Universal Serial Bus Device Class Definition for Video Devices: Video Device Examples

Revision 1.1

June 1, 2005

Contributors

Abdul R. Ismail	Intel Corp.
Akihiro Tanabe	Canon Inc.
Anand Ganesh	Microsoft Corp.
Andy Hodgson	STMicroelectronics
Anshuman Saxena	Texas Instruments
Bertrand Lee	Microsoft Corp.
Charng Lee	Sunplus Technology Co., Ltd
David Goll	Microsoft Corp.
Eric Luttmann	Cypress Semiconductor Corp.
Fernando Urbina	Apple Computer Inc.
Geert Knapen	Philips Electronics
Geraud Mudry	Logitech Inc.
Hiro Kobayashi	Microsoft Corp.
Jean-Michel Chardon	Logitech Inc.
Jeff Zhu	Microsoft Corp.
Ken-ichiro Ayaki	Fujifilm
Mitsuo Niida	Canon Inc.
Nobuo Kuchiki	Sanyo Electric Co., Ltd
Olivier Lechenne	Logitech Inc.
Paul Thacker	STMicroelectronics
Remy Zimmermann	Logitech Inc.
Shinichi Hatae	Canon Inc.
Steve Miller	STMicroelectronics
Tachio Ono	Canon Inc.
Takashi Sato	Philips Electronics
Yoichi Hirata	Matsushita Electric Industrial Co., Ltd

Revision History

Version	Date	Description
1.0	December 9, 2003	Initial release
1.1	June 1st, 2005	Update document for compliance with UVC version 1.08a Added Description of the Controls for the Second Example (Section 3.4). Change VDC to UVC in Tables 2.5 and 3.5. (RR0064)

Table of Contents

1 Introduction 1
1.1 Purpose 1
1.2 Related Documents 1
2 Desktop Video Camera Example 2
2.1 Product Description 2
2.2 Descriptor Hierarchy 3
2.3 Descriptors 5
2.3.1 Device Descriptor 5
2.3.2 Configuration Descriptor 5
2.3.3 Interface Association Descriptor 6
2.3.4 VideoControl Interface Descriptor 6
2.3.4.1 Standard VC Interface Descriptor 6
2.3.4.2 Class-specific VC Interface Descriptor 7
2.3.4.3 Input Terminal Descriptor (Camera) 7
2.3.4.4 Input Terminal Descriptor (Composite) 8
2.3.4.5 Output Terminal Descriptor 8
2.3.4.6 Selector Unit Descriptor 9
2.3.4.7 Processing Unit Descriptor 9
2.3.4.8 Standard Interrupt Endpoint Descriptor 10
2.3.4.9 Class-specific Interrupt Endpoint Descriptor 10
2.3.5 VideoStreaming Interface Descriptor 10
2.3.5.1 Zero-bandwidth Alternate Setting 0 10
2.3.5.1.1 Standard VS Interface Descriptor 10
2.3.5.1.2 Class-specific VS Header Descriptor (Input) 11
2.3.5.1.3 Class-specific VS Format Descriptor 11
2.3.5.1.4 Class-specific VS Frame Descriptor 12
2.3.5.2 Operational Alternate Setting 1 13
2.3.5.2.1 Standard VS Interface Descriptor 13
2.3.5.2.2 Standard VS Isochronous Video Data Endpoint Descriptor 13
2.3.6 String Descriptors 14
2.3.6.1 String Descriptor Zero 14
2.3.6.2 Manufacturer String Descriptor (Index 1) 14
2.3.6.3 Product String Descriptor (Index 2) 14
2.4 Requests 15
2.4.1 Standard Requests 15
2.4.1.1 Set Interface 15
2.4.2 Class-specific Requests 15
2.4.2.1 VideoControl Interface Requests 15
2.4.2.1.1 Set Selector Control Request 16
2.4.2.1.2 Get Selector Control Request 16
2.4.2.1.3 Set Brightness Control Request 17
2.4.2.1.4 Get Brightness Control Request 17
2.4.2.1.5 Set Power Mode Control Request 18
2.4.2.1.6 Get Power Mode Control Request 18
2.4.2.1.7 Request Error Code Control 19

2.4.2.2 VideoStreaming Requests 19
2.4.2.2.1 Set Video Probe Control Request 19
2.4.2.2.2 Get Video Probe Control Request 20
2.4.2.2.3 Set Video Commit Control Request 20
2.4.2.2.4 Get Video Commit Control Request 21
3 Video Camera Player Example 22
3.1 Product Description 22
3.2 Descriptor Hierarchy 23
3.3 Descriptors 24
3.3.1 Device Descriptor 24
3.3.2 Configuration Descriptor 24
3.3.3 Interface Association Descriptor 25
3.3.4 Video Control Interface Descriptor 25
3.3.4.1 Standard VC Interface Descriptor 25
3.3.4.2 Class-specific VC Interface Descriptor 26
3.3.4.3 Input Terminal Descriptor (Camera) 26
3.3.4.4 Input Terminal Descriptor (Media Transport) 27
3.3.4.5 Selector Unit Descriptor 27
3.3.4.6 Output Terminal Descriptor 28
3.3.4.7 Standard Interrupt Endpoint Descriptor 28
3.3.4.8 Class-specific Interrupt Endpoint Descriptor 28
3.3.5 Video Streaming Interface Descriptor 28
3.3.5.1 Zero-bandwidth Alternate Setting 0 29
3.3.5.1.1 Standard VS Interface Descriptor 29
3.3.5.1.2 Class-specific VS Header Descriptor (Input) 29
3.3.5.1.3 Class-specific VS Format Descriptor 30
3.3.5.1.4 Class-specific VS Frame Descriptor 31
3.3.5.1.5 Class-specific Still Image Frame Descriptor 31
3.3.5.1.6 Class-specific Color Matching Descriptor 32
3.3.5.1.7 Standard Bulk Endpoint Descriptor 32
3.3.5.2 Operational Alternate Setting 1 32
3.3.5.2.1 Standard VS Interface Descriptor 32
3.3.5.2.2 Standard VS Isochronous Video Data Endpoint Descriptor 33
3.3.5.2.3 Standard Bulk Endpoint Descriptor 33
3.3.6 String Descriptors 33
3.3.6.1 String Descriptor Zero 34
3.3.6.2 Manufacturer String Descriptor (Index 1) 34
3.3.6.3 Product String Descriptor (Index 2) 34
3.3.6.4 Serial Number String Descriptor (Index 3) 34
3.3.6.5 Product String Descriptor (Index 4) 35
3.4 Requests 35
3.4.1 Standard Requests 35
3.4.1.1 Set Interface 35
3.4.2 Class-specific Requests 35
3.4.2.1 VideoControl Interface Requests 35
3.4.2.1.1 Set Selector Control Request 36

3.4.2.1.2 Get Selector Control Request 36
3.4.2.1.3 Set Zoom (Absolute) Control Request 37
3.4.2.1.4 Get Zoom (Absolute) Control Request 37
3.4.2.1.5 Set Transport Control Request 38
3.4.2.1.6 Get Transport Control Request 38
3.4.2.1.7 Set Media Information Control Request 39
3.4.2.1.8 Get Media Information Control Request 39
3.4.2.1.9 Set Time Code Information Control Request 40
3.4.2.1.10 Get Time Code Information Control Request 40
3.4.2.1.11 Set Power Mode Control Request 41
3.4.2.1.12 Get Power Mode Control Request 41
3.4.2.1.13 Request Error Code Control 42
3.4.2.2 VideoStreaming Requests 42
3.4.2.2.1 Set Video Probe Control Request 42
3.4.2.2.2 Get Video Probe Control Request 43
3.4.2.2.3 Set Video Commit Control Request 43
3.4.2.2.4 Get Video Commit Control Request 44

List of Tables

Table 2-1 Device Descriptor 5
Table 2-2 Configuration Descriptor 5
Table 2-3 Standard Video Interface Collection IAD 6
Table 2-4 Standard VC Interface Descriptor 6
Table 2-5 Class-specific VC Interface Descriptor 7
Table 2-6 Input Terminal Descriptor (Camera) 7
Table 2-7 Input Terminal Descriptor (Composite) 8
Table 2-8 Output Terminal Descriptor 8
Table 2-9 Selector Unit Descriptor 9
Table 2-10 Processing Unit Descriptor 9
Table 2-11 Standard Interrupt Endpoint Descriptor 10
Table 2-12 Class-specific Interrupt Endpoint Descriptor 10
Table 2-13 Standard VS Interface Descriptor 10
Table 2-14 Class-specific VS Header Descriptor (Input) 11
Table 2-15 Class-specific VS Format Descriptor 12
Table 2-16 Class-specific VS Frame Descriptor 12
Table 2-17 Standard VS Interface Descriptor 13
Table 2-18 Standard VS Isochronous Video Data Endpoint Descriptor 13
Table 2-19 String Descriptor Zero 14
Table 2-20 Manufacturer String Descriptor (Index 1) 14
Table 2-21 Product String Descriptor (Index 2) 14
Table 2-22 Set Interface 15
Table 2-23 Set Selector Control Request 16
Table 2-24 Get Selector Control Request 16
Table 2-25 Set Brightness Control Request 17
Table 2-26 Get Brightness Control Request 17
Table 2-27 Set Power Mode Control Request 18
Table 2-28 Get Power Mode Control Request 18
Table 2-29 Request Error Code Control 19
Table 2-30 Set Video Probe Control Request 19
Table 2-31 Get Video Probe Control Request 20
Table 2-32 Set Video Commit Control Request 20
Table 2-33 Get Video Commit Control Request 21
Table 3-1 Device Descriptor 24
Table 3-2 Configuration Descriptor 24
Table 3-3 Standard Video Interface Collection IAD 25
Table 3-4 Standard VC Interface Descriptor 25
Table 3-5 Class-specific VC Interface Descriptor 26
Table 3-6 Input Terminal Descriptor (Camera) 26
Table 3-7 Input Terminal Descriptor (Media Transport) 27
Table 3-8 Selector Unit Descriptor 27
Table 3-9 Output Terminal Descriptor 28
Table 3-10 Standard Interrupt Endpoint Descriptor 28
Table 3-11 Class-specific Interrupt Endpoint Descriptor 28
Table 3-12 Standard VS Interface Descriptor 29

Table 3-13 Class-specific VS Header Descriptor (Input) 29
Table 3-14 Class-specific VS Format Descriptor 30
Table 3-15 Class-specific VS Frame Descriptor 31
Table 3-16 Class-specific Still Image Frame Descriptor 31
Table 3-17 Class-specific Color Matching Descriptor 32
Table 3-18 Standard Bulk Endpoint Descriptor 32
Table 3-19 Standard VS Interface Descriptor 32
Table 3-20 Standard VS Isochronous Video Data Endpoint Descriptor 33
Table 3-21 Standard Bulk Endpoint Descriptor 33
Table 3-22 String Descriptor Zero 34
Table 3-23 Manufacturer String Descriptor (Index 1) 34
Table 3-24 Product String Descriptor (Index 2) 34
Table 3-25 Serial Number String Descriptor (Index 3) 34
Table 3-26 Product String Descriptor (Index 4) 35
Table 3-27 Set Interface 35
Table 3-28 Set Selector Control Request 36
Table 3-29 Get Selector Control Request 36
Table 3-30 Set Zoom (Absolute) Control Request 37
Table 3-31 Get Zoom (Absolute) Control Request 37
Table 3-32 Set Transport Control Request 38
Table 3-33 Get Transport Control Request 38
Table 3-34 Get Transport Control Request 39
Table 3-35 Set Time Code Information Control Request 40
Table 3-36 Get Transport Control Request 40
Table 3-37 Set Power Mode Control Request 41
Table 3-38 Get Power Mode Control Request 41
Table 3-39 Request Error Code Control 42
Table 3-40 Set Video Probe Control Request 42
Table 3-41 Get Video Probe Control Request 43
Table 3-42 Set Video Commit Control Request 43
Table 3-43 Get Video Commit Control Request 44

List of Figures

Figure 2-1 USB Video Camera Topology 2
Figure 2-2 USB Video Camera Descriptor Hierarchy 4
Figure 3-1 USB Video Camera Player Topology 22
Figure 3-2 USB Video Camera Player Descriptor Hierarchy 23

1 Introduction

1.1 Purpose

This document describes in detail example implementations of USB video device that conform to the video device class specification. This document is provided as an aid to implementers of the USB Video Device Class specification and, as such, is informative only. Should a conflict arise between this document and a specification, the specification shall take precedence.

1.2 Related Documents

USB Specification Revision 2.0, April 27, 2000, www.usb.org
USB Device Class Definition for Video Devices, www.usb.org
Interface Association Descriptor ECN, www.usb.org

2 Desktop Video Camera Example

2.1 Product Description

The device described in this section is a full-speed desktop video camera (or "webcam"). This particular implementation has two video sources, a CCD sensor and a composite input connector on the device that can be switched by using a selector unit on the device. It streams video data through an isochronous pipe to the host in MJPEG format at a single frame size (176x144) at a single frame rate (15 fps), and functions as an asynchronous source, using its internal clock as a reference. It is capable of notifying the host of button press events to trigger still-image capture (using Method 1), and contains a processing unit that is capable of adjusting the brightness level of the video stream. This example implementation will assume that we use one Video Interface Collection. The VideoControl interface (interface number 0) and the VideoStreaming interface (interface number 1) are part of this Video Interface Collection.

The following figure represents the internal topology of the camera.

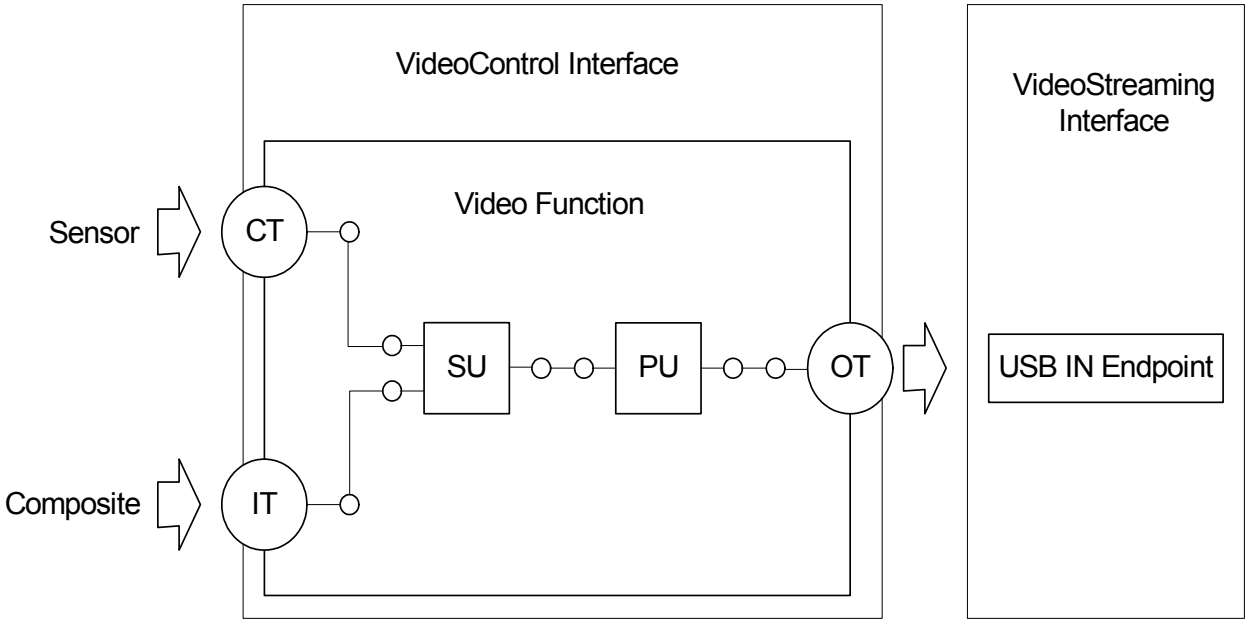

Figure 2-1 USB Video Camera Topology

The video function contains two input terminals, one representing the sensor and the other representing the composite video-input connector. The video streams captured by these terminals go through any necessary analogue-to-digital conversion, and are routed into a selector unit. The selected video stream is then sent to a processing unit for video signal processing. The output is routed to a single output terminal which transmits the video stream to the host via a USB IN endpoint. This endpoint is part of the single VideoStreaming interface that this device contains.

The internals of the video function (unit and terminal topology) are presented to the host through the (mandatory) VideoControl interface.

2.2 Descriptor Hierarchy

This USB camera device uses a Video Interface Collection that includes the VideoControl interface (interface 0) and a single VideoStreaming interface (interface 1). The VideoStreaming interface features two alternate settings. The first alternate setting (0) has zero bandwidth associated with it (implied by the lack of an isochronous endpoint), so switching to this alternate setting frees all allocated bandwidth on the USB for this device. Alternate setting 1 is the operational part of the interface and contains the isochronous endpoint to supply the host with video data.

This figure presents the descriptor hierarchy.

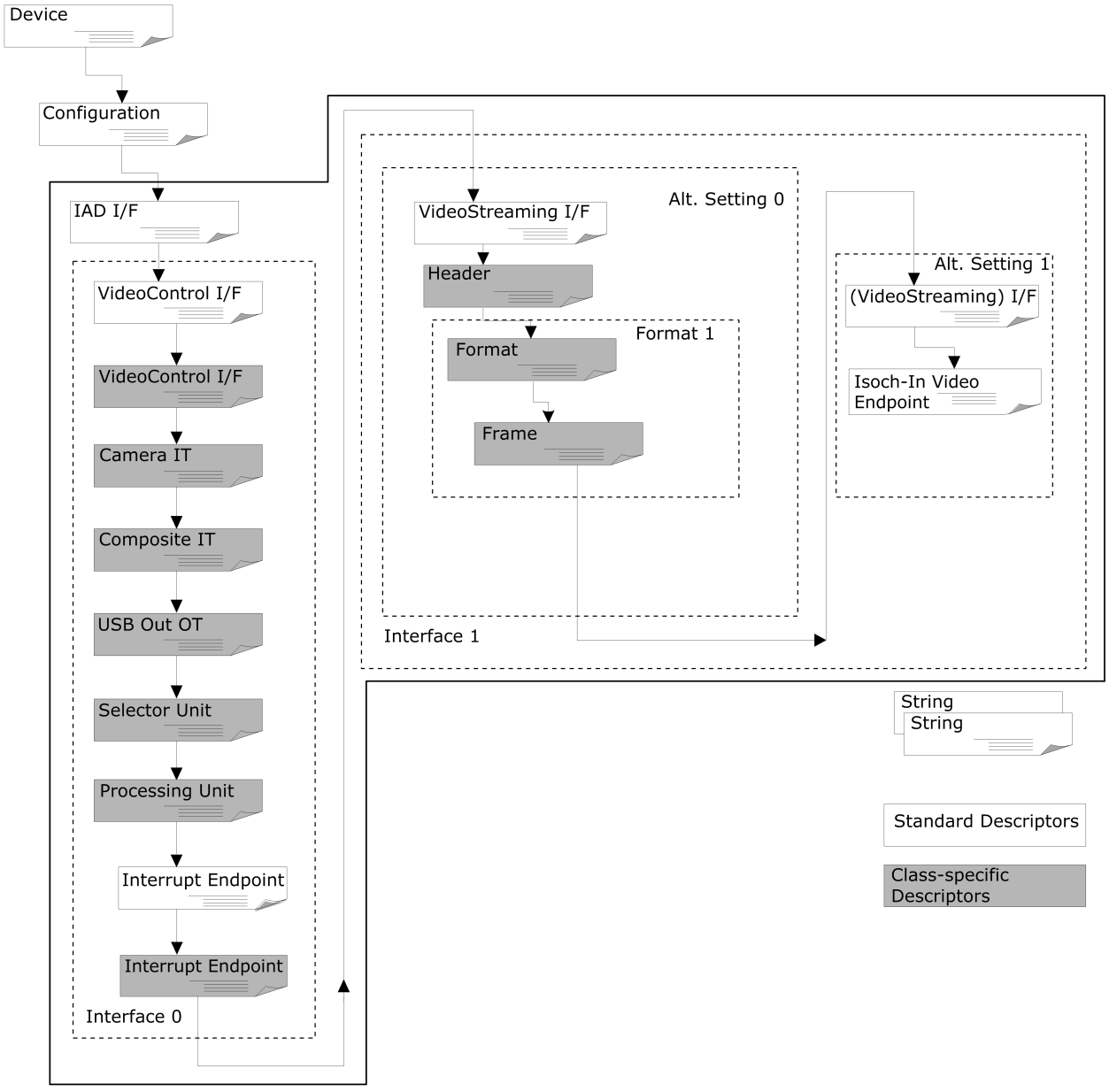

Figure 2-2 USB Video Camera Descriptor Hierarchy

2.3 Descriptors

The following sections present all the descriptors that are used to describe the device.

2.3.1 Device Descriptor

Table 2-1 Device Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x12	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x01	DEVICE descriptor
2	**bcdUSB**	2	0x0200	2.00 – current revision of the USB specification
4	**bDeviceClass**	1	0xEF	Miscellaneous Device Class
5	**bDeviceSubClass**	1	0x02	Common Class
6	**bDeviceProtocol**	1	0x01	Interface Association Descriptor
7	**bMaxPacketSize0**	1	0x08	Control endpoint packet size is 8 bytes
8	**idVendor**	2	0xXXXX	Vendor ID
10	**idProduct**	2	0xXXXX	Product ID
12	**bcdDevice**	2	0xXXXX	Device release code
14	**iManufacturer**	1	0x01	Index to string descriptor that contains the string <Your Name> in Unicode
15	**iProduct**	1	0x02	Index to string descriptor that contains the string <Your Product Name> in Unicode
16	**iSerialNumber**	1	0x00	Unused
17	**bNumConfigurations**	1	0x01	One configuration

2.3.2 Configuration Descriptor

Table 2-2 Configuration Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x09	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x02	CONFIGURATION descriptor
2	**wTotalLength**	2	0x00C0	Length of the total configuration block, including this descriptor, in bytes
4	**bNumInterfaces**	1	0x02	This device has two interfaces
5	**bConfigurationValue**	1	0x01	ID of this configuration
6	**iConfiguration**	1	0x00	Unused
7	**bmAttributes**	1	0x80	Bus-powered device, no remote wakeup capability
8	**bMaxPower**	1	0xFA	500 mA maximum power consumption

2.3.3 Interface Association Descriptor

This device uses an Interface Association Descriptor to describe its Video Interface Collection.

Table 2-3 Standard Video Interface Collection IAD

Offset	Field	Size	Value	Description
0	**bLength**	1	0x08	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x0B	INTERFACE ASSOCIATION Descriptor
2	**bFirstInterface**	1	0x00	Interface number of the VideoControl interface that is associated with this function
3	**bInterfaceCount**	1	0x02	Number of contiguous Video interfaces that are associated with this function
4	**bFunctionClass**	1	0x0E	CC_VIDEO
5	**bFunctionSubClass**	1	0x03	SC_VIDEO_INTERFACE_COLLECTION
6	**bFunctionProtocol**	1	0x00	Not used. Must be set to PC_PROTOCOL_UNDEFINED.
7	**iFunction**	1	0x02	Index to string descriptor that contains the string <Your Product Name> in Unicode. Have to match iInterface field in Standard VC Interface Descriptor.

2.3.4 VideoControl Interface Descriptor

The VideoControl interface describes the device structure (video function topology) and is used to manipulate the video controls.

2.3.4.1 Standard VC Interface Descriptor

The VideoControl interface has no dedicated endpoints associated with it. It uses the default pipe (endpoint 0) for all communication purposes, except for event notification, in which case the interrupt endpoint is used. Class-specific video control requests are sent using the default pipe.

Table 2-4 Standard VC Interface Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x09	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x04	INTERFACE descriptor type
2	**bInterfaceNumber**	1	0x00	Index of this interface
3	**bAlternateSetting**	1	0x00	Index of this setting
4	**bNumEndpoints**	1	0x01	1 endpoint (interrupt endpoint)
5	**bInterfaceClass**	1	0x0E	CC_VIDEO
6	**bInterfaceSubClass**	1	0x01	SC_VIDEOCONTROL

7	**bInterfaceProtocol**	1	0x00	Not used. Must be set to PC_PROTOCOL_UNDEFINED.
8	**iInterface**	1	0x02	Index to string descriptor that contains the string <Your Product Name> in Unicode. Have to match iFunction field of the Standard Video Interface Collection IAD.

2.3.4.2 Class-specific VC Interface Descriptor

The class-specific VC interface descriptor is always headed by a header descriptor that contains general information about the VideoControl interface. It contains all the pointers needed to describe the video interface collection associated with the described video function.

Table 2-5 Class-specific VC Interface Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x0D	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubType**	1	0x01	VC_HEADER subtype
3	**bcdUVC**	2	0x0110	Revision of class specification that this device is based upon. For this example, the device complies with Video Class specification version 1.1.
5	**wTotalLength**	2	0x0042	Total size of class-specific descriptors
7	**dwClockFrequency**	4	0x005B8D80	Use of this field has been deprecated.This device will provide timestamps and a device clock reference based on a 6MHz clock.
11	**bInCollection**	1	0x01	Number of streaming interfaces.
12	**baInterfaceNr(1)**	1	0x01	VideoStreaming interface 1 belongs to this VideoControl interface.

2.3.4.3 Input Terminal Descriptor (Camera)

This descriptor describes the input terminal that represents the CCD sensor (and associated A/D converter). The resulting digital video stream leaves the input terminal through the single output pin.

Table 2-6 Input Terminal Descriptor (Camera)

Offset	Field	Size	Value	Description
0	**bLength**	1	0x11	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x02	VC_INPUT_TERMINAL subtype
3	**bTerminalID**	1	0x01	ID of this input terminal
4	**wTerminalType**	2	0x0201	ITT_CAMERA type. This terminal is

				a camera terminal representing the CCD sensor.
6	**bAssocTerminal**	1	0x00	No association
7	**iTerminal**	1	0x00	Unused
8	**wObjectiveFocalLengthMin**	2	0x0000	No optical zoom supported
10	**wObjectiveFocalLengthMax**	2	0x0000	No optical zoom supported
12	**wOcularFocalLength**	2	0x0000	No optical zoom supported
14	**bControlSize**	1	0x02	The size of the **bmControls** is 2 bytes (this terminal doesn't implement any controls).
15	**bmControls**	2	0x0000	No controls are supported.

2.3.4.4 Input Terminal Descriptor (Composite)

This descriptor describes the input terminal that represents the composite video-input connector (and associated A/D converter). The resulting digital video stream leaves the input terminal through the single output pin.

Table 2-7 Input Terminal Descriptor (Composite)

Offset	Field	Size	Value	Description
0	**bLength**	1	0x08	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x02	VC_INPUT_TERMINAL subtype
3	**bTerminalID**	1	0x02	ID of this input terminal
4	**wTerminalType**	2	0x0401	COMPOSITE_CONNECTOR type. This terminal is the composite connector.
6	**bAssocTerminal**	1	0x00	No association
7	**iTerminal**	1	0x00	Unused

2.3.4.5 Output Terminal Descriptor

This descriptor describes the output terminal that represents the USB pipe to the host.

Table 2-8 Output Terminal Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x09	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x03	VC_OUTPUT_TERMINAL
3	**bTerminalID**	1	0x03	ID of this terminal
4	**wTerminalType**	2	0x0101	TT_STREAMING type. This terminal is a USB streaming terminal.
6	**bAssocTerminal**	1	0x00	No association
7	**bSourceID**	1	0x05	The input pin of this unit is connected to the

				output pin of unit 5.
8	**iTerminal**	1	0x00	Unused

2.3.4.6 Selector Unit Descriptor

This descriptor describes the selector unit that is connected to the processing unit. Either the CCD sensor or composite video connector can be selected as the input.

Table 2-9 Selector Unit Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x08	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE descriptor type
2	**bDescriptorSubtype**	1	0x04	VC_SELECTOR_UNIT descriptor subtype
3	**bUnitID**	1	0x04	ID of this unit
4	**bNrInPins**	1	0x02	Number of input pins
5	**baSourceID(1)**	1	0x01	Input 1 of this unit is connected to unit ID 0x01 – the CAMERA TERMINAL (CMOS sensor).
6	**baSourceID(2)**	1	0x02	Input 2 of this unit is connected to unit ID 0x02 – the composite connector.
7	**iSelector**	1	0x00	Unused

2.3.4.7 Processing Unit Descriptor

This descriptor describes the processing unit that processes the video stream data that is delivered by the selector unit. The only control supported by this implementation is the brightness control.

Table 2-10 Processing Unit Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x0B	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x05	VC_PROCESSING_UNIT
3	**bUnitID**	1	0x05	ID of this unit
4	**bSourceID**	1	0x04	This input pin of this unit is connected to the output pin of unit with ID 0x04.
5	**wMaxMultiplier**	2	0x0000	unused
7	**bControlSize**	1	0x02	Size of the **bmControls** field, in bytes.
8	**bmControls**	2	0x0001	Brightness control supported
10	**iProcessing**	1	0x00	Unused

2.3.4.8 Standard Interrupt Endpoint Descriptor

This descriptor describes the interrupt endpoint used for status returns, in this case to notify the host about button press events.

Table 2-11 Standard Interrupt Endpoint Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x07	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x05	ENDPOINT descriptor
2	**bEndpointAddress**	1	0x81	IN endpoint 1
3	**bmAttributes**	1	0x03	Interrupt transfer type
4	**wMaxPacketSize**	2	0x0008	8-byte status packet
6	**bInterval**	1	0x20	Poll at least every 32ms.

2.3.4.9 Class-specific Interrupt Endpoint Descriptor

This descriptor describes the class-specific information for the interrupt.

Table 2-12 Class-specific Interrupt Endpoint Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x05	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x05	CS_ENDPOINT descriptor
2	**bDescriptorSubType**	1	0x03	EP_INTERRUPT
3	**wMaxTransferSize**	2	0x0008	8-byte status packet

2.3.5 VideoStreaming Interface Descriptor

The VideoStreaming interface has two possible alternate settings, 0 and 1.

2.3.5.1 Zero-bandwidth Alternate Setting 0

Alternate setting 0 is a zero-bandwidth setting, used to relinquish the claimed bandwidth on the bus when the device is not in use. It is the default setting after power-up. The zero bandwidth setting is implied by the omission of an isochronous endpoint in alternate 0. This alternate setting also includes the class-specific format and frame descriptors that describe the video-streaming format capabilities of the device.

2.3.5.1.1 Standard VS Interface Descriptor

Table 2-13 Standard VS Interface Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x09	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x04	INTERFACE descriptor type

2	**bInterfaceNumber**	1	0x01	Index of this interface
3	**bAlternateSetting**	1	0x00	Index of this alternate setting
4	**bNumEndpoints**	1	0x00	0 endpoints – no bandwidth used
5	**bInterfaceClass**	1	0x0E	CC_VIDEO
6	**bInterfaceSubClass**	1	0x02	SC_VIDEOSTREAMING
7	**bInterfaceProtocol**	1	0x00	PC_PROTOCOL_UNDEFINED
8	**iInterface**	1	0x00	Unused

2.3.5.1.2 Class-specific VS Header Descriptor (Input)

This descriptor describes the number of video formats supported by this interface, and the total size of all class-specific descriptors in this interface.

Table 2-14 Class-specific VS Header Descriptor (Input)

Offset	Field	Size	Value	Description
0	**bLength**	1	0x0E	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x01	VS_INPUT_HEADER.
3	**bNumFormats**	1	0x01	One format descriptor follows.
4	**wTotalLength**	2	0x003F	Total size of class-specific VideoStreaming interface descriptors
6	**bEndpointAddress**	1	0x82	Address of the isochronous endpoint used for video data
7	**bmInfo**	1	0x00	No dynamic format change supported
8	**bTerminalLink**	1	0x03	This VideoStreaming interface supplies terminal ID 3 (Output Terminal).
9	**bStillCaptureMethod**	1	0x01	Device supports still image capture method 1.
10	**bTriggerSupport**	1	0x01	Hardware trigger supported for still image capture
11	**bTriggerUsage**	1	0x00	Hardware trigger should initiate a still image capture.
12	**bControlSize**	1	0x01	Size of the **bmaControls** field
13	**bmaControls**	1	0x00	No VideoStreaming specific controls are supported.

2.3.5.1.3 Class-specific VS Format Descriptor

This descriptor describes the video formats supported by the device. Since the device only supports a single video format (MJPG), there is only one format descriptor.

Table 2-15 Class-specific VS Format Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x0B	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x06	VS_FORMAT_MJPEG
3	**bFormatIndex**	1	0x01	First (and only) format descriptor
4	**bNumFrameDescriptors**	1	0x01	One frame descriptor for this format follows.
5	**bmFlags**	1	0x01	Uses fixed size samples.
6	**bDefaultFrameIndex**	1	0x01	Default frame index is 1.
7	**bAspectRatioX**	1	0x00	Non-interlaced stream – not required.
8	**bAspectRatioY**	1	0x00	Non-interlaced stream – not required.
9	**bmInterlaceFlags**	1	0x00	Non-interlaced stream
10	**bCopyProtect**	1	0x00	No restrictions imposed on the duplication of this video stream.

2.3.5.1.4 Class-specific VS Frame Descriptor

This descriptor describes the frame and bandwidth settings supported by the device with the video format described by the preceding format descriptor. Since the device only supports a single frame size (176 x 144), there is only one frame descriptor.

Table 2-16 Class-specific VS Frame Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x26	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x07	VS_FRAME_MJPEG
3	**bFrameIndex**	1	0x01	First (and only) frame descriptor
4	**bmCapabilities**	1	0x03	Still images using capture method 1 are supported at this frame setting. D1: Fixed frame-rate.
5	**wWidth**	2	0x00B0	Width of frame is 176 pixels.
7	**wHeight**	2	0x0090	Height of frame is 144 pixels.
9	**dwMinBitRate**	4	0x000DEC00	Min bit rate in bits/s
13	**dwMaxBitRate**	4	0x000DEC00	Max bit rate in bits/s
17	**dwMaxVideoFrameBufSize**	4	0x00009480	Maximum video or still frame size, in bytes.
21	**dwDefaultFrameInterval**	4	0x000A2C2A	Default frame interval is

				666666ns (15fps).
25	**bFrameIntervalType**	1	0x00	Continuous frame interval
26	**dwMinFrameInterval**	4	0x000A2C2A	Minimum frame interval is 666666ns (15fps)
30	**dwMaxFrameInterval**	4	0x000A2C2A	Maximum frame interval is 666666ns (15fps).
34	**dwFrameIntervalStep**	4	0x00000000	No frame interval step supported.

2.3.5.2 Operational Alternate Setting 1

Alternate setting 1 is the operational setting of the interface. It contains the interface and endpoint descriptors, and specifies a maximum packet size that is able to support the video format being streamed.

2.3.5.2.1 Standard VS Interface Descriptor

Table 2-17 Standard VS Interface Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x09	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x04	INTERFACE descriptor type
2	**bInterfaceNumber**	1	0x01	Index of this interface
3	**bAlternateSetting**	1	0x01	Index of this alternate setting
4	**bNumEndpoints**	1	0x01	0 endpoints – no bandwidth used
5	**bInterfaceClass**	1	0x0E	CC_VIDEO
6	**bInterfaceSubClass**	1	0x02	SC_VIDEOSTREAMING
7	**bInterfaceProtocol**	1	0x00	PC_PROTOCOL_UNDEFINED
8	**iInterface**	1	0x00	Unused

2.3.5.2.2 Standard VS Isochronous Video Data Endpoint Descriptor

Table 2-18 Standard VS Isochronous Video Data Endpoint Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x07	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x05	ENDPOINT
2	**bEndpointAddress**	1	0x82	IN endpoint 2
3	**bmAttributes**	1	0x05	Isochronous transfer type. Asynchronous synchronization type.
4	**wMaxPacketSize**	2	0x01FE	Max packet size of 510 bytes

6	**bInterval**	1	0x01	One frame interval

2.3.6 String Descriptors

In addition to the standard string descriptor zero which contains the list of LANGIDs supported by the device, there are two other string descriptors available. The first string descriptor contains the manufacturer information and the second one contains product information. The following sections present an example of how these descriptors could look like.

2.3.6.1 String Descriptor Zero

Table 2-19 String Descriptor Zero

Offset	Field	Size	Value	Description
0	**bLength**	1	0x18	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x03	STRING descriptor
2	**wLANGID[0]**	2	0x0409	LANGID code zero (US English)

2.3.6.2 Manufacturer String Descriptor (Index 1)

Table 2-20 Manufacturer String Descriptor (Index 1)

Offset	Field	Size	Value	Description
0	**bLength**	1	0x18	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x03	STRING descriptor
2	**bString**	1	0x0054 0x0048 0x0045 0x0020 0x0043 0x004F 0x004D 0x0050 0x0041 0x004E 0x0059	"THE COMPANY"

2.3.6.3 Product String Descriptor (Index 2)

Table 2-21 Product String Descriptor (Index 2)

Offset	Field	Size	Value	Description
0	**bLength**	1	0x0E	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x03	STRING descriptor
2	**bString**	1	0x0043	"Camera"

			0x0061 0x006D 0x0065 0x0072 0x0061	

2.4 Requests

2.4.1 Standard Requests

All standard requests, necessary to operate the device are supported. The next section presents the Set interface request as an example.

2.4.1.1 Set Interface

This request selects the alternate setting on the VideoStreaming interface to control bandwidth allocation.

Table 2-22 Set Interface

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0x01	D7: 0 = Host to device D6..5: 00 = Standard request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x0B	SET_INTERFACE
2	**wValue**	2	0x0000 or 0x0001	0x00 is zero-bandwidth alternate setting. 0x01 is operational alternate setting.
4	**wIndex**	2	0x0001	Interface number of the VideoStreaming interface
6	**wLength**	2	0x0000	No parameter block

2.4.2 Class-specific Requests

The following sections describe the class-specific requests supported by the device in detail. These consist of requests directed to the VideoControl and VideoStreaming interfaces.

2.4.2.1 VideoControl Interface Requests

The class-specific VideoControl interface requests are able to get and set the following controls:

- Selector control in the selector unit
- Brightness control in the processing unit
- Power mode control in the VideoControl interface.

2.4.2.1.1 Set Selector Control Request

This request sets the selector unit control to the desired value.

Table 2-23 Set Selector Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0x21	D7: 0 = Host to device D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x01	SET_CUR
2	**wValue**	2	0x0100	Set the input selector control (01) of this unit.
4	**wIndex**	2	0x0400	Selector Unit ID (04) and VideoControl interface ID (00)
6	**wLength**	2	0x0001	Parameter block length

The one-byte parameter block contains the new bSelector value for the input select control. Since the selector unit has two input pins, the valid range for bSelector is [1,2].

2.4.2.1.2 Get Selector Control Request

This request retrieves the selector unit control parameter.

Table 2-24 Get Selector Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x82 0x83 0x84 0x86	GET_CUR GET_MIN GET_MAX GET_RES GET_INFO
2	**wValue**	2	0x0100	Get the input selector control (01) of this unit.
4	**wIndex**	2	0x0400	Selector Unit ID (04) and VideoControl interface ID (00)
6	**wLength**	2	0x0001	Parameter block length

The actual setting of the Input Select Control is returned in the one-byte parameter block. Since the selector unit has two input pins, the valid range for the returned value is [1,2].

2.4.2.1.3 Set Brightness Control Request

This request sets the brightness control in the processing unit to the desired value.

Table 2-25 Set Brightness Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0x21	D7: 0 = Host to device D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x01	SET_CUR
2	**wValue**	2	0x0200	Set the Brightness control (02) of the unit.
4	**wIndex**	2	0x0500	Processing Unit ID (05) and VideoControl interface ID (00)
6	**wLength**	2	0x0002	Parameter block length

The parameter block contains the new wBrightness value for the brightness control. The valid range for **wBrightness** is detailed in section 4.2.2.3.2 of the *USB Device Class Definition for Video Devices* document.

2.4.2.1.4 Get Brightness Control Request

This request retrieves the brightness control parameter from the processing unit.

Table 2-26 Get Brightness Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x82 0x83 0x84 0x86 0x87	GET_CUR GET_MIN GET_MAX GET_RES GET_INFO GET_DEF
2	**wValue**	2	0x0200	Get the Brightness control (02) of the unit.
4	**wIndex**	2	0x0500	Processing Unit ID (05) and VideoControl interface ID (00)
6	**wLength**	2	0xXXXX	Parameter block length: 0x0001 for GET_INFO request 0x0002 for all other requests

The actual setting of the brightness control is returned in the two-byte parameter block. The valid range for the returned value is detailed in section 4.2.2.3.2 of the *USB Device Class Definition for Video Devices* document.

2.4.2.1.5 Set Power Mode Control Request

This request sets the power mode of the device to the desired value.

Table 2-27 Set Power Mode Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0x21	D7: 0 = Host to device D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x01	SET_CUR.
2	**wValue**	2	0x0100	Set the power control (01) of the interface.
4	**wIndex**	2	0x0000	Only send request to the VideoControl interface (interface ID 00)
6	**wLength**	2	0x0001	Parameter block length

The parameter block contains the new **bDevicePowerMode** value for the power mode control. The valid range for **bDevicePowerMode** is detailed in section 4.2.1.1 of the *USB Device Class Definition for Video Devices* document.

2.4.2.1.6 Get Power Mode Control Request

This request retrieves the device power mode parameter.

Table 2-28 Get Power Mode Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x86	GET_CUR GET_INFO
2	**wValue**	2	0x0100	Get the power control (01) of the interface.
4	**wIndex**	2	0x0000	Only send request to the VideoControl interface (interface ID 00)
6	**wLength**	2	0x0001	Parameter block length

The actual setting of the power mode control is returned in the one-byte parameter block. The valid range for the returned value is detailed in section 4.2.1.1 of the *USB Device Class Definition for Video Devices* document.

2.4.2.1.7 Request Error Code Control

This request retrieves the details of any error conditions pertaining to a Terminal, Unit, interface or endpoint of the video function.

Table 2-29 Request Error Code Control

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x86	GET_CUR GET_INFO
2	**wValue**	2	0x0200	Get the error code control (02) of the interface.
4	**wIndex**	2	0x0000	Only send request to the VideoControl interface (interface ID 00)
6	**wLength**	2	0x0001	Parameter block length

The actual setting of the error code control is returned in the one-byte parameter block. The valid range for the returned value is detailed in section 4.2.1.2 of the *USB Device Class Definition for Video Devices* document.

2.4.2.2 VideoStreaming Requests

The class-specific VideoStreaming interface requests are able to get and set the following controls:

- Video probe control
- Video commit control

2.4.2.2.1 Set Video Probe Control Request

This request sends a set of shadow parameters to the device during negotiation of the active set of parameters for a video stream.

Table 2-30 Set Video Probe Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0x21	D7: 0 = Host to device D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x01	SET_CUR
2	**wValue**	2	0x0100	Probe control selector (01)
4	**wIndex**	2	0x0001	Only send request to the VideoStreaming interface (interface 01).
6	**wLength**	2	0x0022	Parameter block length.

The parameter block contains a new shadow set of stream parameters for the device to use during stream parameter negotiation. The valid data for the shadow set of streaming parameters is detailed in section 4.3.1.1 of the *USB Device Class Definition for Video Devices* document.

2.4.2.2.2 Get Video Probe Control Request

This request retrieves a set of shadow parameters from the device during negotiation of the active set of parameters for a video stream.

Table 2-31 Get Video Probe Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x82 0x83 0x84 0x87 0x85 0x86	GET_CUR GET_MIN GET_MAX GET_RES GET_DEF GET_LEN GET_INFO
2	**wValue**	2	0x0100	Probe control selector (01)
4	**wIndex**	2	0x0001	Only send request to the VideoStreaming interface (interface 01)
6	**wLength**	2	0xXXXX	Parameter block length: 0x0001 for GET_INFO request 0x0022 for all other requests

The parameter block contains a new shadow set of stream parameters for the host to use during stream parameter negotiation. The valid data for the shadow set of streaming parameters is detailed in section 4.3.1.1 of the *USB Device Class Definition for Video Devices* document.

2.4.2.2.3 Set Video Commit Control Request

This request sets a set of working parameters for an active video stream.

Table 2-32 Set Video Commit Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0x21	D7: 0 = Host to device D6..5: 01 = Class request D4..0: 00001 = Recipient is interface

1	**bRequest**	1	0x01	SET_CUR
2	**wValue**	2	0x0200	Commit control selector (02)
4	**wIndex**	2	0x0001	VideoStreaming interface (01)
6	**wLength**	2	0x0022	Parameter block length.

The parameter block contains the stream parameter set for the active video stream. The valid data for the active set of streaming parameters is detailed in section 4.3.1.1 of the *USB Device Class Definition for Video Devices* document.

2.4.2.2.4 Get Video Commit Control Request

This request retrieves a set of working parameters for an active video stream.

Table 2-33 Get Video Commit Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x85 0x86	GET_CUR GET_LEN GET_INFO
2	**wValue**	2	0x0200	Commit control selector (02)
4	**wIndex**	2	0x0001	VideoStreaming interface (01)
6	**wLength**	2	0xXXXX	Parameter block length: 0x0001 for GET_INFO request 0x0022 for all other requests

The parameter block contains the stream parameter set for the active video stream. The valid data for the active set of streaming parameters is detailed in section 4.3.1.1 of the *USB Device Class Definition for Video Devices* document.

3 Video Camera Player Example

3.1 Product Description

The device described in this section is a full-speed video camera player. This particular implementation has two video sources, a CCD sensor and a media transport mechanism on the device that can be switched by using a selector unit on the device. It streams video data through an isochronous pipe to the host in MJPEG format at a single frame size (160x120) at a single frame rate (15 fps), and functions as an asynchronous source, using its internal clock as a reference. It is capable of notifying the host of button press events to trigger still-image capture (using Method 3). This example implementation will assume that we use one Video Interface Collection. The VideoControl interface (interface number 0) and the VideoStreaming interface (interface number 1) are part of this Video Interface Collection.

The following figure represents the internal topology of the video camera player

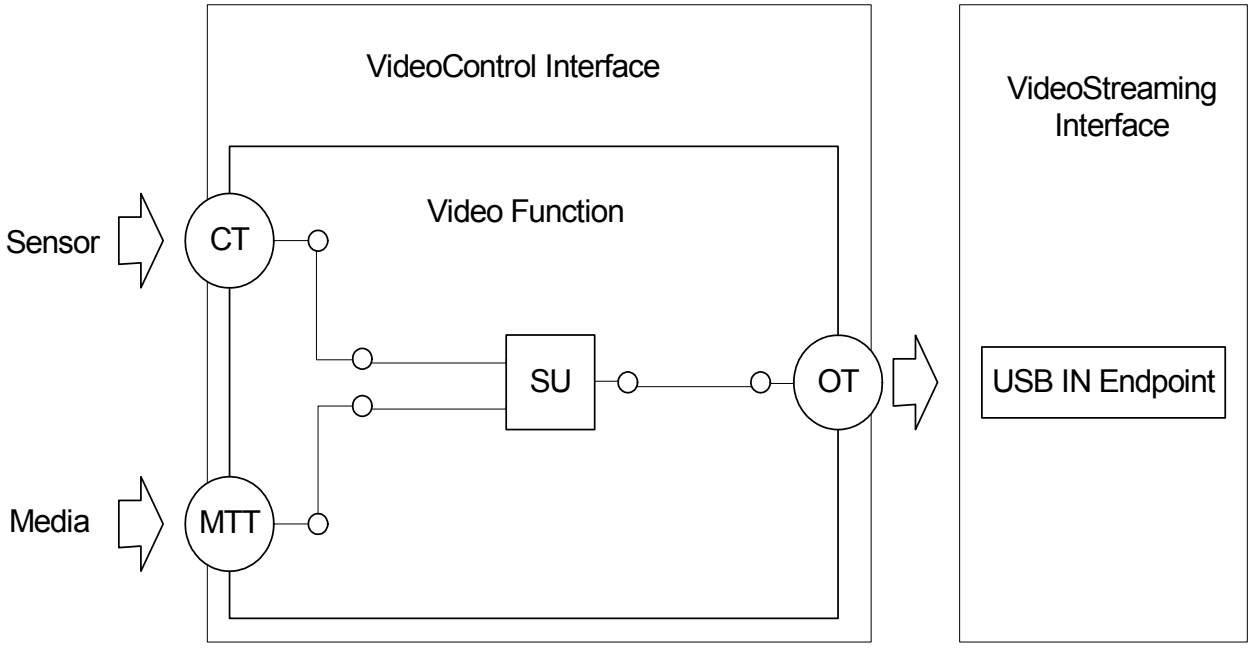

Figure 3-1 USB Video Camera Player Topology

The video function contains two input terminals, one representing the sensor and the other representing the video tape player as a media transport mechanism. The video streams captured by these terminals go through any necessary analogue-to-digital conversion, and are routed into a selector unit. The output is routed to a single output terminal which transmits the video stream to the host via a USB IN endpoint. This endpoint is part of the single VideoStreaming interface that this device contains. The internals of the video function (unit and terminal topology) are presented to the host through the (mandatory) VideoControl interface.

3.2 Descriptor Hierarchy

This figure presents the descriptor hierarchy.

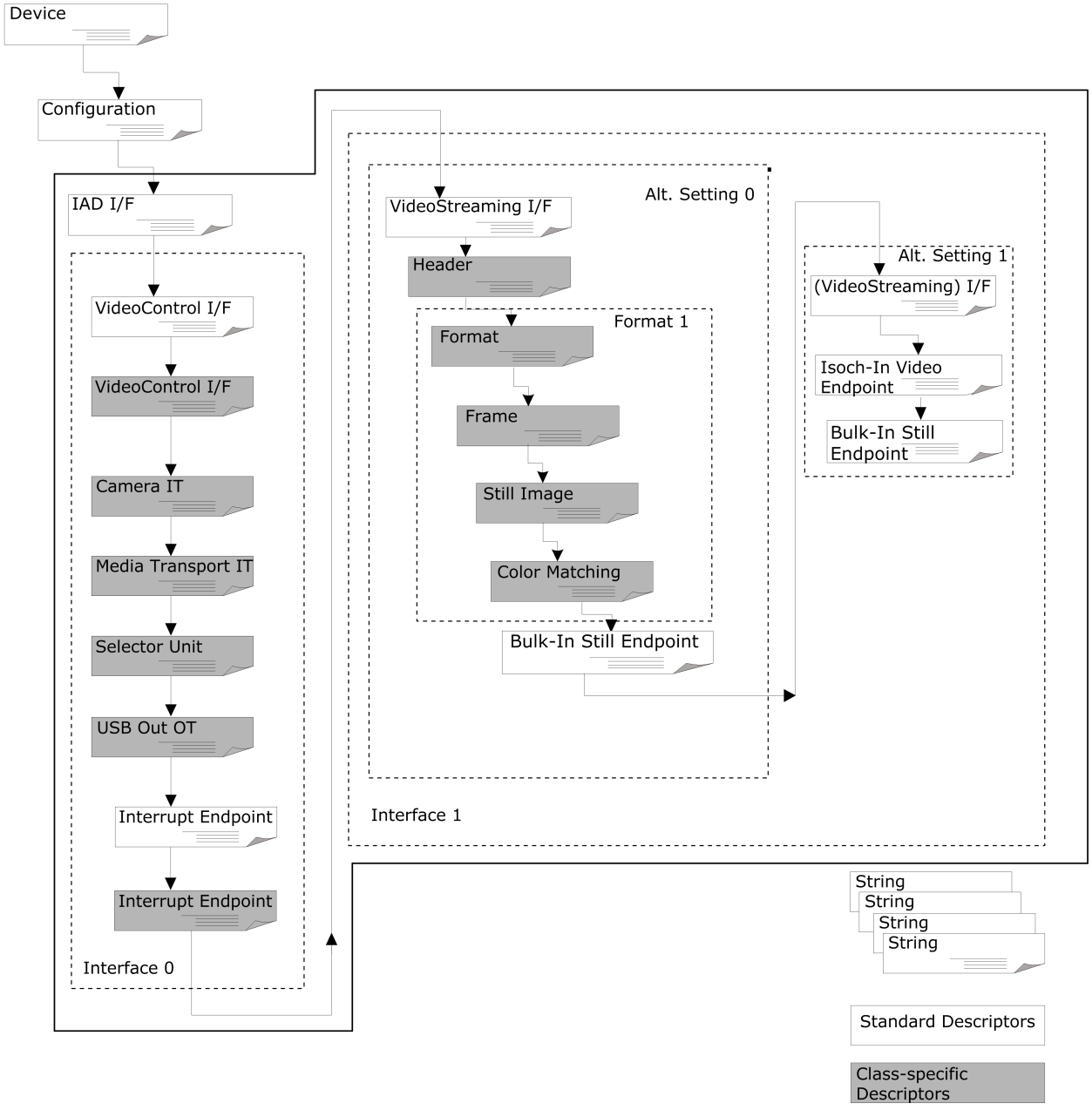

Figure 3-2 USB Video Camera Player Descriptor Hierarchy

3.3 Descriptors

The following sections present all the descriptors that are used to describe the device.

3.3.1 Device Descriptor

Table 3-1 Device Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x12	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x01	DEVICE descriptor
2	**bcdUSB**	2	0x0200	2.00 – current revision of the USB specification
4	**bDeviceClass**	1	0xEF	Miscellaneous Device Class
5	**bDeviceSubClass**	1	0x02	Common Class
6	**bDeviceProtocol**	1	0x01	Interface Association Descriptor
7	**bMaxPacketSize0**	1	0x08	Control endpoint packet size is 8 bytes
8	**idVendor**	2	0xXXXX	Vendor ID
10	**idProduct**	2	0xXXXX	Product ID
12	**bcdDevice**	2	0xXXXX	Device release code
14	**iManufacturer**	1	0x01	Index to string descriptor that contains the string <Your Name> in Unicode
15	**iProduct**	1	0x02	Index to string descriptor that contains the string <Your Product Name> in Unicode
16	**iSerialNumber**	1	0x03	Index String descriptor describing the device's serial number
17	**bNumConfigurations**	1	0x01	One configuration

3.3.2 Configuration Descriptor

Table 3-2 Configuration Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x09	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x02	CONFIGURATION descriptor
2	**wTotalLength**	2	0x00D9	Length of the total configuration block, including this descriptor, in bytes.
4	**bNumInterfaces**	1	0x02	Number of interfaces
5	**bConfigurationValue**	1	0x01	ID of this configuration
6	**iConfiguration**	1	0x00	Unused
7	**bmAttributes**	1	0xC0	Self power
8	**bMaxPower**	1	0x00	Unused

3.3.3 Interface Association Descriptor

This device uses an Interface Association Descriptor to describe its Video Interface Collection.

Table 3-3 Standard Video Interface Collection IAD

Offset	Field	Size	Value	Description
0	**bLength**	1	0x08	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x0B	INTERFACE ASSOCIATION Descriptor.
2	**bFirstInterface**	1	0x00	Interface number of the VideoControl interface that is associated with this function.
3	**bInterfaceCount**	1	0x02	Number of contiguous Video interfaces that are associated with this function.
4	**bFunctionClass**	1	0x0E	CC_VIDEO
5	**bFunctionSubClass**	1	0x03	SC_VIDEO_INTERFACE_COLLECTION
6	**bFunctionProtocol**	1	0x00	Not used. Must be set to PC_PROTOCOL_UNDEFINED.
7	**iFunction**	1	0x04	Index of string descriptor. Must match the iInterface field of the Standard VC Interface Descriptor.

3.3.4 Video Control Interface Descriptor

The VideoControl interface describes the device structure (video function topology) and is used to manipulate the video controls.

3.3.4.1 Standard VC Interface Descriptor

Table 3-4 Standard VC Interface Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x09	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x04	INTERFACE descriptor type
2	**bInterfaceNumber**	1	0x00	Index of this interface
3	**bAlternateSetting**	1	0x00	Index of this setting
4	**bNumEndpoints**	1	0x01	1 endpoint (interrupt endpoint)
5	**bInterfaceClass**	1	0x0E	CC_VIDEO
6	**bInterfaceSubClass**	1	0x01	SC_VIDEOCONTROL
7	**bInterfaceProtocol**	1	0x00	Not used. Must be set to PC_PROTOCOL_UNDEFINED.
8	**iInterface**	1	0x04	Index of String descriptor. Must match the iFuntion field of the Standard Video Interface Collection IAD.

3.3.4.2 Class-specific VC Interface Descriptor

Table 3-5 Class-specific VC Interface Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x0D	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubType**	1	0x01	VC_HEADER subtype
3	**bcdUVC**	2	0x0110	Revision of class specification that this device is based upon. For this example, the device complies with Video Class specification version 1.1
5	**wTotalLength**	2	0x0040	Total size of class-specific descriptors
7	**dwClockFrequency**	4	0xXXXXXXXX	This field has been deprecated.This device will provide timestamps and a device clock reference based on a XXMHz clock.
11	**bInCollection**	1	0x01	Number of streaming interfaces
12	**baInterfaceNr(1)**	1	0x01	VideoStreaming interface 4 belongs to this VideoControl interface.

3.3.4.3 Input Terminal Descriptor (Camera)

This descriptor describes the input terminal that represents the CCD sensor (and associated A/D converter). The resulting digital video stream leaves the input terminal through the single output pin.

Table 3-6 Input Terminal Descriptor (Camera)

Offset	Field	Size	Value	Description
0	**bLength**	1	0x12	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x02	VC_INPUT_TERMINAL subtype
3	**bTerminalID**	1	0x02	ID of this input terminal
4	**wTerminalType**	2	0x0201	ITT_CAMERA type. This terminal is a camera terminal representing the CCD sensor.
6	**bAssocTerminal**	1	0x00	No association
7	**iTerminal**	1	0x00	Unused
8	**wObjectiveFocalLengthMin**	2	0xXXXX	Minimum focal length (objective)
10	**wObjectiveFocalLengthMax**	2	0xXXXX	Maximum focal length (objective)
12	**wOcularFocalLength**	2	0xXXXX	Focal Length (ocular)
14	**bControlSize**	1	0x03	The size of the bmControls is 3 bytes.

15	**bmControls**	3	0x000200	Supported controls.

3.3.4.4 Input Terminal Descriptor (Media Transport)

This descriptor describes the input terminal that represents the media transport mechanism for the video tape player. The resulting digital video stream leaves the input terminal through the single output pin. This terminal supports, for example, sequential media.

Table 3-7 Input Terminal Descriptor (Media Transport)

Offset	Field	Size	Value	Description
0	**bLength**	1	0x10	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x02	INPUT_TERMINAL subtype
3	**bTerminalID**	1	0x03	ID of this input terminal
4	**wTerminalType**	2	0x0202	ITT_MEDIA_TRANSPORT_INPUT type. This terminal is the media transport mechanism.
6	**bAssocTerminal**	1	0x00	No association
7	**iTerminal**	1	0x00	Unused
8	**bControlSize**	1	0x01	Size of **bmControls**, in bytes.
9	**bmControls**	1	0x0D	Supported control requests
10	**bTransportModeSize**	1	0x05	Size of **bmTransportModes**, in bytes.
11	**bmTransportModes**	5	0x007FFFFFAF	Supported control parameters of the Transport control

3.3.4.5 Selector Unit Descriptor

This descriptor describes the selector unit that is connected to the processing unit. Either the CCD sensor or media transport mechanism can be selected as the input.

Table 3-8 Selector Unit Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x08	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE descriptor type
2	**bDescriptorSubtype**	1	0x04	VC_SELECTOR_UNIT descriptor subtype
3	**bUnitID**	1	0x01	ID of this unit
4	**bNrInPins**	1	0x02	Number of input pins
5	**baSourceID(1)**	1	0x02	Input 1 of this unit is connected to unit ID 0x02 – the Camera terminal.
6	**baSourceID(2)**	1	0x03	Input 2 of this unit is connected to unit ID 0x03 – the media transport terminal.
7	**iSelector**	1	0x00	Unused

3.3.4.6 Output Terminal Descriptor

This descriptor describes the output terminal that represents the USB pipe to the host.

Table 3-9 Output Terminal Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x09	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x03	VC_OUTPUT_TERMINAL
3	**bTerminalID**	1	0x04	ID of this terminal
4	**wTerminalType**	2	0x0101	TT_STREAMING type. This terminal is a USB streaming terminal.
6	**bAssocTerminal**	1	0x00	No association
7	**bSourceID**	1	0x01	The input pin of this unit is connected to the output pin of unit 1 – Selector unit.
8	**iTerminal**	1	0x00	Unused

3.3.4.7 Standard Interrupt Endpoint Descriptor

Table 3-10 Standard Interrupt Endpoint Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x07	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x05	ENDPOINT descriptor
2	**bEndpointAddress**	1	0x83	IN endpoint 3
3	**bmAttributes**	1	0x03	Interrupt transfer type
4	**wMaxPacketSize**	2	0x0008	8-byte status packet
6	**bInterval**	1	0x0A	Poll at least every 10ms.

3.3.4.8 Class-specific Interrupt Endpoint Descriptor

This descriptor describes the class-specific information for the interrupt.

Table 3-11 Class-specific Interrupt Endpoint Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x05	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x25	CS_ENDPOINT descriptor
2	**bDescriptorSubType**	1	0x03	EP_INTERRUPT
3	**wMaxTransferSize**	2	0x0020	32-byte status packet

3.3.5 Video Streaming Interface Descriptor

The VideoStreaming interface has two possible alternate settings, 0 and 1.

3.3.5.1 Zero-bandwidth Alternate Setting 0

Alternate setting 0 is the default setting after power-up. It is also includes the class-specific format and frame descriptors that describe the video-streaming format capabilities of the device.

3.3.5.1.1 Standard VS Interface Descriptor

Table 3-12 Standard VS Interface Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x09	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x04	INTERFACE descriptor type
2	**bInterfaceNumber**	1	0x01	Index of this interface
3	**bAlternateSetting**	1	0x00	Index of this alternate setting
4	**bNumEndpoints**	1	0x01	1 endpoints
5	**bInterfaceClass**	1	0x0E	CC_VIDEO
6	**bInterfaceSubClass**	1	0x02	SC_VIDEOSTREAMING
7	**bInterfaceProtocol**	1	0x00	PC_PROTOCOL_UNDEFINED
8	**iInterface**	1	0x04	Index of String descriptor

3.3.5.1.2 Class-specific VS Header Descriptor (Input)

This descriptor describes the number of video formats supported by this interface, and the total size of all class-specific descriptors in this interface.

Table 3-13 Class-specific VS Header Descriptor (Input)

Offset	Field	Size	Value	Description
0	**bLength**	1	0x0E	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x01	VS_INPUT_HEADER
3	**bNumFormats**	1	0x01	One format descriptor follows.
4	**wTotalLength**	2	0x004C	Total size of class-specific VideoStreaming interface descriptors
6	**bEndpointAddress**	1	0x85	Address of the isochronous endpoint used for video data
7	**bmInfo**	1	0x00	No dynamic format change supported.
8	**bTerminalLink**	1	0x04	This VideoStreaming interface

				supplies terminal ID 4 (Output Terminal).
9	**bStillCaptureMethod**	1	0x03	Device supports still image capture method 3.
10	**bTriggerSupport**	1	0x00	Hardware trigger is not supported for still image capture.
11	**bTriggerUsage**	1	0x00	Hardware trigger should initiate a still image capture.
12	**bControlSize**	1	0x01	Size of the **bmaControls** field.
13	**bmaControls**	1	0x00	No VideoStreaming specific controls are supported.

3.3.5.1.3 Class-specific VS Format Descriptor

This descriptor describes the video formats supported by the device. Since the device only supports a single video format (MJPG), there is only one format descriptor.

Table 3-14 Class-specific VS Format Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x0B	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x06	VS_FORMAT_MJPEG
3	**bFormatIndex**	1	0x01	First (and only) format descriptor
4	**bNumFrameDescriptors**	1	0x01	One frame descriptor for this format follows
5	**bmFlags**	1	0x01	Uses fixed size samples.
6	**bDefaultFrameIndex**	1	0x01	Default frame index is 1.
7	**bAspectRatioX**	1	0x00	Non-interlaced stream – not required.
8	**bAspectRatioY**	1	0x00	Non-interlaced stream – not required.
9	**bmInterlaceFlags**	1	0x02	Progressive
10	**bCopyProtect**	1	0x00	No restrictions imposed on the duplication of this video stream.

3.3.5.1.4 Class-specific VS Frame Descriptor

This descriptor describes the frame and bandwidth settings supported by the device with the video format described by the preceding format descriptor. Since the device only supports a single frame size (160 x 120), there is only one frame descriptor.

Table 3-15 Class-specific VS Frame Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x1E	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x07	VS_FRAME_MJPEG
3	**bFrameIndex**	1	0x01	First (and only) frame descriptor
4	**bmCapabilities**	1	0x02	D1: Fixed frame-rate.
5	**wWidth**	2	0x00A0	Width of frame is 160 pixels.
7	**wHeight**	2	0x0078	Height of frame is 120 pixels.
9	**dwMinBitRate**	4	0x00046500	Min bit rate in bits/s
13	**dwMaxBitRate**	4	0x000FA000	Max bit rate in bits/s
17	**dwMaxVideoFrameBufSize**	4	0x00000800	Maximum video or still frame size, in bytes.
21	**dwDefaultFrameInterval**	4	0x000A2C2A	Default frame interval is 666666ns (15fps).
25	**bFrameIntervalType**	1	0x01	Discrete frame interval
26	**dwFrameInterval(1)**	4	0x000A2C2A	Minimum frame interval is 666666ns (15fps).

3.3.5.1.5 Class-specific Still Image Frame Descriptor

This descriptor describes the still image frame supported by the device. Since the device only supports a two frame sizes, there two pair of width and height in this frame descriptor.

Table 3-16 Class-specific Still Image Frame Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x0F	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x03	VS_STILL_FRAME
3	**bEndpointAddress**	1	0x86	IN Endpoint, Endpoint number 6
4	**bNumImageSizePatterns**	1	0x02	Number of Image Size patterns of this format: 2
5	**wWidth**	2	0x0320	Width of frame is 800 pixels
7	**wHeight**	2	0x0258	Height of frame is 600 pixels
9	**wWidth**	2	0xXXXX	Width of frame 2
11	**wHeight**	2	0xXXXX	Height of frame 2

13	**bNumCompressionPtn**	1	0x01	Number of Compression pattern of this format: 1
14	**bCompression**	1	0x64	Compression of the still image in pattern 1: 100

3.3.5.1.6 Class-specific Color Matching Descriptor

Table 3-17 Class-specific Color Matching Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x06	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x0D	VS_COLORFORMAT
3	**bColorPrimaries**	1	0x00	Unspecified
4	**bTransferCharacteristics**	1	0x00	Unspecified
5	**bMatrixCoefficients**	1	0x00	Unspecified

3.3.5.1.7 Standard Bulk Endpoint Descriptor

Table 3-18 Standard Bulk Endpoint Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x07	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x05	ENDPOINT descriptor type
2	**bEndpointAddress**	1	0x86	IN, Endpoint number 6
3	**bmAttributes**	1	0x02	Bulk, No synchronization, Data endpoint
4	**wMaxPacketSize**	1	0x0040	Maximum packet size
6	**bInterval**	1	0x00	Never NAKs

3.3.5.2 Operational Alternate Setting 1

3.3.5.2.1 Standard VS Interface Descriptor

Table 3-19 Standard VS Interface Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x09	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x04	INTERFACE descriptor type
2	**bInterfaceNumber**	1	0x01	Index of this interface
3	**bAlternateSetting**	1	0x01	Index of this alternate setting
4	**bNumEndpoints**	1	0x02	2 endpoints

5	**bInterfaceClass**	1	0x0E	CC_VIDEO
6	**bInterfaceSubClass**	1	0x02	SC_VIDEOSTREAMING
7	**bInterfaceProtocol**	1	0x00	PC_PROTOCOL_UNDEFINED
8	**iInterface**	1	0x04	Unused

3.3.5.2.2 Standard VS Isochronous Video Data Endpoint Descriptor

Table 3-20 Standard VS Isochronous Video Data Endpoint Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x07	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x05	ENDPOINT
2	**bEndpointAddress**	1	0x85	IN endpoint 5
3	**bmAttributes**	1	0x05	Isochronous transfer type. Asynchronous synchronization type.
4	**wMaxPacketSize**	2	0x0080	Max packet size of 128 bytes
6	**bInterval**	1	0x01	One frame interval

3.3.5.2.3 Standard Bulk Endpoint Descriptor

Table 3-21 Standard Bulk Endpoint Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x07	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x05	Bulk endpoint descriptor type
2	**bEndpointAddress**	1	0x86	IN, Endpoint number 6
3	**bmAttributes**	1	0x02	Bulk, No synchronization, Data endpoint
4	**wMaxPacketSize**	1	0x0040	Maximum packet size
6	**bInterval**	1	0x00	Never NAKs

3.3.6 String Descriptors

In addition to the standard string descriptor zero which contains the list of LANGIDs supported by the device, there are four other string descriptors available.

3.3.6.1 String Descriptor Zero

Table 3-22 String Descriptor Zero

Offset	Field	Size	Value	Description
0	**bLength**	1	0xXX	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x03	STRING descriptor
2	**wLANGID[0]**	2	0x0409	LANGID code zero (US English)

3.3.6.2 Manufacturer String Descriptor (Index 1)

Table 3-23 Manufacturer String Descriptor (Index 1)

Offset	Field	Size	Value	Description
0	**bLength**	1	0xXX	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x03	STRING descriptor
2	**bString**	n	0xXXXX 0xXXXX	STRING Values

3.3.6.3 Product String Descriptor (Index 2)

Table 3-24 Product String Descriptor (Index 2)

Offset	Field	Size	Value	Description
0	**bLength**	1	0xXX	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x03	STRING descriptor
2	**bString**	n	0xXXXX 0xXXXX	STRING Value

3.3.6.4 Serial Number String Descriptor (Index 3)

Table 3-25 Serial Number String Descriptor (Index 3)

Offset	Field	Size	Value	Description
0	**bLength**	1	0xXX	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x03	STRING descriptor
2	**bString**	n	0xXXXX 0xXXXX	STRING Value

3.3.6.5 Product String Descriptor (Index 4)

Table 3-26 Product String Descriptor (Index 4)

Offset	Field	Size	Value	Description
0	**bLength**	1	0xXX	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	0x03	STRING descriptor
2	**bString**	n	0xXXXX 0xXXXX	STRING Value

3.4 Requests

3.4.1 Standard Requests

All standard requests, necessary to operate the device are supported. The next section presents the Set interface request as an example.

3.4.1.1 Set Interface

This request selects the alternate setting on the VideoStreaming interface to control bandwidth allocation.

Table 3-27 Set Interface

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0x01	D7: 0 = Host to device D6..5: 00 = Standard request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x0B	SET_INTERFACE
2	**wValue**	2	0x0000 or 0x0001	0x00 is zero-bandwidth alternate setting. 0x01 is operational alternate setting.
4	**wIndex**	2	0x0001	Interface number of the VideoStreaming interface
6	**wLength**	2	0x0000	No parameter block

3.4.2 Class-specific Requests

The following sections describe the class-specific requests supported by the device in detail. These consist of requests directed to the VideoControl and VideoStreaming interfaces.

3.4.2.1 VideoControl Interface Requests

The class-specific VideoControl interface requests are able to get and set the following controls:

- Selector control in the selector unit
- Zoom(Absolute) control in the Camera Terminal
- Transport Control in the Media Transport Terminal
- Media Information Control in the Media Transport Terminal

- Time Code Information Control in the Media Transport Terminal
- Power mode control in the VideoControl interface.

3.4.2.1.1 Set Selector Control Request

This request sets the selector unit control to the desired value.

Table 3-28 Set Selector Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0x21	D7: 0 = Host to device D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x01	SET_CUR
2	**wValue**	2	0x0100	Set the input selector control (01) of this unit.
4	**wIndex**	2	0x0100	Selector Unit ID (01) and VideoControl interface ID (00)
6	**wLength**	2	0x0001	Parameter block length

The one-byte parameter block contains the new bSelector value for the input select control. Since the selector unit has two input pins, the valid range for bSelector is [1,2].

3.4.2.1.2 Get Selector Control Request

This request retrieves the selector unit control parameter.

Table 3-29 Get Selector Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x82 0x83 0x84 0x86	GET_CUR GET_MIN GET_MAX GET_RES GET_INFO
2	**wValue**	2	0x0100	Get the input selector control (01) of this unit.
4	**wIndex**	2	0x0100	Selector Unit ID (01) and VideoControl interface ID (00)
6	**wLength**	2	0x0001	Parameter block length

The actual setting of the Input Select Control is returned in the one-byte parameter block. Since the selector unit has two input pins, the valid range for the returned value is [1,2].

3.4.2.1.3 Set Zoom (Absolute) Control Request

This request sets the Zoom (Absolute) control in the Camera Terminal to the desired value.

Table 3-30 Set Zoom (Absolute) Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0x21	D7: 0 = Host to device D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x01	SET_CUR
2	**wValue**	2	0x0B00	Set the Zoom (Absolute) control (0B) of the Terminal.
4	**wIndex**	2	0x0200	Camera Terminal ID (02) and VideoControl interface ID (00)
6	**wLength**	2	0x0002	Parameter block length

The parameter block contains the new wObjectiveFocalLength value for the Zoom (Absolute) control. The valid range for **wObjectiveFocalLength** is detailed in section 4.2.2.1.11 of the *USB Device Class Definition for Video Devices* document.

3.4.2.1.4 Get Zoom (Absolute) Control Request

This request retrieves the Zoom (Absolute) control parameter from the Camera Terminal.

Table 3-31 Get Zoom (Absolute) Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x82 0x83 0x84 0x86 0x87	GET_CUR GET_MIN GET_MAX GET_RES GET_INFO GET_DEF
2	**wValue**	2	0x0B00	Get the Zoom (Absolute) control (0B) of the terminal.
4	**wIndex**	2	0x0200	Camera Terminal ID (02) and VideoControl interface ID (00)

6	**wLength**	2	0xXXXX	Parameter block length: 0x0001 for GET_INFO request 0x0002 for all other requests

The actual setting of the wObjectiveFocalLength attribute of the control is returned in the two-byte parameter block. The valid range for the returned value is detailed in section 4.2.2.1.11 of the *USB Device Class Definition for Video Devices* document.

3.4.2.1.5 Set Transport Control Request

This request sets the Transport control in the Media Transport Terminal to the desired value.

Table 3-32 Set Transport Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0x21	D7: 0 = Host to device D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x01	SET_CUR
2	**wValue**	2	0x0100	Set the Transport control (01) of the Media Transport Terminal.
4	**wIndex**	2	0x0300	Media Transport Terminal ID (03) and VideoControl interface ID (00)
6	**wLength**	2	0x0001	Parameter block length

The parameter block contains the new bTransportStatebTransportMode value for the Transport control. The valid range for **bTransportStatebTransportMode** is detailed in section 4.1.3.1 of the *USB Device Class Definition for Video Media Transport Terminal*.

3.4.2.1.6 Get Transport Control Request

This request retrieves the Transport control parameter from the Media Transport Terminal.

Table 3-33 Get Transport Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x86	GET_CUR GET_INFO
2	**wValue**	2	0x0100	Get the Transport control (01) of the terminal.
4	**wIndex**	2	0x0300	Media Transport Terminal ID (03) and VideoControl interface ID (00)

6	**wLength**	2	0x0001	Parameter block length.

The actual setting of the bTransportStatebTransportMode attribute of the control is returned in the one-byte parameter block. The valid range for the returned value is detailed in 4.1.3.1 of the *USB Device Class Definition for Video Media Transport Terminal*.

3.4.2.1.7 Set Media Information Control Request

There is no set Request for the Media Information Control.

The Media Information Control is detailed in section 4.1.3.3 of the *USB Device Class Definition for Video Media Transport Terminal*.

3.4.2.1.8 Get Media Information Control Request

This request retrieves the Media Information control parameter from the Media Transport Terminal.

Table 3-34 Get Transport Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x86	GET_CUR GET_INFO
2	**wValue**	2	0x0300	Get the Media Information control (03) of the terminal.
4	**wIndex**	2	0x0300	Media Transport Terminal ID (03) and VideoControl interface ID (00)
6	**wLength**	2	0xXXXX	Parameter block length: 0x0001 for GET_INFO request 0x0002 for all other requests

The actual settings of the bmMediaType and bmWriteProtect attributes of the control are returned in the two-byte parameter block. The valid range for the returned value is detailed in 4.1.3.3 of the *USB Device Class Definition for Video Media Transport Terminal*.

3.4.2.1.9 Set Time Code Information Control Request

This request sets the Time Code Information control in the Media Transport Terminal to the desired value.

Table 3-35 Set Time Code Information Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0x21	D7: 0 = Host to device D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x01	SET_CUR
2	**wValue**	2	0x0400	Set the Time Code Information control (04) of the Media Transport Terminal.
4	**wIndex**	2	0x0300	Media Transport Terminal ID (03) and VideoControl interface ID (00)
6	**wLength**	2	0x0004	Parameter block length

The parameter block contains the new bcdFrame, bcdSecond, bcdMinute and bcdHour values for the Time Code Information control. The valid ranges for **bcdFrame, bcdSecond, bcdMinute and bcdHour** are detailed in section 4.1.3.4 of the *USB Device Class Definition for Video Media Transport Terminal*.

3.4.2.1.10 Get Time Code Information Control Request

This request retrieves the Time Code Information control parameter from the Media Transport Terminal.

Table 3-36 Get Transport Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x86	GET_CUR GET_INFO
2	**wValue**	2	0x0400	Get the Time Code Information control (04) of the terminal.
4	**wIndex**	2	0x0300	Media Transport Terminal ID (03) and VideoControl interface ID (00)
6	**wLength**	2	0xXXXX	Parameter block length: 0x0001 for GET_INFO request 0x0004 for all other requests

The actual setting of the bcdFrame, bcdSecond, bcdMinute and bcdHour attributes of the control are returned in the four-byte parameter block. The valid range for the returned value is detailed in 4.1.3.4 of the *USB Device Class Definition for Video Media Transport Terminal*.

3.4.2.1.11 Set Power Mode Control Request

This request sets the power mode of the device to the desired value.

Table 3-37 Set Power Mode Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0x21	D7: 0 = Host to device D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x01	SET_CUR.
2	**wValue**	2	0x0100	Set the power control (01) of the interface.
4	**wIndex**	2	0x0000	Only send request to the VideoControl interface (interface ID 00)
6	**wLength**	2	0x0001	Parameter block length

The parameter block contains the new **bDevicePowerMode** value for the power mode control. The valid range for **bDevicePowerMode** is detailed in section 4.2.1.1 of the *USB Device Class Definition for Video Devices* document.

3.4.2.1.12 Get Power Mode Control Request

This request retrieves the device power mode parameter.

Table 3-38 Get Power Mode Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x86	GET_CUR GET_INFO
2	**wValue**	2	0x0100	Get the power control (01) of the interface.
4	**wIndex**	2	0x0000	Only send request to the VideoControl interface (interface ID 00)
6	**wLength**	2	0x0001	Parameter block length

The actual setting of the power mode control is returned in the one-byte parameter block. The valid range for the returned value is detailed in section 4.2.1.1 of the *USB Device Class Definition for Video Devices* document.

3.4.2.1.13 Request Error Code Control

This request retrieves the details of any error conditions pertaining to a Terminal, Unit, interface or endpoint of the video function.

Table 3-39 Request Error Code Control

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x86	GET_CUR GET_INFO
2	**wValue**	2	0x0200	Get the error code control (02) of the interface.
4	**wIndex**	2	0x0000	Only send request to the VideoControl interface (interface ID 00)
6	**wLength**	2	0x0001	Parameter block length

The actual setting of the error code control is returned in the one-byte parameter block. The valid range for the returned value is detailed in section 4.2.1.2 of the *USB Device Class Definition for Video Devices* document.

3.4.2.2 VideoStreaming Requests

The class-specific VideoStreaming interface requests are able to get and set the following controls:

- Video probe control
- Video commit control

3.4.2.2.1 Set Video Probe Control Request

This request sends a set of shadow parameters to the device during negotiation of the active set of parameters for a video stream.

Table 3-40 Set Video Probe Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0x21	D7: 0 = Host to device D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x01	SET_CUR
2	**wValue**	2	0x0100	Probe control selector (01)
4	**wIndex**	2	0x0001	Only send request to the VideoStreaming interface (interface 01).
6	**wLength**	2	0x0022	Parameter block length

The parameter block contains a new shadow set of stream parameters for the device to use during stream parameter negotiation. The valid data for the shadow set of streaming parameters is detailed in section 4.3.1.1 of the *USB Device Class Definition for Video Devices* document.

3.4.2.2.2 Get Video Probe Control Request

This request retrieves a set of shadow parameters from the device during negotiation of the active set of parameters for a video stream.

Table 3-41 Get Video Probe Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x82 0x83 0x84 0x87 0x85 0x86	GET_CUR GET_MIN GET_MAX GET_RES GET_DEF GET_LEN GET_INFO
2	**wValue**	2	0x0100	Probe control selector (01)
4	**wIndex**	2	0x0001	Only send request to the VideoStreaming interface (interface 01)
6	**wLength**	2	0xXXXX	Parameter block length: 0x0001 for GET_INFO request 0x0022 for all other requests

The parameter block contains a new shadow set of stream parameters for the host to use during stream parameter negotiation. The valid data for the shadow set of streaming parameters is detailed in section 4.3.1.1 of the *USB Device Class Definition for Video Devices* document.

3.4.2.2.3 Set Video Commit Control Request

This request sets a set of working parameters for an active video stream.

Table 3-42 Set Video Commit Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0x21	D7: 0 = Host to device D6..5: 01 = Class request D4..0: 00001 = Recipient is interface

1	**bRequest**	1	0x01	SET_CUR
2	**wValue**	2	0x0200	Commit control selector (02)
4	**wIndex**	2	0x0001	VideoStreaming interface (01)
6	**wLength**	2	0x0022	Parameter block length

The parameter block contains the stream parameter set for the active video stream. The valid data for the active set of streaming parameters is detailed in section 4.3.1.1 of the *USB Device Class Definition for Video Devices* document.

3.4.2.2.4 Get Video Commit Control Request

This request retrieves a set of working parameters for an active video stream.

Table 3-43 Get Video Commit Control Request

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	0xA1	D7: 1 = Device To host D6..5: 01 = Class request D4..0: 00001 = Recipient is interface
1	**bRequest**	1	0x81 0x85 0x86	GET_CUR GET_LEN GET_INFO
2	**wValue**	2	0x0200	Commit control selector (02)
4	**wIndex**	2	0x0001	VideoStreaming interface (01)
6	**wLength**	2	0xXXXX	Parameter block length: 0x0001 for GET_INFO request 0x0022 for all other requests

The parameter block contains the stream parameter set for the active video stream. The valid data for the active set of streaming parameters is detailed in section 4.3.1.1 of the *USB Device Class Definition for Video Devices* document.