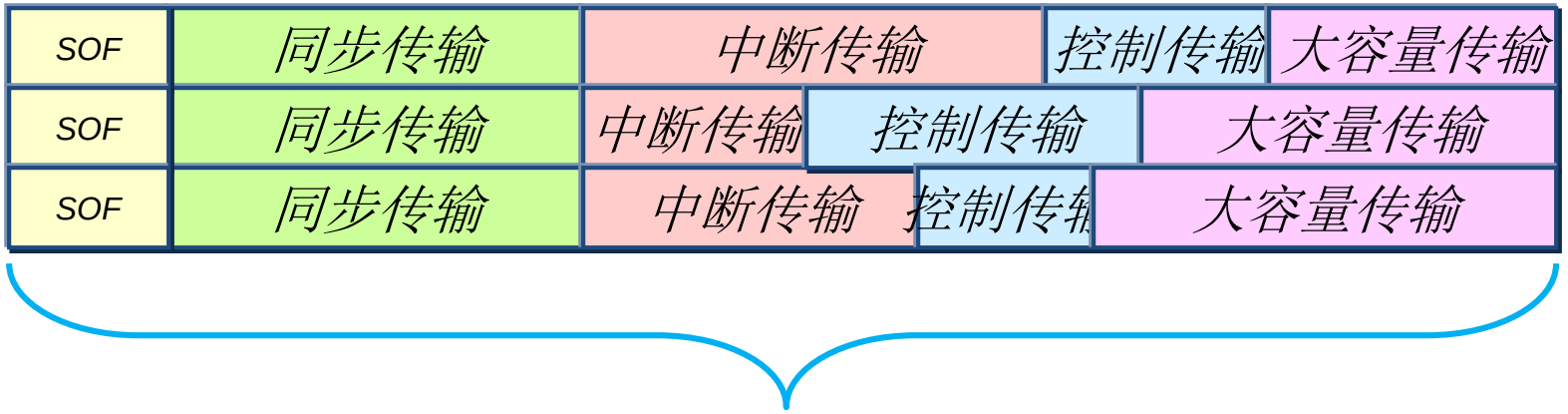
life.augmented

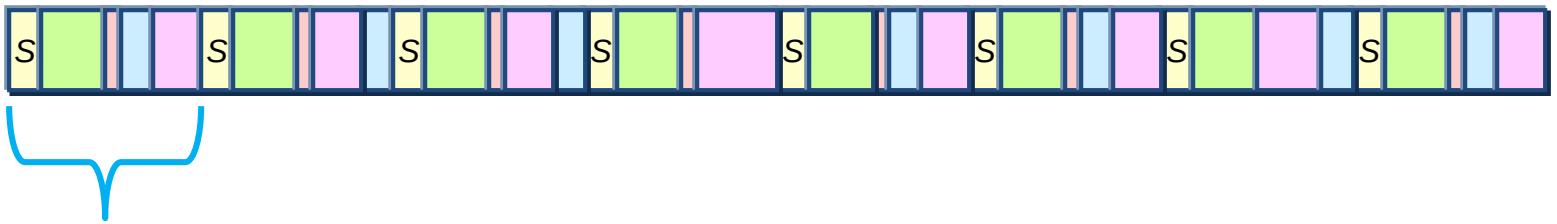
USB通信中的四种transfer类型

	Control			Bulk			Interrupt			isoch		
	HS	FS	LS	HS	FS	LS	HS	FS	LS	HS	FS	LS
带宽	保证			没有保证			有限的保留带宽			保证、传输率固定		
	20%	10%	10%				23.4M B/s	62.5 K B/s	800 B/s		999 K B/s	2.9M B/s
最大数据包长度	64	64	8	512	64	N.A	1024	64	8	1024	1023	N.A
传输错误管理	握手包、PID翻转			握手包、PID翻转			握手包、PID翻转			无错误纠察		
组成	Setup stage + Optional Data stage + Status stage			一个或多个Data transaction (IN or OUT)			一个Data transaction (IN)			一个或多个Data transaction (IN or OUT)		
典型使用场合	枚举			U盘、CDC数据交换			鼠标，HID类设备数据交换			音视频同步数据的传输		

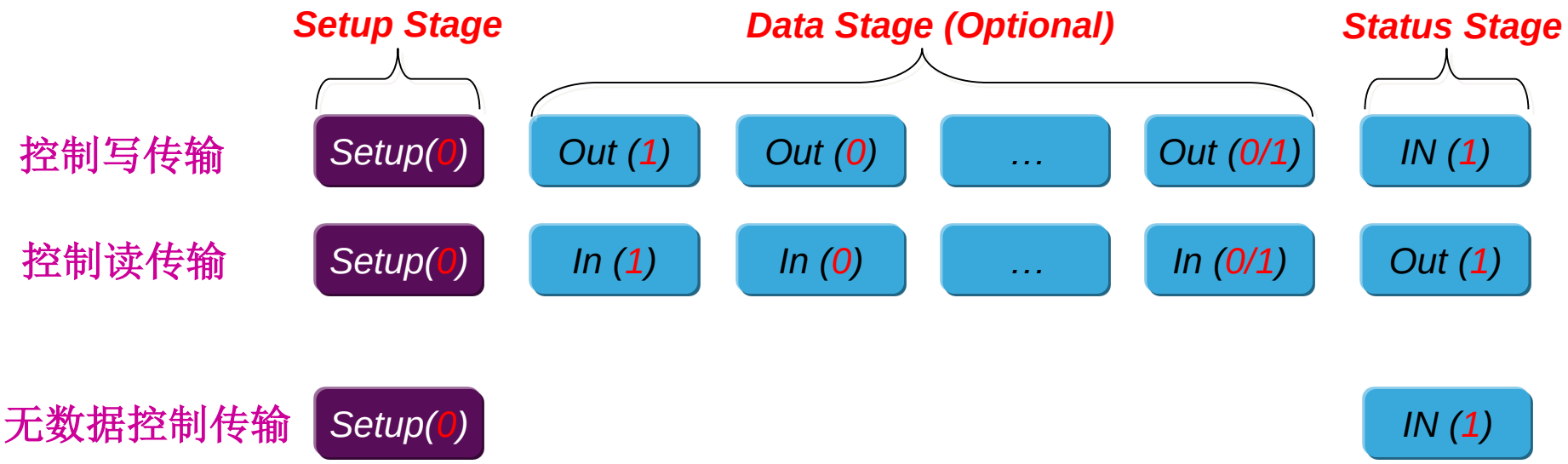
life.augmented

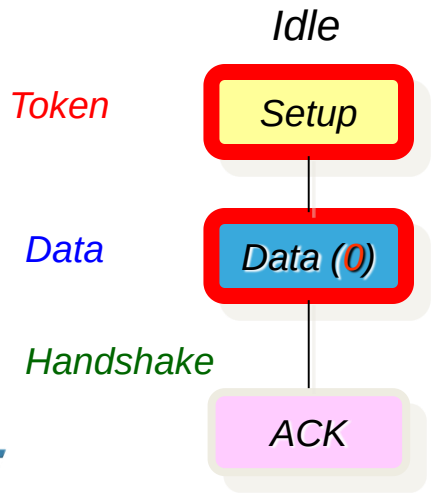
帧格式：由若干transfer组成

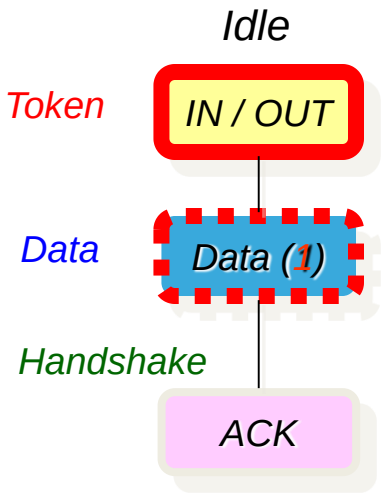


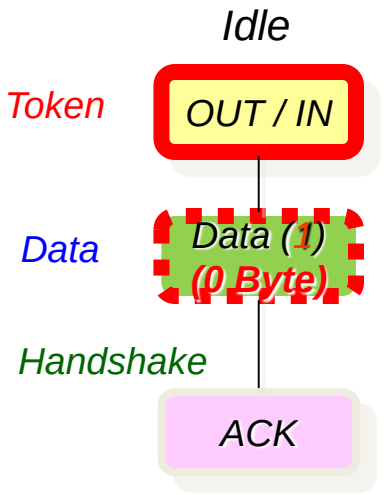


控制传输. 结构









life.augmented

控制传输. 实例

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time Stamp
0	S	GET	0	0	GET_DESCRIPTOR	DEVICE type	0x0000	DEVICE Descriptor	9 . 215 240 050

Transaction	F	SETUP	ADDR	ENDP	T	D	Tp	R	bRequest	wValue	wIndex	wLength	ACK	Time Stamp
0	S	0xB4	0	0	0	D->H	S	D	0x06	0x0100	0x0000	8	0x4B	9 . 215 240 050

Packet	H	F	Sync	SETUP	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
41		S	00000001	0xB4	0	0	0x08	233.330 ns	432.000 ns	9 . 215 240 050

Packet	H	F	Sync	DATA0	Data	CRC16	EOP	Idle	Time Stamp
42		S	00000001	0xC3	8 bytes	0xD729	250.000 ns	218.000 ns	9 . 215 243 382

Packet	D	F	Sync	ACK	EOP	Time	Time Stamp
43		S	00000001	0x4B	250.000 ns	5.973 ms	9 . 215 251 850

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time Stamp
1	S	0x96	0	0	1	8 bytes	0x4B	9 . 221 225 066

Packet	H	F	Sync	IN	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
50		S	00000001	0x96	0	0	0x08	233.330 ns	250.000 ns	9 . 221 225 066

Packet	D	F	Sync	DATA1	Data	CRC16	EOP	Idle	Time Stamp
51		S	00000001	0xD2	8 bytes	0x88BE	250.000 ns	350.000 ns	9 . 221 228 216

Packet	H	F	Sync	ACK	EOP	Time	Time Stamp
52		S	00000001	0x4B	250.000 ns	808.316 µs	9 . 221 236 816

Transaction	F	OUT	ADDR	ENDP	T	Data	ACK	Time Stamp
2	S	0x87	0	0	1	0 bytes	0x4B	9 . 222 045 132

Packet	H	F	Sync	OUT	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
53		S	00000001	0x87	0	0	0x08	233.330 ns	434.000 ns	9 . 222 045 132

Packet	H	F	Sync	DATA1	Data	CRC16	EOP	Idle	Time Stamp
54		S	00000001	0xD2	0 bytes	0x0000	250.000 ns	217.330 ns	9 . 222 048 466

Packet	D	F	Sync	ACK	EOP	Time	Time Stamp
55		S	00000001	0x4B	250.000 ns	5.187 ms	9 . 222 051 600

life.augmented

控制传输 (3)

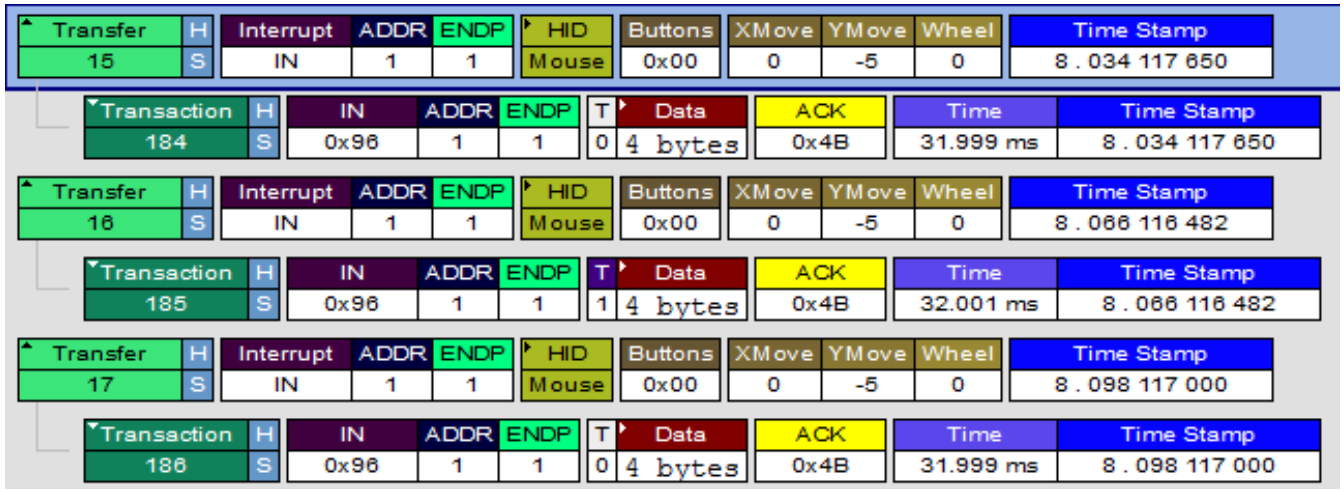
- **特性**
  - 每个USB设备都必须有控制端点，支持控制传输来进行命令和状态的传输
  - USB主机驱动将通过控制传输与USB设备的控制端点通信，完成USB设备的枚举和配置

- **方向**
  - 控制传输是双向的传输，必须由IN和OUT两个方向上的端点来完成两个方向上的数据通信

USB transaction

- Transaction可以分成三类
  - Setup transaction：主机用来向设备发送控制命令
  - Data IN transaction：主机用来从设备读取数据
  - Data OUT transaction：主机用来向设备发送数据
- Transaction的packet组成
  - Token packet：**总是由主机发出**
  - Data packet：包含此次transaction的数据负载，可以来自主机，也可以来自设备
  - **一次transfer要传输的数据量和packet包长，决定了该transfer会由多少个transaction组成**
  - 可选的Handshake packet

若干transaction组成一个transfer



Transfer	H	Interrupt	ADDR	ENDP	HID	Buttons	XMove	YMove	Wheel	Time Stamp
15	S	IN	1	1	Mouse	0x00	0	-5	0	8.034 117 650

Transaction	H	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
184	S	0x96	1	1	0	4 bytes	0x4B	31.999 ms	8.034 117 650

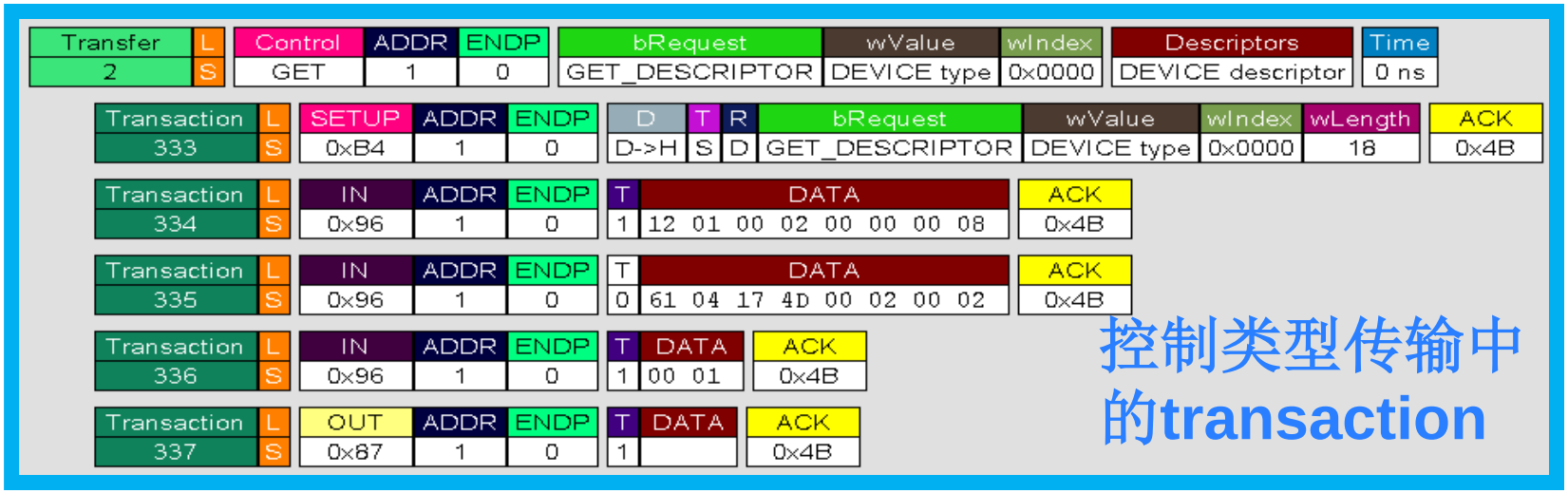
Transfer	H	Interrupt	ADDR	ENDP	HID	Buttons	XMove	YMove	Wheel	Time Stamp
16	S	IN	1	1	Mouse	0x00	0	-5	0	8.066 116 482

Transaction	H	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
185	S	0x96	1	1	1	4 bytes	0x4B	32.001 ms	8.066 116 482

Transfer	H	Interrupt	ADDR	ENDP	HID	Buttons	XMove	YMove	Wheel	Time Stamp
17	S	IN	1	1	Mouse	0x00	0	-5	0	8.098 117 000

Transaction	H	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
186	S	0x96	1	1	0	4 bytes	0x4B	31.999 ms	8.098 117 000

中断类型传输中的transaction



Transfer	L	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time
2	S	GET	1	0	GET_DESCRIPTOR	DEVICE type	0x0000	DEVICE descriptor	0 ns

Transaction	L	SETUP	ADDR	ENDP	D	T	R	bRequest	wValue	wIndex	wLength	ACK
333	S	0xB4	1	0	D->H	S	D	GET_DESCRIPTOR	DEVICE type	0x0000	18	0x4B

Transaction	L	IN	ADDR	ENDP	T	DATA	ACK
334	S	0x96	1	0	1	12 01 00 02 00 00 00 08	0x4B

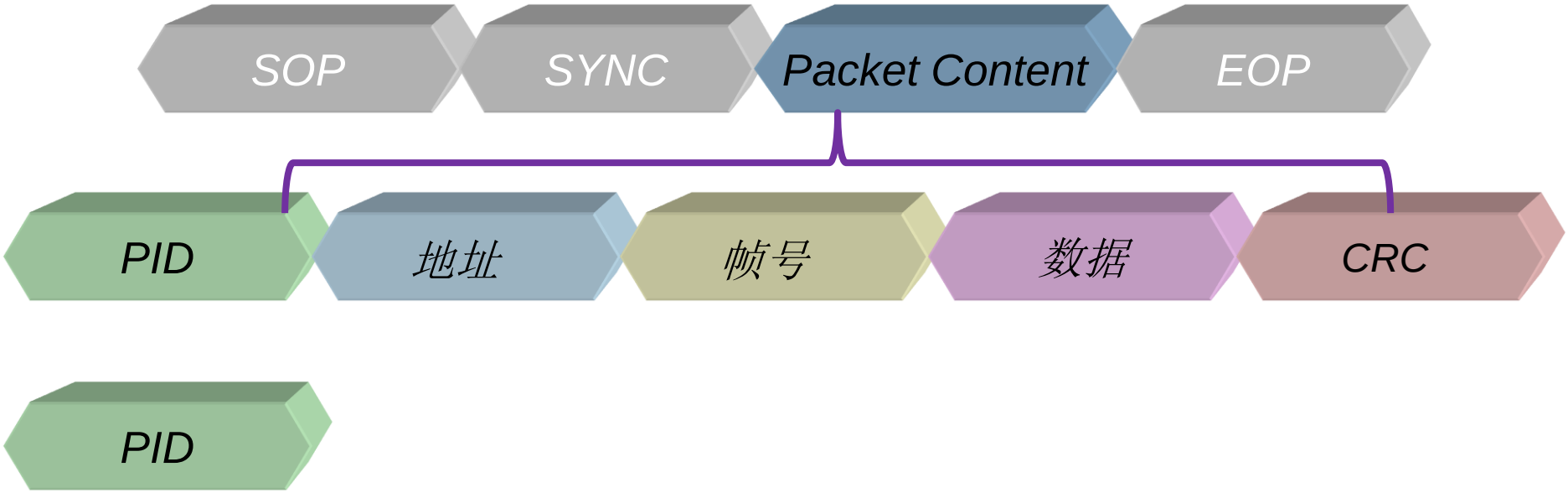
Transaction	L	IN	ADDR	ENDP	T	DATA	ACK
335	S	0x96	1	0	0	61 04 17 4D 00 02 00 02	0x4B

Transaction	L	IN	ADDR	ENDP	T	DATA	ACK
336	S	0x96	1	0	1	00 01	0x4B

Transaction	L	OUT	ADDR	ENDP	T	DATA	ACK
337	S	0x87	1	0	1		0x4B

控制类型传输中的transaction

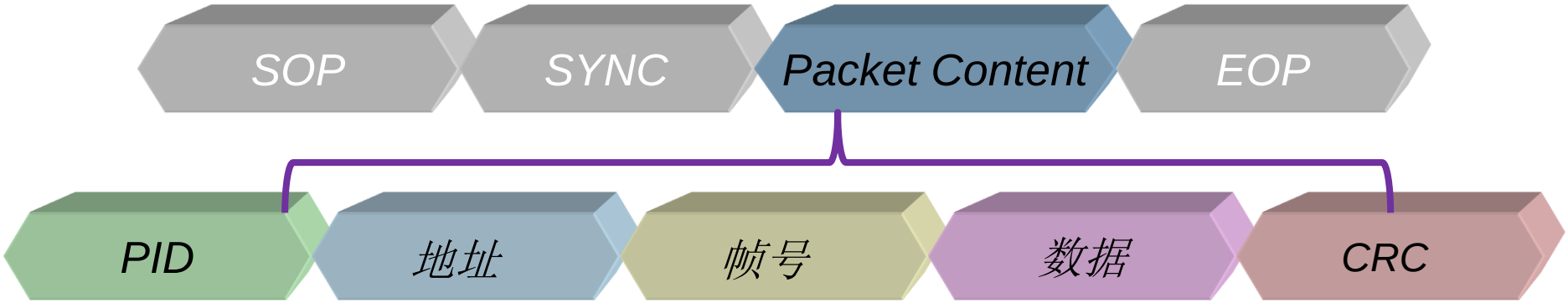
Packet的内容 (1)



PID 类型	PID 名称	对应packet种类	
Token 令牌	OUT /INT /SETUP/ SOF	令牌包、SOF包	不同类型包，以上组成部件有所不同 ……
Data 数据	DATA0 /DATA1 /DATA2/ MDATA	数据包	
Handshake 握手	ACK /NAK /STALL /NYET	握手包	
Special 特殊	PRE /ERR /SPLIT /PING		

life.augmented

Packet的内容 (2)



地址

设备地址

LSB MSB

$Addr_0$	$Addr_1$	$Addr_2$	$Addr_3$	$Addr_4$	$Addr_5$	$Addr_6$

端点地址

LSB MSB

$Endp_0$	$Endp_1$	$Endp_2$	$Endp_3$

LS设备最多3个端点

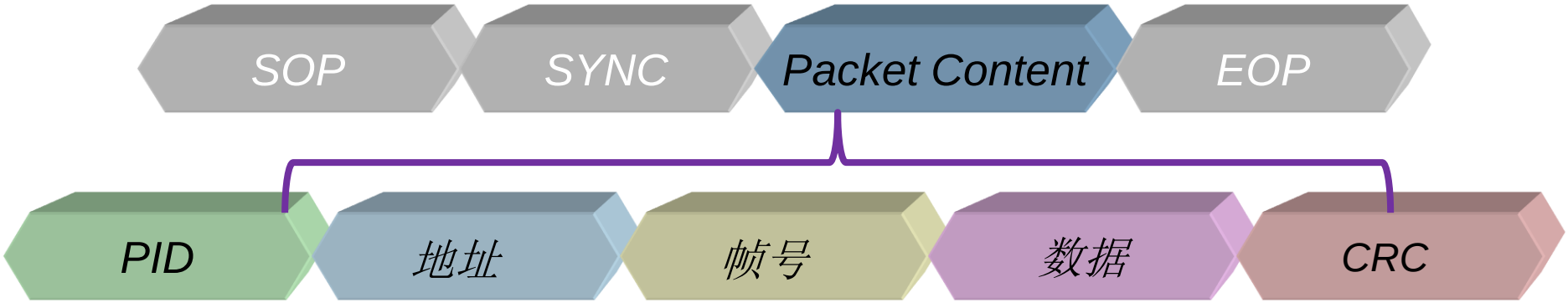
FS/HS设备最多16个端点

帧号

- 11 位
- 主机每发出一个帧，帧号都会自加1
- 当帧号达到7FFH时，将归零重新开始计数
- 仅在每个帧的帧首传输一次SOF包

life.augmented

Packet的内容 (3)



数据

➢ 根据传输类型的不同，数据域的数据长度从0到1024字节不等

	Control			Bulk			Interrupt			isoch		
	HS	FS	LS	HS	FS	LS	HS	FS	LS	HS	FS	LS
数据包长度	64	64	8	512	64	N.A	1024	64	8	1024	1023	N.A

CRC

➢ Token CRC：计算地址域和帧号域的CRC
➢ Data CRC：计算数据域数据的CRC

life.augmented

Packet实例

Sync	ACK	EOP
00000001	0x4B	233.330 ns

Sync	DATA0	Data	CRC16	EOP
00000001	0xC3	80 06 00 01 00 00 12 00	0x072F	250.000 ns

Sync	SOF	Frame #	CRC5	EOP
00000001	0xA5	1611	0x11	250.000 ns

Sync	SETUP	ADDR	ENDP	CRC5	EOP
00000001	0xB4	3	0	0x0A	250.000 ns

一次完整的枚举过程

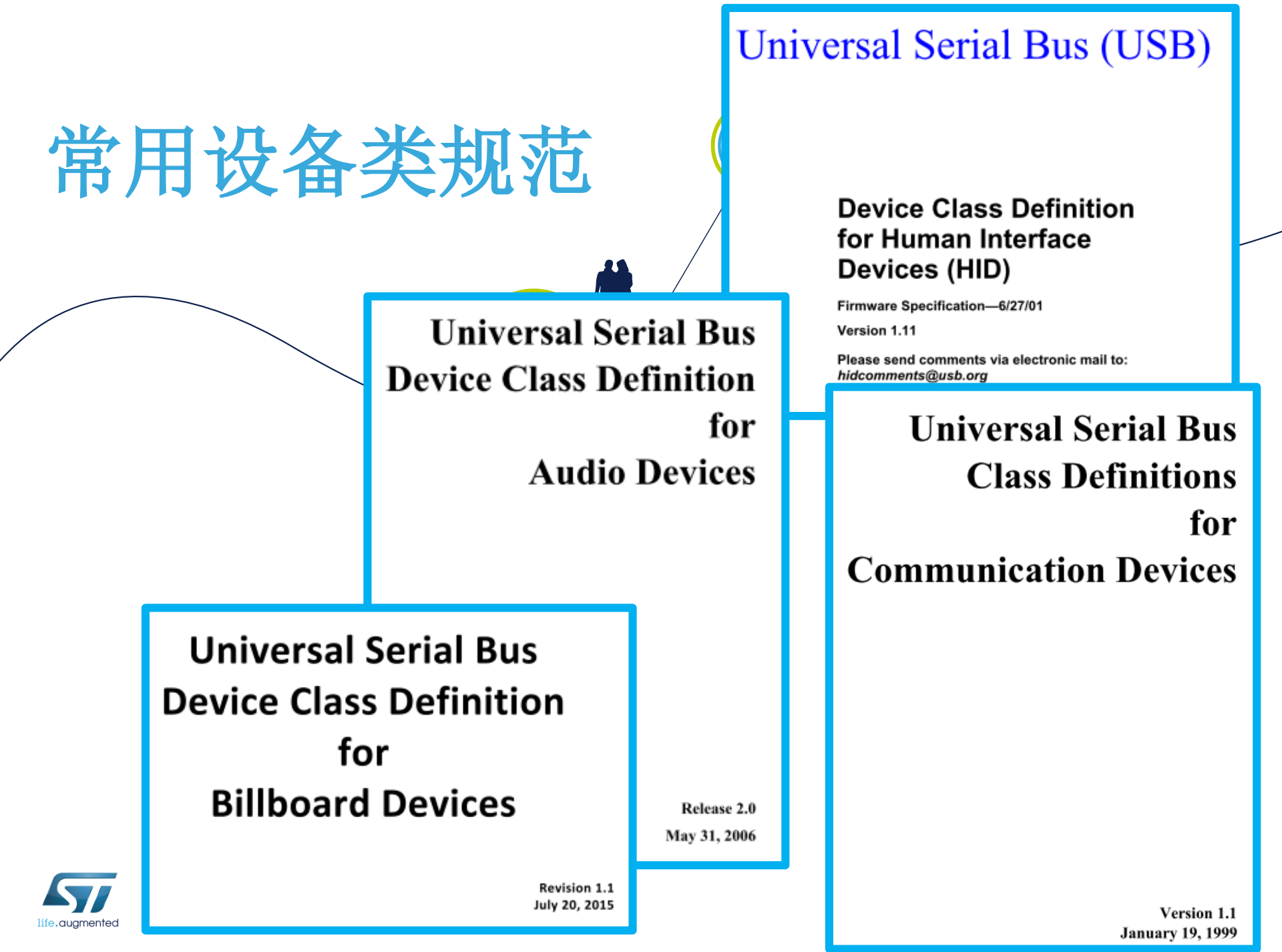
- 完整枚举Tracer file

[C:_Work\4. STM32 USB\[CAE]Mini Project\2. CompositeDevice(KB+Mo...\HID Mouse original demo.usb]

Transfer	H/S	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
7	S	GET	1	0	GET_DESCRIPTOR	DEVICE type	0x0000	DEVICE Descriptor	1.375 ms	4 . 021 155 182
8	S	GET	1	0	GET_DESCRIPTOR	CONFIGURATION type, Index 0	0x0000	CONFIGURATION Descriptor	999.932 us	4 . 022 530 100
9	S	GET	1	0	GET_DESCRIPTOR	CONFIGURATION type, Index 0	0x0000	4 Descriptors	5.503 ms	4 . 023 530 032
10	S	SET	1	0	SET_CONFIGURATION	New Configuration 1	0x0000	0	30.499 ms	4 . 029 033 300
11	S	SET	1	0	SET_IDLE	0x0000	0x0000	0	496.666 us	4 . 059 532 766
12	S	GET	1	0	GET_DESCRIPTOR	REPORT_DESCRIPTOR type	0x0000	REPORT Descriptor	3.319 sec	4 . 060 029 432
13	S	GET	1	0	GET_DESCRIPTOR	STRING type, Index 1	Language ID 0x0409	STMicroelectronics	1.500 ms	7 . 379 347 482
14	S	SET	1	0	SET_IDLE	0x0000	0x0000	0	653.270 ms	7 . 380 847 416

Transfer	H/S	Interrupt	ADDR	ENDP	HID	Buttons	XMove	YMove	Wheel	Time	Time Stamp
15	S	IN	1	1	Mouse	0x00	0	-5	0	31.999 ms	8 . 034 117 650
16	S	IN	1	1	Mouse	0x00	0	-5	0	32.001 ms	8 . 066 116 482
17	S	IN	1	1	Mouse	0x00	0	-5	0	31.999 ms	8 . 098 117 000

常用设备类规范



life.augmented

www.usb.org

- SuperSpeed USB
- USB Power Delivery
- USB Type-C™
- Hi-Speed USB
- USB On-The-Go and Embedded Host
- Tools
- USB-IF eStore
- Documents
  - USB 2.0 Documents
  - Wireless USB Documents
  - USB Device Class Specifications
  - Technical White Papers
  - Presentations / Other Documents
  - Archived Documents
- USB-IF Compliance Program
- USB FAQ
- Events
- Join USB-IF, Inc.
- Resources
- ExpressCard

Home | About USB-IF | Channel | Press | Developers | Members | Products

Home > Developers > **Documents**

USB 3.1 Specification

Universal Serial Bus Revision 3.1 Specification (.zip file format, size 56.6 MB) provides the technical details to understand USB 3.1 requirements and design USB 3.1 compatible products. Modifications to the USB 3.1 specification are made through Engineering Change Notices (ECNs). Enclosed in this zip file are the following documents:

Home | About USB-IF | Channel | Press | Developers | Members | Products

Home > Developers > Documents > **USB 2.0 Documents**

USB 2.0 Documents Index

USB 2.0 Specification
USB 2.0 Adopters Agreement
USB 2.0 On-The-Go Specification Supplement Adopters Agreement
Low Voltage Inter-Chip Adopters Agreement Amendment
High-Speed Inter-Chip Adopters Agreement Amendment
Intellectual Property Rights Notice
USB 2.0 Icon Design Guideline

Home | About USB-IF | Channel | Press | Developers | Members | Products

Home > Developers > Documents > **USB Device Class Specifications**

USB-IF Device Class Documents

The USB Device Working Group ("DWG") enables and promotes increased interoperability and reliability between USB products through documents that augment, enhance and extend USB specifications. USB-IF members may participate in the development efforts of a Working Group only after having executed the IP Agreement corresponding to the specification under development by that Group. If you are interested in working on a Device Class document, you will find the various IP Agreements and can submit your request to join in the members-only section of this site. The current set of Defined 1.0 Class Codes is available.

These specifications recommend design targets for classes of devices. For HID related information, please go to the HID Web page.

Approved Class Specification Documents

USB类编码(USB Class Codes)

- USB定义的class code是用来区分设备功能，并基于其功能加载设备驱动，由三个字段组成：Base class-Subclass-Protocol

- 关于USB类编码信息可以发在设备的两个地方：Device Descriptor和Interface Descriptor（有些编码只能放在设备描述符，有些只能放在接口描述符，有些放在两个地方都行）

Base Class	Descriptor Usage	Description
00	Device	在接口描述符中定义类信息
01	Interface	Audio
02	Both	CDC control
03	Interface	HID
05	Interface	Physical
06	Interface	Image
07	Interface	Printer
08	Interface	Mass Storage
09	Device	Hub
0A	Interface	CDC data

Base Class	Descriptor Usage	Description
OB	Interface	Smart Card
0D	Interface	Security
0E	Interface	Video
0F	Interface	Healthacre
10	Interface	Audio/Video Device
11	Device	Billboard
DC	Both	Diagnostic
E0	Interface	Wireless
EF	Both	Misc.
FE	Interface	App. Spec.
FF	Both	Vendor Spec.

HID设备类，以鼠标应用为例

- HID类编码、类相关描述符
- HID类相关的命令
- HID类相关的数据传输

HID类的Class-Subclass-Interface

- HID Class
  - USB设备被分成不同的设备类别：共享相同的类驱动，且具有类似的数据传输要求
  - 一个USB设备可以是一个单独的类别，或者是多个类别的复合设备
    - e.g. 手机使用HID、音频、telephony等多个类的feature
- HID Subclass
  - HID类不使用subclass来定义协议，而是使用**报告描述符**来表达它的数据协议和类型
  - 设备检测并识别后，HID的类驱动装载报告描述符并解析设备传上来的【报告】
  - 使用Subclass来表示该设备（鼠标或键盘）支持预定义的协议（即是一个Boot设备
- HID Protocol
  - HID设备支持不同的协议
  - 只有 (bInterfaceSubClass @ 接口描述符)表明该设备支持boot interface时，该项才有效；否则就是0

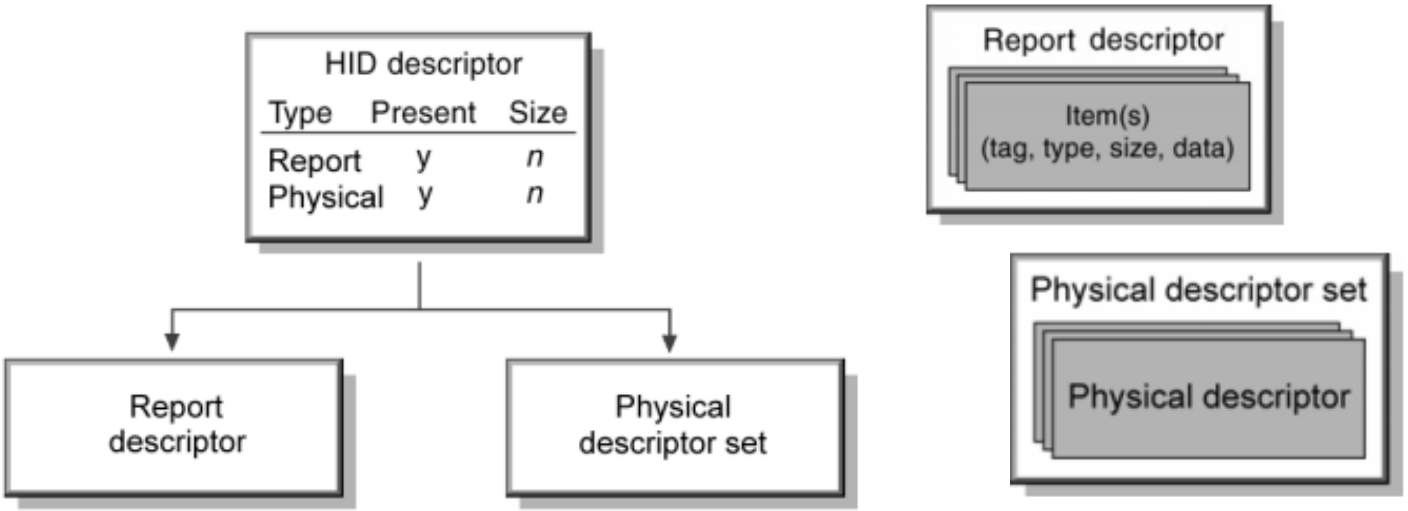
Subclass代码	描述
0	非Subclass
1	Boot Interface Subclass
2~255	保留

Protocol代码	描述
0	NA
1	键盘
2	鼠标
3~255	保留

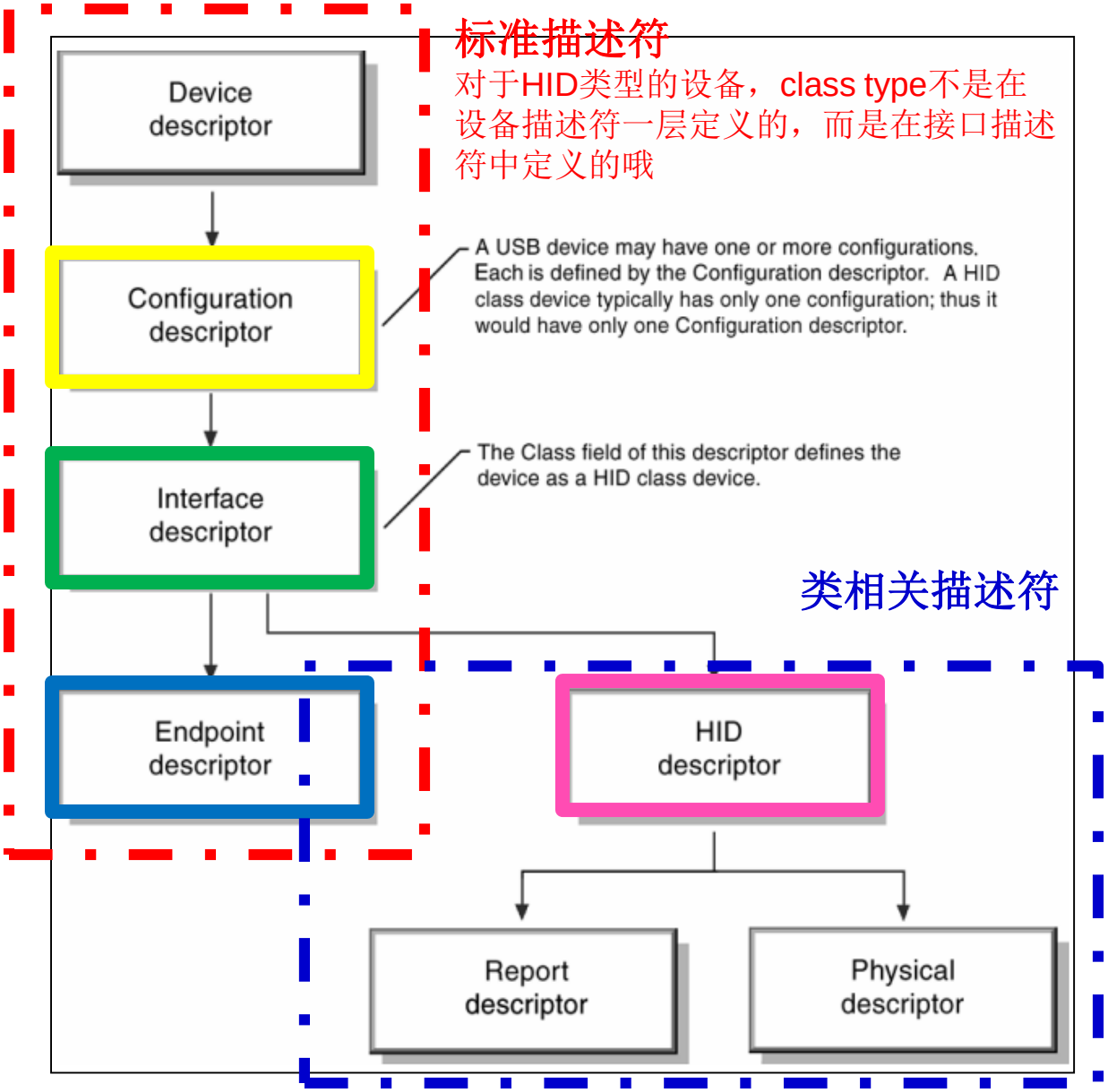
life.augmented

@ HID1_11

- **HID描述符（点击超链）**描绘该应用中还有哪些HID类相关的描述符存在，比如Report描述符和Physical描述符，以及它们所占的字节数
- Report描述符：描述设备产生的每条数据所代表的意思
  - 定义比如描述位置和按钮状态的item，1. 用来告诉主机这些获得数据route给谁，比如给鼠标或者游戏摇杆API；2. 让软件给输入数据分配对应功能，比如使用游戏摇杆的输入来定位坦克
- Physical描述符：可选项，用来告诉主机使用人体的哪些部位来激活设备上的对应控制



HID类设备的描述符结构



life.augmented

HID描述符

- HID描述符告诉主机随后描述符的个数以及类型

Offset	Field	Size	描述	备注
0	bLength	1	该描述符包含的字节数目	……
1	bDescriptorType	1	【HID描述符的类型常数】	**0x21**
2	bcdUSB	2	遵循的HID规范版本号	
4	bCountryCode	1	硬件所在国家	指明硬件是用于哪个国家：大多数硬件无需本地化，因此该值为0；但是键盘可以用该域来表明键盘的语言：page 23 @ HID1_11.pdf
5	bNumDescriptors	1	其后的类相关描述符个数	至少是1：因为至少有一个报告描述符
6	bDescriptorType	1	【报告描述符类型】	**0x22**
7	wItemLength	2	报告描述符的长度	21 00 (33字节的报告描述符)

- Rapoo鼠标
  - 3个HID接口，3个HID描述符

life.augmented

Rapoo的三个接口的HID描述符

Offset	Field	描述	1st 接口	2nd 接口	3rd 接口
0	bLength	该描述符包含的字节数目	0x09 (9)	0x09 (9)	0x09 (9)
1	bDescriptorType	【HID描述符的类型常数】	0x21	0x21	0x21
2	bcdUSB	遵循的HID规范版本号	10 01 (1.1)	10 01 (1.1)	10 01 (1.1)
4	bCountryCode	硬件所在国家	00	00	00
5	bNumDescriptors	其后的HID类描述符个数	01	01	01
6	bDescriptorType	【报告描述符类型】	0x22	0x22	0x22
7	wItemLength	报告描述符的长度	3B 00 (59字节的报告描述符)	94 00 (148字节的报告描述符)	21 00 (33字节的报告描述符)

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors
17	S	GET	4	0	GET_DESCRIPTOR	REPORT_DESCRIPTOR type	0x0000	REPORT Descriptor

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors
19	S	GET	4	0	GET_DESCRIPTOR	REPORT_DESCRIPTOR type	0x0001	REPORT Descriptor

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors
21	S	GET	4	0	GET_DESCRIPTOR	REPORT_DESCRIPTOR type	0x0002	REPORT Descriptor

举例：鼠标的报告描述符

Main item

Global item

Local item

```
Usage Page (Generic Desktop),
Usage (Mouse),
Collection (Application),
        Usage (Pointer),
        Collection (Physical),
                Report Count (3),
                Report Size (1),
                Usage Page (Buttons),
                Usage Minimum (1),
                Usage Maximum (3),
                Logical Minimum (0),
                Logical Maximum (1),
                Input (Data, Variable, Absolute),
                Report Count (1),
                Report Size (5),
                Input (Constant),
                Report Size (8),
                Report Count (2),
                Usage Page (Generic Desktop),
                Usage (X),
                Usage (Y),
                Logical Minimum (-127),
                Logical Maximum (127),
                Input (Data, Variable, Relative),
        End Collection,
End Collection
```

顺序可不固定

1字节输入report

B1	B2	B3	X	X	X	X	X
取值0/1							
数据/变量/绝对值			用于填充的常数				

2字节输入report

取值 -127 ~ 127							
数据/变量/相对值							

life.augmented

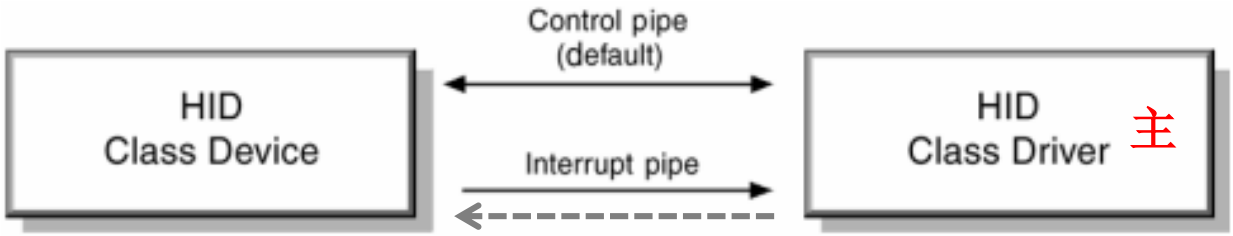
@ HID1_11

- HID类设备和主机端的HID类驱动使用默认的控制pipe或中断pipe通信
- （双向）控制pipe用于
  - 对控制命令和类相关数据进行接收和响应
  - 在HID类驱动使用Get_Report命令进行轮询时，**发送数据**给主机
  - **接收**来自主机的数据

Message data

- 中断pipe用于
  - 主机接收来自设备的异步数据 (unrequested)
  - 发送低延迟的数据给设备
    - 中断类型的OUT pip是可选的，主机可用它来output report；如果没有的话，主机就使用Set_Report来向设备output report

Stream data



HID类相关命令

7.2.1 Get_Report Request

The **Get_Report** request allows the host to receive a report via the **Control** pipe.

7.2.2 Set_Report Request

The **Set_Report** request allows the host to send a report to the device, possibly setting the state of input, output, or feature controls.

7.2.3 Get_Idle Request

The **Get_Idle** request reads the current idle rate for a particular **Input** report (see: **Set_Idle** request).

7.2.4 Set_Idle Request

The **Set_Idle** request silences a particular report on the **Interrupt In** pipe until a new event occurs or the specified amount of time passes.

7.2.5 Get_Protocol Request

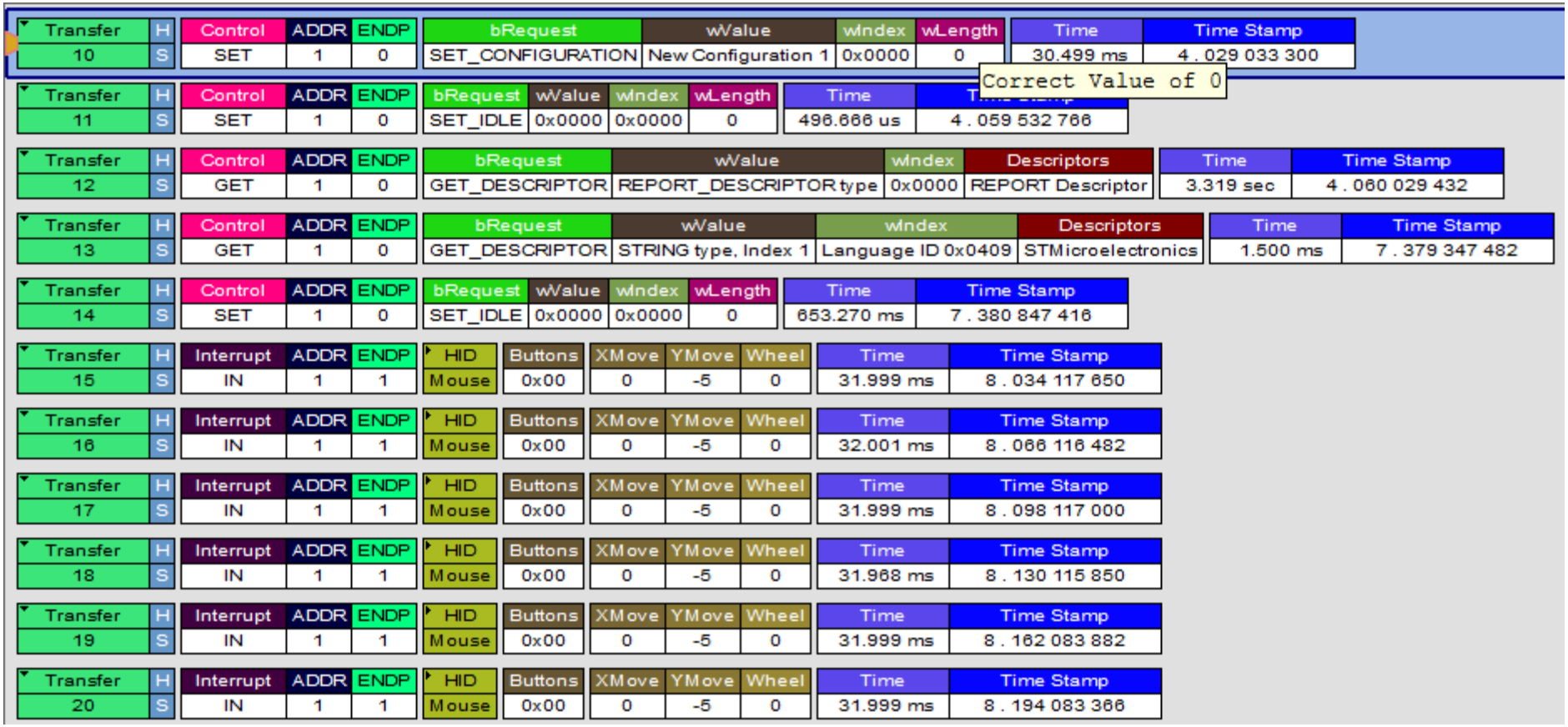
The **Get_Protocol** request reads which protocol is currently active (either the boot protocol or the report protocol.)

7.2.6 Set_Protocol Request

The **Set_Protocol** switches between the boot protocol and the report protocol (or vice versa).

life.augmented

鼠标通信tracer



Transfer	H	Control	ADDR	ENDP	bRequest	wValue	wIndex	wLength	Time	Time Stamp
10	S	SET	1	0	SET_CONFIGURATION	New Configuration 1	0x0000	0	30.499 ms	4 . 029 033 300

Correct Value of 0

Transfer	H	Control	ADDR	ENDP	bRequest	wValue	wIndex	wLength	Time	Time Stamp
11	S	SET	1	0	SET_IDLE	0x0000	0x0000	0	496.666 us	4 . 059 532 766

Transfer	H	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
12	S	GET	1	0	GET_DESCRIPTOR	REPORT_DESCRIPTOR type	0x0000	REPORT Descriptor	3.319 sec	4 . 060 029 432

Transfer	H	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
13	S	GET	1	0	GET_DESCRIPTOR	STRING type, Index 1	Language ID 0x0409	STMicroelectronics	1.500 ms	7 . 379 347 482

Transfer	H	Control	ADDR	ENDP	bRequest	wValue	wIndex	wLength	Time	Time Stamp
14	S	SET	1	0	SET_IDLE	0x0000	0x0000	0	653.270 ms	7 . 380 847 416

Transfer	H	Interrupt	ADDR	ENDP	HID	Buttons	XMove	YMove	Wheel	Time	Time Stamp
15	S	IN	1	1	Mouse	0x00	0	-5	0	31.999 ms	8 . 034 117 650
16	S	IN	1	1	Mouse	0x00	0	-5	0	32.001 ms	8 . 066 116 482
17	S	IN	1	1	Mouse	0x00	0	-5	0	31.999 ms	8 . 098 117 000
18	S	IN	1	1	Mouse	0x00	0	-5	0	31.968 ms	8 . 130 115 850
19	S	IN	1	1	Mouse	0x00	0	-5	0	31.999 ms	8 . 162 083 882
20	S	IN	1	1	Mouse	0x00	0	-5	0	31.999 ms	8 . 194 083 366

Universal Serial Bus Specification

Compaq

Hewlett-Packard

Intel

Lucent

Microsoft

NEC

Philips

Revision 2.0

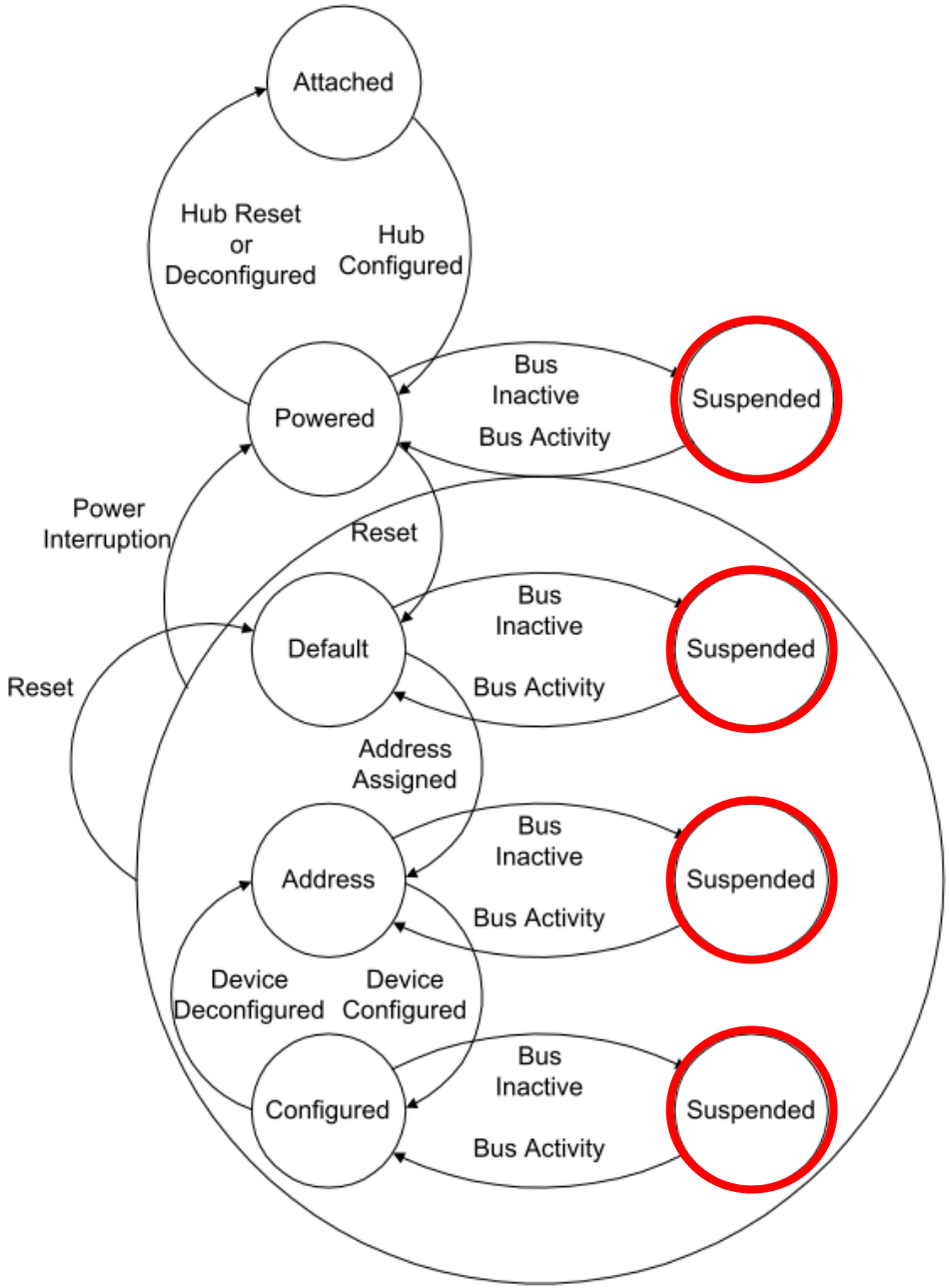
April 27, 2000

通用串行总线 USB 2.0

life.augmented

USB协议...设备的状态...

Attached	Powered	Default	Address	Configured	Suspended	State
No	--	--	--	--	--	Device is not attached to the USB. Other attributes are not significant.
Yes	No	--	--	--	--	Device is attached to the USB, but is not powered. Other attributes are not significant.
Yes	Yes	No	--	--	--	Device is attached to the USB and powered, but has not been reset.
Yes	Yes	Yes	No	--	--	Device is attached to the USB and powered and has been reset, but has not been assigned a unique address. Device responds at the default address.
Yes	Yes	Yes	Yes	No	--	Device is attached to the USB, powered, has been reset, and a unique device address has been assigned. Device is not configured.
Yes	Yes	Yes	Yes	Yes	No	Device is attached to the USB, powered, has been reset, has a unique address, is configured, and is not suspended. The host may now use the function provided by the device.
Yes	Yes	--	--	--	Yes	Device is, at minimum, attached to the USB and is powered and has not seen bus activity for 3 ms. It may also have a unique address and be configured for use. However, because the device is suspended, the host may not use the device's function.



从USB协议来看Suspend

- Suspend是**设备**的众多状态中的一种

- 一旦被power了，设备就要随时准备进入suspend状态，无论它是否已被分配了地址和是否已经完成了来自主机的配置

- 设备是否支持remote wakeup特性，要在自己的配置描述符中说清楚；如果支持的话，那么它也要继续支持host发来的对该特性的使能和禁止的request

Offset	Field	Size	Value	描述	备注
7	bmAttributes	1	位图	D7：保留为1 **D6：是否由总线供电** **D5：是否支持远程唤醒** D4~D0：保留为0	**鼠标通常是0xE0**

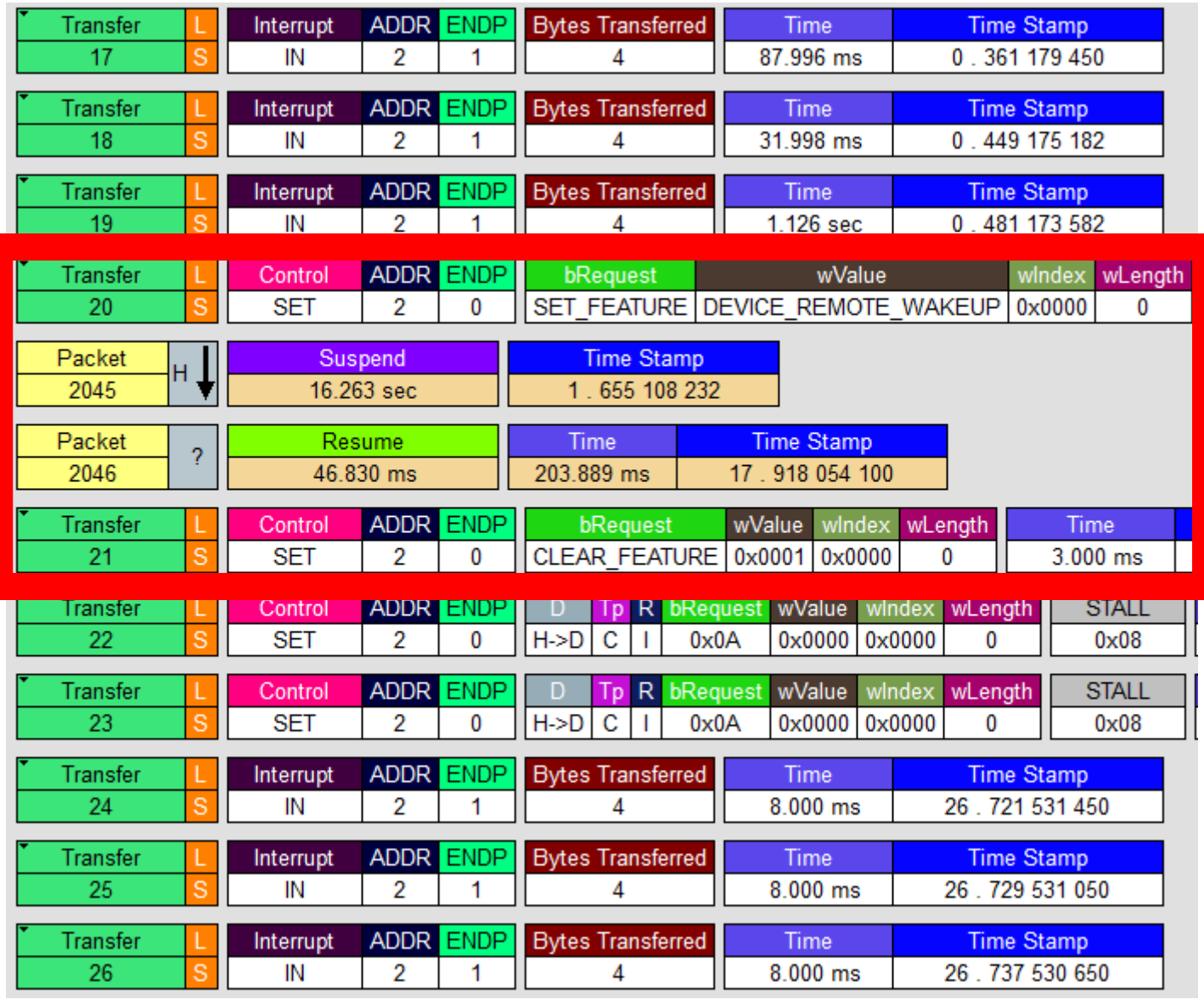
life.augmented

Feature…

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00000000B 00000001B 00000010B	CLEAR_FEATURE	Feature Selector	Zero Interface Endpoint	Zero	None
00000000B 00000001B 00000010B	SET_FEATURE	Feature Selector	Zero Interface Endpoint	Zero	None

Feature Selector	Recipient	Value
DEVICE_REMOTE_WAKEUP	Device	1
ENDPOINT_HALT	Endpoint	0
TEST_MODE	Device	2

PC休眠并自我唤醒



Transfer	L/S	Interrupt	ADDR	ENDP	Bytes Transferred	Time	Time Stamp
17	S	IN	2	1	4	87.996 ms	0 . 361 179 450
18	S	IN	2	1	4	31.998 ms	0 . 449 175 182
19	S	IN	2	1	4	1.126 sec	0 . 481 173 582

Transfer	L/S	Control	ADDR	ENDP	bRequest	wValue	wIndex	wLength
20	S	SET	2	0	SET_FEATURE	DEVICE_REMOTE_WAKEUP	0x0000	0

Packet	H	Suspend	Time Stamp
2045	↓	16.263 sec	1 . 655 108 232

Packet	?	Resume	Time	Time Stamp
2046		46.830 ms	203.889 ms	17 . 918 054 100

Transfer	L/S	Control	ADDR	ENDP	bRequest	wValue	wIndex	wLength	Time
21	S	SET	2	0	CLEAR_FEATURE	0x0001	0x0000	0	3.000 ms

Transfer	L/S	Control	ADDR	ENDP	D	Tp	R	bRequest	wValue	wIndex	wLength	STALL
22	S	SET	2	0	H->D	C	I	0x0A	0x0000	0x0000	0	0x08
23	S	SET	2	0	H->D	C	I	0x0A	0x0000	0x0000	0	0x08

Transfer	L/S	Interrupt	ADDR	ENDP	Bytes Transferred	Time	Time Stamp
24	S	IN	2	1	4	8.000 ms	26 . 721 531 450
25	S	IN	2	1	4	8.000 ms	26 . 729 531 050
26	S	IN	2	1	4	8.000 ms	26 . 737 530 650

life.augmented

PC休眠并被鼠标唤醒

Transfer		Interrupt	ADDR	ENDP	Bytes Transferred	Time	Time Stamp
344	S	IN	1	1	4	63.997 ms	5 . 305 548 916
345	S	IN	1	1	4	63.997 ms	5 . 369 545 850
346	S	IN	1	1	4	1.178 sec	5 . 433 542 782

Transfer		Control	ADDR	ENDP	bRequest	wValue	wIndex	wLength
347	S	SET	1	0	SET_FEATURE	DEVICE_REMOTE_WAKEUP	0x0000	0

Packet		Suspend	Time Stamp
8621	H	14.163 sec	6 . 653 475 332

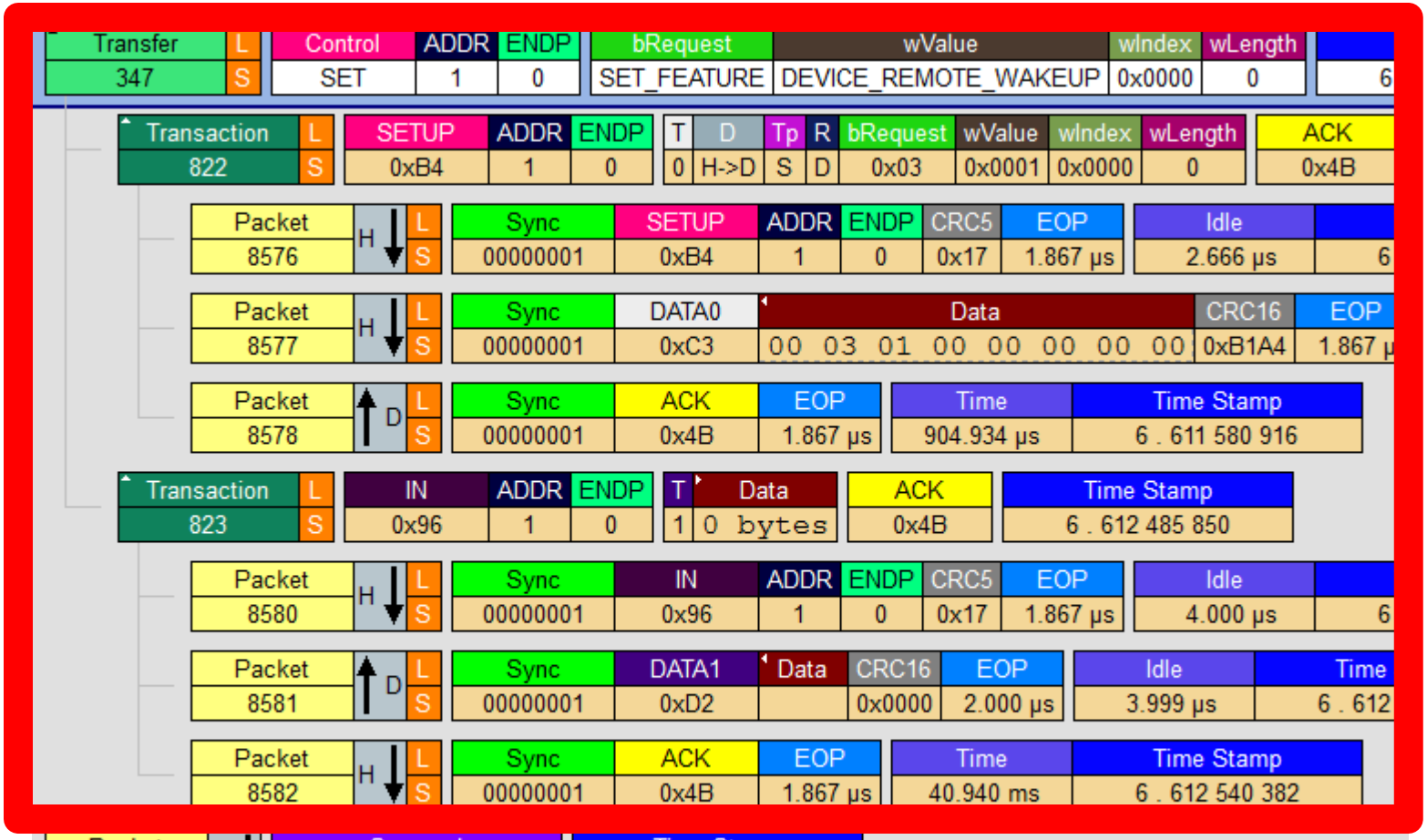
Packet		Resume	Time	Time Stamp
8622	?	1.506 sec	1.664 sec	20 . 816 025 932

Transfer		Control	ADDR	ENDP	bRequest	wValue	wIndex	wLength	Time
348	S	SET	1	0	CLEAR_FEATURE	0x0001	0x0000	0	3.000 ms

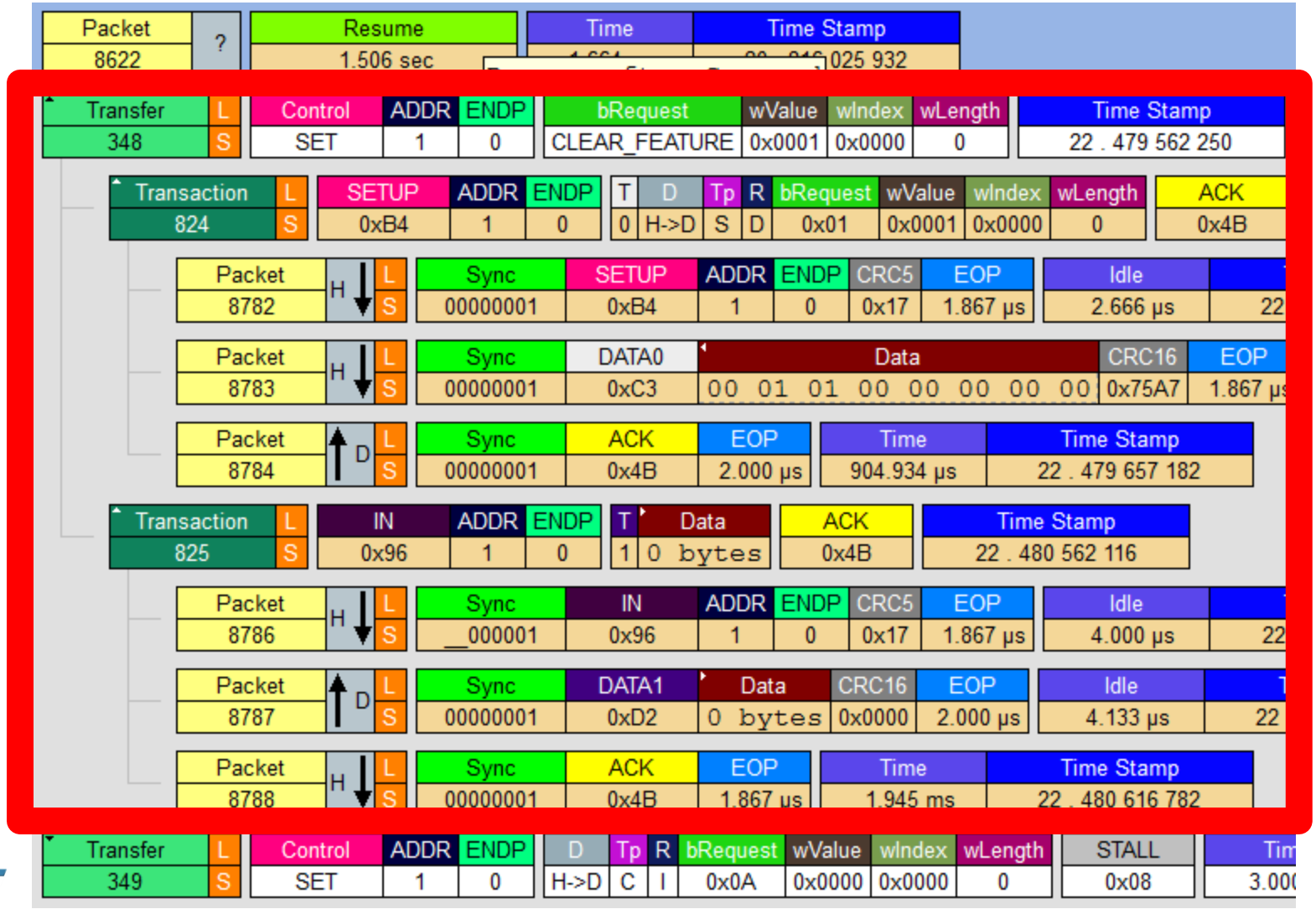
Transfer		Control	ADDR	ENDP	D	Tp	R	bRequest	wValue	wIndex	wLength	STALL
349	S	SET	1	0	H->D	C	I	0x0A	0x0000	0x0000	0	0x08
350	S	SET	1	0	H->D	C	I	0x0A	0x0000	0x0000	0	0x08

Transfer		Interrupt	ADDR	ENDP	Bytes Transferred	Time	Time Stamp
351	S	IN	1	1	4	8.000 ms	22 . 489 561 716
352	S	IN	1	1	4	8.000 ms	22 . 497 561 316
353	S	IN	1	1	4	8.000 ms	22 . 505 560 916

PC在休眠前使能设备的远程唤醒feature



PC被鼠标唤醒后关闭掉其远程唤醒feature



life.augmented