

**Disclaimer** : This is version 1.03 of SYSEX specifications for the Waldorf M Wavetable Synthesizer with included erratas and addons according FW version 1.11. (24.12.2025) All bytes, tables, enums have numeration started with 0 to follow C/C++ standard array numerations.

## **Waldorf M SYSEX Messages structure :**

**SYSEX COMMON HEADER** – FIRST 4 bytes always same for all messages

byte 0 = 0xF0,            - sysex  
byte 1 = 0x3E,           - Waldorf Music ID  
byte 2 = 0x30,           - M ID  
byte 3 = 0x00,

## **SYSEX MSG ID**

byte 4 = <MSG\_ID>    - message ID byte

**SYSEX BODY** – variable amount of bytes

## **SYSEX COMMON FOOTER**

last byte = 0xF7        - always 0xF7, end of sysex message

## **Message Ids are the next :**

Request Active Sound Parameter	0x70
Dump Active Sound Parameter	0x71
Request Single Sound	0x74
Dump Single Sound	0x72
Request Active Multi Parameter	0x7A
Dump Multi Parameter	0x7B
Request Multi Arrangement	0x75
Dump Multi Arrangement	0x73
Request System Parameter	0x7C
Dump Global Parameter	0x7D

0x70

0x74

0x7A

0x75

0x7C

## **SYSEX MESSAGES CONTENT (BODY OF MESSAGE)**

Request Active Sound Parameter            0x70

Message Length = 9 bytes

Body bytes:

byte 5 - part 0...3

byte 6 - LSB (of parameter)

byte 7 - MSB (of parameter)

Dump Active Sound Parameter                      0x71  
Message Length = 12 bytes  
Body bytes:

byte 5 - part 0...3  
byte 6 - LSB (of parameter)  
byte 7 - MSB (of parameter)  
byte 8 - LSB (of value)  
byte 9 - MSB (of value)

The transformation formulas from the M parameter representation to the sysex content LSBs and MSBs in the messages **0x70** and **0x71** are next:

For parameter, according to the table below:

```
unsigned short int rawValue14bit = (int)spId + 8192;  
uint8_t LSB = (uint8_t)(rawValue14bit & 0x7F);  
uint8_t MSB = (uint8_t)(rawValue14bit >> 7);
```

where **spID** = Num of parameter. The number of parameters should be taken from the table below (Sound Parameter)

For value:

```
rawValue14bit = rawValue + 8192;  
LSB = (uint8_t)(rawValue14bit & 0x7F);  
MSB = (uint8_t)(rawValue14bit >> 7);
```

Where **rawValue** is raw signed short int, represented current value of parameter. byte 10 - update screen flag (0x1 to update on receiving dump)

## (!) Erratta Addon Begin

Since FE 1.09 encoding of the 2 «Full-Res» parameters

233	TRANSITION_START_SAMPLE	38, UNIPOLAR_FULLRES	Since FW 1.07 Modern mode only
234	TRANSITION_END_SAMPLE	38, UNIPOLAR_FULLRES	Since FW 1.07 Modern mode only

, are encoded the next way :

```
unsigned short int rawValue15bit = rawValue & ~(1 << 14);  
uint8_t LSB15 = (uint8_t)(rawValue15bit & 0x7F);  
uint8_t MSB15 = (uint8_t)(rawValue15bit >> 7);
```

and the top-most bit of 15 bit Full-Res value placed to the bit 6 of the byte 5 (part encoding byte), using the next code :

```
//since FW 1.09 - top bit of 15-bit values goes to the bit 7 of byte 5
```

```

auto correctionBit = (rawValue >> 14) & 0x1;

//part number - first 2 bits
auto tmp = partNum & 0x3;
tmp |= (correctionBit << 6);
sysexSendMessage.data[5] = tmp;

```

(!) Erratta Addon End

**Request Active Multi Parameter** **0x7A**  
**Message Length = 9 bytes**  
**Body bytes:**

byte 5 - part 0...3  
byte 6 - LSB (of parameter)  
byte 7 - MSB (of parameter)

**Dump Multi Parameter** **0x7B**  
**Message Length = 12 bytes**  
**Body bytes:**

byte 5 - part 0...3  
byte 6 - LSB (of parameter)  
byte 7 - MSB (of parameter)  
byte 8 - LSB (of value)  
byte 9 - MSB (of value)

The transformation formulas from the M parameter representation to the sysex content LSBs and MSBs in the messages **0x7A** and **0x7B** are the same as in **0x70** and **0x71** (regular 14-bit conversion).The number of parameters should be taken from the table below (Multi Parameter)

**Request System Parameter** **0x7C**  
**Message Length = 9 bytes**  
**Body bytes:**

byte 5 - 0x0  
byte 6 - LSB (of parameter)  
byte 7 - MSB (of parameter)

**Dump Global Parameter** **0x7D**

byte 5 - 0x0  
byte 6 - LSB (of parameter)  
byte 7 - MSB (of parameter)  
byte 8 - LSB (of value)  
byte 9 - MSB (of value)

The transformation formulas from the M parameter representation to the sysex content LSBs and MSBs in the messages **0x7C** and **0x7D** are the same as in **0x70** and **0x71** (regular 14-bit conversion).The number of parameters should be taken from the table below (Global Parameter)

**Request Single Sound** **0x74**  
**Message Length = 9 bytes**  
**Body bytes:**

byte 5 - 0x0

byte 6 - bank (1 - 16)  
byte 7 - sound (0 - 127)

There the next rule applied - if bank = 0 and sound = 0 - the M will return current active single sound. Otherwise, the M will return exact defined sound from defined bank.

## Dump Single Sound 0x72

**Message Length = 512 bytes**

bytes 6 ... 31 - Sound Name (26 7-bit ASCII Chars). Although the M will display only 23 first chars of the name.

byte 32 - bank  
byte 33 - sound

There the next rule applied - if bank = 0 and sound = 0 - the M will save a dump into current active single sound. Otherwise, the M will save received sound to the defined bank in the defined slot.

byte 34 - 0x0  
byte 35 - 0x0

bytes 36 - 509 - Sound Parameters in the order, defined in the table below. Each parameter occupied two bytes and represented as:

```
unsigned short int rawValue14bit = rawValue + 8192;
```

```
uint8_t LSB = (uint8_t)(rawValue14bit & 0x7F);  
uint8_t MSB = (uint8_t)(rawValue14bit >> 7);
```

```
sysexSendMessage.data[index++] = LSB;  
sysexSendMessage.data[index++] = MSB;
```

where rawValue is raw value representation of the parameter in the M, according to the sound parameter type (see table below), and index is a byte number in the range 36 - 509 (sysexSendMessage.data is an internal data structure of the M)

byte 510 - 0x7F

## (!Errata begin)

Since FE 1.09 encoding of the 2 «Full-Res» parameters

233	TRANSITION_START_SAMPLE	38, UNIPOLAR_FULLRES	Since FW 1.07 Modern mode only
234	TRANSITION_END_SAMPLE	38, UNIPOLAR_FULLRES	Since FW 1.07 Modern mode only

, are encoded the next way :

```

unsigned short int rawValue15bit = rawValue & ~(1 << 14));
uint8_t LSB15 = (uint8_t)(rawValue15bit & 0x7F);
uint8_t MSB15 = (uint8_t)(rawValue15bit >> 7);

```

and the top-most bit of both these 15 bit Full-Res values placed to the bit 1 and bit 2 of the byte 35 , using the next code :

```

// 15 bit parameters MSB extraction (Bits 1 and 2)
rawValueLast = sendSound.getParameter(SoundParameter::Id::TRANSITION_START_SAMPLE_HIRES);
auto correctionBit1 = (rawValueLast >> 14) & 0x1;

rawValueLast = sendSound.getParameter(SoundParameter::Id::TRANSITION_END_SAMPLE_HIRES);
auto correctionBit2 = (rawValueLast >> 14) & 0x1;

```

```

// Smooth scan parameter (bit 0)

```

```

auto rawValueLast = sendSound.getParameter(SoundParameter::Id::SMOOTH_SCAN_WT_RESERVED_START_SYSEX_V1);
auto correctionBit0 = rawValueLast & 0x1;

```

```

// correction byte composing
auto correctionByte = correctionBit0 | (correctionBit1 << 1) | (correctionBit2 << 2);
sysexSendMessage.data[35] = correctionByte;

```

Hence, the next parameter :

237	SMOOTH_SCAN_WT	2, BOOLEAN	Since FW 1.08 Modern mode only
-----	----------------	------------	--------------------------------

, is encoded as bit0 of the byte 35.

(!Errata end)

The M should be in the single mode to receive / send Dump Single Sound (0x72) messages.

**Request Multi Arrangement** 0x75  
**Message Length = 9 bytes**  
**Body bytes:**

```

byte 5 - current
byte 6 - multi (0 - 127)
byte 7 - 0x0

```

There the next rule applied - if current = 1 the M will return current active multi arrangement. Otherwise, the M will return exact defined multi arrangement from defined (in the byte 6) slot.

**Dump Multi Arrangement** 0x73

**Message Length = 320 bytes**

bytes 6 ... 31 - Arrangement Name (26 7-bit ASCII Chars). Although the M will display only 23 first chars of the name.

```

byte 32 - exact (0-1)
byte 33 - slot (0-127)

```

There the next rule applied – if exact = 1 the M will save an arrangement dump into the slot, defined by next byte (slot). Otherwise, the M will save received arrangement to the defined slot.

byte 34 – 0x0

byte 35 – 0x0

bytes 36 – 291 Multi parameters in block of 64 bytes for each part 0...3 subsequently. Each parameter occupied two bytes and represented as:

```
unsigned short int rawValue14bit = instrumentRecordData[param] + 8192;
```

```
uint8_t LSB = (uint8_t)(rawValue14bit & 0x7F);
```

```
uint8_t MSB = (uint8_t)(rawValue14bit >> 7);
```

```
sysexSendMessage.data[index++] = LSB;
```

```
sysexSendMessage.data[index++] = MSB;
```

where

single part, param is a num of Multi Parameter (0...31), according to the table below and index is a byte number in the range 36-291.

instrumentRecordData[param] is a value of the Multi Parameter of the

bytes 292 – 317 are the dummy (0x00) in the M.

byte 318 – 0x7F

The M should be in the Multi mode to receive / send Dump Multi Arrangement (0x73) messages.

**Special Messages Ids are the next :**

Dump (Receive only) Legacy MW1 Sound Bank	0x50
---	------

Dump (Receive only) Legacy MW1 Single Sound	0x42
---	------

For these two, one can refer to the original Microwave 1 sysex specs.

**Special Messages Ids are the next :**

Dump (Receive only) Legacy MW1 Sound Bank	0x50
---	------

Dump (Receive only) Legacy MW1 Single Sound	0x42
---	------

### Addon for ver 1.03

Since FW 1.11 Waldorf M supports realtime sysex of common MTS (MIDI Tuning Standard) format. The only message, available to send / receive is a realtime Single Note tuning change (0x08 0x02) with no bank. The encoding and decoding code is below (according to frequency data standard as stated in <https://midi.org/midi-tuning-updated-specification>)

Encoding code for tuple <note, frequency> :

```
//calculate closest note to frequency in standard tuning
float midiNoteFloat = 69.0f + 12.0f * log2(frequency / 440.0f);

//split to semitones & cents
int semitone = static_cast<int>(floor(midiNoteFloat));
float cents = (midiNoteFloat - semitone) * 100.0f;

//center = 0x2000
int centValue = static_cast<int>(roundf(cents * 8192.0f / 100.0f)) + 8192;

//clamp
centValue = (centValue > 16383) ? 16383 : (centValue < 0) ? 0 : centValue;

uint8_t msb = (centValue >> 7) & 0x7f;
uint8_t lsb = centValue & 0x7f;

sysexSendMessage.data[7] = note;
sysexSendMessage.data[8] = static_cast<uint8_t>(semitone & 0x7f);
sysexSendMessage.data[9] = msb;
sysexSendMessage.data[10] = lsb;
```

Decoding code for tuple <note, frequency> :

```
uint8_t note = message->data[7];
uint8_t semitone = message->data[8];
uint8_t msb = message->data[9];
uint8_t lsb = message->data[10];

int centValue = (msb << 7) | lsb;
float cents = (centValue - 8192) * 100.0f / 8192.0f;
float midiNoteFloat = semitone + (cents / 100.0f);
float frequency = 440.0f * pow(2, (midiNoteFloat - 69.0f) / 12.0f);
```

The header for message according the MTS standard should be next :

```
// header for the message
const uint8_t data[] =
{
    0xF0,
    0x7F, //universal RT
    0x7F, //universal device
    0x08, //ID#1 : MIDI TS
    0x02, //ID#2 : Single note
    0x00, //Tuning program number (0x00) always for the M
    0x01, //Number of messages (i.e. 1 for single note)
};
```

For enabling external control, since FW 1.09 3 new messages were added :

**Request Multi Arrangement Change** **0x66**  
Message Length = 9 bytes

Body bytes:

byte 5 – arrangement to load (M must be in multi mode) in range : 0-127

byte 6,7 – ignored

after receiving this message, if M is in multi mode – requested arrangement will be loaded.

**Request Mode Change** **0x64**  
Message Length = 9 bytes

Body bytes:

byte 5 – oscillators mode in range 0 – 2 (Classic, Modern, VA)

byte 6,7 – ignored

after receiving this message, the mode for the currently active part will be changed.

**Request Button Press** **0x62**  
Message Length = 9 bytes

Body bytes:

byte 5 – button ID (see the enum below)

byte 6,7 – ignored

after receiving this message, the M emulates the button press. Must be used for remote switch of the screens if needed.

M buttons Ids :

enum class ButtonId

```
{
    // Section select
    ARP,           //0
    LFO,           //1
    ENV,           //2
    OSC,           //3
    WAV,           //4
    MIX,           //5
    VCF,           //6
    VCA,           //7
    // Functions
    CANCEL,        //8
    OK,            //9
    SHIFT,         //10
    RECALL,        //11
    MODE,          //12
    STORE,         //13
    PREV,          //14
    NEXT,          //15
    // Misc
    GLIDE,         //16
    ENV_SELECT,    //17
    OSC_SYNC,      //18
};
```



Request State 0x60  
Message Length = 9 bytes  
Body bytes:

byte 5 – answer type (only one support for now = 1)  
byte 6,7 – ignored  
after receiving this message, the M will answer with current memory state (message 0x61, see below).

Answer Memory State 0x61

```
// header for the message
const uint8_t data[] =
{
    0xF0,
    0x3E,
    0x30,
    0x00,
    0x61 // 0x61 for M memory state answer since FW1.09!
};
sysexSendMessage.data[5] = currentMode&0x2;
sysexSendMessage.data[6] = currentBank&0xF;
sysexSendMessage.data[7] = currentSound&0x7F;
sysexSendMessage.data[8] = currentMulti&0x7F;
//reserved
sysexSendMessage.data[9] = 0x0;
sysexSendMessage.data[10] = 0x0;
sysexSendMessage.data[11] = 0xF7;
```

**Sound Parameter - Types :**

The base type for storage/transmission for all sound parameters is signed int 16. Representation for the type either as is, or given in the commentary and/or tables below the main types table.

Parameter Type Num	Parameter Type Name	Min Value	Max Value	Commentary
0	UNIPOLAR	0	127	
1	BIPOLAR	-64	63	
2	BOOLEAN	0	1	0 - off, 1 - on
3	OCTAVE	-2	2	
4	SEMITONE	-12	12	
5	BEND_RANGE	0	48	In semitones
6	PITCHBASE	0	60	Shift from midi note 36 = C-2 = 65,41 Hz. Range will be then in MIDI notes 36-96, represented by it's name (C-2 to C3)
7	MOD_SOURCE	0	36	Values at table below
8	MOD_QUANTIZE	0	7	
9	<i>OSC_DRIFT</i>	<i>0</i>	<i>7</i>	<i>Not used in M</i>
10	OSC_MODE	0	1	Values at table below
11	WAVE_STARTWAVE	0	63	
12	<i>WAVE_MODE</i>	<i>0</i>	<i>1</i>	<i>Not used in M</i>
13	WAVE_TRAVELMODE	0	3	Values at table below
14	WAVETABLE	0	127	Values at table below
15	MIX_LEVEL	0	112	
16	ENV_RATE	0	3	Values at table below
17	<i>ENV_RESOLUTION</i>	<i>0</i>	<i>2</i>	<i>Not used in M</i>
18	ENV_POINT	0	7	
19	LFO_RATE	1	127	
20	LFO_SHAPE	0	4	Values at table below
21	LFO_HUMANIZE	0	7	
22	LFO_DELAY	0	127	
23	LFO_PHASE_SHIFT	0	90	Represented angle from 0 to 180 (i.e. effective phase angle = value * 2)
24	GLIDE	0	1	0 - off, 1 - on
25	<i>TUNING_TABLE</i>	<i>0</i>	<i>4</i>	<i>Not used in M</i>
26	ARP_ACTIVE	0	1	0 - off, 1 - on
27	ARP_TEMPO	30	300	
28	ARP_MODE	0	3	Values at table below
29	ARP_CLOCK	0	11	Values at table below
30	ARP_DIRECTION	0	3	Values at table below
31	<i>ARP_SORT_ORDER</i>	<i>0</i>	<i>6</i>	<i>Not used in M</i>

32	ARP_VELOCITY_MODE	0	6	Not used in M
33	ARP_PATTERN_LENGTH	0	15	Not used in M
34	ARP_OCTAVE	0	9	
35	ARP_PATTERN_NUMBER	0	15	
36	ARP_MAX_NOTES	0	15	Not used in M
37	TRANSITION_FILENUM	0	63	Transition's Slot num in M
38	UNIPOLAR_FULLRES	0	32767	
39	WAVE_PHASE	0	127	0 – free, 1-127 – start sample num withing wave.
40	LFO_MIDI_SYNC_VALUE	0	28	Values at table below
41	OSC_ASICBUG	0	1	0 – off, 1 – on
42	TIMELEVELMODMODE	0	1	0 – M, 1 – MW. Affects modulation formula of T/L. Details in user manual
43	ENGINEPLAYMODE	0	2	Values at table below
44	ALLOCATOR_STEAL_TYPE	0	3	Values at table below
45	DIGIVCF_FILTER_TYPE	0	17	Values at table below
46	LFO_RANGE_TYPE	0	2	Values at table below
47	TRANSITION_LOOP_TYPE_TYPE	0	4	Values at table below
48	DEREZ	0	14	Bits, removed from 16 bit range.
49	GLIDE_MODE_TYPE	0	1	0 – lin, 1 – exp
50	MVA_ALGORYTHM	0	7	Values at table below

#### Value representation tables for certain types

##### 44, ALLOCATOR\_STEAL\_TYPE

Value	Represented in M	Meaning
0	"M.Late"	Most late note from sounding notes (latest pressed)
1	"M.Early"	Most early note from sounding notes (earliest pressed)
2	"Lowest"	Lowest note from sounding notes
3	"Highest"	Highest note from sounding notes

##### 29, ARP\_CLOCK

Value	Represented in M	Meaning
0	"1/1"	
1	"1/2"	
2	"1/4"	
3	"1/8dot"	
4	"1/4trp"	
5	"1/8"	
6	"1/8trp"	

7	"1/16"	
8	"1/16trp"	
9	"1/32"	
10	"Fast"	
11	"Max"	

### 30, ARP\_DIRECTION

Value	Represented in M	Meaning
0	"Up"	
1	"Down"	
2	"AltUp"	Alternative Up
3	"AltDown"	Alternative Down

### 28, ARP\_MODE

Value	Represented in M	Meaning
0	"Off"	
1	"Normal"	
2	"OneShot"	
3	"Hold"	

### 45, DIGIVCF\_FILTER\_TYPE

Value	Represented in M	Meaning
0	"LP12"	
1	"BP12"	
2	"HP12"	
3	"Notch12"	
4	"LP24"	
5	"BP24"	
6	"HP24"	
7	"Notch24"	
8	"Sin+LP"	Sine wave as Shaper curve followed 12dB LP filter
9	"Dual L/BP"	Dual 12dB LP + 12dB BP in parallel, Extra sets the Frequency offset of BP filter
10	"Bstop12"	Band stop 12db roll-off
11	"S&H+LP12"	S&H for Mixer output, followed with 12dB LP filter, Extra set S&H period.
12	"WSHAPER"	Wavetable-based waveshaper. Shaper curve taken from wave of wavetable 2. Extra set the Wave (Extra / 2)
13	"FM+LP"	FM-capable 12dB LP filter. OSC2 output modulates filter's frequency, Extra set modulation depth.
14	"S&H+HP12"	S&H for Mixer output, followed with 12dB HP filter, Extra set S&H period.
15	"FM+HP"	FM-capable 12dB HP filter. OSC2 output modulates

		filter's frequency, Extra set modulation depth.
16	"FBOOST"	Frequency boost filter. Extra set depth of boost. Roll-off slope around center frequency is 12dB.
17	"MSHAPER"	M-Style Waveshaper. Wavetable-based waveshaper. Shaper curve taken from the ACTUAL wave of wavetable 2. Extra set the Depth of the shaper effect.
18	"Resocomp"	Software resonance compensation – not a filter. Extra parameter regulates depth of effect. Amplifies the input signal for -3dB proportionally to the actual resonance of the voice.

#### 43, ENGINEPLAYMODE

Value	Represented in M	Meaning
0	"Poly"	Standard Polyphonic Mode
1	"MRetrig"	Monophonic Retrig
2	"MLegato"	Monophonic Legato
3	"MPorta"	Monophonic Portamento

#### 16, ENV\_RATE

Value	Represented in M	Meaning
0	"fast"	Rates scaled to 0.01S – 1S
1	"normal"	1x – Standard M rates
2	"slow"	Rates scaled to 0.1S – 1000S

#### 46, LFO\_RANGE\_TYPE

Value	Represented in M	Meaning
0	"fast"	1/10x rates
1	"normal"	1x – Standard M rates
2	"slow"	4x rates

#### 20, LFO\_SHAPE

Value	Represented in M	Meaning
0	"Sine"	
1	"Triangle"	
2	"Pulse"	
3	"Random"	
4	"S&H"	

#### 40, LFO\_MIDI\_SYNC\_VALUE

Value	Represented in M	Meaning
0	"1024bars"	
1	"512 bars"	
2	"256 bars"	
3	"192 bars"	

4	"128 bars"	
5	"96 bars"	
6	"64 bars"	
7	"48 bars"	
8	"32 bars"	
9	"24 bars"	
10	"16 bars"	
11	"12 bars"	
12	"8 bars"	
13	"6 bars"	
14	"4 bars"	
15	"3 bars"	
16	"2 bars"	
17	"6/4"	
18	"4/4"	
19	"3/4"	
20	"1/2"	
21	"3/8"	
22	"1/4"	
23	"3/16"	
24	"1/8"	
25	"3/32"	
26	"1/16"	
27	"3/64"	
28	"1/32"	

## 7, MOD\_SOURCE

Value	Represented in M	Meaning
0	"off"	
1	"ModWh"	Modulation Wheel, CC1
2	"Pbend"	Pitch Bender
3	"LF01"	
4	"LF02"	
5	"AmpEnv"	
6	"FltEnv"	
7	"WavEnv"	
8	"FreeEnv"	

9	"LF01Env"	AD Envelope of the LF01
10	"Sus Pd1"	
11	"Vol Ctr"	
12	"Pan Ctr"	
13	"BrthCtr"	
14	"FootCtr"	
15	"ExprCtr"	
16	"Ctr A"	
17	"Ctr B"	
18	"Ctr C"	
19	"Ctr D"	
20	"Ctr W"	
21	"Ctr X"	
22	"Ctr Y"	
23	"Ctr Z"	
24	"KTrack"	Keytrack, calculated in the range of -1.0 – 1.0 as offset from the middle C. For the middle C = 0.0.
25	"Velo"	Note On velocity
26	"RelVelo"	Note Off velocity
27	"AftTch"	
28	"PlyPres"	Polyphonic aftertouch
29	"Gl.LFO"	Global LFO
30	"min"	
31	"MAX"	
32	"Inverse"	-1 / 1 / -1 / 1 ...etc sequence of 1s, and for each new Note ON the sign will be changed.
33	"CoinFlip"	Random 0 or 1 coin toss at the Note ON
34	"Random"	Random Value within -1.0 ... 1.0 range set once at the Note ON and stay until next retrig of this voice.
35	"Gate"	1 if note is ON, else 0
36	"V.Noise"	Random Value within -0.5 ... 0.5 range, constantly changed (a-la Noise as modulator). Independent for each note.

#### 10, OSC\_MODE

Value	Represented in M	Meaning
0	"MW1"	Microwave 1 ASIC emulation
1	"MW2"	Microwave 2 DSP bandlimited OSC emulation

#### 47, TRANSITION\_LOOP\_TYPE\_TYPE

Value	Represented in M	Meaning
-------	------------------	---------

0	"OFF"	
1	"FWD"	Forward loop
2	"BWD"	Backward (reverse) loop
3	"ST+FWD"	Start from the sample 0, then forward loop
4	"ST+BWD"	Start from the sample 0, then reverse loop

#### 50, MVA\_ALGORITHM\_TYPE

Value	Represented in M	Meaning
0	Varywave	Variable pair of VA waves
1	CS-80 Saw	CS-80 like sawtooth with variable step and shape
2	Waveshpr	Waveshaping oscillator
3	Triple Saw	Three detuned saw
4	S/T fold	Sine / triangle to wavefolder
5	FM 2 OPs	Simple 2 OP FM
6	FM Fback	2 OP FM + feedback
7	Duonoise	Variable noise with resonant bandpass

#### 14, WAVETABLE

Value	Represented in M	Meaning
0	"Resonant"	
1	"Resonant2 LP"	
2	"Mallet Synth"	
3	"Square Sweep"	
4	"Bell"	
5	"Pulse Sweep"	
6	"Saw Sweep"	
7	"Mellow Saw"	
8	"Feedback"	
9	"Add Harm"	
10	"Resonant3 HP"	
11	"Wind Synth"	
12	"High Harm"	
13	"Clipper"	
14	"Organ Synth"	
15	"Square Saw"	
16	"Formant1"	
17	"Polated"	
18	"Transient"	
19	"Elec Piano"	



20	"Robotic"	
21	"Strong Harm"	
22	"Perc Organ"	
23	"Clip'n'Sweep"	
24	"Reso Harm"	
25	"2 Echoes"	
26	"Formant2"	
27	"Formant Voc"	
28	"Micro Sync"	
29	"Micro PWM"	
30	"Glassy"	
31	"Square HP"	
32	"Saw Sync1"	
33	"Saw Sync2"	
34	"Saw Sync3"	
35	"Pulse Sync1"	
36	"Pulse Sync2"	
37	"Pulse Sync3"	
38	"Sine Sync1"	
39	"Sine Sync2"	
40	"Sine Sync3"	
41	"PWM Pulse"	
42	"PWM Saw"	
43	"Fuzz Wave"	
44	"Distorted"	
45	"Heavy Fuzz"	
46	"Fuzz Sync"	
47	"KarpStrong1"	
48	"KarpStrong2"	
49	"KarpStrong3"	
50	"1-2-3-4-5"	
51	"19/20"	
52	"Wavetrip1"	
53	"Wavetrip2"	
54	"Wavetrip3"	
55	"Wavetrip4"	
56	"Male voice"	
57	"Low piano"	
58	"Resonant sweep"	

59	"Xmas bell"	
60	"FM Piano"	
61	"FAT Organ"	
62	"Vibes"	
63	"Chorus2"	
64	"True PWM"	Microwave 2 added table, assymetrical 128 samples, 60 waves PWM representation
65	"Wave23 SawSync"	PPG Wave 2.3 Table 29 Saw Sync
66	"Wave23 UpWaves"	PPG Wave 2.3 Tavle 30 Upper Waves
67	"Q Alt1"	Waveset Alt1 Waldorf Q
68	"Q Alt2"	Waveset Alt2 Waldorf Q
69	"Wave22 Clipper"	«Uncorrupted» clipper table from PPG Wave 2.2
70	"Wave22 PSax"	Piano Sax table from PPG Wave 2.2
71	"Wave23 PSax"	Piano Sax table from PPG Wave 2.3
72	"Mayschoss Wine"	
73	"Parks of Bonn"	
74	"Niederzissen"	
75	"Kripper Ferry"	
76	"Sinziger Organ"	
77	"La Palma's Sun"	
78	"Remagen Works"	
79	"Ahrtal Walks"	
80	"3V Multi Aah"	
81	"3V Clavinet"	
82	"3V Cncert Gtar"	
83	"3V Celeste"	
84	"3V Marimba"	
85	"3V Bell"	
86	"3V Grand Piano"	
87	"3V ElPiano"	
88	"3V Trombone"	
89	"3V Pluck Bass"	
90	"3V GL Bass"	
91	"3V Harp"	
92	"3V Sitar"	
93	"3V ElGitar TTR"	
94	"3V Y-Synth"	
95	"3V HausOrgan"	
96-127	"User WTxX"	XX = Value - 96, i.e. 00 - 31

### 13, WAVE\_TRAVELMODE

Value	Represented in M	Meaning
0	"An.ON"	Analog ON – standard wavescan mode – waves are in range 0 – 63
1	"An.Off"	Analog OFF – analog waves restricted, wavescan in range 0-60
2	"An.Only"	Analog Only – only analog waves allowed, wavescan in range 61-63
3	"Circular"	Wavescan with wrap in both directions. I.e. if desired wave below 0 – it wrapped to 63 and vice versa.

### Sound parameters – Parameters List (at the state of FW 1.11)

Num	Sound Parameter Name	Sound Parameter Type	Commentary
0	OSC1_OCTAVE	3, OCTAVE	
1	OSC1_SEMITONE	4, SEMITONE	
2	OSC1_DETUNE	1, BIPOLAR	
3	OSC1_BEND_RANGE	5, BEND_RANGE	
4	OSC1_PITCH_MODE	2, BOOLEAN	
5	OSC1_FIXEDPITCH	6, PITCHBASE	Since FW 1.06
6	OSC1_MOD1_SOURCE	7, MOD_SOURCE	
7	OSC1_MOD1_CONTROL	7, MOD_SOURCE	
8	OSC1_MOD1_AMOUNT	1, BIPOLAR	
9	OSC1_MOD2_SOURCE	7, MOD_SOURCE	
10	OSC1_MOD2_AMOUNT	1, BIPOLAR	
11	OSC1_MOD2_QUANTIZE	8, MOD_QUANTIZE	
12	DIGIVCF_CUTOFF	0, UNIPOLAR	Since FW 1.06
13	DIGIVCF_RESONANCE	0, UNIPOLAR	Since FW 1.06
14	OSC2_OCTAVE	3, OCTAVE	
15	OSC2_SEMITONE	4, SEMITONE	
16	OSC2_DETUNE	1, BIPOLAR	
17	OSC2_BEND_RANGE	5, BEND_RANGE	
18	OSC2_PITCH_MODE	2, BOOLEAN	
19	OSC2_FIXEDPITCH	6, PITCHBASE	
20	OSC2_MOD1_SOURCE	7, MOD_SOURCE	
21	OSC2_MOD1_CONTROL	7, MOD_SOURCE	
22	OSC2_MOD1_AMOUNT	1, BIPOLAR	
23	OSC2_MOD2_SOURCE	7, MOD_SOURCE	
24	OSC2_MOD2_AMOUNT	1, BIPOLAR	
25	OSC2_MOD2_QUANTIZE	8, MOD_QUANTIZE	
26	OSC_ASICBUG	2, BOOLEAN	Classic mode only
27	DIGIVCF_TYPE	45, DIGIVCF_FILTER_TYPE	Since FW 1.06
28	OSC2_SYNC_TO_OSC1	2, BOOLEAN	Modern mode only
29	OSC2_LINK_OSC1_MODS	2, BOOLEAN	
30	WAVE1_WAVETABLE	14, WAVETABLE	
31	WAVE1_STARTWAVE	11, WAVE_STARTWAVE	
32	WAVE1_STARTPHASE	39, WAVE_PHASE	
33	WAVE1_ENV_AMOUNT	1, BIPOLAR	
34	WAVE1_ENV_VELOCITY	1, BIPOLAR	
35	WAVE1_KEYTRACK	1, BIPOLAR	

36	WAVE1_MOD1_SOURCE	7, MOD_SOURCE	
37	WAVE1_MOD1_CONTROL	7, MOD_SOURCE	
38	WAVE1_MOD1_AMOUNT	1, BIPOLAR	
39	WAVE1_MOD2_SOURCE	7, MOD_SOURCE	
40	WAVE1_MOD2_AMOUNT	1, BIPOLAR	
41	WAVE1_WAVETRAVELMODE	13, WAVE_TRAVELMODE	
42	DIGIVCF_MOD2_SOURCE	7, MOD_SOURCE	Since FW 1.06
43	WAVE2_WAVETABLE	14, WAVETABLE	
44	WAVE2_STARTWAVE	11, WAVE_STARTWAVE	
45	WAVE2_STARTPHASE	39, WAVE_PHASE	
46	WAVE2_ENV_AMOUNT	1, BIPOLAR	
47	WAVE2_ENV_VELOCITY	1, BIPOLAR	
48	WAVE2_KEYTRACK	1, BIPOLAR	
49	WAVE2_MOD1_SOURCE	7, MOD_SOURCE	
50	WAVE2_MOD1_CONTROL	7, MOD_SOURCEs	
51	WAVE2_MOD1_AMOUNT	1, BIPOLAR	
52	WAVE2_MOD2_SOURCE	7, MOD_SOURCE	
53	WAVE2_MOD2_AMOUNT	1, BIPOLAR	
54	WAVE2_WAVETRAVELMODE	13, WAVE_TRAVELMODE	
55	DIGIVCF_MOD2_AMOUNT	1, BIPOLAR	Since FW 1.06
56	WAVE2_LINK_WAVE1_MODS	2, BOOLEAN	
57	MIX_OSC1_LEVEL	15, MIX_LEVEL	
58	MIX_OSC2_LEVEL	15, MIX_LEVEL	
59	MIX_RINGMOD_LEVEL	15, MIX_LEVEL	Modern Mode only
60	MIX_NOISE_LEVEL	15, MIX_LEVEL	
61	MIX_OSC1_MOD_SOURCE	7, MOD_SOURCE	
62	MIX_OSC1_MOD_AMOUNT	1, BIPOLAR	
63	MIX_OSC2_MOD_SOURCE	7, MOD_SOURCE	
64	MIX_OSC2_MOD_AMOUNT	1, BIPOLAR	
65	MIX_RINGMOD_MOD_SOURCE	7, MOD_SOURCE	Modern Mode only
66	MIX_RINGMOD_MOD_AMOUNT	1, BIPOLAR	Modern Mode only
67	MIX_NOISE_MOD_SOURCE	7, MOD_SOURCE	
68	MIX_NOISE_MOD_AMOUNT	1, BIPOLAR	
69	VCF_CUTOFF	0, UNIPOLAR	
70	DIGIVCF_ENABLED	2, BOOLEAN	since FW 1.06
71	VCF_RESONANCE	0, UNIPOLAR	
72	VCF_ENV_AMOUNT	1, BIPOLAR	
73	VCF_ENV_VELOCITY	1, BIPOLAR	
74	VCF_KEYTRACK	1, BIPOLAR	
75	VCF_MOD1_SOURCE	7, MOD_SOURCE	
76	VCF_MOD1_CONTROL	7, MOD_SOURCE	
77	VCF_MOD1_AMOUNT	1, BIPOLAR	
78	VCF_MOD2_SOURCE	7, MOD_SOURCE	
79	VCF_MOD2_AMOUNT	1, BIPOLAR	
80	VCF_RES_MOD_SOURCE	7, MOD_SOURCE	
81	VCF_RES_MOD_AMOUNT	1, BIPOLAR	
82	INSTRUMENT_VOLUME	0, UNIPOLAR	
83	VCA_ENV_AMOUNT	1, BIPOLAR	
84	VCA_ENV_VELOCITY	1, BIPOLAR	
85	VCA_KEYTRACK	1, BIPOLAR	
86	VCA_MOD1_SOURCE	7, MOD_SOURCE	
87	VCA_MOD1_CONTROL	7, MOD_SOURCE	
88	VCA_MOD1_AMOUNT	1, BIPOLAR	
89	VCA_MOD2_SOURCE	7, MOD_SOURCE	
90	VCA_MOD2_AMOUNT	1, BIPOLAR	
91	VCA_PAN	1, BIPOLAR	

92	VCA_PAN_MOD_SOURCE	7, MOD_SOURCE	
93	VCA_PAN_MOD_AMOUNT	1, BIPOLAR	
94	AENV_ATTACK	0, UNIPOLAR	
95	AENV_DECAY	0, UNIPOLAR	
96	AENV_SUSTAIN	0, UNIPOLAR	
97	AENV_RELEASE	0, UNIPOLAR	
98	AENV_ATTACK_MOD_SOURCE	7, MOD_SOURCE	
99	AENV_ATTACK_MOD_AMOUNT	1, BIPOLAR	
100	AENV_DECAY_MOD_SOURCE	7, MOD_SOURCE	
101	AENV_DECAY_MOD_AMOUNT	1, BIPOLAR	
102	AENV_SUSTAIN_MOD_SOURCE	7, MOD_SOURCE	
103	AENV_SUSTAIN_MOD_AMOUNT	1, BIPOLAR	
104	AENV_RELEASE_MOD_SOURCE	7, MOD_SOURCE	
105	AENV_RELEASE_MOD_AMOUNT	1, BIPOLAR	
106	TIMELEVELMODMODE	42, TIMELEVELMODMODE	Changes between MW and M modulation formulae
107	ENV_TIMERESOLUTION	16, ENV_RATE	
108	FENV_DELAY	0, UNIPOLAR	
109	FENV_ATTACK	0, UNIPOLAR	
110	FENV_DECAY	0, UNIPOLAR	
111	FENV_SUSTAIN	0, UNIPOLAR	
112	FENV_RELEASE	0, UNIPOLAR	
113	FENV_DELAY_MOD_SOURCE	7, MOD_SOURCE	
114	FENV_DELAY_MOD_AMOUNT	1, BIPOLAR	
115	FENV_ATTACK_MOD_SOURCE	7, MOD_SOURCE	
116	FENV_ATTACK_MOD_AMOUNT	1, BIPOLAR	
117	FENV_DECAY_MOD_SOURCE	7, MOD_SOURCE	
118	FENV_DECAY_MOD_AMOUNT	1, BIPOLAR	
119	FENV_SUSTAIN_MOD_SOURCE	7, MOD_SOURCE	
120	FENV_SUSTAIN_MOD_AMOUNT	1, BIPOLAR	
121	FENV_RELEASE_MOD_SOURCE	7, MOD_SOURCE	
122	FENV_RELEASE_MOD_AMOUNT	1, BIPOLAR	
123	FREEENV_LOOP	2, BOOLEAN	Belonged to Free Envelope!
124	FREEENV_LOOP_AT_RELEASE	2, BOOLEAN	Belonged to Free Envelope!
125	WENV_P1_TIME	0, UNIPOLAR	
126	WENV_P1_LEVEL	0, UNIPOLAR	
127	WENV_P2_TIME	0, UNIPOLAR	
128	WENV_P2_LEVEL	0, UNIPOLAR	
129	WENV_P3_TIME	0, UNIPOLAR	
130	WENV_P3_LEVEL	0, UNIPOLAR	
131	WENV_P4_TIME	0, UNIPOLAR	
132	WENV_P4_LEVEL	0, UNIPOLAR	
133	WENV_P5_TIME	0, UNIPOLAR	
134	WENV_P5_LEVEL	0, UNIPOLAR	
135	WENV_P6_TIME	0, UNIPOLAR	
136	WENV_P6_LEVEL	0, UNIPOLAR	
137	WENV_P7_TIME	0, UNIPOLAR	
138	WENV_P7_LEVEL	0, UNIPOLAR	
139	WENV_P8_TIME	0, UNIPOLAR	
140	WENV_P8_LEVEL	0, UNIPOLAR	
141	WENV_TIME_MOD_SOURCE	7, MOD_SOURCE	
142	WENV_TIME_MOD_AMOUNT	1, BIPOLAR	
143	WENV_LEVEL_MOD_SOURCE	7, MOD_SOURCE	

144	WENV_LEVEL_MOD_AMOUNT	1, BIPOLAR	
145	WENV_KEYOFF_POINT	18, ENV_POINT	
146	WENV_LOOP_MODE	2, BOOLEAN	
147	WENV_LOOP_BEGIN	18, ENV_POINT	
148	WENV_LOOP_END	18, ENV_POINT	
149	FREEENV_TIME1	0, UNIPOLAR	
150	FREEENV_LEVEL1	0, UNIPOLAR	
151	FREEENV_TIME2	0, UNIPOLAR	
152	FREEENV_LEVEL2	0, UNIPOLAR	
153	FREEENV_TIME3	0, UNIPOLAR	
154	FREEENV_SUSTAIN_LEVEL	0, UNIPOLAR	
155	FREEENV_RELEASE_TIME	0, UNIPOLAR	
156	FREEENV_RELEASE_LEVEL	0, UNIPOLAR	
157	FREEENV_TIME_MOD_SOURCE	7, MOD_SOURCE	
158	FREEENV_TIME_MOD_AMOUNT	1, BIPOLAR	
159	FREEENV_LEVEL_MOD_SOURCE	7, MOD_SOURCE	
160	FREEENV_LEVEL_MOD_AMOUNT	1, BIPOLAR	
161	LFO1_RATE	19, LFO_RATE	
162	LFO1_SHAPE	20, LFO_SHAPE	
163	LFO1_SYMMETRY	1, BIPOLAR	
164	LFO1_HUMANIZE	21, LFO_HUMANIZE	
165	LFO1_RATE_MOD_SOURCE	7, MOD_SOURCE	
166	LFO1_RATE_MOD_AMOUNT	1, BIPOLAR	
167	LFO1_LEVEL_MOD_SOURCE	7, MOD_SOURCE	No amount, always full (as +63)
168	LFO1_SYNC	2, BOOLEAN	
169	LFO1_EG_DELAY	22, LFO_DELAY	
170	LFO1_EG_ATTACK	0, UNIPOLAR	
171	LFO1_EG_DECAY	0, UNIPOLAR	
172	LFO1_RANGE	46, LFO_RANGE_TYPE	
173	LFO2_RATE	19, LFO_RATE	
174	LFO2_SHAPE	20, LFO_SHAPE	
175	LFO2_SYMMETRY	1, BIPOLAR	
176	LFO2_HUMANIZE	21, LFO_HUMANIZE	
177	LFO2_PHASE_SHIFT	23, LFO_PHASE_SHIFT	
178	LFO2_RANGE	46, LFO_RANGE_TYPE	
179	GLIDE	24, GLIDE	
180	GLIDE_RATE	0, UNIPOLAR	
181	GLIDE_MODE	49, GLIDE_MODE_TYPE	
182	GLOBAL_LFO_HUMANISE	21, LFO_HUMANIZE	Since FW 1.08
183	GLOBAL_LFO_RATE	19, LFO_RATE	Since FW 1.08
184	ARP_MODE	28, ARP_MODE	
185	ARP_CLOCK	29, ARP_CLOCK	
186	ARP_TEMPO	27, ARP_TEMPO	
187	ARP_PATTERN_IDX	35, ARP_PATTERN_NUMBER	
188	ARP_PLAY_DIR	30, ARP_DIRECTION	
189	ARP_OCTAVE_RANGE	34, ARP_OCTAVE	
190	___ARP_LENGTH		NOT USED IN M! RESERVED FOR FUTURE
191	___ARP_SORT_ORDER_RESERVED		NOT USED IN M! RESERVED FOR FUTURE
192	___ARP_TIMING_FACTOR_RESERVED		NOT USED IN M! RESERVED FOR FUTURE
193	___ARP_VELO_MODE_RESERVED		NOT USED IN M! RESERVED FOR FUTURE
194	___ARP_PATTERN_LENHT_RESERVED		NOT USED IN M!

			RESERVED FOR FUTURE
195	___ARP_PATTERN_RESET_RESERVED		NOT USED IN M! RESERVED FOR FUTURE
196	___ARP_MAX_NOTES_RESERVED		NOT USED IN M! RESERVED FOR FUTURE
197	GLOBAL_LFO_ENABLE	2, BOOLEAN	
198	GLOBAL_LFO_MIDISYNCD	2, BOOLEAN	
199	GLOBAL_LFO_MIDISYNCDVALUE	40, LFO_MIDI_SYNC_VALUE	
200	MMWMODE	10, OSC_MODE	
201	PLAYMODE	43, ENGINEPLAYMODE	
202	VOICESTEALMODE	44, ALLOCATOR_STEAL_TYPE	
203	GLOBAL_LFO_SHAPE	20, LFO_SHAPE	Since FW 1.08
204	DIGIVCF_EXTRA_MOD_SRC	7, MOD_SOURCE	Since FW 1.10
205	DIGIVCF_EXTRA_MOD_AMOUNT	1, BIPOLAR	Since FW 1.10
206	WAVE1_METAPARAM_ALGO	50, MVA_ALGORITHM_TYPE	Since FW 1.10 MVA Mode only
207	WAVE1_METAPARAM	0, UNIPOLAR	Since FW 1.10 MVA Mode only
208	WAVE1_METAPARAM_MOD_SRC	7, MOD_SOURCE	Since FW 1.10 MVA Mode only
209	WAVE1_METAPARAM_MOD_AMOUNT	1, BIPOLAR	Since FW 1.10 MVA Mode only
210	WAVE2_METAPARAM_ALGO	50, MVA_ALGORITHM_TYPE	Since FW 1.10 MVA Mode only
211	WAVE2_METAPARAM	0, UNIPOLAR	Since FW 1.10 MVA Mode only
212	WAVE2_METAPARAM_MOD_SRC	7, MOD_SOURCE	Since FW 1.10 MVA Mode only
213	WAVE2_METAPARAM_MOD_AMOUNT	1, BIPOLAR	Since FW 1.10 MVA Mode only
214	ANVCF_MPE_PB_TRACK_ENABLE	2, BOOLEAN	Since FW 1.10
215	ANVCF_MPE_PB_TRACK_AMOUNT	0, UNIPOLAR	Since FW 1.10
216	FM_ENABLE_MM	2, BOOLEAN	Since FW 1.11 Modern mode only
217	FM_DEPTH_MM	0, UNIPOLAR	Since FW 1.11 Modern mode only
218	GLIDE_MOD_SOURCE	7, MOD_SOURCE	Since FW 1.11
219	GLIDE_MOD_AMOUNT	1, BIPOLAR	Since FW 1.11
220	DIGIVCF_ENV_AMOUNT	1, BIPOLAR	Since FW 1.06 Modern mode only
221	DIGIVCF_ENV_VELOCITY	1, BIPOLAR	Since FW 1.06 Modern mode only
222	DIGIVCF_KEYTRACK	1, BIPOLAR	Since FW 1.06 Modern mode only
223	DIGIVCF_MOD1_SOURCE	7, MOD_SOURCE	Since FW 1.06 Modern mode only
224	DIGIVCF_MOD1_CONTROL	1, BIPOLAR	Since FW 1.06 Modern mode only
225	DIGIVCF_MOD1_AMOUNT	7, MOD_SOURCE	Since FW 1.06 Modern mode only
226	DIGIVCF_RESMOD_SOURCE	1, BIPOLAR	Since FW 1.06 Modern mode only
227	DIGIVCF_RESMOD_AMOUNT	7, MOD_SOURCE	Since FW 1.06 Modern mode only
228	DIGIVCF_EXTRAPARAM	0, UNIPOLAR	Since FW 1.06 Modern mode only

229	TRANSITION_ENABLE	2, BOOLEAN	Since FW 1.07 Modern mode only
230	TRANSITION_NUMBER	37, TRANSITION_FILENUM	Since FW 1.07 Modern mode only
231	TRANSITION_BASE	6, PITCHBASE	Since FW 1.07 Modern mode only
232	TRANSITION_LOOP_TYPE	47, TRANSITION_LOOP_TYPE_TYPE	Since FW 1.07 Modern mode only
233	TRANSITION_START_SAMPLE	38, UNIPOLAR_FULLRES	Since FW 1.07 Modern mode only
234	TRANSITION_END_SAMPLE	38, UNIPOLAR_FULLRES	Since FW 1.07 Modern mode only
235	GLOBAL_LFO_SYMMETRY	1, BIPOLAR	Since FW 1.08
236	OSCS_DEREZ	48, DEREZ	
237	SMOOTH_SCAN_WT	2, BOOLEAN	Since FW 1.08 Modern mode only

#### System (global) parameters - Parameters List (at the state of FW 1.11)

Num	Global Parameter Name	Global Parameter Range {min, default,max}	Representation
0	DISPLAY_BRIGHTNESS	{16, 255, 255}	integer
1	LEDS_BRIGHTNESS	{16, 255, 255}	integer
2	POPUF_TIME	{0, 2, 5}	integer
3	MASTER_TUNE	{420, 440, 460}	integer
4	TRANSPOSE	{-12, 0, 12}	integer
5	MIDI_CHANNEL	{1, 1, 16}	integer
6	DEVICE_ID	{0, 0, 127}	integer
7	VELOCITY_CURVE	{0, 0, 9}	List, 0=lin.pos. 1=lin.pos. 2=exp.pos. 3=exp.neg. 4=c/f pos. 5=c/f neg. 6=m.exp.+ 7=m.exp.- 8=m.c/f.+ 9=m.c/f.-
8	MIDI_CLOCK_MODE	{0, 0, 2}	list, 0=Auto 1=Int 2=Ext
9	CONTROL_SEND	{0, 1, 3}	list, 0=off 1=DIN 2=USB 3=USB+DIN
10	CONTROL_RECEIVE	{0, 1, 1}	boolean, 0=no 1=yes
11	STEREOWIDTH	{0, 64, 64}	integer
12	ARP_NOTES_OUT	{0, 0, 3}	list, 0=off 1=DIN 2=USB 3=USB+DIN
13	MIDI_CLOCK_SEND	{0, 0, 3}	list, 0=off 1=DIN 2=USB 3=USB+DIN
14	CTRL_A	{102, 102, 119}	integer
15	CTRL_B	{102, 102, 119}	integer
16	CTRL_C	{102, 102, 119}	integer
17	CTRL_D	{102, 102, 119}	integer
18	CTRL_W	{102, 102, 119}	integer
19	CTRL_X	{102, 102, 119}	integer
20	CTRL_Y	{102, 102, 119}	integer
21	CTRL_Z	{102, 102, 119}	integer
22	WTLOADSLOT	{0, 0, 31}	integer
23	WTLOADSKIP	{1, 1, 4}	integer



24	--obsolete--		Obsolete for FW1.11
25	MASTER_VOLUME	{0, 0, 100}	integer
26	INSTANT_LOAD	{0, 0, 1}	boolean, 0=no 1=yes
27	TRANSITIONSOURCEFILE	{0, 0, 999}	integer
28	TRANSITIONLOADSLOT	{0, 0, 63}	integer
29	--reserved--		
30	--reserved--		
31	--reserved--		
32	VCFTCVoice00	{-8191, 0, 8191}	integer
33	VCFTCVoice01	{-8191, 0, 8191}	integer
34	VCFTCVoice02	{-8191, 0, 8191}	integer
35	VCFTCVoice03	{-8191, 0, 8191}	integer
36	VCFTCVoice04	{-8191, 0, 8191}	integer
37	VCFTCVoice05	{-8191, 0, 8191}	integer
38	VCFTCVoice06	{-8191, 0, 8191}	integer
39	VCFTCVoice07	{-8191, 0, 8191}	integer
40	VCFTCVoice08	{-8191, 0, 8191}	integer
41	VCFTCVoice09	{-8191, 0, 8191}	integer
42	VCFTCVoice10	{-8191, 0, 8191}	integer
43	VCFTCVoice11	{-8191, 0, 8191}	integer
44	VCFTCVoice12	{-8191, 0, 8191}	integer
45	VCFTCVoice13	{-8191, 0, 8191}	integer
46	VCFTCVoice14	{-8191, 0, 8191}	integer
47	VCFTCVoice15	{-8191, 0, 8191}	integer
48	MODRNDENABLE	{0, 0, 1}	boolean, 0=no 1=yes
49	LFORNDSPREAD	{0, 0, 100}	integer
50	ENVRNDSPREAD	{0, 0, 100}	integer
51	OSCRNDSPREAD	{0, 0, 100}	integer
52	WAVRNDSPREAD	{0, 0, 100}	integer
53	MIXRNDSPREAD	{0, 0, 100}	integer
54	VCFRNDSPREAD	{0, 0, 100}	integer
55	VCARNDSPREAD	{0, 0, 100}	integer
56	RNDDEFMIDDLE	{0, 0, 1}	boolean, 0=no 1=yes
57	--reserved--		
58	--reserved--		
59	--reserved--		
60	--reserved--		
61	--reserved--		
62	--reserved--		
63	--reserved--		
64	VCASCALEVOICE00	{-500, 0, 0}	integer
65	VCASCALEVOICE01	{-500, 0, 0}	integer
66	VCASCALEVOICE02	{-500, 0, 0}	integer
67	VCASCALEVOICE03	{-500, 0, 0}	integer
68	VCASCALEVOICE04	{-500, 0, 0}	integer
69	VCASCALEVOICE05	{-500, 0, 0}	integer
70	VCASCALEVOICE06	{-500, 0, 0}	integer
71	VCASCALEVOICE07	{-500, 0, 0}	integer
72	VCASCALEVOICE08	{-500, 0, 0}	integer
73	VCASCALEVOICE09	{-500, 0, 0}	integer
74	VCASCALEVOICE10	{-500, 0, 0}	integer
75	VCASCALEVOICE11	{-500, 0, 0}	integer
76	VCASCALEVOICE12	{-500, 0, 0}	integer
77	VCASCALEVOICE13	{-500, 0, 0}	integer
78	VCASCALEVOICE14	{-500, 0, 0}	integer

79	VCASCALEVOICE15	{-500, 0, 0}	integer
80	LFOLEDSBRIGHTNESS	{16, 255, 255}	integer
81	GLOBALPCFILTER	{0, 0, 1}	boolean, 0=no 1=yes
82	MPE_ENABLED	{0, 0, 1}	boolean, 0=no 1=yes
83	MPE_BENDRANGE	{2, 48, 96}	integer
84	MPE_BENDRANGE_OVERRIDE	{0, 0, 1}	boolean, 0=no 1=yes
85	MPE_MASTER_PITCHBEND_ADD	{0, 0, 1}	boolean, 0=no 1=yes
86	--reserved--		
87	--reserved--		
88	--reserved--		
89	TTNOTE	{0, 60, 127}	integer
90	TTFREQINT	{5, 440, 14000}	integer
91	TTFREQFRAC	{0, 0, 999}	integer
92	TTFILENUM	{0, 0, 15}	integer