AXE-FX III MIDI FOR THIRD-PARTY DEVICES

Revision 1.4

SYNPOSIS

Third-party devices can control the Axe-Fx III using standard MIDI commands, i.e. PC, CC, etc. However, the status of effect bypass states, channels, scenes, etc. is not supported by standard MIDI commands. This document outlines the System Exclusive (sysex) commands to query this data from the Axe-Fx III.

Effect states can also be controlled via a sysex message. This allows ease of control as the various CC numbers need not be programmed into the Axe-Fx III's MIDI menu.

This is supported in Axe-Fx III firmware version 1.13 or greater.

The general format of the sysex commands is:
F0 00 01 74 10 cc dd dd dd … cs F7
where cc is the command opcode, dd is a variable number of bytes and cs is the XOR checksum.

See the Appendix for a list of effect IDs.

SET/GET BYPASS (COMMAND 0AH)

Sets and/or gets the bypass state of an effect.

Message format:
F0 00 01 74 10 0A id id dd cs F7.
id id is the effect ID as two 7-bit MIDI bytes, LS first.
dd is the bypass state (0 = engaged, 1 = bypassed).
To query set dd = 7F.

Returns:
F0 00 01 74 10 0A id id dd cs F7
where dd is the current bypass state: 0 = engaged, 1 = bypassed.

SET/GET CHANNEL (COMMAND 0BH)

Sets and/or gets the channel of an effect.

Message format:
F0 00 01 74 10 0B id id dd cs F7.
id id is the effect ID as two 7-bit MIDI bytes, LS first.
dd is the channel (0 – 3).
To query set dd = 7F.

Returns:
F0 00 01 74 10 0B id id dd cs F7
where dd is the current channel (0 - 3).

SET/GET SCENE (COMMAND 0CH)

Sets and/or gets the scene.

Message format:
F0 00 01 74 10 0C dd cs F7.
dd is the desired scene. To query set dd = 7F.

Returns:
F0 00 01 74 10 0C dd cs F7; where dd is the current scene

QUERY PATCH NAME (COMMAND 0DH)

Returns preset name.

Message format:
F0 00 01 74 10 0D dd dd cs F7
where dd dd is the preset number. To query the current preset name let dd dd = 7F 7F.

Returns:
F0 00 01 74 10 0D nn nn dd dd dd … cs F7;
where nn nn is the preset number as two 7-bit MIDI bytes, LS first
and dd dd dd … is 32 characters of name.

QUERY SCENE NAME (COMMAND 0EH)

Returns scene name.

Message format:
F0 00 01 74 10 0E dd cs F7; where dd is desired scene.
To return the current scene name dd = 7F.

Returns:
F0 00 01 74 10 0E nn dd dd dd … cs F7;
where nn is the scene number and dd dd dd … is 32 characters of name.

SET/GET LOOPER STATE (COMMAND 0FH)

Set and or gets the Looper states.

Message format:
F0 00 01 74 10 0F dd cs F7.
where dd is the desired “button” to press:
0: Record

1: Play
2: Undo
3: Once
4: Reverse
5: Half-speed
to query the state set dd = 7F.

Returns:
F0 00 01 74 10 0F dd cs F7
where dd is the looper state:
Bit 0: Record
Bit 1: Play
Bit 2: Overdub
Bit 3: Once
Bit 4: Reverse
Bit 5: Half-speed

TEMPO TAP (COMMAND 10H)

Remotely taps the Tempo button.

Message format:
F0 00 01 74 10 10 cs F7.

TUNER ON/OFF (COMMAND 11H)

Turns the Tuner on or off.

Message format:
F0 00 01 74 10 11 dd cs F7.
where dd = 0 (off) or 1 (on).

STATUS DUMP (COMMAND 13H)

This command requests a status dump of all effects in the current preset.

Message format:
F0 00 01 74 10 13 cs F7.

Returns a variable length message:
F0 00 01 74 10 13 <packet0> <packet1> … <packetN> cs F7
where a packet is three bytes, id id dd, defined as follows:
id = effect ID, 14 bits over two MIDI bytes, LS first
dd:
bit 0: bypass state: 0 = engaged, 1 = bypassed
bit 3-1 channel (0-7, currently the maximum number of channels for any effect is 3).
bit 6-4: number of channels supported for this effect (0-7).

SET/GET TEMPO (COMMAND 14H)

Sets/gets the Tempo.

Message format:
F0 00 01 74 10 14 dd dd cs F7;
where dd dd is the desired tempo as two 7-bit MIDI bytes, LS first.
To query the tempo let dd dd = 7F 7F.

"PUSH DATA"

The Axe-Fx III sends two types of "push data": tempo and tuner. When the Send Realtime Sysex parameter in the Global menu is set to On the Axe-Fx III will push tempo and tuner data to the MIDI Out jack. Note: this data ONLY streams over the MIDI Out jack. It does not stream over the MIDI-Over-USB output. The format of these messages are as follows:

MIDI TEMPO

Sent on the tempo down beat.

Message format:
F0 00 01 74 10 10 F7.

Note that no checksum is sent so as to minimize message length.

MIDI TUNE

Sent when the Tuner is activated.

Message format:
F0 00 01 74 10 11 nn ss cc F7.
where:
nn = note (0 – 11)
ss = string (0 – 5, 0 = low E)
cc = cents (offset binary, 63 = 0, 62 = -1, 64 = +1, etc.)

Note that no checksum is sent so as to minimize message length.

APPENDIX 1: EFFECT IDS

These are enumerated IDs for

```
typedef enum EFFECT_ID_
{
        ID_CONTROL = 2,
        ID_TUNER = 35,
        ID_IRCAPTURE,
        ID_INPUT1,
        ID_INPUT2,
        ID_INPUT3,
        ID_INPUT4,
        ID_INPUT5,              // USB Input
        ID_OUTPUT1,
        ID_OUTPUT2,
        ID_OUTPUT3,
        ID_OUTPUT4,
        ID_COMP1,
        ID_COMP2,
        ID_COMP3,
        ID_COMP4,
        ID_GRAPHEQ1,
        ID_GRAPHEQ2,
        ID_GRAPHEQ3,
        ID_GRAPHEQ4,
        ID_PARAEQ1,
        ID_PARAEQ2,
        ID_PARAEQ3,
        ID_PARAEQ4,
        ID_DISTORT1,
        ID_DISTORT2,
        ID_DISTORT3,
        ID_DISTORT4,
        ID_CAB1,
        ID_CAB2,
        ID_CAB3,
        ID_CAB4,
        ID_REVERB1,
        ID_REVERB2,
        ID_REVERB3,
        ID_REVERB4,
        ID_DELAY1,
        ID_DELAY2,
        ID_DELAY3,
        ID_DELAY4,
        ID_MULTITAP1,
        ID_MULTITAP2,
        ID_MULTITAP3,
        ID_MULTITAP4,
        ID_CHORUS1,
        ID_CHORUS2,
        ID_CHORUS3,
        ID_CHORUS4,
        ID_FLANGER1,
        ID_FLANGER2,
        ID_FLANGER3,
        ID_FLANGER4,
        ID_ROTARY1,
        ID_ROTARY2,
        ID_ROTARY3,
        ID_ROTARY4,
        ID_PHASER1,
        ID_PHASER2,
        ID_PHASER3,
        ID_PHASER4,
        ID_WAH1,
        ID_WAH2,
        ID_WAH3,
        ID_WAH4,
        ID_FORMANT1,
        ID_FORMANT2,
        ID_FORMANT3,
        ID_FORMANT4,
        ID_VOLUME1,
        ID_VOLUME2,
        ID_VOLUME3,
        ID_VOLUME4,
        ID_TREMOLO1,
        ID_TREMOLO2,
        ID_TREMOLO3,
        ID_TREMOLO4,
        ID_PITCH1,
        ID_PITCH2,
        ID_PITCH3,
        ID_PITCH4,
        ID_FILTER1,
        ID_FILTER2,
        ID_FILTER3,
        ID_FILTER4,
        ID_FUZZ1,
        ID_FUZZ2,
        ID_FUZZ3,
        ID_FUZZ4,
        ID_ENHANCER1,
        ID_ENHANCER2,
        ID_ENHANCER3,
        ID_ENHANCER4,
        ID_MIXER1,
        ID_MIXER2,
        ID_MIXER3,
        ID_MIXER4,
        ID_SYNTH1,
        ID_SYNTH2,
        ID_SYNTH3,
        ID_SYNTH4,
        ID_VOCODER1,
        ID_VOCODER2,
        ID_VOCODER3,
        ID_VOCODER4,
        ID_MEGATAP1,
        ID_MEGATAP2,
        ID_MEGATAP3,
        ID_MEGATAP4,
        ID_CROSSOVER1,
        ID_CROSSOVER2,
        ID_CROSSOVER3,
        ID_CROSSOVER4,
        ID_GATE1,
        ID_GATE2,
        ID_GATE3,
        ID_GATE4,
        ID_RINGMOD1,
        ID_RINGMOD2,
        ID_RINGMOD3,
        ID_RINGMOD4,
        ID_MULTICOMP1,
        ID_MULTICOMP2,
        ID_MULTICOMP3,
        ID_MULTICOMP4,
        ID_TENTAP1,
        ID_TENTAP2,
        ID_TENTAP3,
        ID_TENTAP4,
        ID_RESONATOR1,
        ID_RESONATOR2,
        ID_RESONATOR3,
        ID_RESONATOR4,
        ID_LOOPER1,
        ID_LOOPER2,
        ID_LOOPER3,
        ID_LOOPER4,
        ID_TONEMATCH1,
        ID_TONEMATCH2,
        ID_TONEMATCH3,
        ID_TONEMATCH4,
        ID_RTA1,
        ID_RTA2,
        ID_RTA3,
        ID_RTA4,
        ID_PLEX1,
        ID_PLEX2,
        ID_PLEX3,
        ID_PLEX4,
        ID_FBSEND1,
        ID_FBSEND2,
        ID_FBSEND3,
        ID_FBSEND4,
        ID_FBRETURN1,
                ID_FBRETURN2,
                ID_FBRETURN3,
                ID_FBRETURN4,
                ID_MIDIBLOCK,
                ID_MULTIPLEXER1,
                ID_MULTIPLEXER2,
                ID_MULTIPLEXER3,
                ID_MULTIPLEXER4,
                ID_IRPLAYER1,
                ID_IRPLAYER2,
                ID_IRPLAYER3,
                ID_IRPLAYER4,
                ID_FOOTCONTROLLER,
                ID_PRESET_FC,
        } EFFECT_ID;
```