USB ENGINEERING CHANGE NOTICE

Title: USB 2.0 Double Isochronous IN Bandwidth

Applies to: Universal Serial Bus Specification, Revision 2.0

Summary of ECN

This ECN doubles the amount of packets that an Isochronous IN endpoint, within a directly connected inbox native eUSB2 device, can transfer in a microframe.

Reasons for ECN

In order to support higher camera resolutions on the lid of laptops and tablets with minimal change to the USB2 protocol this ECN allows directly connected inbox native eUSB2 devices to transfer higher amounts of IN data per microframe.

Impact on Existing Peripherals and Systems:

There is no impact to existing peripherals and systems.

Hardware Implications:

No change is required of existing host and peripheral hardware.

Hosts and devices that want to take advantage of this feature will need to be updated to support double the number of packets per isochronous IN endpoint in a microframe.

Software Implications:

Host software will need to be updated to realize the benefits of doubling the bandwidth.

Compliance Testing Implications:

Compliance tests will be updated to test compliance of new hosts and devices to this ECN.

Specification Changes

5.6.3 Isochronous Transfer Packet Size Constraints

An endpoint in a given configuration for an isochronous pipe specifies the maximum size data payload that it can transmit or receive. The USB System Software uses this information during configuration to ensure that there is sufficient bus time to accommodate this maximum data payload in each (micro)frame. If there is sufficient bus time for the maximum data payload, the configuration is established; if not, the configuration is not established.

The USB limits the maximum data payload size to 1,023 bytes for each full-speed isochronous endpoint. High-speed endpoints are allowed up to 1024-byte data payloads. A high speed, high bandwidth endpoint specifies whether it requires two or three transactions per microframe. A high bandwidth eUSB2 Isochronous IN endpoint (i.e. an Isochronous IN endpoint within a directly connected inbox native eUSB2 device) can transfer up to 6 transactions per microframe. Table 5-4 lists information about different-sized full-speed isochronous transactions and the maximum number of transactions possible in a frame.

5.6.4 Isochronous Transfer Bus Access Constraints

…

The USB requires that no more than 90% of any frame be allocated for periodic (isochronous and interrupt) transfers for full-speed endpoints. High-speed endpoints can allocate at most 80% of a microframe for periodic transfers. High-speed eUSB2 endpoints can allocate up to 95% of a microframe for periodic transfers[1].

…

A high-speed endpoint can move up to 3072 bytes per microframe (or 192 Mb/s). A high-speed high bandwidth eUSB2 Isochronous IN endpoint can move up to 6144 bytes per microframe (or 384 Mb/s).   A high-speed isochronous endpoint that requires more than 1024 bytes per period is called a high-bandwidth endpoint. A high-bandwidth endpoint uses multiple transactions per microframe. A high-bandwidth endpoint must specify a period of 1x125 μs (i.e., a bInterval value of 1).

See Section 5.9 for more information about the details of multiple transactions per microframe for high-bandwidth high-speed endpoints.
…

5.9 High-Speed, High Bandwidth Endpoints

USB supports individual high-speed interrupt or isochronous endpoints that require data rates up to 192 Mb/s (i.e., 3072 data bytes per microframe). One, two, or three high-speed transactions are allowed in a single microframe to support high-bandwidth endpoints. Directly connected inbox native eUSB2 Devices are allowed to support up to six high-speed transactions per microframe to a high bandwidth eUSB2 Isochronous IN endpoint that require data rates up to 384 Mb/s (i.e. 6144 data bytes per microframe)[2]. See Section 9.6.6 on how to describe the amount of bandwidth requested by a high bandwidth eUSB2 Isochronous IN endpoint.

…

---

[1] Note: This may not leave enough time to perform a max packet size Bulk transaction.
[2] Note: This does not take into account bitstuffing. In the case of worst case bitsuffing, an endpoint will only be able to deliver up to 320 Mb/s (i.e. 5120 data bytes per microframe)

5.9.2.1 <mark>High Bandwidth eUSB2 Isochronous IN Endpoints</mark>

<<This section is added to the end of section 5.9.2>>

The differences in behavior of a high bandwidth eUSB2 Isochronous IN endpoints are described in this section. High bandwidth eUSB2 Isochronous IN endpoints must support data PID sequencing. Data PID sequencing provides the required support for the data receiver to detect one or more lost/damaged packets per microframe. Each packet except the last packet shall be 1024 bytes in length.

Data PID sequencing for a high bandwidth eUSB2 Isochronous IN endpoint uses a repeating sequence of DATA2, DATA1, DATA0 PIDs for the data packet of each transaction in a microframe. If there is only a single transaction in the microframe, only a DATA0 data packet PID is used. If there are two transactions per microframe, DATA1 is used for the first transaction data packet and DATA0 is used for the second transaction data packet. If there are three transactions per microframe, DATA2 is used for the first transaction data packet, DATA1 is used for the second, and DATA0 is used for the third. In case there are four transactions, then after the first three transactions the device shall send a DATA0 for the fourth. In case there are five transactions, then after the first three transactions the device shall send a DATA1 for the fourth and a DATA0 for the fifth. In case there are six transactions, then after the first three transactions the device shall send a DATA2 for the fourth, a DATA1 for the fifth and a DATA0 for the sixth.

In all cases, the data PID sequence starts over again the next microframe.

Errors are treated the same for high bandwidth eUSB2 Isochronous endpoints as they are treated for regular high bandwidth USB2 Isochronous endpoints.

Table 9-1. Standard Device Descriptor

Offset	Field	Size	Value	Description
0	*bLength*	1	Number	Size of this descriptor in bytes
1	*bDescriptorType*	1	Constant	DEVICE Descriptor Type
2	*bcdUSB*	2	BCD	USB Specification Release Number in Binary-Coded Decimal (i.e., 2.10 is 210H). This field identifies the release of the USB Specification with which the device and its descriptors are compliant. <mark>This field shall be set to 0x200 for USB 2.0 Devices</mark> <mark>This field shall be set to 0x201 for USB 2.0 Devices that support the GetDescriptor (BOS Descriptor) request</mark> <mark>This field shall be set to 0x210 for USB 2.0 devices that also support USB 3.x</mark> <mark>This field shall be set to 0x220 for directly connected inbox native eUSB2 Devices that support Double Isochronous IN Bandwidth</mark>
4	…	…	…	…

9.6.6 Endpoint

…

For high bandwidth eUSB2 Isochronous IN Endpoint that require bandwidths above 3KB/microframe, the standard Endpoint descriptor shall declare a zero bandwidth setting, i.e., the wMaxPacketSize field shall be set to 0, and the actual maximum packet size and bandwidth requirements shall be defined in the eUSB2 Iso Endpoint Companion descriptor wMaxPacketSize and dwBytesPerInterval fields, respectively. Note that Devices shall also implement one or more non-zero bandwidth alternate settings with a non-zero wMaxPacketSize (up to 3KB/microframe) in the standard Endpoint descriptor.

This approach enables legacy software to see compatible standard Endpoint descriptors when operating at 3KB/microframe bandwidth or below, and zero bandwidth configurations when operating above. Only an eUSB2 aware stack can take advantage of eUSB2 bandwidth.

For high-speed bulk and control OUT endpoints, the bInterval field is only used for compliance purposes; the host controller is not required to change its behavior based on the value in this field.

9.6.6.1 eUSB2 Isochronous Endpoint Companion Descriptor

This descriptor shall only be returned by high bandwidth eUSB2 Isochronous IN endpoints that support more than 3K/microframe bandwidth. An eUSB2 Iso Endpoint Companion descriptor for an high bandwidth eUSB2 Isochronous IN endpoint shall be declared before the next endpoint or interface descriptor is declared. This descriptor is returned as part of the configuration information returned by a GetDescriptor(Configuration) request and cannot be directly accessed with a GetDescriptor() or SetDescriptor() request.

When an alternate setting is selected that has a high bandwidth eUSB2 Isochronous IN endpoint, with an eUSB2 Iso Endpoint Companion descriptor, the endpoint shall operate with the characteristics as described in the eUSB2 Iso Endpoint Companion descriptor.

Table 9-x: eUSB2 Iso Endpoint Companion Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	Number	Size of this descriptor in bytes (8).
1	bDescriptorType	1	Constant	EUSB2_ISO_ENDPOINT_COMPANION Descriptor Type.
2	wMaxPacketSize	2	Number	The maximum packet size this endpoint is capable of sending or receiving when this configuration is selected. This shall be set to 1024.
4	dwBytesPerInterval	4	Number	The total number of bytes this endpoint will transfer every service interval (SI). This value is used to reserve the bus time in the schedule, required for the frame data payloads per SI. The pipe may, on an ongoing basis, actually use less bandwidth than that reserved. The device reports, if necessary, the actual bandwidth used via its normal, non-USB defined mechanisms. This field shall have a value between 3073 and 6144 (Inclusive)

Table 9-2. Descriptor Types

Descriptor Types	Value
DEVICE	1
CONFIGURATION	2
STRING	3
INTERFACE	4
ENDPOINT	5
DEVICE_QUALIFIER	6
OTHER_SPEED_CONFIGURATION	7

INTERFACE_POWER[3]	8
EUSB2_ISO_ENDPOINT_COMPANION	18

---

[3] The INTERFACE_POWER descriptor is defined in the current revision of the *USB Interface Power Management Specification*.