第五讲：USB主机操作HID类设备

USB技术应用与开发

演讲人：蔡 亮

CONTENTS

01

USB标准请求及描述符解析

USB主机传输

连接检测 → 主机枚举 USB设备 → 有效数据传输、功能控制等

枚举识别

控制传输（标准请求、类请求、厂商请求）

获取设备的各种信息，加载对应的USB类、功能驱动程序

选择当前要用的某个功能（启用配置、选择功能接口）

标准请求

Table 9-3. Standard Device Requests

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

ClearFeature:

清除设备、接口或端点的某种特征（或性能）

GetConfigration:

获取指定设备当前配置值

GetDescriptor:

获取设备某种标准描述符（经常用到）

GetInterface:

获取设备接口当前工作的选择设置值
（每个接口可有多个设置，当前只有一个有效）

GetStatus:

获取设备、接口或端点的某种状态

SetAddress:

为设备设置一个唯一的地址

SetConfiguration:

激活设备的某个配置（当前只有一个配置生效）

SetDescriptor:

为设备新增一个描述符或更新已存在的描述符

SetFeature:

主机启用一个在设备、接口或端点上的特征

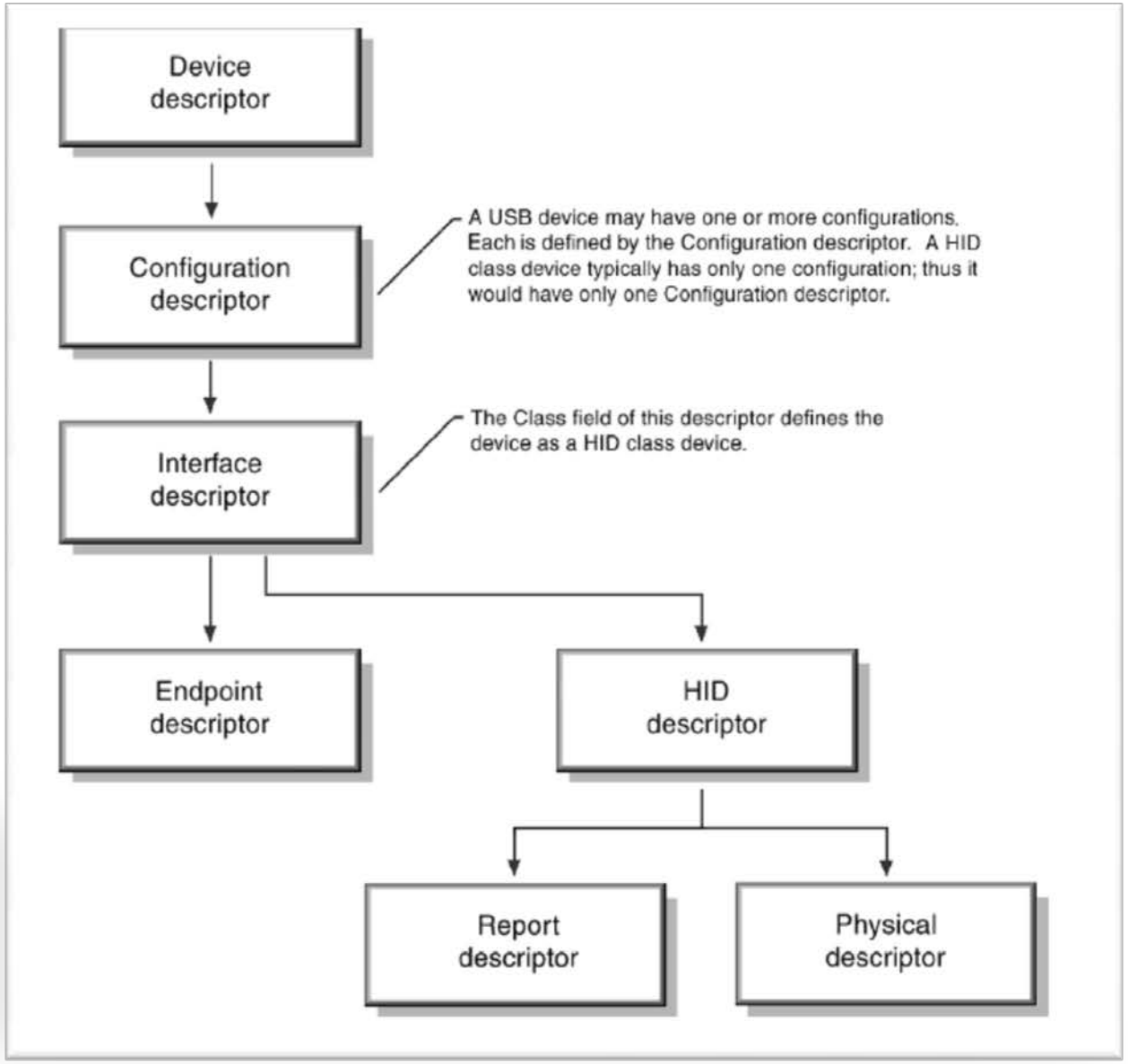
SetInterface:

主机激活设备的某个接口的设置值

SynchFrame:

在实时传输中，用于同步某个帧开始传输序列

HID设备类结构


Device descriptor
Configuration descriptor
A USB device may have one or more configurations. Each is defined by the Configuration descriptor. A HID class device typically has only one configuration; thus it would have only one Configuration descriptor.
Interface descriptor
The Class field of this descriptor defines the device as a HID class device.
Endpoint descriptor
HID descriptor
Report descriptor
Physical descriptor

主机枚举设备——获取设备描述符

0x80, 0x06, 0x00, 0x01, 0x00, 0x00, 0x12, 0x00　　主机请求——获取设备描述符

设备上传设备描述符内容

```
0x12,           // bLength
0x01,           // bDescriptorType
0x10,0x01,      // bcdUSB
0x00,           // bDeviceClass
0x00,           // bDeviceSubClass
0x00,           // bDeviceProtocol
0x08,           // bMaxPacketSize0
0x86,0x1a,      // idVendor
0xe1,0xe6,      // idProduct
0x00,0x01,      // bcdDevice
0x01,           // iManufacturer
0x02,           // iProduct
0x00,           // iSerialNumber
0x01            // bNumConfigurations
```

主机枚举设备——设置设备地址

0x00, 0x05, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00

主机请求——设置设备地址

- 此请求响应执行控制传输中没有数据阶段
- 地址设置范围 1~127
- 设备之后和主机通讯都使用此地址作为标识
- 当有总线复位或设备拔出时，此地址失效回到默认地址0

设备记录该地址

主机枚举设备——获取配置描述符

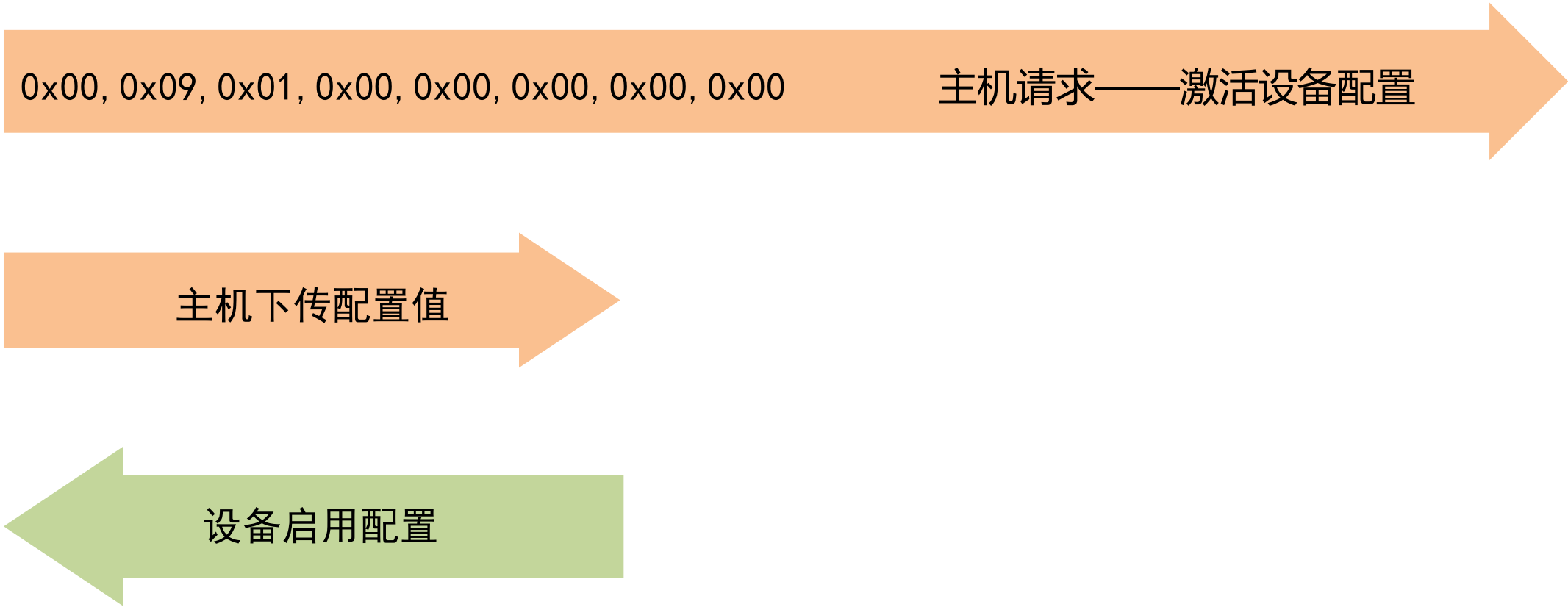
0x80, 0x06, 0x00, 0x02, 0x00, 0x00, 0x09, 0x00　　主机请求——获取配置描述符（集合）

设备上传配置描述符（集合）内容

```
/* 配置描述符 */
0x09,               // bLength
0x02,               // bDescriptorType
0x22,0x00,          // wTotalLength
0x01,               // bNumInterfaces
0x01,               // bConfigurationValue
0x00,               // iConfiguration
0xA0,               // bmAttributes
0x32,               // MaxPower
```

```
/* 配置描述符 */
0x09,               // bLength
0x02,               // bDescriptorType
0x22,0x00,          // wTotalLength
0x01,               // bNumInterfaces
0x01,               // bConfigurationValue
0x00,               // iConfiguration
0xA0,               // bmAttributes
0x32,               // MaxPower
/* 接口描述符，键盘功能 */
0x09,               // bLength
0x04,               // bDescriptorType
0x00,               // bInterfaceNumber
0x00,               // bAlternateSetting
0x01,               // bNumEndpoints
0x03,               // bInterfaceClass
0x01,               // bInterfaceSubClass
0x01,               // bInterfaceProtocol
0x00,               // iInterface
/* HID类描述符 */
0x09,               // bLength
0x21,               // bDescriptorType
0x10,0x01,          // bcdHID
0x00,               // bCountryCode
0x01,               // bNumDescriptors
0x22,               // bDescriptorType
0x3f,0x00,          // wDescriptorLength
/* 端点描述符 */
0x07,               // bLength
0x05,               // bDescriptorType
0x81,               // bEndpointAddress
0x03,               // bmAttributes
EP1_IN_SIZE, 0x00,  // wMaxPacketSize
0x0a,               // bInterval
```

主机枚举设备——激活设备配置



HID类设备主机常见类命令请求

0x21, 0x0A, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00　主机请求——设置HID设备上传速率

0x21, 0x09, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00　主机请求——下传报表（数据）

主机要下传的数据

主机针对HID类设备——获取报表描述符

0x81, 0x06, 0x00, 0x22, 0x00, 0x00, 0xFF, 0x00　　主机请求——获取报表描述符

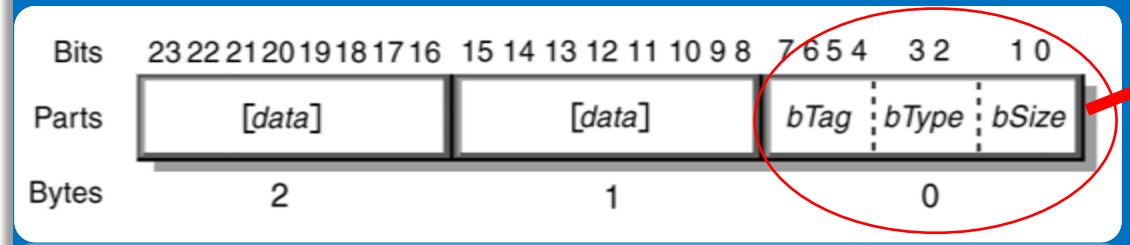
键盘设备上传报表描述符内容

```
0x05,0x01,              // Usage Page (Generic Desktop)
0x09,0x06,              // Usage (Keyboard/Keypad)
0xA1,0x01,              // Collection (Application)
    0x05,0x07,              //   Usage Page (Keyboard/Keypad)
    0x19,0xe0,              //   Usage Minimum..... (224)
    0x29,0xe7,              //   Usage Maximum..... (231)
    0x15,0x00,              //   Logical Minimum..... (0)
    0x25,0x01,              //   Logical Maximum..... (1)
    0x75,0x01,              //   Report Size..... (1)
    0x95,0x08,              //   Report Count..... (8)
    0x81,0x02,              //   Input.....(Data, Variable, Absolute)

    0x95,0x01,              //   Report Count..... (1)
    0x75,0x08,              //   Report Size..... (8)
    0x81,0x01,              //   Input.....(Constant)

    0x95,0x03,              //   Report Count..... (3)
    0x75,0x01,              //   Report Size..... (1)
    0x05,0x08,              //   Usage Page ( LEDs )
    0x19,0x01,              //   Usage Minimum..... (1)
    0x29,0x03,              //   Usage Maximum..... (3)
    0x91,0x02,              //   Output.....(Data, Variable, Absolute)

    0x95,0x05,              //   Report Count..... (5)
    0x75,0x01,              //   Report Size..... (1)
    0x91,0x01,              //   Output.....(Constant)

    0x95,0x06,              //   Report Count..... (6)
    0x75,0x08,              //   Report Size..... (8)
    0x15,0x00,              //   Logical Minimum..... (0)
    0x25,0x65,              //   Logical Maximum..... (101)
    0x05,0x07,              //   Usage Page (Keyboard/Keypad)
    0x19,0x00,              //   Usage Minimum..... (0)
    0x29,0x65,              //   Usage Maximum..... (101)
    0x81,0x00,              //   Input.....(Data, Array)
0xC0                    //    End Collection
```

报表描述符结构

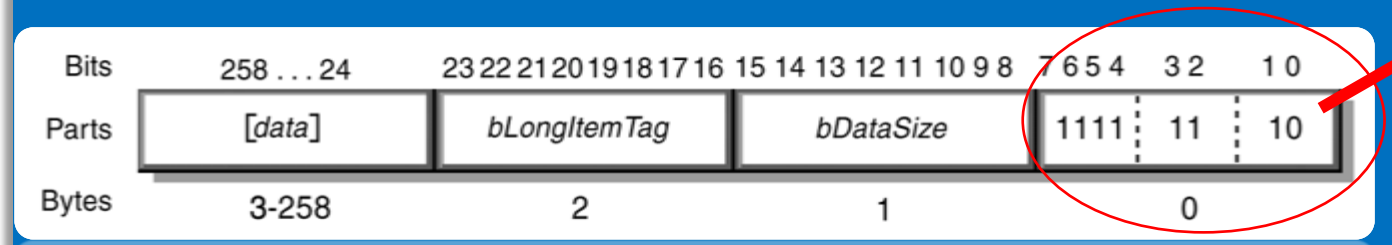
组成单位：条目（item）。

短条目



Part	Description
bSize	Numeric expression specifying size of data: 0 = 0 bytes 1 = 1 byte 2 = 2 bytes 3 = 4 bytes
bType	Numeric expression identifying type of item where: 0 = Main 1 = Global 2 = Local 3 = Reserved
bTag	Numeric expression specifying the function of the item.
[*data*]	Optional data.

长条目



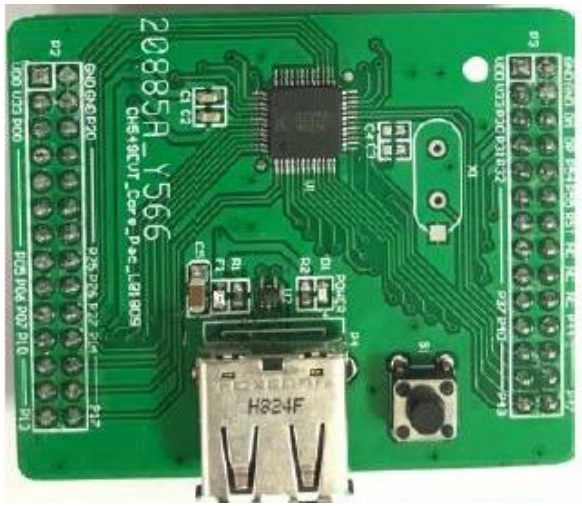
Part	Description
bSize	Numeric expression specifying total size of item where size is 10 (2 bytes); denotes item type as long.
bType	Numeric expression identifying type of item where 3 = Reserved
bTag	Numeric expression specifying the function of the item; always 1111.
[*bDataSize*]	Size of long item data.
[*bLongItemTag*]	Long item tag.
[*data*]	Optional data items.

02

软件实现
及效果演示

硬件平台

应用：以CH549为例，作为USB主机，识别接入的USB设备，并找出HID类设备获取有效数据



供电：3.3V和5V供电系统皆可

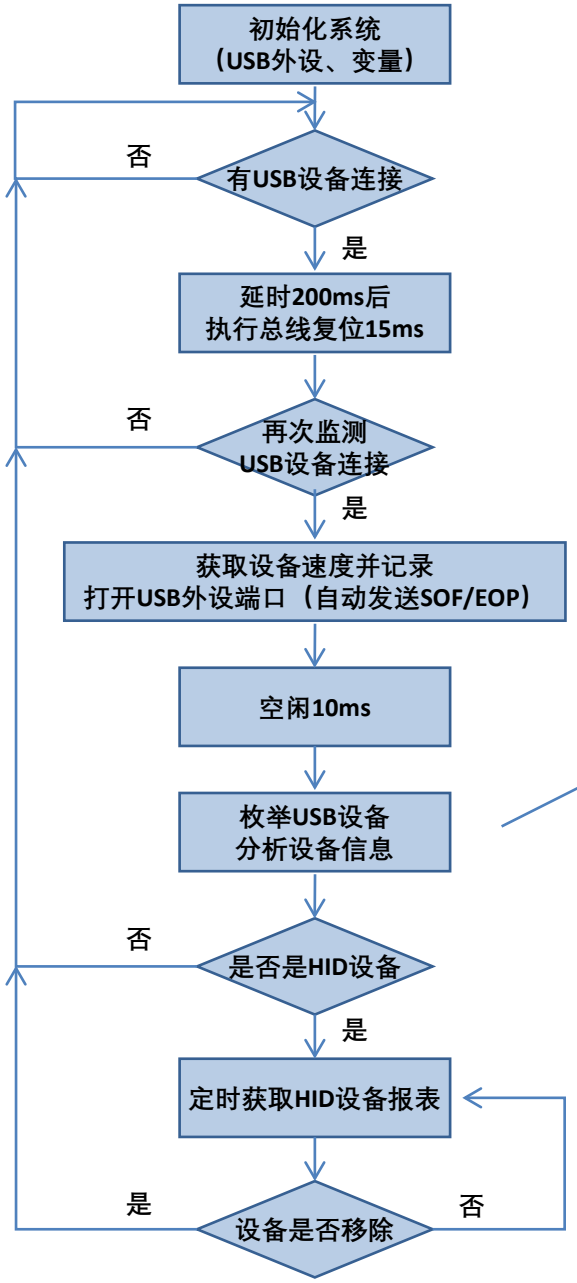
最简外围：只需2个退耦电容

USB功能：USB主设备或者USB从设备

下载方式：串口或者USB口下载

Part NO.		Freq/Max	Flash	RAM	DataFlash	USB	TouchKey	Type-C	ADC	LEDC	Timer	CAP	PWM	UART	SPI	I/O	Built-in OSC/WDOG	VDD/V	Package
CH549	L	12/48MHz	63K	2K+256	1K	1*H/1*D	16	√	16*12b	-	3*16b	3	8	4	1	44	√/√	3.3/5	LQFP48
	F	12/48MHz	63K	2K+256	1K	1*H/1*D	10	√	10*12b	-	3*16b	2	5	4	1	25	√/√	3.3/5	QFN28
	G	12/48MHz	63K	2K+256	1K	1*H/1*D	5	√	5*12b	-	3*16b	2	3	2	1	13	√/√	3.3/5	SOP16

软件框架

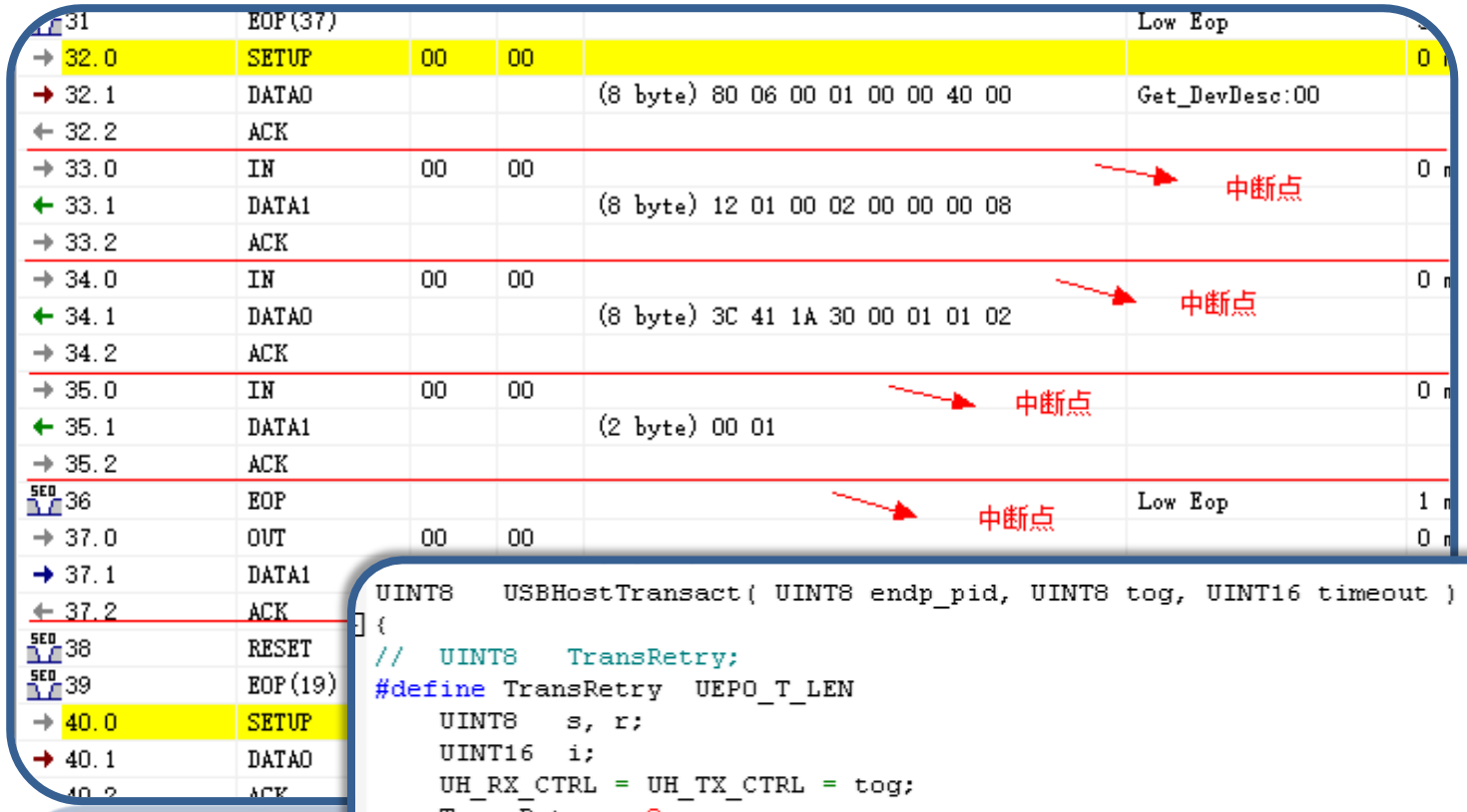


获取设备描述符
设置USB地址
获取配置描述符
设置配置
SetIdle
SetReport（针对HID键盘）
获取报表描述符

保存信息包括：
1、设备端点0最大包长度
2、设备地址
3、设备类、子类、协议号
4、对于HID，是否支持远程唤醒
5、对于HID，接口号、端点号、端点大小，报表长度

```
// 执行基本事务传输函数
    USBHostTransact(X...)

// 执行控制传输函数
    HostCtrlTransfer(X...)
```

软件分析



31	EOP(37)				Low Eop	
32.0	SETUP	00	00			0
32.1	DATA0			(8 byte) 80 06 00 01 00 00 40 00	Get_DevDesc:00	
32.2	ACK					
33.0	IN	00	00			0
33.1	DATA1			(8 byte) 12 01 00 02 00 00 00 08		
33.2	ACK					
34.0	IN	00	00			0
34.1	DATA0			(8 byte) 3C 41 1A 30 00 01 01 02		
34.2	ACK					
35.0	IN	00	00			0
35.1	DATA1			(2 byte) 00 01		
35.2	ACK					
36	EOP				Low Eop	1
37.0	OUT	00	00			0
37.1	DATA1					
37.2	ACK					
38	RESET					
39	EOP(19)					
40.0	SETUP					
40.1	DATA0					

中断点

中断点

中断点

中断点

```
UINT8   USBHostTransact( UINT8 endp_pid, UINT8 tog, UINT16 timeout )
{
//  UINT8   TransRetry;
#define TransRetry  UEP0_T_LEN
    UINT8   s, r;
    UINT16  i;
    UH_RX_CTRL = UH_TX_CTRL = tog;
    TransRetry = 0;

    do {
        UH_EP_PID = endp_pid;
        UIF_TRANSFER = 0;
//  s = WaitUSB_Interrupt( );
        for ( i = WAIT_USB_TOUT_200US; i != 0 && UIF_TRANSFER == 0; i -- );
        UH_EP_PID = 0x00;
//  if ( s != ERR_SUCCESS ) return( s );  // 中断超时,可能是硬件异常
        if ( UIF_TRANSFER == 0 ) return( ERR_USB_UNKNOWN );
```

- 主机发出事务令牌包SETUP+数据包
  设备给出事务应答包ACK
- 主机发出事务令牌包OUT+数据包
  设备给出事务应答包
  ACK/NAK/STALL
- 主机发出事务令牌包IN
  设备给出事务数据包
  主机给出事务应答包

03

知识梳理

USB2.0主机的工作

- 检测到设备的插入动作、移除动作
- 分析设备类别，启用对应的传输协议
- 控制总线状态（复位、挂起…）
- 计算分配总线上的传输、事务带宽
- 所有的事务令牌包都由主机先发起，

  设备给出事务中的数据包或应答包

Thank you

感谢观赏

微信公众号

http://wch.cn

tech@wch.cn

025-84730668

控制传输——*Setup Stage*

控制传输中的 Setup Stage 部分（Setup事务），主机发出8字节命令请求，格式如下：

大小	1byte	1byte	2byte	2byte	2byte
字段	bRequestType	bRequest	wValue	wIndex	wLength

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

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

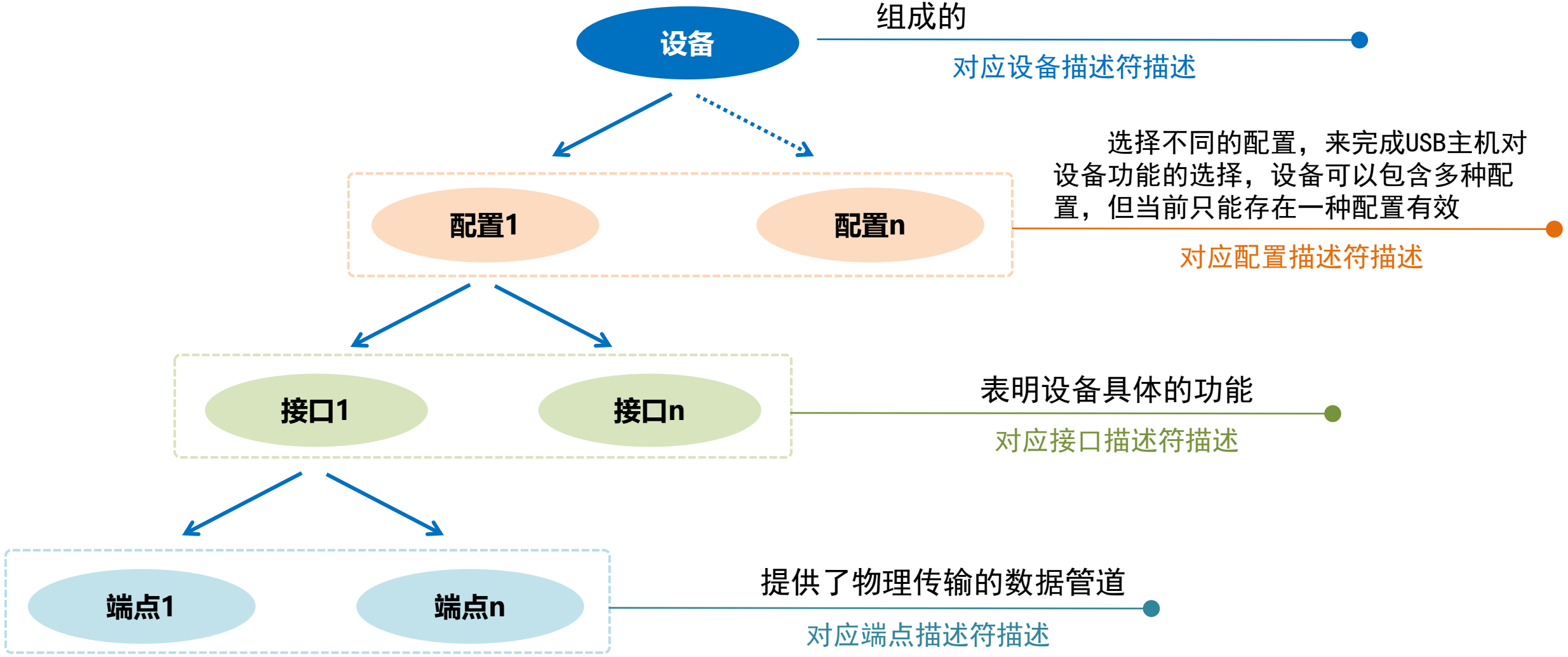
标准请求码

根据bRequest改变，一般作为参数作用

一般表示接口号或端点号

在 Data Stage 阶段的总数据最大字节数

USB从设备



USB设备是由一些配置、接口、端点组成的

对应设备描述符描述

选择不同的配置，来完成USB主机对设备功能的选择，设备可以包含多种配置，但当前只能存在一种配置有效

对应配置描述符描述

表明设备具体的功能

对应接口描述符描述

提供了物理传输的数据管道

对应端点描述符描述

报表描述符结构

6.2.2.4 Main Items

Main items are used to either define or group certain types of data fields within a **Report** descriptor. There are two types of **Main** items: data and non-data. Data-type **Main** items are used to create a field within a report and include **Input**, **Output**, and **Feature**. Other items do not create fields and are subsequently referred to as non-data **Main** items.

Main item tag	One-Byte Prefix (*nn* represents size value)	Valid Data	
Input	1000 00 *nn*	Bit 0	{Data (0) | Constant (1)}
		Bit 1	{Array (0) | Variable (1)}
		Bit 2	{Absolute (0) | Relative (1)}
		Bit 3	{No Wrap (0) | Wrap (1)}
		Bit 4	{Linear (0) | Non Linear (1)}
		Bit 5	{Preferred State (0) | No Preferred (1)}
		Bit 6	{No Null position (0) | Null state(1)}
		Bit 7	Reserved (0)
		Bit 8	{Bit Field (0) | Buffered Bytes (1)}
		Bit 31-9	Reserved (0)
Output	1001 00 *nn*	Bit 0	{Data (0) | Constant (1)}
		Bit 1	{Array (0) | Variable (1)}
		Bit 2	{Absolute (0) | Relative (1)}
		Bit 3	{No Wrap (0) | Wrap (1)}
		Bit 4	{Linear (0) | Non Linear (1)}
		Bit 5	{Preferred State (0) | No Preferred (1)}
		Bit 6	{No Null position (0) | Null state(1)}
		Bit 7	{Non Volatile (0) | Volatile (1)}
		Bit 8	{Bit Field (0) | Buffered Bytes (1)}
		Bit 31-9	Reserved (0)
Feature	1011 00 *nn*	Bit 0	{Data (0) | Constant (1)}
		Bit 1	{Array (0) | Variable (1)}
		Bit 2	{Absolute (0) | Relative (1)}
		Bit 3	{No Wrap (0) | Wrap (1)}
		Bit 4	{Linear (0) | Non Linear (1)}
		Bit 5	{Preferred State (0) | No Preferred (1)}
		Bit 6	{No Null position (0) | Null state(1)}
		Bit 7	{Non Volatile (0) | Volatile (1)}
		Bit 8	{Bit Field (0) | Buffered Bytes (1)}
		Bit 31-9	Reserved (0)
Collection	1010 00 *nn*	0x00	Physical (group of axes)
		0x01	Application (mouse, keyboard)
		0x02	Logical (interrelated data)
		0x03	Report
		0x04	Named Array
		0x05	Usage Switch
		0x06	Usage Modifier
		0x07-0x7F	Reserved
		0x80-0xFF	Vendor-defined
End Collection	1100 00 *nn*	Not applicable. Closes an item collection.	

6.2.2.7 Global Items

Global items describe rather than define data from a control. A new **Main** item assumes the characteristics of the item state table. **Global** items can change the state table. As a result **Global** item tags apply to all subsequently defined items unless overridden by another **Global** item.

Global item tag	One-Byte Prefix (*nn* represents size value)	Description
Usage Page	0000 01 *nn*	Unsigned integer specifying the current Usage Page. Since a usage are 32 bit values, Usage Page items can be used to conserve space in a report descriptor by setting the high order 16 bits of a subsequent usages. Any usage that follows which is defines 16 bits or less is interpreted as a Usage ID and concatenated with the Usage Page to form a 32 bit Usage.
Logical Minimum	0001 01 *nn*	Extent value in logical units. This is the minimum value that a variable or array item will report. For example, a mouse reporting x position values from 0 to 128 would have a Logical Minimum of 0 and a Logical Maximum of 128.
Logical Maximum	0010 01 *nn*	Extent value in logical units. This is the maximum value that a variable or array item will report.
Physical Minimum	0011 01 *nn*	Minimum value for the physical extent of a variable item. This represents the Logical Minimum with units applied to it.
Physical Maximum	0100 01 *nn*	Maximum value for the physical extent of a variable item.
Unit Exponent	0101 01 *nn*	Value of the unit exponent in base 10. See the table later in this section for more information.
Unit	0110 01 *nn*	Unit values.
Report Size	0111 01 *nn*	Unsigned integer specifying the size of the report fields in bits. This allows the parser to build an item map for the report handler to use. For more information, see Section 8, Report Protocol.

6.2.2.8 Local Items

Local item tags define characteristics of controls. These items do not carry over to the next **Main** item. If a **Main** item defines more than one control, it may be preceded by several similar **Local** item tags. For example, an **Input** item may have several **Usage** tags associated with it, one for each control.

Tag	One-Byte Prefix (*nn* represents size value)	Description
Usage	0000 10 *nn*	Usage index for an item usage; represents a suggested usage for the item or collection. In the case where an item represents multiple controls, a Usage tag may suggest a usage for every variable or element in an array.
Usage Minimum	0001 10 *nn*	Defines the starting usage associated with an array or bitmap.
Usage Maximum	0010 10 *nn*	Defines the ending usage associated with an array or bitmap.
Designator Index	0011 10 *nn*	Determines the body part used for a control. Index points to a designator in the Physical descriptor.
Designator Minimum	0100 10 *nn*	Defines the index of the starting designator associated with an array or bitmap.
Designator Maximum	0101 10 *nn*	Defines the index of the ending designator associated with an array or bitmap.
String Index	0111 10 *nn*	String index for a String descriptor; allows a string to be associated with a particular item or control.
String Minimum	1000 10 *nn*	Specifies the first string index when assigning a group of sequential strings to controls in an array or bitmap.
String Maximum	1001 10 *nn*	Specifies the last string index when assigning a group of sequential strings to controls in an array or bitmap.
Delimiter	1010 10 *nn*	Defines the beginning or end of a set of local items (1 = open set, 0 = close set).
Reserved	1010 10 *nn* to 1111 10 *nn*	Reserved.

报表描述符条目解析器（主机HID驱动）