Document Number: DSP0253

Date: 2010-07-21

Version: 1.0.0

Management Component Transport Protocol (MCTP) Serial Transport Binding

Document Type: Specification

Document Status: DMTF Standard

Document Language: en-US

CONTENTS

Figures

Tables

68

Foreword

The *Management Component Transport Protocol (MCTP) Serial Transport Binding Specification* (DSP0253) was prepared by the PMCI 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 Serial Transport Binding Specification* (DSP0253) defines how the MCTP base protocol and MCTP control commands are delivered over a physical serial transport type and medium.

Management Component Transport Protocol (MCTP) Serial Transport Binding Specification

1 Scope

This document provides the specifications for the Management Component Transport Protocol (MCTP) transport binding for Serial UART 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 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

IETF, RFC1662, *PPP in HDLC-like Framing,* July 1994, http://www.ietf.org/rfc/rfc1662.txt

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

Universal Asynchronous Receiver/Transmitter (UART)

A universal asynchronous receiver/transmitter is a piece of computer hardware that translates data between parallel and serial forms. UARTs are commonly used in conjunction with other communication standards such as EIA RS-232.

4 Symbols and Abbreviated Terms

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

4.1

CRC-16-CCITT

Cyclic Redundancy Check 16 bits – Polynomial is $x^{16} + x^{12} + x^5 + 1$

4.2

UART

Universal Asynchronous Receiver/Transmitter

5 Conventions

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

5.1 Reserved and Unassigned Values

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

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

5.2 Byte Ordering

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

6 MCTP over Serial Transport

6.1 General

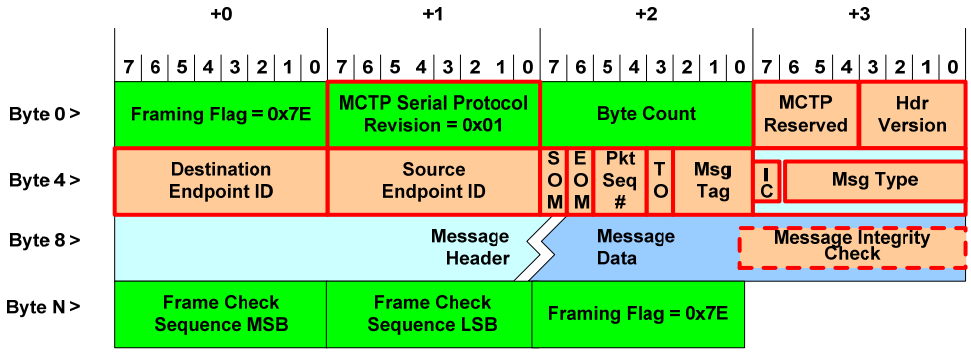
The MCTP over Serial Transport binding defines how MCTP packets are delivered over a physical Serial UART interface using serial transactions. Timing specifications for bus and MCTP control operations are also given.

6.2 MCTP Packet Encapsulation

The MCTP over Serial Transport defines how MCTP Packets are delivered over physical Serial UART using serial transactions. MCTP packet encapsulation for serial shall support the transfer of baseline MTU-sized MCTP packets as specified in DSP0236, which is 64 bytes as of this writing. A 64 byte transmission unit will result in an overall serial packet size of: 64 + 3 (serial header length) + 4 (MCTP transport length) + 3 (serial footer length) = 74 bytes excluding any additional escape characters that are required by framing (see clause 7).

MCTP over Serial packets use the first three bytes to make up the physical serial packet header. The first three fields map to a framing flag, protocol revision, and byte count. Bytes 0–2 and Bytes N–(N + 2) represent the medium specific data. This includes the serial medium specific header (bytes 0–2) and the medium specific trailer (bytes N–(N+2)). Bytes 3–6 in Figure 1 represent fields defined by the Base MCTP Specification and include the MCTP transport header (bytes 3–6). Bytes 7–(N-1) represent the MCTP packet payload which includes the message header and message data.

Table 1 shows the MCTP over Serial Packet header field descriptions.



143

144 **Figure 1 – MCTP over Serial Packet Format**

145 **Table 1 – MCTP over Serial Packet Header Field Descriptions**

Byte	Block Write Field(s)	Description
0	**Framing Flag**	Set to 0x7E to signify the start of a MCTP over Serial packet
1	**MCTP Serial protocol revision**	Set to 0x01 for this initial release. This works with version 0x01 of the MCTP base protocol header defined in DSP0236.
2	**Byte Count**	Byte Count: Byte count for the Serial transaction that is carrying the MCTP packet content. The byte count is the count of bytes that follow the Byte Count field up to, but not including, the Frame check bytes and excluding the escape characters. (See Section 7) 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.)
4	**Data Byte 1**	[7:4] MCTP reserved: This nibble is reserved for definition by DSP0236. [3:0] MCTP header version: Set to 0001b for MCTP devices that are conformant to DSP0236 1.0 and this version of the Serial transport binding. All other values = Reserved.
5	**Data Byte 2**	Destination Endpoint ID (*)
6	**Data Byte 3**	Source Endpoint ID (*)
7	**Data Byte 4**	[7] SOM: Start Of Message flag. (*) [6] EOM: End Of Message flag. (*) [5:4] Packet Sequence Number (*) [3] Tag Owner (TO) bit (*) [2:0] Message Tag (*)
8	**Data Byte 5**	Message Type: (*)

Byte	Block Write Field(s)	Description
9:N-1	**Data Bytes 6:M**	Message Header and Data: (*)
N	**Frame Check Sequence MSB**	CRC 16 CCITT MSB – CRC over bytes 0 – N-1
N+1	**Frame Check Sequence LSB**	CRC 16 CCITT LSB
N+2	**Framing Flag**	Set to 0x7E to signify the end of a Serial over MCTP frame
(*) Indicates a field that is defined by the DSP0236.		

7 Packet Framing and Encapsulation

Serial data is required to have 8 bits of data per byte however parity and stop bit definition are defined by the implementation. Typical implementations use no parity and one stop bit (that is, N-8-1). Software flow control (XON / XOFF) is not supported however hardware flow control may be used. Serial data transmission is required to be full duplex.

Packet framing and encapsulation is a variant of PPP encapsulation. All messages begin and end with a Flag byte of 0x7E.

A properly framed packet has the following properties:

- First Byte is Flag = 0x7E
- Packet size = Byte Count + 4 (Revision field and Byte count field = 2 + Frame Check Bytes = 2) bytes + variable number of escape characters
- Last Byte is Flag = 0x7E

If these conditions are not met upon reception, the packet is silently discarded.

Software uses a flag byte (0x7E) as a start of frame token. Due to this fact if a data byte of 0x7E is encountered it is replaced with a sequence of 0x7D 0x5E. The0x7D character is known as the Escape character. If an Escape character occurs in the data, the sender replaces it with the escape sequence 0x7D 0x5D. The receiver translates 0x7D 0x5E escape sequences back to 0x7E and 0x7D 0x5D escape sequences back to 0x7D.

It is recommended that this serial interface is dedicated for MCTP Serial transfer from the host to the management controller. If other serial data protocols are shared on this physical interface it is up to the implementer to distinguish multiple protocols and subsequent error conditions. It is out of the scope of this specification to define any protocol sharing algorithms.

7.1 Processing Outgoing Packet

The following steps occur before the requester sends out an outgoing packet:

1) FCS computation is calculated on the packet excluding the frame check bytes
2) Escape processing is performed on the packet before transmitting. Note that the byte count field is calculated before the escape sequences are inserted.

7.2 Processing Incoming Packet

The following steps occur after the incoming packet is received:

175 1) Control escape sequences are replaced with actual data bytes
176 2) FCS validation is performed on the modified packet.
177 3) On FCS error the packet is silently discarded.

178 8 Data Integrity

179 A Frame Check Sequence is calculated as a CRC-16-CCITT over the packet to check for bit level errors
180 in the packet. The polynomial is $x^{16} + x^{12} + x^5 + 1$ and is calculated over byte 1 (Serial protocol revision)
181 through the last byte of the data. The framing bytes are not included in the CRC calculation. A description
182 and example code for the CRC can be found in RFC1662 Appendix C. If the receiver's calculation of the
183 checksum does not match the transmitted checksum, the frame is silently discarded.

184 9 MCTP Packet Timing Requirements

185 The timing specifications shown in Table 2 are specific to MCTP packet transfers on Serial. Timing is
186 specified for a "point-to-point" connection. That is, timing is specified as if there were only two endpoints
187 in direct communication on the bus.

188 If the requester detects a break condition on the line, the requester will abort the transmission, flush the
189 transmit buffer and perform a retry of the request. If the responder detects a break condition on the line,
190 the responder will flush the receive buffer and wait for a new request.

191 **Table 2 – Timing Specifications for MCTP Packets on Serial**

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.

192 10 MCTP Control Message Timing Requirements

193 The timing specifications in Table 3 are specific to MCTP control messages on Serial. Timing is specified
194 for a "point-to-point" connection. That is, timing is specified as if there were only two endpoints in direct
195 communication on the bus.

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

198 **Table 3 – Timing Specifications for MCTP Control Messages on Serial**

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 requirment that all retries must occur within MT4, max of the initial request.
Request-to-response time	MT1	–	100 ms	This interval is measured at the responder from the end of the reception of the MCTP Control Protocol request to the beginning of the transmission of the response. This requirement is tested under the condition where the responder can successfully transmit the response on the first try.
Time-out waiting for a response	MT2	MT1 max+ 2*MT3 max	MT4, min[1]	This interval is measured at the requester from the end of the successful transmission of the MCTP Control Protocol request to the beginning of the reception of the corresponding MCTP Control Protocol response. This interval at the requester sets the minimum amount of time that a requester should wait before retrying an MCTP Control Protocol request. NOTE: This specification does not preclude an implementation from adjusting the minimum time-out waiting for a response to a smaller number than MT2 based on measured response times from responders. The mechanism for doing so is outside the scope of this specification.
Transmission Delay	MT3	–	100 ms	Time to take into account transmission delay of an MCTP Control Protocol Message. Measured as the time between the end of the transmission of an MCTP Control Protocol message at the transmitter to the beginning of the reception of the MCTP Control Protocol message at the receiver.

Timing Specification	Symbol	Min	Max	Description
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)

Notations

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