By BEE (Bernard Escaillas 2008)

ION/MICRON PATCH DUMP SYSEX FORMAT

Single Patch Request Message

To request a single patch, send the following Sysex Message to the Ion/Micron
F0 00 00 0E 22 41 0x 00 yy F7

Hexa bytes	Role
F0	Sysex begin tag
00 00 0E	Alesis manufacturer ID
22	Ion/Micron ID
41	Opcode 'Request patch'
0x	From bank / x = { 0...4} for { Red, Green, Blue, Yellow(user), Edit}
00	Single patch
yy	Patch number / yy = {0...127} / {0...3} only for Edit bank
F7	Sysex end tag

Returning message is detailed below.
Length is 1 + (56 + 315) / 7 * 8 + 1 = 434 midi bytes

Bank Request Message

To request a bank (128 patches), send the following Sysex Message
F0 00 00 0E 22 41 0x 01 00 F7

Hexa bytes	Role
F0	Sysex begin tag
00 00 0E	Alesis manufacturer ID
22	Ion/Micron ID
41	Opcode 'Request patch'
0x	From bank / x = { 0...4} for banks { Red, Green, Blue, Yellow(user), Edit}
01	Multiple patch
00	Patch number (not applicable for bank)
F7	Sysex end tag

Length is 1 + (56 + 128 * 315) / 7 * 8 + 1 = 46 154 midi bytes
or for Edit bank : 1 + (56 + 4 * 315) / 7 * 8 + 1 = 1 506 midi bytes

Byte format

Midi bytes have 8 bits and values can range from 0 to 255. But the 8th bit designate midi commands (values h80 to hFF). As a consequence, in midi, **data bytes** only have 7 useful bits.

Since people at Alesis prefer to handle more moderne piece of date than 7bits bytes, they convert their data structures containing bit fields, characters, 8bits, 16bits and 32bits integers to midi bytes by a special trick: they cut the structure to 7 bits pieces to pass it as midi message.

The drawback is that if you want to read an Ion sysex message, you have to restore the original structure.

Ion/Micron sysex messages (not counting the hF0 and the ending hFF bytes) data must be converted to '7of8' format. Every packet of 8 consecutive midi bytes must have their bits rearranged to form 7 full bytes. In fact, the first midi byte contains the 7 MSB (most signifient bit) of the following seven bytes.

If the sysex contains the data m1 to m8 (each letter represents a bit, the 0 represents the unused MSB) :

```
m1=0abcdefg
m2=0ttttttt
m3=0uuuuuuu
m4=0vvvvvvv
m5=0wwwwwww
m6=0xxxxxxx
m7=0yyyyyyy
m8=0zzzzzzz
```

The resulting '7of8' bytes b1 to b7 are:

```
b1=attttttt  = (m1 And 64)*2 Or m2
b2=buuuuuuu  = (m1 And 32)*4 Or m3
b3=cvvvvvvv  = (m1 And 16)*8 Or m4
b4=dwwwwwww  = (m1 And 8)*16 Or m5
b5=exxxxxxx  = (m1 And 4)*32 Or m6
b6=fyyyyyyy  = (m1 And 2)*64 Or m7
b7=gzzzzzzz  = (m1 And 1)*128 Or m8
```

The text below only describes structures for '7of8' bytes. The first byte hF0 of the sysex, as well as the ending byte hF7 are removed.
The inbetween 432 midi bytes are converted to 432 / 8 x 7 = 378 full bytes. The index (or offset) goes from 0 to 377.
Note that messages whose length is not a multiple of 8 should have padding zeros.

Byte ordering & signing

When a data spans several bytes, the index is given for the most signifient byte. For exemple:
- a 16 bits value starting at index 17 is computed by: value at index 17 times 256 plus value at index 18.
- a 32 bits value at that index should compute as: (((b(17) x 256) + b(18)) x 256 + b(19)) x 256 + b(20)

Unless otherwise noted, 8 bits and 16 bits values are signed.
8 bits signed : 0 to 127 = 0 to 127, 128 to 255 = -128 to -1
16 bits signed : 0 to 32767 = 0 to 32767, 32768 to 65535 = -32768 to -1

Type codes

In the table below, the following types tag are used
- 8 : value is 8bits signed integer
- A : value is ASCII characters string, subtype gives length
- U : value is 8bits unsigned integer (used only in enveloppe time values and lists)
- 16 : value is 16bits signed integer
- 32 : value is 32bits integer
- b : several bits. Subtype gives bit mask and shift amount. Value = (byte AND mask) / 2^shift or Value = (byte & mask) >> shift
- * : junk, subtype gives number of dummy bytes

Check sum

To control the check sum, do a binary addition of 32 bit integers at byte index 63,67,71...371.
The binary addition drops the carry that would bring the result above maximum 32 bits value (hFFFFFFFF).
I.e hFFFFFFFF + h00000001 = h00000000. It is also called 'rollover' or 'wrap'.
Note: 32 bits are in 'Network order'. On PC, if you 'memcpy' byte array to 32 bit integer array, you will have to inverse byte order with the 'htonl' function for example.

Sample code in VB :

```
Declare Function MemCpy& Lib "Kernel32" Alias "RtlMoveMemory" (pDest As Any, pFrom As Any, ByVal wBytes&)
Declare Function htonl& Lib "ws2_32.dll" (ByVal u&)
```

[...] note: already having '7of8' bytes in array b(0 to 377)

```
' get check sum complement :
Dim check As Long
MemCpy b(15), check, 4
check = htonl(check)

' compute check sum:
Dim i As Long, v(77) As Long, sum As Currency
MemCpy b(63), v(0), 78 * 4
For i = 0 To 77
    sum = sum + htonl(v(i))
    If sum > 2147483647@ Then sum = sum - 4294967296@ ' rollover highest bound
    If sum < -2147483647@ Then sum = sum + 4294967296@ ' rollover lowest bound
Next i

' final control: adding sum and check should gives 0:
sum = sum + check
If sum > 2147483647@ Then sum = sum - 4294967296@ ' rollover highest bound
If sum < -2147483647@ Then sum = sum + 4294967296@ ' rollover lowest bound
If sum = 0@ Then
```

[...]

MESSAGE HEADER				
Idx	type	subtype	role	description
0	8		Manufacturer ID	Midi standard states that if this value equals 0, it should be groupped with the following value
1	8		(continued)	00 0E : Alesis
2	8		Product ID	22 : Ion/Micron
3	8		Contents	{1,,,3} = {patch,setup,pattern}
4	8		Target bank	{0,,,4} = {red,blue,green,yellow,edit}
5	8		Multiple patches	{0,1} = {no(single patch),yes}
6	8		Target slot	{0,,,127} for banks 0-3, {0,,,3} for bank 4
7	A	/ 8	Check tag	Q01SYNTH . Sysex is rejected if not present
15	32		Check sum	This value, when added to the sum of 32 bits integers from index 63 to 371, should equal 0
19	A	/ 4	Version number	"1.06" on Ion with latest firmware, Was "v1.0" on first edition.
23	*	/ 28		
51	32		Check size	315 per patch. Sysex is rejected if message length does not match.
55	*	/ 8		

VOICE				
Idx	type	subtype	role	description
63	A	/ 15	Name	14 characters and and nul terminator (C style string)
78	b	0000 0001 :0	Poly mode	{0,1} = {mono,poly}
	b	0000 0010 :1	Unison	{0,1} = {on, off}
	b	0000 1100 :2	Unison voices	{0...2} = {2,4,8}
	b	1100 0000 :6	Portamento type	{0...3} = {fixed,scaled,gliss fixed,gliss scaled}
79	b	0111 1111 :0	Unison detune	{0...100} = %
	b	1000 0000 :7	Portamento	{0,1} = {on, off}
80	8		Portamento time	{0...126} = exp(x / 18.38514)*10 ms, {127} = 10 000 ms
81	8		Pitch bender mode	{0,1} = {affects all held keys, affects all playing notes}
82	8		Portamento mode	{0,1} = {normal,legato}
83	8		Analog drift	{0...100} %
84	*	/ 2		
86	8		Patch category	{0...10} = {recent,faves,bass,lead,pad,string,brass,key,comp,drum,sfx}

OSC				
Idx	type	subtype	role	description
87	16		Osc1 fine tune	{-999...+999} = { -99.9%,,,+99.9%}
89	16		Osc2 fine tune	{-999...+999} = { -99.9%,,,+99.9%}
91	16		Osc3 fine tune	{-999...+999} = { -99.9%,,,+99.9%}
93	8		Osc1 shape	{0...100} = %
94	8		Osc2 shape	{0...100} = %
94	8		Osc3 shape	{0...100} = %
95	b	0000 0111 :0	Osc1 octave	{0...6} = {-3,,,+3}
	b	1111 0000 :4	Osc1 semitone	{0...14} = {-7,,,+7}
97	b	0000 0011 :0	Osc1 wave	{0...2} = {sin,tri/saw,pulse}
	b	0011 1100 :2	P. bender to Osc1	{0...12} = semitones
	b	0100 0000 :6	Osc1 sync'ing	{0,1} = {on,off}
	b	1000 0000 :7	Sync'ed osc	{0,1} = {osc2,osc2+osc3}
98	b	0000 0001 :0	Sync type	{0,1} = {soft,hard}
	b	0000 1100 :2	FM algorithm	{0...2} = { 3->2->1 , 2->1<-3 , 2->1 }
	b	0011 0000 :4	Osc2 wave	{0...2} = {sin,tri/saw,pulse}
	b	1100 0000 :6	Osc3 wave	{0...2} = {sin,tri/saw,pulse}
100	b	1000 0000 :7	FM type	{0,1} = {linear,exp}
101	b	0000 0111 :0	Osc2 octave	{0...6} = {-3,,,+3}
	b	1111 0000 :4	Osc2 semitone	{0...14} = {-7,,,+7}
102	b	0000 0111 :0	Osc3 octave	{0...6} = {-3,,,+3}
	b	1111 0000 :4	Osc3 semitone	{0...14} = {-7,,,+7}
103	b	0000 1111 :0	P. bender to Osc2	{0...12} = semitones
	b	1111 0000 :4	P. bender to Osc3	{0...12} = semitones
104	16		FM amount	{0...1000} = {0.0%...100.0%}
106	*	/ 2		

PRE FILTER MIX				
Idx	type	subtype	role	description
108	8		Osc1 level	{0...100}
109	8		Osc2 level	{0...100}
110	8		Osc3 level	{0...100}
111	8		Ring level	{0...100}
112	8		Ext-in level	{0...100}
113	8		Osc1 balance	{-50...+50} = { F1 %-F2 % } with: filter1 = 50+x and filter2 = 50-x
114	8		Osc2 balance	{-50...+50} = { F1 %-F2 % } with: filter1 = 50+x and filter2 = 50-x
115	8		Osc3 balance	{-50...+50} = { F1 %-F2 % } with: filter1 = 50+x and filter2 = 50-x
116	8		Ring balance	{-50...+50} = { F1 %-F2 % } with: filter1 = 50+x and filter2 = 50-x
117	8		Ext-in balance	{0} = F1 100% : F2 100% ; {1...100} = F1 100-x : F2 x ; {-1...-100} = F1 -x : F2 100+x
118	8		Noise balance	{-50...+50} = { F1 %-F2 % } with: filter1 = 50+x and filter2 = 50-x
119	8		F1 to F2 level	{0...100} = %
120	b	0111 1111 :0	Noise level	{0...100}
121	b	1000 0000 :7	Noise type	{0,1} = {pink,white}
122	*	/ 4		

FILTER				
Idx	type	subtype	role	description
126	16		F1 cutoff freq	{0...1022} = Herz where freq = exp(x / 147.933647)*20 Hz ; {1023} = 20 000 Hz
128	16		F2 cutoff freq	{0...1022} = Herz where freq = exp(x / 147.933647)*20 Hz ; {1023} = 20 000 Hz
130	8		F1 res	{0...100}
131	8		F2 res	{0...100}
132	8		Filter env to F1	{-100...+100} = %
133	8		Filter env to F2	{-100...+100} = %
134	16		Keytrack to F1 freq	{-100...+200} = %
136	16		Keytrack to F2 freq	{-100...+200} = %
138	8		F2 offset type	{0,1} = {absolute,offset}
139	8		F1 type	{0,20} = {bypass,mg 4-pole lowpass,ob 2-pole lowpass,ob 2-pole bandpass,ob 2-pole highpass,rp 4-pole lowpass,tb 3-pole lowpass,jp 4-pole lowpass,8-pole lowpass,op 4-pole highpass,8ve dual bandpass,6-pole bandpass,phase warp,comb filter 1,comb filter 2,comb filter 3,comb filter 4,volcal formant 1,vocal formant 2,vocal formant 3,bandlimit}
140	8		F2 type	{0,20} = {bypass,mg 4-pole lowpass,ob 2-pole lowpass,ob 2-pole bandpass,ob 2-pole highpass,rp 4-pole lowpass,tb 3-pole lowpass,jp 4-pole lowpass,8-pole lowpass,op 4-pole highpass,8ve dual bandpass,6-pole bandpass,phase warp,comb filter 1,comb filter 2,comb filter 3,comb filter 4,volcal formant 1,vocal formant 2,vocal formant 3,bandlimit}
141	16		F2 offset freq	{-400...+400} = {-4.00...+4.00 octaves}
143	*	/ 3		

POST FILTER MIX				
Idx	type	subtype	role	description
146	8		F1 level	{0...100}
147	8		F2 level	{0...100}
148	8		Unfiltered level	{0...100} This is called Pre-filter level but you might call it 'Unfiltered' or 'Bypass' or 'Dry'
149	8		Unfiltered source	{0...6} = {osc1,osc2,osc3,F1 input,F2 input,ring,noise}
150	8		F1 polarity	{0,1} = {positive,negative}
151	8		F1 pan	{-100...0...100} = { Left 100%...Middle...Right 100% }
152	8		F2 pan	{-100...0...100} = { Left 100%...Middle...Right 100% }
153	8		Unfiltered pan	{-100...0...100} = { Left 100%...Middle...Right 100% }

* note: index 154 to 164 are listed in the Fx & Misc section

ENV (e1=Amp env, e2 = Filter env, e3 = Pitch/mod env)				
Idx	type	subtype	role	description
168	U		e1 attack time	{0...255} = ms where time = exp(x / 23.177415) / 2
169	U		e2 attack time	{0...255} = ms where time = exp(x / 23.177415) / 2
170	U		e3 attack time	{0...255} = ms where time = exp(x / 23.177415) / 2
171	U		e1 decay time	{0...255} = ms where time = exp(x / 23.177415) / 2
172	U		e2 decay time	{0...255} = ms where time = exp(x / 23.177415) / 2
173	U		e3 decay time	{0...255} = ms where time = exp(x / 23.177415) / 2
174	8		e1 sustain level	{0...100}
175	8		e2 sustain level	{-100...100}
176	8		e3 sustain level	{-100...100}
177	*	/ 6		
183	16		e1 release time	{0...254} = ms where time = exp(x / 25.5188668} ; {255} = 30 000 ; {256} = 'held'
185	16		e2 release time	{0...254} = ms where time = exp(x / 25.5188668} ; {255} = 30 000 ; {256} = 'held'
187	16		e3 release time	{0...254} = ms where time = exp(x / 25.5188668} ; {255} = 30 000 ; {256} = 'held'
189	8		Velocity to e1	{0...100} = %
190	8		Velocity to e2	{0...100} = %
191	8		Velocity to e3	{0...100} = %
192	b	0000 0011 :0	e1 loop	{0...3} = {decay,zero,hold,off}
	b	0000 1000 :3	e1 sust pedal	{0,1} = {on,off}
	b	0111 0000 :4	e1 reset	{0,4} = {mono,poly,key mono,key poly,arp mono}
	b	1000 0000 :7	e1 freerun	{0,1} = {release,freerun}
193	b	0000 0011 :0	e2 loop	{0...3} = {decay,zero,hold,off}
	b	0000 1000 :3	e2 sust pedal	{0,1} = {on,off}
	b	0111 0000 :4	e2 reset	{0,4} = {mono,poly,key mono,key poly,arp mono}
	b	1000 0000 :7	e2 freerun	{0,1} = {release,freerun}
194	b	0000 0011 :0	e3 loop	{0...3} = {decay,zero,hold,off}
	b	0000 1000 :3	e3 sust pedal	{0,1} = {on,off}
	b	0111 0000 :4	e3 reset	{0,4} = {mono,poly,key mono,key poly,arp mono}
	b	1000 0000 :7	e3 freerun	{0,1} = {release,freerun}
195	b	0000 0011 :0	e1 attack slope	{0...2} = {linear,exp+,exp-}
	b	0011 0000 :4	e1 decay slope	{0...2} = {linear,exp+,exp-}
	b	1100 0000 :6	e1 release slope	{0...2} = {linear,exp+,exp-}
196	b	0000 0011 :0	e2 attack slope	{0...2} = {linear,exp+,exp-}
	b	0011 0000 :4	e2 decay slope	{0...2} = {linear,exp+,exp-}
	b	1100 0000 :6	e2 release slope	{0...2} = {linear,exp+,exp-}
197	b	0000 0011 :0	e3 attack slope	{0...2} = {linear,exp+,exp-}
	b	0011 0000 :4	e3 decay slope	{0...2} = {linear,exp+,exp-}
	b	1100 0000 :6	e3 release slope	{0...2} = {linear,exp+,exp-}

* note: index 198 to 202 are listed in the Fx & Misc section

LFO				
Idx	type	subtype	role	description
203	16		Lfo1 rate	{0...1022} = Herz where rate = exp(x / 88.85677) / 100 ; {1023} = 1000 Hz
205	16		Lfo2 rate	{0...1022} = Herz where rate = exp(x / 88.85677) / 100 ; {1023} = 1000 Hz
207	8		Wheel 1 to Lf1	{0...100} = %
208	8		Wheel 1 to Lf2	{0...100} = %
209	8		S/H input	{0...114} = {"note-on vel., release vel., key track, m1 wheel, m2 wheel, pitch wheel, sustain pedal, expression pedal, e1 level, e2 level, e3 level, lfo1 sine, lfo1 cosine, lfo1 triangle, lfo1 cos-triangle, lfo1 saw, lfo1 cos-saw, lfo1 square, lfo1 cos-square, lfo2 sine, lfo2 cosine, lfo2 triangle, lfo2 cos-triangle, lfo2 saw, lfo2 cos-saw, lfo2 square, lfo2 cos-square, sh output, voice random, global random, porta level, porta effect, tracking generator, step track, midi channel pressure, midi poly aftertouch, cc1, cc2, cc3, cc4, cc5, cc6, cc7, cc8, cc9, cc10, cc11, cc12, cc13, cc14, cc15, cc16, cc17, cc18, cc19, cc20, cc21, cc22, cc23, cc24, cc25, cc26, cc27, cc28, cc29, cc30, cc31, cc66, cc67, cc68, cc69, cc70, cc71, cc72, cc73, cc74, cc75, cc76, cc77, cc78, cc79, cc80, cc81, cc82, cc83, cc84, cc85, cc86, cc87, cc88, cc89, cc90, cc91, cc92, cc93, cc94, cc95, cc102, cc103, cc104, cc105, cc106, cc107, cc108, cc109, cc110, cc111, cc112, cc113, cc114, cc115, cc116, cc117, cc118, cc119}
210	16		S/H rate	{0...1022} = Herz where rate = exp(x / 88.85677) / 100 ; {1023} = 1000 Hz
212	8		S/H reset	{0...4} = {mono,poly,key mono,key poly,arp mono}
213	b	0000 0001 :0	Lfo1 tempo sync	{0,1} = {on,off}
	b	1111 1110 :1	S/H smoothing	{0...100} = %
214	b	0000 0100 :2	Lfo2 tempo sync	{0,1} = {on,off}
	b	0001 0000 :4	S/H tempo sync	{0,1} = {on,off}
215	8		Lfo1 sync rate	{0...24} = {16,12,$10^{2/3}$,8,6,$5^{1/3}$,4,3,$2^{2/3}$,2,$1^{1/2}$,$1^{1/3}$,1, 3/4,2/3,1/2,3/8,1/3,1/4,3/16,1/6,1/8,3/32,1/12,1/16}
216	8		Lfo2 sync rate	- " -
217	8		S/H sync rate	- " -
218	b	0000 0111 :0	Lfo1 reset	{0...4} = {mono,poly,key mono,key poly,arp mono}
	b	0111 0000 :4	Lfo2 reset	{0...4} = {mono,poly,key mono,key poly,arp mono}

ARP				
Idx	type	subtype	role	description
220	b	0001 1111 :0	Pattern	{0...31} = {*random*,ant march,teletype,acid bass,spitter,samba,chemical, bodiddle,hats on,hats off,rave stomp,carnaval,stutter,a three and a four, samba march,skip to this,skittering,pipeline,fanfare,swinging,chikka-chikka, fee oh fee,robo-shuffle,deliberate,morse code,hit the 4,heart beep,perka, reveille,vari-poly,tango,hesitant}
	b	1110 0000 :5	Tempo multiplier	{0...6} = {1/4,1/3,1/2,1,2,3,4}
221	b	0000 1111 :0	Length	{0...14} = {2...16}
	b	0111 0000 :4	Octave range	{0...4}
222	b	0000 0011 :0	Octave span	{0...2} = {up,down,centered}
	b	0011 1000 :3	Note order	{0...5} = {forward,reverse,trigger,r-n-r in,r-n-r x,oct jump}
	b	1100 0000 :6	Mode	{0...2} = {on,off,latch}
223	16		Tempo	{500...2500} = {50.0 ... 250.0} bpm

MATRIX				
Idx	type	subtype	role	description
230	8		#1 source	{0...115} = same as S/H input list + 'keytrack extreme' at the end
231	8		#2 source	- " -
232	8		#3 source	- " -
233	8		#4 source	- " -
234	8		#5 source	- " -
235	8		#6 source	- " -
236	8		#7 source	- " -
237	8		#8 source	- " -
238	8		#9 source	- " -
239	8		#10 source	- " -
240	8		#11 source	- " -
241	8		#12 source	- " -
242	8		#1 destination	{0...78} = {voice pitch,O1 pitch full,O2 pitch full,O3 pitch full,O1 pitch narrow,O2 pitch narrow,O3 pitch narrow,O1 shape,O2 shape,O3 shape,fm level,O1 level,O2 level,O3 level,ring mod level,noise Level,ext in level,O1 balance,O2 balance,O3 balance,ring mod balance,noise balance,ext in balance,f1->f2 level,portamento Time,unison detune,F1 freq,F1 res,F1 env mod,F1 keytrack,F2 freq,F2 res,F2 env mod,F2 keytrack,lfo1 rate,lfo1 amplitude,lfo2 rate,lfo2 amplitude,s&h rate,s&h smoothing,s&h amplitude,F1 level,F2 level,pre-filter level,F1 pan,F2 pan,pre-filter pan,Drive level,program level,main/aux balance,pan,e1 amplitude,e1 rate,e1 attack,e1 decay,e1 sust time,e1 sust level,e1 release,e2 amplitude,e2 rate,e2 attack,e2 decay,e2 sust time,e2 sust level,e2 release,e3 amplitude,e3 rate,e3 attack,e3 decay,e3 sust time,e3 sust level,e3 release,dummy,fx mix,fx param a,fx param b,fx param c,fx param d,voice pitch narrow}
243	8		#2 destination	- " -
244	8		#3 destination	- " -
245	8		#4 destination	- " -
246	8		#5 destination	- " -
247	8		#6 destination	- " -
248	8		#7 destination	- " -
249	8		#8 destination	- " -
250	8		#9 destination	- " -
251	8		#10 destination	- " -
252	8		#11 destination	- " -
253	8		#12 destination	- " -
254	16		#1 level	{-1000...+1000} = {-100.0%...+100.0%}
256	16		#2 level	- " -
258	16		#3 level	- " -
260	16		#4 level	- " -
262	16		#5 level	- " -
264	16		#6 level	- " -
266	16		#7 level	- " -
268	16		#8 level	- " -
270	16		#9 level	- " -
272	16		#10 level	- " -
274	16		#11 level	- " -
276	16		#12 level	- " -
278	16		#1 offset	{-1000...+1000} = {-100.0%...+100.0%}
280	16		#2 offset	- " -
282	16		#3 offset	- " -
284	16		#4 offset	- " -
286	16		#5 offset	- " -
288	16		#6 offset	- " -
290	16		#7 offset	- " -

MATRIX				
292	16		#8 offset	- " -
294	16		#9 offset	- " -
296	16		#10 offset	- " -
298	16		#11 offset	- " -
300	16		#12 offset	- " -
302	8		Tracking input	{0...79} = same as Matrix #x destination + 'none' at begining
303	8		Tracking points	{0...1} = {12,16}
304	8		Y for X = -16	{-100...+100} = % - default =-100%
305	8		Y for X = -15	{-100...+100} = % - default =-93%
306	8		Y for X = -14	{-100...+100} = % - default =-87%
307	8		Y for X = -13	{-100...+100} = % - default =-81%
308	8		Y for X = -12	{-100...+100} = % - default =-75%
309	8		Y for X = -11	{-100...+100} = % - default =-68%
310	8		Y for X = -10	{-100...+100} = % - default =-62%
311	8		Y for X = -9	{-100...+100} = % - default =-56%
312	8		Y for X = -8	{-100...+100} = % - default =-50%
313	8		Y for X = -7	{-100...+100} = % - default =-43%
314	8		Y for X = -6	{-100...+100} = % - default =-37%
315	8		Y for X = -5	{-100...+100} = % - default =-31%
316	8		Y for X = -4	{-100...+100} = % - default =-25%
317	8		Y for X = -3	{-100...+100} = % - default =-18%
318	8		Y for X = -2	{-100...+100} = % - default =-12%
319	8		Y for X = -1	{-100...+100} = % - default =-6%
320	8		Y for X = 0	{-100...+100} = % - default =0%
321	8		Y for X = 1	{-100...+100} = % - default =6%
322	8		Y for X = 2	{-100...+100} = % - default =12%
323	8		Y for X = 3	{-100...+100} = % - default =18%
324	8		Y for X = 4	{-100...+100} = % - default =25%
325	8		Y for X = 5	{-100...+100} = % - default =31%
326	8		Y for X = 6	{-100...+100} = % - default =37%
327	8		Y for X = 7	{-100...+100} = % - default =43%
328	8		Y for X = 8	{-100...+100} = % - default =50%
329	8		Y for X = 9	{-100...+100} = % - default =56%
330	8		Y for X = 10	{-100...+100} = % - default =62%
331	8		Y for X = 11	{-100...+100} = % - default =68%
332	8		Y for X = 12	{-100...+100} = % - default =75%
333	8		Y for X = 13	{-100...+100} = % - default =81%
334	8		Y for X = 14	{-100...+100} = % - default =87%
335	8		Y for X = 15	{-100...+100} = % - default =93%
336	8		Y for X = 16	{-100...+100} = % - default =100%
337	8		Tracking preset	{0...9} = {custom,bypass,negate,abs val,neg abs,exp+,exp-,zero,maximum,minimum}
338	*	/ 4		

FX & MISC				
Idx	type	subtype	role	description
154	16		Fx2 param A	<micron>
156	16		Fx2 param B	<micron>
159	8		Drive level	<ion> {0...100}
160	8		Drive type	<ion> {0...6} = {bypass,compressor,rmslimiter,tubeoverdrive,distortion,tubeamp,fuzzpedal}
161	8		Output level	{0...100}
162	U		Knob x param I wonder what happens if Knob x param is affected to Knob y param and Knob y param is affected to Knob x param...	{0...161} = { Polyphony, Unison, Unison Detune, Porta, PortaType, Porta Time, Pitch wheel, Analog drift, Osc sync, FM amount, FM type, O1 wave, O1 shape, O1 octave, O1 transpose, O1 pitch, O1 PWhlRange, O2 wave, O2 shape, O2 octave, O2 transpose, O2 pitch, O2 PWhlRange, O3 wave, O3 shape, O3 octave, O3 transpose, O3 pitch, O3 PWhlRange, O1 level, O2 level, O3 level, Ring level, Noise level, ExtIn level, O1 bal, O2 bal, O3 bal, Ring bal, Noise bal, ExtIn bal, Series level, Noise type, F1 type, F1 freq, F1 res, F1 keytrk, F1 env amt, F2 offset, F2 type, F2 freq, F2 res, F2 keytrk, F2 env amt, F1 level, F2 level, Preflt level, F1 pan, F2 pan, Preflt pan, Preflt src, F1 sign, Drive type, Drive level, Prog level, Fx mix, e1 atk time, e1 atk slope, e1 dec time, e1 dec slope, e1 sus time, e1 sus level, e1 rel time, e1 rel slope, e1 velocity, e1 reset, e1 freerun, e1 loop, e1 pedal, e2 atk time, e2 atk slope, e2 dec time, e2 dec slope, e2 sus time, e2 sus level, e2 rel time, e2 rel slope, e2 velocity, e2 reset, e2 freerun, e2 loop, e2 pedal, e3 atk time, e3 atk slope, e3 dec time, e3 dec slope, e3 sus time, e3 sus level, e3 rel time, e3 rel slope, e3 velocity, e3 reset, e3 freerun, e3 loop, e3 pedal, Lfo1 tempo sync, Lfo1 rate, Lfo1 reset, Lfr1 M1 slider, Lfo2 tempo sync, Lfo2 rate, Lfo2 reset, Lfo2 M1 slider, S/H tempo sync, S/H rate, S/H reset, S/H input, S/H smoothing, Tracking, Trk preset, Trk grid, Trk x=-16, Trk x=-15, Trk x=-14, Trk x=-13, Trk x=-12, Trk x=-11, Trk x=-10, Trk x=-9, Trk x=-8, Trk x=-7, Trk x=-6, Trk x=-5, Trk x=-4, Trk x=-3, Trk x=-2, Trk x=-1, Trk x=0, Trk x=1, Trk x=2, Trk x=3, Trk x=4, Trk x=5, Trk x=6, Trk x=7, Trk x=8, Trk x=9, Trk x=10, Trk x=11, Trk x=12, Trk x=13, Trk x=14, Trk x=15, Trk x=16, Category, Knob X param, Knob Y param, Knob Z param, F2 freq offset, Lfo1 rate sync, Lfo2 rate sync, S/H rate sync}
163	U		Knob y param	- " -
164	U		Knob z param	- " -
198	16		Fx2 param C	<micron>
200	16		Fx2 param D	<micron>
202	16		Fx2 param C sync	<micron>
342	8		Fx1 type	{0...6} = {bypass,super phaser,string phaser,theta flanger,thru 0 flanger,chorus,vocoder}
343	8		Fx1 wet-dry mix	{0...100}={ 0-100%...100-0% }
344	8		Fx1 param A	
345	8		Fx1 param B	
346	8		Fx1 param C	
347	8		Fx1 param D	
348	8		Fx1 param E	
349	8		Fx1 param F	
350	8		Fx1 param G	
351	8		Fx1 param H	
352	8		Fx1 Fx2 balance	<micron>
353	8		Fx2 type	<micron> {0...6} = {bypass,mono delay,stereo delay,split delay,hall revb,plate revb,room revb}
354	8		Fx2 param E	<micron>
355	8		Fx2 param F	<micron>

FX & MISC				
356	8		?	-100
357	8		?	-100
358	8		?	-100
359	8		?	-100
360	8		?	-100
361	8		?	-100
362	8		?	0
363	8		?	0
364	8		?	0
365	8		?	0
366	8		?	0
367	8		?	0
368	8		?	0
369	8		?	0
370	8		?	0
371	8		?	0
372	8		?	0
373	8		?	**{0,1} ?**
374	8		?	0 (probably padding to have a message length multiple of 7)
375	8		?	0 (- " -)
376	8		?	0 (- " -)
377	8		?	0 (- " -)

Effect 1 'super phaser'	
A	Feedback {-100...+100} = %
B	Notch frequency : {0...100} = %
C	Lfo rate : { 0...127 } = exp(n / 20.570845484869301) /100 Hz ; 3 digits kept after point, following digits droped without rounding.
D	Lfo depht : {0...100} = %
E	Lfo shape : {0,1} = {sine,triangle}
F	Stages : {0...5} = {4,8,16,32,48,64}
G	Tempo sync : {0,1} = {on (lfo rate uses param H),off (lfo rate uses param C)}
H	Lfo rate : {0...24} = {16,12,$10^{2/3}$,8,6,$5^{1/3}$,4,3,$2^{2/3}$,2,$1^{1/2}$,$1^{1/3}$,1,3/4,2/3,1/2,3/8,1/3,1/4,3/16,1/6,1/8,3/32,1/12,1/16}

Effect 2 'string phaser'	
A	Feedback {-100...+100} = %
B	Notch frequency : {0...100} = %
C	Lfo rate : { 0...127 } = exp(n / 20.570845484869301) /100 Hz
D	Lfo depht : {0...100} = %
E	Lfo shape : {0,1} = {sine,triangle}
F	
G	Tempo sync : {0,1} = {on (lfo rate uses param H),off (lfo rate uses param C)}
H	Lfo rate : {0...24} = {16,12,$10^{2/3}$,8,6,$5^{1/3}$,4,3,$2^{2/3}$,2,$1^{1/2}$,$1^{1/3}$,1,3/4,2/3,1/2,3/8,1/3,1/4,3/16,1/6,1/8,3/32,1/12,1/16}

Effect 3,4 & 5 'theta flanger', 'thru zero flanger' & 'chorus'	
A	Feedback {-100...+100} = %
B	Manual delay : {0...100} = %
C	Lfo rate : { 0...127 } = exp(n / 20.570845484869301) /100 Hz
D	Lfo depht : {0...100} = %
E	Lfo shape : {0,1} = {sine,triangle}
F	
G	Tempo sync : {0,1} = {on (lfo rate uses param H),off (lfo rate uses param C)}
H	Lfo rate : {0...24} = {16,12,$10^{2/3}$,8,6,$5^{1/3}$,4,3,$2^{2/3}$,2,$1^{1/2}$,$1^{1/3}$,1,3/4,2/3,1/2,3/8,1/3,1/4,3/16,1/6,1/8,3/32,1/12,1/16}

Effect 6 'vocoder'	
A	Analysis sens {-100...+100} = %
B	Sibilance boost : {0...100} = %
C	Decay : {0...100} = %
D	Lfo depht : {0...100} = %
E	Band shift : {-100...100} = %
F	Synthesis signal : {0...3} = { ext-in stereo,ext-in left,aux bus,effects send }
G	Analysis signal : {0...3} = { ext-in stereo,ext-in left,aux bus,effects send }
H	Analysis mix : {0...100} = %

Effect 7 'slap-back'	
A	Delay time : {1...80} = ms
B	Regen : {0...100%} = %
C	
D	
E	
F	
G	
H	