Document Number: DSP0254

Date: 2010-07-21

Version: 1.0.0

Management Component Transport Protocol (MCTP) KCS Transport Binding Specification

Document Type: Specification

Document Status: DMTF Standard

Document Language: en-US

CONTENTS

Forward 5
Introduction 6
1 Scope 7
2 Normative References 7
3 Terms and Definitions 7
4 Symbols and Abbreviated Terms 8
5 Conventions 8
5.1 Reserved and Unassigned Values 8
5.2 Byte Ordering 8
6 MCTP over KCS Transport 8
6.1 MCTP Packet Encapsulation 9
6.2 Error Handling 11
6.3 Interface Related Data 11
6.3.1 SMBIOS Management Controller Host Interface Structure Type 42 11
6.3.2 PCI / PCIe Class Codes 13
6.4 Supported Media 13
7 Transport-Specific Commands 14
7.1 Register Endpoint 14
7.2 Get MCTP Packet 14
7.3 Enable MCTP SMS_ATN 15
7.4 Transport Strings 15
8 Incoming and Outgoing KCS MCTP Packets 15
8.1 Get Incoming MCTP Packet Examples 16
8.1.1 Polling Example 16
8.1.2 SMS_ATN Example 16
9 MCTP KCS Packet Timing Requirements 17
10 MCTP KCS Control Message Timing Requirements 17
ANNEX A (informative) Notation and Conventions 19
ANNEX B (informative) Change Log 20

Figures

Figure 1 – MCTP over KCS Packet Format 9

Tables

Table 1 – MCTP over KCS Packet Header Field Descriptions 10
Table 2 – Management Controller Device Information: Interface Specific Data for KCS 11
Table 3 – Byte-aligned I/O Mapped Register Address Examples 12
Table 4 – 32-bit Aligned I/O Mapped Register Address Examples 12
Table 5 – Supported Media 13
Table 6 – Transport-Specific MCTP Control Command Number 14
Table 7 – Register Endpoint Message Format 14
Table 8 – Get MCTP Packet Message Format 14
Table 9 – Enable MCTP SMS_ATN 15
Table 10 – Timing Specifications for MCTP Packets on KCS 17
Table 11 – Timing Specifications for MCTP Control Messages on KCS 17

Forward

The *Management Component Transport Protocol (MCTP) KCS Transport Binding Specification* (DSP0254) was prepared by the PMCI Subgroup of the Pre-OS Working Group.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. For information about the DMTF, see http://www.dmtf.org.

Introduction

The Management Component Transport Protocol (MCTP) defines a communication model intended to facilitate communication between:

- Management controllers and other management controllers
- Management controllers and management devices

The communication model includes a message format, transport description, message exchange patterns, and configuration and initialization messages.

The *MCTP Base Specification* (DSP0236) describes the protocol and commands used for communication within and initialization of an MCTP network. Associated with the *MCTP Base Specification* are transport-binding specifications that define how the MCTP base protocol and MCTP control commands are implemented on a particular physical transport type and medium, such as SMBus/I^2C, PCI Express™ (PCIe) Vendor Defined Messaging, KCS, Serial, and so on.

Management Component Transport Protocol (MCTP) KCS Transport Binding Specification

1 Scope

This document provides the specifications for the Management Component Transport Protocol (MCTP) transport binding for Keyboard Controller Style (KCS) interface.

2 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

DMTF DSP0134, *System Management BIOS Reference Specification 2.6,* http://www.dmtf.org/standards/published_documents/DSP0134_2.6.pdf

DMTF DSP0236, *Management Component Transport Protocol (MCTP) Base Specification 1.0,* http://www.dmtf.org/standards/published_documents/DSP0236_1.0.pdf

DMTF DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes Specification 1.0,* http://www.dmtf.org/standards/published_documents/DSP0239_1.0.pdf

DMTF, DSP0256, *Management Component Transport Protocol (MCTP) Host Interface Specification 1.0,* http://www.dmtf.org/standards/published_documents/DSP0256_1.0.pdf

IPMI Consortium*, Intelligent Platform Management Interface Specification*, v1.5 Revision 1.1 February 20, 2002, http://download.intel.com/design/servers/ipmi/IPMIv1_5rev1_1.pdf

PCI-SIG, *PCI Local Bus Specification v3.0*, PCI v3.0, February 3, 2004, http://www.pcisig.com/specifications/conventional

SBS Implementers Forum, *System Management Bus (SMBus) Specification v2.0*, SMBus, August 2000, http://www.smbus.org/specs/smbus20.pdf

3 Terms and Definitions

Refer to DSP0236 for terms and definitions that are used across the MCTP specifications. For the purposes of this document, the following terms and definitions apply.

3.1

Keyboard Controller Style Interface (KCS)

A set of bit definitions, and operation of the registers typically used in keyboard microcontrollers and embedded controllers. The term ”Keyboard Controller Style” reflects that the register definition was originally used as the legacy “8742” keyboard controller interface in PC architecture computer systems. This interface is available built-in to several commercially available microcontrollers. Data is transferred across the KCS interface using a per-byte handshake.

136 **3.2**
137 **Logical Endpoint**
138 An endpoint that can be represented by system firmware or system software.

139 4 Symbols and Abbreviated Terms

140 Refer to DSP0236 for symbols and abbreviated terms that are used across the MCTP specifications. For
141 the purposes of this document, the following additional symbols and abbreviated terms apply.

142 **4.1**
143 **KCS**
144 Keyboard Controller Style Interface

145 **4.2**
146 **LUN**
147 Logical Unit Number

148 **4.3**
149 **MCTP**
150 Management Component Transport Protocol

151 **4.4**
152 **PEC**
153 Packet Error Code

154 5 Conventions

155 The conventions described in the following clauses apply to this specification.

156 5.1 Reserved and Unassigned Values

157 Unless otherwise specified, any reserved, unspecified, or unassigned values in enumerations or other
158 numeric ranges are reserved for future definition by the DMTF.

159 Unless otherwise specified, numeric or bit fields that are designated as reserved shall be written as 0
160 (zero) and ignored when read.

161 5.2 Byte Ordering

162 Unless otherwise specified, byte ordering of multi-byte numeric fields or bit fields is "Big Endian" (that is,
163 the lower byte offset holds the most significant byte, and higher offsets hold lesser significant bytes).

164 6 MCTP over KCS Transport

165 The KCS interface is a set of bit definitions, and operation of the registers that is typically used in
166 keyboard microcontrollers. The term ”Keyboard Controller Style” reflects the fact that the host interface
167 was used as the legacy keyboard controller interface in PC architecture computer systems. This interface
168 is available built-in to several commercially available microcontrollers. Data is transferred across the KCS
169 interface using a per-byte handshake.

The MCTP over KCS transport binding defines how MCTP packets are delivered over a physical KCS interface using KCS transactions. Timing specifications for bus and MCTP control operations are also given.

The binding has been designed to be able to share the same bus as devices communicating using earlier KCS management protocols such as IPMI. For a more detailed explanation of the KCS interface with flowcharts see *Intelligent Platform Management Interface Specification*, Chapter 9.

This specification covers MCTP over KCS transport binding only. For further description of the MCTP host interface refer to DSP0256, *Management Component Transport Protocol (MCTP) Host Interface Specification.*

6.1 MCTP Packet Encapsulation

MCTP Packet Encapsulation for KCS shall support the baseline MTU-sized MCTP packet. For example, a baseline transmission unit of 64 bytes will result in a KCS message size of 64 + 3 (KCS header length) + 4 (MCTP transport length) + 1 (KCS PEC) = 72 bytes. MCTP over KCS packets use the first 3 data bytes of the KCS transaction to make up the KCS packet header. The first byte maps to a Group Extension Net Fn / LUN code of 0xB0 as defined in *Intelligent Platform Management Interface Specification*, section 5.1. The second byte maps to the defining body of DMTF = 0x01. This is followed by a byte count. Bytes 0:2 and byte N represent the medium specific data. This includes the KCS medium specific header (bytes 0: 2) and the medium specific trailer (byte n). Bytes 3:6 in Figure 1 represent fields defined by the *Base MCTP Specification* and include the MCTP transport header. Bytes 7: (N-1) represent the MCTP packet payload which includes the MCTP message header and message data.

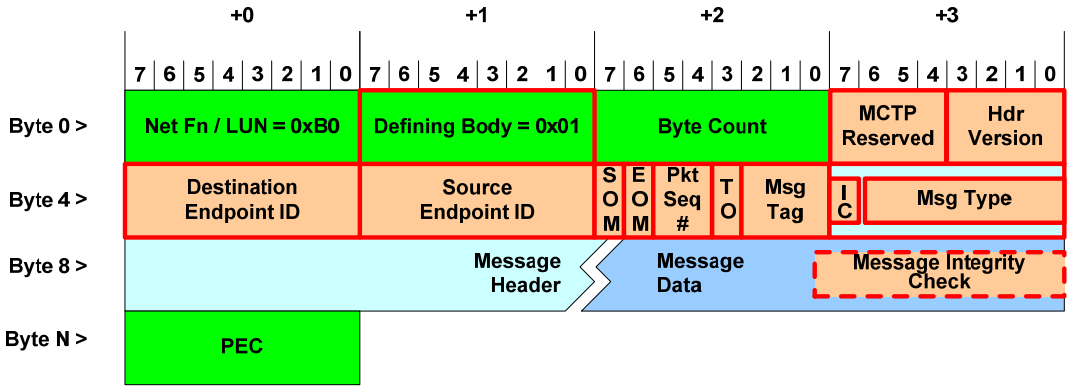

Figure 1 – MCTP over KCS Packet Format

The fields labeled “KCS Medium-Specific Header” and “KCS Medium-Specific Trailer” are specific to carrying MCTP packets using KCS. The fields labeled “MCTP Transport Header” and “MCTP Packet Payload” are common fields for all MCTP packets and messages and are specified in MCTP. This document defines the location of those fields when they are carried over KCS. This document also specifies the *medium-specific* use of the MCTP “Hdr Version” field.

Table 1 lists the KCS medium-specific fields as well as common fields and field values.

198 **Table 1 – MCTP over KCS Packet Header Field Descriptions**

Field	Field Size	Description
Network Function	5 bits	Network Function: Group Extension = 101100b = 0x2C NOTE: MCTP does not use odd network function 0x2D.
LUN	2 bits	Logical Unit Number : Set to 00b for all KCS over MCTP packets
Defining Body	1 byte	Defining Body = DMTF Pre-OS Working Group = 0x01
Byte Count	1 byte	Byte Count: Byte count for the KCS transaction that is carrying the MCTP packet content. This value is the count of bytes that follow the Byte Count field up to, but not including, the PEC byte. For example, if the MCTP packet payload length (starting with byte 7) is 64 bytes then the value in the Length field would be 68. (The count of 68 accounts for 64 bytes of MCTP packet payload plus the four bytes [bytes 3 through 6, inclusive] that comprise the bytes of the MCTP header that follow the Byte Count field.)
MCTP Reserved	4 bits	MCTP reserved: This nibble is reserved for definition by DSP0236.
MCTP Header Version	4 bits	MCTP header version: Set to 0001b for MCTP devices that are conformant to DSP0236 and this version of the KCS transport binding. All other values = Reserved.
Destination Endpoint ID	1 byte	Destination Endpoint ID (*)
Source Endpoint ID	1 byte	Source Endpoint ID (*)
SOM	1 bit	SOM: Start Of Message flag. (*)
EOM	1 bit	EOM: End Of Message flag. (*)
Packet Sequence Number	2 bits	Packet Sequence Number (*)
Tag Owner (TO) bit	1 bit	Tag Owner (TO) bit (*)
Message Tag	3 bits	Message Tag (*)
Message Type	1 byte	Message Type: (*)
Message Header and Data	Varies	Message Header and Data: (*)
PEC	1 byte	Packet error code (PEC), as defined in the *SMBus 2.0 Specification*. The PEC is calculated from Byte 0 to Byte n-1. All KCS MCTP transactions shall include a PEC byte. The PEC byte must be transmitted by the source and checked by the destination.
(*) Indicates a field that is defined by DSP0236, *Management Component Transport Protocol (MCTP) Base Specification.*		

6.2 Error Handling

A packet is required to be dropped if the packet error code (PEC) byte for the transaction is incorrect.

Refer to *Intelligent Platform Management Interface Specification* for further information on error handling on the KCS interface.

6.3 Interface Related Data

The MCTP KCS interface has interface related data for use by the host for discovery. This discovery process is described in DSP0256, *Management Component Transport Protocol (MCTP) Host Interface Specification.* The MCTP KCS Interface related data for SMBIOS, ACPI and PCI / PCIe is described in the following clauses.

6.3.1 SMBIOS Management Controller Host Interface Structure Type 42

The SMBIOS Management Controller Host Interface Structure Type 42 is described in DSP0134, *System Management BIOS Reference Specification*. The KCS interface-specific data is described in Table 2.

Table 2 – Management Controller Device Information: Interface Specific Data for KCS

Offset	Name	Length	Description
00h	Base Address	QWORD	Identifies the base address (either memory-mapped or I/O) of the management controller. If the least-significant bit of the field is a 1, the address is in I/O space; otherwise, the address is memory-mapped.
08h	Base Address Modifier / Interrupt Info	BYTE	Base Address Modifier bit 7:6 – Register spacing 00b = interface registers are on successive byte boundaries 01b = interface registers are on 32-bit boundaries 10b = interface registers are on 16-byte boundaries 11b = reserved bit 5 – Reserved. Return as 0b. bit 4 – LS-bit for addresses 0b = Address bit 0 = 0b 1b = Address bit 0 = 1b Interrupt Info Identifies the type and polarity of the interrupt associated with the host interface, if any. bit 3 – 1b = Interrupt info specified 0b = Interrupt info not specified bit 2 – Reserved. Return as 0b. bit 1 – Interrupt Polarity. 1b = active high, 0b = active low. bit 0 – Interrupt Trigger Mode. 1b = level, 0b = edge.
0Ah	Interrupt Number	BYTE	Interrupt number for MCTP Host interface. 00h = unspecified / unsupported

6.3.1.1 Base Address Field

This field is used to describe the base address for the management controller's host interface. The field can describe both I/O mapped and memory-mapped base addresses. The least significant bit of this field indicates whether the base address is an I/O address or a memory address. The most significant 63-bits of this field holds the most significant 63 bits (bits 63:1) of a 64-bit address. The least significant bit (bit 0) of the base address is kept in the Base Address Modifier field.

All management controller host interface registers are inherently non-cacheable and the register locations must be implemented as non-cacheable addresses.

6.3.1.2 Base Address Modifier Field

This field provides the least-significant bit for the base address, information indicating how the host interface registers are aligned (either on byte, 32-bit, or 16-byte boundaries).

6.3.1.3 Host Interface Register Alignment

Host interface registers can optionally be defined on 32-bit or 16-byte boundaries. In this case, the registers are 32-bits (4 bytes) apart. Base addresses must match the specified register alignment. For example, the base address for a 32-bit aligned interface must have its two least significant address bits = 00b. Thus, the LS bit field in the Base Address Modifier is always 0b for non-byte-aligned addresses.

6.3.1.4 Byte-spaced I/O Address Examples

Table 3 shows how the default host interface addresses would be represented in the SMBIOS Base Address and Base Address Modified fields. Base Address bit 0 = 1b indicates that the base address is an I/O address. The default host interface definition specifies that the host interface registers occupy consecutive byte locations. Thus, the register spacing in the Base Address Modifier is set to 0b. Note that the LS bit field in the Base Address Modifier field matches the least-significant bit listed in the corresponding addresses from the Default Base Address column.

Table 3 – Byte-aligned I/O Mapped Register Address Examples

Interface	Default Base Address	SM BIOS Base Address	LS Bit Field	Register spacing
KCS	0CA2h	0000 0000 0000 CA3h	0b	00b

6.3.1.5 32-bit Spaced I/O Address Examples

Table 4 shows example addresses for a KCS interface implemented with 32-bit aligned registers at I/O base address Cache.

Table 4 – 32-bit Aligned I/O Mapped Register Address Examples

	Example I/O Address	SM BIOS Base Address	LS bit field	Register Spacing
base address	0000 0CACh	0000 0000 0000 0CADh	0b	01b
Data_In	0000 0CACh	0000 0000 0000 0CADh	0b	01b
Data_Out	0000 0CACh	0000 0000 0000 0CADh	0b	01b
Command	0000 0CB0h	0000 0000 0000 0CB1h	0b	01b

Status	0000 0CB0h	0000 0000 0000 0CB1h	0b	01b

6.3.1.6 Memory-mapped Base Address

For memory-mapped host interfaces, the Base Address field and Base Address Modifier are used in the same manner as for an I/O-mapped interface, except that Base Address bit 0 is set to 0b.

6.3.1.7 Interrupt Info Field

This field identifies the type and polarity of the interrupt associated with the MCTP host interface, if any. Refer to Table 2 for individual bit descriptions.

6.3.1.8 Interrupt Number Field

This field holds the interrupt number for the MCTP Host Interface. The field is set to 00h when the number is unspecified or an interrupt is not supported.

6.3.1.9 ACPI MCHI Description Table

The ACPI MCHI Description Table is described in DSP0256, *Management Component Transport Protocol (MCTP) Host Interface Specification*. The interface type record in this structure should be set to Keyboard Controller Style (KCS) for MCTP KCS transport.

6.3.2 PCI / PCIe Class Codes

The PCI SIG (http://www.pcisig.com) has defined class codes for IPMI Host interfaces in Appendix D of the *PCI Local Bus Specification*. PCI/PCIe -based implementations of the MCTP KCS Host interfaces should use the following PCI/PCIe class codes for MCTP KCS:

Class Code = Serial Bus Controllers

Sub Class Code = IPMI Host Interfaces

Interface = IPMI Keyboard Controller Style (KCS) Interface

6.4 Supported Media

This physical transport binding has been designed to work with the media specified in Table 5. Use of this binding with other types of physical media is not covered by this specification. Refer to DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes Specification,* for Physical Medium Identifier values.

Table 5 – Supported Media

Description
KCS / Legacy
KCS / PCI

7 Transport-Specific Commands

Table 6 lists the Transport-specific MCTP control messages for the MCTP KCS transport and the corresponding command code values.

Table 6 – Transport-Specific MCTP Control Command Number

Command Code	Command Name	General Description	Clause
0xF0	Register Endpoint	Registers a UUID with the management controller and receives an MCTP EID and TID in the response	7.1
0xF1	Get MCTP Packet Datagram	Reads an MCTP packet that is available	7.2
0xF2	Enable MCTP SMS_ATN	Enables and disables MCTP over KCS to set the SMS_ATN flag. The default is disabled.	7.3

7.1 Register Endpoint

The Register Endpoint command is used by the system firmware or system software to send a universally unique identifier (UUID), also referred to as a globally unique ID (GUID), to the management controller in order to obtain an MCTP EID. The format of the ID follows the byte (octet) format specified in the *MCTP Base Specification* (DSP0236). The request and response parameters are specified in Table 7.

Table 7 – Register Endpoint Message Format

	Byte	Description
Request data	1:16	UUID bytes 1:16, respectively (see Table 3)
Response data	1	Completion Code
	2	MCTP EID

7.2 Get MCTP Packet

Get MCTP Packet is an MCTP control datagram message that is used by the system firmware or system software to get an incoming MCTP packet from the management controller in the KCS Read state if one is available. For details on the use of this datagram, see clause 8. The request and response parameters are specified in Table 8.

Table 8 – Get MCTP Packet Message Format

	Byte	Description
Request data	–	–
Response data	–	N/A for datagrams

NOTE: An MCTP control datagram is a request message that does not have a corresponding response.

7.3 Enable MCTP SMS_ATN

This command is used by software to enable MCTP over KCS to set the SMS_ATN flag. In order to retain backward compatibility with software for IPMI, the ability for MCTP over KCS to set SMS_ATN is disabled by default. The default state at system reset (for example power cycling of the system or master bus reset), power up and Management Controller re-initialization shall be MCTP SMS_ATN disabled. The request and response parameters are specified in Table 9.

Table 9 – Enable MCTP SMS_ATN

	Byte	Description
Request data	1	MCTP SMS_ATN 0x00 = Disabled 0x01 = Enabled 0x02 – 0xff = Reserved
Response data	1	Completion Code

7.4 Transport Strings

For identifying MCTP host interface the string "MCTP_KCS" is recommended for identifying the KCS interface.

8 Incoming and Outgoing KCS MCTP Packets

Typically the KCS interface is used as an interface between the host and the management controller. Commands can be initiated either by system firmware (such as BIOS or UEFI) or by OS system software. Incoming MCTP packets are returned in the read state of the KCS transaction if a packet is available. In many cases following the KCS write state, data will be available and can be read in the KCS read state immediately following the write state. This will be indicated by a byte count that is larger than zero in the KCS read state MCTP packet. There may be cases when the management controller asynchronously needs to notify the host that it has an MCTP packet available.

In the case where a zero byte count is received in the KCS read state, the *Get MCTP Packet Datagram* can be used to poll for incoming MCTP packets and to retrieve them if available. The *Get MCTP Packet Datagram* can also be used to retrieve incoming MCTP packets when the SMS_ATN flag indicates an incoming MCTP packet is available. The *Get MCTP Packet Datagram* data format is listed in 7.2.

The SMS_ATN bit can also be used when the KCS interface is interrupt driven. Refer to the *Intelligent Platform Management Interface Specification* sections 9.12, “KCS Communication and Non-communication Interrupts”; 9.13, “Physical Interrupt Line Sharing”; and 9.14, “Additional Specifications for the KCS Interface” for additional information on the use and requirements for the SMS_ATN bit.

If IPMI is being used as well as MCTP, the IPMI command *Get Message Flags* may be sent to the Management Controller to identify which IPMI conditions are causing the SMS_ATN flag to be set. If there are no flags set, the software will assume the SMS_ATN was set by the Management Controller for an MCTP packet to be retrieved. All conditions must be cleared (that is, all messages must be flushed) in order for the SMS_ATN bit to be cleared.

8.1 Get Incoming MCTP Packet Examples

The following examples illustrate two methods system firmware or software may use to retrieve an MCTP packet from the management controller. The MCTP packet fields are used to identify the message type and which packets belong to a particular message. It is typically the responsibility of the driver to assemble the incoming packets into MCTP messages as required.

8.1.1 Polling Example

In the following example, the system firmware or software polls for MCTP packets from the management controller:

System firmware or software sends *Get MCTP Packet* periodically. If an incoming MCTP packet is available, the management controller will return this packet in the read state of the KCS transaction. If no available incoming MCTP packet is available, a zero byte count packet will be returned by the management controller in the read state of the KCS transaction.

8.1.2 SMS_ATN Example

In this example the system firmware or driver uses the SMS_ATN flag to retrieve MCTP packets from the management controller.

1) The enable MCTP SMS_ATN command is sent to enable MCTP over KCS to use SMS_ATN bit.
2) System firmware or software detects that there is an MCTP packet available from the management controller. This can be done by either periodically checking the SMS_ATN bit , or for interrupt-driven implementations, getting an interrupt when SMS_ATN becomes set.
3) If the KCS interface is being used for IPMI and MCTP, the IPMI command *Get Message Flags* is sent to the management controller. If any IPMI flags are set, IPMI processing of these flags should occur per the IPMI specification. If no IPMI flags are set, it is assumed that an MCTP packet is available.
4) System firmware or software issues a *Get MCTP Packet Datagram message*. This causes an available incoming MCTP packet to be returned by the management controller in the read state of the KCS transaction. If no available incoming MCTP packet is available, a zero byte count packet will be returned by the management controller in the read state of the KCS transaction.

9 MCTP KCS Packet Timing Requirements

The timing specifications shown in Table 10 are specific to MCTP packet transfers on KCS. Timing is specified for a "point-to-point" connection. That is, timing is specified as if there were only two endpoints in direct communication on the bus.

Table 10 – Timing Specifications for MCTP Packets on KCS

Timing Specification	Symbol	Value	Description
Endpoint packet level retries	PN1	8	Number of times a non-bridge endpoint must retry sending an MCTP packet. This also includes bridges when bridges are transmitting as an endpoint (as opposed to a bridge transmitting from its routing functionality).
Bridge packet level retries	PN2	12	Number of times an MCTP bridge (when transmitting packet for routing) must retry sending an MCTP packet.
Packet transaction originator duration	PT1	250 µs per byte	The overall duration shall be less than the specified interval times the number of bytes in the packet, starting from the byte following the slave byte through and including the PEC byte. Individual data byte transmissions may exceed the specification provided the cumulative duration for the packet is met.

10 MCTP KCS Control Message Timing Requirements

The timing specifications in Table 11 are specific to MCTP control messages on KCS. Timing is specified for a "point-to-point" connection. That is, timing is specified as if there were only two endpoints in direct communication on the bus.

Responses are not retried. A "try" or "retry" of a request is defined as a complete transmission of the MCTP control message.

Table 11 – Timing Specifications for MCTP Control Messages on KCS

Timing Specification	Symbol	Min	Max	Description
Endpoint ID reclaim	$T_{RECLAIM}$	5 sec	–	Minimum time that a bus owner must wait before reclaiming the EID for a non-responsive hot-plug endpoint.
Number of request retries	MN1	2	See Description. .	Total of three tries, minimum: the original try plus two retries. The maximum number of retries for a given request is limited by the requirement that all retries must occur within MT4, max of the initial request.
Request-to-response time	MT1	–	120 ms	This interval is measured from the conclusion of the WRITE_END condition of the request to the end of READ_STATE condition of the response.

Timing Specification	Symbol	Min	Max	Description
Time-out waiting for a response	MT2	MT1 max[1] + 6 ms	MT4, min[1]	This interval is measured from the conclusion of the WRITE_END condition of the request to the end of READ_STATE condition of the response.
Time between request retries	MT3	MT1 max + 6 ms	MT4	This interval sets the minimum amount of time that a requester should wait before retrying an MCTP control request. Measured from the conclusion of the WRITE_END condition of the previous request to the WRITE_START condition of the retry.
Instance ID expiration interval	MT4	5 sec [2]	6 sec	Interval after which the instance ID for a given response will expire and become reusable if a response has not been received for the request. This is also the maximum time that a responder tracks an instance ID for a given request from a given requester.

NOTE 1: Unless otherwise specified, this timing applies to the mandatory and optional MCTP commands.

NOTE 2: If a requester is reset, it may produce the same sequence number for a request as one that was previously issued. To guard against this, it is recommended that sequence number expiration be implemented. Any request from a given requester that is received more than MT4 seconds after a previous, matching request should be treated as a new request, not a retry.

ANNEX A
(informative)

Notation and Conventions

Examples of notations used in this document are as follows:

- 2:N — In field descriptions, this will typically be used to represent a range of byte offsets starting from byte two and continuing to and including byte N. The lowest offset is on the left, the highest is on the right.
- (6) — Parentheses around a single number can be used in message field descriptions to indicate a byte field that may be present or absent.
- (3:6) — Parentheses around a field consisting of a range of bytes indicates the entire range may be present or absent. The lowest offset is on the left, the highest is on the right.
- PCIe — Underlined, blue text is typically used to indicate a reference to a document or specification called out in clause 2 or to items hyperlinked within the document.
- rsvd — Abbreviation for “reserved.” Case insensitive.
- [4] — Square brackets around a number are typically used to indicate a bit offset. Bit offsets are given as zero-based values (that is, the least significant bit [LSb] offset = 0).
- [7:5] — A range of bit offsets. The most significant bit is on the left, the least significant bit is on the right.
- `1b` — The lower case "b" following a number consisting of `0`s and `1`s is used to indicate the number is being given in binary format.
- `0x12A` — A leading "`0x`" is used to indicate a number given in hexadecimal format.

ANNEX B
(informative)

Change Log

Version	Date	Description
1.0.0	2010-07-21	Released as DMTF Standard