FSM System Exclusive Data - Format (Version 2.0)

General setup: (all data: hexadezimal)

F0	00	20	0D	<ID>	07	<CMD>	<Data> … F7
		*1		*2	*3	*4	*5

***1 = Manufacturer - ID:** *00 20 0D* = MIDITEMP
2 = Device-ID**. In case you use several FSM units, each unit can use it's individual ID. Via ID you can adress a single FSM individually. When you use7F*** here, the SysEx – message will be sent to all connected units, by using a specified adress only the unit will accept the message, that was flashed before with an specified ID-number! Usually, when using just one single unit, you always may use ***7F*** Therefore you will find all further examples with this adress ***7F***
***3 = Device type:** *07* = FSM
4 = Command ID.** Defines the function of the SYSEX- message: ***00 = Change data/function for switch 1 ***01*** = Change data/function for switch 2 ***02*** = Change data/function for pedal 1 ***03*** = Change data/function for pedal 1 ***04*** = Change device-ID of the FSM ***05*** = Readout actual firmware version ***06*** = Change data/function for switch 3 ***07*** = Change data/function for switch 4
***5 = Data,** content according to the command - ID

Coding of the midi-data to be sent with the SysEx-datastream:

For inside SysEx-data values >= 0x80 are not allowed, the MIDI statusbytes are coded as 0x00..0x7F. So you can send any MIDI event-types you like. The maximum lenght of a data-stream is 40 bytes for switch1, switch2 and for pedals. Maximum length for switch3 or switch4 is 20 bytes!

MIDI command:	Coding:	Type:
8n NN VV	***0n NN VV***	Note Off
9n NN VV	***1n NN VV***	Note On (VV=0 gilt als Note Off)
An NN VV	***2n NN VV***	Polypressure
Bn NN VV	***3n NN VV***	Control Change
Cn VV	***4n VV***	Program Change
Dn VV	***5n VV***	Channel Pressure
En vv VV	***6n vv VV***	Pitch Bend
F0 data... F7	***70 data...***	SysEx (EOX=***F7*** is not coded here!*)
F1..F7	***71..77***	System Common (if necessary: with 1 or 2 databytes)
F8..FF	***78..7F***	System Realtime

Explanation:

n = MIDI channel No (0..F for channel 1..16)
NN = Note-number or controller-number
VV = data parameter(with notes: velocity)
vv = data parameter LSB

*SyEx-data are applied from the coded statusbyte (**70**) up to the end of the SysEx-message. The final message EOX (**F7**) will be generated automatically and may **NOT** be coded in your data-stream!

CMD = *00* or *01*: Programming the functions of switch 1 or 2:

Here the block of data consists of a„Mode"-databyte, followed by coded midi-data, that are sent by hitting the switches (switch on / switch off):

F0 00 20 0D <ID> 07 00/01 <Mode> [<MIDI Data> …] F7

Mode	**Meaning:**
00	Toggle-mode off, previous stored MIDI data for on/off will not be affected or changed (the box **<MIDI Data>** can be ignored)
01	Toggle mode off, **<MIDI Data>** for on/off or hitting a non-permanent switch will be accepted. In case no **<MIDI Data>** are defined, no data will be sent after "ON"
02	Toggle-mode off, **<MIDI Data>** will be accepted when releasing the switch (OFF). In case no **<MIDI Data>** are defined, no data will be sent after "OFF"
03	Toggle-mode off, **<MIDI Data>** will be accepted for ON **and** for OFF when using a non-permanent switch. Value *VV* of each MIDI-order will be replaced when sending data, concerning to the pedals' status by using ***00*** („off") or ***7F*** („on").
04	Toggle-mode on, prvious stored data will not be affected or changed. (the box **<MIDI Data>** can be ignored)
05	Toggle-mode on, **<MIDI Data>** will be accepted for „ON". If no **<MIDI Data>** are defined, no data will be sent.
06	Toggle-mode on, **<MIDI Data>** will be accepted for "OFF" If no **<MIDI Data>** are defined, no data will be sent.
07	Toggle-mode on, **<MIDI Data>** will be accepted both for switch on and for swtich off. Value *VV* of each MIDI-order will be replaced when sending data, concerning to the pedals' status by using ***00*** („off") or ***7F*** („on").

Toggle-Mode:

In the Toggle-mode (on) a non permanent switch (Latch) works like a switch – each hit onto the pedal does one action, On / OFF. Releasing the latch will not cause any action then. When Toggle-mode is off, you will have two actions when using a latch-pedal (On/Off).
For example you trigger a sample (on) and stop it again by releasing the latch-pedal!

Data values sent by switch1/switch2:

In case you defined seperate data values **mode = 01/02** or **mode = 05/06** for the On/Off action. Your defined MIDI data are going to be sent with each action with your non-permanent switch, as stored at the parameter ***VV***.

When using **mode = 03** or **mode = 07** your non-permanent switch will send out the same data block. However, values of ***VV*** of the MIDI order will here be replaced by the current switch-status. So the stored value ***VV*** = ***00*** will be sent when you hit your switch (ON), by hitting it again value ***00*** will be sent.

If ***VV*** = ***7F*** was stored, switch 1/2 will send data value ***00***, (On), ***7F*** will be sent by sitching again (Off).

CMD = *02* or *03*: Programming the functions of pedal 1 or 2

The data block here consist of a „position"- databyte, followed by coded midi date when using the pedal.

F0 00 20 0D <ID> 07 02/03 <Pos> <MIDI Data> … F7

For Pos = 0: Corresponding to the coding of the data to be sent **<MIDI data>** *,* midi data are created when you move the pedal. From that values of "VV" of the data block will be replaced with new values, concerning to the position of your pedal. Values stored in ***VV=0*** will be sent normal, values stored in ***VV=7F*** will be sent out inverted. That means, the min/max – position of the pedal is swapped now.

For Pos > 0: Data are sent the same way as above, **Pos = 0**. However, for the value of the position here is defined one more addtitional byte. The value of this byte is replaced by the pedal's position then. Basically this only makes sense when you want to send SysEx-data by using a pedal, because you need to define which byte inside your SysEx-data stream has to do the value of ***VV***. The position indication is counted from the first byte of your **<MIDI Data>**.

*Also Pos >0 applies to: Values stored at **VV=0** will be sent normal, values of **VV=7F** will be sent out inverted, means, the min/max – position of your pedal is swapped.*

CMD = 04: Changing the device ID of the FSM

F0 00 20 0D <ID> 07 04 <new ID> F7

CMD = 05: Readout the firmware-version of the FSM (new in firmware-version 2!)

F0 00 20 0D <ID> 07 05 F7

FSM sends back: *F0 00 20 0D 7E 07 05* <*Version*> <*Sub-Version*> *F7*

CMD = 06 *oder* 07: Programming the functions of switch 3 or 4:

Since firmware-version 2 you also can connect switches to the pedal-connectors, so pedal1 = switch3 and pedal2 is = switch 4 now!

F0 00 20 0D <ID> 07 06/07 <Mode> [<MIDI Data> …] F7

<*Mode*> and <*MIDI Data*> correspond to the data for Switch 1 and 2. The maximum lenght of a MIDI-data block for switch3 and switch4 is 20 Bytes, instead 40 bytes!

Examples:

FSM factory-setting using a non-permanent switch at **connector (switch) 1**
FX-Mute f. Yamaha 01V/96
To that, four control-change orders at midi-channel 2 will be created, (CC #. 72 to 75), to switch on/off the effects of this mixer by using a (non-permanent) switch pedal

F0	***00 20 0D 7F 07***	***00***	***07***	***31***	***48***	***00***	***31***	***49***	***00***	***31***	***4A***	***00***	***31***	***4B***	***00***	***F7***
	****1***	****2***	****3***	****4***	****5***	****6***	****4***	****5***	****6***	****4***	****5***	****6***	****4***	****5***	****6***	

***1 =** ID-Feld (MIDITEMP, Device ID and device type - FSM)
***2 =** CMD = 00: Data for switch 1
***3 =** Mode = 07: Toggle-mode (change-over on<->off with each step to the pedal), the next coming data-set is used for On/Off.
***4 =** 31 coded status-byte „Control Change“ at midi-channel 2 (_ B1)
***5 =** Controller No. 48..4B (hexadecimal) = number 72..75 (decimal)
***6 =** Data value (VV) = 0, switch position on<->off will not be swapped

FSM factory-setting using a non-permanent switch at connector (Switch)1: Sends MIDI Start/Stop

To that, two SysEx-orders will be sent to the FSM:

F0	***00 20 0D 7F 07***	***01***	***05***	***7A***	***F7***
	****1***	****2***	****3***	****4***	

F0	***00 20 0D 7F 07***	***01***	***06***	***7C***	***F7***
	****1***	****2***	****5***	****6***	

***1 =** ID-Feld (MIDITEMP, Device ID and device type - FSM)
***2 =** CMD = 01: Data for switch 2
***3 =** Mode = 05: Toggle-mode, the following data will be sent by stepping onto the pedal.
***4 =** 7A = coded status-byte „MIDI Start“ (FA)
***5 =** Mode = 06: Toggle-mode, the following data will be sent by releasing the pedal again.
***6 =** 7C = coded status-byte „MIDI Stop“ (FC)

FSM factory-setting for pedal 1/2: Expression control (control Chg. No. 11) at midi channel 1 or rather channel 2:

F0 00 20 0D 7F 07 02/03 00 30/31 0B 00 F7
****1 *2 *3 *4 *5 *6***

***1 =** ID: (MIDITEMP, Device ID and device type - FSM)
***2 =** CMD = 02 bzw. 03: data for pedal 1 or rather 2
***3 =** Pos = 00: no byte needs to be set at VV, for only controller data will be sent!
***4 =** 30 bzw. 31 = codded status-byte for Control Change at midi-channel 1 or rather 2
***5 =** 0B = Controller No. # 11(decimal): „Expression control“
***6 =** 00 = Controller not inverted!

Universal System Exclusive: Master volume with pedal 2:

F0 00 20 0D 7F 07 03 06 70 7F 7F 04 01 00 00 F7
****1 *2 *3 *4 *5***

***1 =** ID: (MIDITEMP, Device ID and device type - FSM)
***2 =** CMD = 03: Data for pedal 2
***3 =** Pos = 06: the 6th. byte, counted from the next floowing one, is to be marked as: VV
***4 =** 70 = coded status-byte for System Exclusive (=> *F0*)
***5 =** Content (= data) of the SysEx-order

When moving the pedal, the following message will be sent:

F0 7F 7F 04 01 00 VV F7

whereas for VV the value will be sent concerning to the position of the pedal.

More examples and downloads on our website - www.Miditemp.com