

THE SEQUETRON

ADVANCED OPERATION

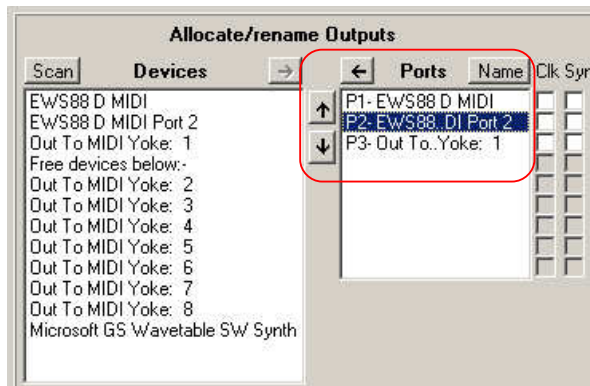
© Phil Tipping, 2011, www.philizound.co.uk

Last updated on 5-Apr-2011 for V7.21

The Sequetron Advanced Operation.....	1
1. Multiple outputs	2
2. Multiple inputs	4
Examples	4
3. Synchronisation.....	5
4. Sequence Parts.....	6
5. Key maps	7
6. Saving & loading MIDI files.....	8
Saving.....	8
Loading	8
Non-Sequetron files	9

1. MULTIPLE OUTPUTS

You can connect different output devices to each sequence and/or the live output, e.g. split the sequences between drum machines and synths.



As you allocate more output ports, they are assigned port ids P1, P2, P3 etc. The up/down arrows move the selected port onto a different port number.

If there is more than one output port, the on-screen config shows drop-down lists for each sequence.

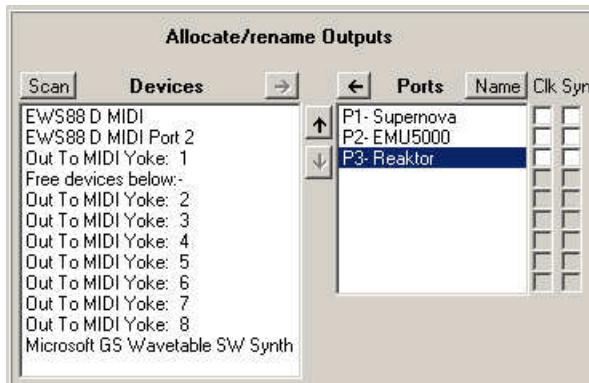
You can preset these at config time...



...or change them at run-time.

At run-time, all on-screen controls disappear to give an uncluttered display; all control is handed over to the MIDI keyboard.

The port command expects a value 1 to 9 which corresponds to the allocated port ids P1 to P9.



The system-assigned port names can sometimes be too long for the display, or not very descriptive. The Sequetron will abbreviate them to fit, but you can rename them as you like by clicking the Name button (or just double-click the entry)...

... giving a more useful display on the main screen in config and run modes:



Alternatively, you can drive multiple devices using a *single* port, with each device set to respond to a different MIDI *channel*. Either daisy-chain the devices via MIDI thru, or preferably drive them individually via a MIDI splitter box. You can now use the Sequetron's channel command instead of the port command.

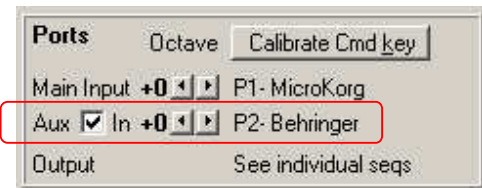
The port, channel & patch for each sequence can be changed on-the-fly either manually via the normal seq commands, or automatically via the recording process; if you change the values for the live output first (e.g. so you can hear the effect), those values are automatically transferred to the target sequence(s) when recording starts.

2. MULTIPLE INPUTS

You can connect up to nine MIDI sources. As you allocate more input ports, they are assigned port IDs P1, P2, P3 etc. The allocation & renaming controls are the same as those used for the output ports.

P1 is always the *main* input, and MIDI from this port is always interpreted for playing & controlling the Sequetron as normal. Ports P3..P9 are 'raw' inputs as they are never interpreted; incoming MIDI can be recorded and/or fed straight through to the Live output.

Port P2 can be configured either as an *aux* interpreted port, or a raw port.



EXAMPLES

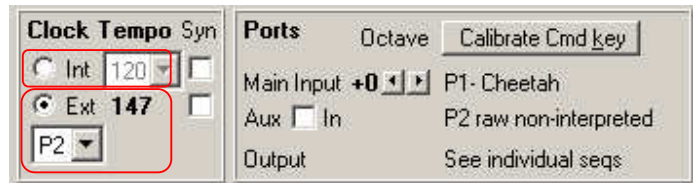
1. Use the aux port to offload control keys, such as cmd, esc, 'stop/play', from the main keyboard to maximise the number of play/record keys, e.g.
 - Connect another device (keyboard, foot controller etc), to the 2nd input port.
 - Assign the reqd. functions to keys outside the range of your main keyboard (e.g. the far ends of the MIDI range C-2 to G8).
 - Program the device (if necessary) to send these note-on & off messages.
2. Program the aux device to send a *complete* command sequence at the push of a button.
3. Use the aux port to record MIDI sequences from external players with the rec-nn command; see Command Reference for details. If you synchronise the external device to the Sequetron, the recording will be locked to the metronome and quantised as reqd. depending on the NQ & SQ settings. You may need to try a few values for 'nn' to give the desired seq length as indicated on the bar length display. The built-in drum patterns were all recorded from an arranger keyboard using this technique with NQ=0, SQ=1/8 to give exact bar-length patterns suitable for looping.

3. SYNCHRONISATION

Synchronisation with external devices uses MIDI time clocks and start, stop & continue messages. MIDI Time Code (MTC) is not yet supported.

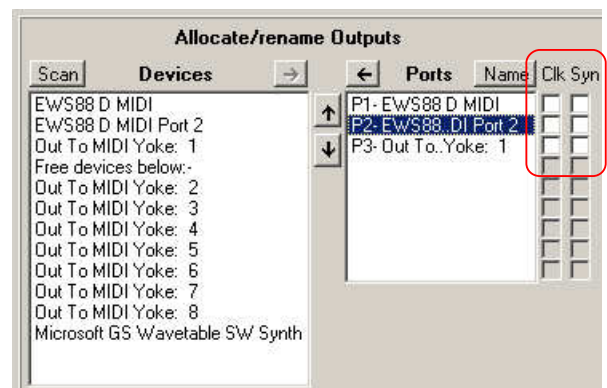
Set the tempo sync option to 'internal' to synchronise output devices to the Sequetron.

Set to external to synchronise the Sequetron to the nominated input port.

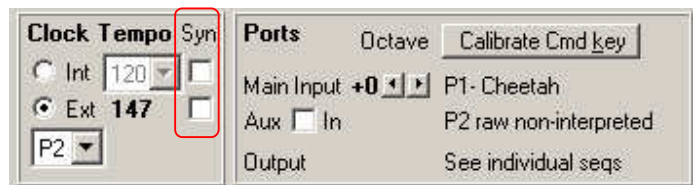


If the clocks are internal, MIDI clocks & start/stop messages can be sent to selected ports by setting options on the Port allocation screen.

A start message is sent whenever the first seq is started.



Incoming start & stop messages can be configured to control the Sequetron, irrespective of the clock source.



An incoming start message acts as a Play command, except that all seqs must first be stopped. Initially the Sequetron needs to know which seqs are to be started this way, so select them first and leave the keyboard in command mode.

An incoming stop message only takes effect in live mode, and stops *all* seqs whether or not they are selected. Any seqs which are running at the time will be automatically 'selected' for subsequent start messages occurring in live mode, although you can override this by selecting them manually as in the previous paragraph.

Seqs cannot be 'paused' mid-sequence, so 'continue' messages can be configured to be either ignored, or treated as a start message.

4. SEQUENCE PARTS

Each visible sequence is the 'active' part out of a set of 9 possible parts. The active part number can be changed with the part command. Non-active parts are 'paused' until they become active again, but all contents, attributes and running states are retained, including port nos.

One use for this is to park your current patterns to one side while you record another set, e.g. assuming all seqs are set to part 1, stop all seqs, then press: **cmd, part, 2**. The part 2 set will appear. To re-instate part 1, press: **cmd, part, 1**.

If any seqs are running, they will switch at the start of their next cycle, producing some interesting morphing effects.

If you precede the command by one or more seq-selects, you can swap out any *combination* of seqs, e.g.

cmd, seq-3, seq-4, part, 8 swaps the parts for seqs 3 & 4 with their part 8's.

This example shows seq2 has part 2 active, and all the others have part 1 active.



Sq.Pt	Ports	Chan	Prog
1.1	P1- Supernova	1	-
2.2	P1- Supernova	1	-
3.1	P1- Supernova	1	-
4.1	P2- EMU5000	1	-
Live	P3- Reaktor	1	-

Control

Status - **Running**

Last -

Input - Live

Load Input Metr Status Diag

Stop

The command can be automated using the sweep modifier; see the Command Reference for more details.

You can also park seqs to one side using save & load, which is more permanent, but currently you cannot load a part if it is running.

5. KEY MAPS

Key mapping has been designed to be as flexible as possible. Functions do not have to be mapped if you don't need them, and the same functions can be mapped to several keys if this is more convenient. The only restriction is that the pitch origin function must be mapped to a single key only.

Skip the rest of this section on first reading if you've had enough mental logistics!

The factory settings map the esc function for each mode onto the same key as the cmd key (bottom C) to simplify the keyboard template, but you can choose to map them to separate keys, or not to have an esc key at all in some modes.

The advantage of having an esc key is that it provides an escape (abort) function if you want to cancel the command sequence and return to live mode immediately. If you don't abort, you have to complete the command by entering any values it requires. The disadvantage with an esc key is that it ties up a key which may otherwise be useful.

e.g. the metr command expects two keys to be pressed for the bar and beat notes. If you assign an esc key in the metr value mode for aborting the command, you cannot use that key as one of the notes. If you don't have an esc key, you can use *all* keys for the notes, but you then have to press two of them to complete the command and return to live mode.

Similarly for record mode, where there are several functions available for stopping the recording and immediately playing it back. If these functions are assigned to keys, then those keys cannot be recorded. The novice factory mapping ties up two keys for control: an esc key for aborting or stopping recording, and a play key for stopping and immediately playing forever. The standard factory mapping ties up an additional key for the mute2 function.

The pitch value mode only needs an esc key if you want to also use the pitch-mode command as it's the only way of aborting from pitch value mode back to live mode. If you don't want the pitch-mode function and don't map an esc key, then all keys are available to be mapped as relative pitch offset values, immediately returning to live mode when pressed.

More ideas/comments are in the config file; see the Configuration guide.

6. SAVING & LOADING MIDI FILES

To see how seqs & parts map to MIDI tracks, start diagnostics before loading/saving.

SAVING

There are two types of save: full and partial, depending on the number of seqs selected. They both create standard MIDI format-1 files, but differ in the amount of information saved.

A *full* save is the default, and stores each part of each seq on a separate track. If you are using the Sequetron full version with 9 parts per seq, the first 9 tracks are the 9 parts for seq 1, the next 9 tracks are the parts for seq 2 etc. The LE version has fewer parts per seq, but uses a similar convention.

Full saves are performed either by the File, Save... menu command, or by the run-time save command if all seqs are selected.

A *partial* save stores the current active part of each selected seq on each track. If a single seq is selected, the file will only contain a single track so a MIDI format-0 file is created.

All files saved at run-time are called scenes. They are automatically named full or partial and stored in a scene folder, holding up to 99 scenes of each type (fewer with the LE version). The 'File, Open...', menu command opens the scene folder for use with external programs if reqd.

LOADING

Standard MIDI files can be loaded at config time via the File, Load... menu, or drag & drop into the main window, or at run-time via the load command.

A full load is the default, and tracks from a *full* file are loaded into all parts of all seqs at config time, or if all seqs are selected at run-time.

Tracks from a *partial* file are loaded in the order they are found into the current active part of selected seqs; all other parts are unchanged. This allows you to re-order seq parts, e.g. if you select seqs 4 & 5, the first two tracks in a partial file will load into the active parts of seqs 4 & 5.

The run-time command loads full or partial files from the scene folder as necessary. You can open the scene folder using 'File, Open...', and use external files renamed as reqd (although all external files are treated as partial files irrespective of their names).

NON-SEQUETRON FILES

These external files *are* loadable but with no quantisation or editing support, so you may need to use other programs to tweak them first, e.g. if the seq length is not synchronised for looping or the time division setting is not scalable by the Sequetron.

If you can 'play' the file with another MIDI player, then quantisation & synchronisation *is* possible using the rec-nn command via one of the raw input ports; see Command Reference for details.

All external files are treated as partial files, so each track loads into the next selected active part. The first channel prefix event in the track assigns the channel for that part; any further channel information within a track is ignored.

A MIDI format-0 file track is currently loaded into a single seq part, so will play back on the single channel assigned for that part.