

MOBILEFISH.COM

HomeShopContactEmulatorsTutorialsDeveloperServicesMobileGamesLoginLegal

Tutorials

3D Studio Max

Abbreviations

ADSL installation

Advertising programs for publishers

BitComet

BTW

Character Encoding Quick Guide

Chinese

Cascading Style Sheet Quick Guide

Top Level Domain Country Codes

Language Codes

Show more ▼

Print this page

MIDI

The Musical Instrument Digital Interface (MIDI) protocol is an industry-standard defined in 1982 to represent musical information.

It is used in electronic instruments (keyboards, synthesizers, drum machines, sound cards), computer applications which produce sound, such as multimedia presentations, computer games and others.

MIDI does not transmit an audio signal or media, it transmits "event messages" such as the pitch and intensity of musical notes to play, control signals for parameters such as volume, vibrato and panning, cues, and clock signals to set the tempo. To create monophonic MIDI files checkout our [online MIDI maker](#).

More information about MIDI can be found at: <http://www.midi.org/>

Quick guides

[MIDI specification](#)

MIDI specification.

Information

A MIDI file (file extension: .mid) consists of a stream of 8-bit bytes. All 16-bit and 32-bit quantities are constructed by reading in two or four 8-bit bytes, respectively. The bytes are joined together in big-endian order.

In this tutorial, we use notation u1, u2, and u4 to mean an unsigned one-, two-, or four-byte quantity, respectively. uN means a variable number of bytes.

An example of a MIDI file created with the [online morse code generator](#):

play

Download MIDI file: morse_code_a.mid

If you download this MIDI file (morse_code_a.mid) and open it with a [HEX editor](#), you will see the following:

| | | | | | | | | | | | | | | | | |
|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------------------|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
| 00000000h: | 4D | 54 | 68 | 64 | 00 | 00 | 00 | 06 | 00 | 00 | 01 | 01 | E0 | 4D | 54 | ; MThd.....àMT |
| 00000010h: | 72 | 6B | 00 | 00 | 00 | 86 | 00 | FF | 03 | 35 | 47 | 65 | 6E | 65 | 72 | 61 ; rk...f.ÿ.SGenera |
| 00000020h: | 74 | 65 | 64 | 20 | 6D | 6F | 72 | 73 | 65 | 20 | 63 | 6F | 64 | 65 | 20 | 77 ; ted morse code w |
| 00000030h: | 77 | 77 | 2E | 6D | 6F | 62 | 69 | 6C | 65 | 66 | 69 | 73 | 68 | 2E | 63 | 6F ; ww.mobilefish.co |
| 00000040h: | 6D | 2F | 67 | 6F | 2F | 6D | 6F | 72 | 73 | 65 | 5F | 63 | 6F | 64 | 65 | 00 ; m/go/morse_code. |
| 00000050h: | FF | 02 | 17 | 28 | 43 | 29 | 20 | 32 | 30 | 31 | 30 | 20 | 4D | 6F | 62 | 69 ; ÿ..(C) 2010 Mobi |
| 00000060h: | 6C | 65 | 66 | 69 | 73 | 68 | 2E | 63 | 6F | 6D | 00 | FF | 58 | 04 | 04 | 02 ; lefish.com.ÿX... |
| 00000070h: | 18 | 08 | 00 | FF | 51 | 03 | 08 | 7A | 23 | 00 | C0 | 4F | 81 | 70 | 90 | 51 ; ...ÿQ..z#.ÀO p Q |
| 00000080h: | 7F | 3A | 80 | 51 | 00 | 3A | 90 | 51 | 7F | 81 | 2E | 80 | 51 | 00 | 82 | 2C ; :EQ.: Q .eQ,,, |
| 00000090h: | 90 | 51 | 00 | 82 | 2C | 80 | 51 | 00 | 00 | FF | 2F | 00 | | | | ; Q,,,eQ..ÿ/. |

A more detailed explanation about this MIDI file can be found in Figure 1 and Table 1.

Figure 1: Delta time and velocity.

velocity

See Table 1 for explanantion A,B,C,D,E and F

7F

A

B

C

D

E

F

0

81 70

3A 3A

81 2E

82 2C

82 2C

delta time (hex)

Table 1: The MIDI file morse_code_a.mid dissected.

| Dec Range [Hex Range] | Hex value | Description |
|-----------------------|-------------|-------------------------------|
| 0-3 [00-03] | 4D 54 68 64 | MIDI header: MThd |
| 4-7 [04-07] | 00 00 00 06 | MIDI header length, always: 6 |
| 8-9 [08-09] | 00 00 | Single multi-channel track |
| 10-11 [0A-0B] | 00 01 | Number of tracks: 1 |
| 12-13 [0C-0D] | 01 E0 | Time-code-based time |
| 14-17 [0E-11] | 4D 54 72 6B | Track header: MTrk |

https://www.mobilefish.com/tutorials/midi/midi_quickguide_specification.html

1/8

| | | |
|--------------------|----------------|---|
| 18-21 [12-15] | 00 00 00 86 | Track header length: 86 = 134 dec |
| 22-25 [16-19] | 00 FF 03 35 | 00 = delta time FF 03 = Track name 35 = String length total 53 bytes |
| 26-78 [1A-4E] | 47 65 | Note: Max: 256 bytes and ISO-8859-1 character set allowed. Track name = "Generated morse code www.mobilefish.com/services/morse_code/morse_code.php " |
| 79-82 [4F-52] | 00 FF 02 17 | 00 = delta time FF 02 = Copyright Notice 17 = String length total 23 bytes |
| 83-105 [53-69] | 28 6D | Note: Max: 256 bytes and ISO-8859-1 character set allowed. Copyright Notice = "(C) 2010 Mobilefish.com" |
| 106-109 [6A-6D] | 00 FF 58 04 | 00 = delta time FF 58 = Time Signature 04 = 4 data bytes |
| 110-113 [6E-71] | 04 02 18 08 | 04 = upper value of time signature 02 = lower value of time signature 18 = number of MIDI clocks in a metronome clock |
| 114-117 [72-75] | 00 FF 51 03 | 00 = delta time FF 51 = Set Tempo in microseconds per quarter note 03 = 3 data bytes |
| 118-120 [76-78] | 08 7A 23 | 0x087A23 = 555555 dec Tempo = 555555 microseconds per quarter note Tempo = 60000000 / 555555 = 108 beats per minute per quarter note |
| 121-123 [79-7B] | 00 C0 4F | 00 = delta time C0 = Select instrument channel 0 4F = Instrument no 79 |
| 124-128 [7C-80] | 81 70 90 51 7F | 81 70 = delta time (240 decimal, see Figure 1: A) 90 = Note on channel 0 51 = note number (More info: Online MIDI and music score maker) 7F = velocity |
| 129-132 [81-84] | 3A 80 51 00 | 3A = delta time (58 decimal, see Figure 1: B) 80 = Note off channel 0 51 = note number (More info: Online MIDI and music score maker) 00 = velocity |
| 133-136 [85-88] | 3A 90 51 7F | 3A = delta time (58 decimal, see Figure 1: C) 90 = Note on channel 0 51 = note number (More info: Online MIDI and music score maker) 7F = velocity |
| 137-141 [89-8D] | 81 2E 80 51 00 | 81 2E = delta time (174 decimal, see Figure 1: D) 80 = Note off channel 0 51 = note number (More info: Online MIDI and music score maker) 00 = velocity |
| 142-146 [8E-92] | 82 2C 90 51 00 | 82 2C = delta time (300 decimal, see Figure 1: E) 90 = Note on channel 0 51 = note number (More info: Online MIDI and music score maker) 00 = velocity |
| 147-151 [93-97] | 82 2C 80 51 00 | 82 2C = delta time (300 decimal, see Figure 1: F) 80 = Note off channel 0 51 = note number (More info: Online MIDI and music score maker) 00 = velocity |
| 152-155 [98-9B] | 00 FF 2F 00 | 00 = delta time FF 2F 00 = End Of Track |

A MIDI file is organized into data chunks and its structure looks like (see also [Table 2](#)):

```
MIDI file
{
  u4 MIDI header or <chunk type>
  u4 MIDI header length or <length>
  u2 MIDI format or <format>
  u2 Number of tracks or <ntrks>
  u2 Time division or <division>
  u4 Track header or <chunk type>
  u4 Track header length or <length>
  Track event or <mtrk event>+

  uN Delta time or <delta-time>
  uN Event or <event>
}
```

Table 2: MIDI file

| # of bytes | Description |
|------------|---|
| | MIDI header or <chunk type> byte range: 0..3 |
| u4 | The MIDI header must have the value MThd represented by the following hex values: <div>4D546864</div> |
| | It is used to identify this file as conforming to the MIDI specification. |
| u4 | MIDI header length or <length> |

byte range: 4..7

For Standard MIDI files (SMF) this value is always 6, represented by the following hex values:

00 00 00 06

It is the number of bytes used by "MIDI format (u2)", "Number of tracks in the MIDI file (u2)" and "The speed of the music (time division) (u2)".
Thus u2+u2+u2=u6.

MIDI format or <format>

byte range: 8..9

The MIDI format describes how the following track information is to be interpreted.

Table 3: MIDI format

| Format | Hex value | Description |
|--------|-----------|---|
| | | The MIDI file contains a single multi-channel track. |
| 0 | 00 00 | All the data is put into only one track (not to be confused with a channel) in an uninterrupted data stream in the file, where all the bytes are next to each other. If you have a simple bass-line and a melody, in format 0, you would alternately encode notes of the bass-line and melody next to each other. |
| u2 | | The MIDI file contains one or more simultaneous tracks (or MIDI outputs) of a sequence. |
| 1 | 00 01 | In format 1 the channel data can be compartmentalized into 1 or more tracks (up to 65535). If you have a simple bass-line and a melody, in format 1, all the melody notes would go into one track, and all the bass notes would go into another track. Also in format 1, you can imitate the format 0 interleaving style in any of the tracks. |
| 2 | 00 02 | The MIDI file contains one or more sequentially independant single-track patterns. A format 2 MIDI file is a sort of a combination of the other two formats. It contains multiple tracks, but each track represents a different sequence which may not necessarily be played simultaneously. This is meant to be used to save drum patterns, or other multi-pattern music sequences. |

Number of tracks or <ntrks>

byte range: 10..11

The number of tracks ranges from: 1 - 65535, represented by the following hex values:

00 01 - FF FF

If MIDI format is 0, the number of tracks is always 1.

Time division or <division>

byte range: 12..13

The time division is used to decode the track event delta times into "real" time.
It has two formats:

• metrical time

Determines the beats (or ticks) per quarter note.

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 <bitno.
0

- Where bit number 0-14 (green) represent the delta time "beats or ticks" which make up a quarter note.
- Where bit number 15 (red) is always 0.

For example:
0x0080 means:
128 beats for a 1/4 note.
64 beats for a 1/8 note.
32 beats for a 1/16 note.
16 beats for a 1/32 note.
256 beats for a 1/2 note.
512 beats for a whole note.

0x0050 means:
80 beats for a 1/4 note.
40 beats for a 1/8 note.
20 beats for a 1/16 note.
10 beats for a 1/32 note.
160 beats for a 1/2 note.
320 beats for a whole note.

• time-code-based time

Determines the number of frames per second SMPTE time and the number of beats (or ticks) per frame.

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 <bitno.
1

- Where bit number 0-7 (blue) represents beats (or ticks) per frame.
- Where bit number 8-14 (green) represents the frames per second SMPTE time. Allowed values: -24, -25, -29 or -30.

The negative values are stored in two complements form.

If value is -24, it represents 24 frames per second.
 If value is -25, it represents 25 frames per second.
 If value is -29, it represents 29 frames per second.
 If value is -30, it represents 30 frames per second.

- Where bit number 15 (red) is always 1.

For example:

0xE878 means:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 <bitno.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Bit number 0-7 (blue) represents 120 beats (or ticks) per frame.

Bit number 8-14 (green) represents 24 frames per second SMPTE time.

The value 24 is calculated as follow:

Step 1:

11101000 Two complements form. Bit number 8-15.

00010111 Inverted two complements form.

Step 2:

00010111 Inverted two complements form.

00000001 Add 1

Step 3:

00011000 Represents value 24.

For example:

0xE764 means:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 <bitno.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Bit number 0-7 (blue) represents 100 beats (or ticks) per frame.

Bit number 8-14 (green) represents 25 frames per second SMPTE time.

For example:

0xE350 means:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 <bitno.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Bit number 0-7 (blue) represents 50 beats (or ticks) per frame.

Bit number 8-14 (green) represents 29 frames per second SMPTE time.

For example:

0xE250 means:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 <bitno.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Bit number 0-7 (blue) represents 50 beats (or ticks) per frame.

Bit number 8-14 (green) represents 30 frames per second SMPTE time.

Track header

byte range: 14..17

- u4 The track header must have the value **MTrk** and marks the start of the track event, where the actual song data are stored. The track header is represented by the following hex values:

| | | | |
|----|----|----|----|
| 4D | 54 | 72 | 6B |
|----|----|----|----|

Track header length

byte range: 18..21

- u4 This represents the total number of bytes used in the track events.

Delta time

u1 ..
u4 byte range: variable

Each <mtrk event> starts with a delta-time.

This represents the amount of time before the following event:

<delta-time><mtrk event><delta-time><mtrk event>....

An event can be a message to play or stop a note, to change the instrument, etc.

All events always starts with a delta-time even if the delta-time is zero (=0x00).

A delta-time is stored in at least 1 byte and maximum 4 bytes. Because the delta-time can be stored in a variable number of bytes, bit no. 7 of each byte has a special use. If this bit is zero then this byte is the last byte of the series. In all of the preceding bytes, bit no. 7 has value 1.

7 6 5 4 3 2 1 0 <bitno.

| | | | | | | | |
|--|--|--|--|--|--|--|--|
| | | | | | | | |
|--|--|--|--|--|--|--|--|

For example:

Delta-time = **0xFF7F** means:

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 <bitno.

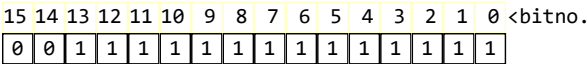
| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Bit number 7 must be 0 because it is the last byte of the series.

Bit number 15 must be 1 because it is NOT the last byte of the series.

To determine what the actual value delta-time=0xFF7F represent, do the following:

If bit number 7 and bit number 15 are "emptied" and the most significant byte is shifted 1 bit to the right, the two bytes looks like:



This represents actual hex value = 3FFF.

So, if a delta-time is between 0-127, it can be represented as one byte. The largest delta-time allowed is 0FFFFFFF, which translates to 4 bytes size. Here are examples of delta-times as 4 bytes values, and the variable length quantities that they translate to:

| Table 4: Delta-times | | | |
|----------------------|-----------------------------|----------------------|------------|
| Delta-time value | Represents actual hex value | Represents dec value | Bytes used |
| 00 | 00000000 | 0 | 1 |
| 3A | 0000003A | 58 | 1 |
| 40 | 00000040 | 64 | 1 |
| 7F | 0000007F | 127 | 1 |
| 81 00 | 00000080 | 128 | 2 |
| 81 2E | 000000AE | 174 | 2 |
| 81 70 | 000000F0 | 240 | 2 |
| 82 2C | 0000012C | 300 | 2 |
| C0 00 | 00002000 | 8192 | 2 |
| FF 7F | 00003FFF | 16383 | 2 |
| 81 80 00 | 00004000 | 16384 | 3 |
| C0 80 00 | 00100000 | 1048576 | 3 |
| FF FF 7F | 001FFFFFFF | 2097151 | 3 |
| 81 80 80 00 | 00200000 | 2097152 | 4 |
| C0 80 80 00 | 08000000 | 134217728 | 4 |
| FF FF FF 7F | 0FFFFFFF | 268435455 | 4 |

u3 **Event**
byte range: variable

There are three types of events:

- [MIDI Control Events or MIDI Channel Events or <MIDI event>](#)
- [System Exclusive Events or <sysex event>](#)
- [Meta Events or <meta-event>](#)

Table 5: MIDI Control Events or MIDI Channel Events or <MIDI event>

| Message description | Status byte (hex) | Data byte 1 (hex) | Data byte 2 (hex) |
|-------------------------------|-------------------|-------------------|-------------------|
| Note Off | 1000nnnn (8n) | 0kkkkkkk | 0vvvvvvv |
| Note On | 1001nnnn (9n) | 0kkkkkkk | 0vvvvvvv |
| Key Pressure or Aftertouch | 1010nnnn (An) | 0kkkkkkk | 0ppppppp |
| Bank Select | 1011nnnn (Bn) | 00000000 (00) | 0ttttttt |
| Modulation Wheel | 1011nnnn (Bn) | 00000001 (01) | 0ttttttt |
| Breath Controller | 1011nnnn (Bn) | 00000010 (02) | 0ttttttt |
| Foot Controller | 1011nnnn (Bn) | 00000100 (04) | 0ttttttt |
| Portamento Time | 1011nnnn (Bn) | 00000101 (05) | 0ttttttt |
| Data Entry Slider (MSB) | 1011nnnn (Bn) | 00000110 (06) | 0ttttttt |
| Main Volume | 1011nnnn (Bn) | 00000111 (07) | 0ttttttt |
| Balance | 1011nnnn (Bn) | 00001000 (08) | 0ttttttt |
| Pan | 1011nnnn (Bn) | 00001010 (0A) | 0ttttttt |
| Expression Controller | 1011nnnn (Bn) | 00001011 (0B) | 0ttttttt |
| Effect Control 1 | 1011nnnn (Bn) | 00001100 (0C) | 0ttttttt |
| Effect Control 2 | 1011nnnn (Bn) | 00001101 (0D) | 0ttttttt |
| General-Purpose Controllers 1 | 1011nnnn (Bn) | 00010000 (10) | 0ttttttt |
| General-Purpose Controllers 2 | 1011nnnn (Bn) | 00010001 (11) | 0ttttttt |
| General-Purpose Controllers 3 | 1011nnnn (Bn) | 00010010 (12) | 0ttttttt |
| General-Purpose Controllers 4 | 1011nnnn (Bn) | 00010011 (13) | 0ttttttt |
| LSB for controllers 0 | 1011nnnn (Bn) | 00100000 (20) | 0ttttttt |
| LSB for controllers 1 | 1011nnnn (Bn) | 00100001 (21) | 0ttttttt |
| LSB for controllers 2 | 1011nnnn (Bn) | 00100010 (22) | 0ttttttt |
| LSB for controllers 3 | 1011nnnn (Bn) | 00100011 (23) | 0ttttttt |
| LSB for controllers 4 | 1011nnnn (Bn) | 00100100 (24) | 0ttttttt |
| LSB for controllers 5 | 1011nnnn (Bn) | 00100101 (25) | 0ttttttt |
| LSB for controllers 6 | 1011nnnn (Bn) | 00100110 (26) | 0ttttttt |
| LSB for controllers 7 | 1011nnnn (Bn) | 00100111 (27) | 0ttttttt |
| LSB for controllers 8 | 1011nnnn (Bn) | 00101000 (28) | 0ttttttt |
| LSB for controllers 9 | 1011nnnn (Bn) | 00101001 (29) | 0ttttttt |

| | | | |
|---|---------------|---------------|----------|
| LSB for controllers 10 | 1011nnnn (Bn) | 00101010 (2A) | 0ttttttt |
| LSB for controllers 11 | 1011nnnn (Bn) | 00101011 (2B) | 0ttttttt |
| LSB for controllers 12 | 1011nnnn (Bn) | 00101100 (2C) | 0ttttttt |
| LSB for controllers 13 | 1011nnnn (Bn) | 00101101 (2D) | 0ttttttt |
| LSB for controllers 14 | 1011nnnn (Bn) | 00101110 (2E) | 0ttttttt |
| LSB for controllers 15 | 1011nnnn (Bn) | 00101111 (2F) | 0ttttttt |
| LSB for controllers 16 | 1011nnnn (Bn) | 00110000 (30) | 0ttttttt |
| LSB for controllers 17 | 1011nnnn (Bn) | 00110001 (31) | 0ttttttt |
| LSB for controllers 18 | 1011nnnn (Bn) | 00110010 (32) | 0ttttttt |
| LSB for controllers 19 | 1011nnnn (Bn) | 00110011 (33) | 0ttttttt |
| LSB for controllers 20 | 1011nnnn (Bn) | 00110100 (34) | 0ttttttt |
| LSB for controllers 21 | 1011nnnn (Bn) | 00110101 (35) | 0ttttttt |
| LSB for controllers 22 | 1011nnnn (Bn) | 00110110 (36) | 0ttttttt |
| LSB for controllers 23 | 1011nnnn (Bn) | 00110111 (37) | 0ttttttt |
| LSB for controllers 24 | 1011nnnn (Bn) | 00111000 (38) | 0ttttttt |
| LSB for controllers 25 | 1011nnnn (Bn) | 00111001 (39) | 0ttttttt |
| LSB for controllers 26 | 1011nnnn (Bn) | 00111010 (3A) | 0ttttttt |
| LSB for controllers 27 | 1011nnnn (Bn) | 00111011 (3B) | 0ttttttt |
| LSB for controllers 28 | 1011nnnn (Bn) | 00111100 (3C) | 0ttttttt |
| LSB for controllers 29 | 1011nnnn (Bn) | 00111101 (3D) | 0ttttttt |
| LSB for controllers 30 | 1011nnnn (Bn) | 00111110 (3E) | 0ttttttt |
| LSB for controllers 31 | 1011nnnn (Bn) | 00111111 (3F) | 0ttttttt |
| Damper Pedal (Sustain Pedal) | 1011nnnn (Bn) | 01000000 (40) | 0ttttttt |
| Portamento | 1011nnnn (Bn) | 01000001 (41) | 0ttttttt |
| Sostenato Pedal | 1011nnnn (Bn) | 01000010 (42) | 0ttttttt |
| Soft Pedal | 1011nnnn (Bn) | 01000011 (43) | 0ttttttt |
| Legato Footswitch | 1011nnnn (Bn) | 01000100 (44) | 0ttttttt |
| Hold 2 | 1011nnnn (Bn) | 01000101 (45) | 0ttttttt |
| Sound Controller 1 (default: Timber Variation) | 1011nnnn (Bn) | 01000110 (46) | 0ttttttt |
| Sound Controller 2 (default: Timber/Harmonic Content) | 1011nnnn (Bn) | 01000111 (47) | 0ttttttt |
| Sound Controller 3 (default: Release Time) | 1011nnnn (Bn) | 01001000 (48) | 0ttttttt |
| Sound Controller 4 (default: Attack Time) | 1011nnnn (Bn) | 01001001 (49) | 0ttttttt |
| Sound Controller 5 | 1011nnnn (Bn) | 01001010 (4A) | 0ttttttt |
| Sound Controller 6 | 1011nnnn (Bn) | 01001011 (4B) | 0ttttttt |
| Sound Controller 7 | 1011nnnn (Bn) | 01001100 (4C) | 0ttttttt |
| Sound Controller 8 | 1011nnnn (Bn) | 01001101 (4D) | 0ttttttt |
| Sound Controller 9 | 1011nnnn (Bn) | 01001110 (4E) | 0ttttttt |
| Sound Controller 10 | 1011nnnn (Bn) | 01001111 (4F) | 0ttttttt |
| General-Purpose Controllers 5 | 1011nnnn (Bn) | 01010000 (50) | 0ttttttt |
| General-Purpose Controllers 6 | 1011nnnn (Bn) | 01010001 (51) | 0ttttttt |
| General-Purpose Controllers 7 | 1011nnnn (Bn) | 01010010 (52) | 0ttttttt |
| General-Purpose Controllers 8 | 1011nnnn (Bn) | 01010011 (53) | 0ttttttt |
| Portamento Control | 1011nnnn (Bn) | 01010100 (54) | 0ttttttt |
| Effects 1 Depth (formerly External Effects Depth) | 1011nnnn (Bn) | 01011011 (5B) | 0ttttttt |
| Effects 2 Depth (formerly Tremolo Depth) | 1011nnnn (Bn) | 01011100 (5C) | 0ttttttt |
| Effects 3 Depth (formerly Chorus Depth) | 1011nnnn (Bn) | 01011101 (5D) | 0ttttttt |
| Effects 4 Depth (formerly Celeste Detune) | 1011nnnn (Bn) | 01011110 (5E) | 0ttttttt |
| Effects 5 Depth (formerly Phaser Depth) | 1011nnnn (Bn) | 01011111 (5F) | 0ttttttt |
| Data Increment | 1011nnnn (Bn) | 01100000 (60) | 0ttttttt |
| Data Increment | 1011nnnn (Bn) | 01100001 (61) | 0ttttttt |
| Non-Registered Parameter Number (LSB) | 1011nnnn (Bn) | 01100010 (62) | 0ttttttt |
| Non-Registered Parameter Number (MSB) | 1011nnnn (Bn) | 01100011 (63) | 0ttttttt |
| Registered Parameter Number (LSB) | 1011nnnn (Bn) | 01100100 (64) | 0ttttttt |
| Registered Parameter Number (MSB) | 1011nnnn (Bn) | 01100101 (65) | 0ttttttt |
| Mode Messages | 1011nnnn (Bn) | 01111001 (79) | 0ttttttt |
| Mode Messages | 1011nnnn (Bn) | 01111010 (7A) | 0ttttttt |
| Mode Messages | 1011nnnn (Bn) | 01111011 (7B) | 0ttttttt |
| Mode Messages | 1011nnnn (Bn) | 01111100 (7C) | 0ttttttt |
| Mode Messages | 1011nnnn (Bn) | 01111101 (7D) | 0ttttttt |
| Mode Messages | 1011nnnn (Bn) | 01111110 (7E) | 0ttttttt |
| Mode Messages | 1011nnnn (Bn) | 01111111 (7F) | 0ttttttt |
| Program Change (=select instrument) | 1100nnnn (Cn) | 0mmmmmmm | [none] |
| Channel Aftertouch or Channel Pressure | 1101nnnn (Dn) | 0rrrrrrr | [none] |
| Pitch Bend | 1110nnnn (En) | 0ffffff | 0ccccccc |

- **nnnn**
Refers to channel 0-15 (0 = for musicians channel no 1, 15 = for musicians no 16)
- **kkkkkkk**
Refers to key/note number 0-127 (60=middle C).
- **vvvvvvv**
Refers to velocity (0-127).
If Note Off this value is ignored.
If Note On, value 127 is the fastest.
- **ppppppp**
Refers to key pressure (0-127) (0 = no pressure, 127 = full pressure).
- **tttttt**
Refers to controller type (0-127).
- **mmmmmmm**
Refers to program number (0-127).
- **rrrrrrr**
Refers to amount (0-127).
- **ffffff**
Refers to value LSB (0-127).
- **ccccc**
Refers to value MSB (0-127).

Table 6: System Exclusive Events or <sysex event>

| Message description | Status byte (hex) | Length byte | Data |
|----------------------------|-------------------|-----------------|------------|
| Normal SysEx Events | FFFF0000 (F0) | variable-length | data bytes |
| Divided SysEx Events | FFFF0111 (F7) | variable-length | data bytes |
| Authorization SysEx Events | FFFF0000 (F0) | variable-length | data bytes |

- **sssssss**
The length byte specifies the number the number of bytes which follow the length byte, not including the status byte and the length byte.

Table 7: Meta Events or <meta-event>

| Message description | Status byte (hex) | Type byte (hex) | Length byte (hex) | Data |
|--|-------------------|-----------------|-------------------|----------|
| Sequence Number | 11111111 (FF) | 00000000 (00) | 00000010 (02) | data |
| Text Event | 11111111 (FF) | 00000001 (01) | len | text |
| Copyright Notice | 11111111 (FF) | 00000010 (02) | len | text |
| Sequence/Track Name | 11111111 (FF) | 00000011 (03) | len | text |
| Instrument Name | 11111111 (FF) | 00000100 (04) | len | text |
| Lyric | 11111111 (FF) | 00000101 (05) | len | text |
| Marker | 11111111 (FF) | 00000110 (06) | len | text |
| Cue Point | 11111111 (FF) | 00000111 (07) | len | text |
| MIDI Channel Prefix | 11111111 (FF) | 00100000 (20) | 00000001 (01) | 0000nnnn |
| End Of Track | 11111111 (FF) | 00101111 (2F) | 00000000 (00) | [none] |
| Set Tempo in microseconds per quarter note | 11111111 (FF) | 01010001 (51) | 00000011 (03) | data |
| SMPTE Offset | 11111111 (FF) | 01010100 (54) | 00000101 (05) | data |
| Time Signature | 11111111 (FF) | 01011000 (58) | 00000100 (04) | data |
| Key Signature | 11111111 (FF) | 01011001 (59) | 00000010 (02) | data |
| Sequencer Specific Meta Event | 11111111 (FF) | 01111111 (7F) | len | data |

- **len**
Refers to the length portion of the meta-event syntax, that is, a number, stored as a variable-length quantity, which specifies how many data bytes follow it in the meta-event.
- **text and data**
Refers to however many bytes of (possibly text) data were just specified by the length.
- **nnnn**
Refers to channel 0-15 (0 = for musicians channel no 1, 15 = for musicians no 16)

Table 8: More information about Messages

| Message | Description |
|--------------|---------------|
| End Of Track | 11111111 (FF) |

[Goto Top](#)

Posted comments (4)**Posted by G at February 19, 2014 17:12:23 CET**

Thank you so much for this. Extremely useful.

G.

Posted by Dan Lee at November 12, 2015 14:46:24 CET

Thank you for this great guide!

I spotted a mistake in the last two examples of the calculation in the time-code-based time:
0xE350 equals 80 beats per frame not 50. And the same applies for 0xE250.

It seems like the binary representation for the bits 0-7 is mixed up too. It should be 01010000 instead of 00110010.

Posted by Top Dowg at January 24, 2020 19:52:58 CET

Very helpful.
Thank you very much.

Posted by Jayant Panhalkar at August 27, 2021 21:00:10 CEST

Here delta time = 240
In real time it's 0.2777775 micro seconds
I got this time from python code
But in real life how to calculate time in micro seconds from delta time?

Post a comment

Please review my [comments policy](#) and make sure your comment is relevant to the topic discussed otherwise it will be deleted. To format your message use [BBCode tags](#). All comments are submitted for approval prior to being published. Please allow up to 24 hours for the approval process to take place.

Name*:

Email (never displayed):

URL (begin with http://):

Comments (you may use [BBCode tags](#) for style, max 5000 characters)*:

To prevent automated submissions an Access Code has been implemented for this tool.

0PQ

Please enter the Access Code as displayed above*:

*** = required**

[Goto Top](#)

[Home](#) | [Shop](#) | [Contact](#) | [Emulators](#) | [Tutorials](#) | [Developer](#) | [Services](#) | [Mobile](#) | [Games](#) | [Login](#) | [Legal](#) | [Sitemap](#)

Copyright © 2002 - 2021, [Mobilefish.com](#)

Please review my [privacy policy](#) before using this site.

For problems or assistance with this site, [send an email to Mobilefish.com](#).