Android USB Camera(2) : UVC协议分析

标签： Android USB-Camera UVC 协议

2016-12-08 19:15 757人阅读 评论(0) 收藏 举报

目录(?) [+]

1. 前言

本文对USB的功能类协议USB Video Class（UVC）的具体设计进行介绍，但不会介绍USB基础协议，所以需要对USB基础协议有一定的了解，包括USB四大描述符以及四种传输方式。分析usb协议的重点在于描述符，这里将以实际设备的描述符为线索，贯穿、分析uvc协议。

2. UVC功能概述

UVC设备都是多Interface设备，这点同普通的u盘不同。UVC设备最起码有两个Interface，VideoControl（VC）Interface和VideoStream(VS) Interface； 这也是最常见的UVC设备。 Spec明确要求一个具有可用的，具有实际UVC功能的设备要有一个VC Interface，一个或多个VS Interface。

VCInterface用于进行配置，操控，设置UVC设备进入不同的功能状态，而VSInterface则负责视频数据流的传输；完整的UVC功能需依赖VS，VC Interfaces的配合才能实现。

3. IAD以及Interfaces

UVC功能需要使用一个VC Interface和一个或多个VS Interface的配合。Spec明确要求UVC设备必须使用一个Interface Association Descriptor（IAD）来描述这个包含了VC和VS的Interfaces集合。

此外USB协议中并没有专门朝设备索取IAD的命令，IAD需作为Configuration描述符的一部分发送给Host端。在该Configuration描述符的排列顺序中，IAD要放在它所包含的VC,VS描述符之前，且同一个IAD中所有的VC,VS Interface的Interface number必须连贯。

```
===>Device Descriptor<===
...

===>Configuration Descriptor<===
...

===>IAD Descriptor<===
bLength: 0x08
bDescriptorType: 0x0B
bFirstInterface: 0x00
bInterfaceCount: 0x02
bFunctionClass: 0x0E -> Video Interface Class
bFunctionSubClass: 0x03 -> Video Interface Collection
bFunctionProtocol: 0x00 -> PC_PROTOCOL_UNDEFINED protocol
iFunction: 0x00

===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x00
bAlternateSetting: 0x00
bNumEndpoints: 0x01
bInterfaceClass: 0x0E -> Video Interface Class
bInterfaceSubClass: 0x01 -> Video Control Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00
...

===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x01
bAlternateSetting: 0x00
bNumEndpoints: 0x00
bInterfaceClass: 0x0E -> Video Interface Class
bInterfaceSubClass: 0x02 -> Video Streaming Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00
...

===>IAD Descriptor<===
bLength: 0x08
bDescriptorType: 0x0B
bFirstInterface: 0x02
bInterfaceCount: 0x02
bFunctionClass: 0x01 -> Audio Interface Class
bFunctionSubClass: 0x01 -> Audio Control Interface SubClass
bFunctionProtocol: 0x00

```

```
iFunction: 0x04
English (United States) "USB2.0 MIC"

===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x02
bAlternateSetting: 0x00
bNumEndpoints: 0x00
bInterfaceClass: 0x01 -> Audio Interface Class
bInterfaceSubClass: 0x01 -> Audio Control Interface SubClass
bInterfaceProtocol: 0x00
CAUTION: This may be an invalid bInterfaceProtocol
iInterface: 0x04
English (United States) "USB2.0 MIC"
...

===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x03
bAlternateSetting: 0x00
bNumEndpoints: 0x00
bInterfaceClass: 0x01 -> Audio Interface Class
bInterfaceSubClass: 0x02 -> Audio Streaming Interface SubClass
bInterfaceProtocol: 0x00
CAUTION: This may be an invalid bInterfaceProtocol
iInterface: 0x04
English (United States) "USB2.0 MIC"
...
```

可以看到我所使用的uvc设备包含两个IAD，第一个IAD是视频输入设备，也就是摄像头。第二个IAD是音频输入设备，也就是麦克风，暂时忽略这个IAD。重点分析第一个IAD，它包含了两个Interface，一个VC Interface和一个VS Interface

4. VC Interface

VC主要用于控制UVC功能中的Units和Terminals，通过对不同的Units和Terminals发下不同的指令，从而影响UVC设备的当前工作状态；Host端发下的所有对UVC设备的操纵控制命令都是经由此Interface完成的

```
===>Interface Descriptor<===
bLength: 0x09
```

```
bDescriptorType: 0x04
bInterfaceNumber: 0x00
bAlternateSetting: 0x00
bNumEndpoints: 0x01
bInterfaceClass: 0x0E -> Video Interface Class
bInterfaceSubClass: 0x01 -> Video Control Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00

===>Class-Specific Video Control Interface Header Descriptor<===
bLength: 0x0D
bDescriptorType: 0x24
bDescriptorSubtype: 0x01
bcdVDC: 0x0100
wTotalLength: 0x0033 -> Validated
dwClockFreq: 0x02DC6C00 = (48000000) Hz
bInCollection: 0x01
baInterfaceNr[1]: 0x01
USB Video Class device: spec version 1.0

===>Video Control Input Terminal Descriptor<===
...

===>Video Control Processing Unit Descriptor<===
...

===>Video Control Output Terminal Descriptor<===
...

===>Endpoint Descriptor<===
bLength: 0x07
bDescriptorType: 0x05
bEndpointAddress: 0x81 -> Direction: IN - EndpointID: 1
bmAttributes: 0x03 -> Interrupt Transfer Type
wMaxPacketSize: 0x0010 = 1 transactions per microframe, 0x10 max bytes
bInterval: 0x0A

===>Class-specific VC Interrupt Endpoint Descriptor<===
bLength: 0x05
bDescriptorType: 0x25
bDescriptorSubtype: 0x03
wMaxTransferSize: 0x0010 = (16) Bytes
```

这个设备包含了一个Processing Unit、一个Input Terminal和Output Terminal，VC Interface需要对它们下发不同的命令，为此VC Interface需要包含一个控制端点（强制性要求），它使用的就是每个USB设备中默认的端点0。

另外一个Interrupt端点则是可选的，用来返回或通知Host端当前的UVC设备内部状态有变化。大部分情况下，一个UVC设备的VC Interface不需一定要实现此端点，但一旦UVC设备需要实现某些特定feature时，Spec会强制性要求实现该interrupt端点。

5. UVC设备内的各种Entity

UVC的Spec在制定时考虑到了诸多不同UVC设备可能存在的特性，为保证Spec具有最大范围的适用性和兼容性，将UVC设备内可能存在的每个功能模块划分为逻辑意义上的功能单元（Entity），并要求具体UVC设备的开发者必须为每个Entity指定ID号，且该ID号能唯一标识某一指定的功能单元，以便Host端的UVC驱动能依此对设备内不同的功能单元进行操作和配置。

Entity粗略可划分为Unit，Terminal两种。这都是逻辑意义上的划分，它们对应UVC设备内的某个功能单元。

5.1 Unit

Unit可以理解为构建出UVC设备功能的各功能单元，多个Unit按照一定的规则连接后就是一个完整的UVC功能设备。Spec规定Unit有一个或多个入口，一个出口；也就是说可以有多个Unit或Terminal作为一个Unit的输入源头，而数据在流经Unit后仅能作为其他Unit或Terminal的一个输入源头。

Spec中定义了三种Unit：Selector Unit (SU)，Processing Unit (PU)，Extension Unit (EU)。

```
===>Video Control Processing Unit Descriptor<===
bLength: 0x0B
bDescriptorType: 0x24
bDescriptorSubtype: 0x05
bUnitID: 0x02
bSourceID: 0x01
wMaxMultiplier: 0x0000
bControlSize: 0x02
bmControls : 0x3F 0x06
D00 = 1 yes - Brightness
D01 = 1 yes - Contrast
D02 = 1 yes - Hue
D03 = 1 yes - Saturation
D04 = 1 yes - Sharpness
D05 = 1 yes - Gamma
D06 = 0 no - White Balance Temperature

```

```
D07 = 0 no - White Balance Component
D08 = 0 no - Backlight Compensation
D09 = 1 yes - Gain
D10 = 1 yes - Power Line Frequency
D11 = 0 no - Hue, Auto
D12 = 0 no - White Balance Temperature, Auto
D13 = 0 no - White Balance Component, Auto
D14 = 0 no - Digital Multiplier
D15 = 0 no - Digital Multiplier Limit
iProcessing : 0x00
```

Processing Unit则代表了对采集的数据进行处理的单元，负责对采集到的图像特性进行调整设置，当Host端要求设定这些Processing Unit负责范围内的功能特性时，Processing Unit作为指定的命令响应者就需要对之进行响应和调整。

Spec定义中要求PU单元负责响应Host端对如下图像特性方面的设置调整要求（在该UVC设备确实支持这些特性可变，可供设置的前提下）

Usr Controls

• Brightness

• Contrast

• Hue

• Saturatio

• Sharpness

• Gamma

• Digital Multiplier (Zoom)

Auto Controls

• White Balance Temperature

• White Balance Component

• Blcklight Compensation

Other

• Gain

• Power Line Frequency

• Analog Video Standard

• Analog Video Lock Status

5.2 Terminal

Terminal是指整个UVC功能的入口和出口，逻辑意义上讲Terminal只有一个出口或者入口，它要么代表整个UVC功能的数据输入点，要么代表数据在流经UVC功能内部的各Unit功能单元后的最终流入点。

Spec中定义了两种Terminal：Input Terminal，Output Terminal。

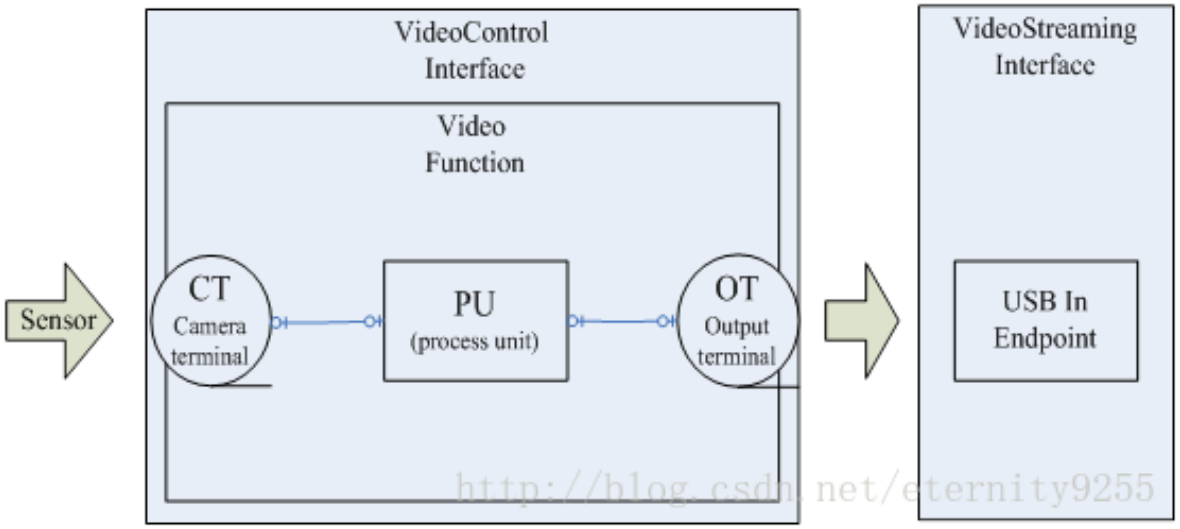
```
===>Video Control Input Terminal Descriptor<===
bLength: 0x12
bDescriptorType: 0x24
bDescriptorSubtype: 0x02
bTerminalID: 0x01
wTerminalType: 0x0201 = (ITT_CAMERA)
bAssocTerminal: 0x00
iTerminal: 0x00
===>Camera Input Terminal Data
wObjectiveFocalLengthMin: 0x0000
wObjectiveFocalLengthMax: 0x0000
wOcularFocalLength: 0x0000
bControlSize: 0x03
bmControls : 0x00 0x00 0x00
D00 = 0 no - Scanning Mode
D01 = 0 no - Auto-Exposure Mode
D02 = 0 no - Auto-Exposure Priority
D03 = 0 no - Exposure Time (Absolute)
D04 = 0 no - Exposure Time (Relative)
D05 = 0 no - Focus (Absolute)
D06 = 0 no - Focus (Relative)
D07 = 0 no - Iris (Absolute)
D08 = 0 no - Iris (Relative)
D09 = 0 no - Zoom (Absolute)
D10 = 0 no - Zoom (Relative)
D11 = 0 no - Pan (Absolute)
D12 = 0 no - Pan (Relative)
D13 = 0 no - Roll (Absolute)
D14 = 0 no - Roll (Relative)
D15 = 0 no - Tilt (Absolute)
D16 = 0 no - Tilt (Relative)
D17 = 0 no - Focus Auto
D18 = 0 no - Reserved
D19 = 0 no - Reserved
D20 = 0 no - Reserved
D21 = 0 no - Reserved

```

```
38  D22 = 0 no - Reserved
    D23 = 0 no  -  Reserved
```

Input Terminal可以理解为整个UVC功能的数据源头，它仅有一个outPin，可以连接到其他Unit的Input Pin作为该Unit的输入源。

```
1  ===>Video Control Output Terminal Descriptor<===
2  bLength: 0x09
3  bDescriptorType: 0x24
4  bDescriptorSubtype: 0x03
5  bTerminalID: 0x03
6  wTerminalType: 0x0101 = (TT_STREAMING)
7  bAssocTerminal: 0x00
8  bSourceID: 0x02
9  iTerminal: 0x00
```

Output Terminal则可以理解为当图像数据流程UVC设备内部的整个处理流程后的输出点，它仅有一个Input Pin， 整个UVC处理流程上的最后一个Unit会将它的Output Pin与Output Terminal的Input Pin连接在一起。



通过如上描述符所获得的信息，我们可以看出该USB Camera大致有三个逻辑功能意义上的Entity：CT(代表硬件上的图像数据采集源，Sensor)，PU（代表Sensor中可对采集所得图像数据进行加工的功能单元），OT(代表实际中USB模块的的ISO In端点)，并无SU单元来控制图像数据采集源的切换和选择（只有一个输入源），此外也没有对采集所得的数据进行个性化处理的功能，故也没有EU单元的存在（无EU描述符）。

6. VS Interface

VS Interface则专注与负责传输UVC设备的Video数据到Host端。若一UVC设备支持n种格式的Video数据，则它需要实现n个VS Interface，每个Interface对应一种专门的数据格式；而每个VSInterface则必须包含一个ISO或bulk端点来传输Video数据，一个可选的bulk端点专门用于传输静态图片数据（在实现了第三种静态图片拍摄机制的前提下）。

```
===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x01
bAlternateSetting: 0x00
bNumEndpoints: 0x00
bInterfaceClass: 0x0E -> Video Interface Class
bInterfaceSubClass: 0x02 -> Video Streaming Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00

===>Video Class-Specific VS Video Input Header Descriptor<===
bLength: 0x0E
bDescriptorType: 0x24
bDescriptorSubtype: 0x01
bNumFormats: 0x01
wTotalLength: 0x010F -> Validated
bEndpointAddress: 0x83 -> Direction: IN - EndpointID: 3
bmInfo: 0x00 -> Dynamic Format Change not Supported
bTerminalLink: 0x03
bStillCaptureMethod: 0x02 -> Still Capture Method 2
bTriggerSupport: 0x01 -> Hardware Triggering Support
bTriggerUsage: 0x00 -> Host will initiate still image capture
bControlSize: 0x01
Video Payload Format 1 0x00
D00 = 0 no - Key Frame Rate
D01 = 0 no - P Frame Rate
D02 = 0 no - Compression Quality
D03 = 0 no - Compression Window Size
D04 = 0 no - Generate Key Frame
D05 = 0 no - Update Frame Segment
D06 = 0 no - Reserved
D07 = 0 no - Reserved

===>Video Streaming MJPEG Format Type Descriptor<===
bLength: 0x0B
bDescriptorType: 0x24
bDescriptorSubtype: 0x06
```

```
bFormatIndex: 0x01
bNumFrameDescriptors: 0x05
bmFlags: 0x01 -> Sample Size is Fixed
bDefaultFrameIndex: 0x01
bAspectRatioX: 0x00
bAspectRatioY: 0x00
bmInterlaceFlags: 0x00
D00 = 0 non-Interlaced stream or variable
D01 = 0 2 fields per frame
D02 = 0 Field 1 not first
D03 = 0 Reserved
D4..5 = 0 Field patterns -> Field 1 only
D6..7 = 0 Display Mode -> Bob only
bCopyProtect: 0x00 -> Duplication Unrestricted

===>Video Streaming MJPEG Frame Type Descriptor<===
--->This is the Default (optimum) Frame index
bLength: 0x2A
bDescriptorType: 0x24
bDescriptorSubtype: 0x07
bFrameIndex: 0x01
bmCapabilities: 0x00
wWidth: 0x0280
wHeight: 0x01E0
dwMinBitRate: 0x0001F400
dwMaxBitRate: 0x00A8C000
dwMaxVideoFrameBufferSize: 0x00096000
dwDefaultFrameInterval: 0x00051615
bFrameIntervalType: 0x04
===>Additional Discrete Frame TypeData
dwFrameInterval[1]: 0x00028B0A
dwFrameInterval[2]: 0x00051615
dwFrameInterval[3]: 0x000A2C2A
dwFrameInterval[4]: 0x001E8480

===>Video Streaming MJPEG Frame Type Descriptor<===
bLength: 0x2A
bDescriptorType: 0x24
bDescriptorSubtype: 0x07
bFrameIndex: 0x02
bmCapabilities: 0x00
wWidth: 0x00A0
wHeight: 0x0078
dwMinBitRate: 0x0001F400
dwMaxBitRate: 0x00A8C000
dwMaxVideoFrameBufferSize: 0x00009600
dwDefaultFrameInterval: 0x00051615
bFrameIntervalType: 0x04
===>Additional Discrete Frame TypeData
```

```
dwFrameInterval[1]: 0x00028B0A
dwFrameInterval[2]: 0x00051615
dwFrameInterval[3]: 0x000A2C2A
dwFrameInterval[4]: 0x001E8480

===>Video Streaming MJPEG Frame Type Descriptor<===
bLength: 0x2A
bDescriptorType: 0x24
bDescriptorSubtype: 0x07
bFrameIndex: 0x03
bmCapabilities: 0x00
wWidth: 0x00B0
wHeight: 0x0090
dwMinBitRate: 0x0001F400
dwMaxBitRate: 0x00A8C000
dwMaxVideoFrameBufferSize: 0x0000C600
dwDefaultFrameInterval: 0x00051615
bFrameIntervalType: 0x04
===>Additional Discrete Frame TypeData
dwFrameInterval[1]: 0x00028B0A
dwFrameInterval[2]: 0x00051615
dwFrameInterval[3]: 0x000A2C2A
dwFrameInterval[4]: 0x001E8480

===>Video Streaming MJPEG Frame Type Descriptor<===
bLength: 0x2A
bDescriptorType: 0x24
bDescriptorSubtype: 0x07
bFrameIndex: 0x04
bmCapabilities: 0x00
wWidth: 0x0140
wHeight: 0x00F0
dwMinBitRate: 0x0001F400
dwMaxBitRate: 0x00A8C000
dwMaxVideoFrameBufferSize: 0x00025800
dwDefaultFrameInterval: 0x00051615
bFrameIntervalType: 0x04
===>Additional Discrete Frame TypeData
dwFrameInterval[1]: 0x00028B0A
dwFrameInterval[2]: 0x00051615
dwFrameInterval[3]: 0x000A2C2A
dwFrameInterval[4]: 0x001E8480

===>Video Streaming MJPEG Frame Type Descriptor<===
bLength: 0x2A
bDescriptorType: 0x24
bDescriptorSubtype: 0x07
bFrameIndex: 0x05
bmCapabilities: 0x00
```

```
wWidth: 0x0160
wHeight: 0x0120
dwMinBitRate: 0x0001F400
dwMaxBitRate: 0x00A8C000
dwMaxVideoFrameBufferSize: 0x00031800
dwDefaultFrameInterval: 0x00051615
bFrameIntervalType: 0x04
===>Additional Discrete Frame TypeData
dwFrameInterval[1]: 0x00028B0A
dwFrameInterval[2]: 0x00051615
dwFrameInterval[3]: 0x000A2C2A
dwFrameInterval[4]: 0x001E8480
```

可以看出这个设备只支持一种视频格式”MJPEG”，所以它只有一个VS Interface，而同一种视频格式下有很多参数可以调整，比如说图像分辨率，每秒的帧速率等。其参数组合信息就是紧跟在Format Type Descriptor之后的Frame Type Descriptor。

在采用不同的参数组合时，UVC设备进行数据传输时，所需占用的USB数据带宽显然有很大的不同；因此一个VS Interface下需要有多个setting，每个settnig对应一个固定的Video参数设置组合，也就对应了一个固定参数值的Video数据，该setting下每秒要传输的帧数（帧速率），图像分辨率等均固定，进而对USB的带宽要求也固定。

```

===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x01
bAlternateSetting: 0x01
bNumEndpoints: 0x01
bInterfaceClass: 0x0E -> Video Interface Class
bInterfaceSubClass: 0x02 -> Video Streaming Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00

===>Endpoint Descriptor<===
bLength: 0x07
bDescriptorType: 0x05
bEndpointAddress: 0x83 -> Direction: IN - EndpointID: 3
bmAttributes: 0x05 -> Isochronous Transfer Type
Synchronization Type = Asynchronous
Bulk Transfer Type
wMaxPacketSize: 0x00C0 = 1 transactions per microframe, 0xC0 max bytes
bInterval: 0x01

```

```
===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x01
bAlternateSetting: 0x02
bNumEndpoints: 0x01
bInterfaceClass: 0x0E -> Video Interface Class
bInterfaceSubClass: 0x02 -> Video Streaming Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00

===>Endpoint Descriptor<===
bLength: 0x07
bDescriptorType: 0x05
bEndpointAddress: 0x83 -> Direction: IN - EndpointID: 3
bmAttributes: 0x05 -> Isochronous Transfer Type
Synchronization Type = Asynchronous
Bulk Transfer Type
wMaxPacketSize: 0x0180 = 1 transactions per microframe, 0x180 max bytes
bInterval: 0x01

===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x01
bAlternateSetting: 0x03
bNumEndpoints: 0x01
bInterfaceClass: 0x0E -> Video Interface Class
bInterfaceSubClass: 0x02 -> Video Streaming Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00

===>Endpoint Descriptor<===
bLength: 0x07
bDescriptorType: 0x05
bEndpointAddress: 0x83 -> Direction: IN - EndpointID: 3
bmAttributes: 0x05 -> Isochronous Transfer Type
Synchronization Type = Asynchronous
Bulk Transfer Type
wMaxPacketSize: 0x0200 = 1 transactions per microframe, 0x200 max bytes
bInterval: 0x01

===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x01
bAlternateSetting: 0x04
bNumEndpoints: 0x01
bInterfaceClass: 0x0E -> Video Interface Class
```

```
bInterfaceSubClass: 0x02 -> Video Streaming Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00

===>Endpoint Descriptor<===
bLength: 0x07
bDescriptorType: 0x05
bEndpointAddress: 0x83 -> Direction: IN - EndpointID: 3
bmAttributes: 0x05 -> Isochronous Transfer Type
Synchronization Type = Asynchronous
Bulk Transfer Type
wMaxPacketSize: 0x0280 = 1 transactions per microframe, 0x280 max bytes
bInterval: 0x01

===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x01
bAlternateSetting: 0x05
bNumEndpoints: 0x01
bInterfaceClass: 0x0E -> Video Interface Class
bInterfaceSubClass: 0x02 -> Video Streaming Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00

===>Endpoint Descriptor<===
bLength: 0x07
bDescriptorType: 0x05
bEndpointAddress: 0x83 -> Direction: IN - EndpointID: 3
bmAttributes: 0x05 -> Isochronous Transfer Type
Synchronization Type = Asynchronous
Bulk Transfer Type
wMaxPacketSize: 0x0320 = 1 transactions per microframe, 0x320 max bytes
bInterval: 0x01

===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x01
bAlternateSetting: 0x06
bNumEndpoints: 0x01
bInterfaceClass: 0x0E -> Video Interface Class
bInterfaceSubClass: 0x02 -> Video Streaming Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00

===>Endpoint Descriptor<===
bLength: 0x07
bDescriptorType: 0x05
```

```
bEndpointAddress: 0x83 -> Direction: IN - EndpointID: 3
bmAttributes: 0x05 -> Isochronous Transfer Type
Synchronization Type = Asynchronous
Bulk Transfer Type
wMaxPacketSize: 0x03C0 = 1 transactions per microframe, 0x3C0 max bytes
bInterval: 0x01

===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x01
bAlternateSetting: 0x07
bNumEndpoints: 0x01
bInterfaceClass: 0x0E -> Video Interface Class
bInterfaceSubClass: 0x02 -> Video Streaming Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00

===>Endpoint Descriptor<===
bLength: 0x07
bDescriptorType: 0x05
bEndpointAddress: 0x83 -> Direction: IN - EndpointID: 3
bmAttributes: 0x05 -> Isochronous Transfer Type
Synchronization Type = Asynchronous
Bulk Transfer Type
wMaxPacketSize: 0x0A80 = 2 transactions per microframe, 0x280 max bytes
bInterval: 0x01

===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x01
bAlternateSetting: 0x08
bNumEndpoints: 0x01
bInterfaceClass: 0x0E -> Video Interface Class
bInterfaceSubClass: 0x02 -> Video Streaming Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00

===>Endpoint Descriptor<===
bLength: 0x07
bDescriptorType: 0x05
bEndpointAddress: 0x83 -> Direction: IN - EndpointID: 3
bmAttributes: 0x05 -> Isochronous Transfer Type
Synchronization Type = Asynchronous
Bulk Transfer Type
wMaxPacketSize: 0x0B20 = 2 transactions per microframe, 0x320 max bytes
bInterval: 0x01

```

```
===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x01
bAlternateSetting: 0x09
bNumEndpoints: 0x01
bInterfaceClass: 0x0E -> Video Interface Class
bInterfaceSubClass: 0x02 -> Video Streaming Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00

===>Endpoint Descriptor<===
bLength: 0x07
bDescriptorType: 0x05
bEndpointAddress: 0x83 -> Direction: IN - EndpointID: 3
bmAttributes: 0x05 -> Isochronous Transfer Type
Synchronization Type = Asynchronous
Bulk Transfer Type
wMaxPacketSize: 0x0BE0 = 2 transactions per microframe, 0x3E0 max bytes
bInterval: 0x01

===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x01
bAlternateSetting: 0x0A
bNumEndpoints: 0x01
bInterfaceClass: 0x0E -> Video Interface Class
bInterfaceSubClass: 0x02 -> Video Streaming Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00

===>Endpoint Descriptor<===
bLength: 0x07
bDescriptorType: 0x05
bEndpointAddress: 0x83 -> Direction: IN - EndpointID: 3
bmAttributes: 0x05 -> Isochronous Transfer Type
Synchronization Type = Asynchronous
Bulk Transfer Type
wMaxPacketSize: 0x1380 = 3 transactions per microframe, 0x380 max bytes
bInterval: 0x01

===>Interface Descriptor<===
bLength: 0x09
bDescriptorType: 0x04
bInterfaceNumber: 0x01
bAlternateSetting: 0x0B
bNumEndpoints: 0x01
bInterfaceClass: 0x0E -> Video Interface Class
```

```
bInterfaceSubClass: 0x02 -> Video Streaming Interface SubClass
bInterfaceProtocol: 0x00
iInterface: 0x00

===>Endpoint Descriptor<===
bLength: 0x07
bDescriptorType: 0x05
bEndpointAddress: 0x83 -> Direction: IN - EndpointID: 3
bmAttributes: 0x05 -> Isochronous Transfer Type
Synchronization Type = Asynchronous
Bulk Transfer Type
wMaxPacketSize: 0x13E8 = 3 transactions per microframe, 0x3E8 max bytes
bInterval: 0x01
```

此外，对于每个VS Interface来讲，必须实现setting0，且setting0不代表任何实际的Video流和参数组合，当指定setting0时就是要求UVC设备暂停数据传输，放弃对USB带宽的占用。

7. 总结

UVC同MSC一样，系USB框架下的功能类协议，但却与MSC有着较大差异。MSC功能采用Control+Bulk传输完成，其枚举流程，描述符配置较为清晰，控制传输阶段简单，定义的类功能控制命令较少。而BULK传输阶段则较为复杂繁琐，出错机制，续传机制等要求较为严格。

而UVC则刚好相反，它采用Control+ISO传输机制实现（BULK和INTR机制为可选特性），其枚举流程，描述符配置较为复杂，繁琐，定义了诸多的类控制命令，Entity等；而具体的数据传输阶段即ISO传输较为简单。