Specification for MIDI over Bluetooth Low Energy (BLE-MIDI)

Version 1.0
November 1 2015

Published By:
The MIDI Manufacturers Association
Los Angeles, CA

PREFACE

Apple introduced support for MIDI over Bluetooth Low Energy in iOS 8 and OS X 10.10 in the summer of 2014 and added features in 2015. A number of products have already been introduced for sale supporting Apple's protocol. The following specification from MMA adopts Apple's implementation as of June 2015 in order to avoid market fragmentation and allow adoption of a BLE MIDI industry standard by MMA member companies.

This specification does not support multiple virtual cables, clock synchronization, or all possible and valid MIDI 1.0 data streams, which might be addressed by an update or second specification, along with improvements in timing resolution and jitter reduction.

MMA/AMEI RP-052

Printed 2015

MMA
POB 3173
La Habra CA 90632-3173 USA

Table of Contents

1. Introduction

This specification outlines a method for encoding and decoding Musical Instrument Digital Interface (MIDI) data for transmission over Bluetooth Low Energy (BLE) connections.

In this document, the terms Bluetooth, Bluetooth LE, and BLE are used interchangeably to refer to Bluetooth low energy technology, as defined in the Bluetooth Core Specification Version 4.0 (dated 30 June 2010). For more information, please visit the Bluetooth Special Interest Group website at http://www.bluetooth.org.

2. Performance

In this protocol, MIDI messages are prepended with timestamps, buffered, and accessed via a BLE characteristic. It is important to note that the frequency with which data can be transmitted (determined by the connection interval) places a lower bound on latency. Millisecond-resolution timestamps allow for jitter comparable to the USB Class Specification for MIDI Devices version 1.0. As Bluetooth technology develops and the usable connection intervals are lowered, latency will improve correspondingly. However, delays in wireless packet delivery may occur unexpectedly at any time, temporarily increasing latency and jitter.

3. BLE Service and Characteristics Definitions

The following service and characteristic are defined:

- MIDI Service (UUID: 03B80E5A-EDE8-4B33-A751-6CE34EC4C700)
- MIDI Data I/O Characteristic (UUID: 7772E5DB-3868-4112-A1A9-F2669D106BF3)
    - write (encryption recommended, write without response is required)
    - read (encryption recommended, respond with no payload)
    - notify (encryption recommended)

4. Connection Interval

The BLE MIDI device must request a connection interval of 15 ms or less. A lower connection interval is preferred in most applications of MIDI. Connection should be established at the lowest connection interval that is currently supported on both the Central and the Peripheral.

5. Initial Connection and Pairing

The Central will attempt to read the MIDI I/O characteristic of the Peripheral after establishing a connection with the accessory. The accessory shall respond to the initial MIDI I/O characteristic read with a packet that has no payload.

6. Maximum Transmission Unit Negotiation

The accessory must support MTU negotiation and must support the MTU Exchange command.

7. Packet Encoding

Unlike legacy MIDI, BLE is a packet based protocol. Incoming messages cannot be instantly forwarded to the receiving party. Instead they must be buffered and transmitted each BLE connection interval, which is negotiated between the sender and receiver. To maintain precise inter-event timing, this protocol uses 13-bit millisecond-resolution timestamps to express the render time and event spacing of MIDI messages.

In transmitting MIDI data over Bluetooth, a series of MIDI messages of various sizes must be encoded into packets no larger than the negotiated MTU minus 3 bytes (typically 20 bytes or larger.)

The first byte of all BLE packets must be a *header byte.* This is followed by *timestamp bytes* and MIDI messages.

Header Byte

bit 7	Set to 1.
bit 6	Set to 0. (Reserved for future use)
bits 5-0	*timestampHigh*:Most significant 6 bits of timestamp information.

The header byte contains the topmost 6 bits of timing information for MIDI events in the BLE packet. The remaining 7 bits of timing information for individual MIDI messages encoded in a packet is expressed by timestamp bytes. Timestamps are discussed in detail in a later section.

Timestamp Byte

bit 7	Set to 1.
bits 6-0	*timestampLow*: Least Significant 7 bits of timestamp information.

The 13-bit timestamp for the first MIDI message in a packet is calculated using 6 bits from the header byte and 7 bits from the timestamp byte.

Timestamps are discussed in detail in a later section.

MIDI Messages

The general form of a MIDI message follows:

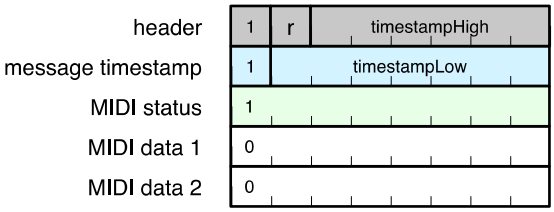
n-byte MIDI Message

Byte 0	MIDI message Status byte, Bit 7 is Set to 1.
Bytes 1 to n-1	MIDI message Data bytes, if n > 1. Bit 7 is Set to 0

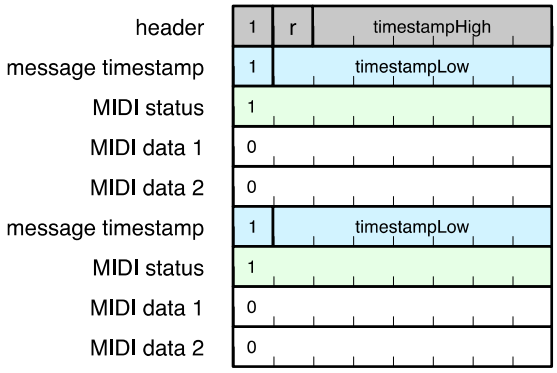
There are two types of MIDI messages that can appear in a single packet: full MIDI messages and Running Status MIDI messages. Each is encoded differently.

A *full MIDI message* is simply the MIDI message with the Status byte included.

BLE Packet with One MIDI Message



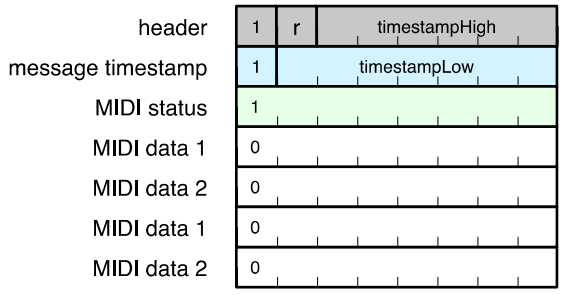
BLE Packet with Two MIDI Messages



A *Running Status MIDI message* is a MIDI message with the Status byte omitted. Running Status MIDI messages may only be placed in the data stream if the following criteria are met:

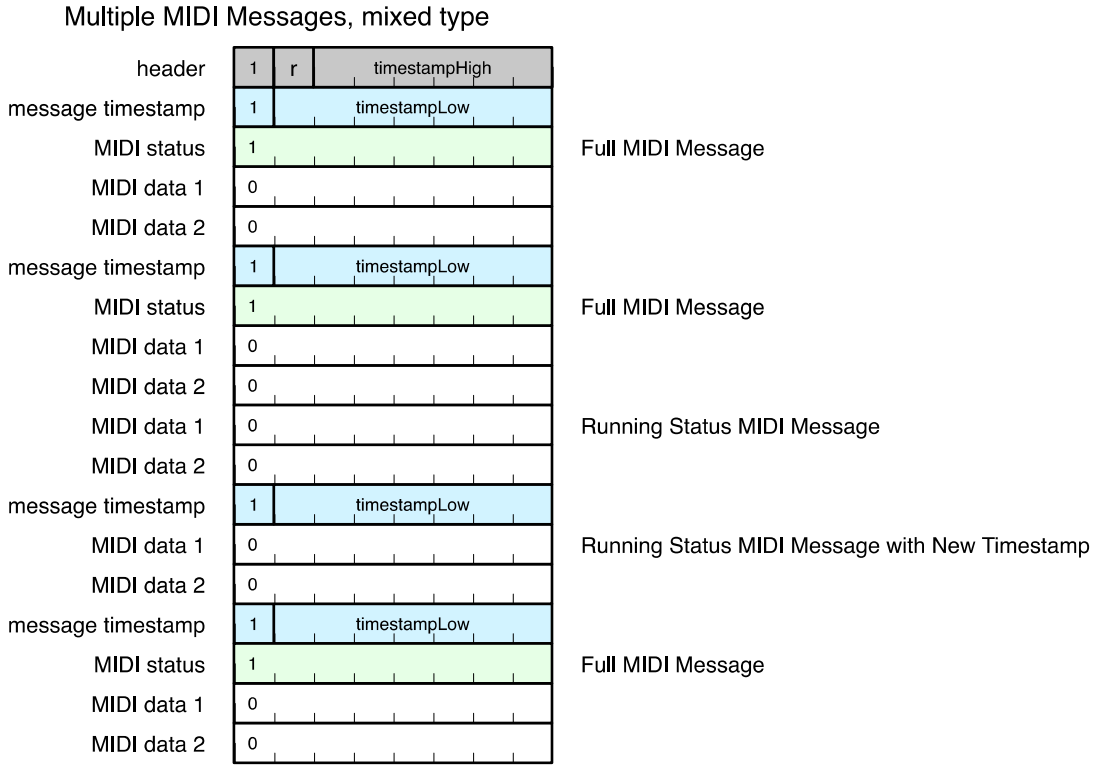
1. The original MIDI message is 2 bytes or greater and is not a System Common or System Real-Time message.
2. The omitted Status byte matches the most recently preceding *full MIDI message*'s Status byte within the same BLE packet.

2 MIDI Messages with Running Status

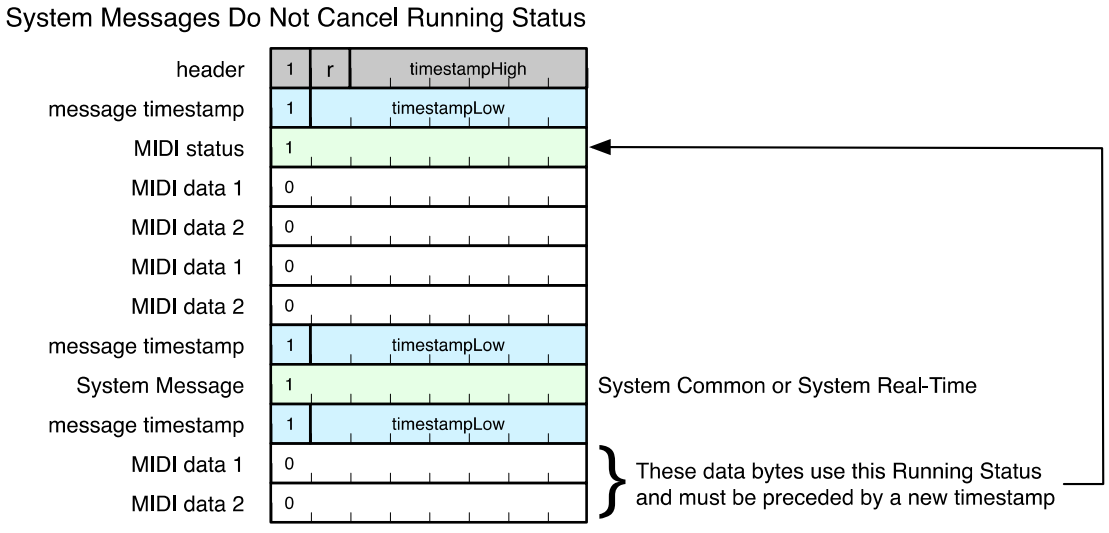


In addition, the following rules apply with respect to Running Status:

1. A Running Status MIDI message is allowed within the packet after at least one full MIDI message.
2. Every MIDI Status byte must be preceded by a timestamp byte. Running Status MIDI messages may be preceded by a timestamp byte. If a Running Status MIDI message is not preceded by a timestamp byte, the timestamp byte of the most recently preceding message in the same packet is used.

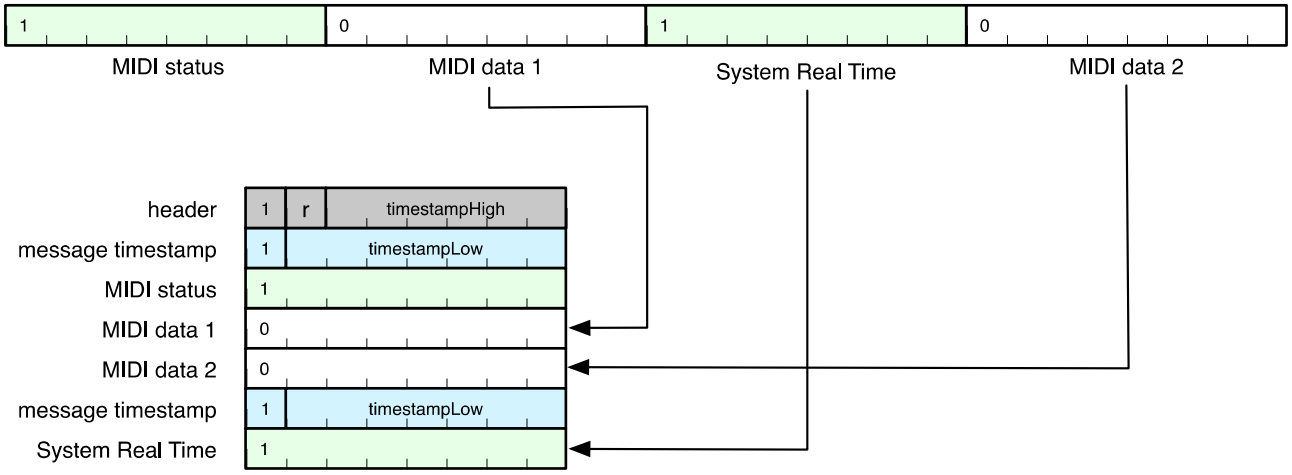


3. System Common and System Real-Time messages do not cancel Running Status if interspersed between Running Status MIDI messages. However, a timestamp byte must precede the Running Status MIDI message that follows.
4. The end of a BLE packet does cancel Running Status.



In the MIDI 1.0 protocol, System Real-Time messages can be sent at any time and may be inserted anywhere in a MIDI data stream, including between Status and Data bytes of any other MIDI messages. In the MIDI BLE protocol, the System Real-Time messages must be deinterleaved from other messages – except for System Exclusive messages.

MIDI Stream with System Real-Time Message in the middle of another MIDI Message

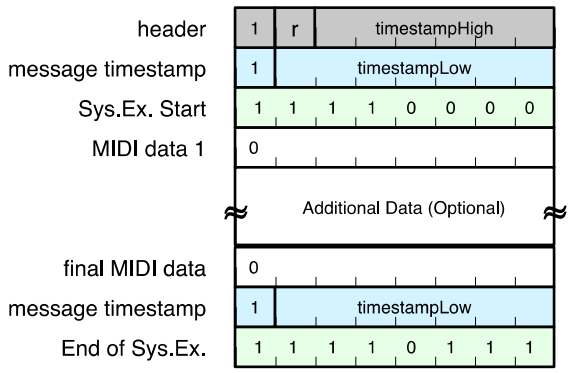


8. Multiple Packet Encoding (SysEx Only)

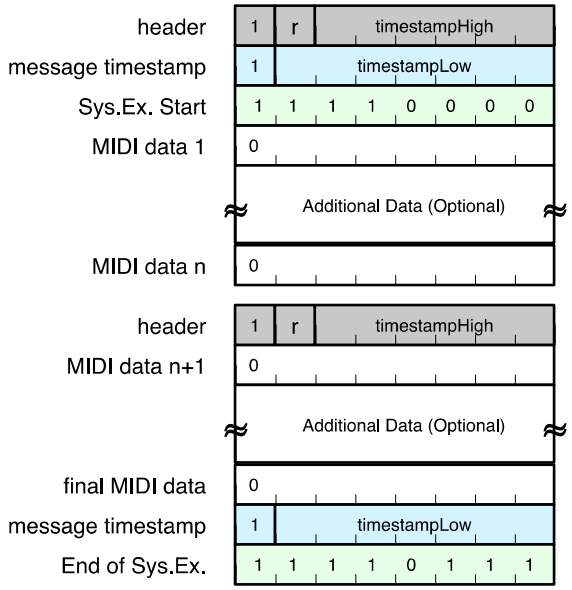
Only a SysEx (System Exclusive) message may span multiple BLE packets and is encoded as follows:

1. The SysEx start byte, which is a MIDI Status byte, is preceded by a timestamp byte.
2. Following the SysEx start byte, any number of Data bytes (up to the number of the remaining bytes in the packet) may be written.
3. Any remaining data may be sent in one or more *SysEx continuation packets*. A SysEx continuation packet begins with a header byte but does not contain a timestamp byte. It then contains one or more bytes of the SysEx data, up to the maximum packet length. This lack of a timestamp byte serves as a signal to the decoder of a SysEx continuation.
4. System Real-Time messages may appear at any point inside a SysEx message and must be preceded by a timestamp byte.
5. SysEx continuations for unterminated SysEx messages must follow either the packet's header byte or a real-time byte.
6. Continue sending SysEx continuation packets until the entire message is transmitted.
7. In the last packet containing SysEx data, precede the EOX message (SysEx end byte), which is a MIDI Status byte, with a timestamp byte.

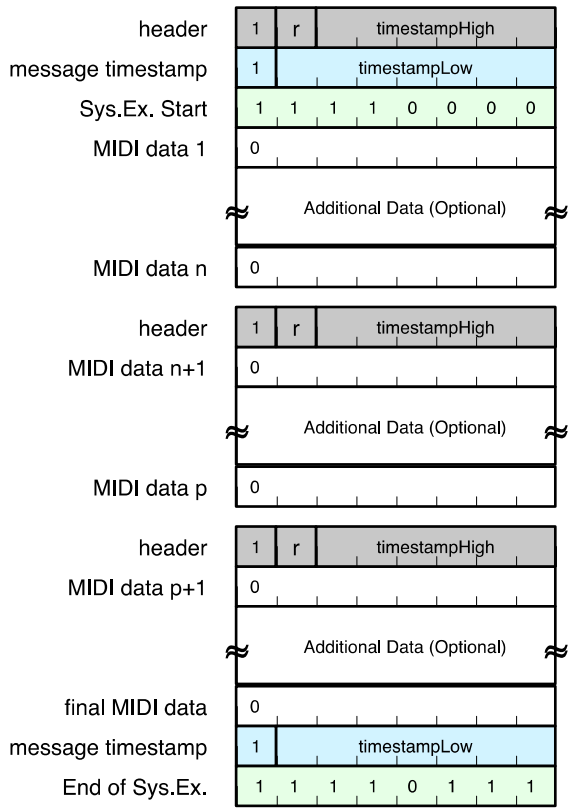
System Exclusive Start & End in 1 Packet



System Exclusive Split Across 2 Packets



System Exclusive Split Across 3 Packets



Once a SysEx transfer has begun, only System Real-Time messages are allowed to precede its completion as follows:

1. A System Real-Time message interrupting a yet unterminated SysEx message must be preceded by its own timestamp byte.
2. SysEx continuations for unterminated SysEx messages must follow either the packet's header byte or a real-time byte.

9. Bluetooth LE MIDI Timestamps

Timestamps are 13-bit values in milliseconds, and therefore the maximum value is 8,191 ms. Timestamps must be issued by the sender in a monotonically increasing fashion.

The 13-bit timestamp for a MIDI message is composed of two parts, a timestampHigh containing the most significant 6 bits and a timestampLow containing the least significant 7 bits. The

timestampHigh is initially set using the lower 6 bits from the header byte while the timestampLow is formed of the lower 7 bits from the timestamp byte. Should the timestamp value of a subsequent MIDI message in the same packet overflow/wrap (i.e., the timestampLow is smaller than a preceding timestampLow), the receiver is responsible for tracking this by incrementing the timestampHigh by one (the incremented value is not transmitted, only understood as a result of the overflow condition).

In practice, the time difference between MIDI messages in the same BLE packet should not span more than twice the connection interval. As a result, a maximum of one overflow/wrap may occur per BLE packet.

Timestamps are in the sender's clock domain and are not allowed to be scheduled in the future.

Correlation between the receiver's clock and the received timestamps must be performed to ensure accurate rendering of MIDI messages, and is not addressed in this document.