SLMKII MIDI

Programmer's Reference

Novation DMS

Edition 0.23

CONTENTS

1. Overview

This manual is for anyone who is familiar with MIDI communications, and who wants to have an understanding of the various Automap MIDI messages used by the SLMKII Keyboard Controller – possibly with a view to monitoring or controlling an SLMKII, or other Novation units which use all or part of the Automap Protocol.

The protocol was originally devised for use with the RemoteSL unit. This was the first Novation unit which had two large LCD displays – which could be set by an application program.

A major part of the protocol was the ability of the unit to load one of 8 special configuration tables, called Templates, when an application sent an ONLINE sysex message. An application program would then receive predetermined MIDI CC messages whenever specific RemoteSL controls were operated – and act accordingly.

Originally the RemoteSL had 6 special Templates which it responded to. These Templates were initially for certain popular DAW programs, such as Reason, Logic, Ableton Live and a HUI simulation. There was also a general purpose special Template – which has evolved into the Automap Template.

Most of the special Templates have been absorbed into using the 'universal' Automap Template. The one, currently, remaining exception being Ableton Live.

The Protocol has been extended, at various stages, as new Novation units have been developed. The following units use part or most of the Automap protocol:

RemoteSL
Compact
SLMKII
Nocturn Keyboard

Most of this document describes Automap operation with respect to the SLMKII Controller Keyboard – because it uses most of the Automap protocol- however , the same principals of operation apply to the other units.

This document describes the various parts and operation of the Automap protocol.

IMPORTANT: In Automap mode, most of the SLMKII control information is normally handled by a 'hidden' USB port – any control information can, however, usually be sent to one of the other normal ports – and still have the same effect. Control Information emanating from the unit in Automap mode, cannot normally be 'seen' by

host programs; as this is normally transmitted on the hidden port – special steps have to be taken by any application program in order to 'see' control changes etc. The Ableton Template is an exception to this, in that the hidden-Port is not used, and control is normally done on Port2.

Most of the control operations described also apply to the Zero units – it's just that the Zeros have no Keyboard, no Touchpad, no Pitch-bend, no Mod-wheel, no Row-select buttons – and the buttons and controls are laid out differently. The ZeroMKII has additional controls: the Cross-fader and the Preview button, which is used in combination with other controls, to perform certain SLMKII operations such as Row-Selects etc.

HEX v DECIMAL NOTATION:
Unless otherwise indicated, byte values are in hex – for those of you who think in decimal there is a Hex to decimal table in the Appendix.

2. General

The SLMKII Keyboard Controller main functions are:

1. To send 'tactile' control changes to DAWs or other MIDI equipment.
2. To display control settings, and other information, on the LCD.
3. To have a number of Configurations, or Templates, customised for specific equipment or DAWs.
4. To respond to the special Automap Protocol – which is a key feature of the units.
5. To indicate various states or setting by LEDs.
6. To have a number of Global Configuration settings – ie independent of Templates.
7. As a basic or advanced Ivory Keyboard.

AUTOMAP MODE:

The SLMKII is capable of holding 32 major customisable configuration tables called Templates – plus a fixed Template – called the Automap Template. Even though the Automap Template can be altered at the SLMKII, it should NOT be altered as it is used in conjunction with the Automap Server host program – in order to follow the established communication protocol between the two.

AUTOMAP PROTOCOL:

This protocol uses specific CC and sysex messages to do the following general things:

1. Report Control and Value Changes
2. Allow a Host Program to turn ON/OFF any LED

3. Allow a Host Program to Set the LCD Text
4. Sends Alerts to the Host.
5. Allow the Host to Request and Change Global Data
6. Allow the Host to Request and Change Template Data
7. Perform Certain High-Level Operations
8. Allow a Host Program to Request the current LCD Text **
9. Allow a Host Program to Request the State of All the LEDs **
10. Allow the unit to respond to Requests for information from a host application.

*Alerts refers to the SLMKII informing the Automap server that certain changes have occurred. Such as a change of Octave setting at the unit itself.

** Currently only used for Test purposes.

ABBREVIATIONS USED:

AMS	Automap Server
RS	Row-Select
LH	Left Hand
RH	Righ Hand
MS	Most-Significant
LS	Least-Significant
CW	Clockwise
ACW	Anti-Clockwise
T-Lock	Transport-Lock

3. Automap CC Messages

BF	CC	VV

All CC (Continuous-Controller) MIDI messages used, to and from the unit in Automap mode follow the above format.

MIDI-CHANNEL NUMBER: All Automap CC messages are sent on MIDI channel #16 ie 'BF' (Ableton uses MIDI channel #1 ie 'B0'.)

CONTROLLER NUMBER: Generally, the CC number indicates a specific SLMKII Control or , in some cases, a particular control group. CC numbers range from 00h to 7Fh.
Please note: the CC protocol control organization, does NOT follow any 'standard' MIDI CC allocations – instead, it is generally a linear allocation of CC numbers of particular SLMKII control groups eg all the Pot CCs are grouped together.

CONTROL VALUE: The value for a particular CC message varies, depending on the type of control and the direction of the message – ie to or from the unit. Some messages have a continuous range of 00-7Fh (eg pots and sliders); some have 00h or 01h, to indicate OFF or ON; some use bit-6 to indicate OFF or ON, or even 'touched, of 'untouched' – some values have single and/or multiple bit fields.
See the individual sections for the exact value usage.

SLMKII PORTS USAGE:
PORT1: This port should be used normally for Keyboard Note/Aftertouch/Pitch-Bend etc messages. Note: Messages received by the unit can be made to route directly, across the unit, to the M1-Out Port. (Useful for passing on MIDI-clocks to MIDI instruments attached to the MIDI port.)

PORT2: This port can be used for MIDI-clock messages to the unit, or to a dedicated PC application – such as Ableton Live

PORT3: Generally, Automap CC messages are sent and received on this 'hidden' port of the SLMKII . The Automap Template is configured to send control events out on the hidden port. Some events, such as the Speed-dial, Transport-lock events etc are 'hard-coded' to use the hidden port only.
Automap messages should be sent to the 'hidden' port, but in general, can be sent to one of the exposed MIDI ports of the unit – and have the same effect.

4. Sysex Headers and Commands

There are three special sysex header groups for controlling and retrieving information from the SLMII:

1. The Original Automap Sysex Header.
2. The Data-Block Change/Request Sysex Header.
3. The TEST 'Simulation' Sysex Header.

The Automap commands are directly associated with special Templates, such as the Main Automap Template, and the Ableton Template.

The Data-Block commands are three pairs of Request and Change memory block commands for retrieving and manipulating Global, Template header and Control data.

The Simulation sysex header is primarily concerned with operations for simulating physical control changes at the SLMKII, mainly for TEST purposes – and is NOT part of the normal operation of the unit. By sending this header to the unit, with additional command numbers and values – the unit is made to behave as if a user had actually operated a physical control. Accompanying these simulation commands are request commands; for requesting the current LCD text, all LED states and memory blocks at specific addresses.

Data-Block and Simulation commands do NOT require a Special-Template to be loaded on the unit.

AUTOMAP SYSEX HEADER:

	Novation MIDI-Id			Automap		Versn - Beta		Tmpl	Spare	Data	
F0	00	20	29	03	03	VV	Bb	02	00	. . . .	F7

All Automap sysex messages have the header above.

VERSION-BETA: these two bytes indicate the main and beta version numbers of the SLMKII when the particular version of the protocol was introduced – currently they should be set to 12h : 00h – and are expressed in BCD form.
(12:00 = New (LEDs) Live Version)

SPECIAL TEMPLATE NUMBER: for Automap specific messages this is set to 02. Originally there were 8 'special' Templates – eg 04h is the Ableton Template – all other special Template numbers are now redundant.

DATA: The Data section will have a Command byte followed by zero or more header and/or data bytes.

MAIN AUTOMAP SYSEX COMMANDS:

Cmd	
01	Online + Offline Commands
02	LCD Text Commands
03	Globals Data Download to RAM
04	Prepare unit for OS Download
05	Tell unit to Upload Globals
06	Globals Data Download to RAM + Flash
07	Tell unit to Upload a Template / All Templates
08	Tell unit to Upload OS

See the Automap Sysex Commands Descriptions section below for details of the various command operations.

DATA-BLOCK + SIMULATION SYSEX HEADER:

	Novation MIDI-Id			Simulation		Versn - Beta		Tmpl	Spare	Data	
F0	00	20	29	03	05	VV	bb	00	00	. . . .	F7

These two sets of commands actually have the same Sysex header – but have a different range of commands. Note: a Special-Template does NOT have to be loaded for these commands to operate.

NOTE the 03:05 difference from the Automap sysex header.

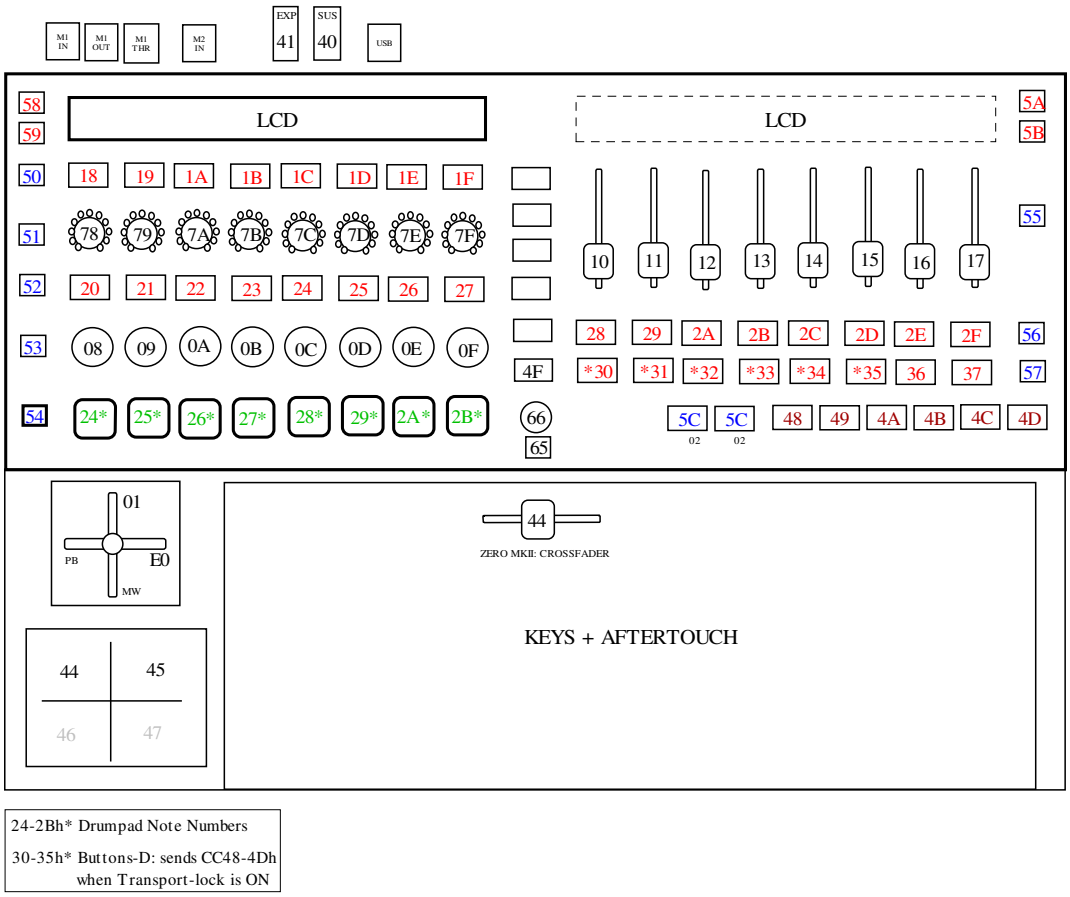
VERSION-BETA: these two bytes indicate the main and beta version numbers of the SLMKII when the particular protocol was introduced – currently they should be set to 12h : 00h.

SPECIAL TEMPLATE NUMBER: Currently this is set to 00h for Data-Block and Simulation commands.

DATA: The Data section will have a Command byte followed a Sub-Command, followed by zero or more header and/or data bytes.

See the Data-Block and Simulation sections below for descriptions of the individual sets of commands.

5. SLMKII Control CC Events to PC



SLMKII to PC AUTOMAP EVENT CC MIDI MESSAGES

The above diagram indicates the various MIDI CC numbers used for messages sent by the SLMKII, on the hidden port3, when the specific controls are operated – when in Automap mode.

The actual values sent for the different type of control are indicated below:

1.	Encoders	BF 78-7Fh 0n / 4nh	Bit6: 0=CW ; 1=ACW ; n = number of clicks eg 02h / 41h
2.	Pots	BF 08-0Fh 0-7Fh	Value = 0-127 decimal
3.	Sliders	BF 10-17h 0-7Fh	Value = 0-127 decimal
4.	Buttons-A	BF 18-1Fh 00/01h	00= Button released ; 01 = Button pressed
5.	Buttons-B	BF 20-27h 00/01h	00= Button released ; 01 = Button pressed
6.	Buttons-C	BF 28-2Fh 00/01h	00= Button released ; 01 = Button pressed
7.	Buttons-D	BF 30-37h 00/01h	T-Lock OFF: 00= Button released ; 01 = Button pressed

8.	Buttons-D *	BF 48-4Dh 00/01h	T-Lock ON: 00= Button released ; 01 = Button pressed
9.	Automap Buttons	BF 48-4Dh 40/41h	00= Button released ; 01 = Button pressed ; Bit6=0=Buttons-D ; Bit6=1=AM buttons
10.	RowSelects	BF 50-54h 00/01h	LH RS: 00= Button released ; 01 = Button pressed Automatically forces LH LCD Display.
11.	RowSelects	BF 56-57h 00/01h	RH RS: 00= Button released ; 01 = Button pressed Automatically forces RH LCD Display.
12.	LH Page Up	BF 58h 00/01h	00= Button released ; 01 = Button pressed
13.	LH Page Dn	BF 59h 00/01h	00= Button released ; 01 = Button pressed
14.	RH Page Up	BF 5Ah 00/01h	00= Button released ; 01 = Button pressed
15.	RH Page Dn	BF 5Bh 00/01h	00= Button released ; 01 = Button pressed
16.	ModWheel	BF 01h 0-7Fh	Value = 0-127 decimal
17.	PitchBend	E0 0/7Fh 0-7Fh	00:00h - 00:7Fh: + 7F:7Fh for Full scale Pitchbend is on Port#1
18.	Sustain Pedal	BF 40 00/7Fh	00= Pedal released ; 7Fh = Pedal pressed
19.	Expression Pedal	BF 41 00-7Fh	Value = 0-127 decimal
20.	Touchpad X1	BF 44 0-7Fh	Value = 0-127 decimal
21.	Touchpad Y1	BF 45 0-7Fh	Value = 0-127 decimal
22.	Touchpad X2	BF 46 0-7Fh	Value = 0-127 decimal – Disabled control.
23.	Touchpad Y2	BF 47 0-7Fh	Value = 0-127 decimal – Disabled control.
24.	CrossFader	BF 44 0-7Fh	Value = 0-127 decimal *ZeroMKII ONLY ! (same as X1)
25.			
26.	Alerts	BF 5C 0-04h **	00 = MIDI Channel Changed manually 01 = Keyboard Transpose changed. 02 = Octave setting changed 03 = AfterTouch changed. 04 = Velocity Curve changed.
27.	Speed-Dial	BF 66 0n/4nh	Bit6: 0=CW ; 1=ACW ; n = number of clicks

	Encoder		eg 02h / 41h
28.	Speed-Dial Button	BF 65 00/01h	00= Speed-dial released ; 01 = Speed-dial pushed in.

*T-Lock = Transport-Lock mode
(for compliance with, original, RemoteSL Transport button events protocol.)

** Implemented to varying extents on the different units – NOT present on the RemoteSL.

NOTES:

1. The Ableton Live Template sends CC38-3Fh for Encoder1-8 changes.
2. On the RemoteSL and ZeroSL there are NO dedicated Automap buttons for Learn, View etc.
3. On the RemoteSL and ZeroSL the dedicated Transport buttons use the SAME CC event messages as the SLMKII Automap buttons.

6. Miscellaneous SLMKII Events to PC

Here is a list of Miscellaneous Automap CC Messages sent to the PC:

CC	Purpose
4F	Transport-Lock status
5E	MS Tempo Setting
5F	LS Tempo Setting
63	Echo CC Message Response *
67	Parameter Request Response
6A	Encoders-D Row-Select – Compact unit only
6B	CC Off/OnLine Message.

* Originally designed for the Reason Special-Template.

Transport-Lock status:

BF	4F	00	Transport-Lock is OFF
BF	4F	01	Transport-Lock is ON

This message is sent:

1. Whenever the Transport-Lock button is pressed.
2. In response to a Parameter Request+Transport-Lock status Request from a host.

Tempo CC Change Pair:

BF	5E	0-7F	MSB Tempo setting
BF	5F	0-7F	LSB Tempo setting

This pair of CCs are sent:

1. As one of the last operations performed when the unit powers-up.
2. Whenever the current Tempo setting is changed.

The pair of values combine to give a 14-bit value of the current Tempo setting with a range of 20-320 BPM (0014h to 0240h).

Echo CC Message Response:

BF	63	00-7Fh

This command was originally used by the Reason LUA script to simulate an externally triggered event. All it does is to echo back the received CC message and value.

Parameter Request Response:

BF	67	RR

This CC is sent to the PC in response to a Parameter Request (RR) sent to unit.

Currently there are two Parameter Request values that can be sent to the SLMKII:

BF	67	00	Request the Unit-Product-Type
BF	67	01	*Request the Transport-Lock State

*Not implemented on the RemoteSL+ZeroSL

A Request for the Unit-Product-Type responds as follows:

BF	67	00	If unit is a RemoteSL/SLMKII
BF	67	01	If unit is a ZeroSL/ZeroMKII
BF	67	02	If unit is a Compact

A Request for the Transport-Lock status responds as follows:

BF	4F	00	Transport-Lock is OFF
BF	4F	01	Transport-Lock is ON

Off/OnLine CC Message from the SLMKII:

BF	6B	00	The last special-template has been unloaded
BF	6B	01	A special-template has been Loaded

This command supplement s the off/online sysex commands, and is sent to the current control-ports.

7. Miscellaneous Automap CCs to the SLMKII

Here is a list of Miscellaneous Automap CC Messages sent to the SLMKII:

CC	Purpose
4E	Turn OFF ALL LEDs *
60	LH Row-Select LEDs Bit-map
61	RH Row-Select LEDs Bit-map
63	Echo CC Message Request **
67	Parameter Request
68	Available Row-Select 1-5 bits **
69	Available Row-Select 6-8 bits **
6A	Encoders-D Row-Select – Compact unit only

*NOT implemented on the RemoteSL or ZeroSL
** Originally designed for the Reason Special-Template – no longer used.

Turn OFF ALL LEDs:

BF	4E	00	Turns OFF ALL Button + RS + Ring LEDs

Echo CC Message Response:

BF	63	00-7Fh

This command was originally used by the Reason LUA script to simulate an externally triggered event. All it does is to echo back the received CC message and value.

Currently there are two Parameter Request values that can be sent to the SLMKII:

BF	67	00	Request the Unit-Product-Type
BF	67	01	*Request the Transport-Lock State

*Not implemented on the RemoteSL+ZeroSL

See above for the Parameter Request Response.

LH Row-Select LEDs Bit Map:

BF	60	01	RS1(Encoders) LED bit = ON
BF	60	00	RS1(Encoders) LED bit = OFF
BF	60	04	RS3(Buttons-B) LED bit = ON
BF	60	10h	RS5(Drumpad) LED bit = ON
BF	60	12h	RS2+RS5 LED bits = ON – all others =OFF

RH Row-Select LEDs Bit Map:

BF	61	01	RS6(Sliders) LED bit = ON
BF	61	00	RS6(Sliders) LED bit = OFF
BF	61	02	RS7(Buttons-C) LED bit = ON
BF	61	04	RS5(Buttons-D) LED bit = ON
BF	61	06	RS7+RS8 LED bits = ON – all others =OFF

These commands are a more efficient alternative to the 8, individual, RS LED commands.

CC60 + CC61h BIT-MAPPED ROWSELECT + 'OTHER' VALUE BITS:

LH RS: BF 60 bb	b7	b6	b5	b4	b3	b2	b1	b0
	0	-	-	RS5	RS4	RS3	RS2	RS1

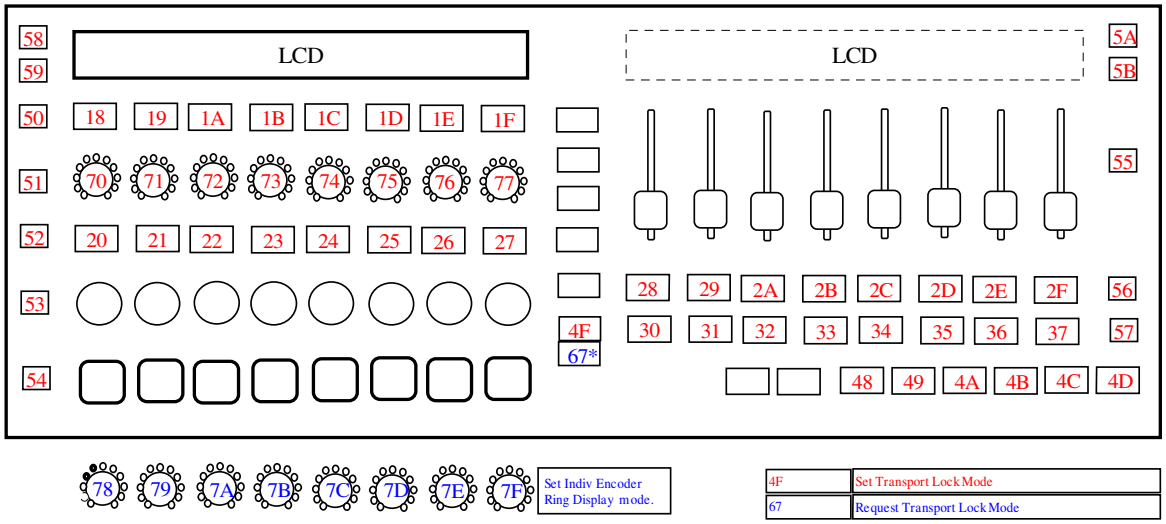
RH RS: BF 61 bb	b7	b6	b5	b4	b3	b2	b1	b0
	0	*	*	-	REC	RS8	RS7	RS6

REC = Record LED Bit

* Allocated of Reason operation – redundant.

8. Button LED Commands to SLMKII

The following diagram shows the specific CC messages which are sent to the SLMKII to turn ON/OFF specific button LEDs:



PC to SLMKII AUTOMAP LED MIDI MESSAGES

Button LED ON/OFF Messages :

BF	CC	00/01	00=LED OFF; 01 = LED ON

NOTES:

1. On the RemoteSL and ZeroSL the Record LED CC message is the same as on the SLMKII ie 4Ch.
2. On the RemoteSL and ZeroSL the Row-Select LED CC messages are the same as on the SLMKII ie 50-57h.
3. CC 70-77h and CC78-7Fh: see 'Encoder LED-Ring CC Commands' section below.

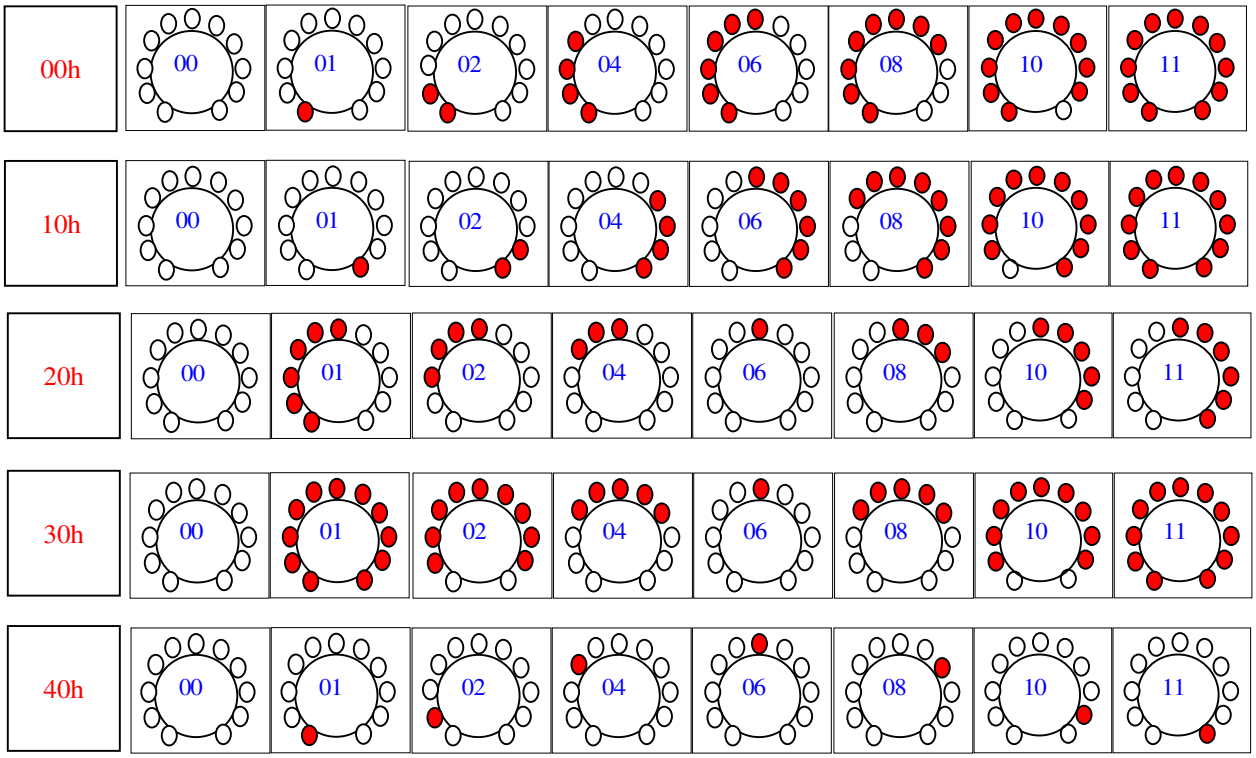
9. Encoder LED-Ring CC Commands to SLMKII

There are two sets of CC commands for controlling the 8 Encoder LED-Rings:

1. 8 commands for setting the Display-Mode of each LED-Ring.
2. 8 commands for setting the value of each LED-Ring – based on its current Display-Mode.

Encoder Ring #		CC	Mode value
1	BF	78h	mode
2	BF	79h	mode
3	BF	7Ah	mode
4	BF	7Bh	mode
5	BF	7Ch	mode
6	BF	7Dh	mode
7	BF	7Eh	mode
8	BF	7Fh	mode

Ring Display-Mode	Ring LED pattern
00h	0-11 Continuous Band (CW)
10h	11-0 Continuous Band (ACW)
20h	L/R Centred Band
30h	Double Centred Band
40h	0-11 Single LED (CW)



Example Ring-LED Modes v Values

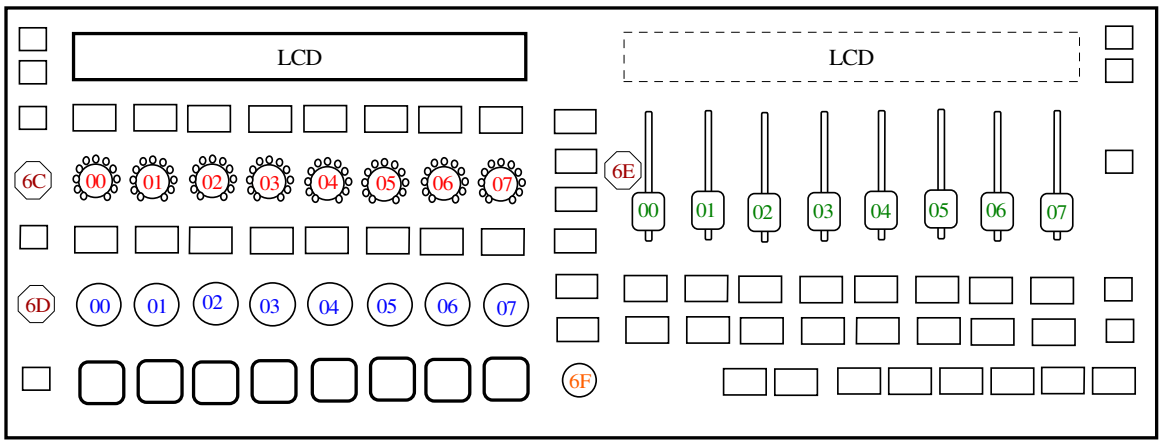
Note: Sending a value of Zero always turns OFF all Ring-LEDs wahtever the mode.

ENCODER RING-LED VALUE COMMANDS:

Encoder Ring #		CC	Ring value
1	BF	70h	00-0Bh
2	BF	71h	00-0Bh
3	BF	72h	00-0Bh
4	BF	73h	00-0Bh
5	BF	74h	00-0Bh
6	BF	75h	00-0Bh
7	BF	76h	00-0Bh
8	BF	77h	00-0Bh

10. Touch Sensor Reports

The following diagram show the CC message numbers and values sent by the SLMKII when a specific Touch sensitive control is touched or 'untouched':



SLMKII to PC AUTOMAP TOUCH MIDI MESSAGES

The Touch-Sensor events are indicated in four groups of CC messages; with the CC number indicating the group; and the least significant nibble value indicating the individual control number in the group (0-7); if bit-6 of the value is set, then the control has just been Touched, if the value is clear then the control has just been 'unTouched'.

Currently only ONE sensor is considered 'Touched' at a time – even if more than one is actually touched. While the Speed-dial is being touched, no other touch sensor can cause a 'Touched' message.

			1	2	3	4	5	6	7	8
Encoders1-8	BF	6C	40/00	41/01	42/02	43/03	44/04	45/05	46/06	47/07
Pots1-8	BF	6D	40/00	41/01	42/02	43/03	44/04	45/05	46/06	47/07
Sliders1-8	BF	6E	40/00	41/01	42/02	43/03	44/04	45/05	46/06	47/07
Speed-dial	BF	6F	40/00							
Cross-fader*	BF	6F	41/01							

All table values are in hex 4v/0v = 'Touched' / 'unTouched'

*Applies to Zero MKII only.

11. Automap Sysex Command Descriptions

01 Online + Offline Commands:

	Novation MIDI-Id			Automap		Versn - Beta		Tmpl	Spare	Cmd	value	
F0	00	20	29	03	03	VV	bb	nn	00	01	00 01	F7

For Automap, Tmple (special Template) = 02 (04h for Abelton)

These two pairs of command have different meaning – depending on the direction in which it is being sent:

		Value	Operation
To SLMKII	ONLine	01	Always forces the unit to go into Automap mode, even when in Advanced Template mode.
To SLMKII	OFFLine	00	Tells the unit that the Automap server is no longer running – the unit displays 'Automap is OFFLINE'

To AMS/Live	ONLine	01	Tells the host program that the unit has loaded a special Template – so it can resume LCD text updates.
To AMS/Live	OFFLine	00	Tells the host program that the unit has just unloaded a special Template – so there is no need for the associated application to send LCD text updates.

02 LCD Text Commands:

	Novation MIDI-Id			Automap		Versn - Beta		Tmpl	Spare	Cmd	Sub Cmd	Data	
F0	00	20	29	03	03	VV	bb	nn	00	02	ss	. . . .	F7

After the Text sysex header, a series of commands and data will follow relating to the type of text operation to be performed. It is possible for multiple commands to be performed in the Text sysex message eg a cursor-addressing command could be followed by a text string etc.

Commands will be followed by zero or more associated data bytes.
Most commands have an 'inherent' length based on the command type.

IMPORTANT: Even though the SLMKII etc have only one LCD – text for BOTH the LEFT and RIGHT set of controls should still be sent to the unit. This is because the unit AUTOMATICALLY switches the LH/RH LCD text displayed to match the last LH or RH control operated – of course there is no need for switching on the RemoteSL, as this has two LCDs.

Text Sub- Commands:

00	End of Text String
01	Cursor Address.
02	Clear Text Command
03	Cursor Blinking Command*
04	Text String
F7	End of Sysex

*NOT implemented

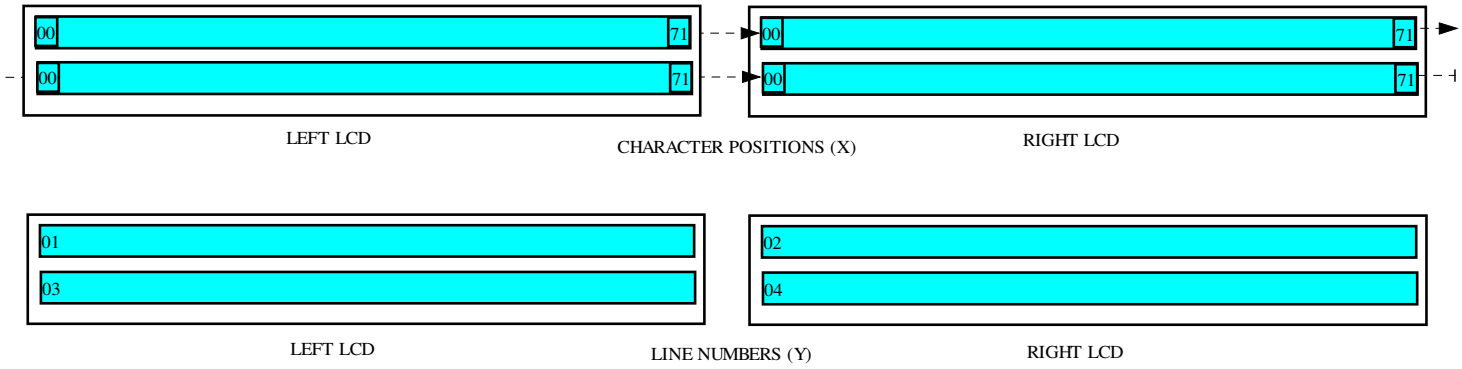
01 Cursor Addressing:

This is a 2 byte command for setting the current cursor position on the LCDs:

First byte	Column Position: 00 – 71
Second byte	LCD line number : 01-04

01	Left LCD - Top Line
02	Right LCD - Top Line
03	Left LCD - Bottom Line
04	Right LCD - Bottom Line

eg 01 23 02 (Hex) – will position the cursor in the middle of the top line of the Right LCD.



02 Clear LCD Display Functions:
These are 1 or 2 byte sub-commands for clearing sections of the LCDs.

		Cursor position after clear cmd:
01	Clear both displays	Set cursor to Top-Left of Left LCD
02	Clear the entire Top-Line of both LCDs	Set cursor to Top-Left of Left LCD
03	Clear the entire Bottom-Line of both LCDs	Set cursor to the Bottom -Left of Left LCD.
04	Clear the entire Left LCD	Set cursor to its Top-Left of Left LCD.
05	Clear the entire Right LCD	Set cursor to the Top-Left of Right LCD.
06	Clear the Top-Line of the Left LCD	Set cursor to its Top-Left of Left LCD.
07	Clear the Bottom-Line of the Left LCD	Set cursor to the Bottom-Left of Left LCD.
08	Clear the Top-Line of the Right LCD	Set cursor to its Top-Left of Right LCD.
09	Clear the Bottom-Line of the Right LCD	Set cursor to the Bottom-Left of Right LCD.
10	Clear from the current cursor address for the number of character positions in byte #2	Go back to the original cursor position. ** Two-Byte sub-command. Eg: 02 0A 10 = clear 16 characters from the current cursor position.

03 Cursor Blinking Command:
**** NOT implemented**

04 Variable Length Text:
This can be a null terminated string of ASCII text, and will be a maximum of 144 characters long.
Automatic character wrapping will occur from the extreme top-right of the Right LCD to the extreme bottom-left of the Left LCD.

Characters placed beyond the extreme bottom-right of the Right LCD– will be ignored.

Example **LCD Text Message:**

Button2 Button3 1 ON

Short LCD sysex message:

```
F0 00 20 29 03 03 12 00 02 00 02 02 04 01 09 01 04 42 75 74 74 6F 6E
20 32 00 01 09 03 04 20 20 20 31 00 01 12 01 04 42 75 74 74 6F 6E 20
33 00 01 12 03 04 20 20 20 4F 4E 00 F7
```

Breakdown:

F0 00 20 29 03 03 12 00 02 00	Sysex Header
02	LCD Commands
02 04	Clear Entire Left Display
01 09 01	Cursor Posn XY = Char 9, Top LH Line
04 42 75 74 74 6F 6E 20 32 00	Text = "Button 2 "
01 09 03	Cursor Posn XY = Char 9, Bottom LH Line
04 20 20 20 31 00	Text = " 1"
01 12 01	Cursor Posn XY = Char 18, Top Line
04 42 75 74 74 6F 6E 20 33 00	Text = "Button 3"
01 12 03	Cursor Posn = Char 18, Bottom Line
04 20 20 20 4F 4E 00	Text = " ON"
F7	End of Sysex

03 Globals Data Download to RAM:

	Novation MIDI-Id			Automap		Versn -	Beta	Tmpl	Spare	Cmd	Data (1024*)	
F0	00	20	29	03	03	VV	bb	nn	00	03	. . . .	F7

*For the RemoteSL this is 256 bytes

The Data payload has to be the EXACT number of bytes for the particular unit.
ALL Data payload bytes have bit-7 clear.

04 Prepare unit for OS Download:

	Novation MIDI-Id			Automap		Versn -	Beta	Tmpl	Spare	Cmd	
F0	00	20	29	03	03	VV	bb	nn	00	04	F7

This command is normally used by the Automap Server to put a unit into a state for receiving an Operating System download update – using the hidden Port3.
The message 'Waiting for OPERATING SYSTEM UPDATE via USB 3' will be displayed. Note: the unit must be re-powered to exit this state.

05 Tell unit to Upload Globals:

	Novation MIDI-Id			Automap		Versn -	Beta	Tmpl	Spare	Cmd	
F0	00	20	29	03	03	VV	bb	nn	00	05	F7

This command causes the unit to put the entire 256/1024 bytes of Global data in RAM.
The response is exactly the same as command #3 – only it comes from the unit.

06 Globals Data Download to RAM + Flash:

	Novation MIDI-Id			Automap		Versn -	Beta	Tmpl	Spare	Cmd	
F0	00	20	29	03	03	VV	bb	nn	00	06	F7

This is identical to command 03 – except that the received Global data goes to BOTH the RAM and is saved to immediately to FLASH.

07 Tell unit to Upload a Template / All Templates:

	Novation MIDI-Id			Automap		Versn - Beta		Tmpl	Spare	Cmd	Template Num	
F0	00	20	29	03	03	VV	bb	nn	00	07	TT	F7

This command makes the unit UPLOAD ALL or ONE Template from Flash – in exactly the same way as if the Globals Template Upload option had been used – except for the uploading of the current Template in RAM.
TT = 00 = Upload ALL Templates.
TT = 01 - 33 = Upload specific SINGLE Template (includes Automap Template).

08 Tell unit to Upload OS

	Novation MIDI-Id			Automap		Versn - Beta		Tmpl	Spare	Cmd	
F0	00	20	29	03	03	VV	bb	nn	00	08	F7

This command puts the unit directly into a state where it the OS firmware can be sent to a PC – by pressing the PROCEED button – ONLY to the MIDI1-Ouput Port.

12. Data-Block Sysex Command Descriptions

Data-Block Sysex Header

	Novation MIDI-Id			Simulation		Versn - Beta		Tmpl	Spare	main Cmd	sub Cmd	Data	
F0	00	20	29	03	05	VV	bb	00	00	mc	sc	. . . .	F7

Main Cmd	sub Cmd			DATA CHANGE COMMANDS:
68h	0	DataRange	Data	Control-Data- Change command to unit
68h	1	DataRange	Data	Template-Header - Change command to unit
68h	2	DataRange	Data	Global–Change command to unit
68h	3	DataRange		Control Data Request.
68h	4	DataRange		Template-Header Request
68h	5	DataRange		Global Request
				DATA REQUEST BLOCK CMDS:
69h	3	DataRange	Data	Response from the unit to a Control-Data Request
69h	4	DataRange	Data	Response from the unit to a Template-Header Data Request.
69h	5	DataRange	Data	Response from the unit to a Global-Data Request

Three PAIRS of Sysex Commands have been added for the SLMKII units – there are three commands for requesting single or multiple data bytes FROM the unit, and three for sending single or multiple blocks of data TO the unit.

The three types of data are for:

1. Individual or consecutive bytes to / from a specific Control in RAM.
2. Individual or consecutive bytes to / from the Template Header currently in RAM
3. Individual or consecutive bytes to / from the Global data area in RAM.

With the Data Request commands, the host sends the unit a sysex message; and the unit Responds by sending back a sysex message, with an identical front end, but with an additional number of data bytes copied from the requested area.

** WARNING: Altering Data-Blocks incorrectly can cause the unit to CRASH !!

With the Data Change commands, the host sends the unit a similar sysex message to the Request message plus additional single or multiple data bytes to be copied into the particular data area, at the specified offset into the area – there is NO Response message.

GENERAL DATA-BLOCK REQUEST FORMAT:
The general format for the Template-Header and Global Data Requests are:

Example: "F0 00 20 29 03 05 12 00 00 00 68 05 00 52 00 08 F7":

Request to unit:

Novation Sysex Header	main Cmd	sub Cmd	Offset into Data Area (14 Bits)	Number of Bytes (14 Bits)	
F0 00 20 29 03 05 12 00 00 00	**68**	05	00 52	00 08	F7

"F0 00 20 29 03 05 00 00 00 00 **69** 05 00 52 00 08 20 20 20 20 20 20 20 20 F7":

Response from unit:

Sysex Header	Data payload	
F0 00 20 29 03 05 00 00 00 00 **69** 05 00 52 00 08	20 20 20 20 20 20 20 20	F7

Example is of receiving 8 - Global Drumpad Threshold setting bytes.

For the Control Data Requests, the general format is similar:
Example: " F0 00 20 29 03 05 10 05 00 00 68 03 42 00 00 10 F7":
Request to unit:

Novation Sysex Header	main Cmd	sub Cmd	Control Number	Offset into Control	Not used	Num of bytes	
F0 00 20 29 03 05 12 00 00 00	**68**	03	42	00	00	10	F7

" F0 00 20 29 03 05 00 00 00 00 **69** 03 42 00 00 10 53 75 73 41 42 65 64 20 01 00 7F 00 00 04 58 00 F7":
Response from unit:

Sysex Header	Data payload ('SusABed .. ..' was 'Sus Ped .. ..')	
F0 00 20 29 03 05 00 00 00 00 **69** 03 42 00 00 10	53 75 73 41 42 65 64 20 01 00 7F 00 00 04 58 00	F7

GENERAL SYSEX DATA CHANGE FORMAT:
The general format for the Template-Header and Global Data Change are:

Example: "F0 00 20 29 03 05 10 05 00 00 68 02 00 52 00 08 12 20 20 20 20 20 20 34 F7"
Change to unit:

Novation Sysex Header	main Cmd	sub Cmd	Offset into Data Area	Number of Bytes	New Data Payload	
F0 00 20 29 03 05 10 05 00 00	68	02	00 52	00 08	12 20 20 20 20 20 20 34	F7

Example is for changing the 8 - Global Drumpad Threshold setting bytes.

For the Control Data Change, the general format is similar:
Example: " F0 00 20 29 03 05 10 05 00 00 68 00 42 03 00 02 41 42 F7":
Change to unit:

Novation Sysex Header	main Cmd	sub Cmd	Control Number	Offset into Control	Not used	Number of bytes	New Control Data Payload ("AB")	
F0 00 20 29 03 05 10 05 00 00	68	00	42	02	00	02	41 42	F7

Example is for changing two bytes of the Sustain Pedal Control Name.

NOTES:

1. All Data Payload bytes are restricted to 0-7Fh values.
2. Offsets are Base-0 numbers ie 00h is the first data byte in a data area.
3. The Template-Header and Globals Data Offsets, and number of bytes; are restricted to 14 bits values.
4. The Control Offsets and Number of bytes are restricted to 127 bytes.
5. The Control Number is a Base-1 value. Valid control numbers are 1-90 inc.
6. These Request and Change Commands can be performed in BOTH StandAlone and Automap Modes.
7. The 'Not used' byte of the Control Data Messages 'could' be used in future to request a number of 'consecutive' controls.
8. IMPORTANT: Refer to the following documents for the various members, and their OFFSETs, of the Template-header, Control-Strip and Global Data:
   1. 'SLMKII Template Offsets .pdf'
   2. 'SL Control Members etc.pdf'
   3. 'SLMKII Global Offsets.pdf'

   THESE DOCUMENTS CAN ONLY BE PERSONALLY REQUESTED FROM NOVATION DMS.

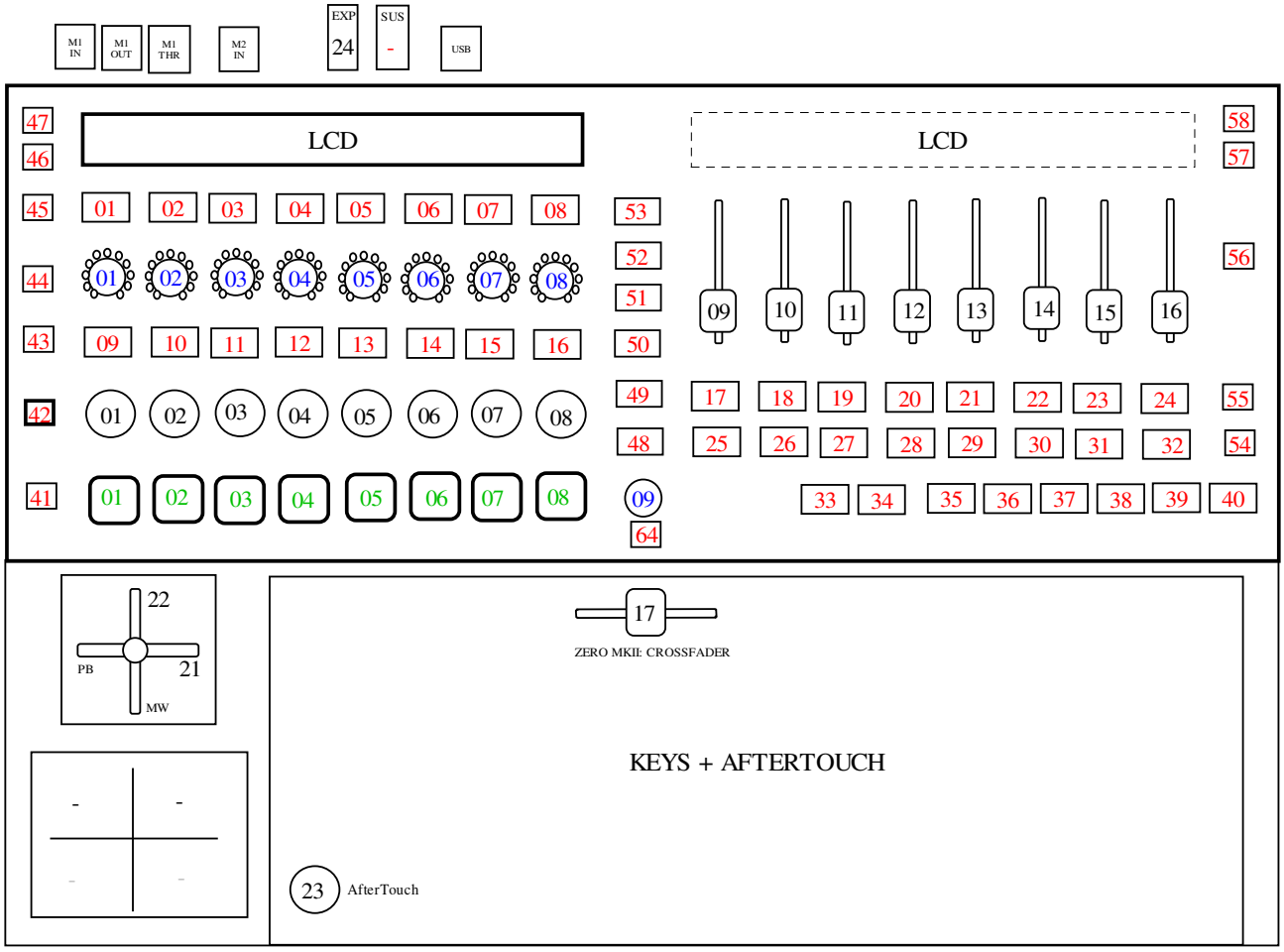
13. Simulation Sysex Command Descriptions

Simulation Commands Sysex Header

	Novation MIDI-Id			Simulation		Versn - Beta		Tmpl	Spare	main Cmd	sub Cmd	Data	
F0	00	20	29	03	05	VV	bb	00	00	mc	sc	. . . .	F7

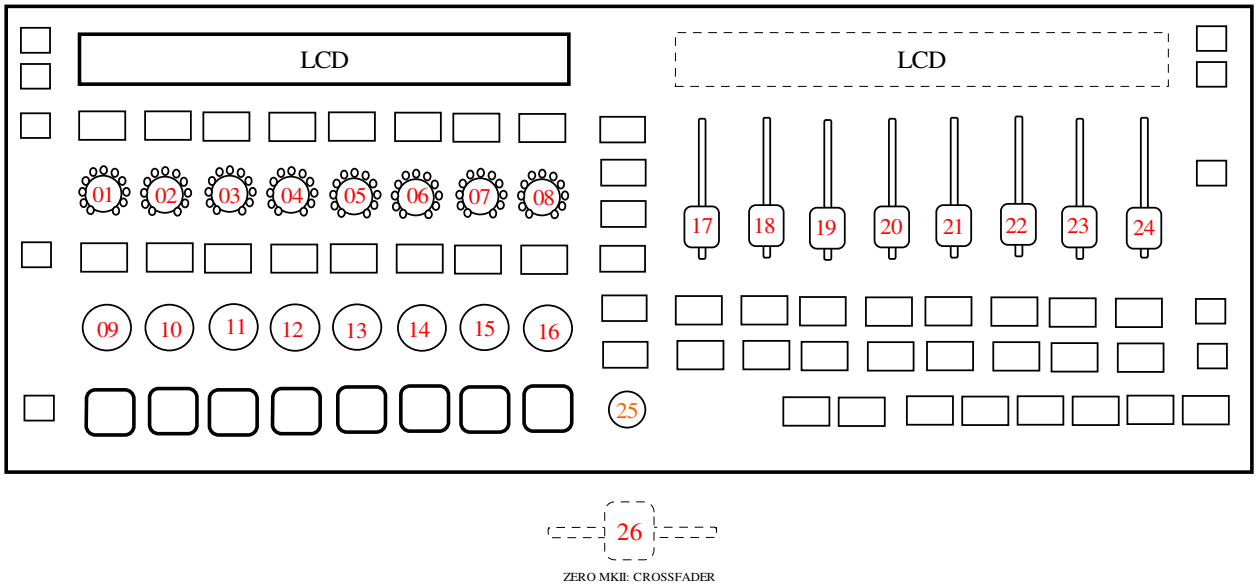
Main Cmd	Sub Cmd	Param #1 (Cn num)	Param #2 (value)	SIMULATION CMDS: (to the unit)
66h	01	1-58	00 / 01	Simulate Button #n Release or Press
66h	02	1-24	0-127	Simulate New Pot/Slider/Misc #n Position 0-7Fh
66h	03	1-9	+/- 64	Simulate Encoder Change = number of 'clicks'
66h	04	-	-	LCD Text Request to the unit.
66h	05	-	-	LCD Text Response from the unit.
66h	06	-	-	LED Status Bit Map Request to the unit.
66h	07	-	-	LED Status Bit Map Response from the unit
66h	08	1-25	0-127	Simulate Ivory Key #n plus velocity ** Non-linear**
66h	09	X=0-127	Y=0-127	Simulate New TouchpadX+Y setting
66h	10	1-8	0-127	Simulate Drumpad #1-8 + value event
66h	11	1	00/01	Simulate Sustain-Pedal Release + Press.
66h	12	1-26/127	-	Simulate Touch-Sensor 01-26(decimal) 127= NO sensor is Touched.
				ADDITIONAL HIGH-LEVEL CMDS:
6Ah	00	-	-	Save RAM Globals to Flash
6Ah	01	-	-	Save Current RAM Template to Flash
6Ah	02	-	-	Update Octave LEDs
6Ah	03	-	-	Force the Unit to Play Mode
6Ah	04	-	-	Send the Current Template to the Host

** NOTE: the sub-cmd,Param1 and param2 values are in decimal.
See Appendix-2 – Simulation Controls Map.



RED = Buttons
BLUE = Encoders
BLACK = Pots / Sliders etc
GREEN = Drumpads

PC to SLMKII SIMULATION CONTROL NUMBERS



PC to SLMKII SIMULATION TOUCH MIDI MESSAGES

04 SAMPLE LCD REQUEST + RESPONSE:

```
F0 00 20 29 03 05 10 05 00 00 66 04 00 00 20 F7

F0 00 20 29 03 05 00 00 00 00 66 05 00 00 20

20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 41 75 74 6F 6D 61 70 20 69 73 20 4F 46 46 4C 49 4E 45 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 4D 61 6B 65 20 73 75 72
65 20 41 75 74 6F 6D 61 70 20 69 73 20 6C 6F 61 64 65 64 20 6F 6E 20
79 6F 75 72 20 63 6F 6D 70 75 74 65 72 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 F7
```

Currently the above EXACT Request message must be used: The X+Y+Length parameters – have NOT been fully implemented. This Request ALWAYS returns the the FULL set of text on BOTH LCDs.

06 SAMPLE LED BITMAP REQUEST + RESPONSE:

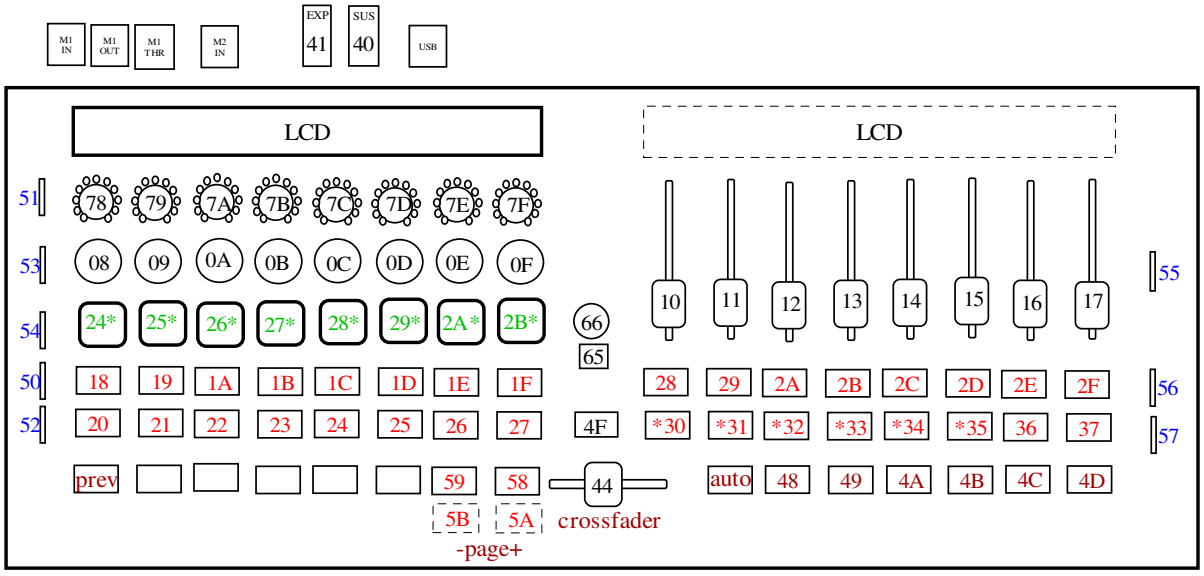
```
F0 00 20 29 03 05 10 05 00 00 66 06 00 00 20 F7


F0 00 20 29 03 05 00 00 00 00 66 07 00 00 14

00 00 00 00 00 60 10 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F7
```

The Response is a copy of the 20 LED 8-bit bytes in the unit – split into 20 7-bit values plus 3 7-bit bytes with the status of the MS-bits of the 20 LED bytes, plus a spare byte. The 20 bytes contain ALL Button and ALL 88 Ring-LED ON/OFF statuses.

Appendix 1 – ZeroMKII Button LED Commands:



(BLUE = SLMKII CC+LED CODES)

ZeroMKII to PC AUTOMAP EVENT CC MIDI MESSAGES

Notes:

1. Generally speaking, the same Event and LED CC codes are used on the ZeroMKII as on the SLMKII – it's just that the controls are physically swapped around.
2. Row-Selects are sent when the Preview Button is held down and : an Encoder or Pot or Slider is Touched; a MIDI Control Button is pressed – the same as pressing a Row-select on an SLMKII.
3. Holding the Preview button down, causes the Page buttons to operate in RH mode, instead of the normal (Preview button up) LH Page Button mode.

Appendix2 – Hex Byte to Decimal Value Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1x	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2x	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3x	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4x	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5x	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6x	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7x	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8x	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9x	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
Ax	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
Bx	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
Cx	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
Dx	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
Ex	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
Fx	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255