USB 的控制传输详解

1 USB 插入检测与枚举

当一个 USB 设备插入到接口上时，主机首先进行**插入检测**，检测完毕之后，设备采用 **0 号端点**，以**缺省地址**与主机进行**控制传输**。进行传输的这条通道叫做**控制通道**。

主机完成插入检测之后，对设备进行枚举，以了解设备，并加载其驱动。这时，主机希望了解 USB 设备的情况，需要知道它是一个什么设备，是一个大容量存储设备，如优盘，还是一个 USB 鼠标，它的数据传输能力怎么，这些要通过获得一系列的描述符来得到。这些描述符包括设备描述符、配置描述符、接口描述符等。想要获得**一个设备描述符**，就要进行**一次控制传输**。注意，上面这句话里所提到的“一次控制传输”是有特殊含义的。“传输”是的一个专用的名词，一个传输通常由多个“阶段”组成，根据传输类型的不同，包含的阶段也不同。一个“阶段”通包含一个或多个“事务”，以这段话中加引号的都是专用词汇，更多及详情请参：http://www.usr.cc/thread-51674-1-1.html

1.1 控制传输的分类：

控制传输的用途是获得设备信息、对设备进行配置。其基本的操作就是①**控制读传输**和②**控制写传输**，此外还有一些情况不涉及数据的传输，即称为③**无数据控制传输**。以上三个词汇对应的英文表述是“Control Write”、“Control Read”和“No-data Control”。

1.2 控制传输的总体格式

控制读传输包含三个**阶段:建立阶段**、**数据阶段**和**状态阶段**。

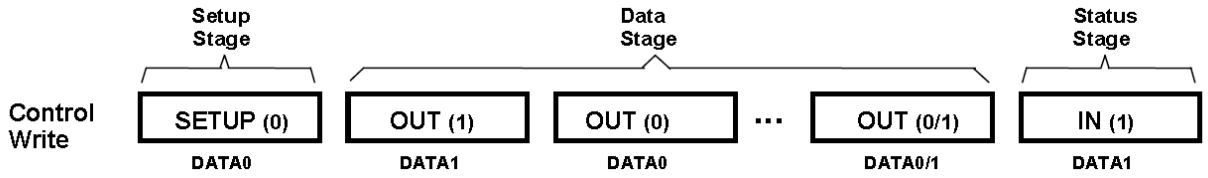
建立阶段只有一个**建立事务**。建立事务参：http://www.usr.cc/thread-51675-1-1.html

数据阶段有一个或多个 IN 或 OUT **事务**。这里之所以会有多个数据**事务**，是因为一个事务可能传不完所有的数据，这时才需要多个事务。从而也决定了，这多个事务是同类型的，一个是 IN,其它的就都是 IN.反之亦然。比如建立事务表明要获得**设备描述符**，那么这个数据阶段就都是 IN 事务，从设备得到描述符。

状态阶段有一个与**数据阶段**相反的事务。如果**数据阶段**是 IN,从设备读取了数据，那么这个阶段就是一个 OUT 事务，从主机发给设备数据，告诉设备读取数据是否收到了。相反，如果数据阶段是 OUT 事务，主机向设备传数据了，那么状态阶段就是一个 IN 事务，设备告诉主机数据是否收到。

1.3 控制写传输

基于 1.2 中的描述，控制写传输的格式如下：



第一个阶段是建立阶段，其中包含一个 8 字分的 DATA0 数据包。

第二个阶段是数据阶段，里面全是 OUT 事务。值得注意的是，这里每个事务的数据包是 DATA0 包和 DATA1 包轮流出现的。SETUP 事务中的数据包是 DATA0,OUT 事务中的依次为 DATA1,DATA0,DATA1….。数据阶段出现多个 OUT 事务是因为一个数据包可能无法传送完所有的数据。因此拆分成多个数据多，在多个事务中进行传送。

这里最后一个事务有讲究：如果最后一个事务传的数据小于允许的最大数据包大小（这是通常的情况）则主机认为数据的传输结束了。如果最后一个包恰好也是允许的最大数据包大小，即所要传的数据总量正好是允许传的最大数据包大小的整数倍，则要再传一个 0 数据包的 OUT 事务，来暗示数据传输的结束。

第三个阶段是状态阶段，这个阶段是一个 IN 事务。其中的数据包恒是 DATA1 数据包，不与上面轮换的。

状态阶段讲究更多一些，因为这一阶段要返回数据传输成功与否。具体如下：

1. 写数据成功　　主机发送 IN 令牌包，**设备回复 0 长度数据包**，主机 ACK.
2. 数据传送出错　　主机发送 IN 令牌包，**设备回复 STALL**.
3. 设备忙(比如正在写数据)　　主机发送 IN 令牌包，**设备 NAK.** 这时主机应该继续发送 IN 令牌包，持续状态阶段.

上段文字中，第二个相(phase)被加精显示，表示设备决定控制写操作成功与否。

1.4 控制读传输

基于 1.2 中的描述，控制读传输的格式如下：

第一个阶段是**建立阶段**，其中包含一个 8 字分的 DATA0 数据包。

第二个阶段是**数据阶段**，里面全是 IN 事务。值得注意的是，这里每个事务的数据包是 DATA0 包和 DATA1 包轮流出现的。SETUP 事务中的数据包是 DATA0,IN 事务中的依次为 DATA1,DATA0,DATA1….。数据阶段出现多个 IN 事务是因为一个数据包可能无法传送完所有的数据。因此拆分成多个数据多，在多个事务中进行传送。

这里最后一个事务有讲究：如果最后一个事务传的数据小于允许的最大数据包大小（这是通常的情况）则主机认为数据的传输结束了。如果最后一个包恰好也是允许的最大数据包大小，即所要传的数据总量正好是允许传的最大数据包大小的整数倍，则要再传一个 0 数据包的 IN 事务，来暗示数据传输的结束。

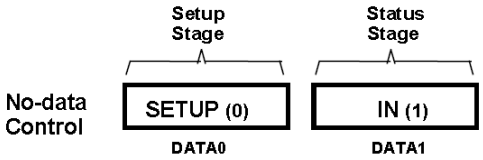
第三个阶段是**状态阶段**，这个阶段是一个 OUT 事务。其中的数据包恒是 DATA1 数据包，不与上面轮换的。

状态阶段讲究更多一些，因为这一阶段要返回数据传输成功与否。具体如下：

1. 读数据成功　　主机发送 OUT 令牌包(PING 令牌包,高速情况下)，主机发送 0 长度数据包，**设备 ACK**.
2. 数据传送出错　　主机发送 OUT 令牌包(PING 令牌包,高速情况下)，主机发送 0 长度数据包，**设备 STALL**.
3. 设备忙(比如正在写数据)　　主机 OUT 或(PING 令牌包,高速情况)，主机发送 0 长度数据包，**设备 NAK**. 这时主机应该继续发送 IN 令牌包，持续状态阶段。

1.5　无数据控制

基于 1.2 中的描述，无数据控制的格式如下：



控制并不一定需要很多数传输的，有些控制可能只是告诉诉设备要做一件事，这个命令只有包含在建立阶段的建立事务的 8 字节数据中就可以了，而设备只要告诉主机收到命令与否就可以，所以这种传输不需要数据阶段，直接进入状态阶段。

第一个阶段是**建立阶段**，其中包含一个 8 字分的 DATA0 数据包。

第二个阶段是**状态阶段**，这个阶段是一个 IN 事务。其中的数据包恒是 DATA1 数据包。其讲究与控制写传输相似。

1.6　USB 的枚举

USB 插入后，主机进行总线复位，然后就进入枚举阶段：

1. 第一次获取设备描述符

   这次获取设备描述符只需获取前 8 个字节就可以，不需获得全部。而一般来说设备一次传数据包至少也是 8 字节的，所以此次控制传输的数据阶段只有一个 IN 事务。设备描述符总共计 18 字节，如果设备的缓存小于 18 字节，则会以最大缓存的长度发来一个数据包，而主机接到这个包后就不再发出 IN 事务，直接进入状态阶段。

2. Set Address

主要第一次获取设备描述符成功之后，就知道设备是真实存在的了，然后会分配给设备一个地址，并进行第二个控制传输，为设备设定这个地址。

3. 第二次获取设备描述符

这次是获得全部设备描述符，并且是从新地址来获得的。

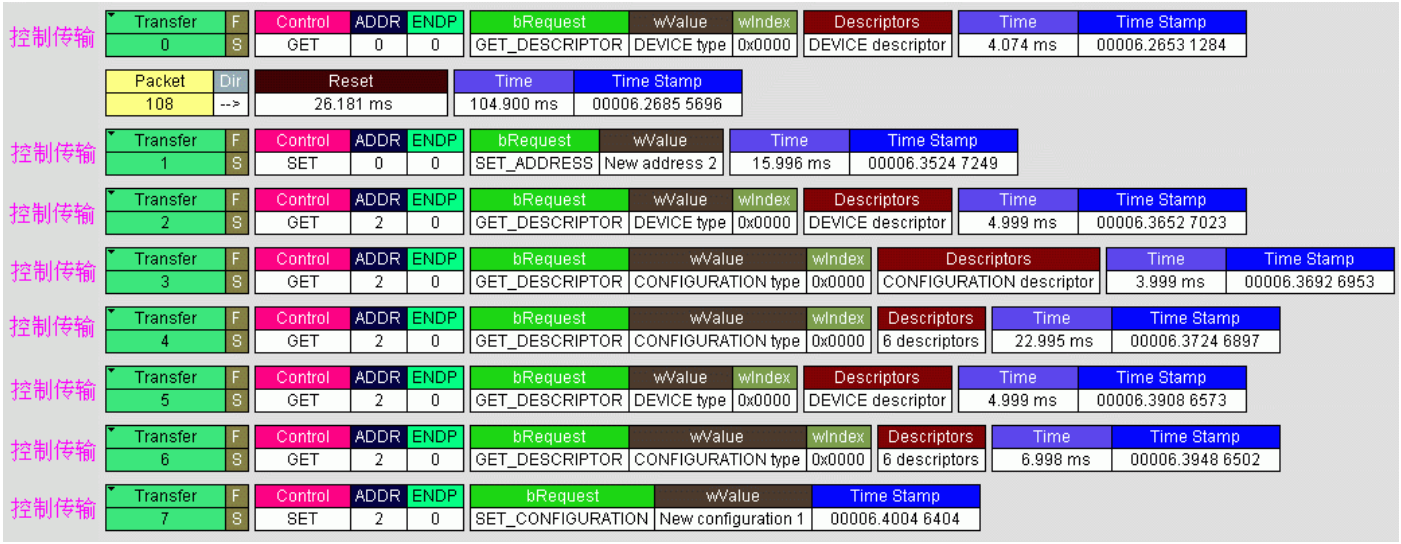
4. 获取配置描述符

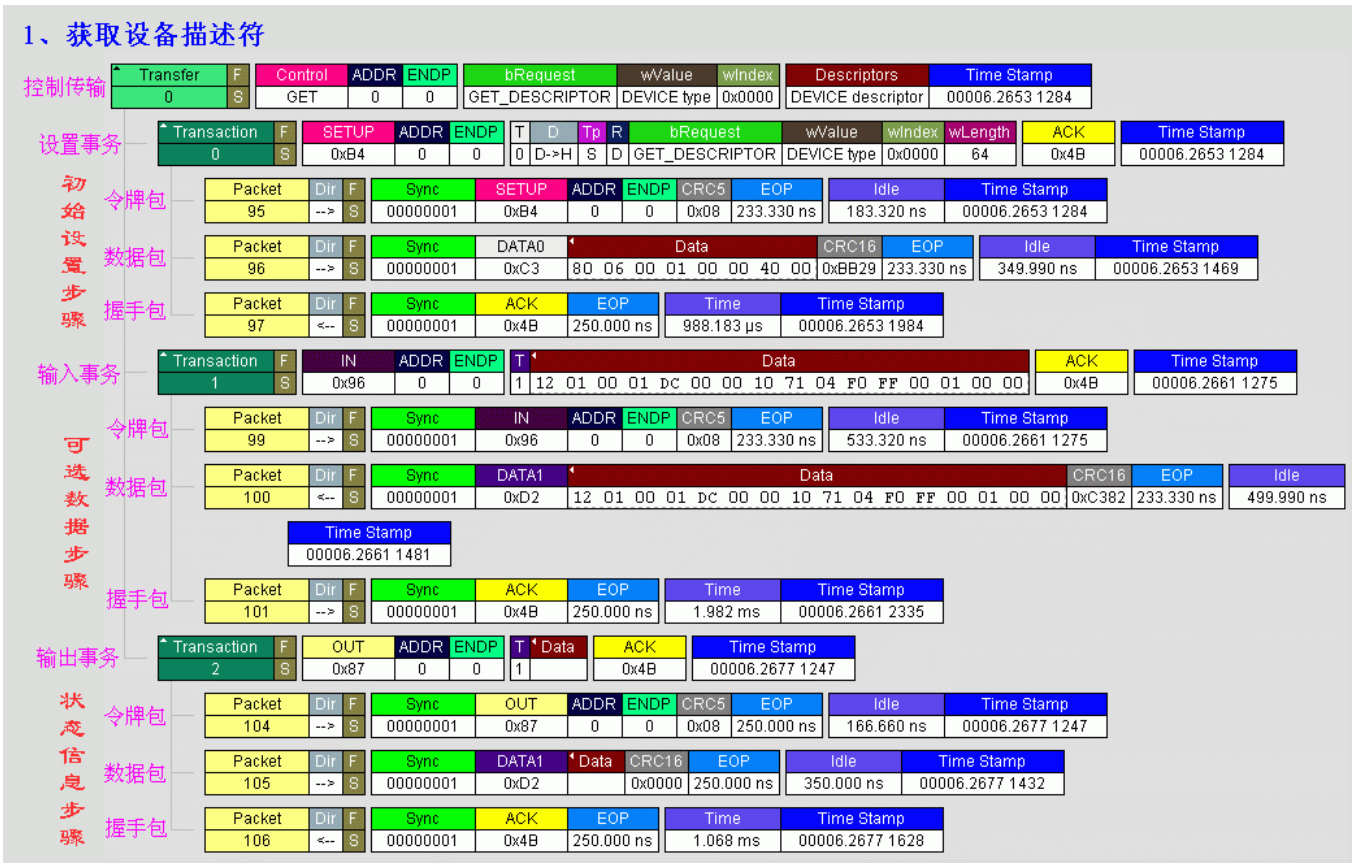
5. 获取描述符集合

配置描述符可能会包含一些其它描述符，这时要一起获取过来。

6. 获取其它描述符。

整个枚举过程基本就获取各种描述符，描述符的意义及作用在其它文档中描述。下面给出网上得到的一组图片：





2、设置地址

控制传输

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	Time Stamp
1	S	SET	0	0	SET_ADDRESS	New address 2	00006.3524 7249

设置事务

Transaction	F	SETUP	ADDR	ENDP	T	D	Tp	R	bRequest	wValue	wIndex	wLength	ACK	Time Stamp
3	S	0xB4	0	0	0	H->D	S	D	SET_ADDRESS	New address 2	0x0000	0	0x4B	00006.3524 7249

初始设置步骤

令牌包

Packet	Dir	F	Sync	SETUP	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
188	-->	S	00000001	0xB4	0	0	0x08	250.000 ns	166.670 ns	00006.3524 7249

数据包

Packet	Dir	F	Sync	DATA0	Data	CRC16	EOP	Idle	Time Stamp
189	-->	S	00000001	0xC3	00 05 02 00 00 00 00 00	0xD768	250.000 ns	333.320 ns	00006.3524 7434

握手包

Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp
190	<--	S	00000001	0x4B	250.000 ns	988.100 µs	00006.3525 0449

输入事务

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time Stamp
4	S	0x96	0	0	1		0x4B	00006.3532 7235

状态信息步骤

令牌包

Packet	Dir	F	Sync	IN	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
192	-->	S	00000001	0x96	0	0	0x08	250.000 ns	500.000 ns	00006.3532 7235

数据包

Packet	Dir	F	Sync	DATA1	Data	CRC16	EOP	Idle	Time Stamp
193	<--	S	00000001	0xD2		0x0000	250.000 ns	499.990 ns	00006.3532 7440

握手包

Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp
194	-->	S	00000001	0x4B	250.000 ns	14.990 ms	00006.3533 0145

3、获取设备描述符

控制传输

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
2	S	GET	2	0	GET_DESCRIPTOR	DEVICE type	0x0000	DEVICE descriptor	4.999 ms	00006.3652 7023

设置事务

Transaction	F	SETUP	ADDR	ENDP	T	D	Tp	R	bRequest	wValue	wIndex	wLength	ACK	Time Stamp
5	S	0xB4	2	0	0	D->H	S	D	GET_DESCRIPTOR	DEVICE type	0x0000	18	0x4B	00006.3652 7023

初始设置步骤

令牌包

Packet	Dir	F	Sync	SETUP	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
210	-->	S	00000001	0xB4	2	0	0x15	250.000 ns	166.660 ns	00006.3652 7023

数据包

Packet	Dir	F	Sync	DATA0	Data	CRC16	EOP	Idle	Time Stamp
211	-->	S	00000001	0xC3	80 06 00 01 00 00 12 00	0x072F	250.000 ns	350.000 ns	00006.3652 7208

握手包

Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp
212	<--	S	00000001	0x4B	250.000 ns	988.083 µs	00006.3653 0224

输入事务

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time Stamp
6	S	0x96	2	0	1	12 01 00 01 DC 00 00 10 71 04 F0 FF 00 01 00 00	0x4B	00006.3660 7009

可选数据步骤

令牌包

Packet	Dir	F	Sync	IN	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
214	-->	S	00000001	0x96	2	0	0x15	250.000 ns	533.330 ns	00006.3660 7009

数据包

Packet	Dir	F	Sync	DATA1	Data	CRC16	EOP	Idle
215	<--	S	00000001	0xD2	12 01 00 01 DC 00 00 10 71 04 F0 FF 00 01 00 00	0xC382	233.330 ns	483.330 ns

Time Stamp
00006.3660 7216

握手包

Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp
216	-->	S	00000001	0x4B	233.330 ns	982.100 µs	00006.3661 0569

输入事务

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time Stamp
7	S	0x96	2	0	0	00 01	0x4B	00006.3668 6995

可选数据步骤

令牌包

Packet	Dir	F	Sync	IN	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
218	-->	S	00000001	0x96	2	0	0x15	250.000 ns	533.330 ns	00006.3668 6995

数据包

Packet	Dir	F	Sync	DATA0	Data	CRC16	EOP	Idle	Time Stamp
219	<--	S	00000001	0xC3	00 01	0xFCF1	233.330 ns	566.660 ns	00006.3668 7202

握手包

Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp
220	-->	S	00000001	0x4B	233.330 ns	991.433 µs	00006.3668 7495

输出/状态

Transaction	F	OUT	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
8	S	0x87	2	0	1	0 bytes	0x4B	2.000 ms	00006.3676 6981

4、获取配置描述符

控制传输

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time Stamp
3	S	GET	2	0	GET_DESCRIPTOR	CONFIGURATION type	0x0000	CONFIGURATION descriptor	00006.3692 6953

设置事务

Transaction	F	SETUP	ADDR	ENDP	T	D	Tp	R	bRequest	wValue	wIndex	wLength	ACK	Time Stamp
9	S	0xB4	2	0	0	D->H	S	D	GET_DESCRIPTOR	CONFIGURATION type	0x0000	9	0x4B	00006.3692 6953

初始设置步骤

令牌包

Packet	Dir	F	Sync	SETUP	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
227	-->	S	00000001	0xB4	2	0	0x15	233.330 ns	183.320 ns	00006.3692 6953

数据包

Packet	Dir	F	Sync	DATA0	Data	CRC16	EOP	Idle	Time Stamp
228	-->	S	00000001	0xC3	80 06 00 02 00 00 09 00	0x7520	233.330 ns	366.660 ns	00006.3692 7138

握手包

Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp
229	<--	S	__000001	0x4B	250.000 ns	988.167 µs	00006.3693 0154

输入事务

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time Stamp
10	S	0x96	2	0	1	09 02 2E 00 01 01 00 60 01	0x4B	00006.3700 6944

可选数据步骤

令牌包

Packet	Dir	F	Sync	IN	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
231	-->	S	00000001	0x96	2	0	0x15	233.330 ns	550.000 ns	00006.3700 6944

数据包

Packet	Dir	F	Sync	DATA1	Data	CRC16	EOP	Idle	Time Stamp
232	<--	S	00000001	0xD2	09 02 2E 00 01 01 00 60 01	0xA01E	233.330 ns	483.320 ns	00006.3700 7151

握手包

Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp
233	-->	S	00000001	0x4B	233.330 ns	986.933 µs	00006.3701 0214

输出事务

Transaction	F	OUT	ADDR	ENDP	T	Data	ACK	Time Stamp
11	S	0x87	2	0	1		0x4B	00006.3708 6930

状态信息步骤

令牌包

Packet	Dir	F	Sync	OUT	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
235	-->	S	00000001	0x87	2	0	0x15	233.330 ns	183.320 ns	00006.3708 6930

数据包

Packet	Dir	F	Sync	DATA1	Data	CRC16	EOP	Idle	Time Stamp
236	-->	S	00000001	0xD2		0x0000	250.000 ns	350.000 ns	00006.3708 7115

握手包

Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp
237	<--	S	00000001	0x4B	233.330 ns	1.993 ms	00006.3708 7311

5、获取配置描述符其他内容

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time Stamp
4	S	GET	2	0	GET_DESCRIPTOR	CONFIGURATION type	0x0000	6 descriptors	00006.3724 6897

Transaction	F	SETUP	ADDR	ENDP	T	D	Tp	R	bRequest	wValue	wIndex	wLength	ACK	Time Stamp
12	S	0xB4	2	0	0	D->H	S	D	GET_DESCRIPTOR	CONFIGURATION type	0x0000	255	0x4B	00006.3724 6897

Packet	Dir	F	Sync	SETUP	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
240	-->	S	00000001	0xB4	2	0	0x15	250.000 ns	166.660 ns	00006.3724 6897

Packet	Dir	F	Sync	DATA0	Data	CRC16	EOP	Idle	Time Stamp
241	-->	S	00000001	0xC3	80 06 00 02 00 00 FF 00	0x9725	250.000 ns	333.330 ns	00006.3724 7082

Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp
242	<--	S	00000001	0x4B	233.330 ns	988.017 µs	00006.3725 0102

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
13	S	0x96	2	0	1	09 02 2E 00 01 01 00 60 01 09 04 00 00 04 00 00	0x4B	999.750 µs	00006.3732 6883

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
14	S	0x96	2	0	0	00 00 07 05 81 03 08 00 C8 07 05 01 03 08 00 C8	0x4B	999.767 µs	00006.3740 6868

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
15	S	0x96	2	0	1	07 05 82 02 40 00 00 07 05 02 02 40 00 00	0x4B	2.000 ms	00006.3748 6854

Transaction	F	OUT	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
16	S	0x87	2	0	1		0x4B	17.996 ms	00006.3764 6826

6、获取设备和配置描述符

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time Stamp
5	S	GET	2	0	GET_DESCRIPTOR	DEVICE type	0x0000	DEVICE descriptor	00006.3908 6573

Transaction	F	SETUP	ADDR	ENDP	T	D	Tp	R	bRequest	wValue	wIndex	wLength	ACK	Time Stamp
17	S	0xB4	2	0	0	D->H	S	D	GET_DESCRIPTOR	DEVICE type	0x0000	18	0x4B	00006.3908 6573

Packet	Dir	F	Sync	SETUP	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
278	-->	S	00000001	0xB4	2	0	0x15	250.000 ns	166.660 ns	00006.3908 6573

Packet	Dir	F	Sync	DATA0	Data	CRC16	EOP	Idle	Time Stamp
279	-->	S	00000001	0xC3	80 06 00 01 00 00 12 00	0x072F	250.000 ns	350.000 ns	00006.3908 6758

Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp
280	<--	S	00000001	0x4B	233.330 ns	988.067 µs	00006.3908 7274

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
18	S	0x96	2	0	1	12 01 00 01 DC 00 00 10 71 04 F0 FF 00 01 00 00	0x4B	999.767 µs	00006.3916 6558

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
19	S	0x96	2	0	0	00 01	0x4B	999.767 µs	00006.3924 6544

Transaction	F	OUT	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
20	S	0x87	2	0	1		0x4B	2.000 ms	00006.3932 6530

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time Stamp
6	S	GET	2	0	GET_DESCRIPTOR	CONFIGURATION type	0x0000	6 descriptors	00006.3948 6502

Transaction	F	SETUP	ADDR	ENDP	T	D	Tp	R	bRequest	wValue	wIndex	wLength	ACK	Time	Time Stamp
21	S	0xB4	2	0	0	D->H	S	D	GET_DESCRIPTOR	CONFIGURATION type	0x0000	137	0x4B	999.767 µs	00006.3948 6502

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
22	S	0x96	2	0	1	09 02 2E 00 01 01 00 60 01 09 04 00 00 04 00 00	0x4B	999.850 µs	00006.3956 6488

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
23	S	0x96	2	0	0	00 00 07 05 81 03 08 00 C8 07 05 01 03 08 00 C8	0x4B	999.683 µs	00006.3964 6479

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
24	S	0x96	2	0	1	07 05 82 02 40 00 00 07 05 02 02 40 00 00	0x4B	2.000 ms	00006.3972 6460

Transaction	F	OUT	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
25	S	0x87	2	0	1		0x4B	2.000 ms	00006.3988 6432

7、设置配置

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	Time Stamp
7	S	SET	2	0	SET_CONFIGURATION	New configuration 1	00006.4004 6404

设置事务

Transaction	F	SETUP	ADDR	ENDP	T	D	Tp	R	bRequest	wValue	wIndex	wLength	ACK	Time Stamp
26	S	0xB4	2	0	0	H->D	S	D	SET_CONFIGURATION	New configuration 1	0x0000	0	0x4B	00006.4004 6404

初始设置步骤

令牌包

Packet	Dir	F	Sync	SETUP	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
317	-->	S	00000001	0xB4	2	0	0x15	250.000 ns	166.660 ns	00006.4004 6404

数据包

Packet	Dir	F	Sync	DATA0	Data	CRC16	EOP	Idle	Time Stamp
318	-->	S	00000001	0xC3	00 09 01 00 00 00 00 00	0xE4A4	250.000 ns	333.340 ns	00006.4004 6589

握手包

Packet	Dir	F	Sync	ACK	EOP	Time	Time Stamp
319	<--	S	00000001	0x4B	250.000 ns	988.083 µs	00006.4004 7104

输入事务

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time Stamp
27	S	0x96	2	0	1		0x4B	00006.4012 6389

状态信息步骤

令牌包

Packet	Dir	F	Sync	IN	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
321	-->	S	00000001	0x96	2	0	0x15	250.000 ns	516.660 ns	00006.4012 6389

数据包

Packet	Dir	F	Sync	DATA1	Data	CRC16	EOP	Idle	Time Stamp
322	<--	S	00000001	0xD2		0x0000	250.000 ns	483.330 ns	00006.4012 6595

握手包

Packet	Dir	F	Sync	ACK	EOP	Time Stamp
323	-->	S	00000001	0x4B	250.000 ns	00006.4012 6799