Universal Serial Bus
Device Class Definition
for
Video Devices:
Stream Based Payload

Revision 1.1

June 1, 2005

Contributors

Abdul R. Ismail	Intel Corp.
Akihiro Tanabe	Canon Inc.
Allison Hicks	Texas Instruments
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
1.1	June 1st, 2005	Initial release

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Purpose

This document defines a general extensibility mechanism by which vendors can support Stream Based payload formats not defined by the *USB Device Class Definition for Video Devices* standard payload format specifications. A Stream Based payload format is any format where the video data is transferred as a Byte- or Packet-oriented stream. If the packetization scheme allows for multiplexed streams, then multiple video, as well as non-video, streams may be transported over a single endpoint.

1.2 Scope

The payload format and associated header information is fully specified by this document. This includes:

- Payload Header
- Payload Data

1.3 Related Documents

USB Specification Revision 2.0, April 27, 2000, www.usb.org
USB Device Class Definition for Video Devices, www.usb.org
ISO/IEC International Standard 11172; "Coding of moving pictures and associated audio for digital storage media up to about 1,5Mbits/s", November 1993.
ISO/IEC International Standard 13818; "Generic coding of moving pictures and associated audio information", November 1994.
Pr ETS 300 468, Digital broadcasting systems for television, sound and data services – Specification for service information (SI) in digital video broadcasting (DVB) systems

2 Video Class-Specific Information

2.1 Payload Header

The following is a description of USB VC Header definition for Stream Based format.

Table 2-1 Header Definition for Stream Based Format

HLE	Header Length							
BFH [0]	EOH	ERR	STI	RES	SCR	PTS	EOF	FID

HLE

Size: 1 byte, Value: unit number in bytes

The header length field specifies the length of the header, in bytes. This field shall be set to 2.

BFH[0]

Size: 1 byte, Value: bit field

FID: Frame ID

When the D0 bit of the **bmFramingInfo** field of the Video Probe and Commit Control is set, this field is used to indicate codec-specific segments, such that the value will remain constant throughout a codec-specific segment, then toggle at the beginning of the next segment. Otherwise, this field is ignored and shall be set to zero. For detailed information, see section 4.3.1.1 “Video Probe and Commit Controls" in the *USB Device Class Definition for Video Devices* specification.

EOF: End of Frame

When the D1 bit of the **bmFramingInfo** field of the Video Probe and Commit Control is set, this field is used to indicate the end of a codec-specific segment. Otherwise, this field is ignored and shall be set to zero. For detailed information, see section 4.3.1.1 “Video Probe and Commit Controls" in the *USB Device Class Definition for Video Devices* specification.

PTS: Presentation Time Stamp

This bit shall be set to zero.

SCR: Source Clock Reference

This bit shall be set to zero.

RES: Reserved.

This bit shall be set to zero.

STI: Still Image

This bit shall be set to zero.

ERR: Error Bit
This bit, when set, indicates an error in the streaming device.

EOH: End of Header
This bit shall be set to 1.

2.2 Payload Data

Payload Data is either *Packet-Oriented* or *Byte-Oriented*. When **dwPacketLength** (see Table 3-1) is non-zero, the stream is Packet-Oriented and shall be interpreted as a sequence of format-specific packets. When **dwPacketLength** is zero, the stream is Byte-Oriented and shall be interpreted as a sequence of bytes, without regard to format-specific boundaries (if any) within the stream. A stream consisting of variable length packets is considered to be a Byte-Oriented stream. Out-of-band boundary indicators can be applied to both Byte-Oriented and Packet-Oriented streams via the FID and EOF bits of the Payload Header (see section 2.1, "Payload Header" of this specification for additional details).

2.2.1 Packet-Oriented Payload Data

Packet-Oriented streams have additional constraints, governed by the relationship between the value of **dwPacketLength** and the size of the Payload Data portion of a Payload Transfer.

When **dwPacketLength** is greater than the Payload Data size, the start of a format-specific packet must coincide with a new payload transfer (i.e. the first byte of a format-specific packet shall be the first byte after the Payload Header in a Payload Transfer). The final byte of a format-specific packet may fall anywhere within the Payload Data.

When **dwPacketLength** is less than or equal to the Payload Data size, the Payload Data portion of a Payload Transfer shall contain an integral number of format-specific packets. If, due to timing considerations, no format-specific packets are available over a period of one or more (micro)frames, those (micro)frames shall be empty.

2.2.2 Byte-Oriented Payload Data

Byte-Oriented streams do not have any Payload Data boundary constraints. Payload Transfers can be of variable length within the limit set by **dwMaxPayloadTransferSize** of the Video Probe and Commit Control. A sequence of one or more variable length packets within a Byte-Oriented stream can be indicated via the FID and EOF bits of the Payload Header. For additional details, see section 4.3.1.1 of the *USB Device Class Definition for Video Devices* specification.

3 Payload-Specific Information

3.1 Descriptors

3.1.1 Stream Based Format Descriptor

The Stream Based Format Descriptor defines the characteristics of the Stream Based payload. A USB IN or OUT endpoint, and the interface it belongs to, supports one or more format definitions.

Stream Based Format Descriptor is defined in Table 3-1.

Table 3-1 Stream Based Format Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	Number	Size of this Descriptor, in bytes: 24
1	**bDescriptorType**	1	Constant	CS_INTERFACE Descriptor type
2	**bDescriptorSubtype**	1	Constant	VS_FORMAT_STREAM_BASED Descriptor subtype
3	**bFormatIndex**	1	Number	Index of this Format Descriptor
4	**guidFormat**	16	GUID	Globally Unique Identifier used to identify stream-encoding format
20	**dwPacketLength**	4	Number	If non-zero, indicates a format-specific packet size in a Packet-Oriented stream. If zero, indicates that the format-specific data is either Byte-Oriented, or consists of variable size format-specific packets. See section 2.2, “Payload Data” of this specification for more details.

3.2 Video Samples

Since this specification addresses Stream Based Format over USB Video Class, Video sample information, such as aspect ratio, picture position, and so on, are not included in this Stream Based format specification. This information is encapsulated within the format-specific stream.

For example, this information is described as a profile in ISO/IEC 11172 for MPEG-1 SS streams.

4 Examples

4.1 Isochronous Transfer IN

The following example shows the relationship between Payload Transfers, the token and data packets when receiving isochronous transfers from the device. This example shows high-speed, high-bandwidth transfers, but this is only illustrative and not a requirement of the Stream Based payload format. The actual bandwidth usage will vary according to the requirements of the device.

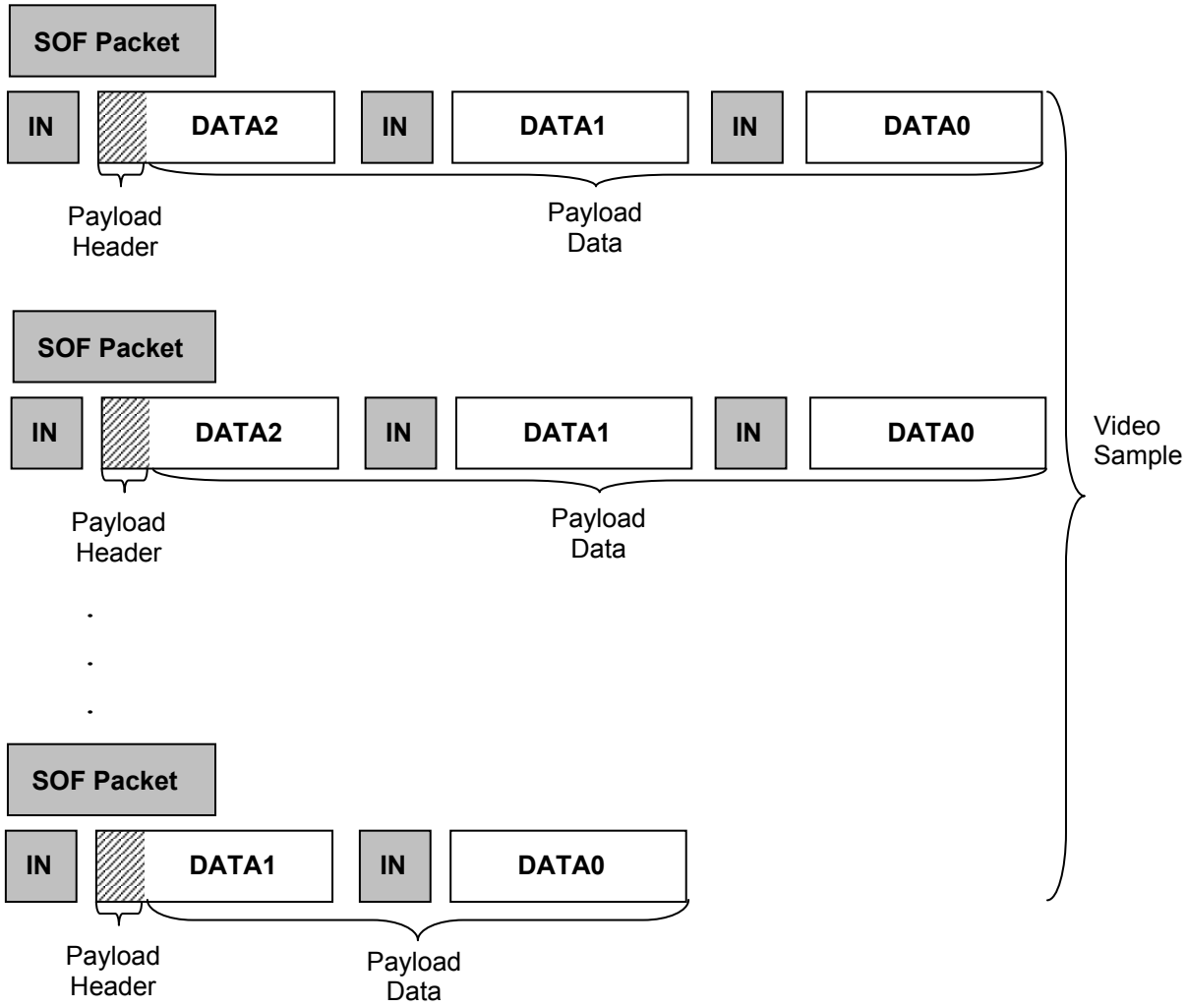

Figure 4-1 Example Stream Based Isochronous Transfer, IN Endpoint

4.2 Isochronous Transfer OUT

The following example shows the relationship between Payload Transfers, the token and data packets when sending isochronous transfers to the device. This example shows high-speed, high-bandwidth transfers, but this is only illustrative and not a requirement of the Stream Based payload format. The actual bandwidth usage will vary according to the requirements of the device.

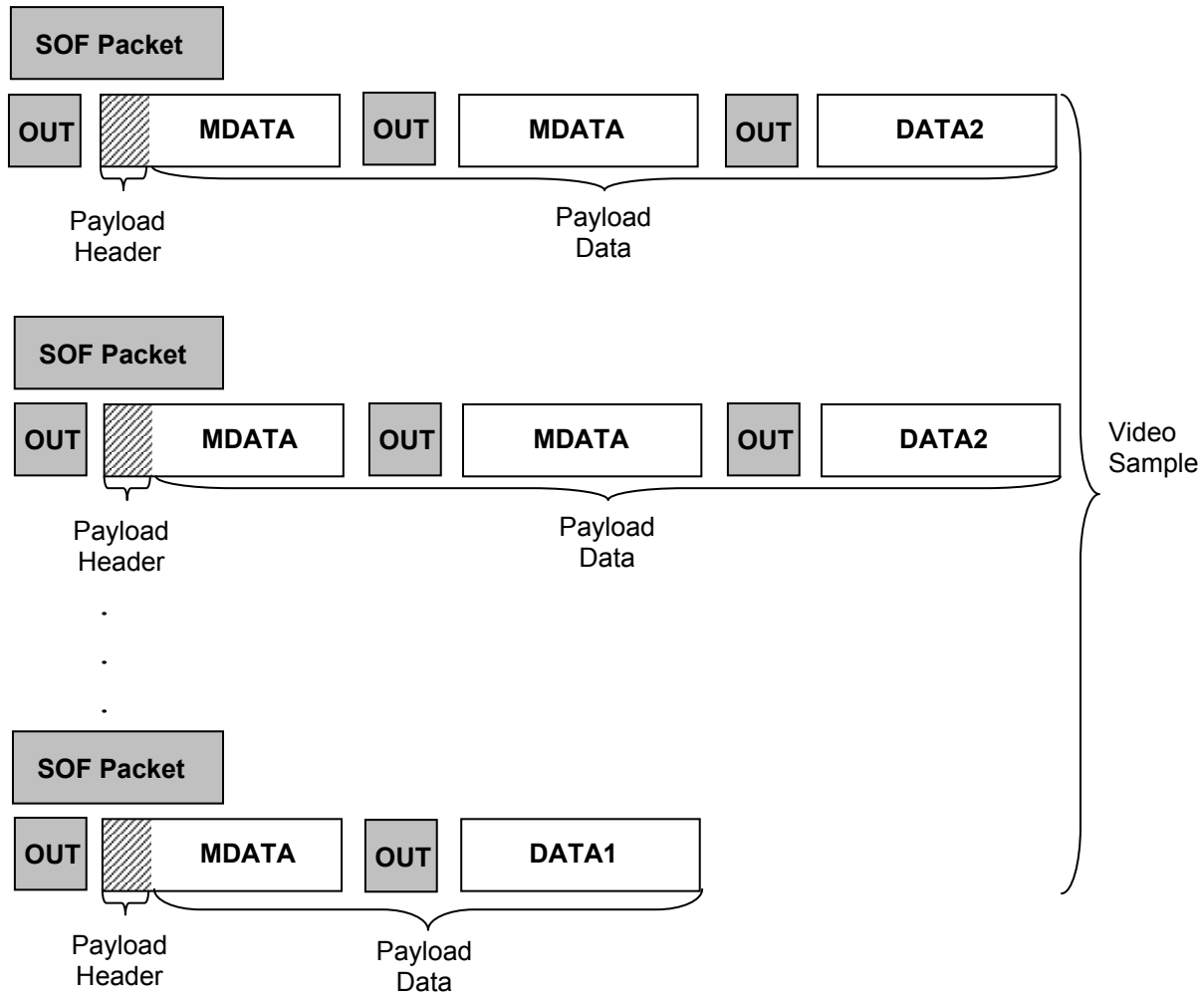

Figure 4-2 Example Stream Based Isochronous Transfer, OUT Endpoint

4.3 Bulk Transfer IN

The following example shows the relationship between Payload Transfers, the token and data packets of the Stream Based payload format when receiving bulk transfers from a device. Handshake packets are not shown for the sake of clarity.

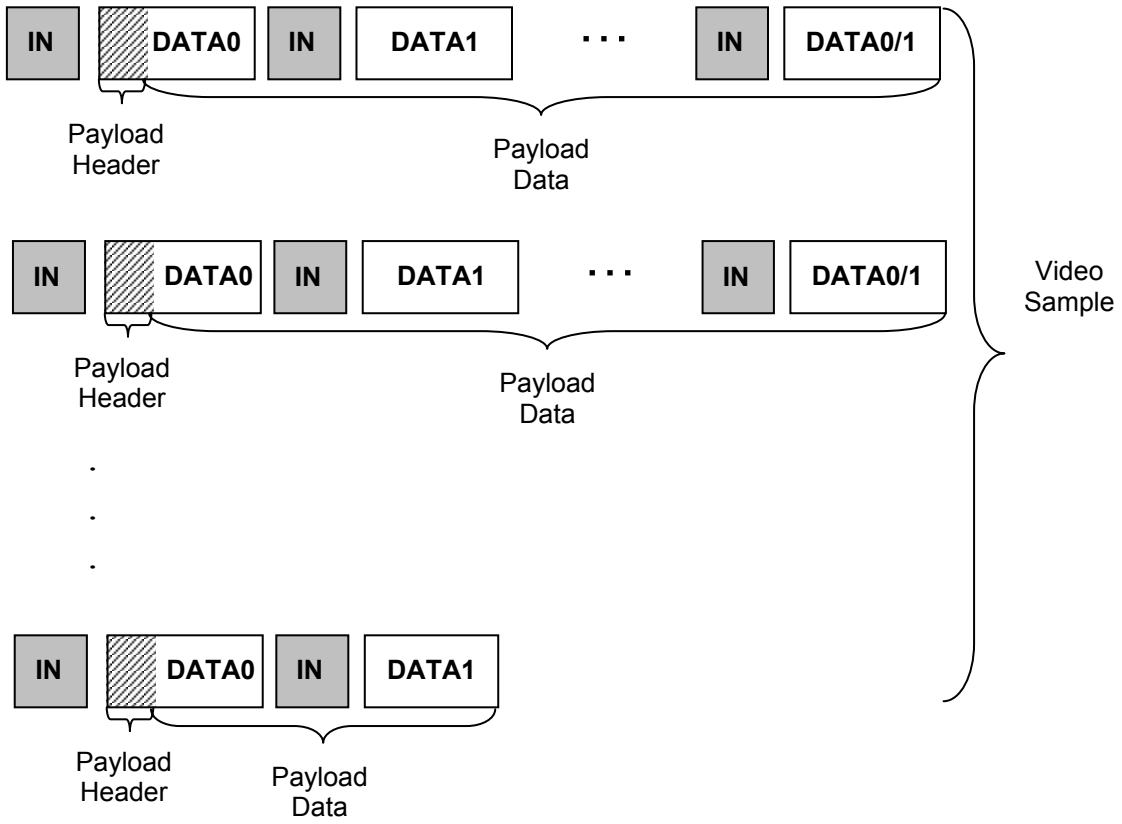

Figure 4-3 Example Stream Based Bulk Transfer, IN Endpoint

4.4 Bulk Transfer OUT

The following example shows the relationship between Payload Transfers, the token and data packets of the Stream Based payload format when sending bulk transfers to the device. Handshake packets are not shown for the sake of clarity.

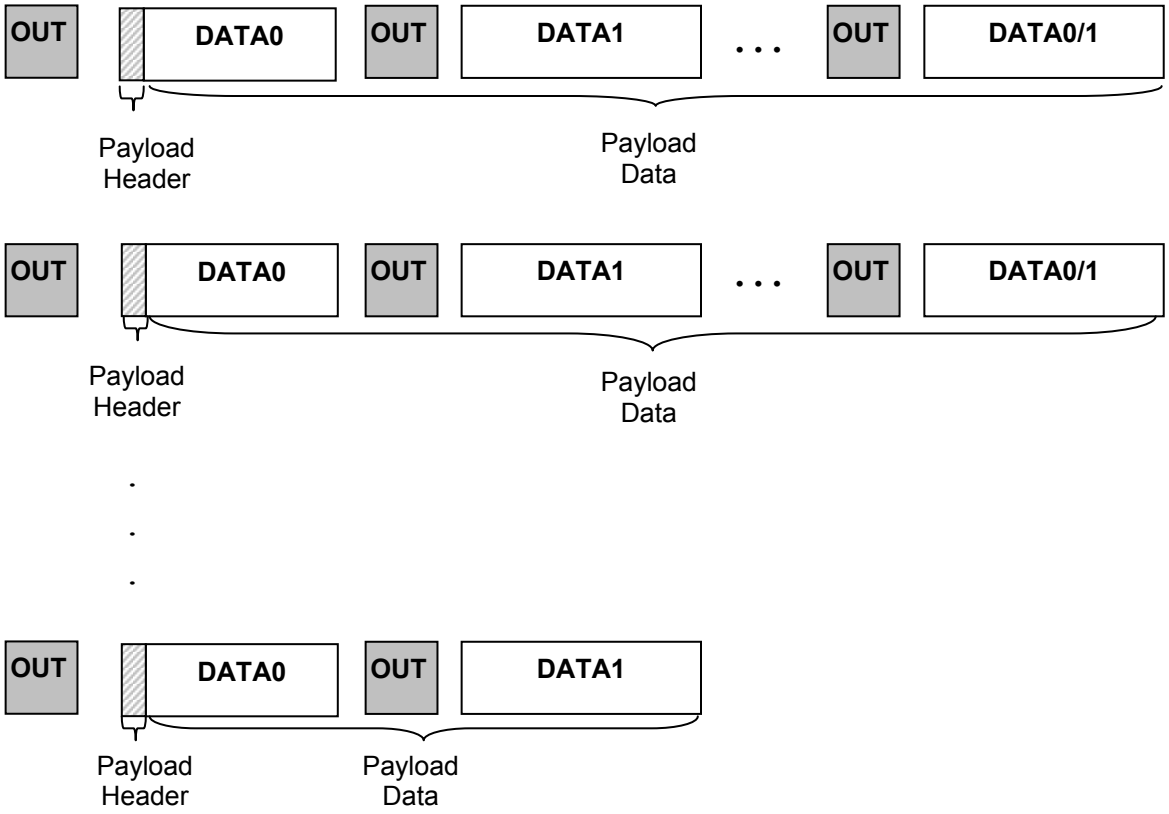

Figure 4-4 Example Stream Based Bulk Transfer, OUT Endpoint