

# Sequencomat V1 manual & documentation

## *Preface*

„Sequencomat“ is the name of a MIDI step sequencers series I have made in [cycling74s Max/MSP](#) with touchscreen control surface on a Lemur. Original designed for the [Jazzmutant Lemur](#) these sequencers also work with [Liines Lemur App for iPad](#). For detailed information about my work please visit my page [music-interface.com](#)

All Sequencomat maxpatches work with the [free runtime](#) and also work parallel with max4live. As a user you do not need any scripting skills in max to run that sequencer, but the patches are open for editing if you want to. SequencomatV1 is released under [creative commons license BY – NC - SA](#) .

**SequencomatV1** was my first attempt of an 8 track stepsequencer and released september 2009. Meanwhile I released version 2 and 3 of this step sequencers which are more elaborated.

However, that version1 - with its direct controls – has its own charme. So as in december 2011 Liine ported the Lemur to iPad, I decided to re-release SequencomatV1 for the iPad. I adapted the control surface in resolution, but also in appearance and added some functions (physics on trackvalues). On the maxpatch I fixed some minor bugs, but in common it reflects my scripting abilities from the date of first release.

I hope you all enjoy sequencing with it.

For questions refer to the forum at Liine or Jazzmutant, or write me an email:  
[mat@music-interface.com](mailto:mat@music-interface.com)

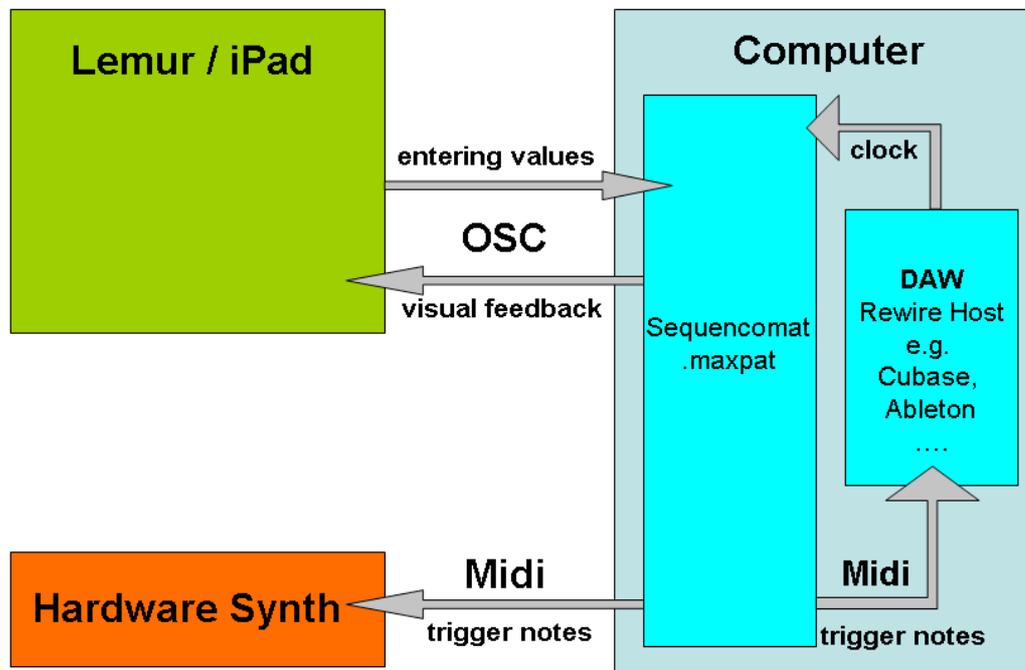
Dortmund, 18<sup>th</sup> march 2012

Matthias Wille

## Inhaltsverzeichnis

- Sequencomat V1 manual & documentation.....1
- Preface.....1
- How does it work?.....3
- Setup.....4
- Rewire host.....4
- Jazzeditor.....4
- OSC settings on Lemur.....5
- OSC and rewire settings on maxpatch.....6
- Midi .....8
- in short: setup for experienced users.....9
- The Lemur module.....10
- Overview.....10
- Grid.....11
- Pattern.....11
- Midichannel and Pitch.....12
- Rollpad.....13
- Trackvalues (Volume – Length – Random).....14
- Demo videos.....16
- FAQ.....17
- Lemur scripting.....19
- Pitch.....19
- Velocity.....19
- Length.....19
- Moving function containers.....20
- Presetsontop.....20
- Clear.....21
- Presets.....22
- Random.....23
- Max patching.....25
- p sequencomat .....26
- p ticks (subpatch in sequencomat).....26

## How does it work?



Sequencomat is a combination of a maxpatch and a Lemur control template. While values are controlled by the Lemur, all triggering and processing is done in max. That means it is **not** possible to send out Midinotes directly from the Lemur, you need a computer where the maxpatch runs, even if you use the sequencer for hardware.

Between maxmsp and the Lemur control surface is a bidirectional OSC connection. This sends all your changes on the Lemur to max and also visual feedback (e.g. steplight) from max to the Lemur. That OSC connection has to be set up first.

All MIDI is sent by the maxpatch to any Midiport installed on the system and chosen on the maxpatch surface. The Midisignal can be routed internal within the computer to trigger software synths. If a Midiinterface is installed, hardware synths can be triggered.

SequencomatV1 got no own clock section, so clock signal must be coming from a rewire host. Tempo and transport can be controlled on the Sequencomat Lemur template or within your DAW.

Setting up might look a bit complicated, if you are new to this. But no worries made, we will go through the setup and connection step by step next.

If you are completely new to the Lemur please refer to the Lemur manual for connecting Lemur and computer in common.

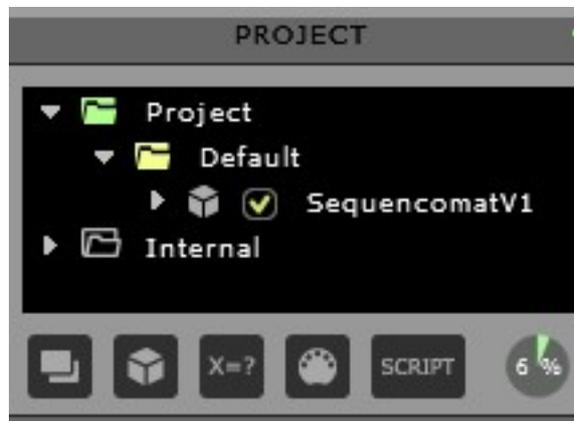
## Setup

### *Rewire host*

As SequencomatV1 has no own clock section it needs a rewire host (e.g. Ableton, Logic, Cubase) to sync to. Always **open this rewire host before starting max!** Only this way max knows that there is a rewire host it can sync to.

### *Jazzeditor*

Open the Jazzeditor (software to build controls and projects), create a new blank interface and insert the “SequencomatV1.jzlib” into it.



The interface (here: “Default”) can be named however you like, but never change the name of the “SequencomatV1” container, on the top hierarchical level of the module. That container name is part of the path for the OSC objects and therefore it is necessary.<sup>1</sup>

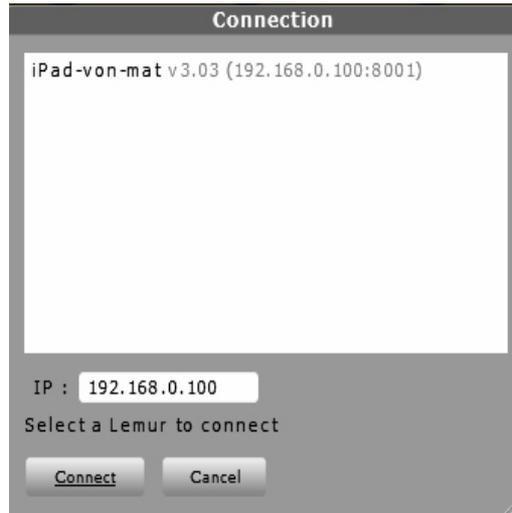
**Sequencomat is a module.** It can be insert into other existing projects by just adding an interface as described above or you can build a new project with it.

After insert SequencomatV1.jzlib connect the Jazzeditor to the Lemur and send the project, so it appears on the touchscreen.

You can see you Lemur IP in that popup menur that will appear during sync (see screenshot below). Remember that Lemur IP because you have to enter it later in the maxpatch.

---

<sup>1</sup> If you enter more than one Sequencomat in one project, the Jazzeditor will automatically change the container name to have clear addresses. That illustrates that you can run only one Sequencomat within the same project!

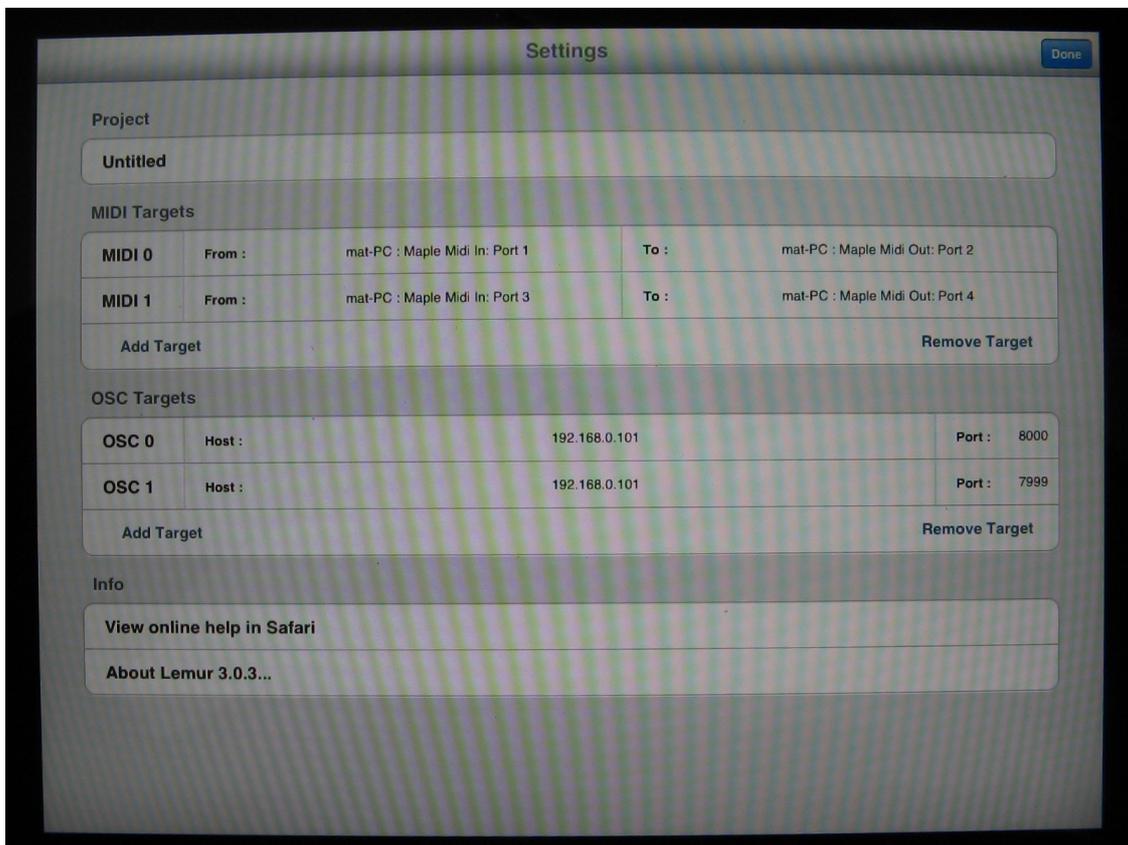


### OSC settings on Lemur

On the Lemur up to 8 OSCtargets for communicating with different software and/or computers can be defined.

**SequencomatV1 uses OSCtarget0 for Lemur objects** to communicate with the maxpatch. This OSCtarget0 is the standard target for objects – most modules use it – so probably it is already defined on your system.

You can check OSCtarget0 by going to the setting in your Lemur iPad App (button at top right) or on your hardware Lemur (hardware settings button / OSC page):



Within your OSCtargets you have to fill in 2 values:

- IP of your computer  
(check your network settings if unsure about your IP)
- port  
(use standard 8000)

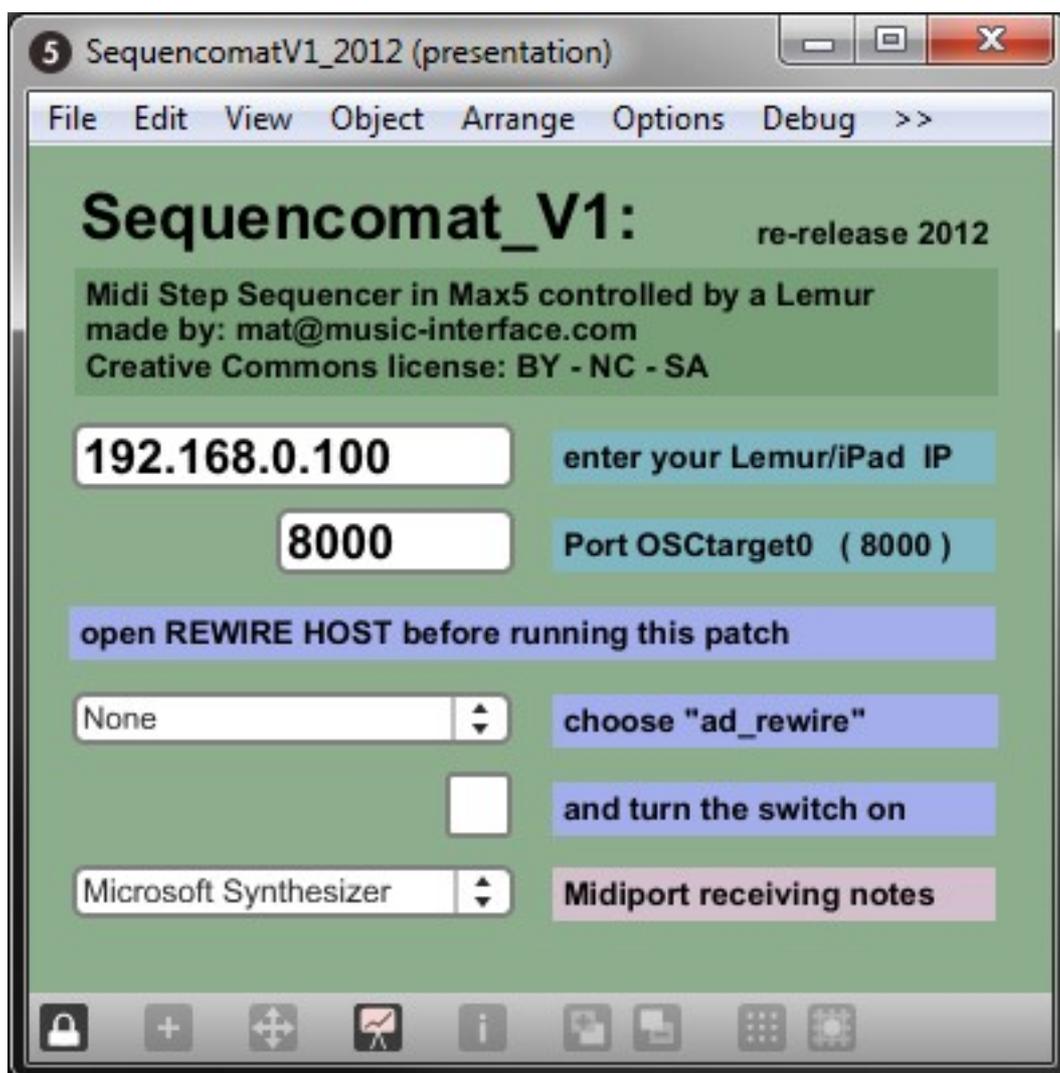
In the above picture my settings in Liines Lemur App are shown.

In this case OSCtargets 0 and 1 are defined. I use both with same computer IP but different ports. OSCtarget1 is only used for running V3 parallel, not necessary for V1....

### ***OSC and rewire settings on maxpatch***

Open cycling74s max/msp or max runtime. Max runtime is a free player for maxpatches you can [download at the cycling74s website](#). So you do not need to buy max to run the patch. However, if you want to edit or expand the patch you need full version of max.

**Load the “SequencomatV1\_2012.maxpat”**. You should have put it into a folder in the max library or in a folder that max knows (if you use another location max have to learn the path).



On the maxpatch **fill in your Lemur/iPad IP**.

If you are unsure about that IP you can check your settings on Lemur/iPad or you can also see it when you connect the Jazzeditor to your Lemur.

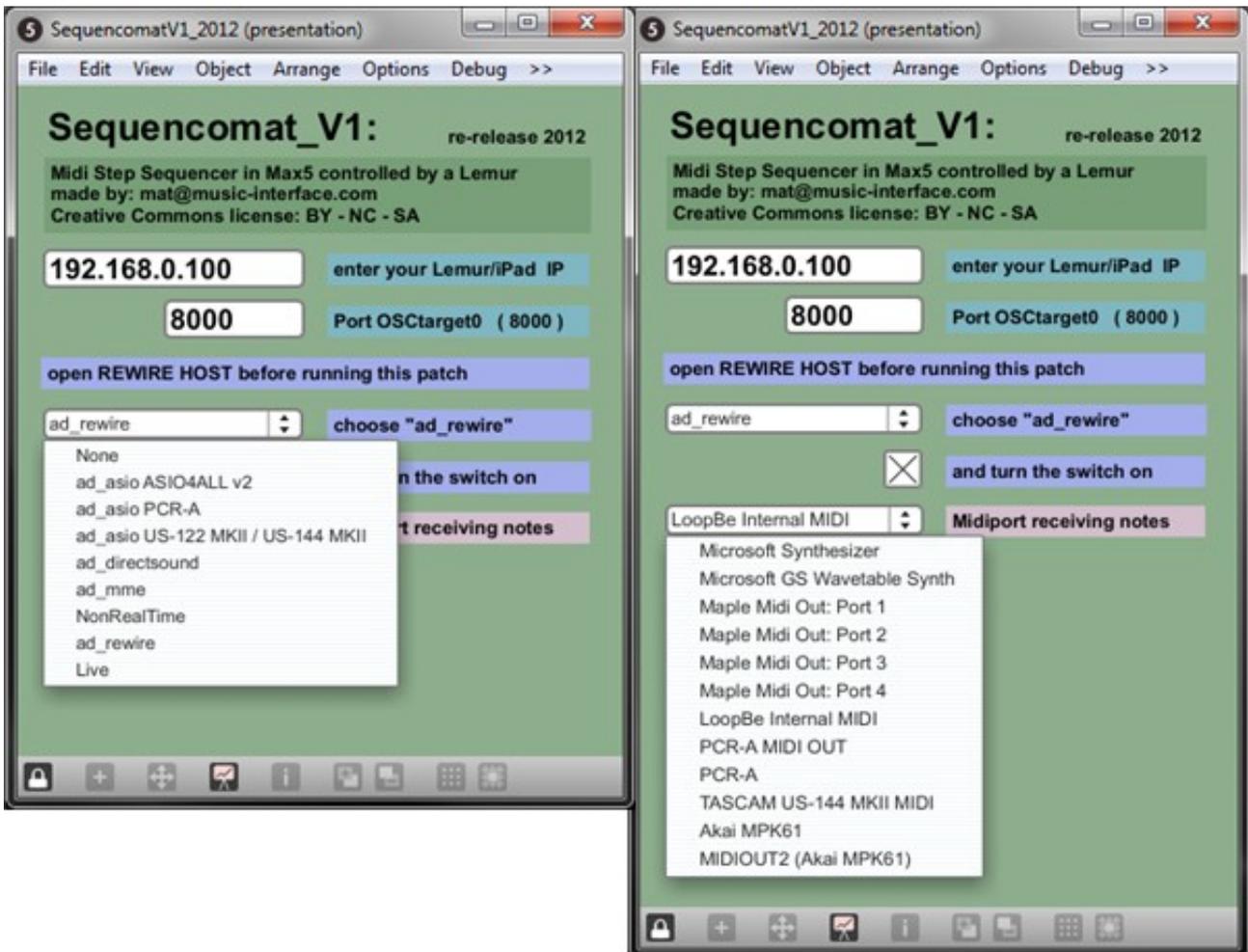
After entering the IP click somewhere in the patch to enter (pressing “enter” key does not work proper!)

As **port** use the same port you have used for OSCtarget0 on the Lemur.

Standard is 8000, and by default the maxpatch binding to that port.

Next open the drop down menu below and **choose “ad\_rewire”**.

Then turn on the engine with the switch (“X”).



Now the OSC connection and clock setup should already work. **Test it by pressing play on the Lemur**. If OSC connection and rewire setup is correct, you can control play/stop with the Lemur a yellow position index will be moving across the stepmatrix in play. **But no Midi notes yet...** (this will be enabled in the next chapter).

**If Sequencomat does not work at that point, check your OSC settings again!**

One **typical problem** is to mix up the IP addresses: In the maxpatch enter your iPad/hardware Lemur IP. In the OSC settings of Lemur (target0) enter your Computer IP.

Another problem can appear if max does not register your rewire host. In that case deactivate the engine (uncheck X), choose “none” instead of “ad\_rewire”, then choose “ad\_rewire” again and activate the engine again. If that does not help, close max/msp (runtime) and your DAW and start them again in the right order: DAW first, then max.

## **Midi**

On the maxpatch you choose a Midiport with the lowest dropdown menu. That Midiport will receive all notes.

**On Mac** you can use “from max runtime” (a virtual midiport that comes along with max, but only for Mac users) or use an IACbus (virtual Midiports of Apple OS). Do not use the Jazzaemon ports that will also appear in the drop down menu on Mac. These Jazzaemon ports are only for sending Midi directly from the Lemur, they do not work in max.

**On PC** you have to use 3<sup>rd</sup> party software as virtual Midiports. You can use:

- Loopbe (<http://nerds.de/en/loopbe1.html>)
- Maple Midi ([http://www.maplemidi.com/Maple\\_driver.html](http://www.maplemidi.com/Maple_driver.html)) or
- MidiYoke (<http://www.midiox.com/>)
- or any other virtual Midi cable..

You should use a different Midiport than you use for Lemurs Miditargets (that send Midi direct from the Lemur). In case you have to install them, do so first before starting max or DAW.

If you got a hardware Midiinterface you can also send Midi to that port and trigger external gear with the sequencer.

**In your DAW** you have to open Midi IN for that port you have chosen on the maxpatch. Look at the preferences of your DAW. You will find a sheet where you can see again all Midiports on your system. Open/activate the note IN for that port. Now your DAW receives notes from the sequencer in common, but you may still activate a track.

**On the track** choose again that Midiport for receiving notes and set a special Midichannel. *(This is only done to differ between tracks. For the beginning you can choose “all ins” and “all channels” for the track, but later on it is good to address one track to e.g. channel 2 with a drumrack and another track with a synth to channel 3. That way you can switch the instrument from the Lemur by changing Midichannels.)*

Furthermore you have to **activate that track**. You can monitor the IN of that Miditrack (in Ableton it is the orange button) or press record. Just the same as if you want to play notes with an USB keyboard.

**That's it.** Now everything should work. If it does not send Midinotes, check your Midiport again: is it used by another application? Is IN activated for that port? If still not work, try another Midiport. Look if it is activated in max (by default all Midiports are activated).

If you once have setup it correctly, next time it will be easier. OSCtargets on Lemur are saved within settings. Unfortunately the maxpatch of V1 does not save any values. You have to fill it in each time (you will remember your IP soon ;) ). SequencomatV3 – my actual sequencer – can save all settings in max.

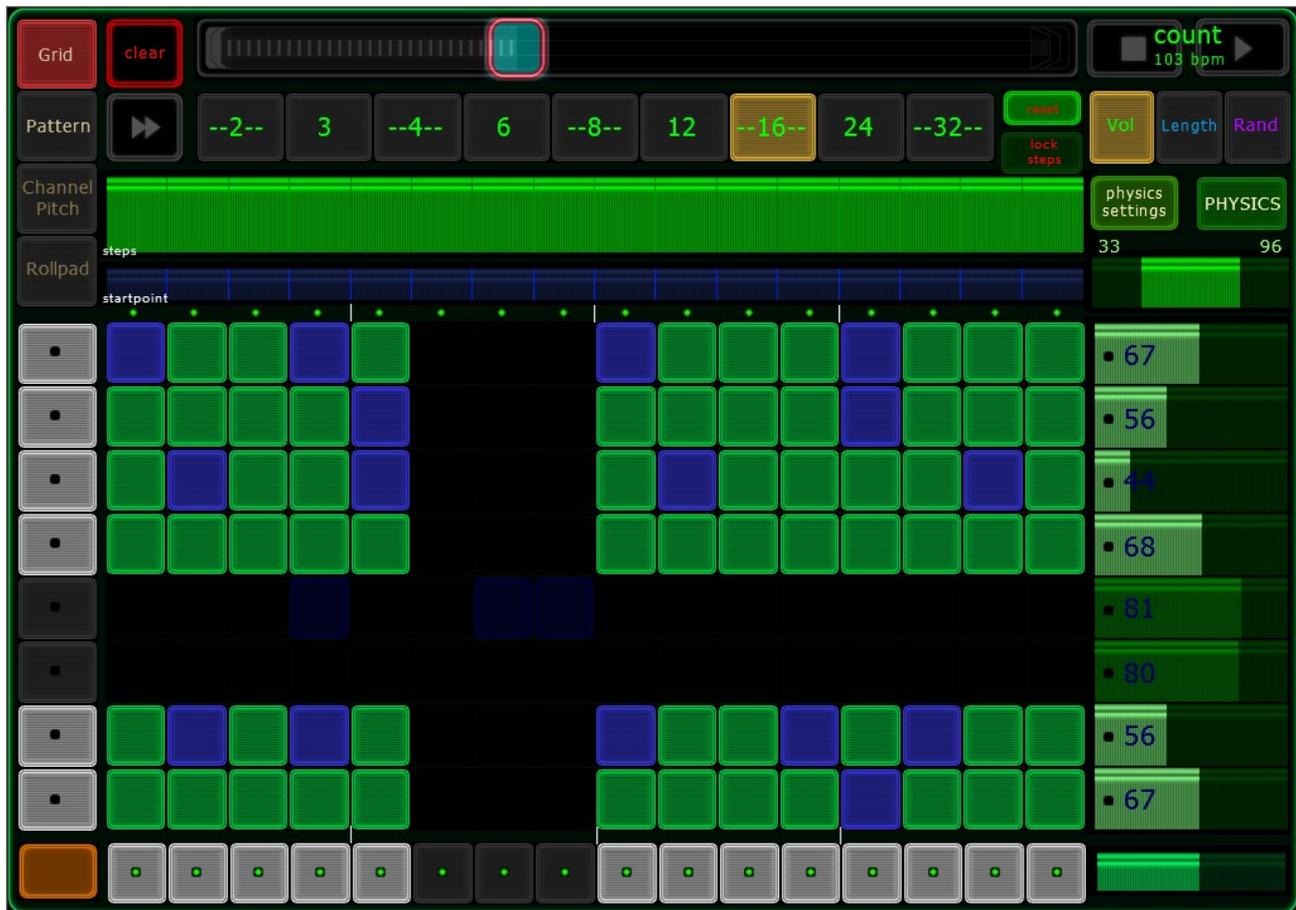
***in short: setup for experienced users***

It might be a good guideline for experienced users,  
but also a nice sum up and checklist for new users:

- Open rewire host before starting maxpatches
- insert „SequencomatV1.jzlib“ into a new interface in your lemur project
- set OSCtarget0 on lemur to your computer IP and port 8000
- load „SequencomatV1\_2012.maxpat“ with max/msp (free runtime)
  
- on maxpatch:
  - fill in your lemur IP (see lemur IP while sync jazzeditor and lemur)
  - enter port (default is 8000)
  - choose „ad\_rewire“ in drop down menu
  - X activate
  - => press play to test OSC/rewire communication
  - enter Midiport for note output
  - use another Midiport than those for lemur Miditygets
  
- in DAW:
  - open track IN for that Midiport in preferences
  - choose that Midiport on a track (and set Midichannel if needed)
  - activate track and record

## The Lemur module

### Overview



The main part of the Lemur module is a **8 x 16 step matrix**. Here the steps are set, which trigger the notes. The background is green and active steps are blue. If the sequencer is operating, a yellow position index will be shown.

On left side and bottom of the interface the **mute switches** for tracks and steps are located. If pressed the according track or step is muted and getting dark in the mainmatrix. Here track 5 and 6 and steps 6-8 are muted.

The switch at the left bottom corner will de-mute all tracks and steps at once (usefull e.g. at the end of a break)

The **top 1/3 of the interface** host some **main functions in moving containers** (Grid – Pattern – Channel/Pitch - Rollpad). The controls of this section can be changed by pressing the left radioswitch. Here Grid is shown, which lays in the background and has no container.

The **right 1/6 of the interface** host some **functions for trackvalues** (Volume – Length – Random). The content of the left side is changed by the switch below play/stop.

The **play/stop buttons** will be always in front. Both other sections can be switched independently.

## Grid

On Grid global tempo can be controlled with a big horizontal fader. (start/stop is right of it but not shown here, cause it is not part of this container)



The beat grid of the 16steps is controlled by a radioswitch. It includes triplet grids.

Left of that is a switch to inverse the playdirection.

Below the grid is the steprange. By default it is set to 16 – full range. But the range can also be shrunk, using less steps than 16, e.g. 12 for  $\frac{3}{4}$  rhythms or other for some square stuff. The steprange can be locked by a small switch on the right side. If locked, the range can be moved with one finger sliding above parts of the pattern. The reset will unlock the range again and set it to full range. The blue startpoint at the bottom can shift the whole pattern in 16<sup>th</sup>.

At the left top is a clear switch. It will clear all steps, set velocity, length and random to default and reset range. Notes and Midichannel are not cleared, as the basic settings might differ. You can expand the clear behavior within a script located at that button (see scripting).

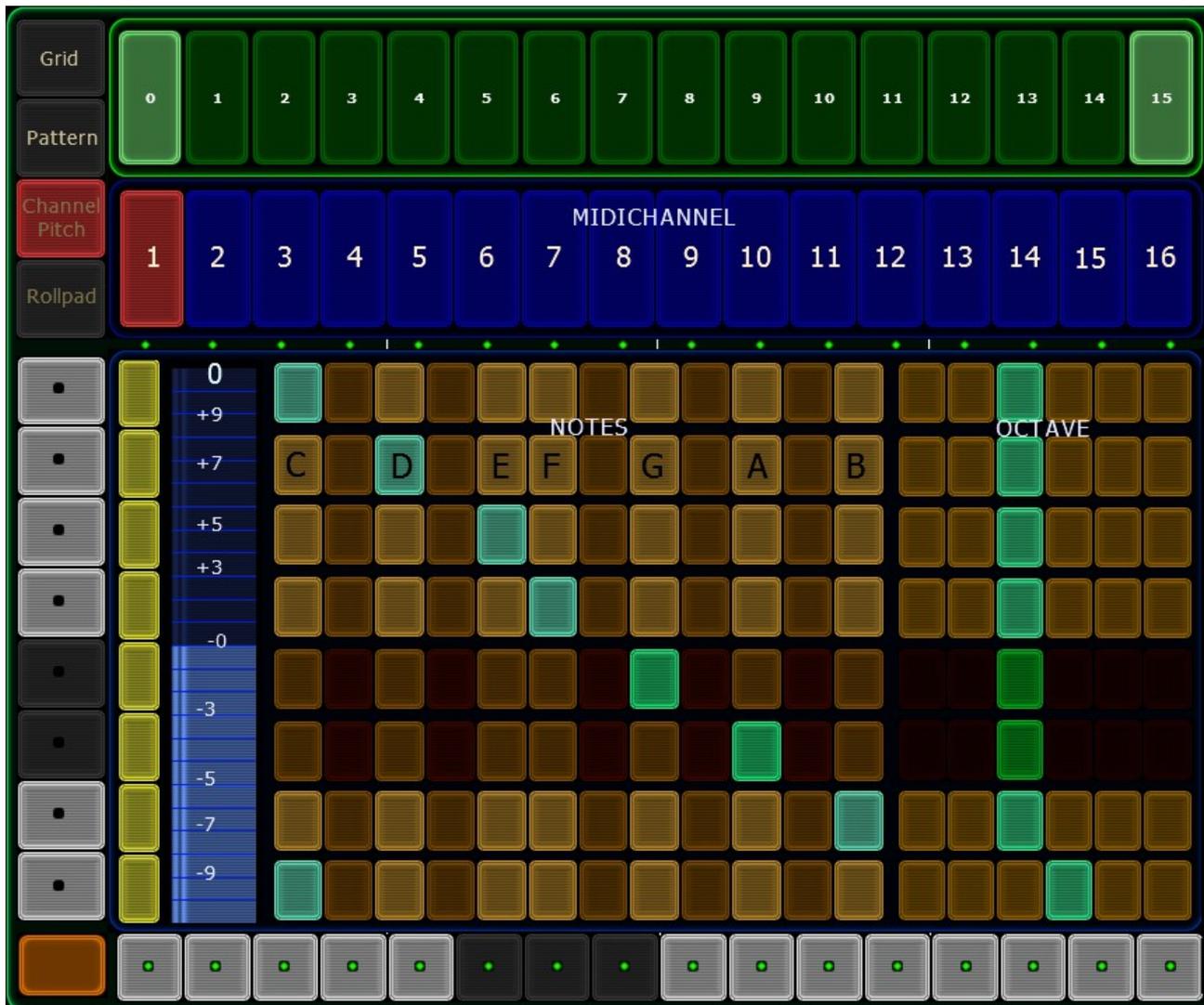
## Pattern



SequencomatV1 got 16 patterns. Save or load can be done during playback. To save just press a red slot. Slots already containing a pattern will be highlighted. To load press a green slot. Load is executed immediately, not with next bar. *(To reset pattern highlighting press 0 and 15 parallel on green load slots as shown in the picture)*

Presets are saved as values of the control objects on the Lemur. So preset handling is all done on Lemur, while maxpatch trigger those values. If the project is saved on the hardware Lemur or iPad App all presets will be saved within. If presets should be saved within jzlib – e.g. to enter the sequencer with all presets into another project – the Jazzeditor has to be opened and in sync with the Lemur all time during creating patterns!

## Midichannel and Pitch



On „Channel Pitch“ the main step matrix is covered by the pitch control. Steps will still be played and indicated by the LEDs (flashed whenever a tone is played), but to change steps in the matrix „Grid“ or „Pattern“ must be chosen.

The Midichannel of the output can be controlled by the blue switches. Jumping Midichannel enables to address different instruments on different patterns. The Midichannel can be changed during playback, because all notes on the old channel got a noteoff message when changing.

While Midichannel is set for all tracks, pitch can be set for each of the 8 tracks. Pitch is divided into note and octave. The notes buttons show a color according to white and black keys on a piano for orientation. The control objects of muted tracks are shaded dynamically (see track 5-6 in screenshot).

On the left side is a light blue transpose slider. All tracks can be transposed +/- 12 with that slider during playback for trying harmonic variations. It will be also saved within patterns. Transpose will be only done to those tracks which are activated with the yellow switches on far left. This enables to transpose only parts of the pattern.

The mute switches are still accessible while „Channel Pitch“ is opened. On top the „load preset“-container is shown. *If you prefer to see the grid control on top, deactivate the „presetontop“ script as shown in Lemur scripting.*

## Rollpad



„Rollpad“ is a performance XYpad for making flams and rolls. Not only with drums, it is also nice with synths. :) Again „Rollpad“ covers the stematrix, but playback will continue and mute switches are still available.

The XYpad sends notetriggering. The position on the horizontal axis sends the timing of the trigger: the more right, the faster the repeated triggering gets. The vertical axis sends volume: higher touches are getting louder. If a movement is made from left bottom corner to top right corner a line of repeated triggers getting faster and louder will be produced.

The **pitch** of the Rollpad is controlled by the green switches on the left. They play the pitch according to their tracksettings. You can change pitch during performance, e.g. playing melodies with the switches while performing on Rollpad.

The green range controls the range of **velocity** from top to bottom. Here it is 0-127, but if a variation only from 33 as lowest velocity to 57 is needed, just shrink the range! The velocity is shown in the green monitor at the left top and also in the position of the green pixel (SignalScope).

Parallel the output of the X axis – **speed of triggering** – can be also ranged. Below the Rollpad is a light blue range for that. The minimum and maximum time between triggers in milliseconds is indicated by the two light blue monitors on the right and left bottom of the Rollpad. The monitor in the middle shows the actual speed of triggering according to ball position.

At the bottom of the Rollpad is also a **slider for length** (purple). Length is not controlled by the ball position, only by this slider. If the slider is fully opened, the length of the triggers is nearly the time between 2 triggers. If the slider is turned down the triggers get shorter. This enables nice effects e.g. on HiHats or Snares. You can control the triggering with one hand and change length parallel with the other hand.

On the right side is a **red section for an additional ControlChange output**. This control change is not send by the maxpatch. It is send via Miditarget0 from Lemur directly!

Set the Midichannel and Controllernumber via the dropdownmenus at the top. Set the range of the CC output (according to ball position) with the red range object. CC output can be muted with the switch at the bottom and CC value will be shown in red monitor object on right top and the red pixel.

If the rollpad is touched, a CC will be send. Use this CC e.g. to open a distortion effect, which comes in more, the higher the ball is. If the finger leaves the touchpad, the CC falls back to the minimum value. So this is thought for a special effect on triggering only. If you like to play a Multiball for CC in common I suggest using my „4balls“ or „loopomat“ module from the user library on another interface.

On the top left are 3 little buttons. With „Physics“, **physics of the ball** can be activated which will give a more smooth variation of triggering. Use 2 or more fingers to control the ball in physics mode. If „set physics“ is pressed (and hold!) a popup container will appear where the 3 physic attributes can be set.

If „**Sync**“ is activated the **triggering is in sync with the sequencer**. Vertical lines will appear on the Rollpad representing e.g. 16<sup>th</sup>, 8<sup>th</sup> or also triplet grids. This mode is only triggering during playback.

The range for speed of triggering will disappear (cause it got no function in this mode) and the length Slider will get bigger. Length is still working the same way, but now is also relative to the speed of triggering (e.g. half length of a 8<sup>th</sup> or 16<sup>th</sup> note according to balls horizontal position).

### ***Trackvalues (Volume – Length – Random)***

On the left 1/6<sup>th</sup> of the Main interface is a second container section which can be moved independently from the above described layers for global controls. They are controlled by a switch on top of them (but below the play/stop section of the main interface).

See picture for an overview about those containers.

The big horizontal 8time MultiSlider will appear in line with its track. They control volume and length of the notes on the track and a function called „random“.

So it is important to know, that velocity and length are controlled „trackwise“ in V1.<sup>2</sup> To trigger the same note (drum) with 2 velocities, use same pitch on 2 tracks...

---

<sup>2</sup> On version 3 of Sequencomat these values are controlled step for step on each track („stepwise“).



The **bottom slider sets all** values of the MultiSlider. Above the MultiSlider is a **range for the output** of all values. Values are shown in **monitors** across each track.<sup>3</sup> **LEDs** will light up each time a trigger is send (so you see sending, even if stepmatrix is covered by Rollpad or Note).

**Length is done in absolute milliseconds** and is *not relative to grid/tempo!*

Range of Length is not linear, to enable a wide distribution of values.

**Overlapping notes** with same pitch will cut later notes based on midi noteoff message!

<sup>3</sup> In last version „range“ and „set-all“ changed place. Done to avoid setting all trackvalues accidentally by corner touch...

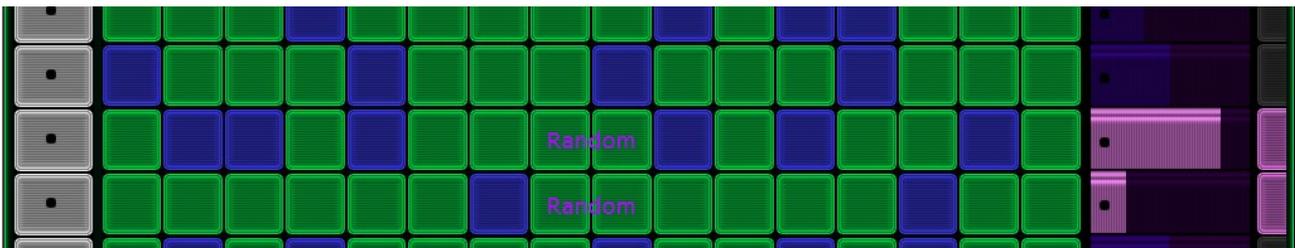
Volume and Length got also 2 switches for **physics**.

If physics is activated by the right switch the MultiSlider will start to move if touched (and therefore **variate values** according to interaction). Use range to control the minimum and maximum of output and bottom slider to set all trackvalues in a row. A nice trick is to stop physics – it freezes – and set some values special, then activate again. The physics will now follow that. Values are created in realtime<sup>4</sup>, so fast physics are more randomly while slow physical movement is more effective.

Press and hold the „**physics settings**“ button and a popup container will show where tension and friction can be set. Use slow tension for slow movement. While editing settings the button has to be hold. Leave your finger from popup container before lifting it from that button! (auto movement of container can cause value)

### Random variates the step position in the stepmatrix for a track.

Activate random by the switches on the far right. You will see the blue steps on the according tracks start to move and flicker: Step position is variated randomly. A small monitor across the track shows „Random“ as additional visual feedback to the jumping behaviour.



To **change the number of steps in variation** deactivate the switches. Otherwise steps will always fall back to the count at that time random was activated.

The MultiSlider controls the speed of variation. Slower values will lead to more replays with same position before next variation.

## Demo videos

You will find a lot of videos showing sequencers in action on my youtube channel:  
<http://www.youtube.com/user/tonvibration>

While shooting my videos I always try to demonstrate some functions. Sometimes I also explain how it works (english, with „slight“ german accent...) . While reading manuals is important (thank you for doing so!) watching videos is also nice and can be helpfull.

For V1 theres is a playlist (<http://www.youtube.com/playlist?list=PL55F20A2D3AFE7AE>)  
 The videos here show mostly old surface on hardware lemur, but function is the same.

<sup>4</sup> Well, more or less realtime... but the values are calculated on the lemur and send to max for triggering. In V3 to values are created in max and the lemur becomes more a „plain“ controller section. This makes interaction a bit faster and also reduces lemur memory.

## FAQ

### **Do I need to buy max/msp or max4live to play this patch?**

No. Cycling74s max/msp got a free runtime. This player can play patches, while you can not edit them. For using the sequencer as it is you need just the player :)

Max4live is a special unit of max/msp for Ableton Live. It is very integrated into Ableton which has its own pros and cons.... also got no free player :(

My patches are done in normal max/msp.

### **Does it run parallel with max4live?**

Yes. Normal maxpatches run perfect parallel with max4live.

If you like to use e.g. the sequencer within the MU framework, you can add it into MU as a custom interface (see MU manual for details).

### **Boohyah, all that max and OSC things....why not a sequencer on Lemur only?**

The Lemur controller got some nice scripting possibilities. You can move containers, handle values, store them, request and much more. But if it comes to clock accuracy – most important on a sequencer – a maxpatch is more stable and reliable.

Furthermore each line of scripting will shrink the lemur memory. \*hope for updates soon\* So it is better to let the Lemur „only“ be a controller sending and receiving OSC.

Running one more software does not harm. The free max runtime as player will not drop your system performance critically. And yes, loading one more thing during setup sucks, o.k.... but freedom in use and combination of controllers wins ;)

### **Is it possible to edit the patch?**

If you are into max/msp and own the full version you can edit. My patch is open. I release it under creativecommons BY – NC – SA. So feel free to add functions or take the whole patch into pieces for your own needs.

When sharing with others, please make sure to drop my name as original author ([mat@tonvibration.de](mailto:mat@tonvibration.de)). Commercial releases are not allowed without consulting me. But if you want to share with the community, I will appreciate that.

### **Where is the difference between all that versions of Sequencomat?**

I name all my stepsequencer SEQUENCOMAT. I like the name because it reflects the typically „Automat“ ending in german language and also drops „sequence“....and furthermore my shortname is „mat“.

The versions are according to their release date. With each change of version I had to do some major changes in the framework. In V3 I did a big jump while porting all value calculation and handling into max. Each version got some updates in following number.

In common the higher the version the more details you can control.

	v1	v2	v3 lite	v3 full
clock	rewire	rewire midimaster midislave	rewire midimaster midislave	rewire midimaster midislave
swing	-	-	global	trackwise
tracks	8	8	6	8
patterns	16	16	50	100
global values	midichannel grid playdirection steprange startpoint			
trackvalues	pitch velocity length random	midichannel grid playdirection steprange startpoint pitch velocity length random	midichannel grid playdirection steprange startpoint random	midichannel grid playdirection steprange startpoint random
stepvalues	steps	steps	steps pitch velocity length	steps pitch velocity length delay propability controlchange  + random for all stepvalues
avialable	legacy / iPad	legacy (discontinued)	legacy / iPad	legacy / iPad
release date	sept2009	nov2009	sept2011	may2010
last update <i>march2012</i>	iPad re-release march2012	V2.5 march2010	dec2011	V3.3 june2011
manual	(you read it)	<a href="#">online</a>	<a href="#">online</a>	

See a video comparison between SequencomatV1 and SequencomatV3.1 here:  
<http://www.youtube.com/watch?v=v4pdzmYzrxQ> (shot with legacy version august 2010)

## Lemur scripting

The triggering of this sequencer is done in max. But most variables in SequencomatV1 are calculated on the Lemur and also interaction is programmed here. So this chapter should give a short inside into this programming.

**If you just want to use the template as it is, it is not necessary to read this section!**

In case you want to change e.g. the appearance of the moving function containers or the actions associated with the clear button, this may help.

However, I recommend to go through the available tutorials about scripting on the lemur before. You may find those in the Lemur manual or here:

[http://www.jazzmutant.com/workshop\\_tutorialslist.php](http://www.jazzmutant.com/workshop_tutorialslist.php) [march 2012]

Especially those are very helpfull:

[http://www.jazzmutant.com/workshop\\_tutorialslist.php?id=storing](http://www.jazzmutant.com/workshop_tutorialslist.php?id=storing)

[http://www.jazzmutant.com/workshop\\_tutorialslist.php?id=multilineA](http://www.jazzmutant.com/workshop_tutorialslist.php?id=multilineA)

[http://www.jazzmutant.com/workshop\\_tutorialslist.php?id=multilineB](http://www.jazzmutant.com/workshop_tutorialslist.php?id=multilineB)

### **Pitch**

Is calculated for each track in a variable inside Settings. Variables are called note0.....note7 and contain a single concrete value reaching from 0-127.

**firstof(key0.x)+(firstof(oct0.x)\*12)+36+(transon.x[0]\*trans.out)**

It is based on the according key and oct values + keynote + Transpose-slider, if the according transon button for that track is pressed.

You can change the way pitch is calculated. As long as the variables are named the same and are within same position (pathway), max will use this.

### **Velocity**

Is calculated on lemur too and send to max. There are again 8 variables, this time located within Volume.Velocity. Variable names are vel0.....vel7 and are concrete values 0 - 127

**range(Velocity.x[0], Range.x[0], Range.x[1])\*127+1**

It ranges the according Slider (Velocity.x[0] for track 0) to the range, multiply with 127 and +1 to avoid Velocity 0 (which will be no note). Again these variables can be changed, but should stay in same path and with same name.

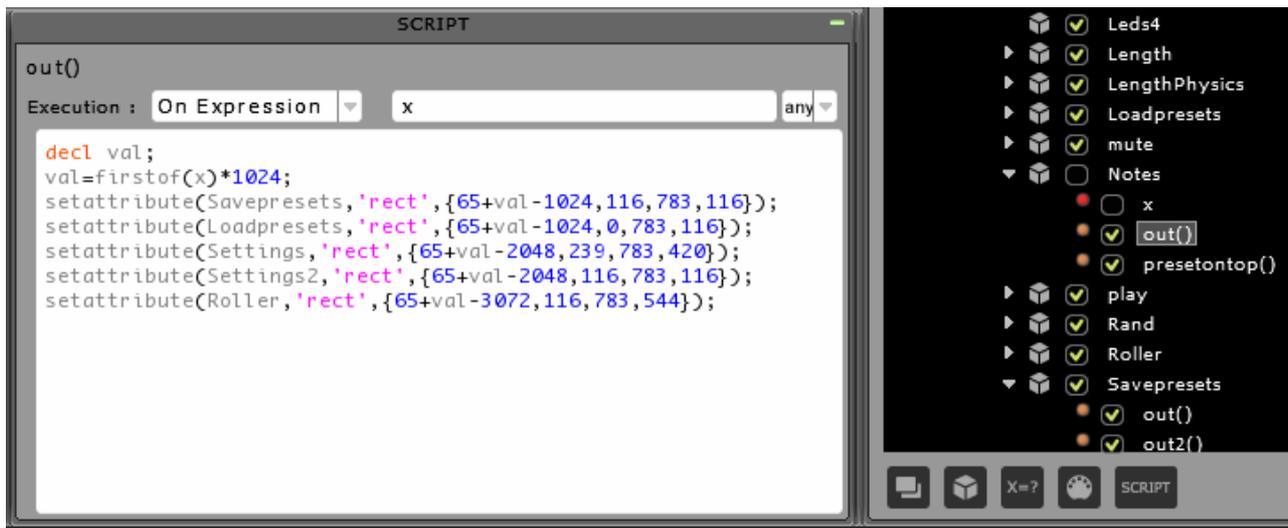
### **Length**

Again length values for notes are calculated on the lemur. They are located within Length.Velocity. Variable names are also vel0....vel7. So the pathway is important (Length vs Volume)

**range(Velocity.x[0]\*Velocity.x[0], Range.x[0], Range.x[1])\*3000+15**

The length variables send concrete values in milliseconds for each track. The math is a bit different to open control to a wide range of lengths.

## Moving function containers



Position of the function container is defined by a script under the „Notes“ switch – a radio switch which changes the shown container.

The first 2 line declare a **value („val“)** based on the **firstof-function** of the 4times „Notes“ Radioswitch. That firstof will output 0 if the first/highest switch is activated, upscending with each switch by 1. (here:0-3 for handling 4 functiongroups)

This is **multiplied with 1024 for the horizontal resolution of the iPad.** (for legacy lemur this had to be set to 800)

As the script lays within the Notes-switches, the term „x“ has not to be defined in pathway. For adressing objects outside an object use „Objectname.x“ The script is executed on any change of x – whenever the radioswitch changes.

All other lines work like: ***setattribute(Containername, 'rect', {x,y,width,height});***

The containers attributes are only changed in its x-position dynamically.

X is the horizontal position of the container from left to right. Here all Containers will show up 65 pixel from the left side. Followed by functions like „+val-1024“, „+val-2048“, „+val-3072“. The numbers are multipliers of 1024, referring to Containers position. This script will cause the containers to move in a horizontal line from left to right, where only the corresponding function is in sight, while the others lay in  $x < -1024$  or  $x > 1024$ ; not displayed.

## Presetontop

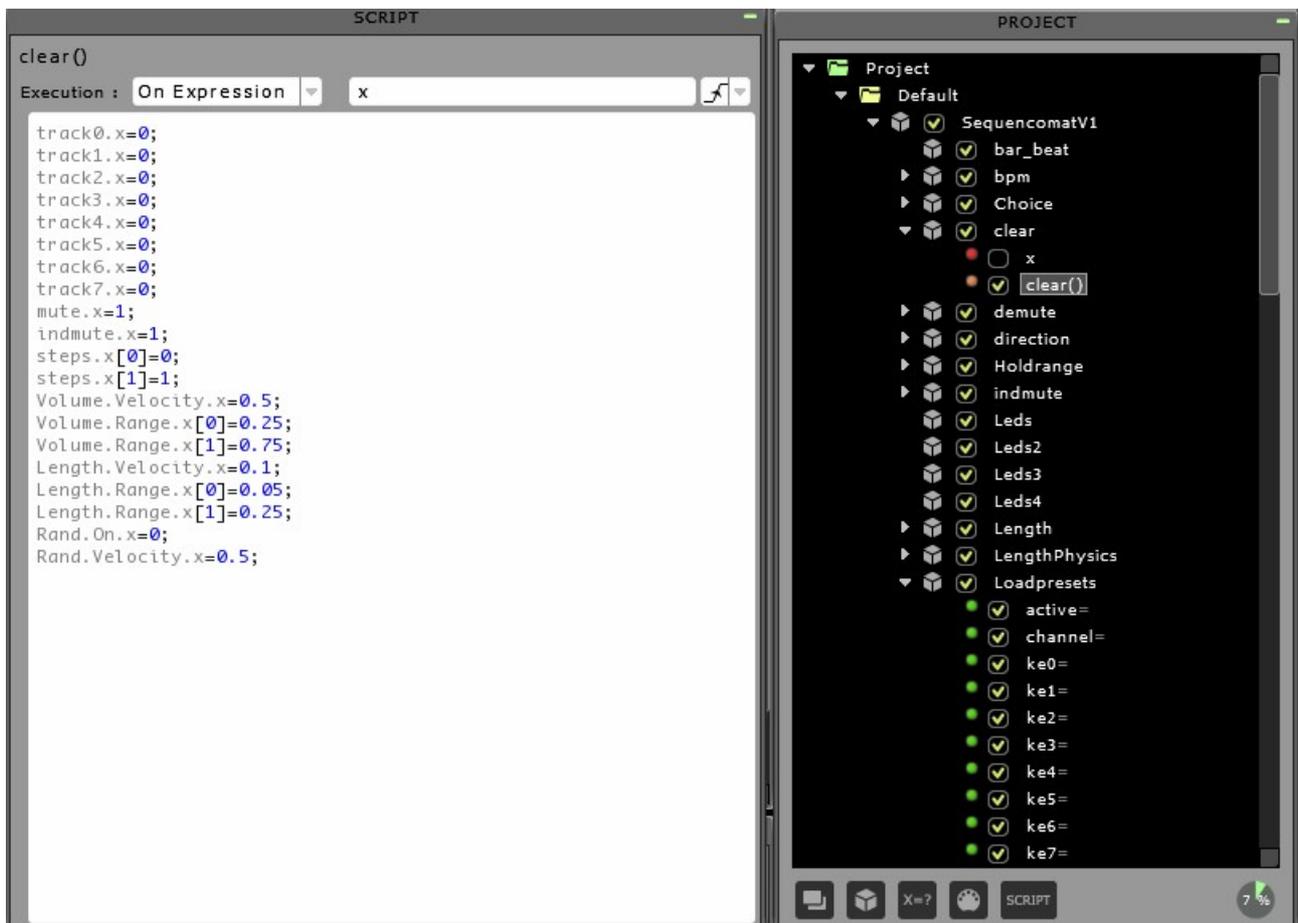
A second script named „presetontop“ makes the Loadpreset container appear whenever the firstof value is 2 or higher. That means the Loadpresets is also visible if function „Channel Pitch“ or „Rollpad“ is chosen.

I like to have presets available for pattern jumping when performing on Rollpad or changing notes, but if you prefer showing the grid, bpm slider and clear instead, you can just uncheck the script on the right side under „Notes“.



## Clear

Clear is also done within Lemur.



As you see in the screenshot below the clear script is relative to the clear button. *Most of the action scripts in the sequencer are within the button.*

It is activated if the x of the clearbutton goes from 0 to 1, so if it is pressed. Release the button makes no action.

The lines set the Lemur control object to special positions. As the max will listen to that control objects values are also cleared/changed.

track0.x is the first/highest track. By setting it to 0 it erases all steps. If you want to set it to a special pattern use a vector like {1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0} for a 4 on the floor.

Same for the other tracks and for mute (trackmute) and indmute (stepmute, „individual“ mute)

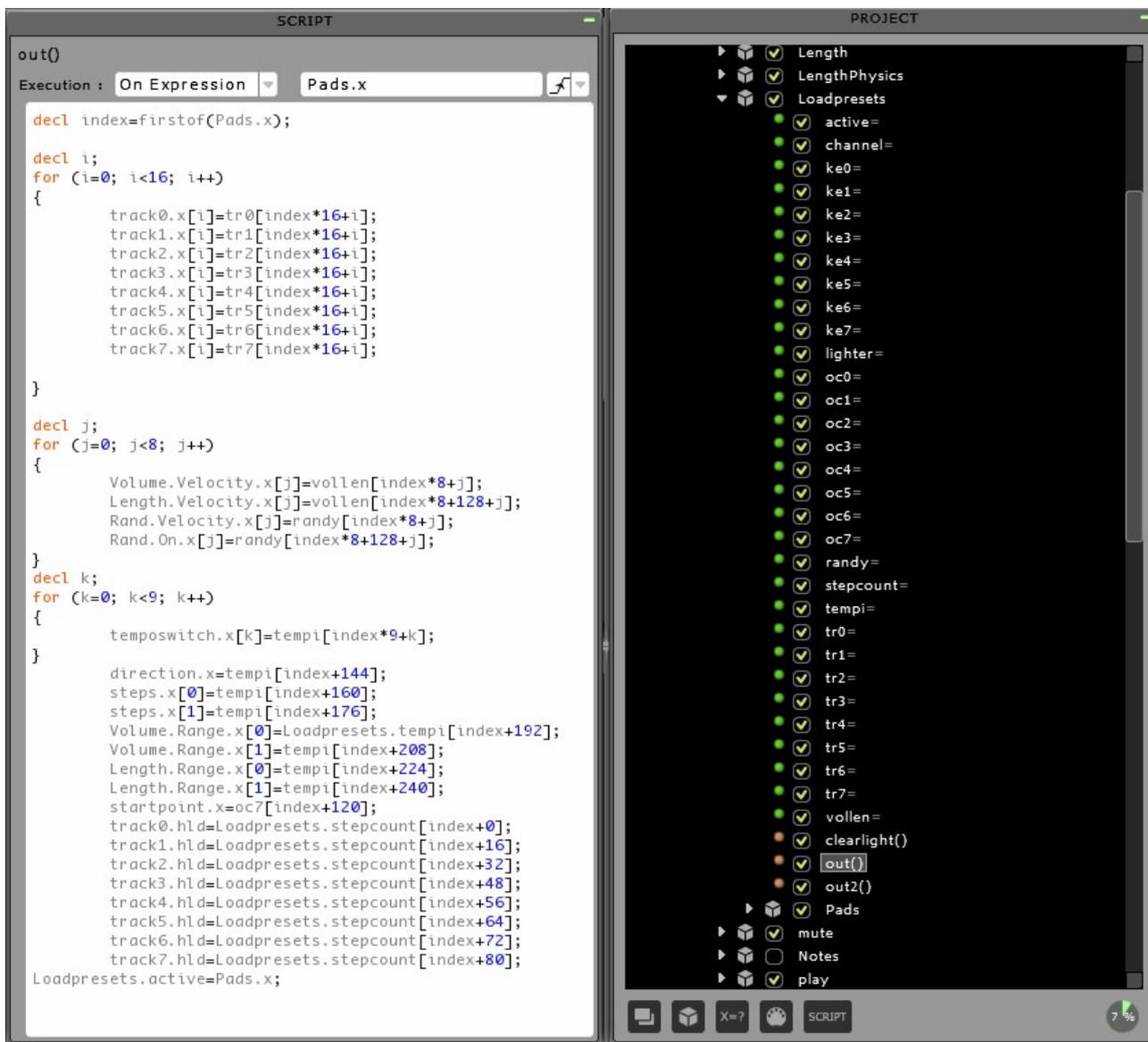
steps.x[0] and x[1] are minimal and maximal value of steprange. Alternativ you can write it in one line with steps.x={({0,1})}

The other lines control the trackvalue by same principles.

Notes and Octave are not reseted, cause they depend on setting of the track which will change often. But you can add that to the script, using vectors.

## Presets

All values are saved as positions of the control object on the lemur.



The saving/loading scripting is based on a special tutorial in Jazzmutants old workshop section: [http://www.jazzmutant.com/workshop\\_tutorialslist.php?id=storing](http://www.jazzmutant.com/workshop_tutorialslist.php?id=storing)

Loading presets is done with scripts inside the Loadpresets container. Also all variables that store and recall the values are located here.

For an overview I show loading. Saving functions are within the Savepresets container. They work nearly the same, just reverse and will be self explaining if loading presets is understood.

The first line ***decl index=firstof(Pads.x);*** will output a variable according to the pressed loadpattern slot between 0 and 15.

***decl i;***

***for (i=0; i<16; i++)***

will cause a loop going through 0-15 for everything which comes behind in brackets { .. }.

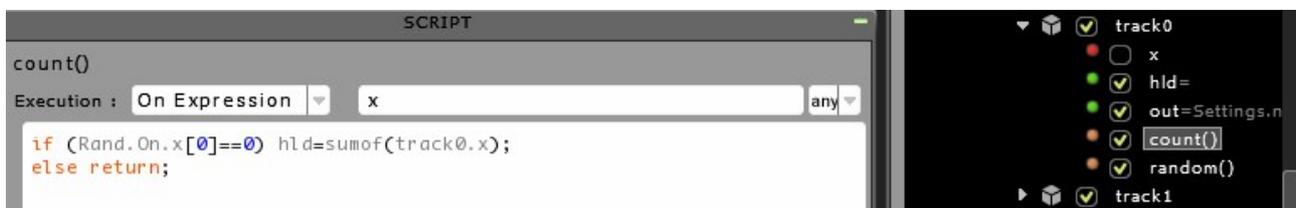
***track0.x[i]=tr0[index\*16+i];***

that will set the steps on track 0 [from x[0] to x[15] – all individual steps] to values on the variable „tr0“. This variable will contain a vector with 256 values. It will get the values from that position in vector which refers to [index\*16+i]. Startposition is defined by index\*16, showing a value according to pressed preset slot (0,16,32,48,64....). Then again a loop [i] will update all 16 steps.

All tracks work that way and velocities and length work alike.

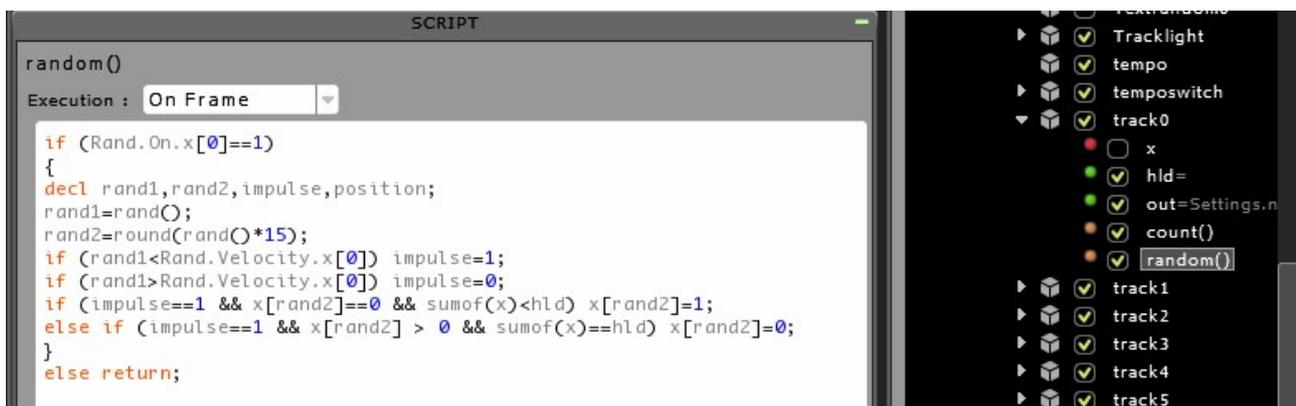
## Random

The random variation on step position is also programmed inside the lemur. The scripts are relative to the 8 tracks and are triggered on frame. See screenshots below for track0.



„hld“ is a variable. A constant that reflects the number of steps activated on that track.

It will be updated any time that a step is entered or deleted (on expression x) as long as the random switch on the according track [0] is off. Step count is not updated if random is active. That will make sure, that numbers of steps in variation stays the same during random.



Whenever the according random switch on a track (here: „Rand.On.x[0]“) is activated, it will declare 4 variables:

- rand1 = a random value between 0 and 1
- rand2 = a random concrete value from 0 - 15
- impulse
- position

if (rand1<Rand.Velocity.x[0]) – if the random value on frame is smaller than the Rand-MultiSlider-value on the according track [0] – „impulse“ will be set to 1.

With next line, it will be otherwise set to 0.

if impulse is 1 AND and a random step [rand2] is off (0) AND the sum of all steps is smaller than „hld“ THEN activate the step.

else if impulse is 1 AND and a random step [rand2] is on (>0) AND the sum of all steps is exactly „hld“ THEN deactivate the step.

**out = Settings.note0\*x\*mute.x[0]\*indmute.x has nothing to do with random. It sends the pitch and mute information to max via OSC.**

## Max patching

To give you a start with lemur and sequencing I suggest the following tutorials in the Jazzmutant section:

[http://www.jazzmutant.com/workshop\\_tutorialslist.php?id=maxstepA](http://www.jazzmutant.com/workshop_tutorialslist.php?id=maxstepA)

[http://www.jazzmutant.com/workshop\\_tutorialslist.php?id=maxstepB](http://www.jazzmutant.com/workshop_tutorialslist.php?id=maxstepB)

These are great tutorials Mikael Björk (LoFi Massakah) once gave.

I expanded the tutorial by some more functions,

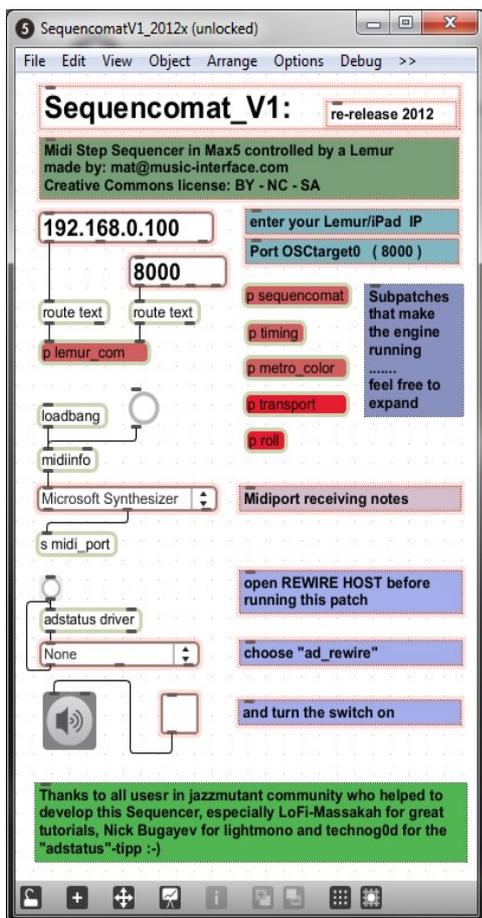
„Monomat“: <http://liine.net/en/community/user-library/view/129/>

Threat in JM: <http://forum.jazzmutant.com/viewtopic.php?t=2264>

After that SequencomatV1 was my next release. I updated and expanded the functions several times, but there is no version log. The development can also be seen in the videos: (<http://www.youtube.com/playlist?list=PL55F20A2D3AFE7AE>)

For rerelease 2012 I only changed the surface appearance of the patch and fixed a bug on rollpad triggering.

If you open the maxpatch, you will see the red subpatches. Doubleclick on them to open:



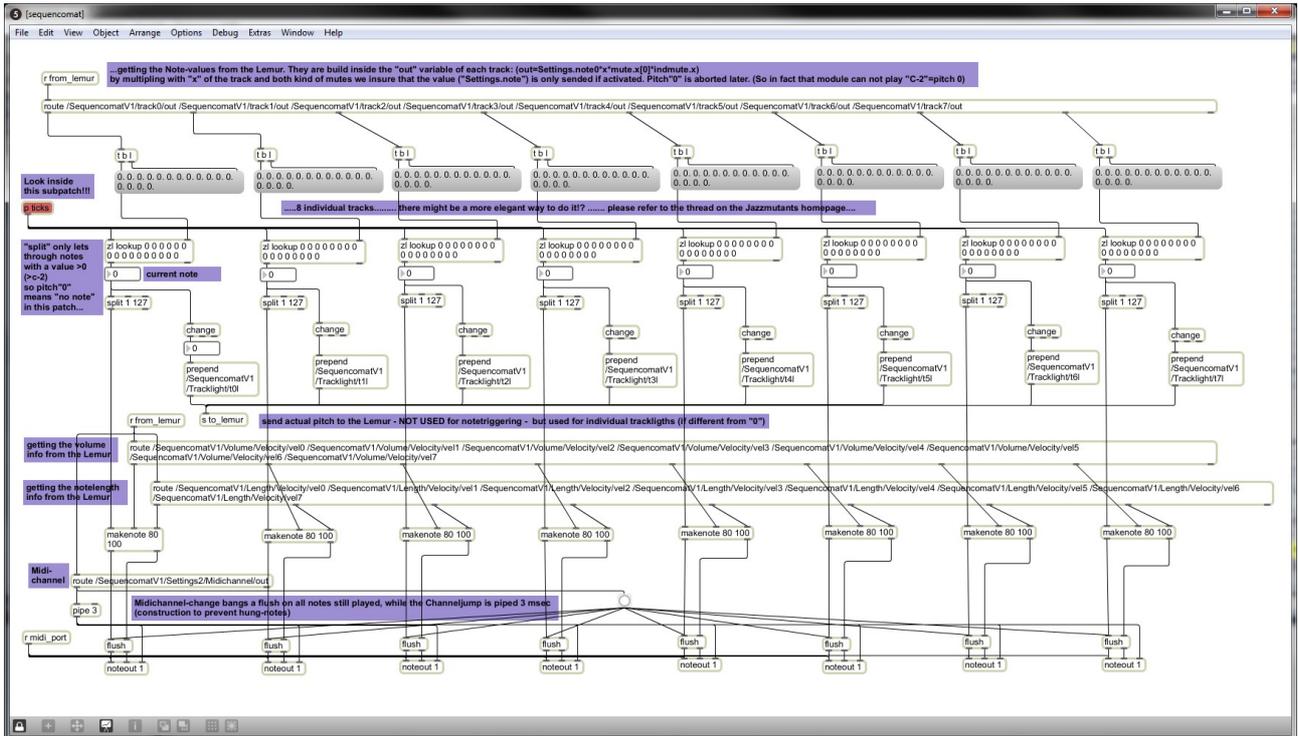
- **p lemur\_com** handles all OSC communication and is taken directly from Mikael's tutorials
- **p sequencomat** combines all 8 tracks in stripes
  - **subpatch tick** calculates the actual step, based on timing, grid, steprange, startpoint.
- **p timing** sends out 9 variables („2n“, „2tn“, „4n“, „4tn“, „8n“, „8tn“, „16n“, „16tn“, „32n“). they reflect „steps since songstart“ according to different grid. variables are used within „tick“ in combination with % to set actual step.
- **p metro\_color** hosts the main matrix color incl. Steplight and LED
- **p transport** communication with rewire host
- **p roll** contains pitch and trigger information of the Rollpad

Some parts of the patches are commented, which might help a bit to understand the structure.

If you like to expand the patch or use parts of it in your own, I would appreciate. It is released under creative commons BY-NC-SA. I will be happy if you contact me with any news on this sequencer or music made with it.

Contact: [mat@tonvibration.de](mailto:mat@tonvibration.de)

# p sequencomat



# p ticks (subpatch in sequencomat)

