MSQ-CC-LRE

MotionSeQuencer for ControlChanges for 2xLRE8x2 Boards
Synth-Patch-Editor & Motion-Sequencer 4 ControlChange (= CC-Automation)



Video

Introduction

Controls and automate a Nord Drum2 (Drum-Synth)NORD DRUM 2

It acts as:
- **Midi Merger** NTE,CLK,PC merge with CC... - **Patch Manager** it replaces the Synths internal Patch Storage, each PC Number from your Sequencer is added by the BANK CC (CC 32), where each Nr is ADD 128 PC Numbers more...
- **Motion Sequencer** Record your Controller Movements in a Sequence in 32th Resultion @ maximal

256 Steps length

Features

- **Remote your Synths** by: 8x Midichannels with up to 32x Control Change (CC)
- **Save the Patches** and dump it to Synth
- **Load hundrets of Patches** via received Program Change + the Bank-CC (CC32)
- **Save Patches** vie CC24 + CC value 0-127... when sending before a BankCC32 you can expand that to 128×128 patches
- **Record CC-Motion-Sequences** - **PLAY Motions-Sequences** up to 256 steps @ 32th rate - **VELOCITY MORPH** Add Velocity-Ammount to CCs
- **MERGE** incoming Midi-Notes/Clock/Pitchbend with Automated CCs
- **Set Sequencer Beatstructure** - how to interprete Clock-ticks (4/4, 5/4, 6/4, 7/4...) - CC23
- **Global Page**: for example you use 8 simular Drum-Voices, with the Global you have 8 channel strips with dedicated Controlls, for example:
8xVolume, 8xTone/Noise-Mix, 8xDistortion, 8xClick
if you have one Synth over 2xMSQ_CC_LRE Tracks(booth set to MidiChannel 0, to get 64CCs instead of 32), then the Global Page: have the abillity to show/edit a parameter from Track1Voice on Track1Global, and from Track2Voice on Track2Global... it depends how you set the Midichannel in the Systemsettings (which are hardcodet but via Mapping Array changeable)
- for one multipart-synth, MSQ_CC_LRE do all the Preset Store, and Automations, so it is one Unit > to use the Unit in a other way would make all the Patches (128×128 patches) useless, so once done, its bound to it, load all with Programchange! minimal is better here, there will be other **MSQs** outthere see MSQ-CC-BCR

Hardware Requirements

External Requirement:(for example)

- Melody/Clock Source with ProgramChange-Output: midibox_seq_v4l oops that dont do PC...
- Melody/Clock Destination: NordDrum 2

Midibox:

- core32
- 1xMidi IO connect 1 midicontroller and 1 Note/Clock-Source/Destination
- SD-Card, formated with FAT32, and the file “bcr1.syx” on it
- Soldering Iron, Wires, PCB....
- USB Power Supply... I tried to use the Midi-BUS-Power from BCR2000 but it is too weak!
- 2x LRE 8×2 mb-lre8x2cs_pcb
- 3 extra Encoders and Ledrings (to controll the unit) + Pushfunction inclusive Button LED
- 8 Momentary Buttons without LED
- 1 Momentary Button with LED
- 1x DINX4
- 1x DOUTX4

Setting

Cabeling

MidiIO PortB Out »> Synth Midi IN
MidiIO PortB In »> Clock+Notes

Frontpanels

Stickers

LCD

it would be possible to add 16 OLEDs but... but i dont have the money for that right now....

MBHP

Software

Firmware

V1. from 9.05.2018msq_cc_lre_v0.norddrum2.zip
hardcodet for a NordDrum2 - but change-able in Mapping via a Array in Sourceode:
this is the maping which says wich of the 32 internal CCs are one of the outhernal CCs (0-127):

```
// 4 CC_Route_Mode = 0:    = 1x Synthedit0r
const u8 CC_Map0[8][32] = { // CC_Map0 [Part]       [Internal CC Nr] = value of external CC  >>
                            // CC_Map1 [MidiChannel] [Remote/Source]  = Value of Synth/Destination
                              //  1st Row Horizontal      (top-row)          //  2nd Row Horizontal                  //  3rd Row Horizontal                  //  4th Row Horizontal
// CC-on-LRE:  0    1    2    3    4    5    6    7        8    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31
              { 57 , 56 , 24 , 23 , 26 , 25 , 10 ,  7 ,   15 , 17 , 14 , 16 , 18 , 19 , 20 , 21 ,  46 , 52 , 53 , 47 , 48 , 54 , 55 , 254 ,  50 , 51 , 49 , 30 , 58 , 255 , 255 , 255},  // Nord Drum 2   Voice 1   MidiCh  7
              { 57 , 56 , 24 , 23 , 26 , 25 , 10 ,  7 ,   15 , 17 , 14 , 16 , 18 , 19 , 20 , 21 ,  46 , 52 , 53 , 47 , 48 , 54 , 55 , 254 ,  50 , 51 , 49 , 30 , 58 , 255 , 255 , 255},  // Nord Drum 2   Voice 2   MidiCh  8
              { 57 , 56 , 24 , 23 , 26 , 25 , 10 ,  7 ,   15 , 17 , 14 , 16 , 18 , 19 , 20 , 21 ,  46 , 52 , 53 , 47 , 48 , 54 , 55 , 254 ,  50 , 51 , 49 , 30 , 58 , 255 , 255 , 255},  // Nord Drum 2   Voice 3   MidiCH  9
              { 57 , 56 , 24 , 23 , 26 , 25 , 10 ,  7 ,   15 , 17 , 14 , 16 , 18 , 19 , 20 , 21 ,  46 , 52 , 53 , 47 , 48 , 54 , 55 , 254 ,  50 , 51 , 49 , 30 , 58 , 255 , 255 , 255},  // Nord Drum 2   Voice 4   MidiCh 10
              { 57 , 56 , 24 , 23 , 26 , 25 , 10 ,  7 ,   15 , 17 , 14 , 16 , 18 , 19 , 20 , 21 ,  46 , 52 , 53 , 47 , 48 , 54 , 55 , 254 ,  50 , 51 , 49 , 30 , 58 , 255 , 255 , 255},  // Nord Drum 2   Voice 5   MidiCh 11
              { 57 , 56 , 24 , 23 , 26 , 25 , 10 ,  7 ,   15 , 17 , 14 , 16 , 18 , 19 , 20 , 21 ,  46 , 52 , 53 , 47 , 48 , 54 , 55 , 254 ,  50 , 51 , 49 , 30 , 58 , 255 , 255 , 255},  // Nord Drum 2   Voice 6   MidiCh 12
              { 255 , 255 , 255 , 255 , 255 , 255 , 255 , 255 ,   255 , 255 , 255 , 255 , 255 , 255 , 255 , 255 ,  255 , 255 , 255 , 255 , 255 , 255 , 255 , 255 ,  255 , 255 , 255 , 255 , 255 , 255 , 255 , 255},  //not in Use
              { 255 , 255 , 255 , 255 , 255 , 255 , 255 , 255 ,   255 , 255 , 255 , 255 , 255 , 255 , 255 , 255 ,  255 , 255 , 255 , 255 , 255 , 255 , 255 , 255 ,  255 , 255 , 255 , 255 , 255 , 255 , 255 , 255}};  //not in Use
```

This Mapping says which one of the 32 internal CCs are positioniered in the Mixer/Overview/Channelstrip-Mode

```
  // 4 CC_Route_Mode = 1: = 8x Channelstrip
  const u8 CC_Map1[8][32] = { //CC_Map1 [Active_Strip_Set] [CC to remap to Map0]
                              //  1st Row Horizontal                  //  2nd Row Horizontal                  //  3rd Row Horizontal                  //  4th Row Horizontal
                              {  1 ,  1 ,  1 ,  1 ,  1 ,  1 , 32 , 32 ,   28 , 28 , 28 , 28 , 32 , 32 , 32 , 32 ,   3 ,  3 ,  3 ,  3 , 32 , 32 , 32 , 32 ,   7 ,  7 ,  7 ,  7 ,  7 ,  7 , 32 , 32 },   // Channel-Strip-Set1 (Mixer)
                              {  0 ,  0 ,  0 ,  0 ,  0 ,  0 , 32 , 32 ,   9 ,  9 ,  9 ,  9 , 32 , 32 , 32 , 32 ,   4 ,  4 ,  4 ,  4 , 32 , 32 , 32 , 32 ,   5 ,  5 ,  5 ,  5 ,  5 ,  5 , 32 , 32 },   // Channel-Strip-Set2 (Filter)
                              { 15 , 15 , 15 , 15 , 15 , 15 , 32 , 32 ,  19 , 19 , 19 , 19 , 19 , 19 , 32 , 32 ,  24 , 24 , 24 , 24 , 24 , 24 , 32 , 32 ,  25 , 25 , 25 , 25 , 25 , 25 , 32 , 32 },   // Channel-Strip-Set2 (Decay)
                              {  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 },  //not used
                              {  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 },  //not used
                              {  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 },  //not used
                              {  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 },  //not used
                              {  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,  32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 }};//not used
        // CC_Map1[8][0]=1 > shows CC_Map0[x][1] = Synth-CC:56 = ClickLevel-CC

// to get:   CC_Map0[0-7][ CC_Map1[0][0-7] ] = 56CC-Nr.= ClickGain   CC_Map0[0-7][ CC_Map1[8][8-15] ] = 58CC-Nr. = Bal      in the End: CC_Map0[0-7][ CC_Map1[Active_Strip_Set][0-32] ] = ..CC-Nr.
//                                  1                                                                  28
// to get:   CC_Map0[0-7][ CC_Map1[1][0-7] ] = 15CC-Nr.= N.Filt     CC_Map0[0-7][ CC_Map1[1][8-15] ] = 17CC-Nr.= N.Res     For Value:  beat[8].CC_Store[32]
//                                  0                                                                  9                                beat[0-7].CC_Store[ CC_Map1[Active_Strip_Set][0-32] ]
// to get:   CC_Map0[0-7][ CC_Map1[2][0-7] ] = 21CC-Nr.= N.Decay    CC_Map0[0-7][ CC_Map1[2][8-15] ] = 47CC-Nr.= TimDec
//                                  15                                                                 19
// to get:   CC_Map0[0-7][ CC_Map1[3][0-7] ] = 255CC-Nr.= Nothing-will be filtered out! > blank out LEDRING
//                                  32
```

there are 8 deep edit pages, and 8 overview pages.

CC Routing to Synths

MSQ_CC_LRE internal i have 8×32 CCs, they are always identical.
but with a simple input output matrix i can decide which CC it gets in real world.
each of the 8 Part can have midichannle 0-15...
So we talking about Mapping... in the moment it is made in the source code with a simple array.

To Do

alot, but since it is base on MSQ-CC-BCR! most is done, and its running solid

maybe scale min max values for CC: for example different synths have only 0-3value instead of 0-127, by different functions like WAVEFORM...) - this will be interesting when using other synths then nord drum...

Resources

Community users working on it

- **Phatline** = Programming, Documentation...

Just let a Private message on the forum to user already involved, the sourcecode is includet in the firmware .zip!!!

From:
https://www.midibox.org/dokuwiki/ - **MIDIbox**

Permanent link:
https://www.midibox.org/dokuwiki/doku.php?id=msq-cc-lre&rev=1539132843

Last update: **2018/10/10 00:54**