# **SmartFade MIDI specification**

## **SmartFade version 1.50**

27 April 2005

## **Contents:**

Specification:
MIDI
Master / Slave operation:
Master / Slave functions and commands:
MIDI channel usage:
Midi IN and OUT functions
Midi Controllers
Controller types:
Default controller patch:
Program change
Select memory page and Stack operation:
Note On / Note Off:
Memory bumps:
Midi Protocol:
Midi channel:
Running status:7

## Revisions:

- 0.10 First implementation
- 0.11 OMNI receive mode deleted
  - Detailed list of feature set in SmartFade version 1.5
- 0.12 Edited spec for public distribution with details of SmartFade 1.5.0 implementation

## Specification:

## MIDI

MIDI facilities are included to provide two primary functions:

- Show control systems: The unit may be set to respond to MIDI data from a time-based sequencer or show control system.
- Slave mode for SmartFade (or vice versa): SmartFade may be set to MIDI slave and respond to changes on a Master SmartFade console, or vice versa. Master operations are followed by the slave. Because all SmartFades include a DMX input, any type may be connected to feed DMX to the master to be merged.

The user need not be concerned with the detailed setup of these modes. In each operating mode it shall suffice to make a MIDI connection and set one unit as a master and the other(s) as slave(s).

### Master / Slave operation:

A unit set to Master mode will transmit key presses and fader movements on its OUT port.

A master ignores data on the IN port.

A slave will respond to inputs on its IN port and re-transmit them (MIDI thru) to its OUT port.

A slave does not produce any messages of its own. The only data appearing on the OUT port of a slave is the MIDI thru data copied from the IN port.

#### Master / Slave functions and commands:

A master can operate all functions internally.

A Master sends a subset of the available functions via MIDI. Memory and Stack playback are supported but not sequence stepping and some others functions.

A unit set to slave may still be operated locally.

#### MIDI channel usage:

A master may be set to transmit on any channel. The default is Channel 1. The transmit channel may be set in a menu.

A slave may be set to respond to any channel. The default is Channel 1. The receive channel may be set in a menu.

## Midi IN and OUT functions

First version:

Midi data is restricted to two main functions: GO/PAUSE/BACK operation of the Stack and assigning faders to controllers and buttons to Note-on/off commands.

Future versions will provide full remote control as required.

Specific fader and button movements may be sent via Midi.

Midi commands supported are:

- Controllers
- Program change
- Note On / Of

No system-real-time or system-exclusive commands are used. Masters do not send these commands. Slaves must discard these commands.

### Midi Controllers

Positions of virtual faders are sent as Control change messages. All levels are coded as 7 bit unsigned.

Control change messages used to slave two SmartConsoles together are pre-patched to specific controller numbers that may not be modified by the user.

#### Controller types:

#### Paged virtual faders:

Control change messages are sent to control memories. Each page corresponds to the page of faders. Messages are only sent for matched faders. Unmatched faders must first be matched on the Master console before a new memory level message is sent.

Paged Virtual faders are assigned to fixed controller numbers and may not be patched.

#### Aux. Controls:

The 4 master faders are assigned to fixed controllers and may not be patched by the user.

#### Default controller patch:

Ctrl change	Transmit	(Master):			
1011nnnn	Ommmmmm	0vvvvvv			
			mmmmmm	=	Fader number:
			*****	=	Level 0-7F, top 7 bits of level or bank number
		(*)	0	=	reserved
		(*)	1-48	=	Paged Virtual fader level, 1-48
		(*)	49-123	=	reserved
		(*)	124	=	XF down
		(*)	125	=	XF up
		(*)	126	=	Bumps master
		(*)	127	=	Grand master

Ctrl change Receive (Slave): Ommmmmmm

1011nnnn

0vvvvvv			
	mmmmmm	=	Fader number:
	<b>vvvvvv</b>	=	Level 0-7F, top 7 bits of level or bank number
(*)	0	=	reserved
(*)	1-48	=	Paged Virtual fader level, 1-48
(*)	49-123	=	reserved
(*)	124	=	XF down
(*)	125	=	XF up
(*)	126	=	Bumps master
(*)	127	=	Grand master

(\*) Code set implemented in SmartFade version 1.5

(\*) These codes are reserved for future implementation, not to be included in initial SmartFade Midi release.

### Program change

Program change messages are used to set the current virtual fader page, the current virtual channels page or to jump to a Stack step.

#### Select memory page and Stack operation:

Prog chang	<u>ge</u>			
1100nnnn	Оррррррр			
		ррррррр	=	Memory page:
	(*)	0	=	Goto Stack step zero (dead state to allow next fade to step one)
	(*)	1-99	=	Stack step number.
	(*)	100-123	=	Memory page number 1-24
	(*)	124	=	GO
	(*)	125	=	PAUSE
	(*)	126	=	Black Out ON
	(*)	127	=	Black Out OFF

(\*) Code set implemented in SmartFade version 1.5

As the user turns the dial on a Master to jump to a step no data is sent to the Slave. When the user releases the Stack button the Master sends the message.

## Note On / Note Off:

#### Memory bumps:

Note-Off T	<u>ransmit</u>			
1000nnnn	Ommmmmmm	0vvvvvv		
 <u>Note On T</u>	ransmit:			
1001nnnn	Ommmmmm	0vvvvvv		
		<b>vvvvvv</b>	= =	normally any non-zero value zero for running status Note-Off
		mmmmmmm	=	Fader number / button number:
	(*) (*)	0-47 48-127	= =	Mems Bump 1-48 reserved

(\*) Code set implemented in SmartFade version 1.5

(\*) Code set for future implementation

If SOLO mode = ON in the slave console receiving a Memory Bump On command will kill all other sources except other memories that have been, and are still, bumped on.

All bumps operate as momentary. When the button is pressed one Note-on command is sent. When the button is released the Note-off Command is sent.

### Midi Protocol:

#### Midi channel:

For all messages the user may set a single Midi Transmit and a single Midi receive channel.

#### Running status:

Slaves shall support continuous running status messages.

Master shall generate running status messages by default. This is to reduce MIDI traffic and minimise apparent delays. Status bytes shall only be transmitted when the message status changes or after continuous running commands have been sent for one second.

Note-Off messages may be sent for running status as Note-On with velocity of zero.