

HID+CDC 复合设备在 WIN10 的识别问题

1 问题现象

有一客户使用 STM32F405 参照 USB 标准库下的 HID+CDC 的示例代码做产品，发现在 WIN7 上使用得好好的，可放到 WIN10 上，CDC 第一次能够识别，再次拔插后就不能再识别，且此后无论插拔多少次都无法再识别，除非再次上电，又会重复上述现象，只有板子上电后第一次才能正确被识别，后续均不行。

2 问题分析

客户使用 ST 官方示例代码 STM32_USB-Host-

Device_Lib_V2.2.0\Project\USB_Device_Examples\Composite_Examples\CDC_HID_Composite（下载页面：

http://www.stmicroelectronics.com.cn/content/st_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32-standard-peripheral-library-expansion/stsw-stm32046.html）。当我尝试使用此示例代码重现客户所遇到的问题时，发现此代码在 WIN7 运行 OK，但与客户不同的是，我测试到的情况是在 WIN10 下 CDC 一次都无法识别，HID 却一直可以识别。

下面来分析下问题，既然 WIN7 下 HID 和 CDC 都能正常识别，放在 WIN10 上才不正常，那么初步可以判断，此问题应该与 WIN10 操作系统的 USB 主机驱动实现有关。

通过 USB 分析仪分析客户代码在 WIN10 下 USB 枚举异常的数据通讯：

Sp	Index	ms.ms.us	Len	Err	Dev	Ep	Record	Summary	Alt
FS	109	0:20.827.734	36 B		05	00	Get String Descriptor	Index=2 Length=255	
FS	123	0:20.827.918	0 B		05	00	Control Transfer (STALL)	Index=0 Length=10	
FS	132	0:20.828.195	90.0 ms				[91 SOF]	[Frames: 1220 - 1310]	
FS	133	0:20.918.371	18 B		05	00	Get Device Descriptor	Index=0 Length=18	
FS	147	0:20.918.497	9 B		05	00	Get Configuration Descriptor	Index=0 Length=9	
FS	161	0:20.918.653	100 B		05	00	Get Configuration Descriptor	Index=0 Length=100	
FS	180	0:20.919.208	21.0 ms				[22 SOF]	[Frames: 1311 - 1332]	
FS	181	0:20.937.176	0 B		05	00	Set Configuration	Configuration=1	
FS	191	0:20.941.211	1.00 ms				[2 SOF]	[Frames: 1333 - 1334]	
FS	192	0:20.942.638	4 B		05	00	Get String Descriptor	Index=2 Length=4	
FS	206	0:20.942.803	36 B		05	00	Get String Descriptor	Index=2 Length=36	
FS	220	0:20.943.212	63.0 ms				[64 SOF]	[Frames: 1335 - 1398]	
FS	221	0:21.006.931	4 B		05	00	Get String Descriptor	Index=1 Length=4	
FS	235	0:21.007.040	36 B		05	00	Get String Descriptor	Index=1 Length=36	
FS	249	0:21.007.221	53.0 ms				[64 SOF]	[Frames: 1399 - 1452]	
FS	250	0:21.060.433	0 B		05	00	Set Idle	Duration=Indefinite Report=0	
FS	260	0:21.060.551	74 B		05	00	Get Report Descriptor	Index=0 Length=138	
FS	279	0:21.061.228	68.0 ms				[68 SOF]	[Frames: 1453 - 1521]	
FS	280	0:21.126.974	2 B		05	00	Get Device Status		
FS	294	0:21.130.238	2.08 us				[1 SOF]	[Frame: 1522]	
FS	295	0:21.130.556	7 B		05	00	Get Line Coding	115200bps 8N1	
FS	309	0:21.130.722	0 B		05	00	Set Control Line State		
FS	319	0:21.130.851	7 B		05	00	Set Line Coding	115200bps 8N1	
FS	338	0:21.131.008	7 B		05	00	Get Line Coding	115200bps 8N1	
FS	352	0:21.080.234	1.99 s		05	03	[250 IN-NAK]	[Periodic Timeout]	
FS	353	0:21.131.238	1.99 s				[2000 SOF]	[Frames: 1523 - 1474] [Periodic Timeout]	
FS	354	0:23.080.518	1.99 s		05	03	[250 IN-NAK]	[Periodic Timeout]	
FS	355	0:23.131.522	1.99 s				[2000 SOF]	[Frames: 1475 - 1426] [Periodic Timeout]	

Figure 1 第一次 USB 枚举过程

上图是客户代码第一次正常枚举的通讯数据，从中可以看出，WIN10 USB 主机在正常获取 HID 报告描述符后，紧接着会获取虚拟串口状态和设置波特率，这样就正常枚举结束了。接着继续采集异常 USB 枚举过程进行对比：

FS #	Time	Duration	Address	Length	Device	Operation	Details
FS # 367	0:32:754.523					<Full-speed>	
FS # 368	0:32:754.864	15.0 ms				[16 SOF]	[Frames: 60 - 75]
FS # 369	0:32:770.597	18 B	00	00		Get Device Descriptor	Index=0 Length=64
FS # 383	0:32:770.828	54.8 ms				<Reset> / <Target disconnected>	
FS # 384	0:32:825.713					<Full-speed>	
FS # 385	0:32:825.874	15.0 ms				[16 SOF]	[Frames: 131 - 145]
FS # 388	0:32:841.721	0 B	00	00		SetAddress	Address=06
FS # 396	0:32:841.877	12.0 ms				[13 SOF]	[Frames: 147 - 159]
FS # 397	0:32:851.897	18 B	05	00		Get Device Descriptor	Index=0 Length=18
FS # 411	0:32:854.878	6.00 ms				[7 SOF]	[Frames: 160 - 166]
FS # 412	0:32:861.402	100 B	06	00		Get Configuration Descriptor	Index=0 Length=255
FS # 431	0:32:861.706	26 B	08	00		Get String Descriptor	Index=3 Length=255
FS # 445	0:32:861.879	2.08 us				[1 SOF]	[Frame: 167]
FS # 446	0:32:861.891	4 B	06	00		Get String Descriptor	Index=0 Length=255
FS # 460	0:32:862.062	36 B	06	00		Get String Descriptor	Index=2 Length=255
FS # 474	0:32:862.249	0 B	06	00		Control Transfer (STALL)	Index=0 Length=10
FS # 483	0:32:862.879	237 ms				[238 SOF]	[Frames: 188 - 405]
FS # 484	0:33:099.849	18 B	06	00		Get Device Descriptor	Index=0 Length=18
FS # 498	0:33:100.062	9 B	06	00		Get Configuration Descriptor	Index=0 Length=9
FS # 512	0:33:100.184	100 B	06	00		Get Configuration Descriptor	Index=0 Length=100
FS # 531	0:33:100.913	2.08 us				[1 SOF]	[Frame: 406]
FS # 532	0:33:103.915	246 ms		T		<Suspend>	
FS # 533	7:24:625.985	18.5 us				<Reset> / <Chrp J> / <Tiny J>	
FS # 534	7:24:626.004	164 ms		T		<Reset> / <Target disconnected>	
FS # 535	7:26:530.652					<Full-speed>	
FS # 536	7:26:533.652	129 ms				<Suspend>	
FS # 537	7:26:563.646	54.8 ms				<Reset> / <Chrp J> / <Tiny J>	
FS # 538	7:26:718.530					<Full-speed>	

Figure 2 异常枚举过程

上图是 WIN10 下异常枚举过程。由上图可以看出，WIN10 系统上 USB 主机在获取到设备描述符和配置描述符后直接将设备挂起了。很明显，WIN10 操作系统的 USB 主机驱动实现对设备描述符或者配置描述符的内容并不认可，才会导致无法识别 HID+CDC 复合设备。

于是，我们首先检查客户代码的设备描述符：

Device Descriptor		Radix: auto
bLength	18	
bDescriptorType	DEVICE (0x01)	
bcdUSB	2.00 (0x0200)	
bDeviceClass	Miscellaneous (0xef)	
bDeviceSubClass	Common (0x02)	
bDeviceProtocol	Interface Association Descriptor (0x01)	
bMaxPacketSize0	64	
idVendor	0x0483	
idProduct	0x3256	
bcdDevice	2.00 (0x0200)	
iManufacturer	Not Requested (1)	
iProduct	Not Requested (2)	
iSerialNumber	Not Requested (3)	
bNumConfigurations	1	

Figure 3 获取的设备描述符

复合设备的 class,subclass,protocol 必须为 0xef,0x02,0x01,这里 VID=0x0483,PID=0x3256(Cube 库下为 0x5740,但这个不重要),接下来看配置描述符:

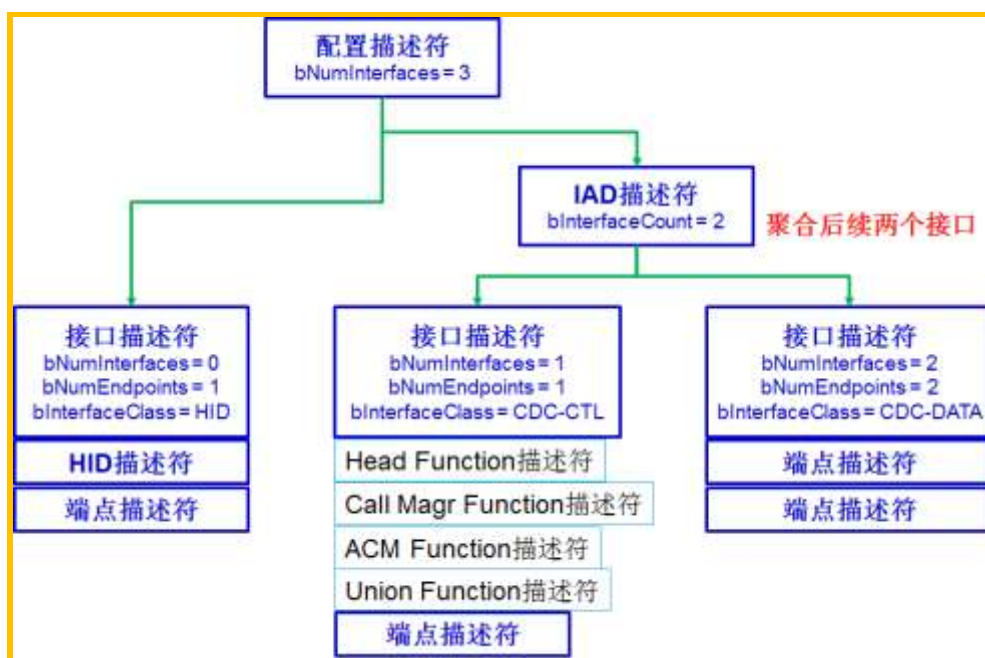


Figure 4 win10 不能识别的配置描述符

由此可见，客户的描述符是 HID interface + IAD + CDC interfaces 结构。对于 WIN7，这种结构可以识别，但对于 WIN10，这种结构 WIN10 未必能够兼容，我们尝试在 HID interface 外部加上一层 IAD 结构，使其成为 IAD1 + HID interface + IAD2 + CDC interfaces 结构，此时客户的问题得以解决，在 WIN10 也可以正确识别了，修改后的描述符结构如下：

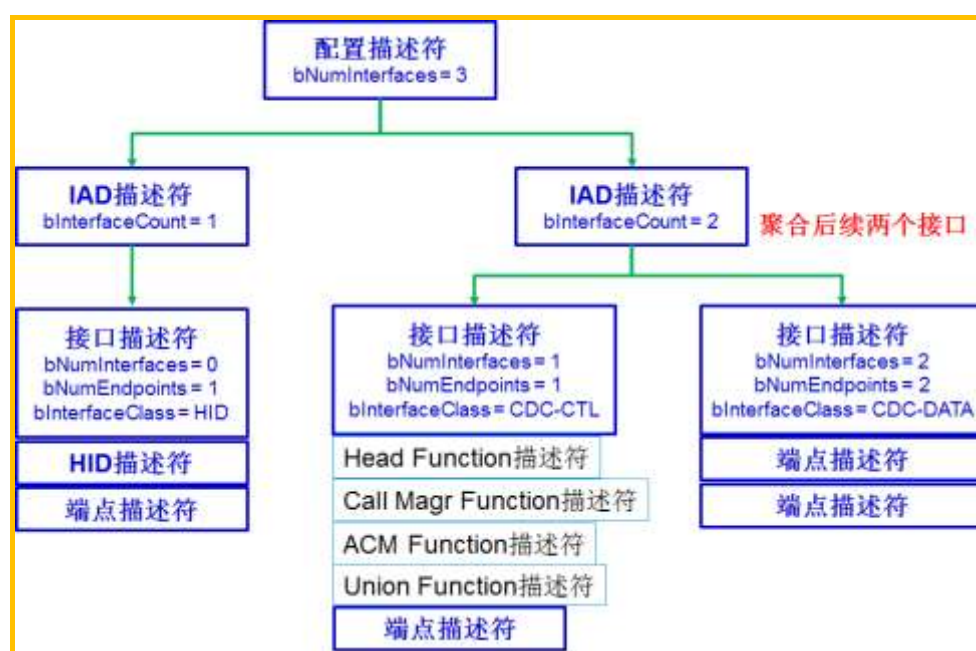


Figure 5 win10 能够正确识别的配置描述符

结束本篇实战经验之前，让我们再次回顾 IAD 的概念：

IAD（Interface Association Descriptor），为 USB 设备定义了一个标准来表述捆绑在一个逻辑功能（比如这里的 CDC 虚拟串口）上的多个接口的聚合的方法。USB 协会分配了一个设备级别的类编码（即图 3 中 0xEF），使用 IAD 的设备必须使用它（如图 3 的设备描述符）；这样可以很容易在设备枚举时就能识别出采用了 IAD 的设备。IAD 描述符通常放在它所捆绑的多个接口的接口描述符之前。

3 结论

在 WIN10 系统中，建议复合设备每个逻辑功能的接口描述符前都搭载一个 IAD 描述符，不论这个逻辑功能是单个接口描述符完成（比如这里的 HID 功能）还是要由多个接口描述符完成（比如这里的 CDC 功能）。

重要通知 – 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。