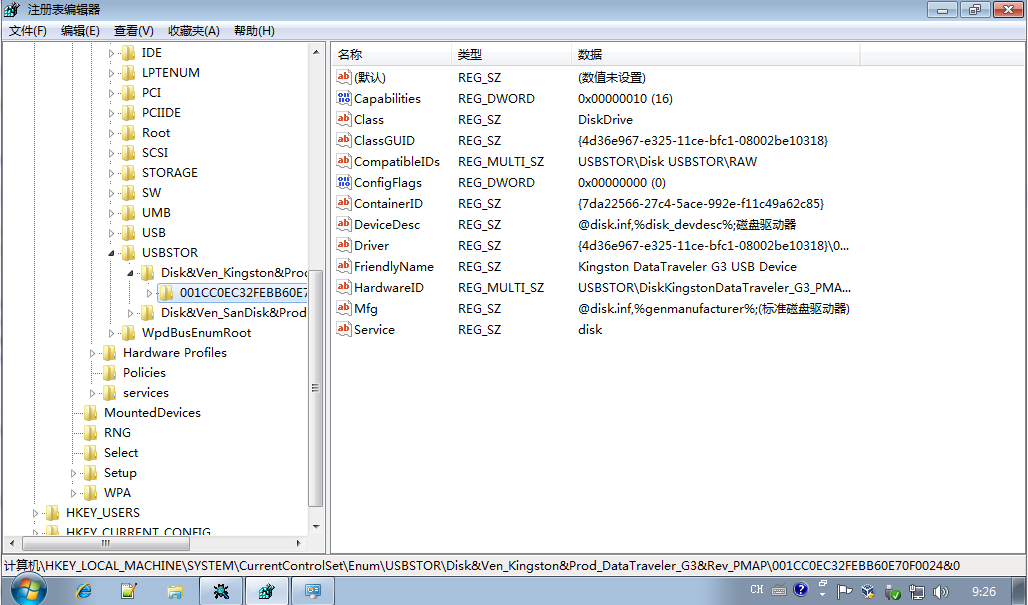
# 过滤驱动的加载

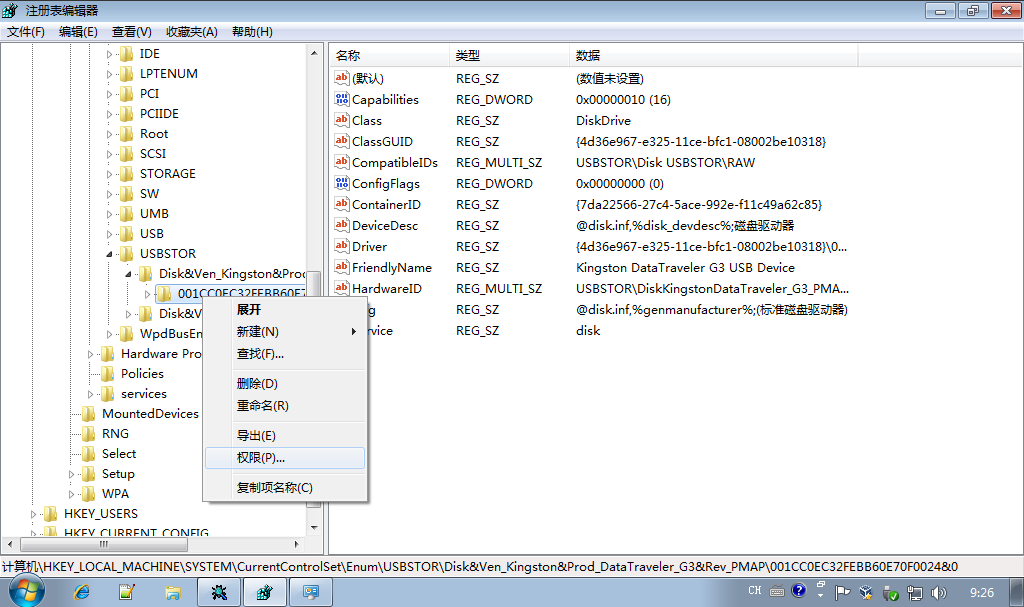
WDM式的过滤驱动的加载比较特殊，需要修改注册表，使得设备在Enum阶段就可以找到相关的驱动信息。之后把这个设备的对象传给驱动的AddDevice例程，之后再进行挂载的过程，但是Win7对注册表哦权限管理比较严格，现在遇到的问题是无法随意添加键值。

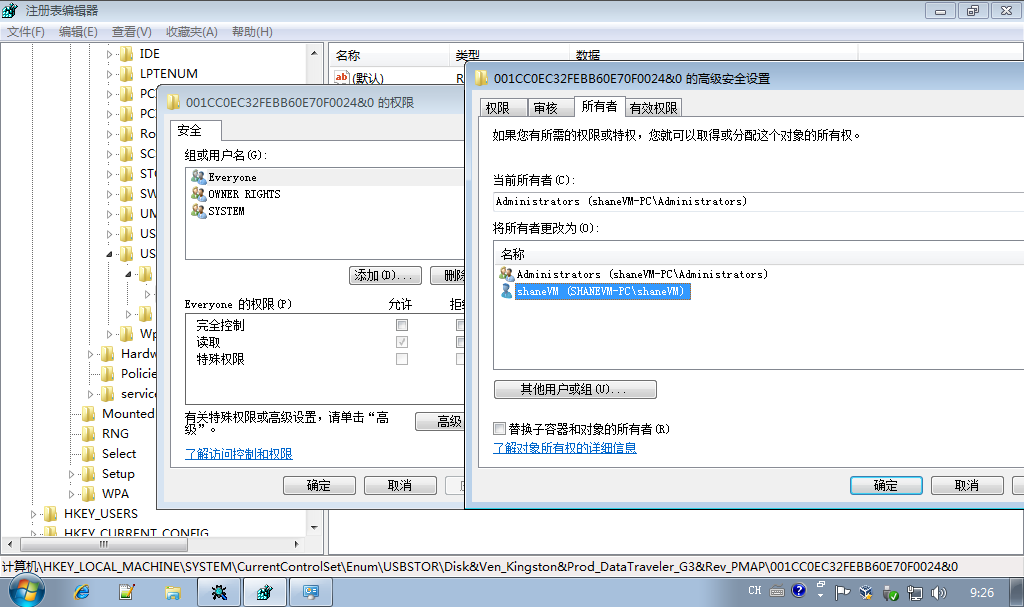


具体的方法为，首先定位到要修改的地方，在我们这里就是：

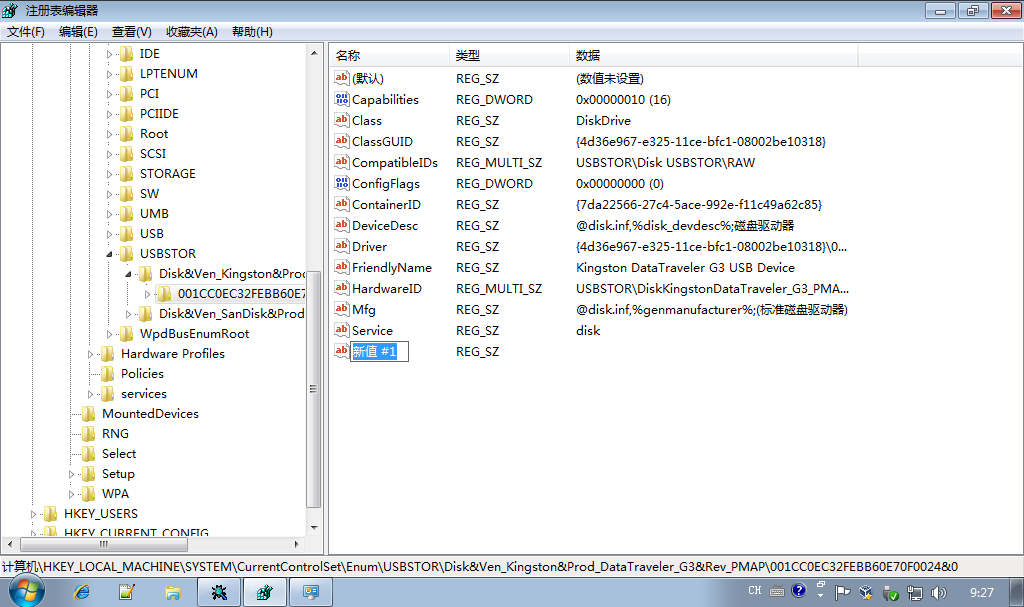


之后修改目录的所有者。

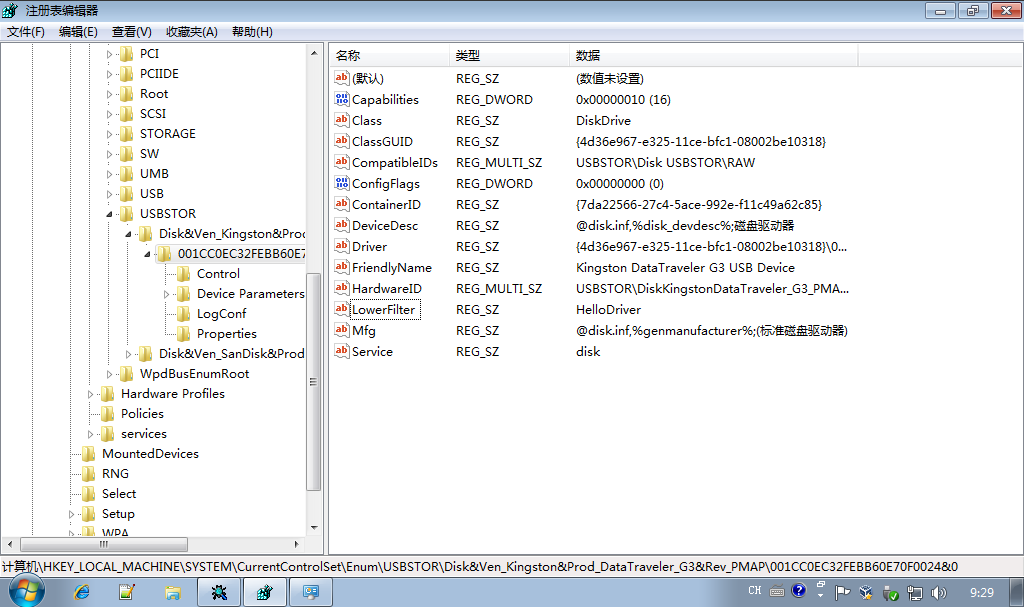




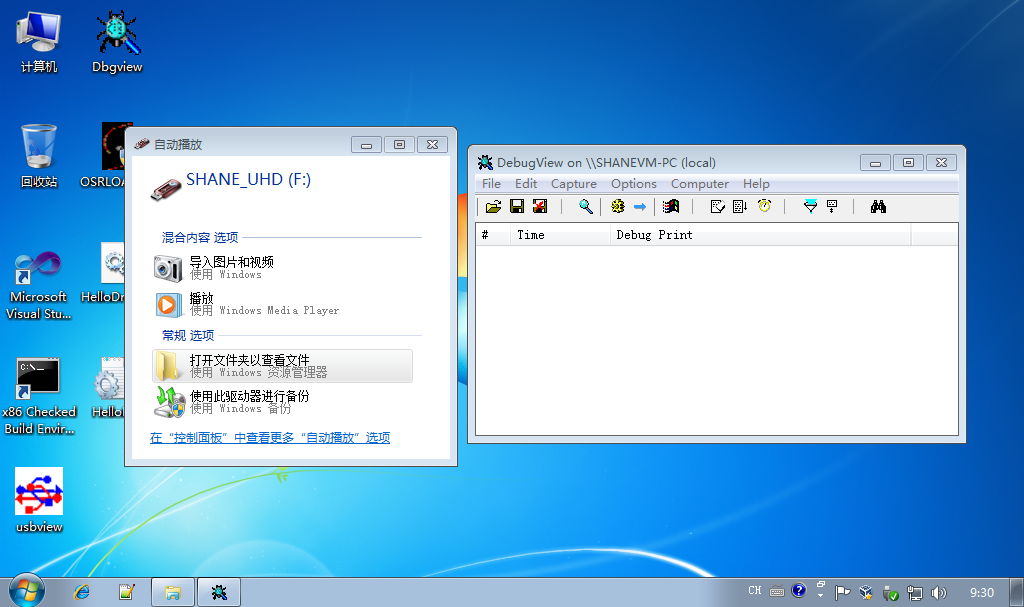




之后就可以进行相关的操作了。

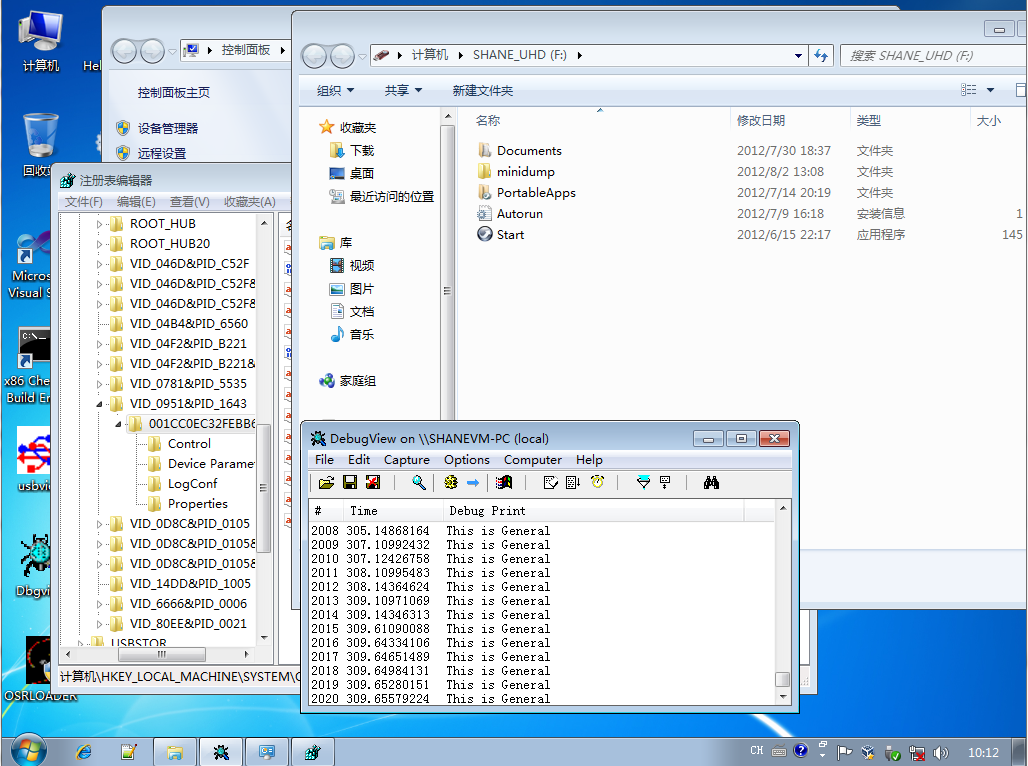


之后打开DBGView接入设备进行观察。



很明显。第一次启动的时候并未实现自动挂载。重启再试。

根据Tim Robert的提示。在Win7中，键值的位置已经改变。



可以看到我们的设备在USB设备枚举的过程中由于是一个混合设备，产生了两个子键，一个为HID设备，另外一个才为USBAudio的设备。在USBAudio设备下挂载我们的USB过滤驱动。

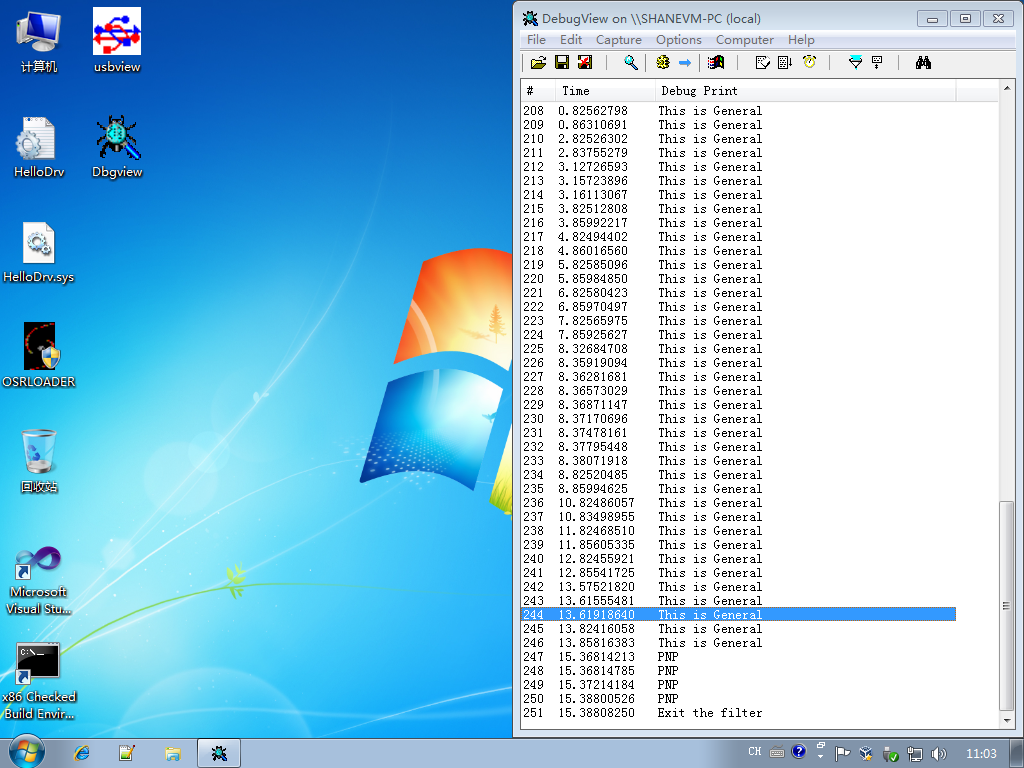


可以看到LowerFilter已经正常的挂载上了。产生了信息。之后就是要确认声音牵扯到的是哪个信号，之后提取出来就好了。

但是在出声之后系统就卡死了。

<http://www.osronline.com/showthread.cfm?link=226286>

再次进行U盘的测试。

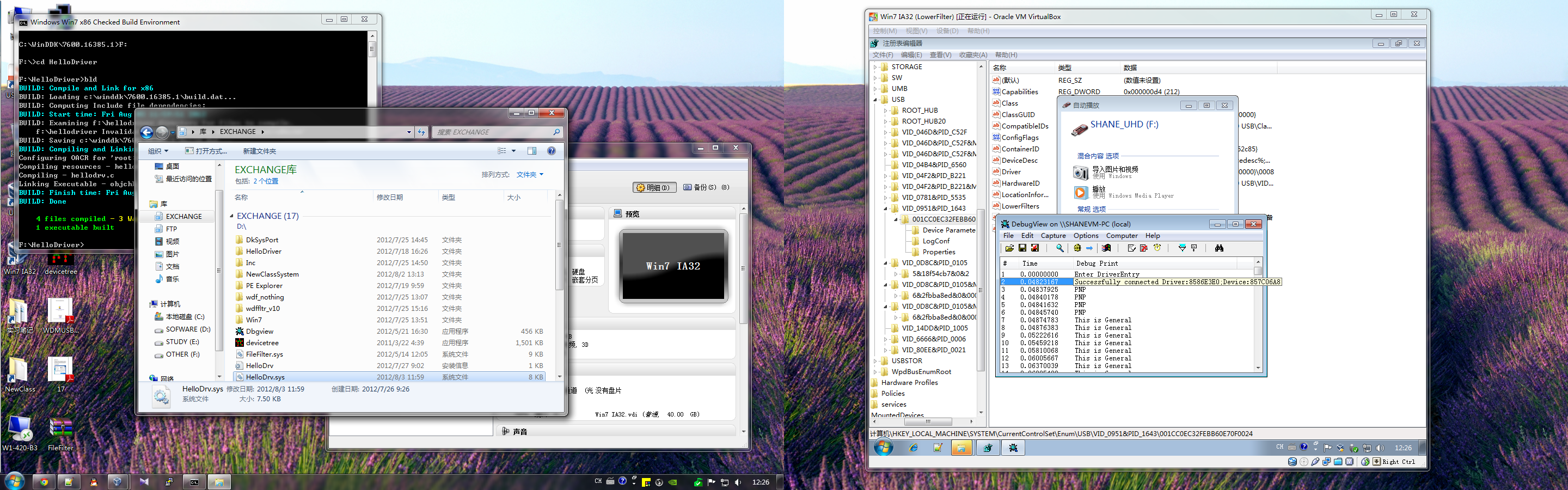


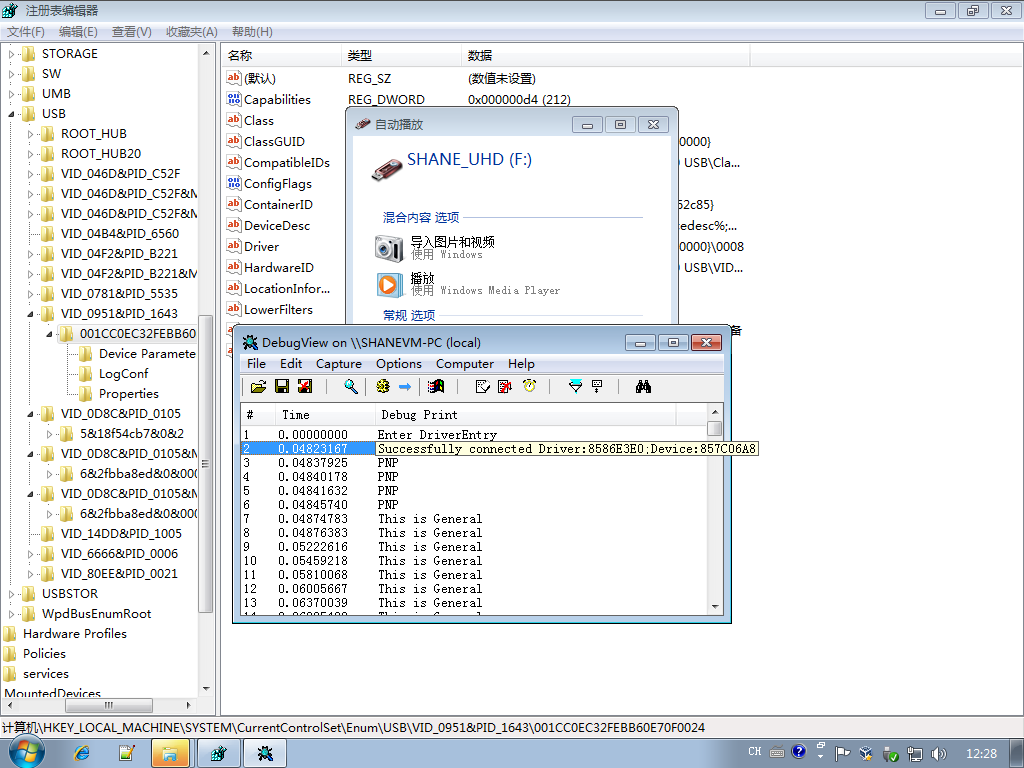
也能够正常的退出。

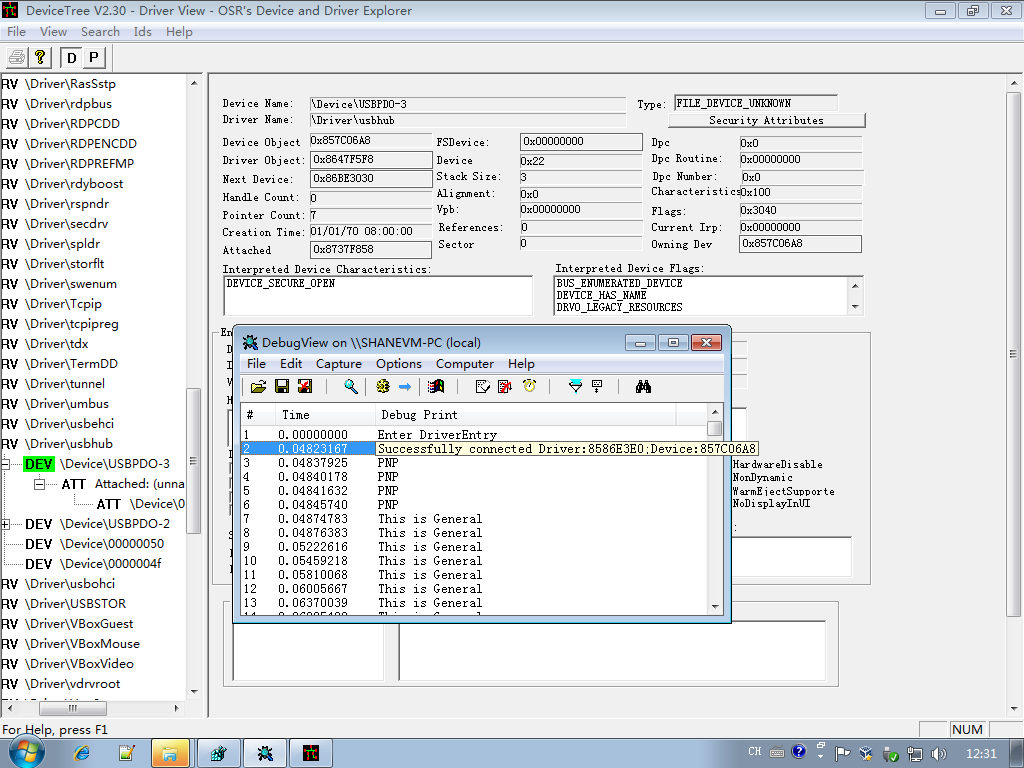
之后大概会长生每秒两个的General事件。

使用正常。而且在复制文件进入的时候，明显能够觉察到包的个数变多，而且读写均正常。

打印驱动及设备对象的结果为：



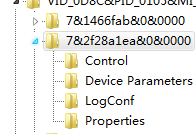




最后可以看到，设备最终是挂载到USBHUB底下特定的USBPDO上面的。

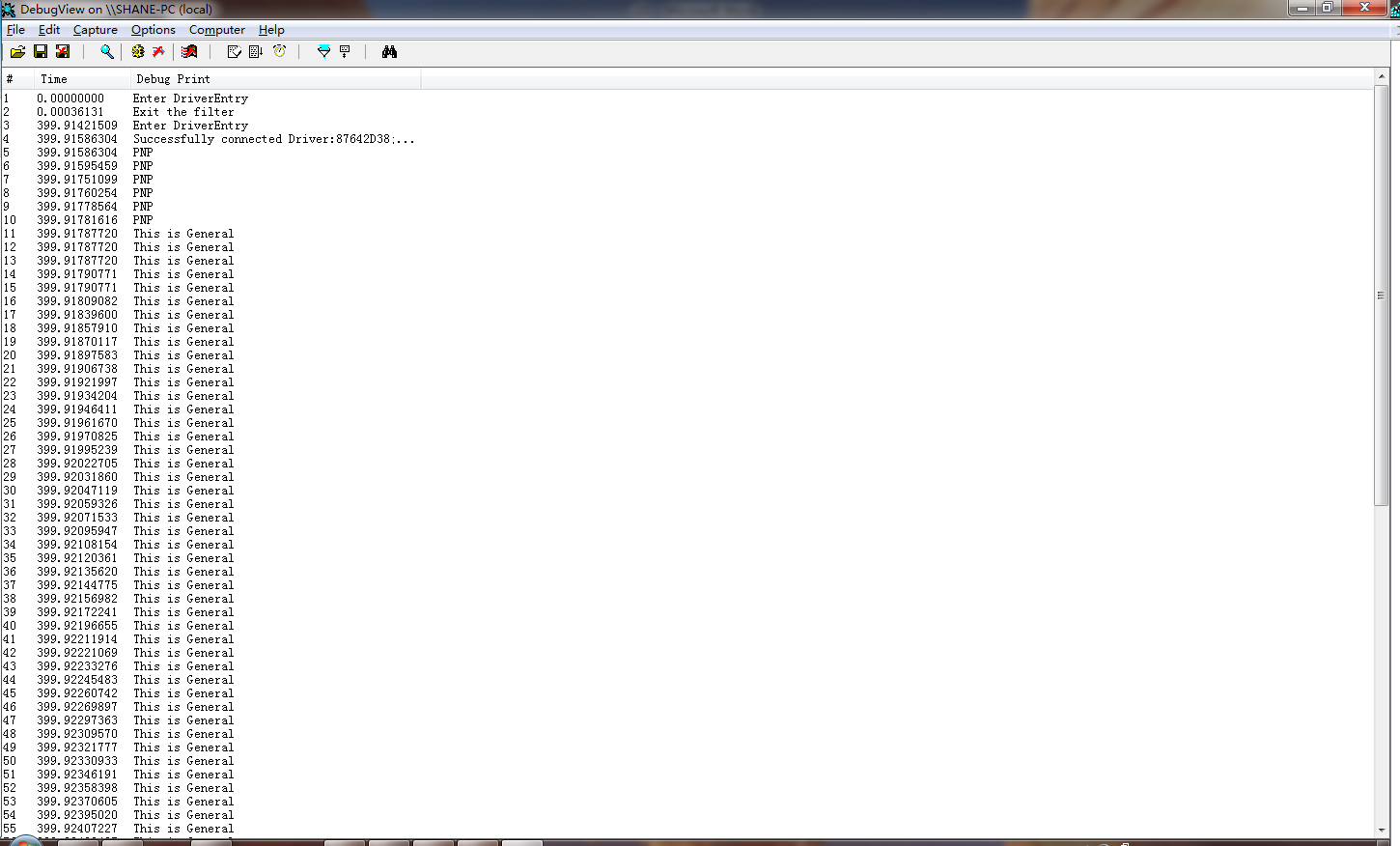
之后在物理机上面进行测试。

可以看到在这里面一共有三个子键，推测为不同接口上面产生的多个子键。

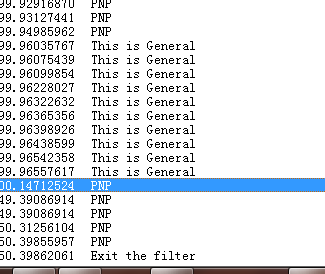


之后修改了子键最多的一项。

经测试，正为此项。

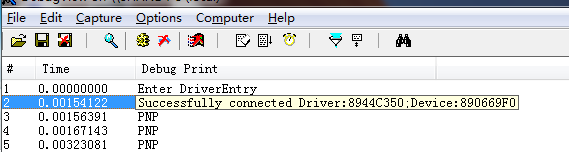


设备最后成功加载。



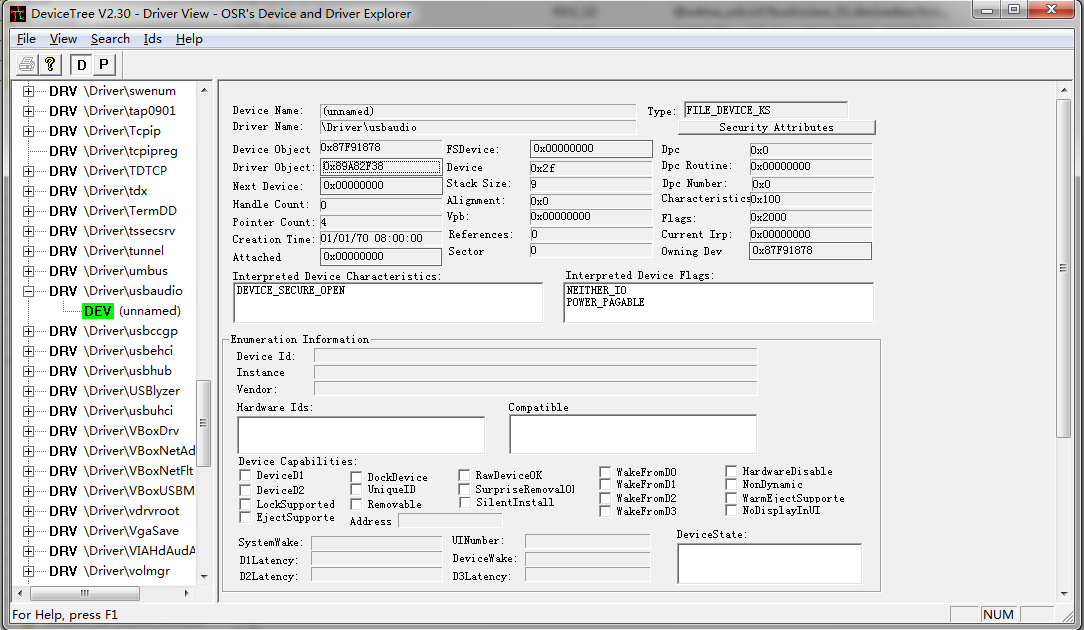
以及退出。

之后在物理机中查看硬件以及驱动的挂载层次。

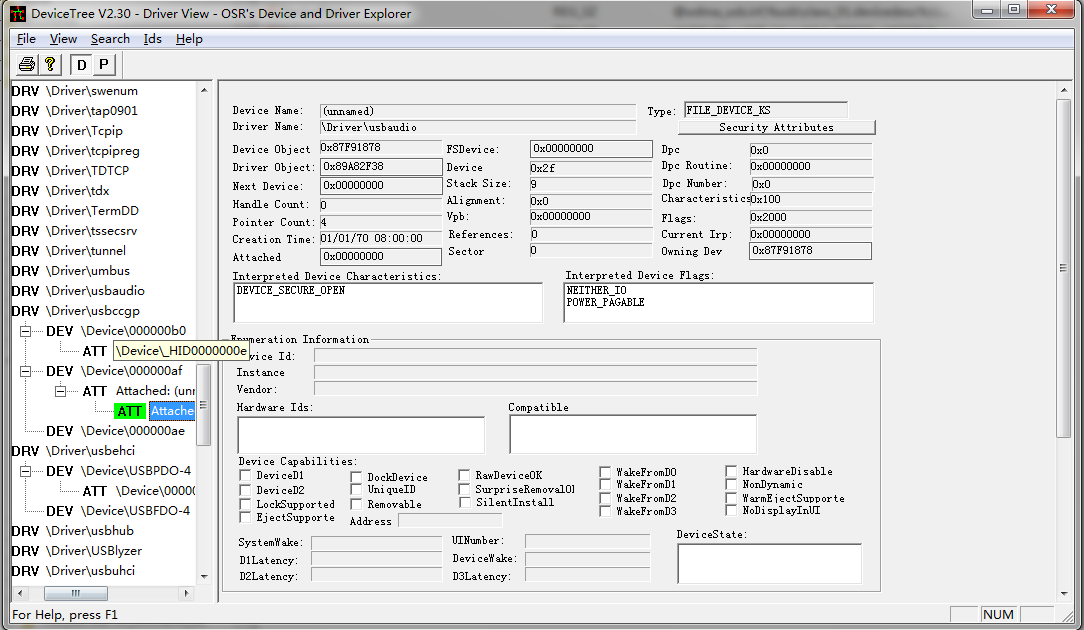


驱动及设备对象地址如图。

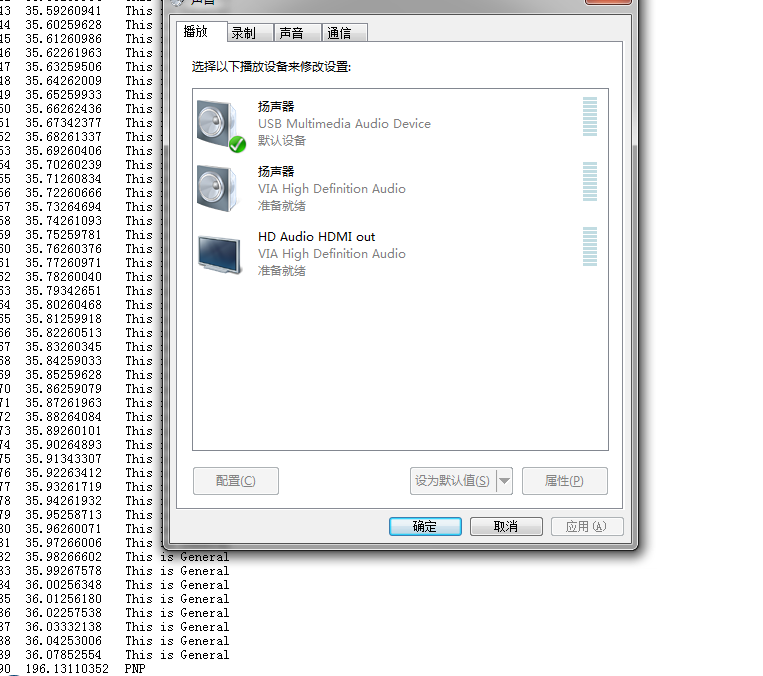
在USB Audio下，并没有挂载的对象。



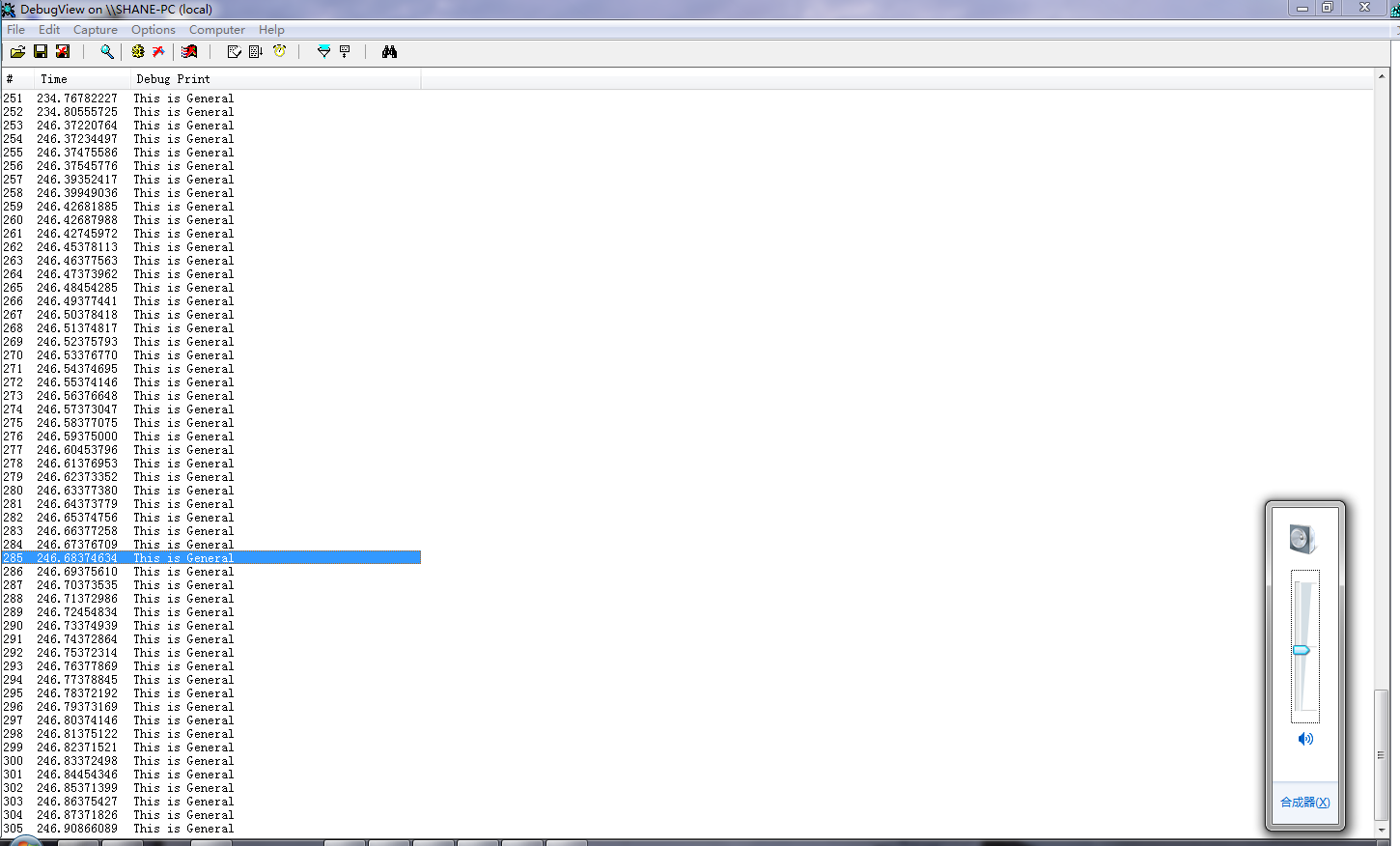
最后的USB Audio的挂载层次可以查明是在CGGP上面。



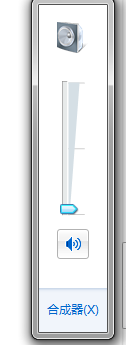
查看播放设备，会调用一次PNP。但是并不会像虚拟机一样死机。



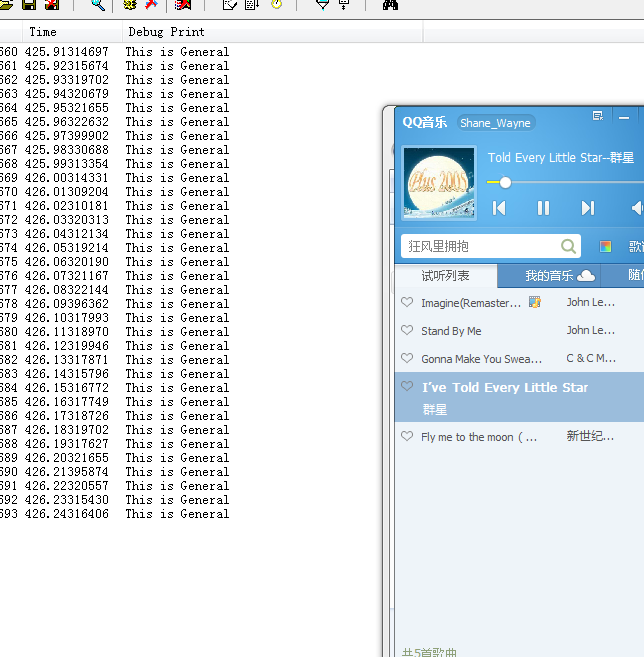
在播放测试声音后，General例程会大量调用，但是并不会死机。



每个测试声音大概会调用50次例程。



即使是调整音量也会产生1-2个普通例程调用。



一旦开始播放声效，会产生很多很多的例程调用。

> only to check whether the Streaming data passes through this filter or not.

For now, I am not particularly interested to process any streaming data.

The streaming data will pass through the filter, as URBs (USB request

blocks). URBs are sent as IRP\_MJ\_INTERNAL\_DEVICE\_CONTROL requests. The

URB will be for URB\_FUNCTION\_ISOCH\_TRANSFER.

What do you plan to do with the video here? Usbvideo.sys expects to

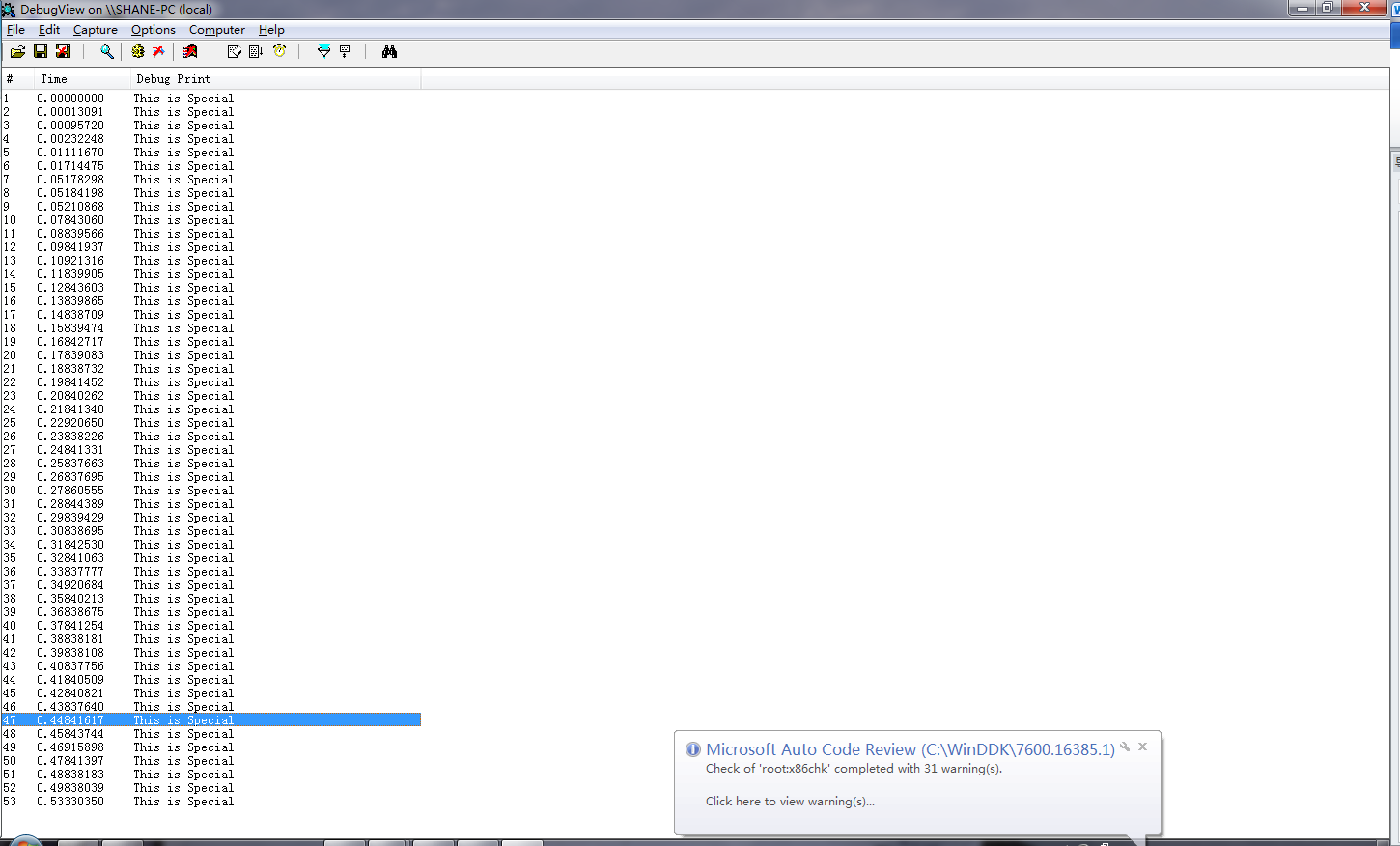
receive USB packets in the UVC format.

现在主要的问题，是如何从众多的General例程中分辨出携带有我们需要的音频信息的调用例程。

这个就得一步一步去细化调用分支，之后在分支中取得具体的信息。

可以看到在更改了例程函数之后可以知道大多数的控制例程还是在IRP\_MJ\_INTERNAL\_DEVICE\_CONTROL这个例程里面。

一个测试声音，总共有53次调用：



之后在驱动代码做如下更改

PIO\_STACK\_LOCATION stack = IoGetCurrentIrpStackLocation(Irp);

NTSTATUS status;

ULONG dwControlCode = stack->Parameters.DeviceIoControl.IoControlCode;

if(dwControlCode == IOCTL\_INTERNAL\_USB\_SUBMIT\_URB)

{

DbgPrintEx(DPFLTR\_DEFAULT\_ID,DPFLTR\_ERROR\_LEVEL,"TransFer\n");

}

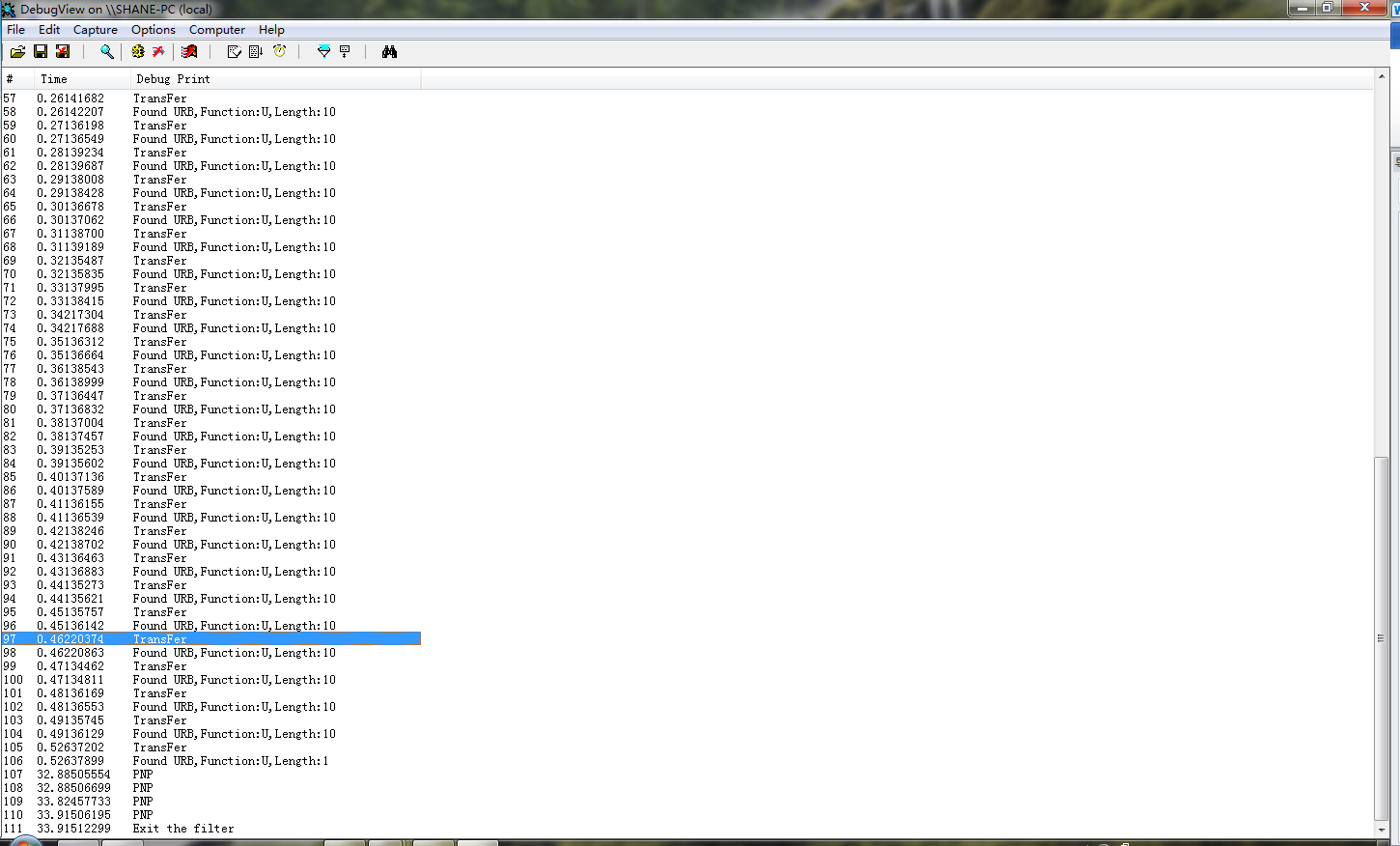
else

{

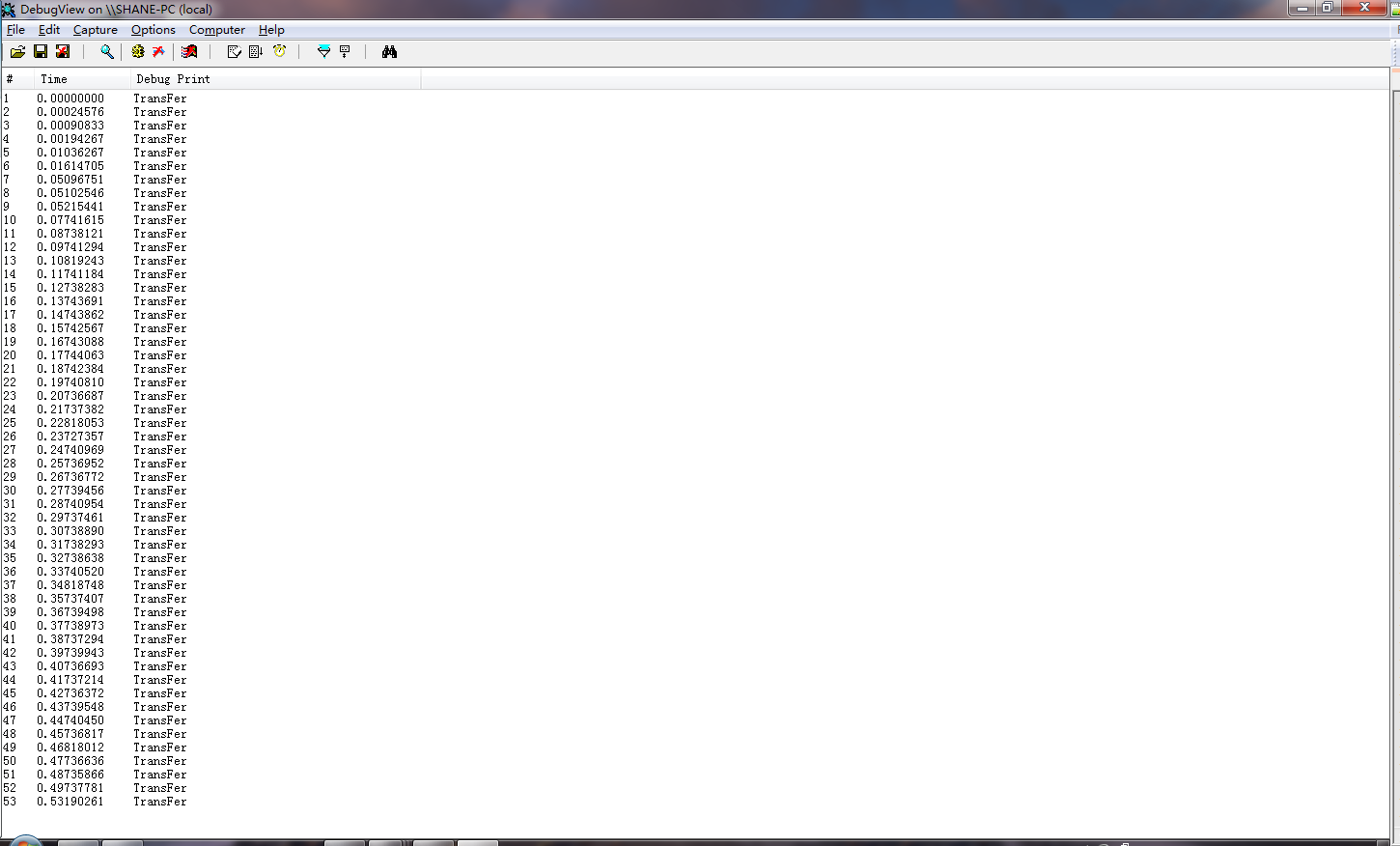
DbgPrintEx(DPFLTR\_DEFAULT\_ID,DPFLTR\_ERROR\_LEVEL,"This is Special\n");

}

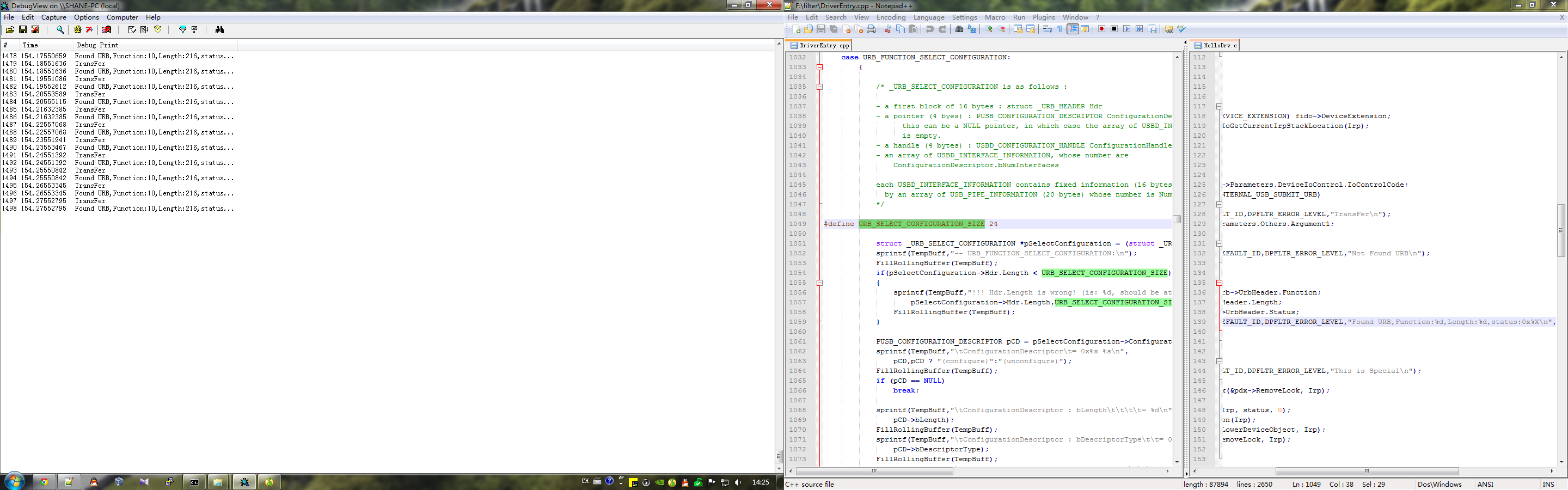
这段代码能够侦测URB的传送情况。



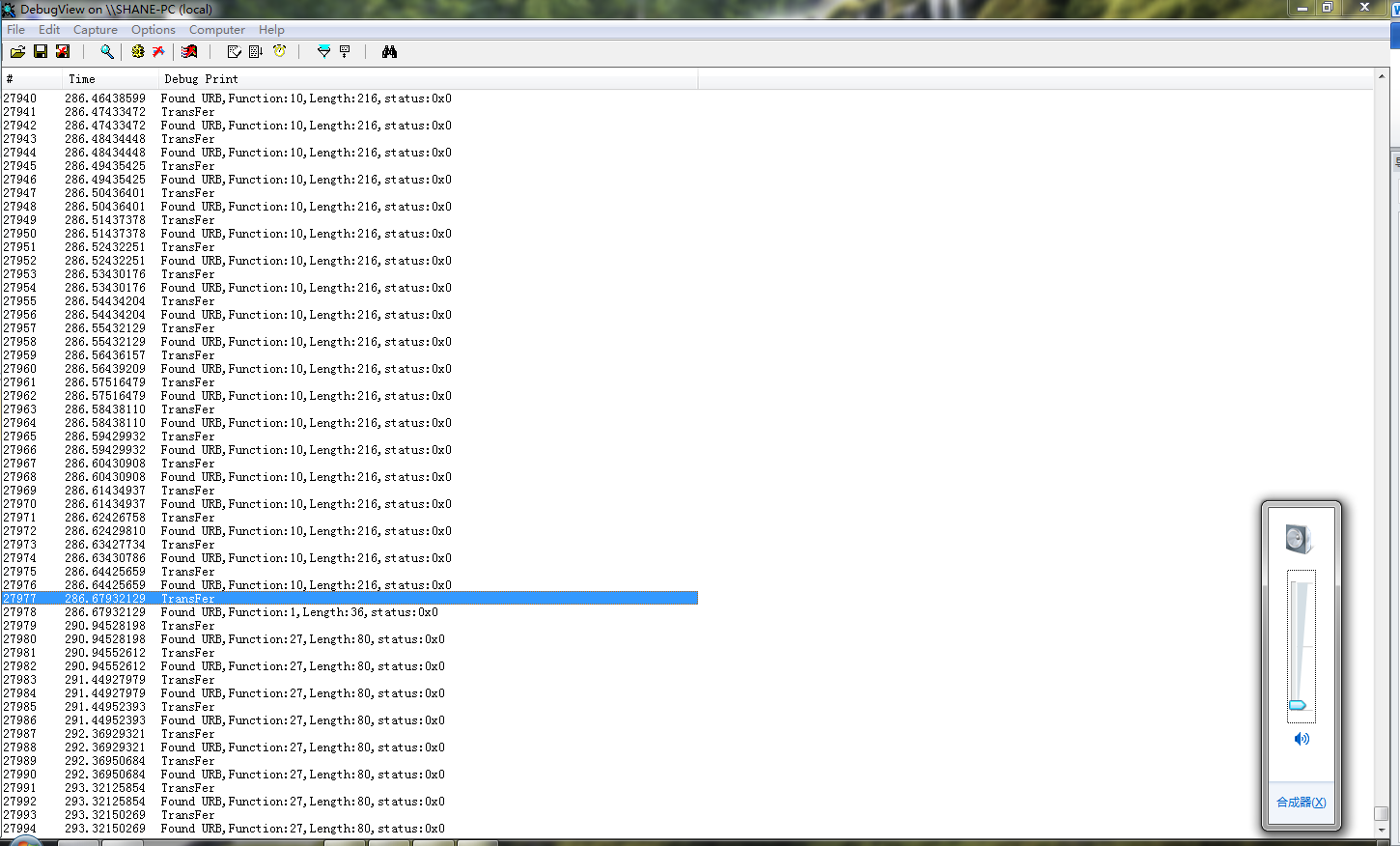
最后可以看到Length的信息，还有功能的信息。



可以看到，又都成了Transfer。。。



可以看到最后同步传输时候的function为10。长度为216。具体应该还牵扯到usb的标准。



调节音量时URB的Function Code为27。

最后查询官方文档，可以得到URB的具体声明。

typedef struct \_URB {

union {

struct URB\_HEADER  UrbHeader;

struct URB\_SELECT\_INTERFACE  UrbSelectInterface;

struct URB\_SELECT\_CONFIGURATION  UrbSelectConfiguration;

struct URB\_PIPE\_REQUEST  UrbPipeRequest;

struct URB\_FRAME\_LENGTH\_CONTROL  UrbFrameLengthControl;

struct URB\_GET\_FRAME\_LENGTH  UrbGetFrameLength;

struct URB\_SET\_FRAME\_LENGTH  UrbSetFrameLength;

struct URB\_GET\_CURRENT\_FRAME\_NUMBER  UrbGetCurrentFrameNumber;

struct URB\_CONTROL\_TRANSFER  UrbControlTransfer;

struct URB\_CONTROL\_TRANSFER\_EX  UrbControlTransferEx;

struct URB\_CONTROL\_TRANSFER\_EX  UrbControlTransferEx;

struct URB\_BULK\_OR\_INTERRUPT\_TRANSFER  UrbBulkOrInterruptTransfer;

struct URB\_ISOCH\_TRANSFER  UrbIsochronousTransfer;

struct URB\_CONTROL\_DESCRIPTOR\_REQUEST  UrbControlDescriptorRequest;

struct URB\_CONTROL\_GET\_STATUS\_REQUEST  UrbControlGetStatusRequest;

struct URB\_CONTROL\_FEATURE\_REQUEST  UrbControlFeatureRequest;

struct URB\_CONTROL\_VENDOR\_OR\_CLASS\_REQUEST  UrbControlVendorClassRequest;

struct URB\_CONTROL\_GET\_INTERFACE\_REQUEST  UrbControlGetInterfaceRequest;

struct URB\_CONTROL\_GET\_CONFIGURATION\_REQUEST  UrbControlGetConfigurationRequest;

struct URB\_OS\_FEATURE\_DESCRIPTOR\_REQUEST  UrbOSFeatureDescriptorRequest;

struct URB\_OPEN\_STATIC\_STREAMS  UrbOpenStaticStreams;

};

} URB, \*PURB;

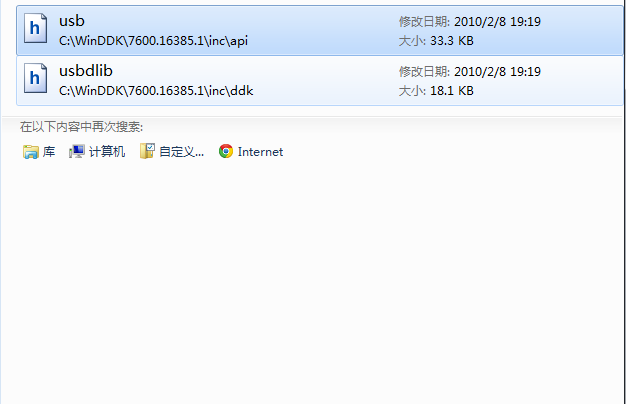
可以看成为一个联合体。

**Function**

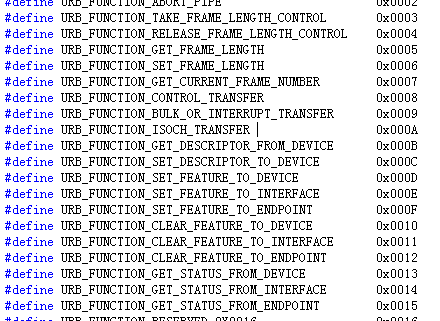
Specifies a numeric code indicating the requested operation for this URB. One of the following values must be set:

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| **URB\_FUNCTION\_SELECT\_CONFIGURATION** | Indicates to the host controller driver that a configuration is to be selected. If set, the URB is used with [**\_URB\_SELECT\_CONFIGURATION**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540422(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_SELECT\_INTERFACE** | Indicates to the host controller driver that an alternate interface setting is being selected for an interface. If set, the URB is used with [**\_URB\_SELECT\_INTERFACE**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540425(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_ABORT\_PIPE** | Indicates that all outstanding requests for a pipe should be canceled. If set, the URB is used with[**\_URB\_PIPE\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540419(v=vs.85).aspx) as the data structure. This general-purpose request enables a client to cancel any pending transfers for the specified pipe. Pipe state and endpoint state are unaffected. The abort request might complete before all outstanding requests have completed. Do not assume that completion of the abort request implies that all other outstanding requests have completed. |
| **URB\_FUNCTION\_TAKE\_FRAME\_LENGTH\_CONTROL** | This URB function is **deprecated** in Windows 2000 and later operating systems and is not supported by Microsoft. Do not use. If you specify this function with an URB request, the request will fail and the system will report an error. |
| **URB\_FUNCTION\_RELEASE\_FRAME\_LENGTH\_CONTROL** | This URB function is **deprecated** in Windows 2000 and later operating systems and is not supported by Microsoft. Do not use. If you specify this function with an URB request, the request will fail and the system will report an error. |
| **URB\_FUNCTION\_GET\_FRAME\_LENGTH** | This URB function is **deprecated** in Windows 2000 and later operating systems and is not supported by Microsoft. Do not use. If you use this function with a URB request, the request will fail and the system will report an error. |
| **URB\_FUNCTION\_SET\_FRAME\_LENGTH** | This URB function is **deprecated** in Windows 2000 and later operating systems and is not supported by Microsoft. Do not use. If you use it with a URB request, the request will fail and the system will report an error. |
| **URB\_FUNCTION\_GET\_CURRENT\_FRAME\_NUMBER** | Requests the current frame number from the host controller driver. If set, the URB is used with[**\_URB\_GET\_CURRENT\_FRAME\_NUMBER**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540401(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_CONTROL\_TRANSFER** | Transfers data to or from a control pipe. If set, the URB is used with [**\_URB\_CONTROL\_TRANSFER**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540384(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_CONTROL\_TRANSFER\_EX** | Transfers data to or from a control pipe without a time limit specified by a timeout value. If set, the URB is used with [**URB\_CONTROL\_TRANSFER\_EX**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540387(v=vs.85).aspx) as the data structure.  Available in Windows Vista and later operating systems. |
| **URB\_FUNCTION\_BULK\_OR\_INTERRUPT\_TRANSFER** | Transfers data from a bulk pipe or interrupt pipe or to an bulk pipe. If set, the URB is used with[**\_URB\_BULK\_OR\_INTERRUPT\_TRANSFER**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540352(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_BULK\_OR\_INTERRUPT\_TRANSFER\_USING\_CHAINED\_MDL** | Transfers data to and from a bulk pipe or interrupt pipe, by using chained MDLs. If set, the URB is used with[**\_URB\_BULK\_OR\_INTERRUPT\_TRANSFER**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540352(v=vs.85).aspx) as the data structure. The client driver must set the**TransferBufferMDL** member to the first [**MDL**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff554414(v=vs.85).aspx) structure in the chain that contains the transfer buffer. The USB driver stack ignores the **TransferBuffer** member when processing this URB.  Available in Windows 8. For information about using chained MDLs, see [How to Send Chained MDLs](http://msdn.microsoft.com/en-us/library/windows/hardware/hh450848(v=vs.85).aspx). |
| **URB\_FUNCTION\_ISOCH\_TRANSFER** | Transfers data to or from an isochronous pipe. If set, the URB is used with [**\_URB\_ISOCH\_TRANSFER**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540414(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_ISOCH\_TRANSFER\_USING\_CHAINED\_MDL** | Transfers data to or from an isochronous pipe by using chained MDLs. If set, the URB is used with[**\_URB\_ISOCH\_TRANSFER**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540414(v=vs.85).aspx) as the data structure. The client driver must set the **TransferBufferMDL** member to the first [**MDL**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff554414(v=vs.85).aspx) in the chain that contains the transfer buffer. The USB driver stack ignores the **TransferBuffer**member when processing this URB.  Available in Windows 8. For information about using chained MDLs, see [How to Send Chained MDLs](http://msdn.microsoft.com/en-us/library/windows/hardware/hh450848(v=vs.85).aspx). |
| **URB\_FUNCTION\_RESET\_PIPE** | See URB\_FUNCTION\_SYNC\_RESET\_PIPE\_AND\_CLEAR\_STALL. |
| **URB\_FUNCTION\_SYNC\_RESET\_PIPE\_AND\_CLEAR\_STALL** | Resets the indicated pipe. If set, this URB is used with[**\_URB\_PIPE\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540419(v=vs.85).aspx).  **Note**  This URB replaces URB\_FUNCTION\_RESET\_PIPE.  The bus driver accomplishes three tasks in response to this URB:  First, for all pipes except isochronous pipes, this URB sends a CLEAR\_FEATURE request to clear the device's ENDPOINT\_HALT feature.  Second, the USB bus driver resets the data toggle on the host side, as required by the USB specification. The USB device should reset the data toggle on the device side when the bus driver clears its ENDPOINT\_HALT feature. Since some non-compliant devices do not support this feature, Microsoft provides the two additional URBs: URB\_FUNCTION\_SYNC\_CLEAR\_STALL and URB\_FUNCTION\_SYNC\_RESET\_PIPE. These allow client drivers to clear the ENDPOINT\_HALT feature on the device, or reset the pipe on the host side, respectively, without affecting the data toggle on the host side. If the device does not reset the data toggle when it should, then the client driver can compensate for this defect by not resetting the host-side data toggle. If the data toggle is reset on the host side but not on the device side, packets will get out of sequence, and the device might drop packets.  Third, after the bus driver has successfully reset the pipe, it resumes transfers with the next queued URB.  After a pipe reset, transfers resume with the next queued URB.  It is not necessary to clear a halt condition on a default control pipe. The default control pipe must always accept setup packets, and so if it halts, the USB stack will clear the halt condition automatically. The client driver does not need to take any special action to clear the halt condition on a default pipe.  All transfers must be aborted or canceled before attempting to reset the pipe.  This URB must be sent at PASSIVE\_LEVEL. |
| **URB\_FUNCTION\_SYNC\_RESET\_PIPE** | Clears the halt condition on the host side of a pipe. If set, this URB is used with [**\_URB\_PIPE\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540419(v=vs.85).aspx) as the data structure.  This URB allows a client to clear the halted state of a pipe without resetting the data toggle and without clearing the endpoint stall condition (feature ENDPOINT\_HALT). To clear a halt condition on the pipe, reset the host-side data toggle and clear a stall on the device with a single operation, use URB\_FUNCTION\_SYNC\_RESET\_PIPE\_AND\_CLEAR\_STALL.  The following status codes are important and have the indicated meaning:  USBD\_STATUS\_INVALID\_PIPE\_HANDLE  The **PipeHandle** is not valid  USBD\_STATUS\_ERROR\_BUSY  The endpoint has active transfers pending.  It is not necessary to clear a halt condition on a default control pipe. The default control pipe must always accept setup packets, and so if it halts, the USB stack will clear the halt condition automatically. The client driver does not need to take any special action to clear the halt condition on a default pipe.  All transfers must be aborted or canceled before attempting to reset the pipe.  This URB must be sent at PASSIVE\_LEVEL.  Available in Windows XP and later operating systems. |
| **URB\_FUNCTION\_SYNC\_CLEAR\_STALL** | Clears the stall condition on the endpoint. For all pipes except isochronous pipes, this URB sends a CLEAR\_FEATURE request to clear the device's ENDPOINT\_HALT feature. However, unlike the URB\_FUNCTION\_SYNC\_RESET\_PIPE\_AND\_CLEAR\_STALL function, this URB function does not reset the data toggle on the host side of the pipe. The USB specification requires devices to reset the device-side data toggle after the client clears the device's ENDPOINT\_HALT feature, but some non-compliant devices do not reset their data toggle properly. Client drivers that manage such devices can compensate for this defect by clearing the stall condition directly with URB\_FUNCTION\_SYNC\_CLEAR\_STALL instead of resetting the pipe with URB\_FUNCTION\_SYNC\_RESET\_PIPE\_AND\_CLEAR\_STALL. URB\_FUNCTION\_SYNC\_CLEAR\_STALL clears a stall condition on the device without resetting the host-side data toggle. This prevents a non-compliant device from interpreting the next packet as a retransmission and dropping the packet.  If set, the URB is used with [**\_URB\_PIPE\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540419(v=vs.85).aspx) as the data structure.  This URB function should be sent at PASSIVE\_LEVEL  Available in Windows XP and later operating systems. |
| **URB\_FUNCTION\_GET\_DESCRIPTOR\_FROM\_DEVICE** | Retrieves the device descriptor from a specific USB device. If set, the URB is used with[**\_URB\_CONTROL\_DESCRIPTOR\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540357(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_GET\_DESCRIPTOR\_FROM\_ENDPOINT** | Retrieves the descriptor from an endpoint on an interface for a USB device. If set, the URB is used with[**\_URB\_CONTROL\_DESCRIPTOR\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540357(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_SET\_DESCRIPTOR\_TO\_DEVICE** | Sets a device descriptor on a device. If set, the URB is used with [**\_URB\_CONTROL\_DESCRIPTOR\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540357(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_SET\_DESCRIPTOR\_TO\_ENDPOINT** | Sets an endpoint descriptor on an endpoint for an interface. If set, the URB is used with[**\_URB\_CONTROL\_DESCRIPTOR\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540357(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_SET\_FEATURE\_TO\_DEVICE** | Sets a USB-defined feature on a device. If set, the URB is used with [**\_URB\_CONTROL\_FEATURE\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540361(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_SET\_FEATURE\_TO\_INTERFACE** | Sets a USB-defined feature on an interface for a device. If set, the URB is used with[**\_URB\_CONTROL\_FEATURE\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540361(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_SET\_FEATURE\_TO\_ENDPOINT** | Sets a USB-defined feature on an endpoint for an interface on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_FEATURE\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540361(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_SET\_FEATURE\_TO\_OTHER** | Sets a USB-defined feature on a device-defined target on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_FEATURE\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540361(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_CLEAR\_FEATURE\_TO\_DEVICE** | Clears a USB-defined feature on a device. If set, the URB is used with [**\_URB\_CONTROL\_FEATURE\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540361(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_CLEAR\_FEATURE\_TO\_INTERFACE** | Clears a USB-defined feature on an interface for a device. If set, the URB is used with[**\_URB\_CONTROL\_FEATURE\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540361(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_CLEAR\_FEATURE\_TO\_ENDPOINT** | Clears a USB-defined feature on an endpoint, for an interface, on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_FEATURE\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540361(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_CLEAR\_FEATURE\_TO\_OTHER** | Clears a USB-defined feature on a device defined target on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_FEATURE\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540361(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_GET\_STATUS\_FROM\_DEVICE** | Retrieves status from a USB device. If set, the URB is used with [**\_URB\_CONTROL\_GET\_STATUS\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540378(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_GET\_STATUS\_FROM\_INTERFACE** | Retrieves status from an interface on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_GET\_STATUS\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540378(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_GET\_STATUS\_FROM\_ENDPOINT** | Retrieves status from an endpoint for an interface on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_GET\_STATUS\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540378(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_GET\_STATUS\_FROM\_OTHER** | Retrieves status from a device-defined target on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_GET\_STATUS\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540378(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_VENDOR\_DEVICE** | Sends a vendor-specific command to a USB device. If set, the URB is used with[**\_URB\_CONTROL\_VENDOR\_OR\_CLASS\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540393(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_VENDOR\_INTERFACE** | Sends a vendor-specific command for an interface on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_VENDOR\_OR\_CLASS\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540393(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_VENDOR\_ENDPOINT** | Sends a vendor-specific command for an endpoint on an interface on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_VENDOR\_OR\_CLASS\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540393(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_VENDOR\_OTHER** | Sends a vendor-specific command to a device-defined target on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_VENDOR\_OR\_CLASS\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540393(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_CLASS\_DEVICE** | Sends a USB-defined class-specific command to a USB device. If set, the URB is used with[**\_URB\_CONTROL\_VENDOR\_OR\_CLASS\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540393(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_CLASS\_INTERFACE** | Sends a USB-defined class-specific command to an interface on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_VENDOR\_OR\_CLASS\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540393(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_CLASS\_ENDPOINT** | Sends a USB-defined class-specific command to an endpoint, on an interface, on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_VENDOR\_OR\_CLASS\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540393(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_CLASS\_OTHER** | Sends a USB-defined class-specific command to a device defined target on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_VENDOR\_OR\_CLASS\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540393(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_GET\_CONFIGURATION** | Retrieves the current configuration on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_GET\_CONFIGURATION\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540365(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_GET\_INTERFACE** | Retrieves the current settings for an interface on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_GET\_INTERFACE\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540373(v=vs.85).aspx) as the data structure.  Available in Windows 2000, and Windows Vista and later operating systems. Not available in Windows XP. |
| **URB\_FUNCTION\_GET\_DESCRIPTOR\_FROM\_INTERFACE** | Retrieves the descriptor from an interface for a USB device. If set, the URB is used with[**\_URB\_CONTROL\_DESCRIPTOR\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540357(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_SET\_DESCRIPTOR\_TO\_INTERFACE** | Sets a descriptor for an interface on a USB device. If set, the URB is used with[**\_URB\_CONTROL\_DESCRIPTOR\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540357(v=vs.85).aspx) as the data structure. |
| **URB\_FUNCTION\_GET\_MS\_FEATURE\_DESCRIPTOR** | Retrieves a Microsoft OS feature descriptor from a USB device or an interface on a USB device. If set, the URB is used with [**\_URB\_OS\_FEATURE\_DESCRIPTOR\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540417(v=vs.85).aspx)as the data structure.  Available in Windows XP and later operation systems. |
| **URB\_FUNCTION\_OPEN\_STATIC\_STREAMS** | Opens streams in the specified bulk endpoint. If set, the URB is used with [**\_URB\_OPEN\_STATIC\_STREAMS**](http://msdn.microsoft.com/en-us/library/windows/hardware/hh406294(v=vs.85).aspx) as the data structure.  Available in Windows 8. For information about formatting an URB for an open-stream request, see [How to Open and Close Static Streams in a USB Bulk Endpoint](http://msdn.microsoft.com/en-us/library/windows/hardware/hh450846(v=vs.85).aspx). |
| **URB\_FUNCTION\_CLOSE\_STATIC\_STREAMS** | Closes all opened streams in the specified bulk endpoint. If set, the URB is used with[**\_URB\_PIPE\_REQUEST**](http://msdn.microsoft.com/en-us/library/windows/hardware/ff540419(v=vs.85).aspx) as the data structure.  Available in Windows 8. For information about formatting an URB for a close-stream request, see [How to Open and Close Static Streams in a USB Bulk Endpoint](http://msdn.microsoft.com/en-us/library/windows/hardware/hh450846(v=vs.85).aspx). |

10号功能对应的应该是**URB\_FUNCTION\_BULK\_OR\_INTERRUPT\_TRANSFER之后在这里进行单独的处理。**



在以上的文件中有我们需要的API的详细生命，最后查得10的Code对应的功能为



<http://msdn.microsoft.com/en-us/library/windows/hardware/ff540414(v=vs.85).aspx>