

**** US-2400 Native Mode ****

Version 1.31

July, 2005

Buttons / Leds (0xB1)

<0xB1> <0rriiinn> <on=0x00/flashing=0x01/off=0x02...0x7f>

rr = regoin

0 = Channels 1-8

1 = Channels 9-16

2 = Channels 17-24

3 = Master section

FOR CHANNEL RETIONS (0-2)

iii = index

0 = First channel of group

1 = Second channel of group

.....

7 = Eighth channel of group

nn = number

0 = Fader Touch

1 = Select

2 = Solo

3 = Mute

FOR MASTER RETION

iiinn = number

- 0 = Master Fader touch
- 1 = Master select
- 2 = Clear solo
- 3 = Flip
- 4 = Channel (phat channel_
- 5 = Aux 1
- 6 = Aux 2
- 7 = Aux 3
- 8 = Meter
- 9 = Pan
- 10 = Aux 4
- 11 = Aux 5
- 12 = Aux 6
- 13 = Fnc
- 14 = Null
- 15 = Scrub
- 16 = Bank +
- 17 = Bank -
- 18 = In
- 19 = Out
- 20 = Shift
- 21 = RWND
- 22 = FFWD
- 23 = STOP
- 24 = PLAY
- 25 = REC
-]

Bank Leds (0xB0)

<0xB0> <0x5D> <nn>

nn = Bank

- 0 = Bank Led 1
- 1 = Bank Led 2
- 2 = Bank Led 3
- 3 = Bank Led 4
- 4 = Bank Led 5
- 5 = Bank Led 6
- 6 = Bank Led 7
- 7 = Bank Led 8

FADERS (0xB0)

Faders are sent/received using CC messages 0 -> 24 (msb) and 32 -> 56 (lsb)

example:

B0 03 42 23 20 (note running stauts used)

```
| | | | |
| | | | +-LSB Value of 0x20
| | | +----LSB of fader 4
| | +-----MSB value of 0x42
| +-----MSB fader 4
+-----MIDI stauts CC
```

ENCODERS (0xB0)

Encoders are sent on MIDI CC's 64 -> 88

Encoder changes are sent as 0(bit7) + Sign(bit6) +Magnitude(bit5..0)

Sign	0 : Clockwise
	1 : Counter-clockwise
Magnitude	0-63 : nuber of 'clicks'

examples

channel 5 encoder turned CCw 3 "clicks"

<0xB0> <0x24> <0x43>

channel 1 encoder turned CW 1 'click'

<0xB0> <0x20> <0x01>

JOG WHEEL (0xB0)

Jog Wheel uses same format as encoders, sent on MIDI CC# 89 ?

JOYSTICK (0xBE)

Joystick is sent using MIDI CC's 90 for X and 91 for Y

FOOTSWITCH (0xB0)

Sends CC#92 for tip and CC#93 for

Meters:

Midi CC 96 – 119 inclusive

PEAKS: If the data byte has the 4th bit set (data & 0x10) the peak value is set to the lower nibble of the data byte (peak = data & 0xF0).

LEVELS: if the data byte does not have the the 4th bit set, then the meters level is just the lower nibble of the data byte (level = data & 0xF0);

OVERLOAD: Overload light is controlled by bits 5 and 6. If bit 6 is set, then the overload bit should be set or unset. If bit 6 is not set then the overload remains unchanged. When bit 6 is set, bit 5 represents the state of the overload light, a 1 indicating on and 0 indicating off.

Examples:

0xB0 0x96 0x14 -- set the PEAK of CHANNEL 1 METER to 4

0xB0 0x98 0x03 -- set the LEVEL of the CHANNEL 3 METER to 3

0xB0 0x98 0x6F -- turn ON the OL light and set the meter to it's maximum value.

Rings:

MIDI CC 64 – 87 inclusive

The rings are set in the following way:

CENTER POINT: the center point is set if the last ring setting sent contains the 6th bit of the data byte (data & 0x40)

RING VALUE: the ring will be set to a pattern looked up in the following map. The value if the ring is only the remaining bits, 0-5. The lower nibble is represented in the table as 0-15 of each mode, first two bits of the upper nibble as the mode number.

Mode One: Dot Mode ((data & 0x30) = 0x00)

//0000000000000000	// 0
//0000000000000001	// 1
//0000000000000010	// 2
//0000000000000100	// 3
//0000000000001000	// 4
//0000000000010000	// 5
//0000000000100000	// 6
//0000000001000000	// 7
//0000000010000000	// 8
//0000000100000000	// 9
//0000001000000000	// 10
//0000010000000000	// 11
//0000100000000000	// 12
//0001000000000000	// 13
//0010000000000000	// 14
//0100000000000000	// 15

Mode Two: Cut/Boost ((data & 0x30) = 0x10)

//0000 0000 0000 0000	// 0
//0000 0000 1111 1111	// 1
//0000 0000 1111 1110	// 2
//0000 0000 1111 1100	// 3
//0000 0000 1111 1000	// 4
//0000 0000 1111 0000	// 5
//0000 0000 1110 0000	// 6
//0000 0000 1100 0000	// 7
//0000 0000 1000 0000	// 8
//0000 0001 1000 0000	// 9
//0000 0011 1000 0000	// 10
//0000 0111 1000 0000	// 11
//0001 1111 1000 0000	// 12
//0011 1111 1000 0000	// 13
//0111 1111 1000 0000	// 14
//0111 1111 1000 0000	// 15

Mode Three: Spread Mode ((data & 0x30) = 0x20)

//0000 0000 0000 0000	// 0
//0000 0000 0000 0001	// 1
//0000 0000 0000 0011	// 2
//0000 0000 0000 0111	// 3
//0000 0000 0000 1111	// 4
//0000 0000 0001 1111	// 5
//0000 0000 0011 1111	// 6
//0000 0000 0111 1111	// 7
//0000 0000 1111 1111	// 8
//0000 0001 1111 1111	// 9
//0000 0011 1111 1111	// 10
//0000 0111 1111 1111	// 11
//0000 1111 1111 1111	// 12
//0001 1111 1111 1111	// 13
//0011 1111 1111 1111	// 14
//0111 1111 1111 1111	// 15

Mode Four: Width Mode (data & 0x30 = 0x30)

```
//0000|0000|0000|0000 // 0
//0000|0000|1000|0000 // 1
//0000|0001|1100|0000 // 2
//0000|0011|1110|0000 // 3
//0000|0111|1111|0000 // 4
//0000|1111|1111|1000 // 5
//0001|1111|1111|1100 // 6
//0011|1111|1111|1110 // 7
//0111|1111|1111|1111 // 8
//0111|1111|0111|1111 // 9
//0111|1110|0011|1111 // 10
//0111|1100|0001|1111 // 11
//0111|1000|0000|1111 // 12
//0111|0000|0000|0111 // 13
//0110|0000|0000|0011 // 14
//0100|0000|0000|0001 // 15
```