

SEQUENTIX GmbH

CIRKLON SEQUENCER

OPERATION MANUAL

Firmware V1.22

© 2024

Table Of Contents

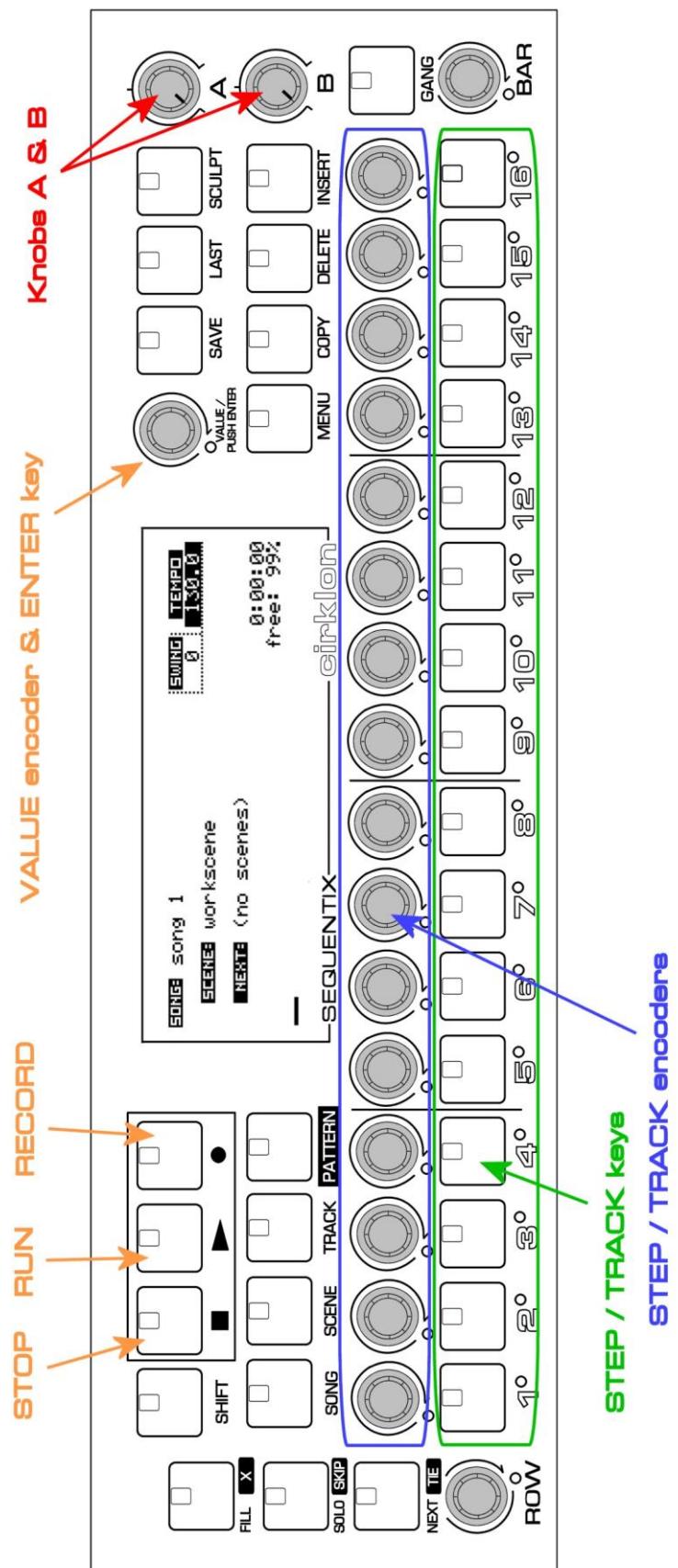
Table Of Contents.....	1-3
Getting Started with Cirklon	1-9
1. The Cirklon Concept.....	1-11
Song Play Mode.....	1-14
2. The Main Pages.....	2-1
Using more than 16 tracks.....	2-3
Track Muting and Status	2-4
Solo mode	2-4
Changing Song Values	2-5
Auto Name Generation	2-7
Creating and Assigning an Instrument.....	2-8
Deleting An Instrument.....	2-12
Poly Spread	2-12
Creating and Assigning a Pattern.....	2-14
Fill Patterns.....	2-16
Track Swap	2-17
3. Pattern Edit.....	3-1
What's in a P3 Pattern ?.....	3-2
Editing Pattern Values.....	3-3
Editing Multiple Steps	3-6
Gang	3-6
Slope Edit.....	3-8
Pattern Level Values	3-9
Directions.....	3-10
Timebase.....	3-11
Last Step.....	3-12
Polyrhythm Timebase.....	3-12
Bar Length	3-12
Bar Edit	3-13
Bar Copy	3-14
Bar Extend and Copy	3-14
Bar Values.....	3-15
Bar Loops.....	3-15
Inserting empty bars	3-16
More On Step Editing	3-16
Copy / Insert.....	3-16

Sculpting and Randomisation	3-19
Audition	3-19
Configuring the Aux Rows	3-20
P3 Edit Menu Options.....	3-22
First step.....	3-22
ReSync	3-23
Sync.....	3-23
Apply FTS.....	3-23
Aux Macro & Accumulator Config.....	3-23
PATTERN CLEAR Short-cut	3-23
Saving Patterns	3-24
Saving Patterns – to SD card / internal flash	3-25
 4. Using Scenes.....	4-1
Gbar	4-2
Length.....	4-2
Song Advance	4-2
Force To Scale.....	4-2
Xpose	4-4
Saving a SCENE.....	4-4
Mute Hold.....	4-5
The Scene List.....	4-7
Move SCENE.....	4-8
Scene Loops	4-8
Recall / Copy To Workscene	4-9
Deleting A Scene.....	4-9
Change View / Scene Select View	4-10
Editing a Saved Scene	4-12
Pattern “Saved” Status.....	4-13
 5. Creating Another Song	5-1
Song Copy.....	5-2
 6. Play Mode Menu	6-3
Scene Auto Save.....	6-3
Edit Track Thru Over-ride.....	6-3
Tune.....	6-3
MIDI Reset.....	6-3
Card / Sysex	6-4
More.....	6-4
User Configuration.....	6-6
Safe Stop.....	6-8

7.	MIDI configuration	7-1
	Port Configuration	7-1
	soft thru.....	7-2
	thru FTS	7-3
	record enable.....	7-3
	mclk send.....	7-3
	MIDI Clock Reception	7-4
	MTC – MIDI Time-code	7-4
	MIDI Options	7-5
	Remote Control MIDI Input	7-6
8.	CVIO and Drum Trig Configuration.....	8-1
	CVIO Config.....	8-1
	CV Output Settings	8-2
	Envelope Generation	8-5
	Gate Port Settings	8-8
	CV Input	8-9
	Drum Trig Config.....	8-11
9.	Track values	9-1
	Adding a track MIDI value.....	9-1
	Assigning Knobs A & B	9-6
	Saving Track Values.....	9-7
	Track Control Values	9-9
	Track Mixer Page.....	9-11
10.	CK patterns.....	10-1
	Drum Grid Edit.....	10-2
	Row View	10-5
	Quick Note Number Edits	10-8
	Bar Length	10-8
	Last Step.....	10-9
	Looping of Bars & Steps	10-10
	View and Signature Timebase	10-10
	Changing the Signature Timebase	10-12
	GANGing Notes in CK Edit.....	10-13
	GANGing by Step or Row	10-14
	Clearing the Ganged Selection	10-14
	Copying, Deleting and Inserting Notes	10-16
	Copying by Bar or Ganged Selection	10-18
	Cutting / Deleting Notes	10-18
	Ripple Edit.....	10-19

Fixing and Labelling Rows.....	10-20
Other Edit Menu Functions.....	10-21
Apply Quantisation.....	10-21
Delete Doubles	10-22
First Step and Pattern Rotate	10-22
Continuous Controller Editing.....	10-23
 11. CK Event List Edit	11-1
 12. Real- and Step-time record.....	12-1
CK Pattern Record	12-1
Quantised Recording	12-1
P3 Record	12-4
P3 Step-time record (Arp capture).....	12-5
 13. P3 Patterns & Aux events.....	13-1
Randomizing Pattern Values	13-1
How Events Work	13-3
Inter Track Events.....	13-6
A Quick Grab	13-7
Auxes With Double Values	13-7
Target Values For Grab, Swap and Push	13-9
Transpose By Events	13-10
Aux Notes	13-12
Chord Memory	13-13
Polyphonic Real-time Record	13-13
Accumulator Events.....	13-14
What Is An Accumulator?	13-14
Putting An Accumulator To Work.....	13-15
Relative Accumulation.....	13-17
Accumulator Configuration.....	13-18
Auxiliary Value Scaling – No More	13-23
The Accumulator Events	13-24
Accumulator Mask Events	13-25
Knob Mask Events.....	13-25
Knob Grab Events.....	13-26
Set Step Value Events.....	13-27
Repeat Events.....	13-27
Redirect Aux Events.....	13-29
Pattern Control.....	13-29
Global Control.....	13-29

14.	Loading and Saving Data.....	14-1
	Wipe Songs.....	14-2
	Card SAVE	14-2
	Card LOAD	14-7
	MIDI Sysex	14-10
	Factory Reset / Memory Initialisation	14-11
15.	Firmware update	15-1



Cirklon Sequencer front panel

Getting Started with Cirklon

To follow these instructions for the Cirklon sequencer, it will help to familiarise yourself with the layout of the front panel, and the names of the various controls.
Refer to the diagram on the previous page.

The horizontal row of keys along the lower part of the panel labelled with numbers 1 to 16 are the STEP or TRACK KEYS.

The knobs located directly above the step keys, are the STEP or TRACK ENCODERS.
In play mode, where they are used to alter track settings such as muting and un-muting tracks, they will be referred to as TRACK keys/encoders.
In pattern edit modes, where they are used to alter the settings for the steps of a pattern, they will be referred to as STEP keys/encoders.

The encoders rotate continuously and have no marker positions.
They are used to make relative changes to numeric values, and to scroll and select in lists.
All encoders have a centre-push action, so they can be pressed as well as turned.
In some cases, the encoder may also be held in while it is turned, to enable coarse adjustment of values in larger steps.

In contrast, the two KNOBs, labelled A & B, which are used for absolute control of values, have a 300-degree range, and have no press-switch.

The encoder to the immediate right of the display is the VALUE encoder.
The centre push switch of the VALUE encoder is the ENTER key.
This encoder is often turned to alter a numeric value, then pressed to confirm the change.
It is also used to confirm operations where it was not used to edit any value.

All other keys and encoders are labelled with their respective names, except for the transport keys, where the commonly used square, triangle and circle symbols denote the keys for STOP, RUN and RECORD.

1. The Cirklon Concept

Cirklon is a multi-track MIDI and CV/gate hardware sequencer.

From 2020, the original Cirklon model has been replaced by Cirklon 2.

Cirklon 2 adds a higher-power CPU, full colour touch-screen, more RAM memory and additional connectivity options.

The Cirklon 2 OS was developed directly from the existing Cirklon code-base, and development will continue in parallel on both machines, so the two versions are very similar in operation. This manual covers both versions, with extra features of Cirklon 2 described as needed.

Cirklon has a maximum of **64 TRACKS** (configurable in banks of 16), each of which can play a **PATTERN** of notes and/or controller messages on a connected **INSTRUMENT**.

Instruments can be connected to one of 5 independent MIDI ports, an optional multi-channel CV/gate port, one of 6 virtual ports on the class-compliant USB-MIDI device port, or to an optional drum-trigger interface connected to the SYNC port.

Cirklon 2 adds a USB host port, which may be used for the connection of a class-compliant USB-MIDI interface with up to 16 additional MIDI input and output ports

The USB host interface also supports USB storage devices or the connection of a USB keyboard for naming songs, patterns, etc.

Cirklon 2 also has a 100Base-T ethernet LAN port. Support for RTP-MIDI transfer is still under development and will appear in a future OS update.

All the MIDI ports can send MIDI tempo clock or timecode.

The SYNC port can be used to slave DIN SYNC equipped devices at 24 or 48 ppqn.

Cirklon has 2MB of battery-backed RAM, which holds all the song and instrument data, and configuration options.

Cirklon 2 increases the RAM capacity to 8MB.

The contents of RAM are retained while the power is off, so you are ready to return to the music you were working on the moment it is powered up.

Songs, instrument definitions and configuration settings can be saved to, and loaded from an internal micro-SD card, or a regular size SD or MMC card in the rear panel slot.

Cirklon 2 can additionally save and load data from USB storage devices.

The entire contents of memory can also be saved as a single .SYX file to internal or card storage, or backed up and restored via MIDI system exclusive dump.

All the patterns which make up one piece of music are grouped together in a **SONG**. Multiple songs can be loaded in Cirklon's memory, but only one can play at a time.

As well as the patterns, each song also has an instrument assignment for each track, and a set of **SCENES**.

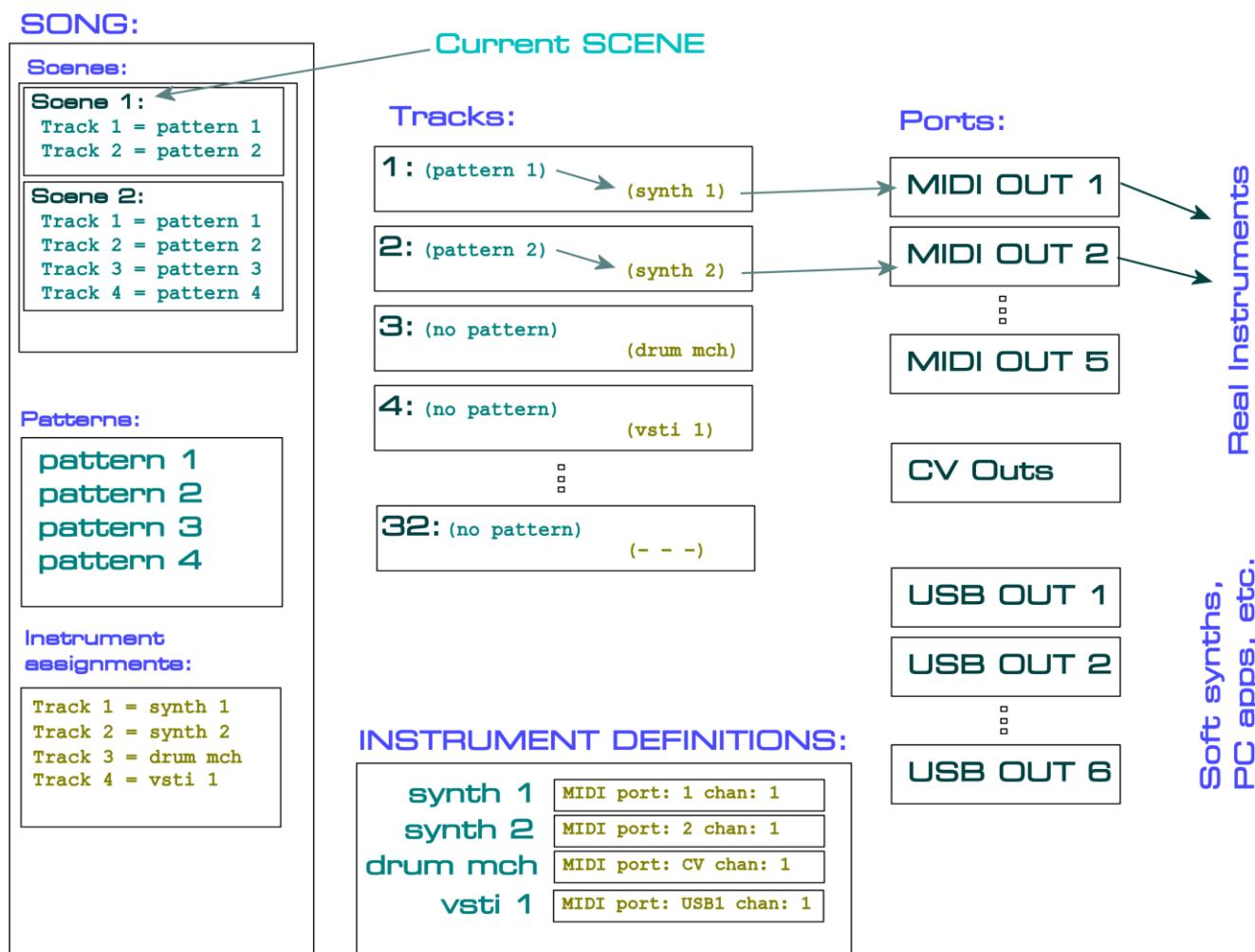
The instrument assignments determine which instrument will play the pattern on each track.

Scenes are used to recall the pattern selections for every track, along with the initial active/muted status, and some further optional values.

By creating a sequence of scenes in a song, a whole piece can be assembled from smaller sections, each composed of a number of patterns playing across the tracks.

The following figure shows a simplified example of what is going on inside Cirklon while playing a song.

After describing what is there, we'll go through the stages needed to get to this point.



The box on the left shows the contents of a simple song.

All the information needed to play a piece of music is contained in the current **song**, and the **instrument definitions**.

The **tracks** are part of the sequence engine, which controls pattern playback and generates all the outgoing MIDI events.

The **instrument definitions** describe the real instruments which Cirklon is to control, most importantly the **MIDI port** and **channel** to which the instrument is connected.

The instrument definitions also include some options to alter track behaviour, and can store user-defined note and controller labels to aid editing, and a set of ‘track values’ for real-time control. We will look at these advanced settings later.

Instrument definitions are *global* - the choice of which instrument to use on each track is made as part of each song, but every song loaded has access to the same set of instruments.

In this example, the song contains:

- 2 scenes *(scene 1 and scene 2)*
 - 4 patterns *(pattern 1, pattern 2, pattern 3 and pattern 4)*
 - 4 instrument assignments

When the song is loaded:

- the instrument assignments tell each track which instrument to use
 - *Scene 1* becomes the current scene, telling each track which pattern (if any) it should play

The tracks look at the instrument definitions to see which MIDI port and channel they should send note and controller messages to.

For this song, track 1 would use the settings held for *synth 1*, track 2 for *synth 2*, track 3 for *drum mch*, and track 4 for *vsti 1*.

Scene 1 tells track 1 to play *pattern 1*, and track 2 to play *pattern 2*.

Recalling scene 2 will assign patterns to tracks 3 and 4, which were not used during scene 1.

In scene 2, tracks 1 and 2 continue to play the same patterns they were assigned during scene 1.

Note that the scenes only tell the tracks which patterns to play – the patterns themselves are not part of the scene.

The same pattern can be used by many different scenes in a song.

Song Play Mode

Any of the scenes in a song can be manually recalled at any time.

To play through all the scenes of a song automatically, for a controlled song structure, Cirklon must be in **song play mode**.

In song play mode, each scene will play for a chosen number of bars, then advance to the next scene in the list.

Some scenes may be set to require 'manual advance', so you can have a song which plays automatically through some sections but repeats certain scenes indefinitely until you choose to proceed. You may also set a scene loop, which will cause playback to loop round a number of consecutive scenes in the song.

By default, Cirklon will enter song play mode whenever you are on the SONG page.

You can change this behaviour by holding the SONG key and pressing the RUN key.

This combination will not start the sequencer running, it only changes the song play setting.

There are three choices:

<i>auto</i>	scenes advance only while on the SONG page
<i>song</i>	scenes always advance, except in pattern edit
<i>work</i>	scenes must be recalled or advanced manually

To get to the stage of having a first song play on a new Cirklon, we'll need to cover the following tasks:

- creating and assigning some instruments
- creating and assigning some patterns
- saving the pattern assignments as a scene

So, let's get started...

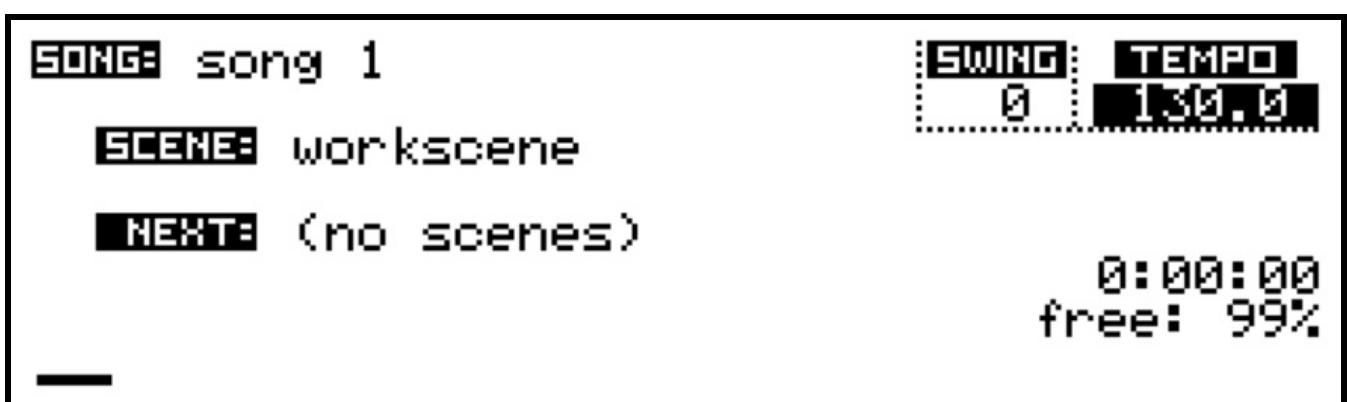
2. The Main Pages

There are four main pages which can be viewed on the display.
These are:

- SONG
- SCENE
- TRACK
- PATTERN

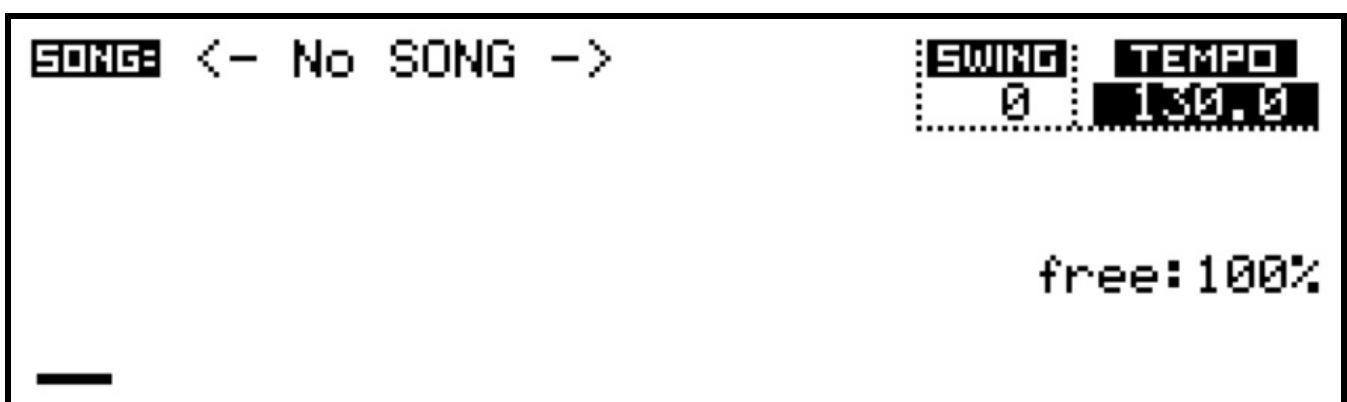
These pages are selected using four dedicated keys on the left of the panel.

When you first power on a new Cirklon, you should see the SONG page, like this:



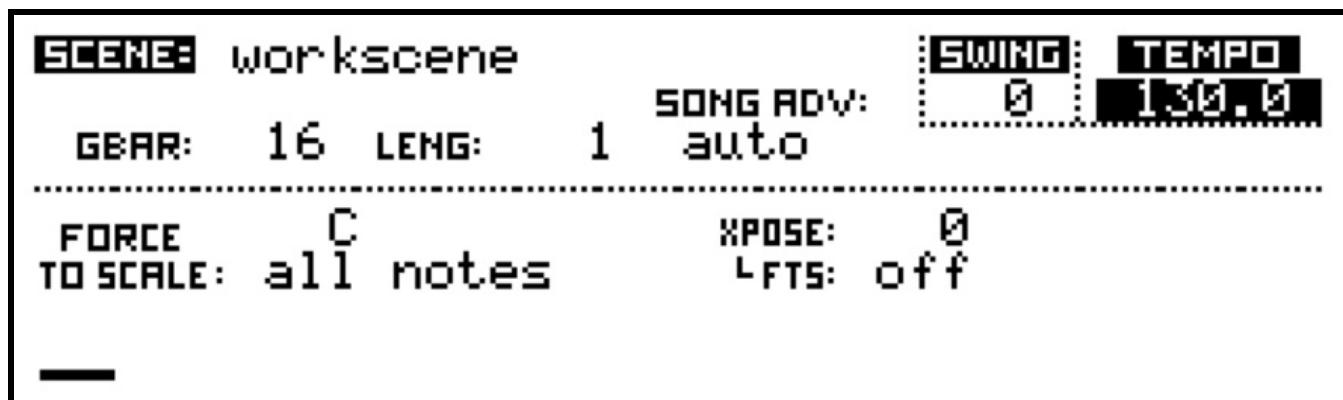
The LED in the SONG key will be lit to show you are on the SONG page.

Note: If Cirklon's memory has been wiped and there are no songs present, the SONG page may look like this:



In this case, you will need to create a new song. Hold SHIFT and press SONG, then press encoder 6, below the text "create new", which should have appeared on the display.

Press the SCENE key, and you will switch to the SCENE page, which should look like this:

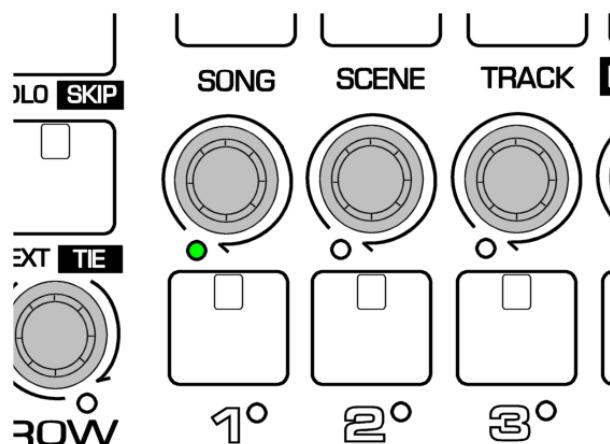


Pressing the TRACK key will switch to the TRACK page, which should look like this:



The TRACK page shows information for the **current edit track** – the single track that is active for pattern selection, editing and real-time recording.

The lower left of the display above shows the current edit track is number 1. This will also be shown by a green LED, just below and left of track encoder 1:



Note: Where they are used on the SONG, SCENE or TRACK pages, the 16 keys and encoders will be referred to as the track keys/encoders, since they each control one of 16 tracks.

In pattern edit, they will be referred to as the step keys/encoders, since they edit the step values of the pattern.

The edit track selection can be changed on the SONG, SCENE or TRACK pages by pressing the track encoders, or by turning the BAR encoder.

Using more than 16 tracks

The total number of tracks available is user-configurable to 16, 32, 48, or 64.

To change the number of tracks, see section 6 – Play Mode Menu, User Configuration.

Since there are only 16 track keys and encoders, if the number of tracks is set to 32 or more, the tracks are mapped to the track keys and encoders in banks of 16.

You can step through the banks by pressing the BAR encoder.

A message on the display will show which tracks are selected.

Note that this may result in the edit track selection being outside the current track bank selection, in which case none of the track encoder LEDs will be lit.

You may also change bank along with the edit track selection by turning the BAR encoder.

The single green track encoder LED will scroll across the tracks and continue into the next bank as you go past the end.

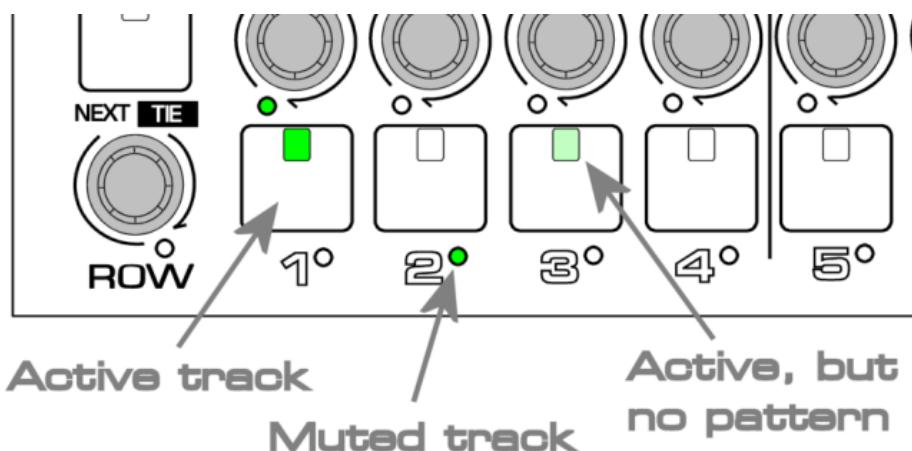
Track Muting and Status

The track keys are used to control the active/muted status of each track.

A pattern assigned to a track will not play unless the track is active, with the key LED on.

If a pattern is assigned to a track which has been muted, the LED below the track key will light green to show the track is in use.

If a track has been set active, but no pattern is assigned, nothing will play, so the key LED will be lit at half brightness.



It is possible to “arm” a muted track, so that it will become active at the next global bar loop. Do this by holding the SHIFT key as you press the track key for the track to be unmuted. The LED below the track key will be amber until the track becomes active.

The SHIFT + <track key> function will not mute an active track unless a user configuration option for this is enabled – see the later section on user config.

Solo mode

Solo mode adds an extra layer of track mute control, similar to the solo function on a mixing desk. It allows you temporarily listen to a small selection of active tracks, without affecting the mute status of the other tracks.

Each track has a solo status, which over-rides the mute status when solo mode is active - only tracks set to solo will play, all other tracks will be muted.

To enable solo mode, hold the SOLO key, and press the track key of one or more active tracks. The SOLO key and the track keys for the selected tracks will start to flash. Although other tracks remain active, only the flashing, soloed tracks will play. While solo mode is active, pressing the track key for an active track will mute the track.

Pressing the track key for a muted track, will both un-mute, and enable solo play for that track.

To exit solo mode, simply press and release the SOLO key by itself.

When you exit solo mode, the tracks will retain their active solo status, so another press of the SOLO key will return you to the same state.

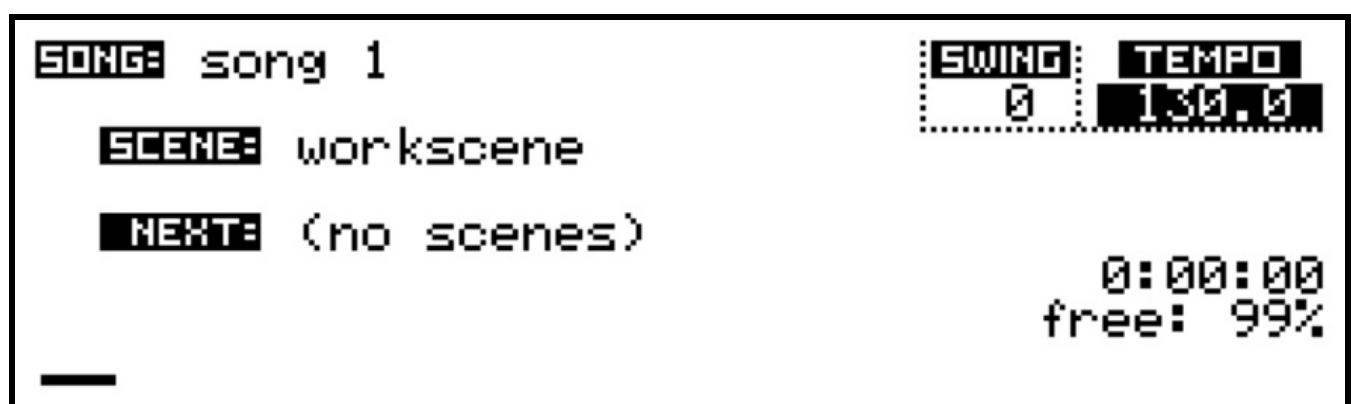
If you want to exit solo mode and clear the solo status for all tracks, hold the SHIFT key and press SOLO.

Changing Song Values

Changes to the main values on each of the play pages are made using the VALUE encoder.

The selected value is shown with an inverse or highlighted background.

Go back to look at the SONG page:



There are 3 values which can be adjusted here:

- the TEMPO
- the SWING setting
- the song name

The display above shows the selection highlight over the TEMPO value, but it is not yet active for editing. Press the VALUE encoder (ENTER), and arrow symbols will appear around the TEMPO value:



You will also see that the selection highlight no longer covers the fractional part of the tempo. By default, tempo is adjusted in steps of 1 BPM. Turn the VALUE encoder and you should see the value change.

It is possible to adjust the tempo in coarser or finer steps.

Hold the SHIFT key and turn the VALUE encoder one step to the left, then release SHIFT.

The highlight should now look like this:



Adjustment of tempo will be in steps of 10 BPM.

Now hold SHIFT and turn VALUE back two steps to the right.

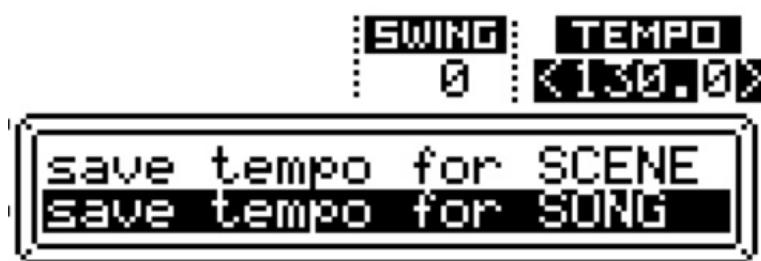
The highlight will now cover all the digits, and you will find tempo will change by steps of 0.1 BPM.

It is also possible to set the tempo using the SCULPT key.

Tap the SCULPT key at least twice, at the desired tempo. The display will show the tempo detected.

While the TEMPO value is selected for editing, it is possible to save the value for the current song or scene by pressing the SAVE key.

A popup window will show two options, with song selected by default:



To save the current tempo value for this song, press ENTER.

When a song has a stored tempo, the label [SONG] will appear below the value.

You can remove the stored value by pressing the DELETE key while the tempo value is active for edit. When the song is later recalled, the tempo will be adjusted to its stored tempo value.

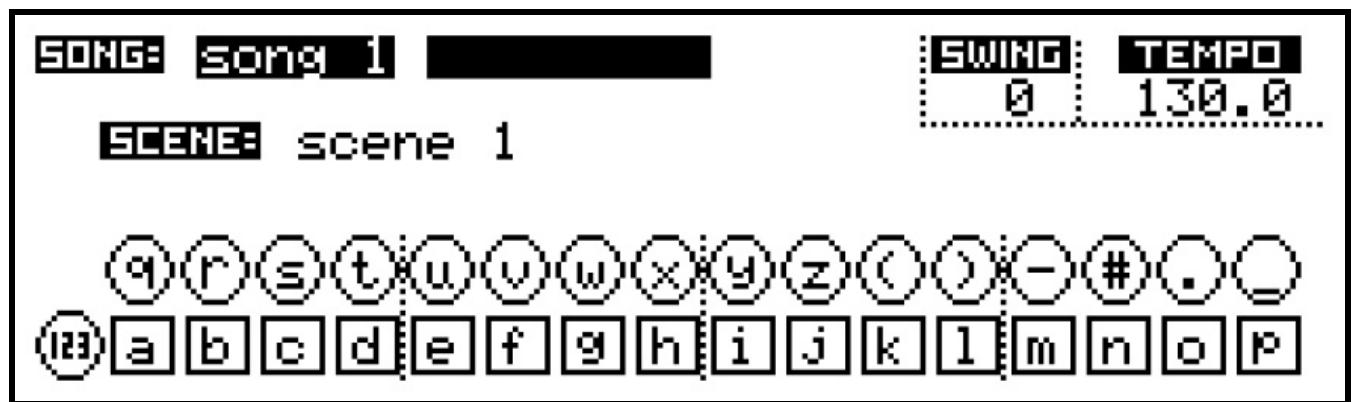
Once the tempo has been set as required, press VALUE/ENTER again.

The arrows will disappear, and the encoder is once more active for selecting the other values on the page.

The SWING setting is adjusted in the same way as tempo, and can be saved for the song or scene.

Swing has a range of 0 (off) to 16, and adds a 16th note shuffle feel to the playback of patterns.

If you select the song name, then press VALUE/ENTER, you will see a display like this:



The flashing cursor at the end of the song name, and the visual representation of the step keys and encoders at the bottom of the display show you are in **text input mode**.

All track key, encoder and lower LEDs also light up in this mode.

In text input mode, you can enter or edit the name using the keys and encoders.
The character on each key is shown in the visual aid.

Press DELETE to delete the character before the cursor.

Hold the SHIFT key to enter upper case letters.

Pressing the ROW encoder toggles between letters and numbers – the visual aid will switch accordingly.

You can move the cursor within the text by turning the VALUE encoder.

When the cursor is within existing text, the INSERT key will switch between new characters being inserted into, or over-writing the text.

To confirm the edited text, press ENTER (the VALUE encoder).

To exit without affecting the stored name, press the MENU key.

This mode of text entry is used for naming various objects, such as songs, scenes, patterns and instruments.

Auto Name Generation

If you wish, Cirklon can invent names for your patterns, songs or scenes as follows: from within the name field, simply hit the SOLO button. A name will be automatically generated. Keep hitting SOLO until you see something you like, then save as if you'd typed it yourself.

Creating and Assigning an Instrument

Let's return to the task of creating and assigning an instrument.

This is done from the TRACK page. If not already selected, press TRACK to get there.



There are four values shown on the main TRACK page:

- the pattern selection, labelled **PATTERN**:
- the fill-in pattern selection, labelled **FILL**:
- the track number (or name), labelled **TRACK**:
- the current instrument assignment, labelled **INSTRUMENT**:

Remember the values shown here are for the current edit track, in this example track 1.

One of the values will be highlighted to show it is active for editing.

As on the song page, to move the selection highlight turn the VALUE encoder.

Turn the VALUE encoder clockwise, until the instrument field is highlighted.

The instrument field will show “---” where no instrument is assigned.

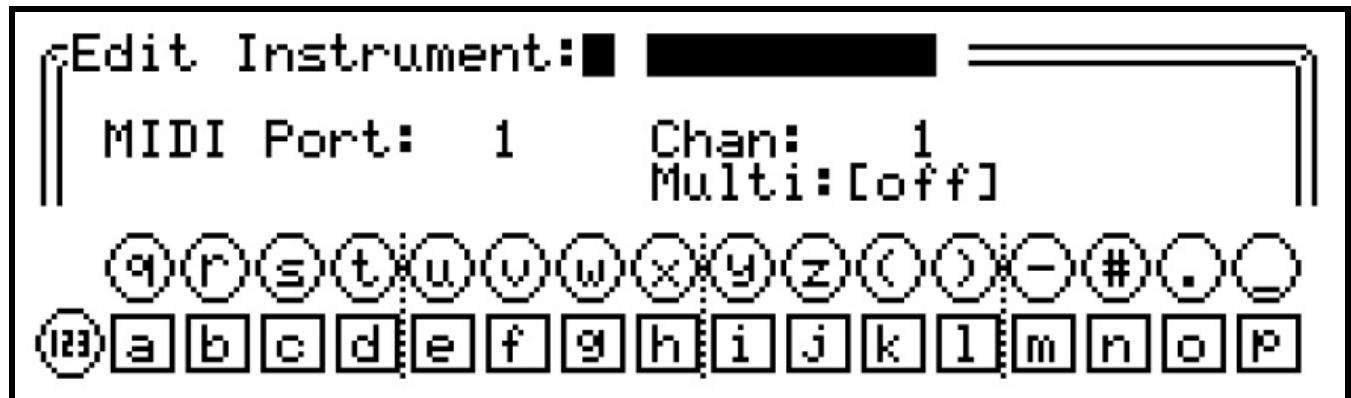
Then press VALUE/ENTER to access the **instrument list**:



The instrument list will be empty on a new machine, so the only choices are to assign “no instrument” to the track, or to create a new instrument.

“create new...” will be selected by default, so press ENTER to define the first instrument.

The instrument definition page will first appear like this:



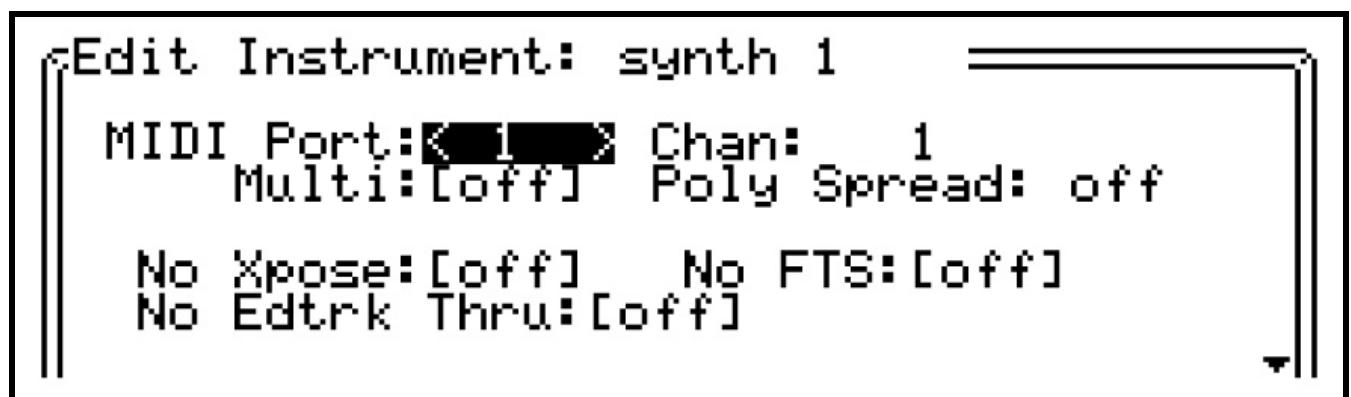
The field on the top line of the display is active in text input mode, waiting for the name of the new instrument. Using the track keys and encoders, enter an instrument name in the same way as for the song name.

Once a name has been entered, the rest of the instrument definition values will be accessible. Use the VALUE encoder to move the highlight selection between the fields, then press to edit the values as before.

Arrows will appear around a numeric value to show it is active for editing.

When highlighted, the values between square brackets will toggle on each press of ENTER.

In this view, the MIDI port value is active:



The MIDI port should match the port the instrument is connected to. This can be one of the 5 standard MIDI ports, one of 6 ‘virtual’ ports on the USB MIDI device interface, or the CV port, which is used to route note and controller output to the optional CVIO expansion for CV/gate output.

On Cirklon 2, available USB host MIDI ports will appear in the form “**Hstnn**”, where *nn* is the port number.

The **Multi** option is intended for use with multi-timbral synthesizers which may respond to MIDI messages on several channels at the same time. When multi is enabled, an extra field to specify the MIDI channel number will appear alongside the instrument assignment on the track page. This saves any need to redefine the same instrument for each channel it is active on.

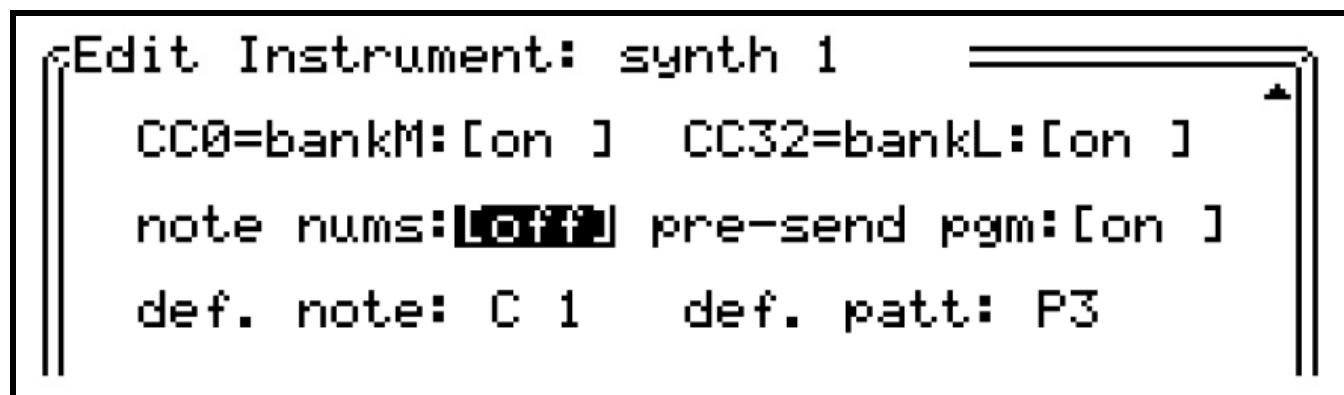
The **Poly Spread** option allows a number of monophonic sounds, assigned on consecutive MIDI channels, to be treated as a single polyphonic instrument.

More details of this function are below.

The **No Xpose** and **No FTS** options globally disable the effect of any pattern transposition or force-to-scale for the instrument. This is most used with drum machines or percussion modules, where specific sounds are mapped to each note number.

The **No Edtrk Thru** option disables automatic soft-thru of incoming MIDI notes to the instrument, even if it is assigned to a track selected as the current edit track.

Two further instrument values can be found by scrolling the value selection further, indicated by the down-arrow at the bottom right:



The **CC0=bankM** and **CC32=bankL** options enable the use of CC numbers 0 and 32 for the bank select MSB and LSB messages which some instruments use in conjunction with MIDI program change messages to select from more than 128 internal sounds.

When either bank select CC is enabled and a value is active, it will be transmitted and stored along with the program change message, as described in the later chapter on Track Values.

Note numbers might be preferable when dealing with some types of instrument. Thus, setting **note nums** to **on** tells Cirklon to display numbers rather than notes in the pattern editors.

Some pattern-based sequencers which can respond to program change messages to select other patterns or scenes may take a short time to respond to those messages.

This can mean that program changes recalled with an advancing Cirklon scene don't take effect until 1 bar later.

By setting the **pre-send pgm** value to **on**, any stored program changes for the instrument will be sent well in advance of the next Scene change.

Users of Elektron boxes, Korg Electribes or other groove-type-things, might find this very welcome.

Default Note – the note value that should be set up in empty patterns ready for you to start working. This can be especially useful if an instrument is a single drum voice as all you need to do to get started is activate a few steps.

If, instead, you set this to **off**, the default note will match that of the current scene's Root note (which is needed for Force-to-scale operations).

Default Pattern – choose between two types: Step-time (P3) or Real-time, polyphonic (Ck), or have a prompt to select pattern type appear with each new pattern.

Once the correct MIDI port and channel have been set, press the SAVE key to store the new instrument. This will return you to the TRACK page, with the newly created instrument assigned to the current track:



If you wish to create and assign instruments for other tracks, press another track encoder. The display will switch to the new track, with the instrument assignment field still active for selection.

Press ENTER to access the instrument selection for the new track. Then choose the "create new..." option once more.

If you wish to edit the settings for an existing instrument, select the track with the instrument already assigned, or assign it to the current track.

Press the ENTER key to access the instrument list, then hold the SHIFT key while pressing ENTER again. This will open the instrument definition page for the stored instrument, where you can change any of its values.

Press SAVE to confirm the changes, or MENU to cancel the changes and restore the previous settings.

For the example song, we would continue to create the instruments *synth 2*, *drum mch* and *vsti 1*.

As each new instrument is defined, they will appear in the instrument list, along with their port and channel values, like this:



Deleting An Instrument

If an instrument definition is no longer required, you can delete it by highlighting the instrument in the selection list, then pressing the DELETE key.

A two-step confirmation page will appear to make sure you don't do this by accident.

Also, if the instrument is used by any loaded songs, you will be prompted to choose an alternative instrument with which to replace the existing assignments.

Poly Spread

The poly spread function takes a polyphonic stream of notes, either from MIDI soft-thru or the output of a pattern, and spreads them as single notes across a range of MIDI channels, similar to the way voices are allocated in a polyphonic synthesizer.

This allows, for example, a polyphonic pattern of chords to be played on a number of

monophonic synths set up with a similar (or entirely different!) patch, on the same MIDI port.

The poly spread value can range from 2 to 16 channels.

The channels are assigned from the base channel number up.

For example, an instrument set with a base channel number 4, and poly spread of 4, will play up to 4 simultaneous notes, which will be sent on channels 4, 5, 6 and 7.

There is a choice of five assignment modes to control how the notes are allocated across the channels. The mode can be changed using MIDI CC number 110.

The five modes correspond to these CC values:

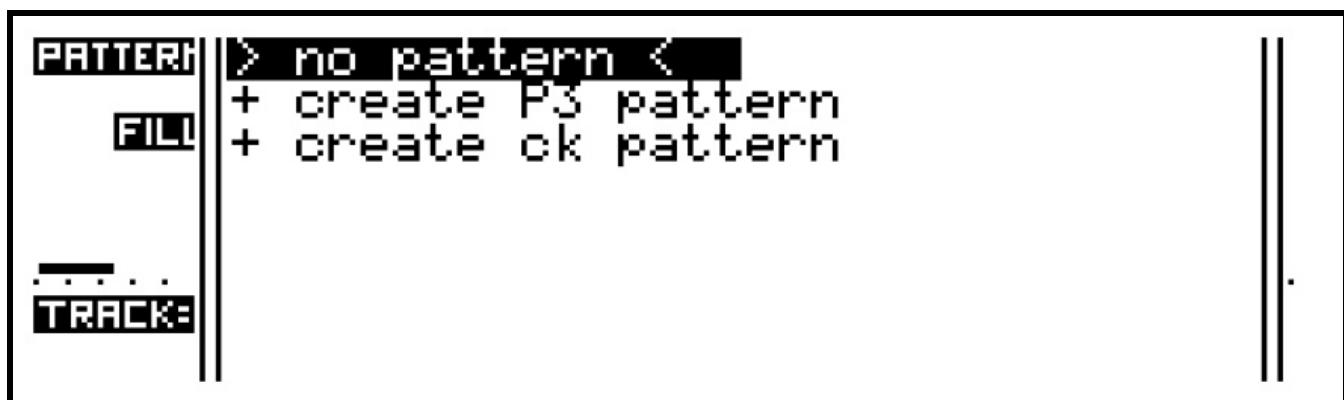
0	default	each new note is assigned to the least recently used channel
1	reset order	least-recently-used order is reset for each new group of notes
2	reassign same note	a new note will be reassigned to any free channel it was previously assigned to
3	reset & reassign	a combination of options 1 & 2
4	dynamic unison	all available channels are played in unison for single notes, with the channels/voices divided among further notes to maximise the use of available voices

Creating and Assigning a Pattern

With the instruments created and assigned to tracks, we can create some patterns. On the TRACK page, use the VALUE encoder to highlight the PATTERN: field.

As with the instrument assignment, pressing ENTER on the pattern selection will bring up the **pattern list**.

In a newly created song, or on a track which has not yet had a pattern created for it, this list will be empty, except for the options shown:



There are two different types of pattern which can be created.

The first is the P3 pattern, which is an enhanced version of the patterns found on the Sequentix P3 Sequencer.

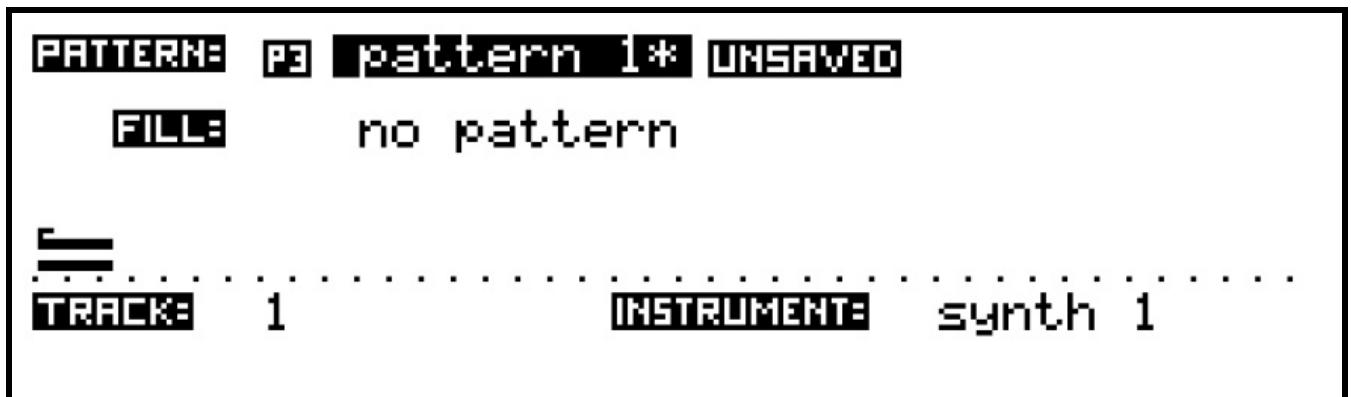
The P3 was intended to allow pattern creation in a similar fashion to analogue step sequencers, with added means for patterns to modulate and interact in ways only possible with a processor-based machine.

The second type is the CK pattern, which is an event-list pattern better suited to polyphonic keyboard and percussion parts, and un-quantised real-time recording.

These will be described in depth later.

Normally the pattern list will show only the patterns that were created on the current track. You can expand the list to include all patterns in the song by holding SHIFT as you press ENTER to bring up the selection list. Additionally, if you press the INSERT key while the list is visible, a LOAD screen appears. This provides a means of inserting patterns stored on your SD card, perhaps patterns created in another song (see also: Saving Patterns – to SD card a little later)

To create the first pattern, scroll to “+ create P3 pattern” and press ENTER. The display will return to the track page, with the newly created P3 pattern assigned to the track:



The pattern field now shows a symbol for the pattern type (P3), and its default name - “pattern 1”.

You will also see an asterisk following the pattern name, and a further symbol indicating the pattern status is unsaved.

If a pattern is used in a number of different scenes in a song, making changes to the pattern could cause unexpected changes elsewhere in your song.

To prevent this, a pattern may be marked as saved. This can be done manually in pattern edit, or it will happen automatically when a scene is saved which uses the pattern.

When you edit a saved pattern, an edit copy of the pattern is made.

This means you can easily undo any changes since the point of saving by recalling the saved version. Or you can save the edited pattern as a new version, leaving the original pattern as it was before editing.

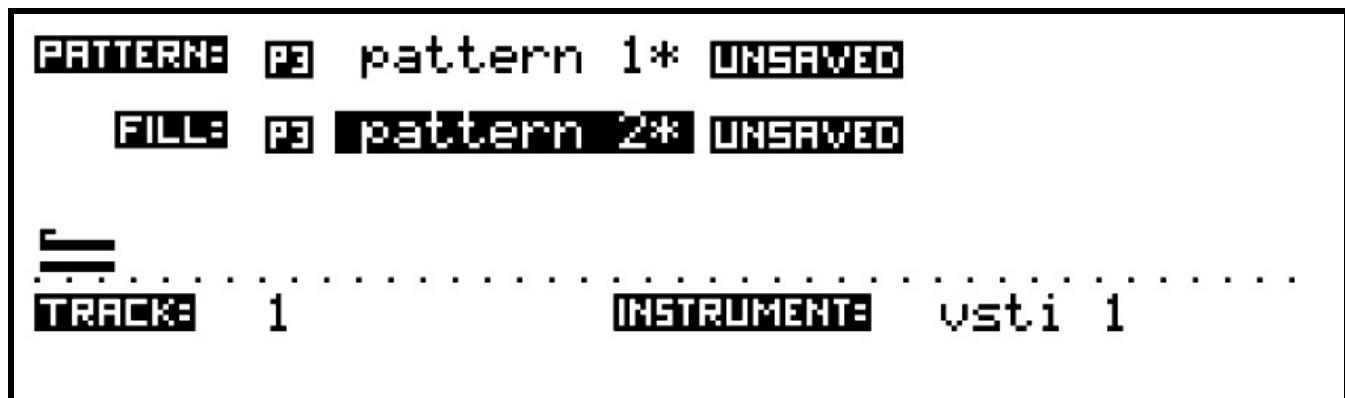
The asterisk after the pattern name shows that there are unsaved changes to the pattern, either because it is newly created, or because edits have been made since it was last saved. The saved status of each pattern can also be seen in the pattern selection list.

If you make changes to any saved patterns and then save the scene, you will be prompted to choose whether you want to save the changes to the pattern, save-as a new version of the pattern, skip saving to leave the edit copy in place, or lose the edits.

As well as protecting against unexpected changes to a song, this behaviour also simplifies the creation of incrementally changing versions of a pattern over the course of a number of scenes.

Fill Patterns

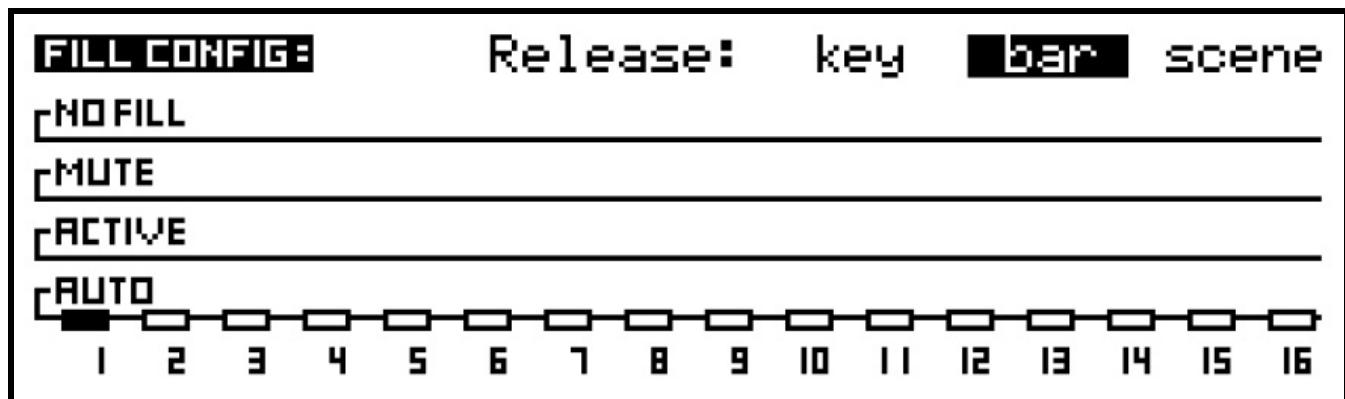
In addition to the main pattern, you may optionally assign a fill pattern to each track. Fill patterns are selected in the same way as the main pattern, from the same pattern list, by selecting the FILL: field.



Any track assigned a fill pattern will switch to playing that pattern during a fill-in.

A fill-in can be manually triggered while in play mode using the FILL key. Once triggered, the FILL key will light until the fill-in completes, or turn off immediately if there are no tracks with a fill pattern assigned. Fills will trigger continuously if you hold the FILL key.

There are options to control how each track behaves during a fill-in, and when the fill-in should finish. These options are set on the fill config page, which is accessed by holding the SHIFT key then pressing FILL. The fill config page looks like this:



The track numbers for each track in the current bank are shown along the bottom of the display, with a block above.

Tracks which have a fill pattern assigned will have a filled-in block, as shown here for track 1. The vertical position of each block on the four lines, labelled *auto*, *active*, *mute* or *no fill* shows the **fill mode** for that track. Change the mode by turning the encoder above each track key.

The behaviour for each mode is as follows:

- auto* - the fill pattern will play if the track is active (un-muted)
- active* - the fill pattern will always play
- mute* - the track will be muted during a fill-in
- no fill* - the track will ignore the fill-in

At the top right of the fill config page, the fill **release** setting is shown.

This determines when the tracks will return to their main pattern after a fill has been triggered, as follows:

- key* - fill patterns play only while the FILL key is held
- bar* - fill patterns play until the end of the next global bar
- scene* - fill patterns play for the rest of the current scene

Change the release option using the VALUE encoder.

Note that in pattern edit, the FILL key is assigned a different purpose, you can only trigger a fill-in while on the SONG, SCENE or TRACK pages.

Pattern edit and real-time record will act on the fill pattern if the FILL field is selected on the TRACK page when you enter pattern edit or record modes.

The fill pattern on the current edit track will play continually while you are editing or recording it if the user config option “play fills in edit/record” is enabled – see the section on “Play Mode Menu” for details of setting user config options.

Track Swap

It is possible to swap the position of any track with another, perhaps to group related instruments together or even if you just happen to enjoy moving things around. The swap process is very simple and performed from the TRACK page as follows:

Ensure the track you wish to move is selected, just as if you were about to edit its pattern or Track Values.

Perform the swap by holding the COPY key and pressing another track encoder at the same time – the instruments, pattern assignments, and any stored track values for the two tracks will be swapped.

Note that this operation will not alter track numbers used for inter-track events in P3 patterns - those must be manually changed if you are using those events.

3. Pattern Edit

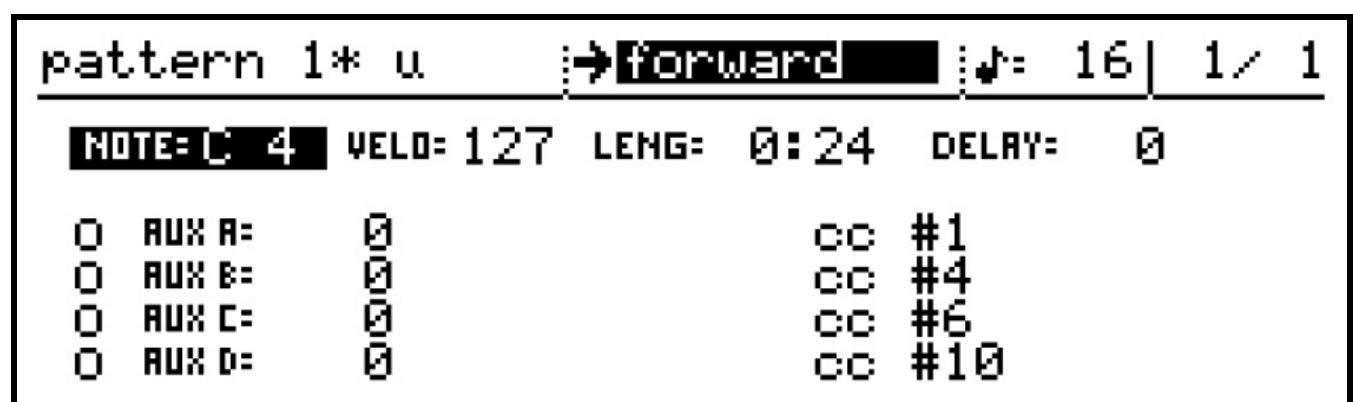
With an empty pattern created and assigned to the track, we need to edit the pattern to get some notes playing. Switch to pattern edit by pressing the PATTERN key.

Normally the pattern assigned to the current edit track will be edited.

However if the FILL field is selected on the track page, the fill pattern assigned to the edit track will be edited instead.

As a shortcut to create a pattern, if you press the PATTERN key on a track with no pattern assigned, a new P3 pattern will be created and assigned as you enter pattern edit.

At first, the edit page for a new P3 pattern will look like this:



The top line of this view shows:

- the pattern name
- the current direction
- the timebase setting (duration of each step as a division of the master tempo)
- the current bar number and bar length of the pattern

The lower part of the display shows all the current values for a single step of the pattern. This is the **step view**.

What's in a P3 Pattern ?

Each bar of a P3 pattern can be thought of as a matrix of values, as shown below. There are 16 steps in each bar, shown as the columns of the matrix.

For each step there are a number of different values.

There are numeric values for **note**, **velocity**, **length**, **delay**, and four **auxes**, labelled **A** to **D**.

There are status or flag (on/off) values for **gate**, **tie**, **skip**, **X** and the four **auxes**.

These values make up the rows of the matrix:

ROWS	STEPs															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
note																
velo																
length																
delay																
aux A																
aux B																
aux C																
aux D																
gate / tie / skip / X																
aux status * 4																

The purpose and range of each of the row values is as follows:

row	purpose	range
note	Sets the MIDI note to be played	C0 to G#10 (MIDI notes 0 to 127)
velo	Sets the velocity value for the note message	1 to 127
length	Sets the duration of the note. value is shown in the form <i>s:tt</i> <i>s</i> shows a whole number of steps, <i>tt</i> shows 48 th divisions of a step	0 to 47 ticks, to max 96:0 or "hold"
delay	Allows the note start to be delayed by a number of clock ticks (at 192 ppqn)	0 to 47
aux A – aux D value	Set the values for assigned MIDI controller messages, or aux events	MIDI CCs : 0 to 127 Aux event ranges vary
aux A – aux D status	Trigger the associated CC or event	on / off
gate	Trigger the note	on /off
tie	Force the current note to sustain until just after the start of the next step	on /off
skip	Skip over this step to the next un-skipped step	on / off
X	transpose defeat – prevent transpose of note value by scene xpos, FTS	on / off

Note that the length value can be longer than a single step period.

Assuming that the tie flag is not active, a note will continue to play for its full length,
overlapping any other notes in following steps.

This means the P3 pattern type can play overlapping, sustaining notes – a feature not
available on the P3 itself.

If the same note is triggered again while still playing, it will be released then re-triggered with
the new step length.

Editing Pattern Values

In pattern edit the 16 step encoders are used to change the numeric values.

The 16 step keys are used to toggle the flag values on and off.

The LEDs in the step keys indicate the current flag status.

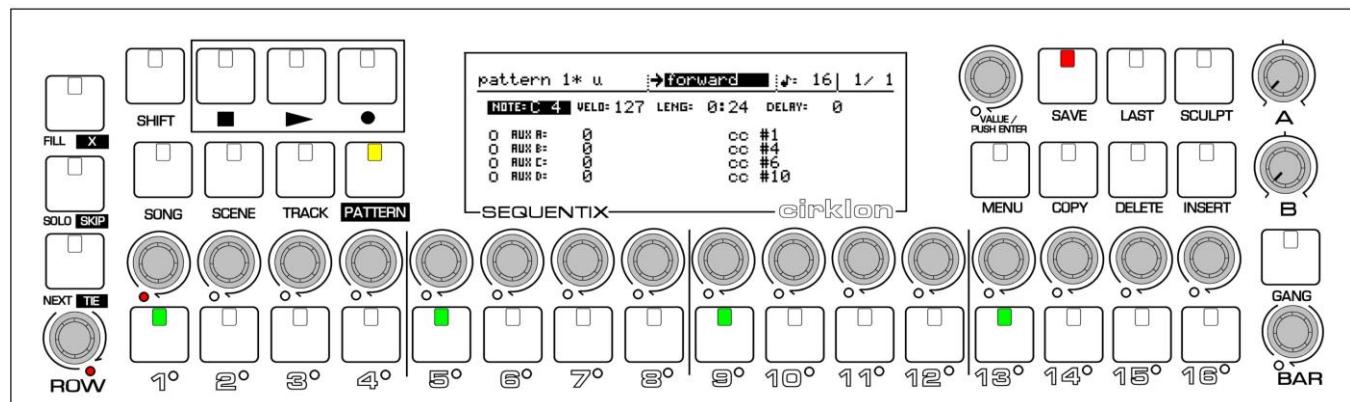
In a newly created pattern, the step keys will be assigned to the gate flags, and the encoders to
the note values.

Turn on some of the steps by pressing the step keys.

Then press the RUN key to start Cirklon running.

If you have created and assigned your instrument correctly (remembering to connect an instrument to the correct MIDI port and channel), it should start playing notes on the active steps.

For example, if you press steps 1, 5, 9 and 13, the front panel should look like this:



You can then turn some of the step encoders, to adjust the note values.

As the encoder above a step is turned to edit the value on that step, the display position will change to show all the values for the edited step, with the newly changed value updated.

A red LED below one step encoder indicates the current step.

You can switch to view the other steps without editing their values by pressing the step encoders.

To select which row value you wish to edit, turn the ROW encoder.

In step view, you will see this moves the selection highlight through the different values.

Normally the step keys are assigned to the gate flags.

These control whether a note is triggered on each step.

If you select one of the four auxiliary rows, the step keys will be assigned to the status flags for that aux row.

These control whether a CC message will be sent, or an aux event triggered on each step.

The selection of gate and the four auxiliary flags is therefore linked to the row selection.

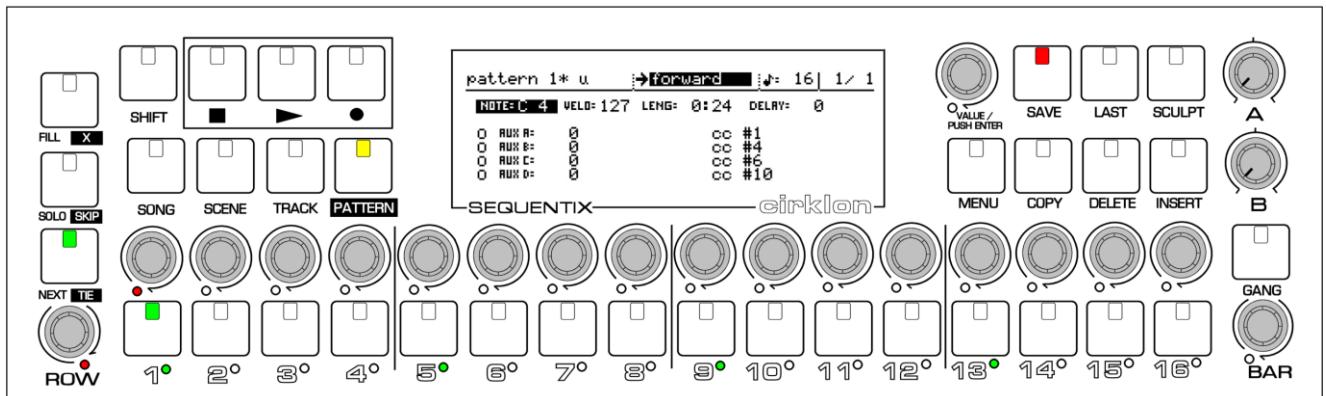
To access the remaining three flag values – tie, skip and X, there are dedicated keys on the far left of the front panel.

When one of these keys is pressed, its LED will light.

The LEDs in the step keys will now show the status for the selected flag.

The current gate status will appear on the LEDs just below the step keys, so you can still see which steps are active as you decide where to apply ties, skips and X's.

In the example above, if you press the TIE key, then set the tie flag on step 1 only, the panel should look like this:



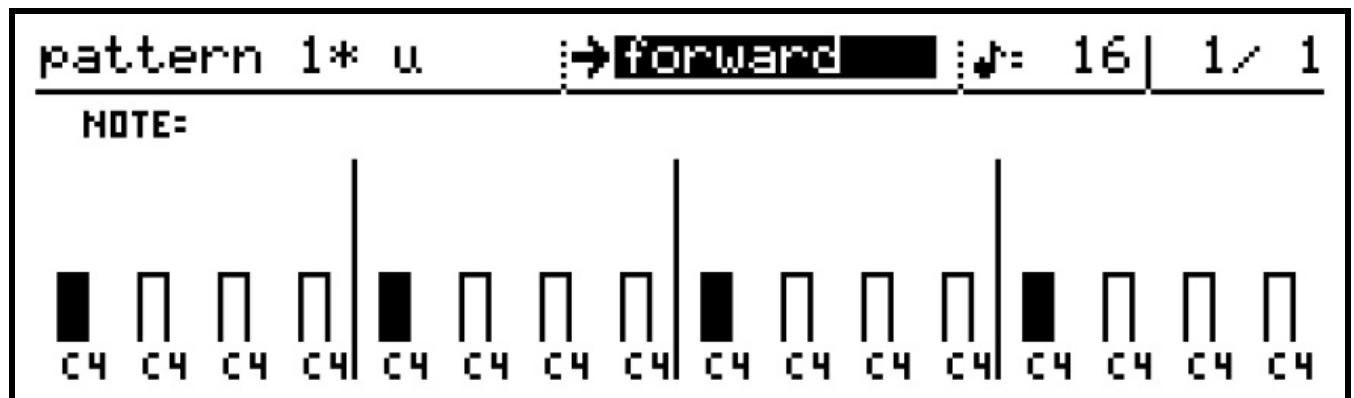
The TIE key is lit to show the step keys are assigned to the tie flags.

The LEDs under the steps show the gate flags.

Step 1 is lit to show tie is active on that step only.

While the step view shows every value for single step, the **row view** shows the current row values for every step in the current bar.

Press the ROW encoder, or the PATTERN key, to switch to the row view:



Pressing ROW once more will switch back to the step view.

In the row view, the 16 bars for note, velocity, length or delay will be filled in if the gate flag for the step is active – as can be seen for the active steps 1, 4, 9 and 13.

On the aux rows, the bars will be filled on each step that the aux is enabled.

The appearance of the bars is slightly altered for the length values.

Since the length value can cover a wide range, from fractions of a step up to whole numbers, the width of the bars is used to indicate whether the length is one or more steps long.

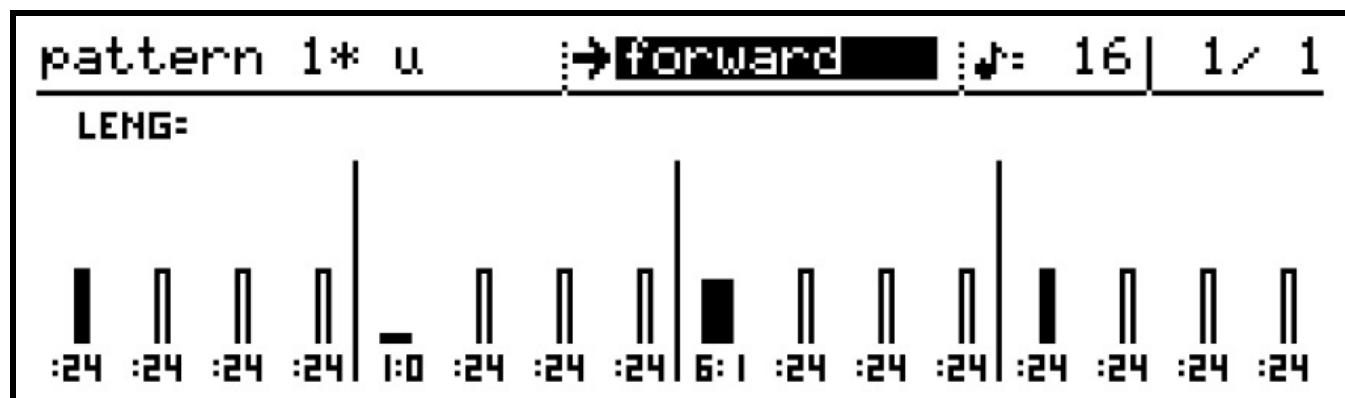
The number indicator also changes range to suit.

In this example, step 1 has a length of 24/48ths of a step, shown as :24

Step 5 has a length of 1 whole step, shown as 1:0

Step 9 has a length of 6 and 1/4 steps, shown as 6:1 – once the length is beyond one step, it increments in quarters, rather than 48ths of a step.

The bars for a length of less than 1 step are narrower than the normal bars.



In addition to the normal range of length values, there is also a special “hold” setting.

This means that the note should be held until the start of the next active step in the pattern.

Set the length on a step to hold by holding SHIFT as you turn an encoder to edit the length value. The value will appear as “hold” in step view, and a full size bar with a value of :00 in the row view.

Turn the step encoder without the shift key to return the step to the last numeric length value.

Editing Multiple Steps

Pattern edit has a couple of labour-saving features for times when you want to change the values on a number of steps by the same amount, or want to create a rising or falling ramp in consecutive values, but don’t want to go through every step making individual adjustments. These are the **gang** and **slope** edit functions.

Gang

Gang mode is enabled in pattern edit using the GANG key, just above the BAR encoder.

The gang function allows you to link a number of pattern steps together, so that any adjustment made using the encoder for one of the ganged steps will affect all ganged steps equally.

To enable gang mode for the first time, hold down the GANG key, then press some of the step encoders in turn, though don't hold more than one track encoder together – that has a different effect.

As you press each encoder, the LED below it should light green.

This shows the step has joined the gang.

After selecting a number of steps this way, release the GANG key.

The LED in the GANG key should remain lit, as will the green encoder LEDs for the ganged steps.

Now edit the value on any of the ganged steps using its encoder.

The values of **all** the ganged steps will change by the same relative amount.

The effect is easiest to see in the row view.

Note that gang only applies to numeric values – not to the on/off flag values on the step keys. Any steps which are not ganged can be still be edited as normal, without affecting any other steps.

To turn off gang mode, press the GANG key again.

The GANG LED will go off, as will the green step encoder LEDs, restoring the single red LED on the current step.

There are various ways to change the gang selection:

- to toggle steps in and out of the gang, hold GANG while pressing the step encoders
- to remove all the steps from the gang, hold the DELETE key and press GANG
- to invert the gang, i.e. to toggle the gang state of every step, hold SHIFT and press GANG

A further option allows you to quickly gang a repeating pattern of steps.

If you hold the GANG key, then hold one step encoder and press another step encoder so both encoders are held at the same time, those two steps will be added to the gang, along with all the other steps in the bar at multiples of the spacing between the two selected.

For example, to gang every even numbered step:

- hold GANG
- hold step encoder 16
- press step encoder 14

Steps 14 and 16 will be ganged.

The gap between steps 14 and 16 is 2 steps, so every other step a multiple of 2 steps from steps 14 and 16 will also be ganged.

In this case, that means every even step.

Try it with other patterns.

Note that holding GANG and pressing two adjacent steps will gang every step.

Slope Edit

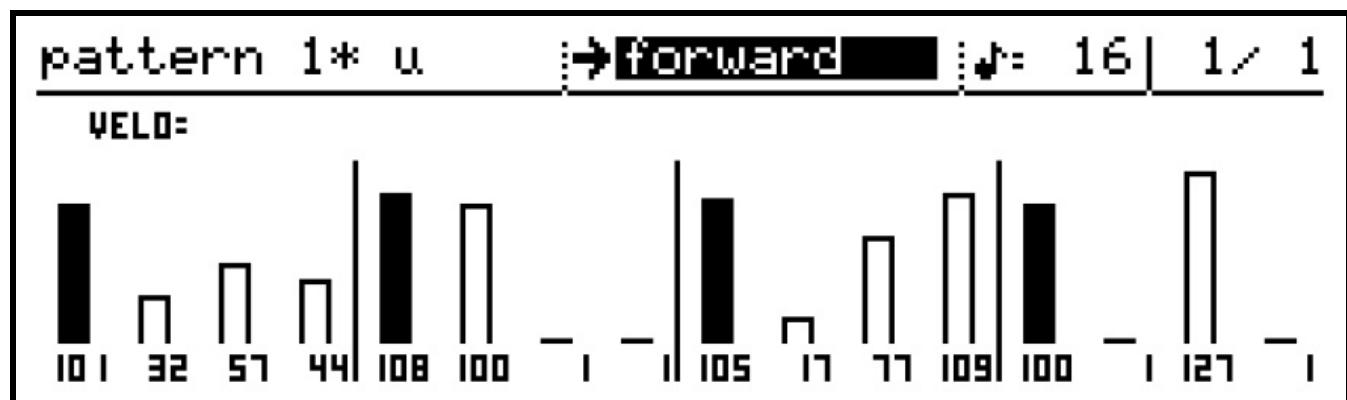
The second multiple-step editing function is slope edit.

This option allows you to adjust the first or last value in a range of steps such that the step values in between follow a straight line between the first and last values.

To use slope edit, hold the step encoder for the first or last step in the range of values you wish to edit. Then, adjust the step encoder at the other end of the range.

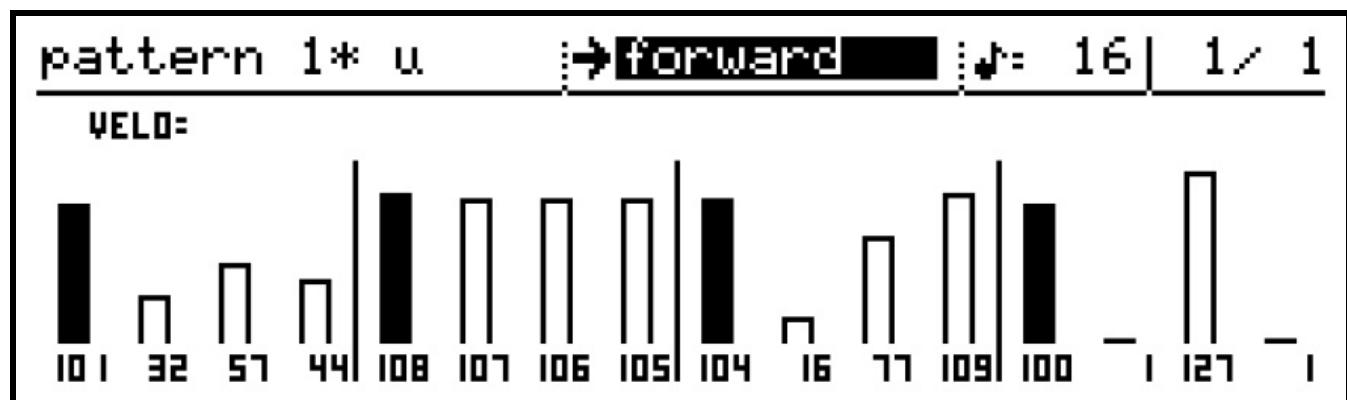
This is best done in row view, so you can easily see the effect.

For example, here is a pattern with varying velocity values:



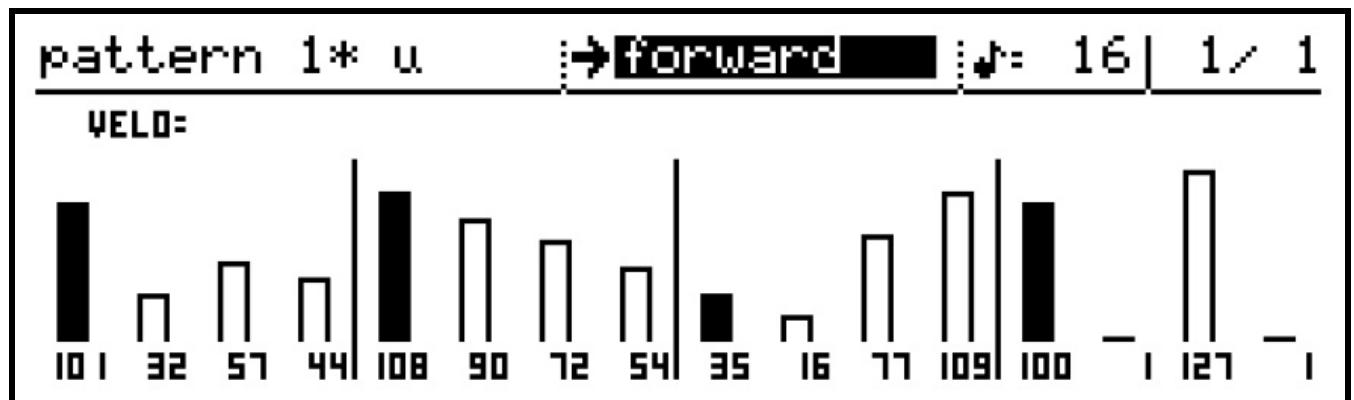
If you were to hold the encoder for step 5 (which has a value of 108), then turn the step encoder for step 9 (which has a value of 105) down by one, step 9 will have a new value of 104.

However, since step 5 was held, all the values between step 5 and step 9 will form a straight line:



As there are 4 steps from step 5 to step 9, and a difference of 4 between the first and last step values, each new value changes by 1 from step to step.

If we were then to hold step 5, and turn to reduce step 10's value by one, this would be the result:



Step 10 has decreased by 1, to 16.

Step 5 holds its value, but steps 6, 7, 8 and 9 are fitted to a straight line between the step 5 and 10 values.

This function is particularly effective for creating chromatic runs of notes and ramping velocity or controller values.

Slope edit works with coarse editing of the values.

That is, if you press and hold a step encoder that you are turning, it will adjust the value in larger increments. Indeed, both the first and last encoders can be held in and turned simultaneously, for rapid adjustment of slopes.

Pattern Level Values

We'll take a break from editing pattern steps to look at the pattern-level values.

Most important of these are the **direction** and **timebase** values.

These values appear on the top line of the pattern edit page.

Here the current direction is forward, and the timebase is 16:

→ forward | ⌂: 16 |

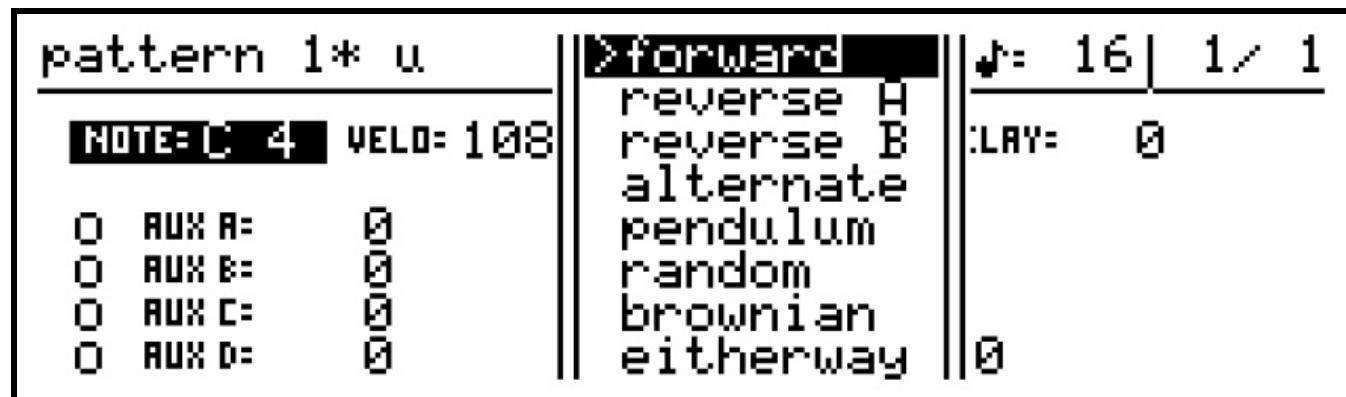
These are edited in the same way as the main values on the play pages – select the value to edit with the VALUE encoder, then press to edit the value.

Both values use a selection list, so the new value selection does not take effect until VALUE/ENTER is pressed to select it.

You can cancel without changing the selection using the MENU key.

Directions

The direction setting for each pattern determines what order the steps will be played in. The available options are shown in the selection list:



The currently active direction is shown with an arrow before it.

The behaviour of each direction is shown in the following table:

Direction	Behaviour
forward	Steps play in ascending order, looping back to step 1 after the last step is reached
reverse A	The pattern will play in reverse, starting from step 1, and looping back to the last step after step 1
reverse B	The pattern will play in reverse, starting from the last step, and looping back to the last step after step 1
alternate	The pattern will play in forward then reverse B directions, as described above
pendulum	The pattern will play forward until the last step, and then reverse direction without playing the last step twice
random	Steps will play randomly
brownian	After each step the pattern will randomly: <ul style="list-style-type: none"> • move to the next step (50% chance) • move to the previous step (25% chance) • repeat the current step (25% chance)
eitherway	There is a 50% chance of moving to the next, or previous step. It could go either way.

Note that steps with **skip** set are skipped in all direction modes.

References in the above table to step 1 really mean the lowest numbered step without **skip** active.

Changes of **direction** take effect immediately. This could lead to a pattern getting out of step with the patterns on other tracks.

To prevent this, after a change of direction, the pattern is flagged for re-synchronisation at the next global bar loop.

The global bar will be described fully in a later section, but briefly it provides a reference for the start of a common 1-bar loop, so irregular length patterns can be kept in step.

Timebase

The **timebase** setting determines the duration of each step as a division of the tempo.

The available timebase values are shown below:

1	2	4	8	16	32	64	2T	4T	8T	16T	32T	64T	Prh
---	---	---	---	----	----	----	----	----	----	-----	-----	-----	-----

The default timebase is **16**.

The numbers relate to the equivalent note length of one step.

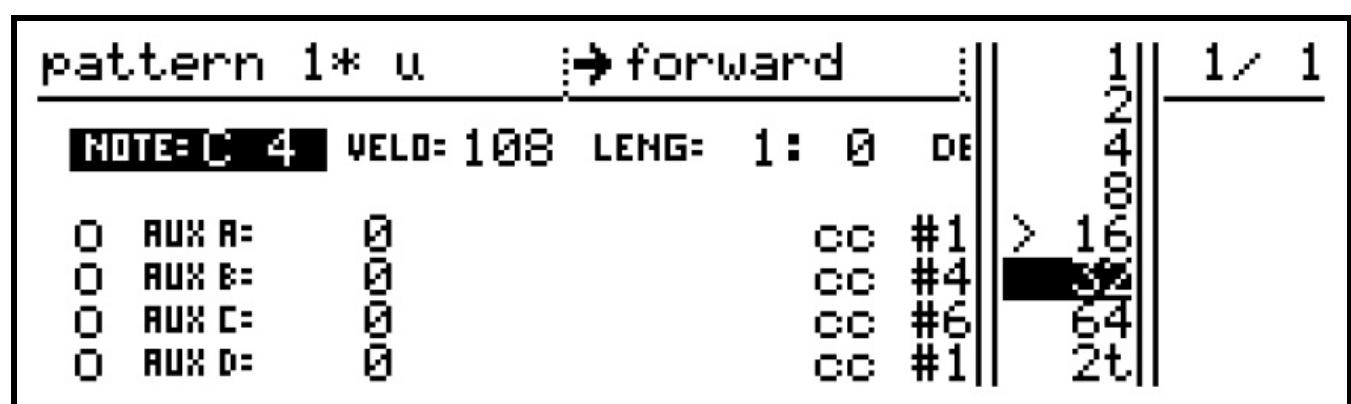
For example, timebase **16** means each step lasts for one **sixteenth** note.

The timebase settings with a **T** after the number are **triplet** settings.

Lastly, Cirklon includes a special timebase setting '**Prh**', for Polyrhythm.

This special option is used along with the "Last step" option, and is described on the following page.

There are more timebase values available than will fit in the height of the display, so you will find that the selection list will scroll up or down to allow access to all values:



As with direction, changes to the timebase value take effect immediately, but the pattern will be flagged for re-sync with the global bar.

Last Step

Although each bar of a P3 pattern has up to 16 steps, you may want to use fewer for a shorter pattern.

This could be done by setting later steps to skip, but a simpler method is to use the bar's **last step** value.

To set the last step, hold down the LAST key. While LAST is held, all the step LEDs will go off, except for the LED in the step which is currently set as last step.

Press any step key to set that as last step.

A useful tip here is that the LAST key is double-click-sticky.

For one-handed operation, you can press the LAST key twice in rapid succession.

The LED will lock on, allowing you to select a new last step in the pattern without having to hold both keys at once.

Another useful tip: SHIFT and LAST will specify a new length for the current bar only.

Polyrhythm Timebase

When the Prh timebase is selected, each step lasts for an equal division of the whole bar, according to the number of steps set by the Last Step operation.

Typically you'd use this to spread an odd number of steps across a regular 4/4 bar.

The easiest way to experience its effects is to create a 1 bar **Prh** pattern then listen to the timing as you dynamically vary the Last Step.

As you reduce the Last Step value, the notes that remain active are forced into sometimes unusual positions in order to fill the bar. The results can be odd, interesting or even startling.

Bar Length

A P3 pattern can be up to sixteen bars in length – a maximum of 256 steps.

To adjust the bar length of a pattern, hold the LAST key as you turn the BAR encoder.

New bars will be added to the end of the pattern, and the new length of the pattern in bars will be updated in the top right of the display:

To select one of the new bars for editing turn the BAR encoder.

You can set pattern edit to automatically follow the current bar in the pattern as it plays using the **bar follow** option.

To activate bar follow, press and hold the BAR encoder, turn it one step in either direction, then release it.

A message box should briefly appear in the display to confirm that bar follow is active:



The BAR encoder LED will also turn red.

You can use the same gesture (press, hold & turn BAR) to deactivate it again.

Bar follow will be deactivated automatically if you turn any of the step encoders.

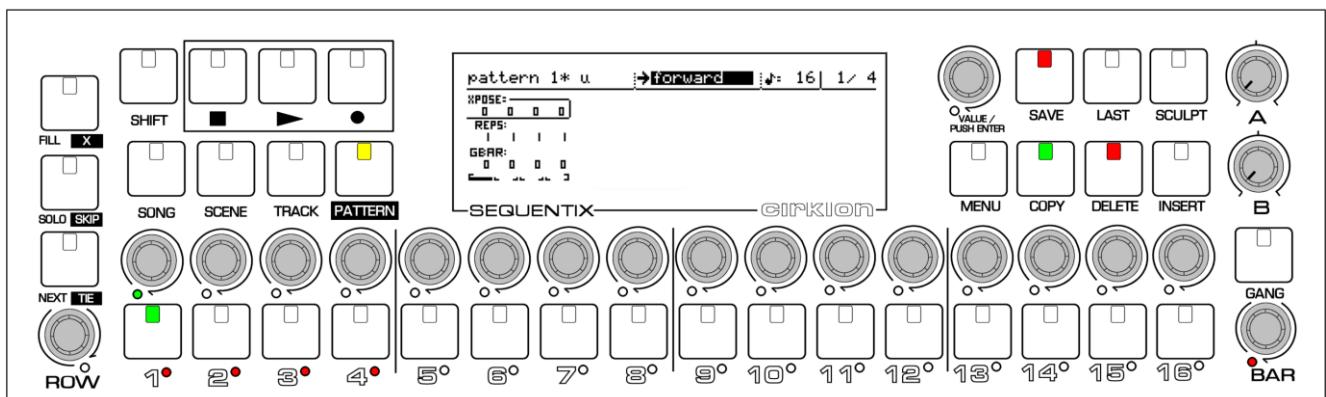
This is to prevent inadvertent edits to steps in the wrong bar if an encoder is being turned as the bar advances.

Now that there are multiple bars in our P3 pattern, we can take a look at some values specific to each bar. These are found in the bar edit page.

Bar Edit

To access the bar edit mode, press the BAR encoder.

The display switches to a new view, shown here with the full panel, as it would appear for a 4 bar pattern:



Note that the ROW encoder LED, which was red in step edit, has turned off.

The BAR encoder LED will be green, unless bar follow is active, in which case it will remain red.

The step keys and encoders take on different functions while in bar edit.

The current bar playback position is shown by a single step key LED.

The length of the pattern in bars is shown in red by the LEDs under the step keys.

The encoder LEDs indicate the current bar selected for editing in green.

You can press either the step keys or encoders to change the current bar selection.

The COPY and DELETE keys also light to show that they are active for use in bar edit.

To delete a bar from the pattern, hold the DELETE key and press the step key for the bar to be removed.

Bar Copy

To copy the contents of one full pattern bar to another bar in the pattern, first ensure the bar you wish to copy is selected as the current bar, either by pressing the step key or encoder, or turning the BAR encoder.

Hold the COPY key, then press the step key for the bar you wish to copy the current bar into. The current bar selection will remain on the source bar.

You can confirm the bar was copied by selecting the destination bar, then switching back to step edit mode by pressing the ROW or BAR encoders. All step values and flags will match the source bar.

Bar Extend and Copy

You can simultaneously increase the bar length of a pattern and copy the current bar into all new bars added using the extend and copy function.

This operation is performed by holding COPY, then pressing a step key to select a destination bar which is beyond the current last bar of the pattern.

The bar length will be increased, and any new bars added will be copies of the current bar within the previous pattern length.

Bar Values

Let's look more closely at the bar edit display:



There are three rows of values in the lower part of the display, labelled XPOSE, REPS and GBAR, with one of each value for every bar in the pattern.

These values are edited using the step encoders for each bar number.

You can select which row of values to edit by turning the ROW encoder.

The XPOSE values allow you to transpose each bar up or down by up to 5 octaves.

The REPS values allow you to set each bar to repeat up to 99 times.

The GBAR value can be 0 or 1, to represent OFF or ON.

This is the global bar sync option. When active (set to 1), the bar will be reset to its first step every time the global bar loops.

This will only have an effect if there are steps skipped in the bar, or its last step has been set such that the total number of steps playing in the bar is not a factor of global bar length.

The practical result is that an odd-length bar will play for the full length of the global bar before the pattern advances to the next bar.

Remember that the global bar value is set on the scene page – by default it is 16. Larger values are worth experimenting with, especially if you wish odd-length bars to play multiple times before resetting.

Bar Loops

It is possible to set a bar loop within a P3 pattern.

Any bars before the start of the loop will play only once when the pattern starts to play.

Once in the loop section, the pattern will return to the first bar of the loop after the last.

Any bars in a pattern after the loop end will never play.

To set a bar loop, hold the SHIFT key, then press two of the step keys in turn, to set the first and last bars of the loop section.

The LEDs below the looped bars will remain red, but bars before or after the loop section will show amber.

You may have noticed the graphic below the bar values.

This shows the length of the pattern in bars and the current bar playing, and any bar loop.

For example, if we loop bars 3 and 4 of a 4 bar pattern, the value display would look like this:



- the larger marks around bars 3 and 4 indicates the loop limits.

The bar 4 marks are filled to show it's the current playback position.

Inserting empty bars

It's possible to insert a new bar in your pattern, moving the following bars along.

Hold INSERT and press a step key to insert an empty bar at that point.

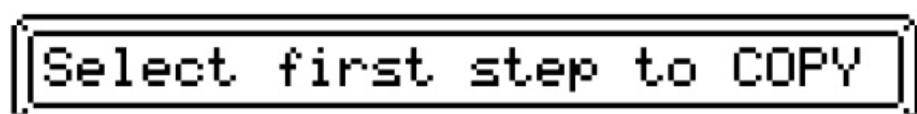
More On Step Editing

Let's return to step edit, and look at some more advanced operations.

Copy / Insert

Any number of steps in a P3 pattern can be copied to a clipboard, then inserted (pasted) at another location in the same pattern, or into another P3 pattern.

To copy steps, press the COPY key. This message will appear briefly:



The COPY key will flash to show that a copy operation is expected, and the top line of the display will show:

COPY: <select>

| 1/ 4

Press the first step of the section to be copied.

For example, if you press step 1 of the first bar in the pattern, the top line will now show:

COPY:bar 1 step 1 to <select>

| 1/ 4

Then press the last step of the section to be copied.

Note that you can use the bar encoder to move to another bar of the pattern before pressing the step key.

As soon as you press the second step key, a brief message will confirm the length in steps of the copied section:

Copied 64 steps to clipboard

Now you are ready to insert the clipboard contents elsewhere in the pattern.

Or you can return to play mode to select another pattern for editing if you wish to copy steps from one pattern to another.

To insert the clipboard contents into a pattern, press the INSERT key.

The top line of the display will change to something like this:

INSERT: 64 steps of all

| 1/ 4

It shows the number of steps on the clipboard, and a filter for the values to be inserted.

When you copy steps, the values from every row are copied for each step, but when you come to insert them, you can choose which values are to be changed.

The filter option is changed using the VALUE encoder.

The options are:

- **all** all row and flag values
- **all-aux** all row and flag values, except for the aux rows
- **auxes** only the aux rows
- **note** only the note values
- **velo** only the velocity values

- **length** only the length values
- **delay** only the delay values
- **flags** only the gate, tie, skip and X flags
- **aux A** only the aux A values and flags
- **aux B** only the aux B values and flags
- **aux C** only the aux C values and flags
- **aux D** only the aux D values and flags

To insert the selected values into the pattern, press the step key where the first step in the clip should be written. The full length of the clip will over-write each step from that point on.

If the end of the pattern is reached before the full clip has been inserted, insertion will wrap to the start of the pattern and continue.

When copying steps to the clipboard, if you select a last step before the first step, the selected steps are copied to the clipboard in reverse order.

Small sections, or the pattern can be reversed in this way.

Note that bar values are not copied using step copy and insert.

Sculpting and Randomisation

The sculpt feature allows you to use knobs A and B to continuously edit the values of a pattern as it plays.

Sculpt must first be enabled by holding the SHIFT key and pressing SCULPT.

The SCULPT key will light to show it is now active.

The top line of the display will also switch to show the sculpt *mode*, and a preview of the values for the current knob positions.

While sculpt is active, and the pattern playing, holding the SCULPT key will modify the value for the current row and step, according to the knob positions.

There are four sculpt modes. You can change mode by pressing VALUE/ENTER, then selecting the new mode by turning and pressing ENTER once more.

The four modes are:

Sculpt ABS - the position of knob A will over-write the value in each playing step.

Sculpt REL - the position of knob A will adjust the stored value by a **relative** amount. This can be a positive or negative offset, with zero effect at the mid-position of the knob.. If you hold SCULPT as the pattern repeats, each step will move incrementally further from its original value.

Random ABS - each playing step value will be over-written with a value randomly chosen from the range of values set by knobs A and B. It doesn't matter whether knob A or B is greater.

Random REL - each step is adjusted by a random offset, again in the range between knobs A and B. Multiple passes will cause a cumulative change to the stored values.

The best way to appreciate the effect of these functions is to experiment with them while looking at the row view.

Audition

This is a useful feature if you are editing a P3 pattern while playback is stopped.

If you hold a step encoder, then press the step key below it, the note value on that step will be sent to the assigned instrument, so you can confirm the note selection by ear.

Configuring the Aux Rows

In a new P3 pattern, the aux rows are assigned to some commonly used MIDI CC numbers. Each aux row can be configured to use any other CC number if required.

They can also be used to send MIDI after-touch, pitch-bend, program change and NRPN messages, or they can be assigned to a range of “aux events”, which will be described in a later chapter.

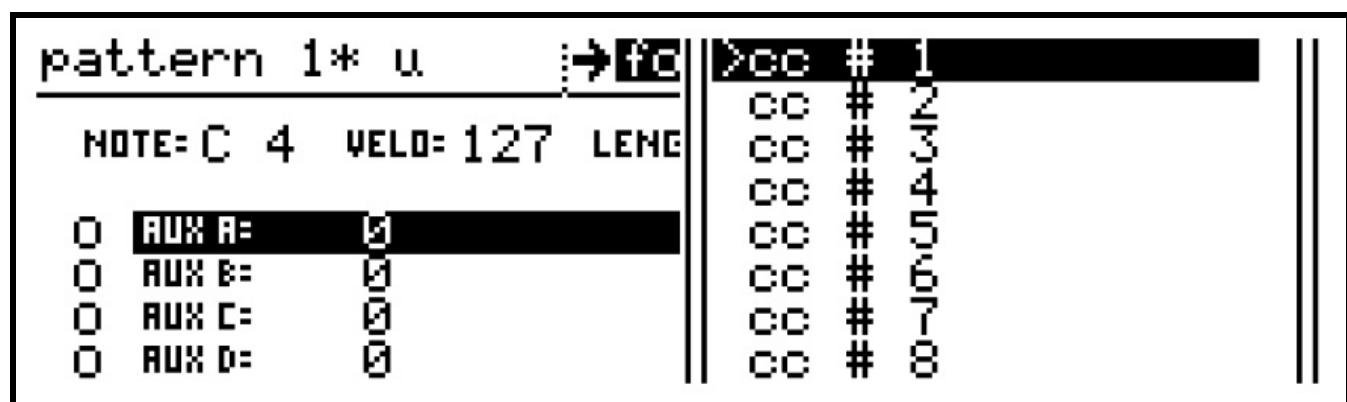
To change the assignment of an aux row, first make sure it is selected as the current row using the ROW encoder.

Then press and hold the ROW encoder.

While *holding* it in, *turn* it one step in either direction.

This gesture will bring up a selection list on the right of the display.

For a row currently assigned to a MIDI CC, the list will look like this:



The arrow next to “cc # 1” shows that this is the current assignment for the row.

Use the ROW encoder to scroll to a new CC number to assign, and press ROW to confirm the selection.

If you want to cancel the operation, press the MENU key.

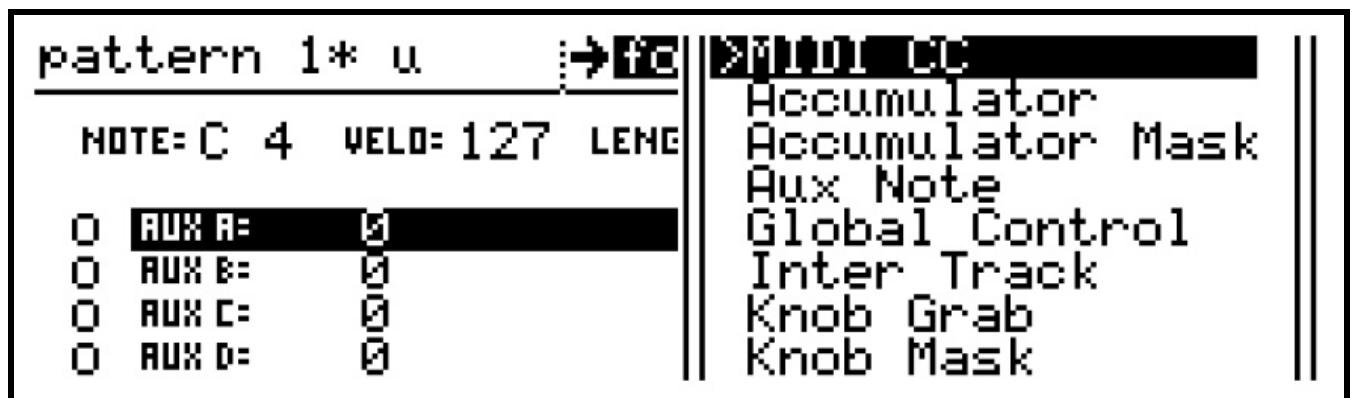
Scrolling through this list, you will see only MIDI CC values.

For the other MIDI message types and the aux events, there is a second level selection to access groups of all possible aux assignments.

Press, hold and turn the ROW encoder to access the first level selection list.

Once in the list, press, hold and turn ROW a second time.

The group list should then appear:

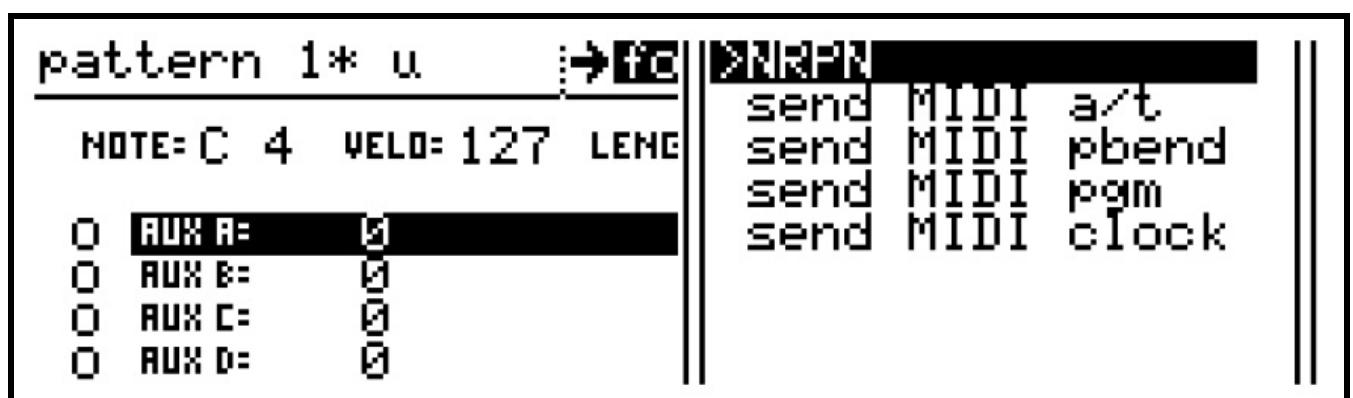


Most of the entries are groups of **aux events**. Their use is covered in detail in a later section.

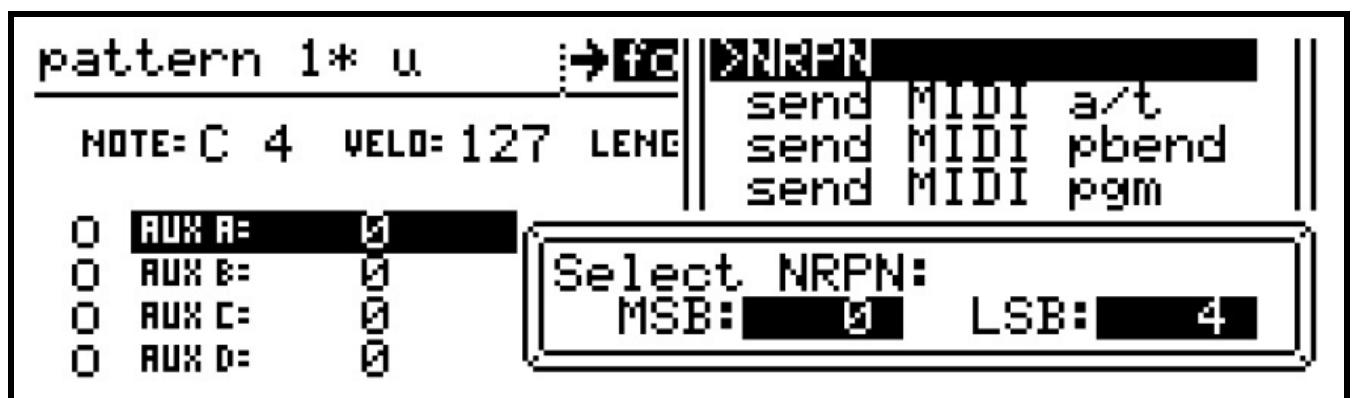
To send other types of MIDI message, scroll down and select the “MIDI Send” group.

Pressing ROW will take you back to the first level list for the new group.

In the MIDI Send group, the options are:



NRPN messages require one or two parameter selection numbers (MSB and LSB), so if you choose the NRPN option, an extra step is required to configure those values:



While the “Select NRPN:” option is shown, step encoders 9 and 11 (which are directly below the MSB and LSB values on the display) will light red.

They are used to select the required MSB and LSB values.

With the selection made there, press ROW once more to confirm the new assignment of an NRPN to the aux row.

P3 Edit Menu Options

Further edit operations can be found on the edit menu.

This is accessed from the top level edit pages (step or bar edit) by pressing the MENU key. While the menu is active, a number of options appear along the bottom of the display:



The step encoders under each of these options will light red as a visual cue that the menu is active. The encoder below each option is pressed to select it.

First step

The first step function allows you to pick any step in a pattern, and have the entire pattern rotated such that the chosen step becomes the first step of the first bar of the pattern.

After pressing encoder 6 to select this option, a brief message will appear, then the top line of the display will show:

Select new FIRST STEP | 1 / 1

Press the step key for the step you wish to become the new first step of the pattern, and all steps will move around accordingly.

To cancel the operation before selecting a new step, press the MENU key.

Note that bar values are not moved in this operation.

ReSync

The resync menu option toggles the automatic resynchronisation of patterns on or off.

Patterns are normally resynchronised at the end of the global bar whenever an edit is made that changes the effective length of the pattern.

Such edits include changes to direction, timebase, step skipping and bar length.

Sync

The sync function allows you to manually reset pattern playback to the first step of the bar.

Apply FTS

This option permanently applies the current “force to scale” setting to the notes of the pattern. Force To Scale is a scene-level setting, and will be covered in the following chapter on scenes.

Aux Macro & Accumulator Config

The aux macros are pre-defined sets of aux event assignments for the aux rows.

The accumulator config page allows you to alter pattern settings related to advanced use of aux events. Please refer to the later section on aux events.

Having covered just about everything in pattern edit for P3 patterns, let's get back to some higher-level operations.

PATTERN CLEAR Short-cut

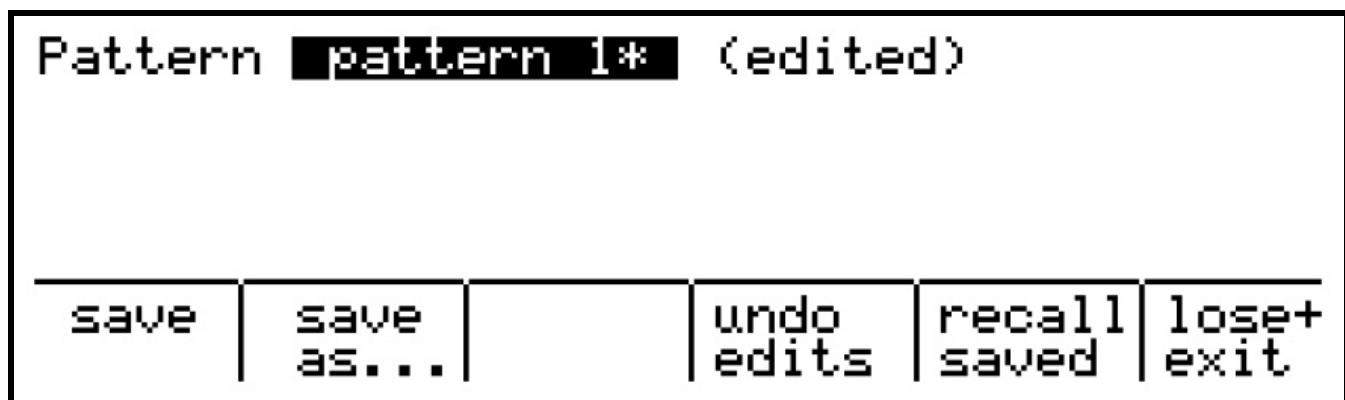
To quickly wipe the contents of the current pattern, retaining only the bar length setting, you can hold the DELETE key, and press PATTERN.

A confirmation page will prevent accidental use. This feature is handy when you just need to start from scratch.

Saving Patterns

A newly created pattern will not be lost when you exit from pattern edit mode. However, saving a pattern allows you to protect the pattern so that further edits can always be undone, restoring the pattern to the state it was in at the time of saving. You can also save an edited version of a pattern as a new pattern, leaving the original pattern as it was before the recent changes were made.

The pattern save menu can be accessed in pattern edit by pressing the SAVE key:



Depending on the state of your pattern, not all of these options will appear. It is always possible to undo changes made to a pattern since you most recently entered pattern edit mode. For a pattern which has been saved, there is also a second level of undo, which allows you to reverse any changes made since the point at which the pattern was saved.

A newly created pattern will be **unsaved**

In this case, the save menu will only show the choices of “save” and “save as...”.

The “save” option will mark the pattern as saved.

“save as...” will prompt for a new pattern name, then create a copy of the pattern under this new name, also marking this new pattern as saved.

After you make changes to a pattern which has previously been edited, you will see the “undo edits” option.

This option undoes only the changes made to the pattern since you most recently entered pattern edit. This option will appear for saved or unsaved patterns.

If you edit a saved pattern, the “recall saved” option appears.

This option reverses all changes made to the pattern since it was saved.

Finally, the “lose+exit” option will undo the most recent changes to a pattern, and exit from pattern edit mode. This will be equivalent to “undo edits” if there have been changes made

since you entered pattern edit, but it will restore the saved version of the pattern if no changes had been made.

Saving Patterns – to SD card / internal flash

There's an extra option for saving patterns as a separate file to an SD card in the rear panel slot, or the internal flash storage (according to the current setting chosen on the card LOAD or SAVE menu – see the later chapter on storage for more details).

This can be useful when there are patterns you wish to use in multiple songs – e.g. specific metronome grooves or favourite Aux event combinations.

In pattern edit, press SAVE, then COPY. A menu similar to this will appear:

```
LEAVE pattern: "HH-RAND"
to file: HH-RAND.CKP
[Save path]:
SD:/
```

It shows the current pattern name and the filename that will be created in the folder specified by "Save path". You can change the filename or path by turning the VALUE encoder to select either value and pressing to edit the filename or select a different folder.

If the idea of saving these 'template patterns' appeals, consider adopting meaningful names from the start. Life's confusing enough.

To load a pattern saved in this way elsewhere, go to the track where you'd like to use it and go to the pattern selection list. Then press INSERT, and the file load menu will appear. Select the pattern to load, and press ENTER to confirm.

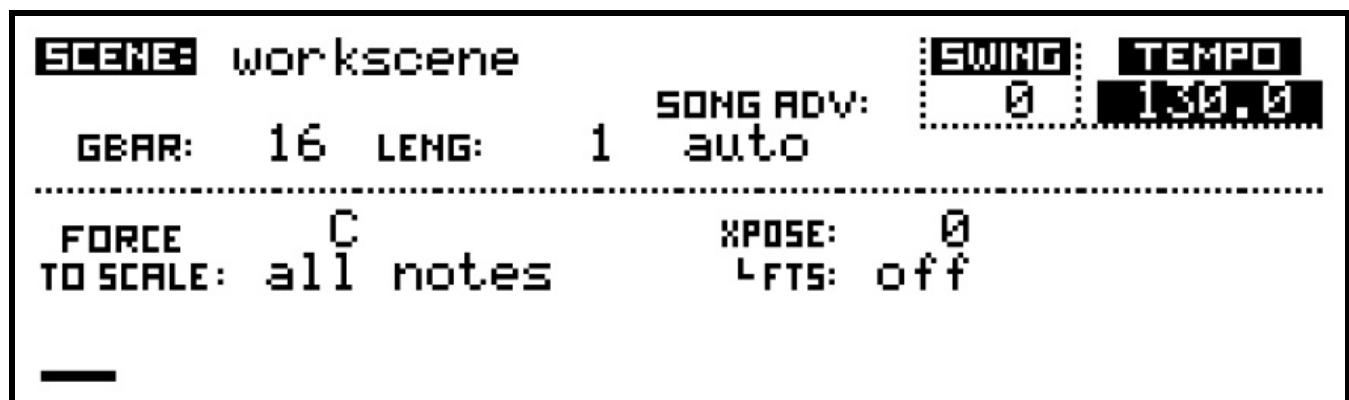
```
LOAD SD:/
[temp]
4-4.CKP          13/05/2013  20:25
HH-RAND.CKP      02/02/2020  14:55
MADMETRO.CKP    02/02/2020  14:57
PRC-REPC.CKP    02/02/2020  14:56
PUSH-FALL.CKP   02/02/2020  14:58
```


4. Using Scenes

Now that we have instruments assigned to tracks, and have created and edited some patterns to play on those instruments, we can look at how to assemble groups of patterns into a longer piece of music, using SCENEs.

Remember that the patterns themselves are not part of a scene - scenes are simply a record of which patterns are to be played by each track, the initial active/muted status for each track, and a number of other scene-level values.

The main scene values are viewed and edited on the SCENE page:



The VALUE encoder is used for selection and editing of these values just as on the song page.

The top line of the display shows the scene name.

In this case, we are looking at the **workscene**.

The workscene is a temporary holding area for the values and pattern assignments that make up a scene.

You begin each new song in the workscene, since there will not yet be any saved scenes.

As you save the workscene to real scenes, you will remain in the workscene.

Each save will store the current state of the workscene as a new scene at the end of the scene list.

This allows you to build a song from a sequence of scenes by gradually adding or changing patterns on some of the tracks, while leaving other tracks unchanged, without messing around with a lot of scene save-as operations.

Once you are working with a stored scene, you will be able to change the name here, but the workscene cannot be renamed.

Gbar

This is the global bar length - the length of one bar in the scene, in 16th note steps. In song play mode, it is used along with the length value to control how long a scene will play for before advancing to the next scene.

It also provides a common reference for the start of a bar for odd-length P3 patterns that you wish to resynchronise with the other tracks at regular intervals. This is achieved by turning on the bar-level gbar setting described the in P3 pattern edit section.

Length

The scene length is the number of bars (of Gbar steps) the scene will play for before advancing to the next scene in song play mode.

It can take a value from 1 to 200.

The current scene length is also used as the default length when creating new patterns. If you typically work with patterns of 4, 8, 16 bars, etc., you can set this value to your preferred length to save the need to edit the bar length of each new pattern manually.

Song Advance

This option controls whether or not the scene will advance to the next scene while in song play mode. Set to the default of “auto”, the scene will advance after one pass. Set to “man”, the scene will loop indefinitely, until a new scene is selected from the scene list, or you select the next scene using the shortcut of holding the NEXT key then pressing SCENE.

Force To Scale

With Force to Scale (FTS) active, all notes played will be forced into the chosen scale (or key signature).

If a note is not included in the current scale, it will be moved to the nearest note which is.

There are two FTS values - the root note and the scale. Each is selected from a list. There are a number of preset scales to choose from, and also space for 8 user defined scales. The user scales are stored at the song level – that is, each is accessible only to the song it is defined in.

Any of the scales can be edited, though edits will only be saved permanently to user scales. To edit a scale, access the selection list as if to assign the scale, then hold SHIFT as you press ENTER.

In FTS edit, the step keys are used to toggle each note of the chromatic scale in the FTS scale. Note that the root note cannot be disabled.

FTS has a number of uses:

- Making sure all notes played are in key while making live pattern edits
- Changing the FTS root or scale while playing for instant key changes
- Editing the notes in the FTS scale in real-time as a performance tool

What FTS Does

To explain the effect of FTS, let's say you have a pattern with 12 notes in a rising chromatic scale:

Step:	1	2	3	4	5	6	7	8	9	10	11	12
Note:	C	C#	D	D#	E	F	F#	G	G#	A	A#	B

With FTS set to a **root note of C** and the **Major scale**, the notes in the pattern above which are not in the key of C Major will be moved to the nearest notes which are.

The actual notes played would therefore be:

Step:	1	2	3	4	5	6	7	8	9	10	11	12
Note:	C	C	D	D	E	F	F	G	G	A	A	B

Note that **the pattern itself is not changed** – the notes are corrected **dynamically** as the pattern plays.

This may be a problem for certain instruments, such as drums, where you don't want the wrong sounds to be played. The option "No FTS" in the instrument definition allows you to disable force-to-scale for any such instrument.

You can also disable FTS on any step of a P3 pattern, by using the transpose defeat, or X, flag.

Xpose

This is the scene level transpose value.

All patterns will be transposed by the selected value, unless the instrument option "No Xpose" is enabled, or the X-flag is set on some pattern steps.

The FTS option linked to the transpose value controls whether the force-to-scale settings are applied before (pre) or after (post) the scene transposition.

With the Xpose FTS option is set to post, in effect, the root note setting for FTS is also transposed.

Saving a SCENE

Whenever you create a new song, you will start in the workscene.

Once you have created some patterns for the first section of your song, the workscene can be saved to a stored scene.

To do this, ensure you are on the SCENE page, then press the SAVE key.

You will be prompted to enter a name for the scene:



You can edit this name in the usual way, or just press ENTER to accept the auto-generated name.

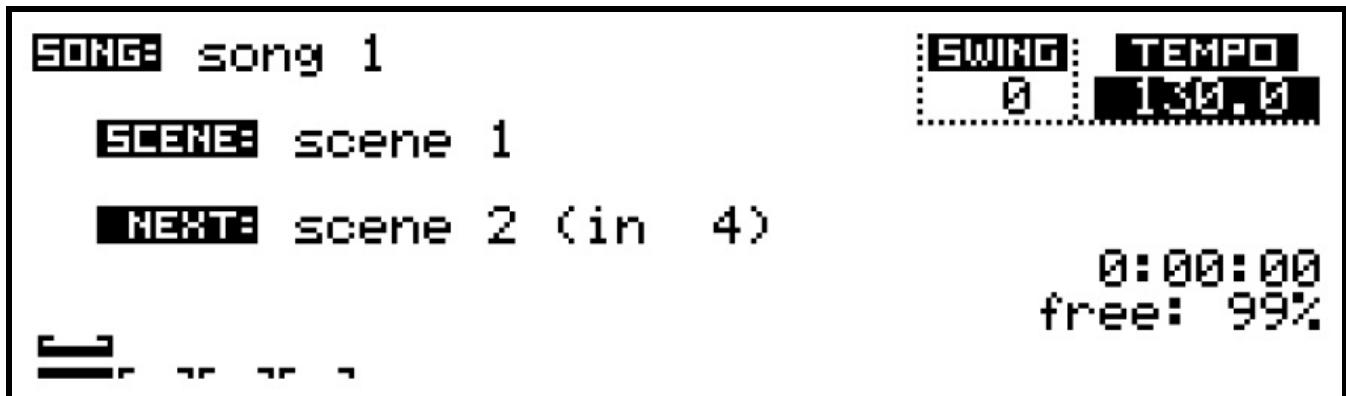
Once the scene has been saved, you will remain in the workscene, as described before.

This is intended to simplify the progress of the initial building of a song.

Once a number of scenes have been added, you can switch to song play mode by switching to the SONG page, or manually enabling song play.

Whenever you enter song play mode while in the workscene, at the end of the next global bar, the first scene of the song will be loaded and the pattern assignments and initial active/mute status of the tracks for the scene recalled.

The names of the current and next scenes are shown on the SONG page:



A countdown of bars until the next scene change is shown to the right of the next scene name. In this example, Scene 1 is the current scene.

Scene 2 is set to follow after 4 bars (4 repeats of the gbar length defined for Scene 1).

The graphic at the very bottom of the display provides a visual indication of the current scene length, and the playing bar position in the scene.

The single 'bar' indicator second from bottom shows the length and current position of the pattern assigned to the current edit track.

If the Song Adv option for scene 1 was set to manual, it would not advance, and the text **manual** would appear in place of the countdown.

A manual scene allows you to have a point in a song where the patterns of a scene will repeat indefinitely until you manually advance to the next scene.

You do this by holding the NEXT key and pressing the SCENE key.
A brief message will confirm the pending scene change.

Mute Hold

As each scene is recalled during a song, the initial active or muted status for each track is also recalled.

The Mute Hold feature allows you to over-ride this behaviour if you wish.

To change the mute hold option, hold the SHIFT key and press SCENE.

There are three possible settings:

OFF - mute settings are always recalled

Next - for the next scene to be recalled, the current mute status will be preserved, then mute hold will revert to OFF

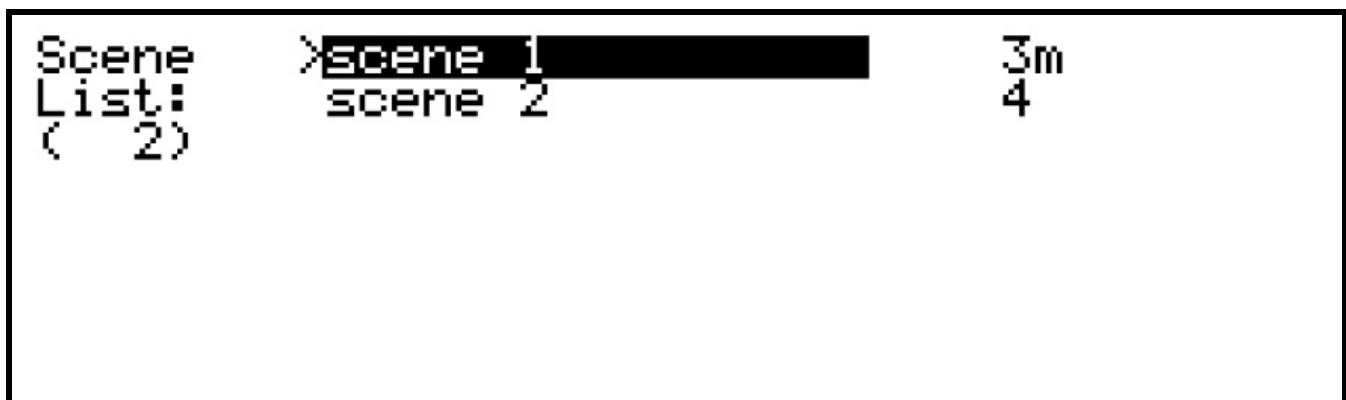
LOCK - for all scenes recalled, the current mute status will carry over

As you change the mute hold setting, a popup message on screen will confirm the new mode.

The Scene List

In order to view all the scenes in a song, or to manually recall a scene other than the next scene in the song, you must use the scene list.

The scene list is accessed from the SCENE page by pressing the SCENE key. If you are not on the SCENE page, two presses of SCENE will take you to the scene list.



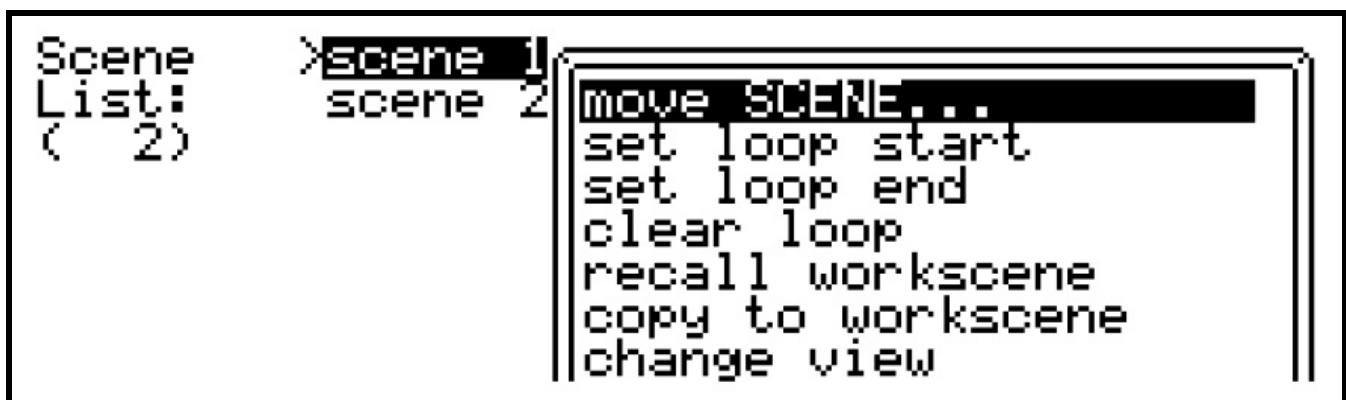
This list shows our example song, with two scenes.

The number to the right of the scene name shows the length of the scene, except for the scene currently playing. For the current scene, the value counts down, to show how many bars will play before the scene loops (or advances, if song mode is enabled). A scene set to manual advance will show a letter 'm' next to the length/count number.

Using the VALUE encoder, you can scroll to any scene on the song, then press to recall that scene on the next global bar loop, or immediately if Cirklon is stopped.

The arrow just before the scene name indicates the current scene playing.

The scene list provides some other scene operations, accessed by pressing the MENU key:



Move SCENE

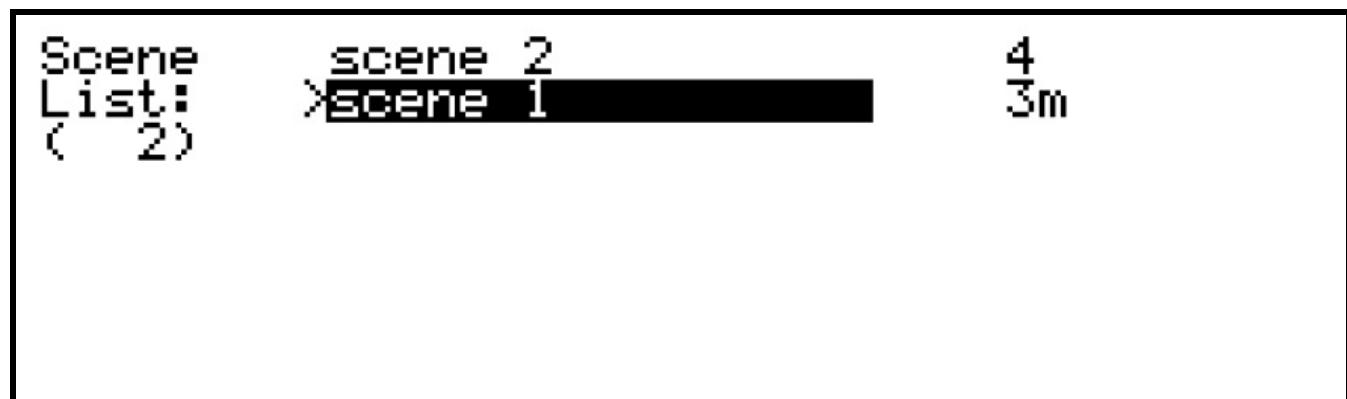
This function allows you to move the selected scene to another position in the song. Select the scene in the list you wish to move, press MENU, then select move SCENE.

The display will show the full list of scenes. Scroll to the scene **before** the new location for the selected scene.

An extra entry for “- start of song -“ allows you to move a scene to the very start of the song. In this example, Scene 1 has been selected for moving:



...pressing VALUE/ENTER will reposition Scene 1 *after* Scene 2, thus changing the order of the song:



Scene Loops

Normally in song play mode, a song will start at the first scene, play through all scenes, and then repeat the final scene indefinitely.

A scene with all tracks muted can be saved as last scene to set a definite end to a song if required.

The menu options “set loop start” and “set loop end” allow you to set a smaller section of the song to loop.

First select the scene you wish to start or end the loop with in the list.

Then press MENU, and select one of the “set loop ...” options.

The labels “loop...” and “...end” beside the reps counts will show the loop points.

The “clear loop” option removes both loop settings from the song.

Recall / Copy To Workscene

These two options allow you to return to the workscene, to continue building a song.

If you switched to song play mode while you are working on a scene that you hadn’t saved, “Recall workscene” will restore the workscene so you can continue to edit or save it.

“Copy to workscene” copies all the pattern assignments and values for the current scene into the workscene and switches to it.

This allows you to return to the building phase of a song, incrementally saving variations on the workscene as new scenes at the end of the scene list, without having to use save-as and manually naming each scene.

Deleting A Scene

You can delete the currently selected scene in the list by pressing the DELETE key.

A confirmation screen will follow.

Note that only the scene itself is deleted.

Any patterns assigned in the scene will remain in memory, whether they are used in other scenes or not.

Change View / Scene Select View

For rapid access to a larger number of scenes, the scene list has an alternative view, where scene names are shown in two columns of up to eight scenes.

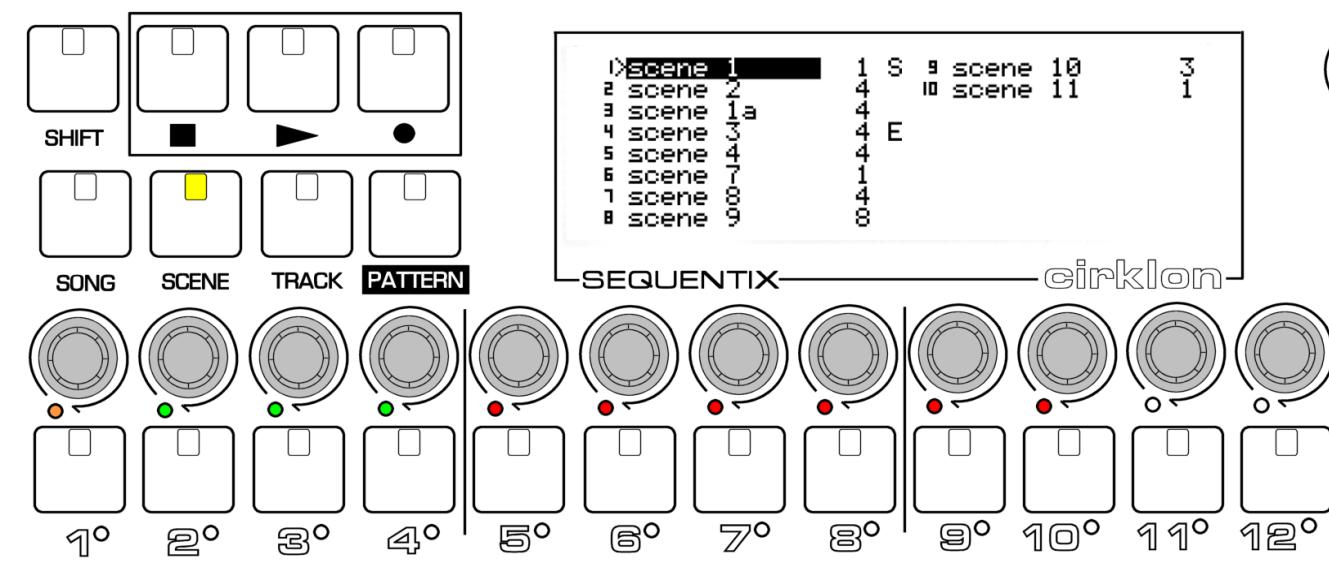
This is the **scene select view**.

Scene selections can still be made using the VALUE encoder as before, but the step encoders can be used to select scenes directly, and to quickly set scene loop start and end points at the same time.

The “change view” menu option is used to switch between the two views.

Your choice of view is stored permanently.

An example of how Cirklon might look in scene select is shown below:



This example song has a total of 10 scenes, not numbered contiguously as some scenes have been edited and saved-as, while others have been deleted.

In the scene select view, each scene is mapped to one of the step encoders.

The current scene is shown by a flashing amber LED.

All other scenes are shown with a red LED, unless they are within a scene loop, in which cases they have a green LED.

On the display, the scene names are shown in two columns (if required), with the number of the step encoder the scene is mapped to just before the name.

The scene length (or bars remaining count for the current scene) is shown after the name, with the letter 'm' to indicate a scene set for manual advance.

The first scene in any scene loop has a letter 'S', and the last scene in a loop a letter 'E'. A '>' symbol appears between the step key number and the name of the current scene.

Any scene shown on the display can be selected directly by pressing the step encoder mapped to it.

Unlike the scene list view, selecting a new scene here will not return you to the main scene page.

While a change to another scene is pending, an asterisk (*) will appear before that scene's name.

You can also scroll the highlight selection through the scenes using VALUE, and select the next scene by pressing ENTER.

If there are more than 16 scenes in a song, the VALUE encoder can be used to move back and forth through all the scenes. As you do so, a different bank of scenes will be mapped to the step encoders for direct selection.

A scene loop can be set quickly here by holding one step encoder, then pressing and releasing the encoder for a later scene.

The first scene selected will be queued as the next scene to play, and also set as the scene loop start. The later scene will be set as the scene loop end.

This mode may be preferable to the basic scene list for improvising the structure of a song in real-time, rather than sticking to a pre-determined scene order.

When jamming with scenes, the next scene usually starts at the end of the global bar. You may, however, prefer the scene to run for its full length before switching. If this is the case, hold SHIFT when selecting the next scene. **Remember that Scene length is calculated by multiplying the LENG field with the GBAR value – currently it is always a multiple of GBAR.**

Editing a Saved Scene

When you have recalled a saved scene, either manually or by entering song play mode, you can make changes to the pattern assignments and mute status of tracks, or change the scene level values.

Changes made to a saved scene will **not update the saved version of the scene**.

You must normally re-save the scene if you want any changes to be permanent.

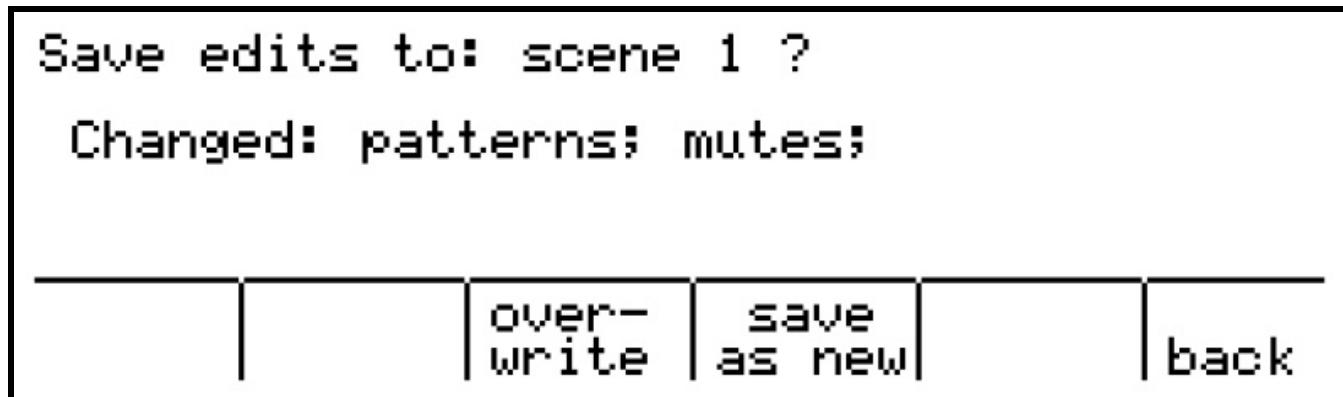
This prevents unexpected changes to finished scenes in songs where you might want to make temporary changes to mutes, etc. during playback.

In some cases, for example where you are still working on the structure of a song making changes to various scene values, you may find it easier to have changes to scenes save automatically.

To enable this, you can turn on the “Scene auto-save” option found in the play mode menu – see the later section for details.

When any changes are made to a scene so that it differs from the saved version, the SAVE key will light when you are on the SCENE page.

When you press the SAVE key, a display like this will appear:



This shows a reminder of what you have changed, in case you lost track.
The step encoders under the three options will light red.

You can:

- over-write save the changes to the scene
- save as new save the altered scene to a new scene
- back return to the scene page without saving

When you choose the over-write or save-as-new options, further confirmation pages may then appear.

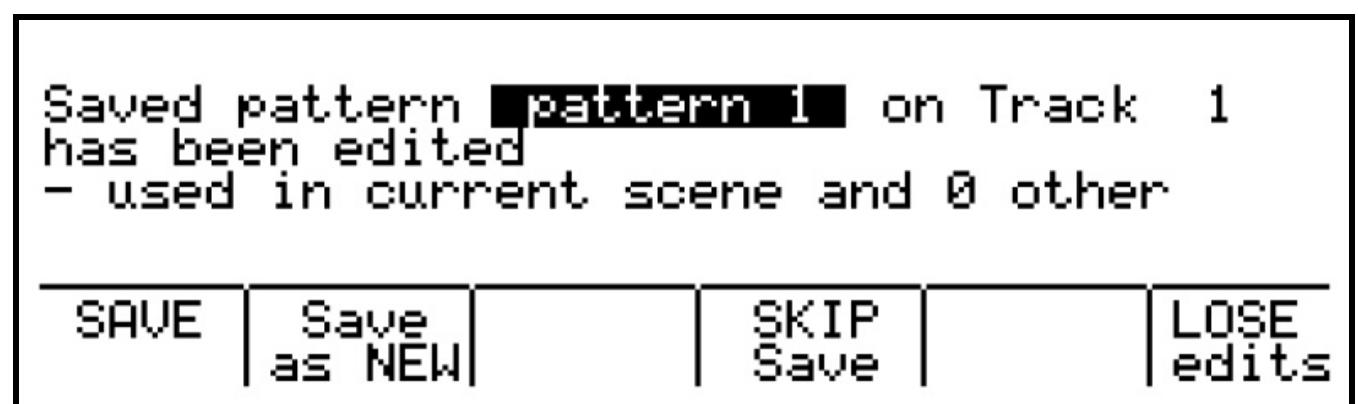
Pattern “Saved” Status

As described earlier, each pattern has a status value which records if the pattern has been saved. An additional value records how many scenes make use of the pattern.

When you save a scene, all the patterns assigned in that scene will normally have their status updated to saved, and their scene usage count updated.

You should normally save any edits made to patterns at the same time as the scene is saved, if they have not already been manually saved in pattern edit.

For each saved pattern which has been edited, the scene save process will present you with the following options:



You can:

- save the edits to the original version of the pattern, thus affecting all scenes that use the pattern
- save the edited version of the pattern to a new pattern
- skip saving for now, leaving the edited version in place (not recommended if you don't want to risk unexpected changes if you later save this pattern for another scene)
- lose the changes, and restore the original version of the pattern

To stop this process becoming too long-winded, you are not prompted to edit the pattern names either when saving, or saving as new.

New patterns will have their names auto-generated based on the existing name.

If you are choosy about your pattern names, remember to edit the name and save your patterns manually when you finish working on the pattern itself.

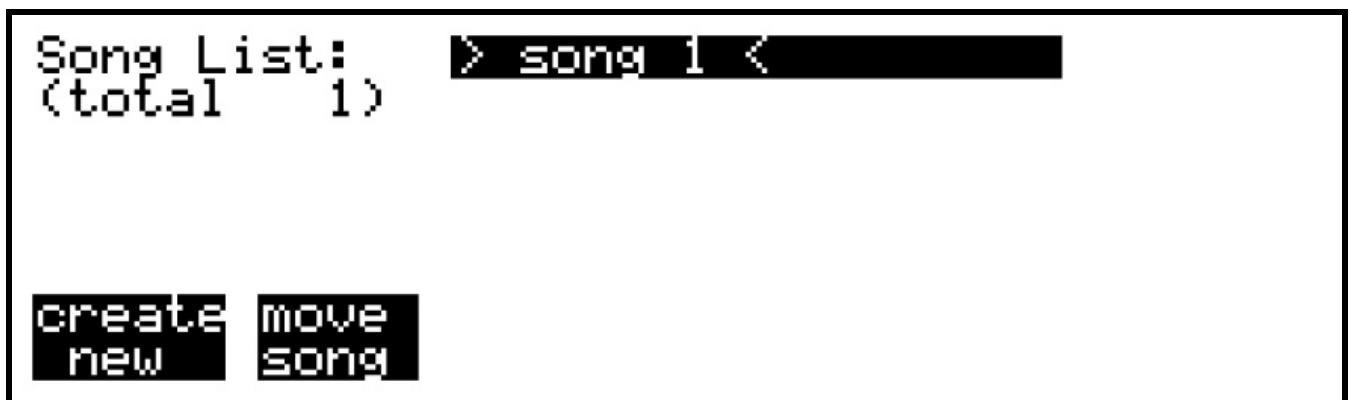
5. Creating Another Song

Having covered the basics of creating the first song from scenes and patterns, let's look at how to start a new song, so we can do the whole thing again.

Existing songs are selected, and new songs created, from the SONG list. The SONG list is accessed by holding SHIFT while pressing the SONG key.

Note that you can access the SONG list without first going to the SONG page – that is, you won't automatically activate song play mode if you don't want to.

The song list will look like this:



Pressing encoder 6, under “create new”, will create “Song 2”, and return you to the play mode page you were on (SONG, SCENE or TRACK)

Once there is more than one song in memory, you can recall any other song from the list. You may also change the position of any song in the list by pressing encoder 7, under “move song”. As a short-cut for song selection, you can advance to the next SONG in the list by holding the NEXT key and pressing SONG, in the same way as for scene advance. Using the move song option allows you to re-order the songs in memory as a set for live performance.

To delete a song from memory, select the song in the list, then press the DELETE key. A confirmation page follows, where you must press encoders 6 and 7 together to remove the song.

The first task needed for any new song is to assign instruments to one or more tracks. To ease this task, there is an option to copy the instrument assignments from any other song into the current song.

The “import instruments” option is found on the play mode menu, as described in the following section.

Song Copy

You may find it useful to create an empty song or songs, with instruments assigned to the tracks as you prefer, but no scenes or patterns, to be used as a template for new songs.

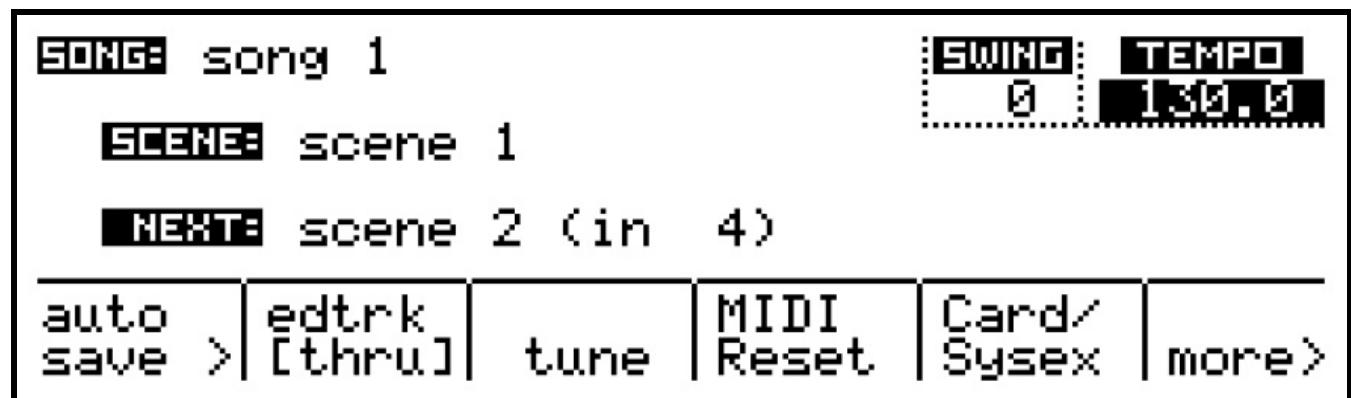
To make a copy of such a template, or indeed any existing song, go to the song list and select the song you wish to copy. Then press the COPY button.

The song name dialogue will appear so you can choose a new name for the copy.

6. Play Mode Menu

The play mode menu is accessed from the SONG, SCENE or TRACK pages, by pressing the MENU key.

The main options appear along the bottom of the display:



Scene Auto Save

This option allows you to enable automatic saving of changes to saved scenes, as described in the earlier section, Using Scenes.

A popup list shows and alters the current setting. This option resets to OFF at power-on.

Edit Track Thru Over-ride

This option allows you to temporarily disable the soft-thru transmission of notes from a master keyboard on the current edit track.

While edit-track thru is disabled, the current edit track selection LED will flash.

Tune

This function is only accessible while Cirklon is stopped.

It will send a MIDI note C 4 on any active track. This is to give owners of unstable analogue synthesizers a fighting chance of keeping them in tune.

MIDI Reset

While Cirklon is stopped, this option will send MIDI all-notes-off messages on all channels, to clear any hanging notes.

This option will also reset some internal state information that is used to minimise the use of MIDI bandwidth.

For example, if the same MIDI CC value is sent repeatedly by a pattern to a MIDI output, the CC message is sent only once. There is usually no point sending the same value that has just been sent.

However, after MIDI reset is pressed, the most-recently-sent state is cleared, so any unchanging values will be re-transmitted.

This can be of use where a stored program change message has been sent to an instrument which has then been manually changed to a different patch.

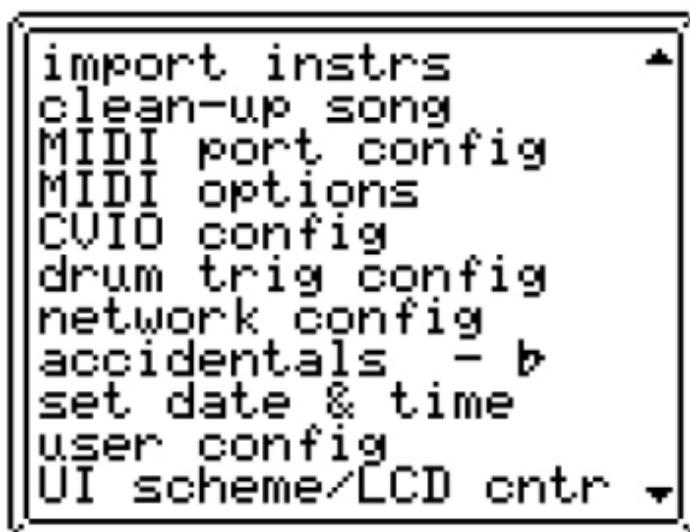
The stored program change message will be re-sent after a MIDI reset is performed.

Card / Sysex

This option gives access to the data saving and loading features, which are described in the later section, “Loading and Saving Data”.

More...

Some less commonly used functions are to be found on the “more>” sub-menu.
Pressing this option brings up a further selection list:



Use the VALUE encoder to select from these options.

“import instrs” allows you to copy the instrument assignments from any other song in memory.

The MIDI port config, MIDI options, CVIO Config and drum trig config pages will be described in detail in the following sections.

Only on Cirklon 2, the network config page is currently for test purposes.

The “accidentals” option allows you to choose whether note names are shown as sharps or flats. This is a song-level setting.

The “set date & time” option allows you to adjust the internal real-time clock, used by Cirklon for time-stamping files saved to SD-card.

Only on Cirklon 2, the “UI scheme” option switches the displayed text between a smoothed colour font, and the original LCD font used on Cirklon 1.

For later revisions of Cirklon 1, the “LCD contrast” option appears last. This allows the display contrast to be adjusted and stored.

User Configuration

The “user config” option gives access to a number of preference options:

User Configuration:

no auto pattern on edit	[off]
name patterns after track	[off]
no pattern after delete	[off]
show MIDI note numbers	[off]
aux edit sets flag	[on]
ck edit, auto clear gang	[off]
ck edit, COPY+DELETE = CUT	[off]
ck edit, RIPPLE bar length	[off]
play fills in edit/record	[on]
no select with SCENE loop	[off]
SCENE advance in edit	[off]
global STOP x 2 resets SONG	[off]
select main ptn on trk chg	[off]
force save menu after edit	[off]
always auto-name save-as-new	[off]
full manual bar follow	[off]
ignore MIDI STOP if master	[off]
treat MIDI CONT as START	[off]
48ppqn DIN SYNC	[off]
Number of TRACKS	[32]
Edit track bank follow	[off]
SHIFT+[trk] mutes active	[off]

Scroll down the list using the VALUE encoder, and press it to toggle each setting. The list extends off the bottom of the display, but is shown here in full.

The effect of each setting is as follows:

no auto pattern on edit - P3 patterns are not created automatically when you attempt to edit the pattern on a track with no pattern assigned.

name patterns after track - by default, patterns are named “pattern nn”, or “cirklon nn”, where nn is an auto-generated number. With this option enabled, the name of the track is included in the name. For example, a P3 pattern may be named “Trk1 P1”, a CK pattern

“Trk1 C1”. If a custom track name has been defined, this will be used in place of the default “Trkn”.

no pattern after delete - normally after deleting a pattern, the previous pattern will be assigned to the track. With this option enabled, the track will have no pattern assigned.

show MIDI note numbers - MIDI note values will be shown as numbers, rather than note names.

aux edit sets flag - when adjusting the numeric value of an aux step in a P3 pattern, inactive steps will be turned on automatically.

ck edit, auto clear gang - in CK pattern edit, the selection of ganged notes is cleared when gang mode is turned off.

ck edit, COPY+DELETE = CUT - in CK pattern edit, the default behaviour is for DELETE to perform a cut operation – selected events are removed to the clipboard. With this option enabled, DELETE will remove and delete notes. The CUT operation is performed by holding COPY and DELETE together.

ck edit, RIPPLE bar length - in CK pattern edit, the use of RIPPLE INSERT and DELETE functions will affect the overall bar length of the pattern.

play fills in edit/record - this option is enabled by default, and causes a track with a fill pattern assigned to switch to the fill pattern during pattern edit or real-time record. It can be turned off if this behaviour is not desired, for example during live performance.

no select with SCENE loop - with this option enabled, setting a loop in scene select view does not cue the loop start scene to play next - you must manually select the next scene.

SCENE advance in edit - by default, Cirklon will prevent any Scene changes while you are in edit mode, to avoid confusion when the pattern being edited is not assigned in the new scene. This option enables you to edit patterns without holding up Scene changes.

Global STOP x 2 resets SONG - when on the SONG page, while stopped, double-pressing the STOP key will return playback to the first SCENE in a song. This option enables that behaviour in all screens.

select main pptrn on trk chg - if you have selected the fill pattern field, switching to another track will return the track page selection to the main pattern field.

force save menu after edit - this option will cause the pattern SAVE menu to appear when you exit pattern edit mode after making any changes to a pattern.

always auto-name save-as-new - when saving a pattern as new in pattern edit, an auto-generated name will be used, and you will not be prompted to confirm it.

full manual bar follow - some pattern edit operations cancel the 'bar follow' setting, to prevent unintended edits of pattern steps as the bar position advances. This option leaves it up to the user to manually turn off bar follow.

ignore MIDI stop if master - when running from its internal tempo clock, Cirklon will normally stop in response to an incoming MIDI STOP message. With this option enabled, it wont.

treat MIDI CONT as START – some PC-based sequencer software sends MIDI continue messages with song position pointer, instead of MIDI start. Setting this option will allow Cirklon to slave correctly to such software.

48ppqn DIN sync - DIN SYNC normally outputs 24 pulse per quarter note, as used by most Roland equipment. This option doubles the output clock rate to 48ppqn, suitable for use with equipment made by Korg and others.

number of TRACKS - this option allows you to vary the maximum number of tracks available. The value can be 16, 32, 48 or 64. When more than 16 tracks are enabled, the track keys and encoders will switch between banks of 16 tracks as described in section 2. Note that lowering this value may affect playback of existing songs which use track numbers above the new limit.

edit track bank follow - when changing the bank of tracks assigned to the track keys and encoders, the current edit track will be changed. This can prevent confusion when the edit track is not visible in the current bank.

SHIFT+[trk] mutes active - holding SHIFT and pressing a track key to "arm" a track for unmuting will also apply to active tracks, muting them at the next global bar loop.

Safe Stop

The Safe Stop feature is intended to help those of us with shaky, or imprecise fingers avoid sudden awkward silences at gigs

To enable the feature, while stopped, hold SHIFT and press the STOP key.

From that point onwards, Cirklon will no longer stop when you touch the STOP key by itself

– you must hold SHIFT and press STOP.

Turn off the feature using the same key combination once stopped – the display will reassure you that Safe Stop is off again.

7.MIDI configuration

Port Configuration

The MIDI port configuration page (accessed via the play mode “more” menu described in the previous section) looks like this:

1	edtrk	off	on/thru	off	off
2	off	off	on/thru	norm1	off
3	thru	on	on/thru	tempo	off
4	3-8	off	on/thru	rsyn1	off
5	5-nn	off	on/thru	rsyn2	off
MIDI port	soft thru	thru FTS	record enable	mclk send	send MTC

All options are edited using the encoders below the display.

Encoder 6 selects the port to edit, with the settings for that port highlighted.

Not all ports can be shown at once, so the display will scroll as required.

The first ports, numbered 1 to 5, are the standard serial MIDI ports on the rear panel.
Next follows the CV port, which represents the internal CVIO option, if present.

Next there are 6 ports assigned to the USB device port, which acts as a class-compliant USB MIDI interface, when connected to a suitable host.

These are labelled “usb1” to “usb6”.

For Cirklon 2, there are further rows for up to 16 MIDI ports on a class-compliant USB MIDI interface, attached to the USB host port. These are labelled “hst1” to “hst16”.

Note that only the number of ports supported by the attached device will be active.

The MIDI port configuration page allows you to choose the soft-thru behaviour and selectively enable real-time recording for each IN port, and to set the MIDI clock and MTC transmission options for each OUT port.

The purpose of each setting is as follows:

soft thru

This option allows incoming MIDI messages (notes, controllers, etc.) to be re-transmitted at one of the MIDI OUT ports, on a different MIDI channel if required.

Pressing encoder 7 will step through five main soft-thru options.

Encoder 7 can also be turned to select port and channel numbers where needed.

The options are:

off - no incoming messages will be re-sent

edtrk - incoming messages will be re-sent to the instrument on the current edit track.

This is primarily for use with a master keyboard, so it can rapidly be routed to any instrument by selecting its track.

thru - incoming messages will pass straight through, on the same MIDI channel, to the same numbered MIDI OUT port. This option might be used with an instrument that has a two-way connection to a Cirklon port and local-control turned off, so that the keyboard always controls its internal sound generation, but with the messages routed through Cirklon for real-time recording.

<p>-nn - incoming messages pass on their original MIDI channel, but to a different MIDI port.

When this option is selected, you must turn encoder 7 to change the destination port number. The label '*<p>*' shown here represents a port number when setting the option.

The text 'nn' is as it appears, and shows that messages remain on their original channel.

This option may be used to re-route incoming messages from one of the USB ports to one of the serial ports, or the CV port. This would allow a connected PC to send messages to instruments attached directly to Cirklon. If you really had to.

<p>-<nn> - this option will pass incoming messages to another port, re-channelised to a single MIDI channel. When this option is selected, turn encoder 7 to step through all possible port and channel numbers.

Caution: setting up MIDI thru routes where external MIDI paths exist between Cirklon inputs and outputs may mean messages can become stuck in an infinite loop, causing buffer overflows.

It's a good idea to think your MIDI configuration through in advance so you don't get any nasty surprises.

thru FTS

This option will cause any notes that are being soft-thru'd on a port to have the current Force-To-Scale setting applied.

record enable

Each port can be enabled for real-time recording.

There are three possible settings – *off*, *on*, and *on/thru*

When real-time record is active, the normal soft-thru settings for a record-enabled port do not apply.

Instead, the *mthru* option on the real-time record menu controls the passing of any messages. To allow soft-thru while recording, ports must have the *on/thru* option set here.

mclk send

This option configures out-going MIDI clock.

MIDI clock can be sent whether Cirklon is acting as tempo master, or slaved to another device.

There are a number of options for clock transmission to support manual pausing and re-synchronisation of slaved devices using the transport keys.

The options are:

off - MIDI clock is not sent on this port

norml - standard MIDI clock is sent.

tempo - only MIDI clock tempo ticks are sent, without START or STOP messages. This is useful for devices such as effects processors with a MIDI clock sync function.

By default, Cirklon sends MIDI clock ticks even when stopped. This allows suitably capable slave devices to lock to the correct tempo before starting.

If you don't wish to send MIDI clock ticks when stopped, this option can be toggled by holding SHIFT and pressing the encoder below <*mclk send*>.

When this option is active, < and > characters will appear around the clock send mode.

A further three clock options, *rsyn1*, *rsyn2* and *rsyn3*, provide various re-synchronisation effects for slaved devices while Cirklon is already running.

When Cirklon is started, ports set to *rsyn1* or *rsyn2* will send a MIDI clock START as normal. With the *rsyn3* option selected, MIDI clock does not start when Cirklon is first started.

While Cirklon is running, you can manually stop the sync output for any port set to *rsyn* by holding SHIFT and pressing RUN.

The effect varies as follows -

rsyn1 - MIDI clock will be stopped just before the end of the current bar, then automatically restart at the beginning of the following bar. This is useful for slaves which have somehow got out of step with Cirklon. For example, if you are using odd-length patterns on a TB303.

rsyn2 - MIDI clock will be stopped immediately when SHIFT + RUN is pressed. A further press of RUN will restart sync at the start of the next bar. This allows, for example, a slaved drum machine to be dropped out, then later re-started in sync.

rsyn3 - MIDI clock will not start when Cirklon is first started. You need to press RUN, or SHIFT + RUN once more after starting, to start slaved devices. Further presses of RUN / SHIFT + RUN behave as for *rsyn1*

For a slave that can be manually stopped, pressing RUN allows it to be re-started in sync with Cirklon. This is especially useful for machines like the Roland TR909, which must be stopped to switch between play and edit modes. You can manually stop the 909 with its STOP key, switch between play/edit, then re-start it in sync from Cirklon without having to restart everything.

MIDI Clock Reception

There is no setting to enable MIDI clock reception for a MIDI input port, as it is handled automatically.

If Cirklon is stopped, and a MIDI clock START message is received at any input port, Cirklon will switch to MIDI clock slave mode and start running in time with the incoming tempo.

MTC – MIDI Time-code

MIDI time code is a linear time reference often used by audio recorders and computer-based sequencer applications.

It has some advantages, and some disadvantages compared to MIDI clock.

MTC does not carry any tempo information. To use MTC, the tempo settings on master and slave devices must be set to the same value manually.

Cirklon can send MIDI time-code on any port, and will also respond to incoming MTC by starting, and running at its internally configured tempo.
Encoder 11 is pressed to toggle the transmission of MTC for each port.

MTC supports a number of different time-code frame rates.

The frame rate for transmission is selected on the MIDI options page described below.

MIDI Options

Some further global MIDI settings can be found on the MIDI options page, accessed via the play mode “more” menu.



Latency matching delay

In a situation where Cirklon is sequencing both hardware instruments connected via the hardware MIDI, CVIO or drum trigger ports, and virtual instruments running on a PC or Mac connected via the USB MIDI port, the inevitable audio latency of the virtual instruments will cause their audio output to be delayed relative to the proper instruments.

To compensate for this, the *Latency matching delay* setting allows you to set independent output delay times for every port, in increments of 0.25ms, up to a limit of 64ms.
Encoder 7 is used to adjust the delay time.

When the **port** setting is set to *all*, a global delay value will be applied to all output.

Encoder 6 may be used to select a specific port, which can then have its delay value adjusted independently of the others. If one or more ports has a different value from the others, the delay value will be shown as “not =” when port is “all”.

Remote Control MIDI Input

With remote control MIDI input enabled, notes played on a MIDI keyboard will control the current scene transpose value, and program change messages will select one of the first 128 scenes in the current song.

Turn encoder 9 anti-clockwise from the *off* position to selection the input port, then use encoder 10 to set the desired MIDI channel for the remote control source.

MTC FPS

This setting allows you to choose the frame rate for transmission of MIDI time-code, where it is enabled for one or more of the MIDI output ports.

All ports must use the same frame rate.

The frame-rate setting is included in the MTC messages, so incoming frame-rate can be detected automatically.

The out-going frame rate will be changed automatically to match the frame rate received from an external source.

8.CVIO and Drum Trig Configuration

The optional CVIO expansion gives you 16 high-resolution control voltage and 8 gate outputs via a 25-pin “analogue bus” connector on the rear panel.

A further choice of external break-out box (BOB) connected via a multicore cable gives access to the outputs on 3.5mm or quarter-inch sockets, without concentrating a mass of cables behind the sequencer itself.

The same pattern types used for MIDI instruments are also used for controlling CV-attached instruments.

Note, controller and pitch-bend messages routed to a channel of the “CV” MIDI port control the CV and gate outputs directly through a high-speed internal connection.

The 16 channel range supports independent control of a number of instruments by grouping their CV and gate outputs on a unique channel.

Each CV output can generate a voltage proportional to the note, velocity or controller values, or a variable combination of all three.

You can think of the CVIO as a MIDI to CV convertor built into Cirklon, without the bandwidth limit of a serial MIDI connection.

The drum trigger (or drum mux) module is a further option that provides multiple trigger pulse and CV outputs for driving analogue percussion synths or drum machines.

Cirklon’s SYNC output is used in an alternate mode to transmit drum trigger data.

The drum mux module receives and outputs the triggers to 16 pairs of 3.5mm sockets. It also features a DIN SYNC output, so the original SYNC capability can be retained.

Multiple drum trigger modules can be attached in parallel, up to a maximum of 48 separate trigger outputs.

As with the CVIO, the trigger output is driven by note output to a chosen channel on the CV port.

CVIO Config

Setting up the CV and gate outputs for use with an instrument is a two-step process.

The first step is to create an instrument definition to route the output of a track to a channel on the CV port.

The second step is to configure one or more CV outputs to respond to messages on the chosen channel for the desired control of pitch, gate, etc.

Let's work through a basic example –

Synth1 has pitch CV, gate and filter CV inputs.

The pitch CV input is connected to CV output 1, the gate input to Gate output 1, and the filter CV to CV output 2.

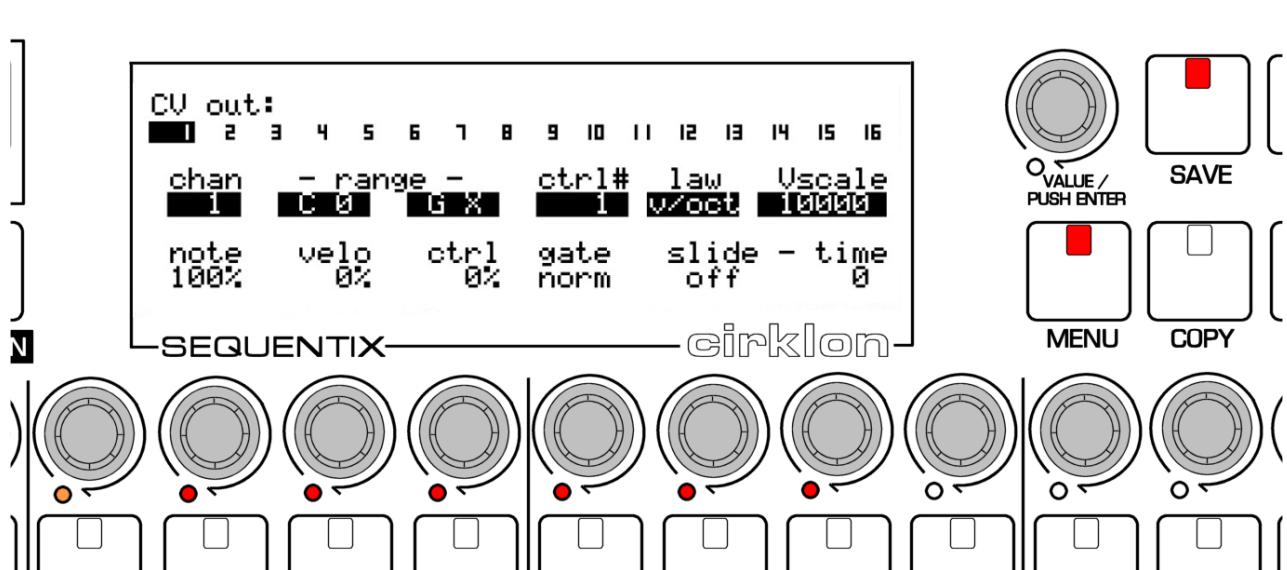
First, you must create an instrument definition, as described in the earlier section "Creating and Assigning an Instrument".

Name the instrument "synth1", set the MIDI Port to "CV", and MIDI channel to 1.

With the instrument assigned to a track, and a pattern playing, MIDI messages will now be sent to the CVIO on channel 1.

The next step is to configure the CV outputs to respond to those messages.

This is done from the CVIO Config page:



At the top of this display, the numbers 1 to 16 show which CV output has been selected for configuration. As shown, output number 1 is selected.

The other outputs can be selected by turning the VALUE encoder.

CV Output Settings

There are two rows of six settings for each CV output.

The row is selected by turning the ROW encoder (or encoder 5).

All values in the selected row are highlighted, and the six encoders below the display used to adjust the setting above.

Starting at the lower row, the first three values are **note**, **velo** and **ctrl**.

With these controls, you can mix three different MIDI values into the single CV output at the same time.

If you want to use an output purely for pitch CV control, as in our example synth1, the **note** value should be set to 100%, and both **velo** and **ctrl** set to 0%.

For simple velocity or controller to CV outputs, set either **velo** or **ctrl** to 100%, and the others to 0%.

An example of where you might want to mix two different values would be for a synth with a CV input for filter cutoff.

Mixing some of the note value with velocity or a controller will give you variable filter keyboard tracking and simultaneous filter modulation.

By default, the gate outputs follow the CV output of the same number.

That is, gate output 1 will turn on when a note is being played on CV output 1, and turn off when the note is released.

If over-lapping notes are sent to the CV output, i.e. a new note starts before the previous note ends, the most recent note will take priority.

If the new note is released while a previous note is still held, the CV output will return to the previous note CV.

The **gate** setting controls how the gate output behaves when over-lapping notes are played.

Set to *norm*, the gate output will remain high while any note is being played.

Set to *retrig*, the gate output will briefly turn off then on again whenever a new note is played, to re-trigger any envelopes.

The **slide** settings give a choice of smooth transitions from the CV for one note to the next, also known as glide or portamento.

There are two slide settings, for mode and time.

There are five choices for slide mode:

off - no slide

a-lin - automatic, linear mode. Auto-slide modes only activate the slide when notes sent to the CVIO overlap. This can be achieved by playing legato, setting the length of one note in a pattern so it is held beyond the start of the next note, or by using the P3 pattern TIE flag.

The linear part means that the change from one CV to the next happens in a straight line, also known as 'fixed rate' slide. The time value determines the time taken to slide over a fixed range of notes. It will take 12 times as long to slide over a full octave as it does to slide over one semitone.

a-exp - automatic, exponential mode. Exponential slide is also known as ‘fixed time’ slide. The shape of the transition from one CV value to the next follows an exponential decay curve. As the difference between the two notes increases, the initial rate of slide will increase, with the result that the time taken to slide between two notes is roughly the same, regardless of the interval between them.

The remaining *lin* and *exp* options are as above, but slide is always active.

Moving on to the top row of values:

The first setting is **chan** – for MIDI channel.

This sets the MIDI channel number that this output will respond to.

For our example “synth1”, this should be set to channel 1.

The **range** setting defines the range of MIDI notes that this output will respond to when used for note or velocity output.

The range shown here is from C0 to GX, where ‘X’ is short for octave 10.

This represents the absolute maximum range of CV output possible – 10 and a half octaves, giving just over 10.5 volts for a 1 volt/octave synth.

The lower note limit for the CV output determines the note which will correspond to minimum output voltage.

The appropriate range will vary widely for different synths that may be used, so a bit of trial and error may be required.

The next setting is **ctrl#**, for controller number.

This is the MIDI CC number whose value will be mixed into the CV according to the **ctrl** amount on the lower row.

Alternatively, turning one step before CC #0 will select **bend**. Pitch-bend will be added to the CV output according to the amount value. The amount value will be shown in semitones, rather than percentage depth.

The **law** setting gives a choice of:

- | | |
|--------------|---|
| <i>V/oct</i> | - volts per octave (Moog / Roland type) |
| <i>Hz/V</i> | - Hertz per volt (Korg / Yamaha type) |
| <i>gate</i> | - allows the use of a CV output as a gate output |
| <i>env</i> | - outputs a fully user-controllable ADSR envelope |

As the CV outputs are capable of outputting a higher voltage than the gate outputs, the **gate** option can be used to drive synths which need a gate voltage higher than 5 volts.

Envelope Generation

The envelope option allows any CV output to be used as an ADSR envelope generator. When the *env* law is selected, some of the settings on the CV config page will change to appear like this:

CV out:															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
chan 1	-	range C 3	-	G X	ctrl# 10	law Env	gate retrg								
att 0	dec 50	sus 63	rel 50	amt 127	velo 0										

The lower row now has values for **attack time**, **decay time**, **sustain level**, **release time**, **envelope amount (amt)** and **velocity sensitivity**.

The amount control sets a fixed level of envelope depth. The velocity control adds a variable amount of envelope depth according to the velocity of each note.

Setting amount to 127 and velocity to 0 gives maximum envelope depth, with no velocity sensitivity. Setting amount to 0 and velocity to 127 gives maximum velocity control.

On the top row, the **channel** and **range** settings are used in the same way as for CV or gate output - they determine the channel and note range that will trigger the envelope.

The **gate** option determines whether the envelope generator will re-trigger with over-lapping notes.

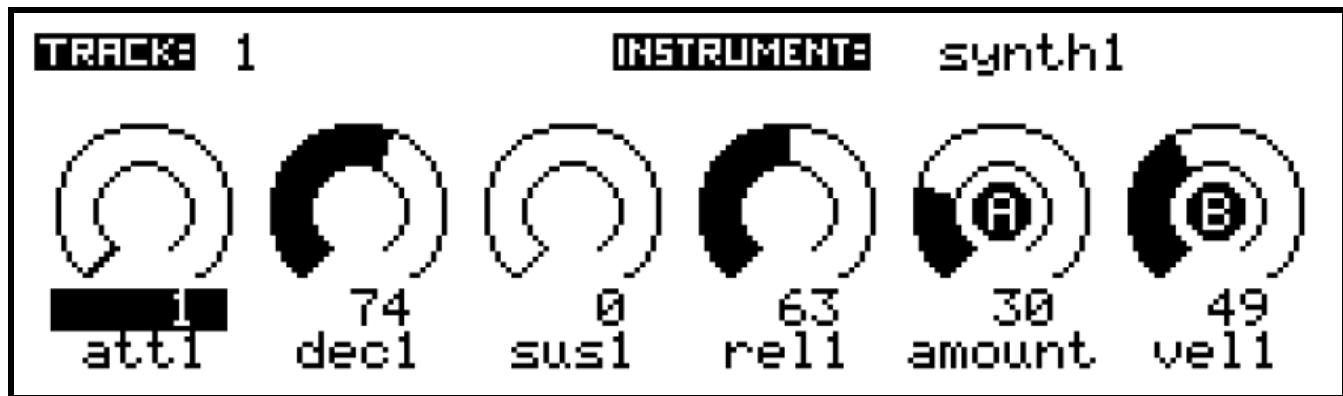
The **ctrl#** value allows you to define a group of consecutive CCs to adjust the envelope parameters from track control values, patterns or an external controller.

The base CC number set here maps to the attack control, with the next 5 CC numbers mapped to decay, sustain, release, amount and velocity respectively.

By using different CC numbers for envelope generators on CV outputs assigned to the same channel, multiple envelopes can be generated for a single instrument.

It is very useful to assign track control values for the envelope CCs at the instrument level.

An example of such a mapping is shown here:



By labelling the CC track values, the track values page will give quick access to view or edit the current settings for any envelope outputs for an instrument.

Refer to chapter 9 – Track Values, for full details of how to set up these controls.

CV Calibration

The **Vscale** setting is a soft adjustment of the voltage scaling of the CV outputs.

This value is shown on the main CV config page when the V/oct law is selected.

If you switch the law to Hz/V, Vscale is replaced by the **Offset** value.

Although many synths may use 1 volt per octave in theory, variations in calibration of test equipment mean that often there are slight differences in scaling between different synths.

The Vscale and Offset settings allow you to adjust each CV output individually to match the calibration of the connected synth. This saves having to laboriously range and scale each new synth you use against a common reference.

The CV output by each port is calculated as:

$$V_{out} = CV * (Vscale / 10000) + Offset$$

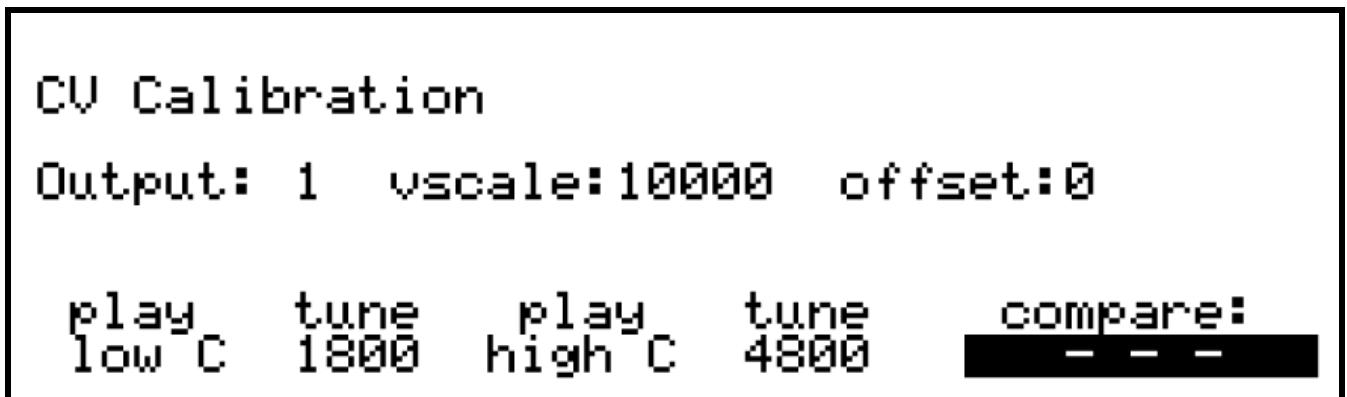
For a Volts-per-octave synth, the Vscale value alters the octave spacing, and the Offset value gives an absolute change in pitch.

For a Hertz-per-volt synth, where the CV output follows an exponential curve to control frequency, the effect of the two values is swapped – Vscale gives an absolute tuning adjustment and Offset adjusts the octave spacing.

But you don't need to worry about the maths, there is an extra page where the correct calibration values can be found by matching the tuning of two notes against a reference MIDI-connected synth, or by using an external tuner.

In the CVIO config view, you can manually adjust the value by turning the encoder, but this is slow, and not easily done by ear.

If you press the encoder, you will enter the CV Calibration page:



On the calibration page, the encoders below the display have the following functions:

encoder 6 – press to play a low C reference note. Repeated presses will toggle the note on and off. The 'low C' label will be highlighted while the note is active.

encoder 7 – turn to adjust the tuning of the low reference note.

This value is used to calculate the shift in absolute tuning needed to bring the synth in tune. Only the minimum adjustment needed to get to the correct pitch should be used – if you need to move an octave or more, you should adjust the octave setting of the synth first to get as close as possible to the starting reference.

The tune value shown is a 4 digit, hexadecimal number. You don't need to worry what this means, just adjust the tuning as required.

(the value is the 16-bit digital code being sent to the CVIO digital-to-analogue convertor to give the current output voltage).

encoder 8 – press to play a high C reference note, **two octaves above** the low note. Repeated presses will toggle this note on and off.

encoder 9 – turn to adjust the tuning of the high reference note. This value is used to calculate the correct octave scaling.

encoder 11 – press this encoder to choose a MIDI-attached instrument on which to play the reference low and high C notes. Note that an instrument on the CVIO interface cannot be used.

As you adjust the tuning of the reference notes using encoders 7 and 9, the difference between the two tune values is used to calculate the correct Vscale and Offset values for the CV output to give accurate scaling, taking into account the V/oct or Hz/v mode of the output.

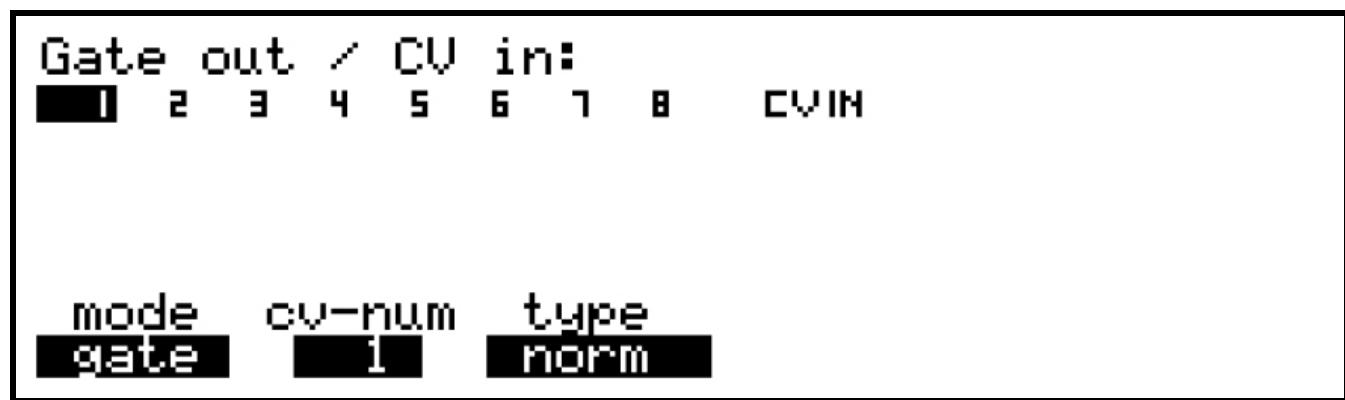
You shouldn't need to worry about the technicalities of this too much, as long as you can get your synths to play in tune.

When both reference notes are in tune, press the MENU key to return to the main CVIO Config page, with the new values for the output set.

Gate Port Settings

To access the settings for the gate ports, turn the VALUE encoder clockwise beyond CV output 16.

The display should look like this:



Selection of the gate port is made with VALUE as before, and the settings adjusted using the usual encoders under the display.

The **mode** setting gives a choice of *gate* or *clock*.

The **type** option sets the electrical behaviour of the output, giving the choice of:

<i>norm</i>	5 volt output when the gate is active, 0 volts when off
<i>S-trig</i>	"short-to-ground" (Moog-type) gate output
<i>inverse</i>	this gives the opposite polarity to norm, for Yamaha's CS40M and CS15D

Gate Mode

Gate outputs will usually be linked to the CV output being used to control the pitch of the attached synth. This is done using the **cv-num** value, which has a range of 1 to 16.

For example, if you have set up CV output 1 to control the pitch of a synth in response to notes played, setting gate output 1 to gate mode with cv-num set to 1, gate output 1 will turn on when a note is played that matches the channel and range setting on CV output 1.

It is also possible to set a gate output to work independently of the CV outs, responding to any notes played on a chosen MIDI channel.

To do this, adjust the value for **cv-num** beyond 16 – the label will switch to **chan**, with channel numbers 1 to 16 now available for selection.

Clock Mode

In *clock* mode, the gate outputs can be used to generate a clock pulse based on a division of the tempo clock, or a DIN-sync type run or reset signal.

When clock mode is selected the cv-num value is replaced by *ppqn* (pulses per quarter note). This sets the clock division for the pulse output between 48 and 1.

Two further values of *run* and *rst* give the control signals needed to start or reset an attached sequencer or drum machine.

Run gives a constant “on” signal when the sequencer is running.

Rst gives a short pulse when playback is started.

Note that the *mclk send* settings for the CV port **must be configured** on the MIDI port config page for clock outputs to work.

CV Input

The CVIO processor features a built-in analogue to digital convertor.

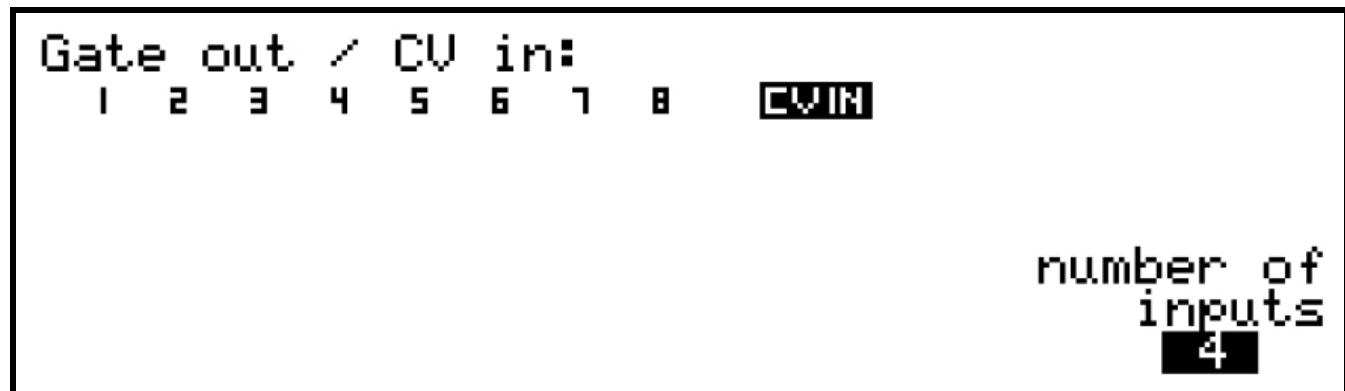
Up to 4 of the gate outputs can be re-configured as analogue inputs.

The voltage at each input, which can range from 0 to 5 volts, will then be converted to the value of a MIDI CC message, which appears as the input from the CV port.

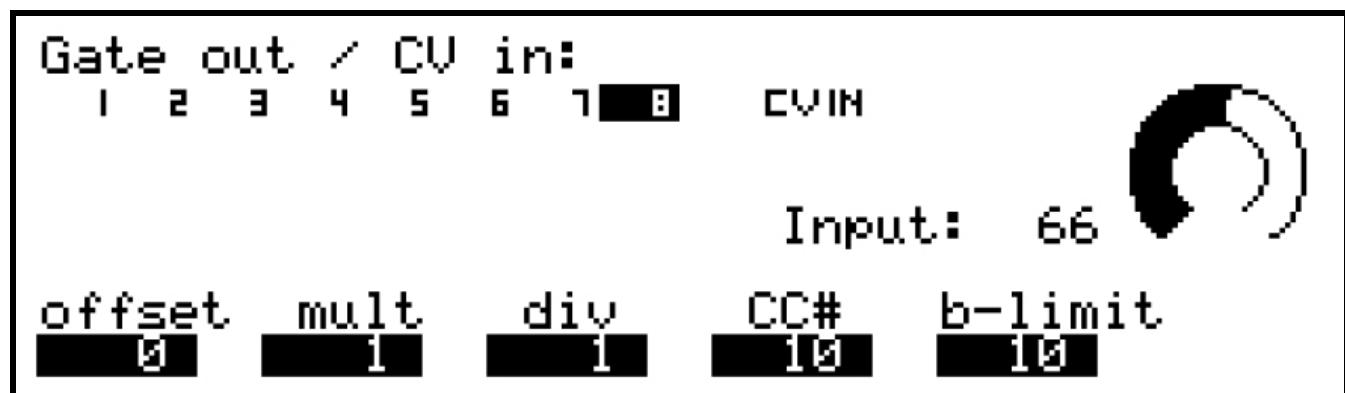
Using the MIDI port configuration, CCs can then be soft-thru routed to another output or captured for real-time recording.

The inputs are enabled starting from gate 8 down.

The number of gates to configure as inputs is chosen by scrolling to the CV IN option:



Once you have enabled some gates as input pins using encoder 11, the page for that gate connection will show a different set of values.



The **CC#** field selects which CC number will be sent from this input.

The **b-limit** value sets the minimum time, in milliseconds, before a new CC will be sent after a change of input voltage. This applies a bandwidth limit to the MIDI output.

Rapidly changing voltages can produce a stream of CCs that will quickly saturate a MIDI bus, so adjust this parameter to the largest value that is responsive enough.

The **offset**, **mult** and **div** values allow you to arithmetically adjust how the input voltage is converted to a CC value.

For example, if a voltage input doesn't cover the full range of CC values, the **mult** (multiply) and **div** (divide) values allow you to scale the CC values to fill the range.

Offset allows you to lower the range of mapped CC values, in case the input voltage doesn't reach zero.

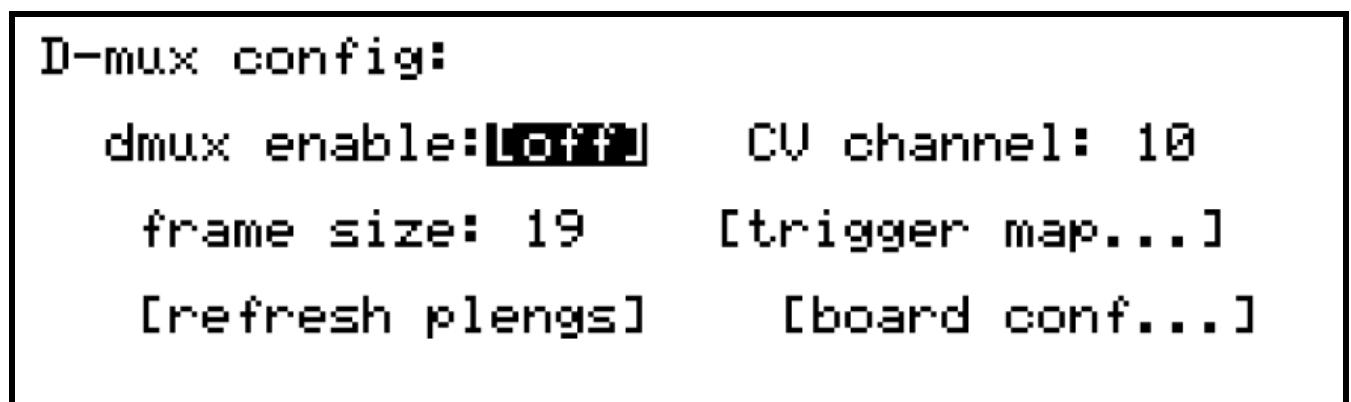
Drum Trig Config

The drum trig config page is used to configure settings for the optional drum mux module.

“Mux” is short for multiplex, which refers to the transmission method used. The SYNC port can switch to an alternate mode where bursts of multiplexed trigger pulses and accent CVs can be transmitted to one or more drum mux modules. Each module can output up to 16 pairs of trigger pulses and accent CVs, plus three further trigger pulses for DIN SYNC output (to replicate the function of the re-purposed SYNC socket).

The principle advantage of triggering percussion sounds using the drum mux output, as opposed to connecting them via a MIDI-to-trigger interface, is that the drum mux output has a much higher transmission speed than MIDI, and uses frame-based synchronisation to ensure every trigger pulse is tightly synchronised with all other triggers on the same tick. This gives a clearly audible advantage over MIDI when as few as 4 or 5 sounds are being triggered at the same instant.

The drum trig config page looks like this:



Select the **dmux enable** option to switch the SYNC socket to dmux mode.

Once the dmux output is active, normal DIN SYNC devices will no longer respond to the SYNC output directly.

The **CV channel** option selects which channel on the CV port the drum triggers will respond to. The selected channel will no longer route note messages to the CVIO expansion, so make sure to use a channel that is not already assigned there.

The **frame size** sets the total number of triggers to be transmitted in each burst.

The drum mux module has 16 accent/trigger output pairs, with a further 3 trigger channels used for the DIN sync output, so the default frame size is 19.

The frame size will only need to be changed if additional modules are connected, in which case you will also need to change at least one setting on the extra modules.
This is described in more detail in the Board Level Config section.

Drum Trigger Map

The **trigger map...** option takes you to a sub-page like this:

Output	Note	velo	pulse	(note2)	acc
> 1 <	C 4	1	32	---	pls
2	D 4	1	32	---	pls
3	E 4	1	32	---	pls
4	F 4	1	32	---	pls
5	G 4	1	32	---	pls
6	A 4	1	32	---	pls
7	B 4	1	32	---	pls

For each trigger and accent output pair, you can select which **note** will trigger it, choose a **velocity** response curve, adjust the trigger **pulse** length, and set an optional second trigger note.

Select the output to configure using encoder 6, then edit the values with the encoders below each column.

Any changes made on the trigger map page apply immediately.
To return to the main drum trig config page, press the **MENU** key.

There are currently four **velocity** map options.

Option 1 has a higher minimum voltage, rising linearly to the maximum voltage output.
Option 2 has a lower minimum voltage rising to a lower maximum.
Option 3 is similar to option 1, with a slightly reduced maximum voltage, to suit the lower possible voltage range with a Euro-rack 12V supply.
Option 4 is similar to option 2, with a maximum voltage of 5V.

Select the curve which best suits the attached drum sound when triggered over the full range of velocity.

The **pulse** length is specified in sixteenths of a millisecond (0.0625ms steps).
The default pulse length of 32 is equal to 2ms.

The maximum length of 240 gives a 15ms pulse.

Turning the length value as if to select zero will select “gate” mode, where the trigger output will act as a normal gate, with the pulse length controlled by note length.

For D-mux version 3 boards on, there is an extra setting to enable pulsed accent.

Normally the accent CV outputs jump to each new velocity level just before the trigger, and hold that level until the next note trigger.

With the **acc** value set to “pls”, the accent CV output will pulse on then off (0V) with the pulse length controlled by the pulse length setting .

This is useful for drum machines or percussion modules that need a single trigger input for each sound, with velocity sensitivity according to the pulse voltage.

Although the pulse lengths and pulsed accent settings are stored internally in Cirklon’s memory, these settings are also stored in the drum mux module itself.

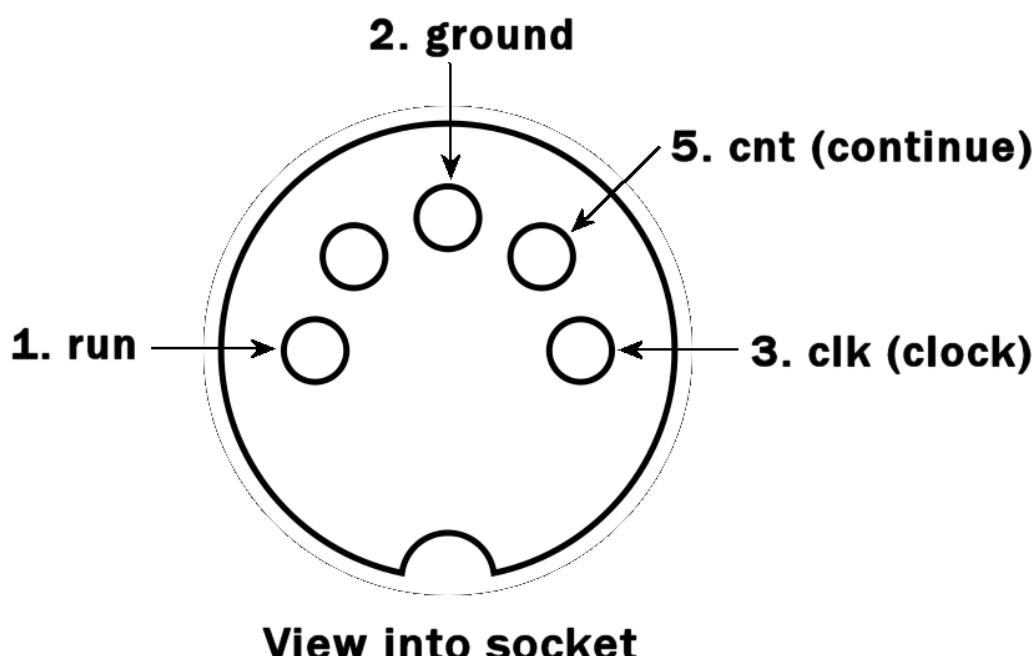
As you adjust the pulse length values, a message is sent to the drum mux module to update its internal setting.

This may briefly disturb the transmission of the normal trigger pulse bursts.

DIN SYNC Output

Scrolling to the end of the default trigger map settings, you’ll see outputs 17, 18 and 19 mapped to special non-note values of *D-run*, *D-cnt* and *D-clk*.

These last three outputs are the signal pins of the SYNC OUT port on the dmux module:



With their default mapping to the *D-* values, the sync output will transmit DIN SYNC following the MIDI clock on the CV port.

Make sure **mclk send** is enabled for the CV port in MIDI port config.

If you don't need DIN SYNC, these three outputs may be used as extra trigger outputs when mapped to notes, however they will not respond to velocity.

The *D-* options are found at the end of the note range, after G X (G 10)

Refresh Pulse Lengths

Although any new pulse length or accent setting for an output is sent to the drum mux module as it is changed, if any settings are changed in Cirklon when the drum mux module is not connected or powered up, or a different drum mux configuration is loaded from card or sysex, the settings may become mismatched.

The **refresh plengs** function will force an update of all stored values according to the current settings held in Cirklon's memory to ensure that all settings in the module are as shown in the menu.

It is strongly recommended that you run this function after loading a different drum mux configuration.

Board Config

The board config page is used to configure the unique settings for each drum mux module in a multiple module setup.

The options here send configuration commands that all connected modules will respond to, so it is important to only connect the unit to be configured while using this page:

```
Board Level Config:  
frame start pos: 1 trig count: 16  
DIN aux enable:[on ] [send to board]
```

Note that the drum mux output is purely one-way - there is no way for the current settings to be read back from a module.

The most recently used settings will always appear in the config page, even if you have connected a different module.

In order to use multiple modules, the “frame size” (on the main drum trig config page) must first be increased to match the total number of trigger outputs, up to a maximum of 48.

Each drum mux module must then have its **frame start pos** value set to determine the unique range of triggers it will respond to.

Let’s look at adding a second drum mux module, where the first module has the default **frame start pos = 1, trig count = 16** and DIN output enabled.

The default frame size of 19 is made up of the 16 accent/trigger pairs plus the 3 triggers used for the DIN output on the first module.

The **frame start pos** for the second board should therefore be 20.

You would set the value to 20 on the board config page, then use the **[send to board]** option to update its configuration.

You may choose to use fewer than the maximum 16 of the trigger outputs using the **trig count** option.

You can choose whether to output DIN SYNC from the second module by setting the **DIN aux enable** option.

This option must be enabled to use the DIN output either for DIN SYNC, or for extra note-triggered outputs.

The output numbers for the 3 pins of the DIN output will always follow the last normal trigger output set with the **trig count** value.

For example, if trig count is set to 8 with DIN aux enable on, the three DIN pins will be triggers 9, 10 and 11 for that module – the 3.5mm trigger and accent outputs 9 to 16 will be inactive.

Assuming both modules will use all 16 outputs and send DIN SYNC, the total frame size should be set to 38. The first module will respond to trigger numbers 1 to 19, the second to trigger numbers 20 to 38.

The additional trigger settings will become accessible on the trigger map page as soon as the frame size is increased.

Once configured with the correct frame start pos, each board will respond to pulse length adjustments according to its own trigger number range. The pulse length values are edited with all modules connected.

9. Track values

Track values are user-configurable controls with a variety of uses.

Although you add and use them entirely on the TRACK page, they are stored as part of each instrument definition. Once a track control has been added to an instrument, it will appear on any track to which that instrument has been assigned.

There are two types of track values:

Track MIDI values let you control parameters on your instruments by sending MIDI controller messages.

Track control values give access to internal settings that dynamically alter pattern playback.

Adding a track MIDI value

A track MIDI value can send any MIDI CC or a program change message.

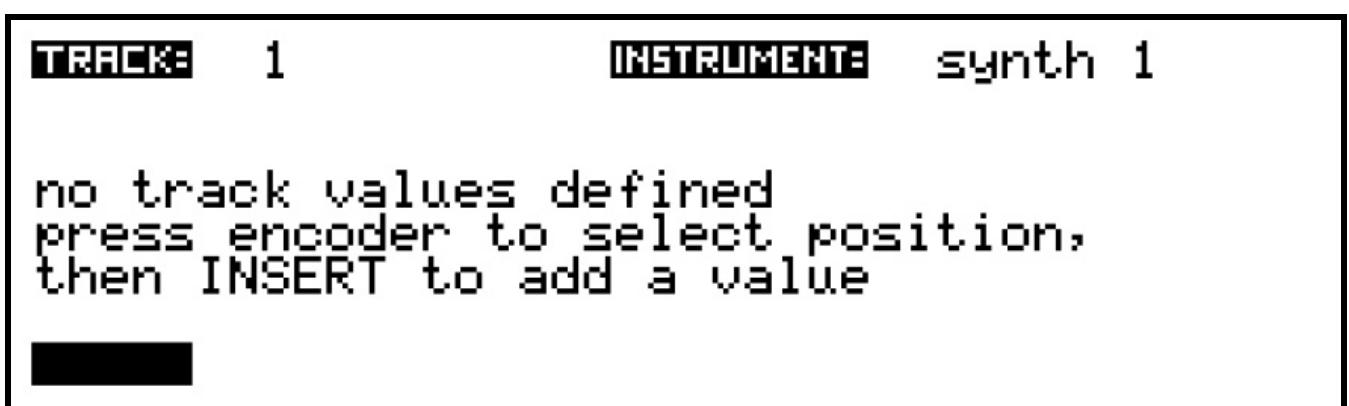
They can be used for real-time control, and can also be saved for the current SONG, or SCENE.

Stored values will be sent when a song is first recalled, or at the start of chosen scenes.

The track values appear on a sub-page of the TRACK page, called (unsurprisingly) the track values page.

You access this page by pressing TRACK when you are already on the TRACK page.

Before you have added any track values for the instrument assigned to a track, the track values page will look like this:

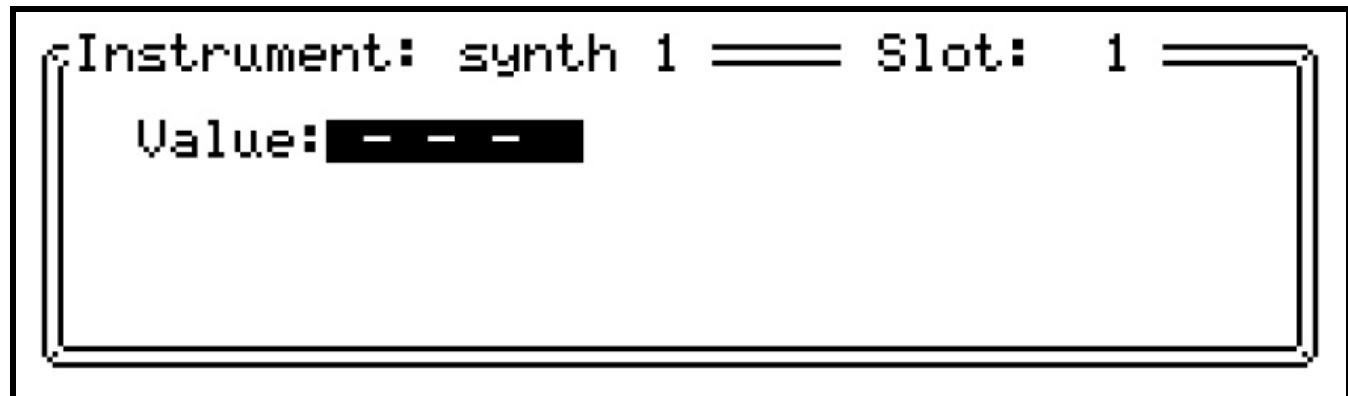


The display is divided into six ‘slots’ that can each hold a track value. The encoders under the display are used to edit the actual values while on this page, and the six encoder LEDs light red as a reminder of this.

The selection highlight shows the current slot position – this is where any new value will be added. The highlight above is on slot 1.

The INSERT key is used to add a new value (or to change or remove an existing value).

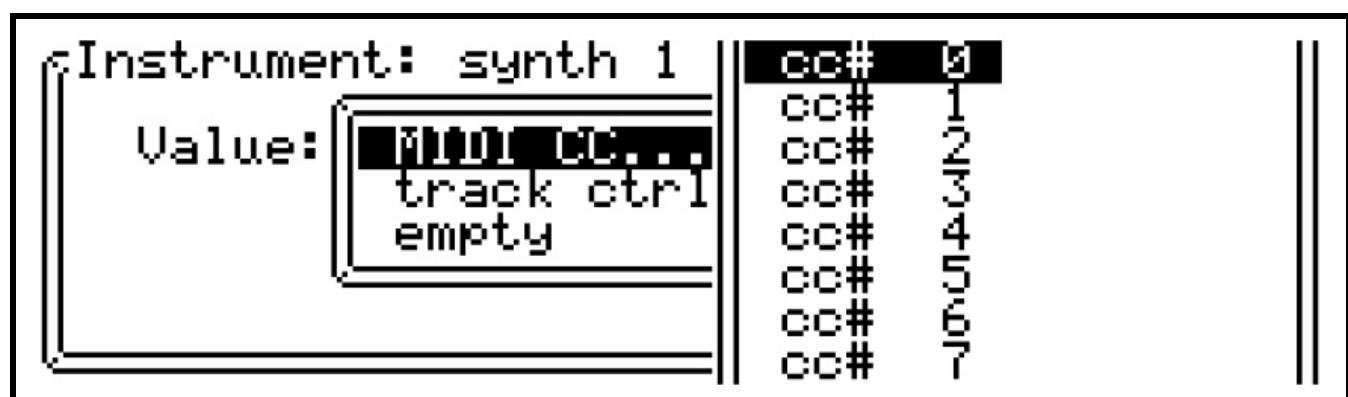
Let's add a track MIDI value by pressing INSERT. This page should appear:



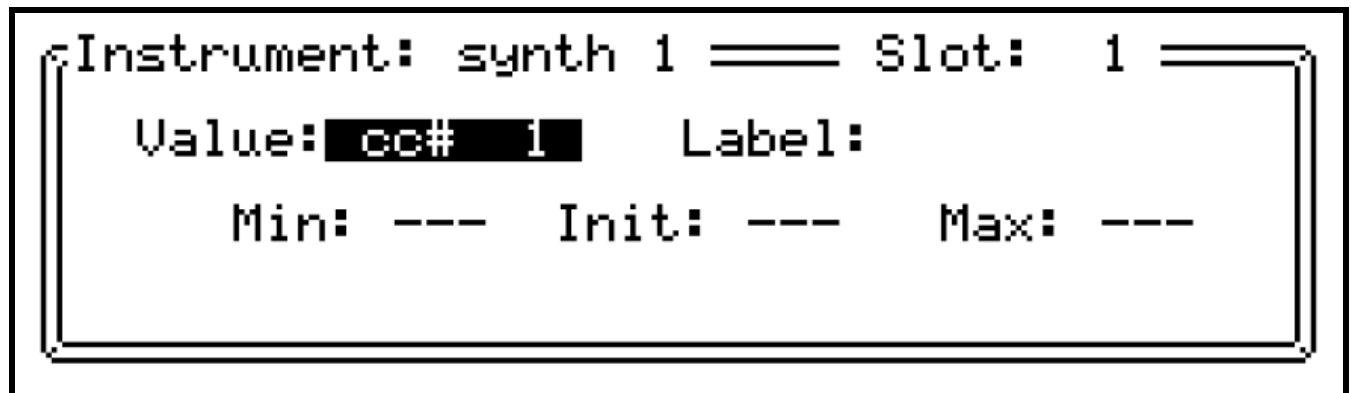
The top line shows the track value will be added to the instrument "synth1", in slot 1.
Press ENTER to open the select list for the value type:



Then press ENTER again to pick a CC number:



Use VALUE to scroll one step to “cc# 1” and press to confirm.
The CC select list will close, leaving a new “label” field visible.



Ignore this for now, and press SAVE to return to the TRACK values page:



The track value now appears in slot 1, with dashes showing that no value has yet been set. The text “modwh” is a preset label for CC #1, as this CC is most commonly used for modulation wheel control, as defined by the MIDI standard. There are preset labels for some other CCs – “vol”, “pan”, “bankM” and “bankL” will appear for CCs 7, 10, 0 and 32 respectively.

To send a CC #1 / modwheel message, turn encoder 6.

An initial value will appear on the display, as well as being sent to the instrument:



Keep turning the encoder, and each new value will be sent as a CC message to the instrument. You now have real-time control of this parameter.

In case you need to re-send the current value to an instrument, perhaps because a manual change was made on a synth, quickly double-pressing the encoder will force a re-send.

Let's add a second track MIDI value.

Press encoder number 7, to select the second slot on the display.

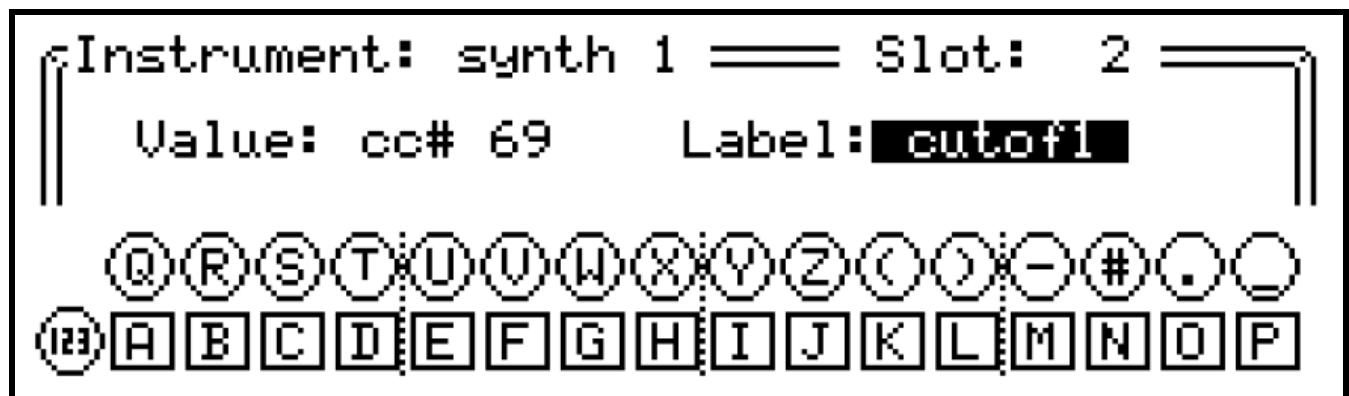
Then press INSERT, choose the MIDI CC option again, and select "cc #69".

Press SAVE to return to the track values, and you'll see this CC has no preset label:



Since "cc# 69" is a bit unhelpful, and anyway "synth1" maps this CC to the cutoff frequency of filter 1, let's give it a custom label.

Press INSERT to edit the value. Then turn VALUE to move the selection to the label field, and press:

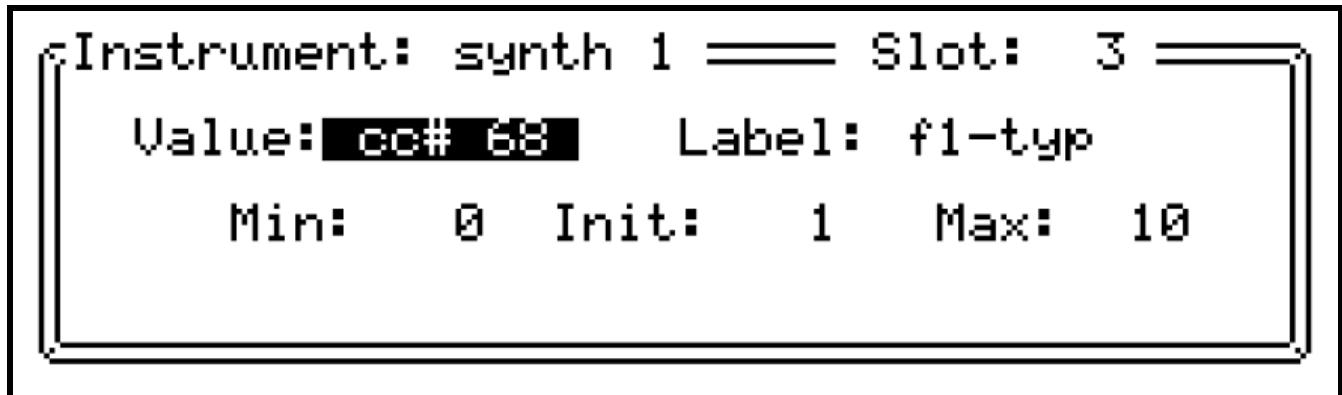


The usual text input mode is used to enter label text with the step keys and encoders. Enter the letters "cutoff1".

Press ENTER to confirm the text, then SAVE to return to the track value page, where your new label should appear:



You may continue to add further track values for any other parameters you want to control on an instrument.



Some of your CCs might not be utilised over their full 7-bit range. For example, the synth we're calling synth 1 offers different filter types, selected with CC number 68.

The range is only 0-10 so that's what we'll use for the Min and Max values.

Init is a field you'd ordinarily use for a middle or initial value (e.g. if this were a pan control, you might like the initial position to be central – 63). In our example, Init isn't meaningful but as the first filter type of this synth is Bypass, I added a value of '1'. It could just as easily have been left blank.

If you have filled the first 6 slots, or just want to put some values in a separate row, each instrument has space for 30 rows of values, accessed by turning the ROW encoder.

However, only rows which have values present are normally accessible.

To move beyond the last populated row, hold SHIFT while turning the ROW encoder one step clockwise. This will take you to an empty row, where you can INSERT new values. Once the first value is inserted on a row, that row will be accessible without using SHIFT.

Assigning Knobs A & B

To provide smooth, absolute control over track values, it is possible to assign the front panel knobs A & B to any two of the track values.

These assignments will be remembered for each track, in each song.

Whenever you are on one of the play pages (SONG, SCENE or TRACK), the knobs assigned for the current edit track will control their values directly, even though the track value page is not visible.

To assign a knob, hold SHIFT then press the encoder under the track value to which you want to assign the knob.

The display will show this message:



Simply move knob A or B, and the assignment is done.

You can see that the knob has been assigned from the A or B label that now appears in the centre of the value indicator:



You can continue to use the encoder to change the value after a knob is assigned.

But if you do, the knob will enter “pass-through” mode.

This means that moving the knob will have no effect until you pass through the current value set using the encoder. An indicator line will appear to show the (inactive) knob position until the current value is passed.

To de-assign the knob, hold SHIFT and press the encoder a second time. A message will tell you the knob has been de-assigned and the central label will disappear too.



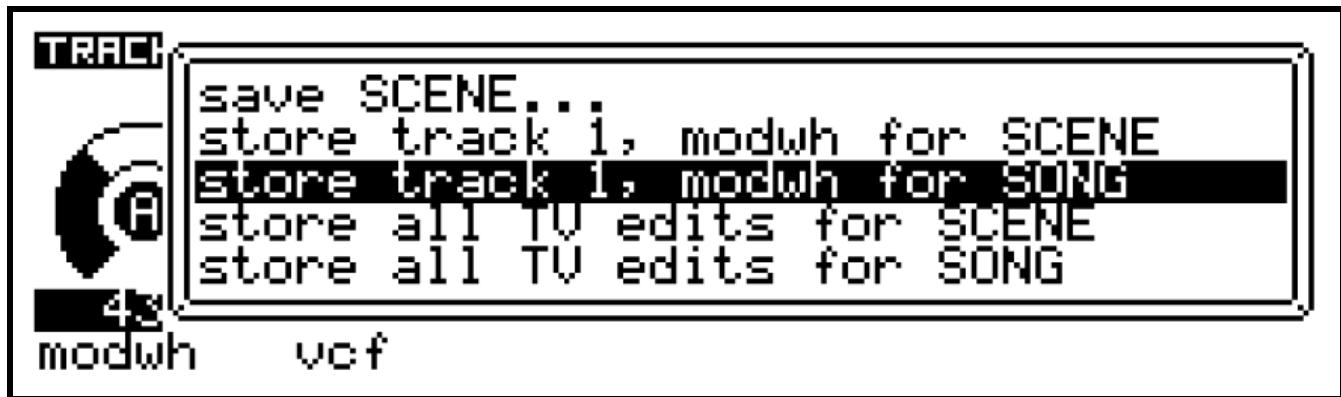
Saving Track Values

Track values can be saved at the song or scene level, to allow re-call of chosen settings at the start of the song, or any of the scenes.

To save a current value on the track values page, press the SAVE key.

Normally on the TRACK page, SAVE is used to save the current scene.

On the track values page, SAVE brings up a list of five options:



Use the VALUE encoder to select the option you want. The first, "save SCENE..." gives access to the normal scene save function.

The next two options show the current edit track number and track value name or label, and let you store the value for the SONG or SCENE.

The final two options let you store **all edited track values across all tracks**, at the SONG or SCENE level.

Values saved at the SONG level are stored permanently as soon as you select a SONG option.

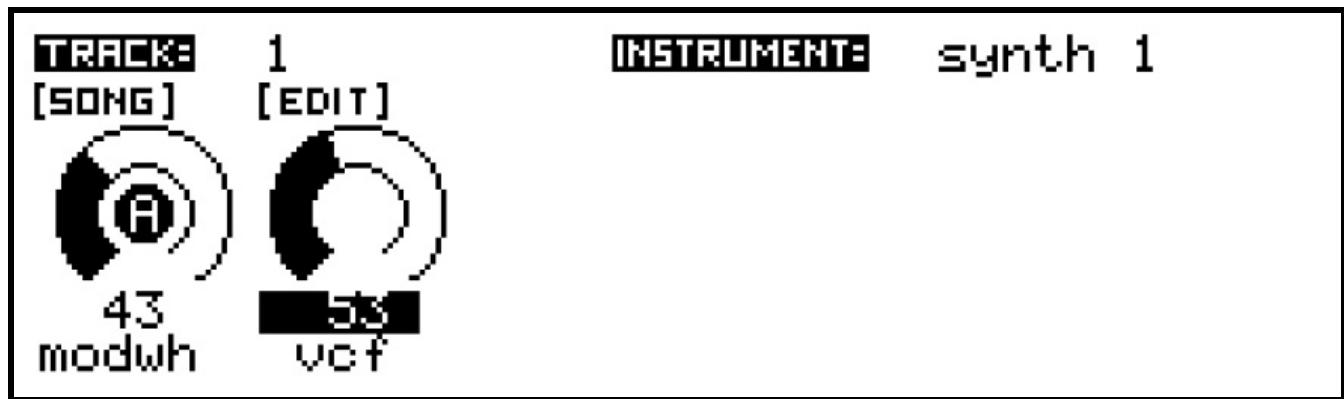
For the SCENE level, **you must also make sure to save the SCENE itself** to make the stored value permanent, unless you have the scene auto-save option enabled.

Once a value has been saved, a new label will appear above it.

Here, for example, the current "modwh" (CC #1) value has been saved at the SONG level, and the "vcf" (CC #2) value at the SCENE:



If either value is now adjusted using the encoders or knobs, a label will appear to show that the value has been manually changed:



Saved settings for the selected value can be removed by pressing the DELETE key. This works incrementally – if a saved value has been EDITed, the first press of DELETE will revert to the saved setting. A further press of DELETE will then remove the saved setting.

Deleted settings saved at the SCENE level require the SCENE to be re-saved to make their removal permanent.

Track Control Values

The second type of track values are potentially far more interesting than MIDI CCs. They give access to a number of internal controls in the sequence engine which modify the behaviour of tracks as they play patterns.

They are added in much the same way as the MIDI values – pick an empty slot, press INSERT, then choose “track control”, and select one from the list. The track control list includes one entry which is really a MIDI control – **pgm**, for sending program change messages.

Track control values can also have knobs A or B assigned to control them directly, with appropriate scaling of the knob position for those controls with wide value ranges. And their values can be saved at the SONG or SCENE levels, as with the MIDI values.

The track control values are as follows:

value	purpose	range
pgm	send MIDI program change message	0 - 127
quant%	<p>applies real-time, variable quantisation to CK patterns. The default quantisation grid is taken from the pattern, but can be changed using the tbase track control value.</p> <p>the Swing control from the main play menu operates in tandem with this control.</p> <p>For P3 patterns, this will scale the step delay values, with 100% corresponding to 0 delay.</p>	0 – 100%
note%	transposes notes as a proportion of the interval between the note and a centre note. (default centre note is C 4)	0 – 255%
noteC	sets the centre note for the note% scaling control.	C0 – G X (G 10)
velo%	adjusts velocity values as a proportion of the distance from a centre velocity value.	0 – 255%
veloC	sets the centre velocity value for velo%	1 - 127
leng%	scales the lengths of notes	0 – 2000%
tbase	sets the spacing for the grid used by the quant% value.	as tbase
xpos	transpose by semitones	-12 to +12
octave	transposes notes by octaves.	-4 to +4
knob1	the knob1 value used for knob mask and grab events in P3 patterns.	0 to 127

knob2	the knob2 value used for knob mask and grab events in P3 patterns.	0 to 127
fts-R	The root note for the track's FTS process – applied separately from Scene-level FTS	C-B
fts-S	The scale for your track – with all meaning all notes are allowed through (has the same effect as turning off the scene-level FTS).	all, maj, aeo, har, mel, blu, pen, mix, lyd, bl2, aug, arb, bal, hng, ori, rga, chi, hn2, jap, jp2, per, dim, who, u01-u08
reich	Named after Steve Reich, this TV alters the length of the playback loop for a CK pattern by +/- 12 ticks, leading to incremental phasing between tracks.	+/-12

The effects of some of these controls will be fairly obvious.

Others are best experimented with, to see what uses you may find for them.

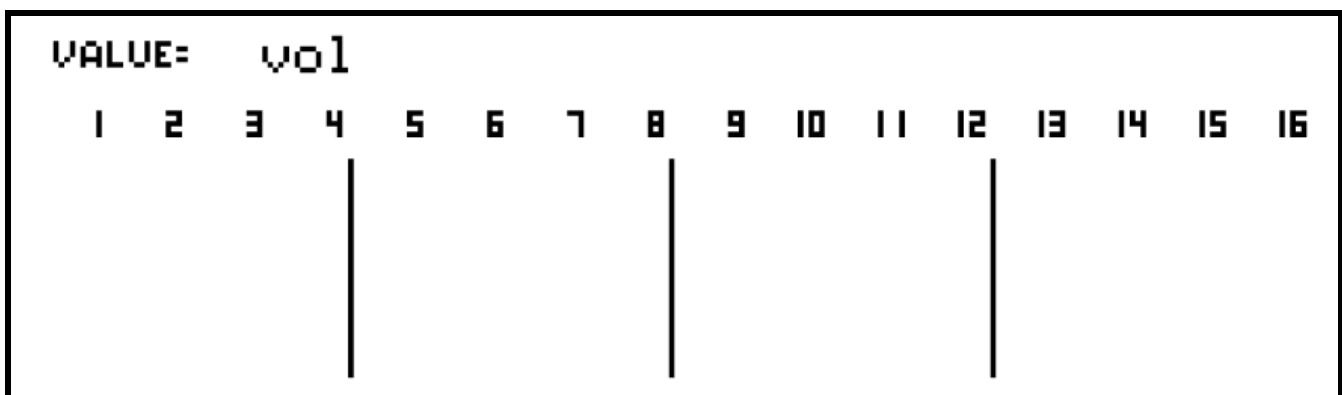
Track Mixer Page

The track mixer page gives you quick access to the MIDI volume and pan controls for every track in the current bank.

These are available whether or not the volume and pan CCs have been added to the track values for each instrument, although they will have no effect on tracks which don't have an instrument assigned.

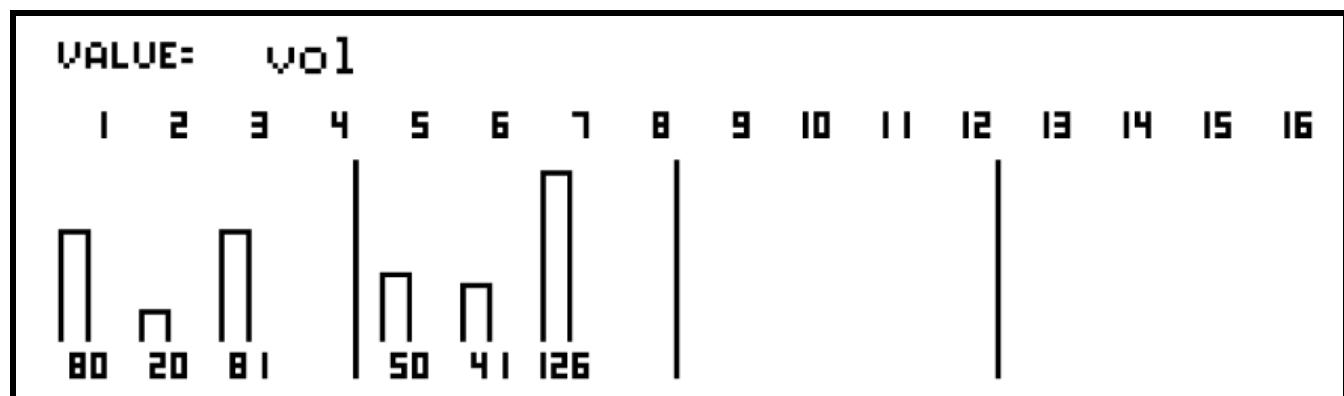
To access the mixer page, from the TRACK page, hold SHIFT and press TRACK again.

Assuming no volume or pan CCs are in use, the display will look like this:



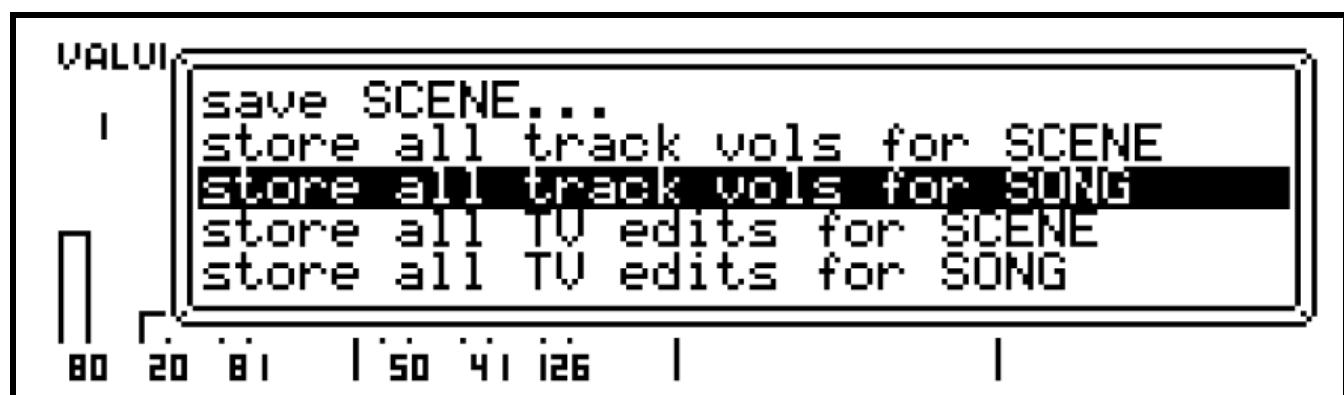
The ROW encoder can be used to switch between the vol and pan controls.

The encoder for each track will send the selected controller, and update the display with the new value:

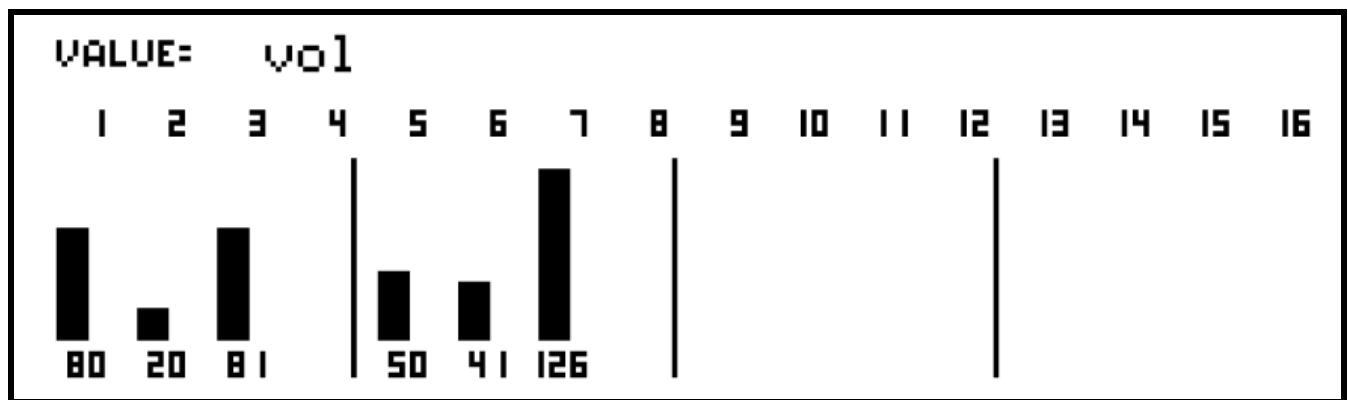


If the same instrument is assigned with the same MIDI channel to more than one track, you will notice that the value for each of its tracks change together when any is edited.

The current values for each track can be saved by pressing the SAVE key, then selecting the store all track vols/pans for SONG/SCENE option:



When the current value matches the stored value, the display will show filled bars:



To remove a value, hold the DELETE key and press the track encoder. The delete process will remove the edited/current value, then any stored value in the same way as the normal track values.

10. CK patterns

As mentioned briefly in the section “Creating and Assigning a Pattern”, Cirklon has two different pattern types.

A P3 pattern is made up of between 1 and 16 bars, each bar holding 16 steps of the various row values described in the P3 pattern edit section.

In contrast, a CK pattern is stored as an “event list”.

An event in this context refers to a MIDI message to be sent at a specific time.

A CK pattern is a list of MIDI events – MIDI messages stamped with the time at which they should be sent during playback of the pattern.

In the case of note events, they also have a length value. A note-off message must be sent to release a note that has been triggered with a note-on. The length value of a note event determines the time between the sending of the note-on and note-off messages.

The CK pattern can hold any number of events, from zero, up to the available limit of memory. Its bar length value can take a much wider range than a P3 pattern, since there is no fixed allocation of memory required for each bar. Memory is only used as needed for the number of events added to the pattern.

The CK pattern can also be set to “infinite” length – the events will play through once, for as many bars as required, but the pattern will never loop.

The two pattern types both have a distinct set of advantages.

In the case of P3 patterns, the fact that the pattern data is held in a rigid structure makes possible their non-linear direction modes, step skipping, and aux event processing.

For CK patterns, the event list structure means that there is no theoretical limit to the number of notes (or other events) that can occur at the same time. This makes the CK pattern ideal for polyphonic keyboard and percussion patterns, and patterns with streams of rapidly changing controller values.

(Note that for practical reasons, a limit of 16 simultaneously held notes per track applies)

CK patterns also provide the best way to recreate a live keyboard performance by capturing a pattern with real-time record.

When you play a note on a MIDI keyboard, a note-on message is sent specifying a note number and velocity value.

If you move the modulation wheel, a controller message is sent with the new position of the wheel. Various other message types may carry pitch-bend, channel pressure (aftertouch), polyphonic pressure, and program change information.

If these messages are recorded in a CK pattern, and sent back to the originating instrument, it should recreate the original performance exactly.

Every note will be recorded with precise timing information, and a unique velocity value, in contrast to the limited polyphony and shared velocity value in P3 patterns.

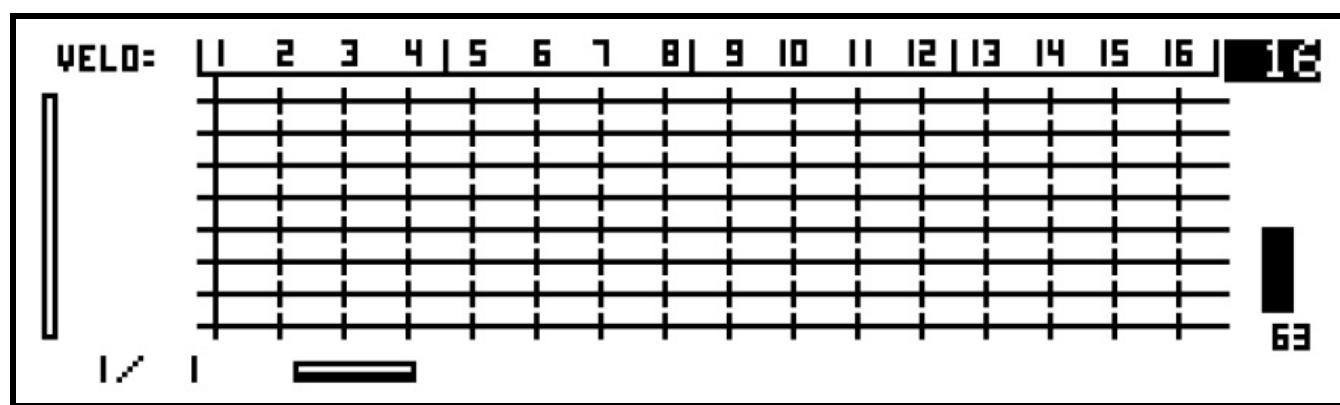
One disadvantage of the event list format is that it is possible to rapidly eat through memory if you attempt to record too much information.

Drum Grid Edit

The primary reason for the addition of the CK pattern type in Cirklon was to overcome the limitations of the P3 Sequencer for creating polyphonic drum patterns.

Therefore, the main pattern edit mode for a CK pattern takes the form of a drum grid editor which borrows heavily from some of the classic Japanese drum machines of the 1980s.

If you create and assign a CK pattern to a track, then press the PATTERN key, you will enter drum grid edit. For a newly created pattern, with no notes entered or recorded, the display will look like this:



Typically for a percussion pattern, you will be using a relatively fixed set of notes, each of which is mapped to a drum sound on a synth, sampler or drum machine.

Drum grid edit shows each note present in the pattern on the rows of the grid, in ascending order.

Since there are no notes in this new pattern, there are no note names beside any of the rows.

To manually enter new notes into the pattern, you first need to add a row for that note.

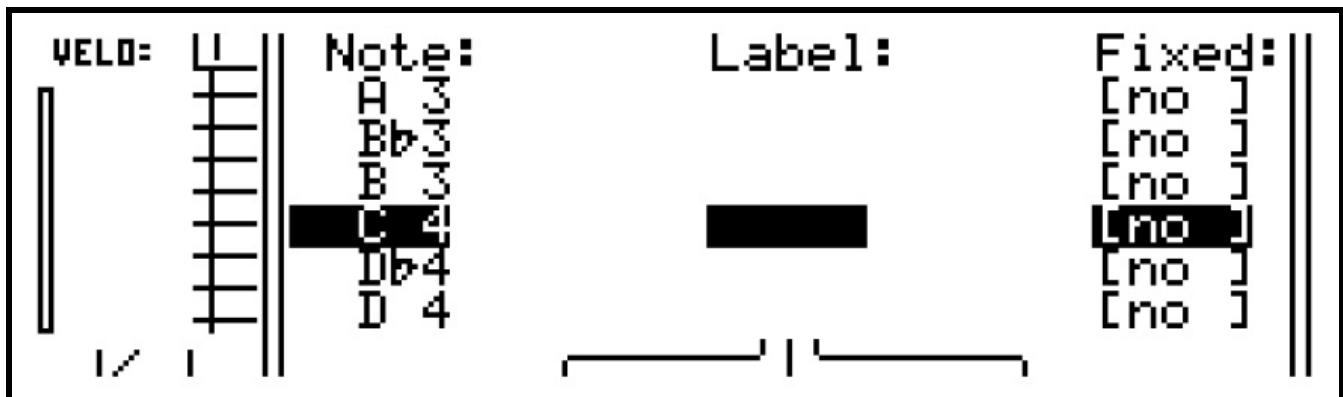
If you press one of the step keys before any rows are assigned, the message, "add note row first!", will appear, followed by a page for adding notes.

Otherwise, the add note page can be accessed via the edit menu.

Press the MENU key, and this list of options will appear:

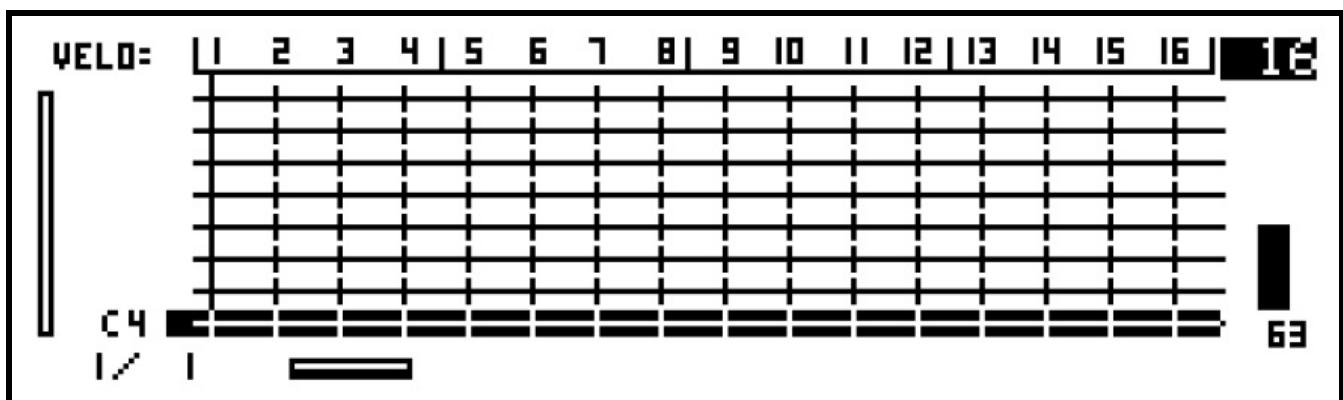


Use the VALUE encoder to scroll down to “add/edit rows”, and press to select this option. A list will appear with three columns for **Note**, **Label** and **Fixed**:



The VALUE encoder is used to scroll the selection up and down this list. Encoders below the display can be used to edit the settings for each note row. The purpose of the Label and Fixed values will be explained later. For now, just press ENTER to add the default note C 4.

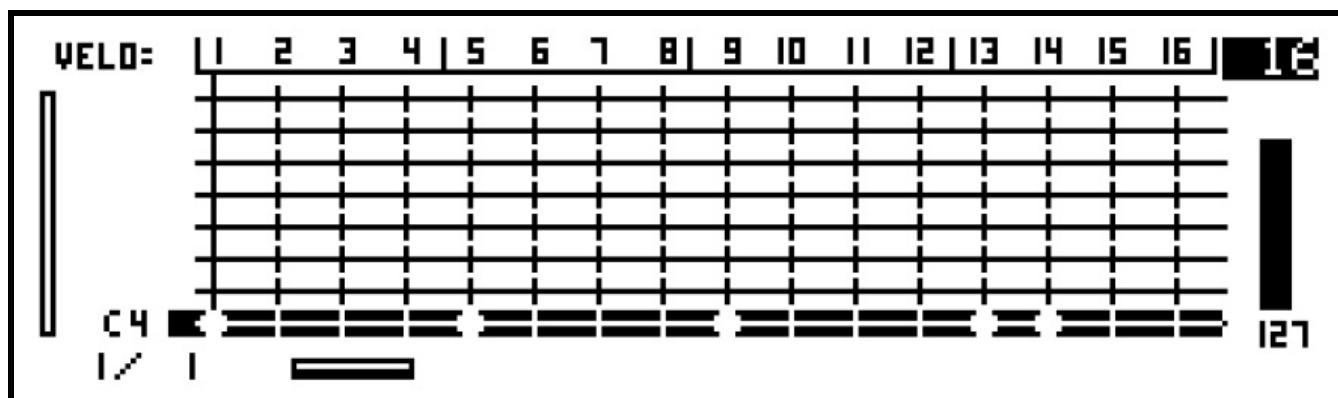
You will return to the grid view, with the bottom row of the grid now labelled, “C 4”. This row of the grid is also highlighted, to show it is the current row for editing:



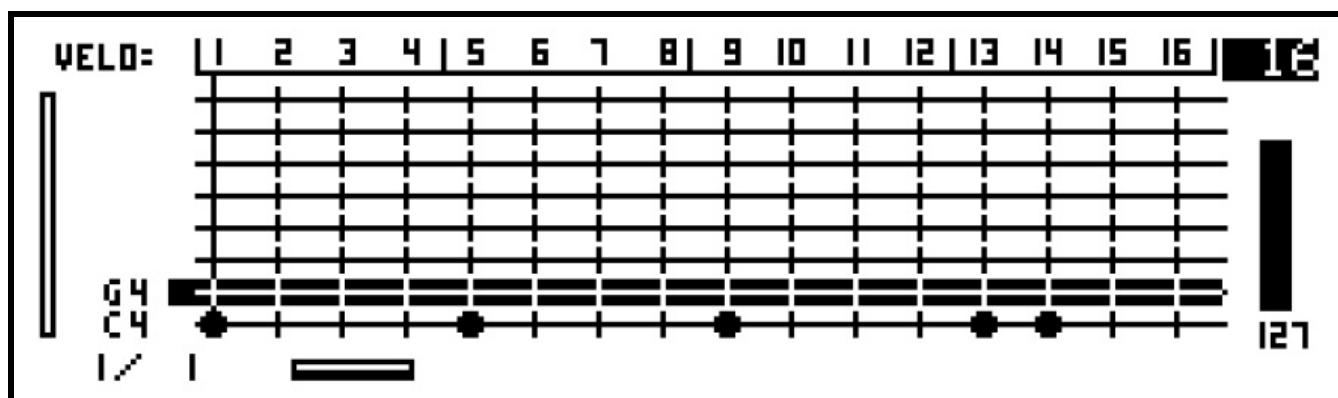
Now you can enter note C 4 at any of the 16 step positions in the bar by pressing the step keys. Since note events need a velocity value, for newly added notes the velocity is taken from the current position of front panel knob A.

You can see in advance what this value will be from the indicator on the far right of the screen – as shown here knob A is set to give a velocity of 63.

Press some of the step keys now, and you will see the step LEDs light for each step where a note is added, and a blob appears in the corresponding position in the display:

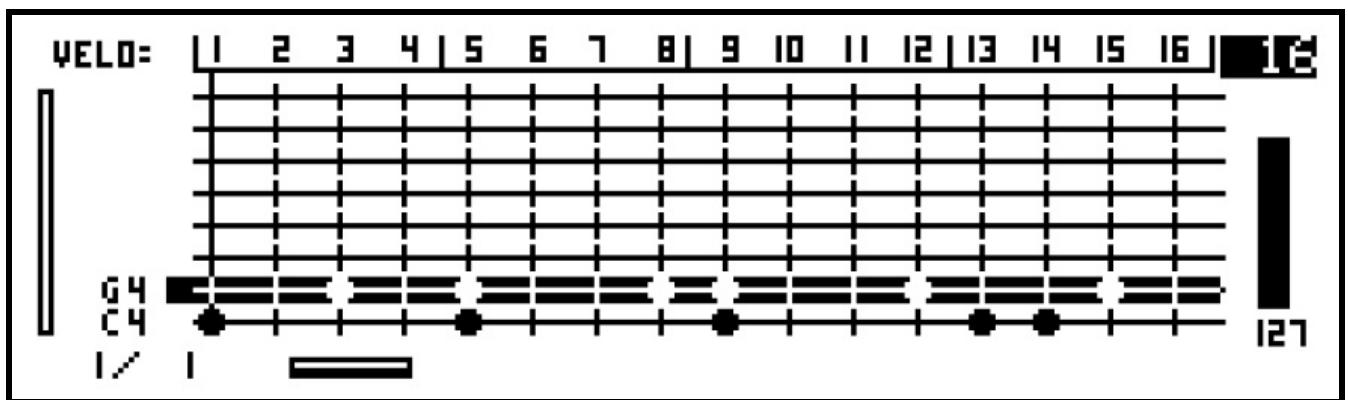


Now let's add the row for another note, using the MENU, "add/edit rows" option as before. This time, select note "G 4".



After adding the new row, you will see the label for "G 4" appear above the first note, and the selection highlight moves to the new row.

You can now use the step keys to enter note G 4 into the pattern:



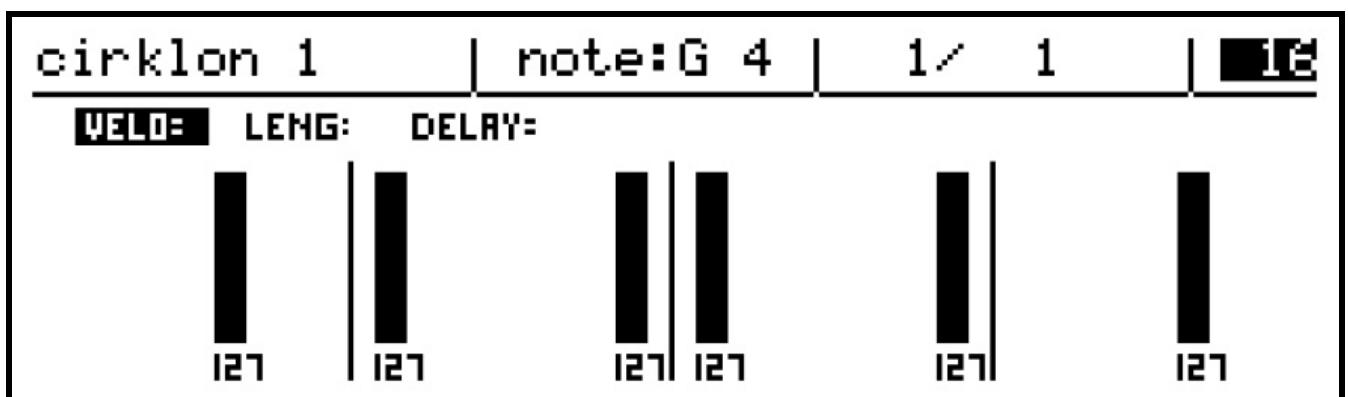
Turn the ROW encoder to move the selection between the two rows. The step key LEDs will show the status of notes for the current row, while the keys toggle those notes on and off (active or muted).

An important point to note is that, once a note event has been added on a step, turning that step off does not remove the note from the pattern, it simply “mutes” the note so that it does not play. This is so that the velocity value set for the step is not lost if you experiment with turning the steps of a pattern off and on.

Row View

To view the velocity values for the current note row, you must switch to the **row view**. You can switch manually by pressing the ROW encoder, but the view will switch automatically if you adjust any of the velocity values using the step encoders, since the velocity value is not directly visible in the grid view.

Turn one of the step encoders while on the G 4 row, and you should see a display something like this:



Encoder editing of the velocity values is similar to P3 pattern editing.

Turn the encoders to adjust the value, or hold in and turn the encoders for steps of 10.

You can also hold the encoder for one step while adjusting another step to force any values between the two steps to follow a straight line – the slope edit function.

Note that if you hold a step that has no value present, slope edit assumes a value of 0.

If you have turned any of the steps in the pattern on then off, leaving the note muted, the saved velocity value will appear in the row view as an un-filled bar.

While working on a pattern, you may want to clear out any muted notes.

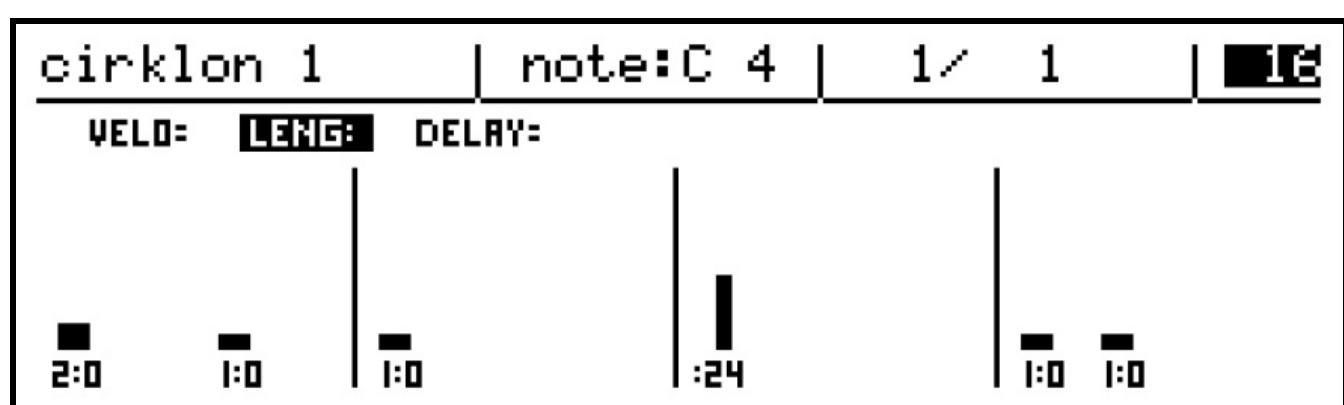
This can be done using the “purge muted notes” option, on the edit menu.

As well as showing the velocity values for each step of the current row, the row view also shows the pattern name, note name for the current row, current bar position and bar length of the pattern.

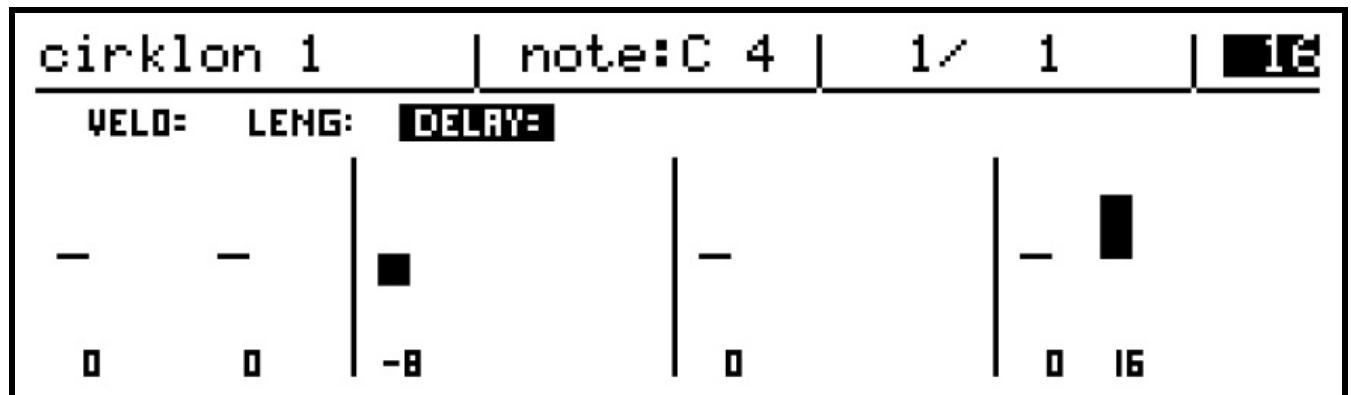
Notice also the three labels, VELO, LENG and DELAY in this view.

These three sub-rows give access to the velocity and length values for each note in the pattern, and allow the precise timing of the event to be moved up to halfway before or after the step position.

Turning the ROW encoder while in the row view will step through the velocity, length and delay values for each note in turn, with the current value highlighted:



The DELAY value can be positive, to show a note that will trigger after the step time, or negative, to show it will trigger before the step:

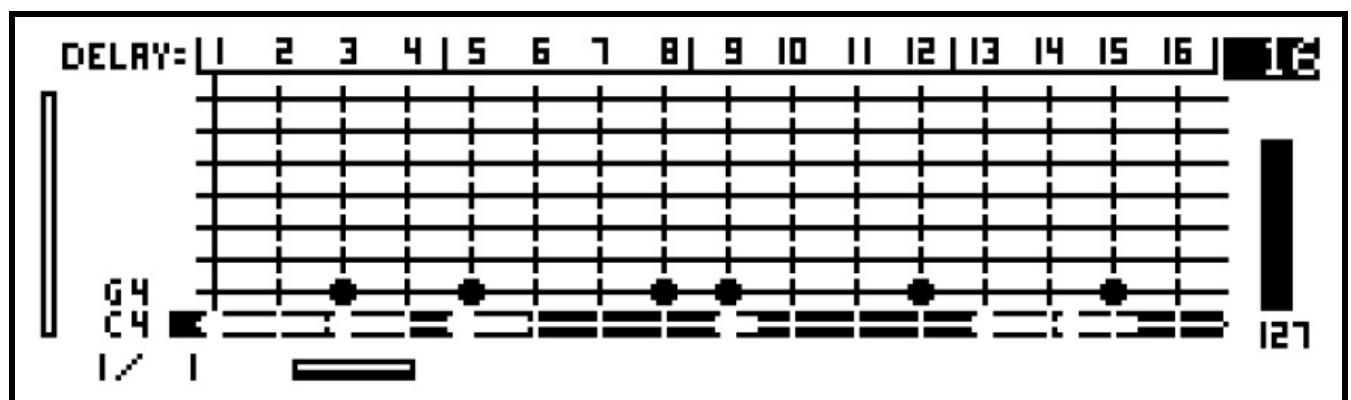


When you switch back to the grid view after selecting the length or delay sub-row, the selected sub-row will remain active for editing.

Turning the ROW encoder will move up or down the note rows as before, with the encoders adjusting the velocity, length or delay as shown by VELO, LENG or DELAY at the top right of the display.

When you make edits to the length or delay values using the encoders in grid view, the display will not switch to the row view as it did for velocity, as the changes to these values can be seen in the grid display.

The blobs that show each active note on the grid have tails to show their length, and their position may be slightly ahead or behind the vertical lines for each step position to show they have a non-zero delay value:



Note that a tail for a long note will be truncated where it overlaps a following note on that row.

Quick Note Number Edits

There is a shortcut in CK edit to allow a note event to be transposed up or down quickly. This saves you having to add a new row for the note you want to transpose to, adding a new note then manually muting the old one.

To do this, hold the SHIFT key while turning the step encoder for the note on the current row you wish to move.

If a note being moved this way passes through another row with a note present on the same step, it may be the other note that is then moved with further turns of the encoder, taking a different velocity, length or delay with it.

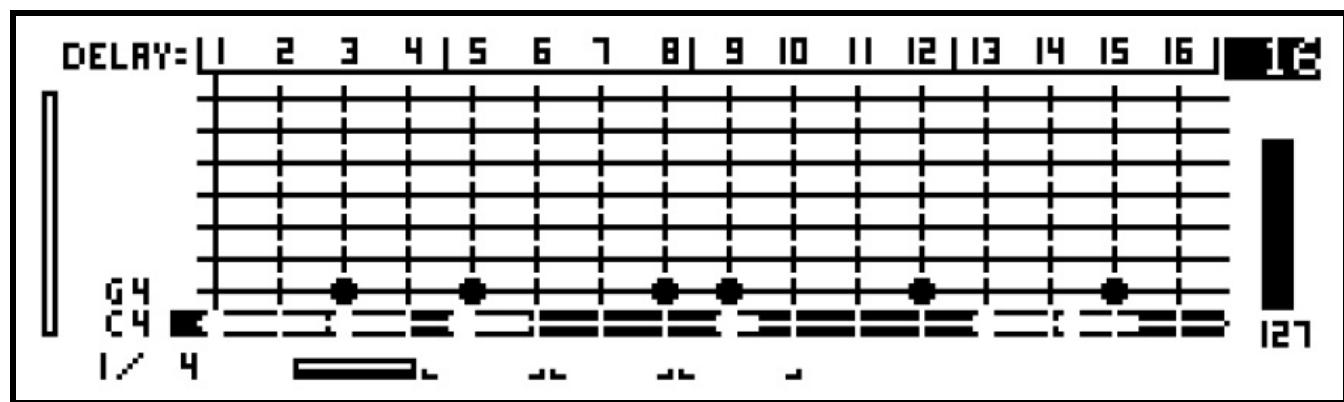
The visible rows for the pattern are automatically updated as notes are moved in this way.

Bar Length

As in P3 edit, the BAR encoder is used to select which bar of the pattern you want to edit in a pattern longer than one bar.

Again, the length in bars of a CK pattern can be adjusted by holding the LAST key, then turning the BAR encoder.

Let's set the bar length to 4 bars. Hold the LAST key and turn BAR three steps clockwise, until the display looks like this:



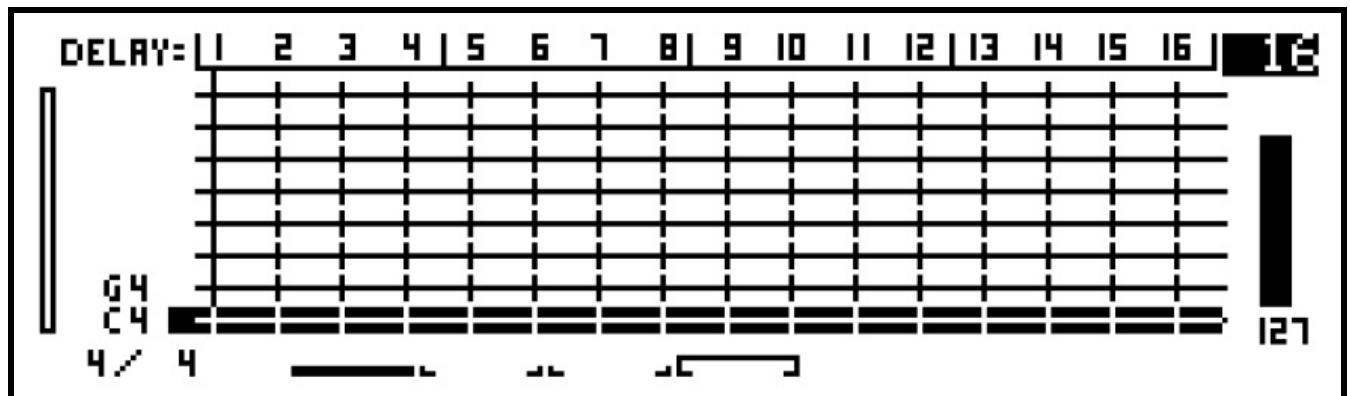
The current edit bar, total length in bars and playback position are shown along the bottom of the grid view display.

The first number from the left shows the current edit bar (which stayed at bar 1 as the length was increased), followed by a slash (/), then the bar length of the pattern – now 4 bars.

To the right of the numbers, the edit bar selection and playback position are shown graphically. For a pattern longer than 8 bars, the graphic will show the pattern in blocks of up to 8 around the edit bar. The bar currently playing is shown as a solid line.

In our newly lengthened pattern, all bars except bar 1 will be empty. The two rows added for notes C 4 and G 4, will be visible in all bars.

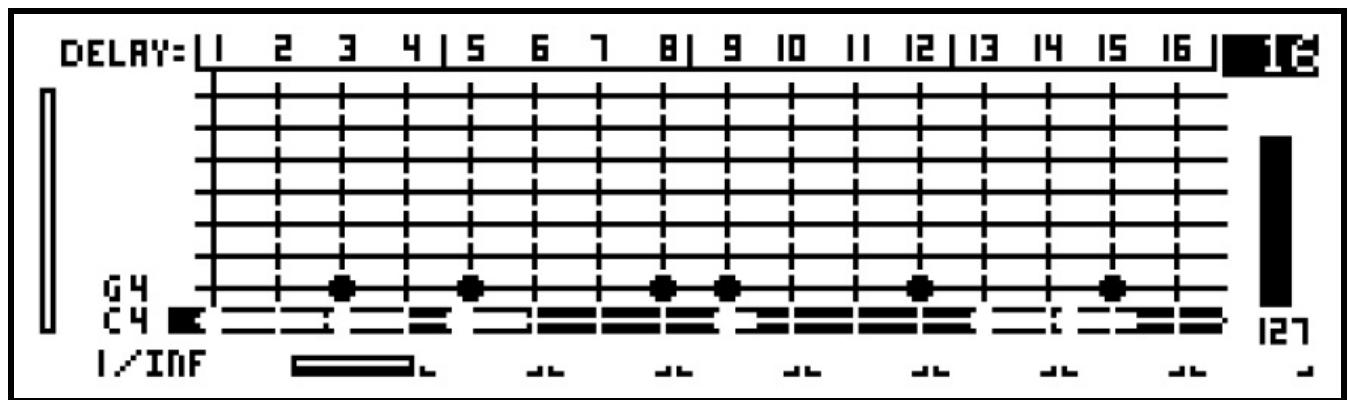
Turning the BAR encoder clockwise to advance to bar 4 of the pattern, the display should look like this:



This shows you are viewing bar 4 of the 4 bar pattern. The solid white line shows the playback position still at bar 1, while the horizontal bracket above bar 4 confirms the edit position.

It is possible to set a CK pattern to have an “infinite” length so the pattern will play through only once, and never loop.

This is done by adjusting the bar length as if to zero, by holding LAST and turning BAR anti-clockwise until the bottom line of the display looks like this:



Last Step

It is possible to vary the number of steps in each bar of a CK pattern, from 1 to 16.

This is done by holding the LAST key then pressing the step key number for the desired length.

Note that for a CK pattern, changing the last step value will alter the overall length of the pattern, and how the pattern is divided into bars for editing.

However, the note events will retain their absolute start position within the pattern.

This differs from a P3 pattern, where each bar always holds 16 steps, but some steps are passed over during playback when the last step value is less than 16.

Looping of Bars & Steps

Often, especially when working on long patterns, it's helpful to loop a few bars or a few steps so you don't have to wait for a whole pattern loop to hear the effect of your edits.

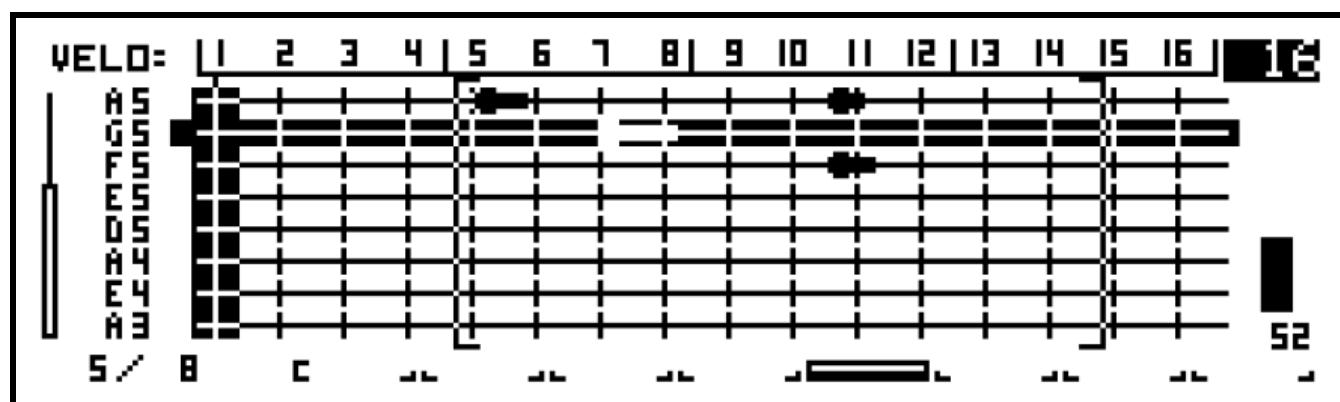
To achieve this, the following key combinations have been provided from within Ck Pattern Edit:

FILL + <step encoder> sets loop start

LAST + <step encoder> sets loop end

Also:

FILL + BAR sets the loop points to the current bar



The loop points are shown as thin, enclosing brackets around the chosen steps.

Repeating the action will **clear** the loop point(s) you just added.

View and Signature Timebase

The time duration of each bar of a CK pattern is equal to the number of steps per bar (as determined by the last step function above), multiplied by the duration of each step, as determined by the **timebase** setting.

By default, a new pattern will contain bars of 16 steps of 16th notes – equivalent to a 4/4 time signature.

It can be useful when editing a pattern to change the timebase setting to “zoom” into parts of the pattern to add more rapid notes.

In order that changing the timebase in this way does not cause an immediate change in the length of a bar, and therefore the whole pattern, CK patterns have two separate timebase values, the **view timebase** and the **signature timebase**.

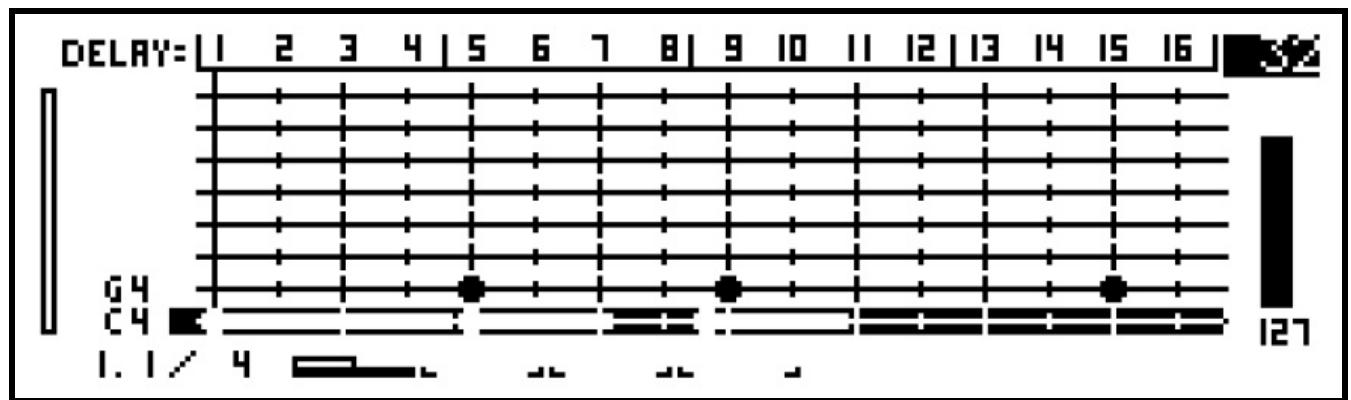
The view timebase determines the time represented by each step in the grid for editing.

The signature timebase, along with the last step value, determines the duration of each whole bar of the pattern, and therefore the overall length of the pattern.

The view timebase always appears in the top right corner of the display.

You can select any of the timebase values available in P3 patterns by turning the VALUE encoder.

Let's look at what happens if you change the view timebase for our example pattern from 16 to 32, by turning the VALUE encoder one step clockwise:



With each step now representing a 32nd note, but the grid only able to show a maximum of 16 steps, only one half of a bar of the pattern can be shown at a time.

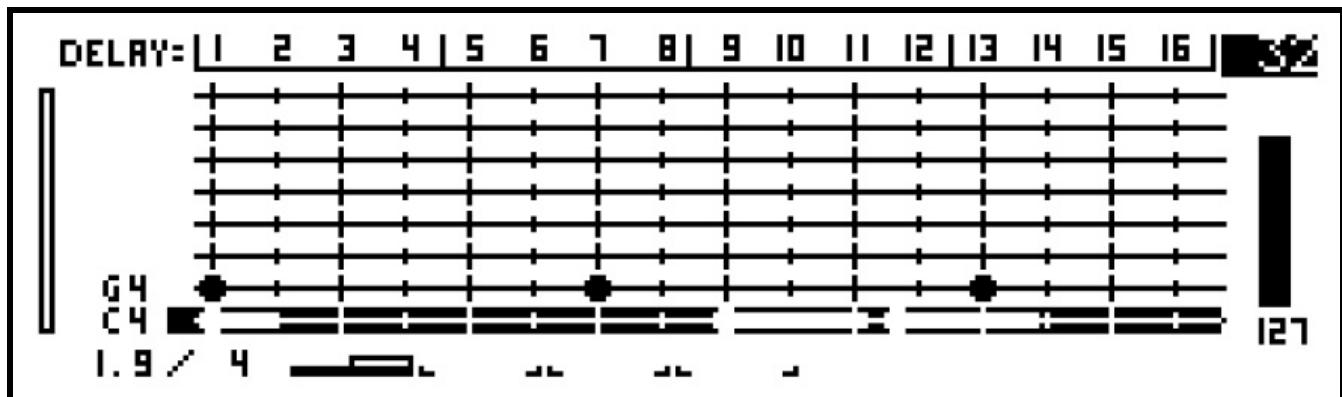
This is indicated by the addition of a point and a second number to the edit bar position, and by a change in the horizontal bracket above the current edit bar graphic.

The edit bar position 1.1 means you are looking at bar 1, from step 1.

The step number here refers to the step number at the *signature timebase*.

The horizontal bracket above the current edit bar graphic shows that the grid view is only covering half of the length of the bar.

If you turn the BAR encoder one step clockwise from this position, the display should change to this:



You can see you are still on bar 1, but now looking at the second half – starting from what would be step 9 of the pattern viewed at its signature timebase.

The graphic now shows that the grid view covers the second half of the bar. You may also notice there is a difference in the style of the vertical lines for each step position. The lines for the signature timebase step positions are more solid than those for the extra steps shown at the higher view timebase setting.

Changing the Signature Timebase

The signature timebase value is updated whenever you alter the last step value. The current view timebase at the point where you change last step is taken as the new signature timebase, re-defining the length of a bar.

As you adjust the view timebase setting, the grid view will divide the view of the bar into as many sections as it needs to show the full bar in whole signature timebase steps.

This may mean odd time signatures are shown in sub-bars of uneven lengths when zoomed. For example, if you select a view timebase of 8, then set last step to 11, you have in effect set a time signature of 11/8 for your pattern.

If you change the view timebase to 16, the first part of each bar will be shown across 12 steps, and the second over 10 steps, rather than as two halves of 11 steps each.

This ensures the two 16th note steps required to show each of the 8th note steps at the signature timebase appear together.

GANGing Notes in CK Edit

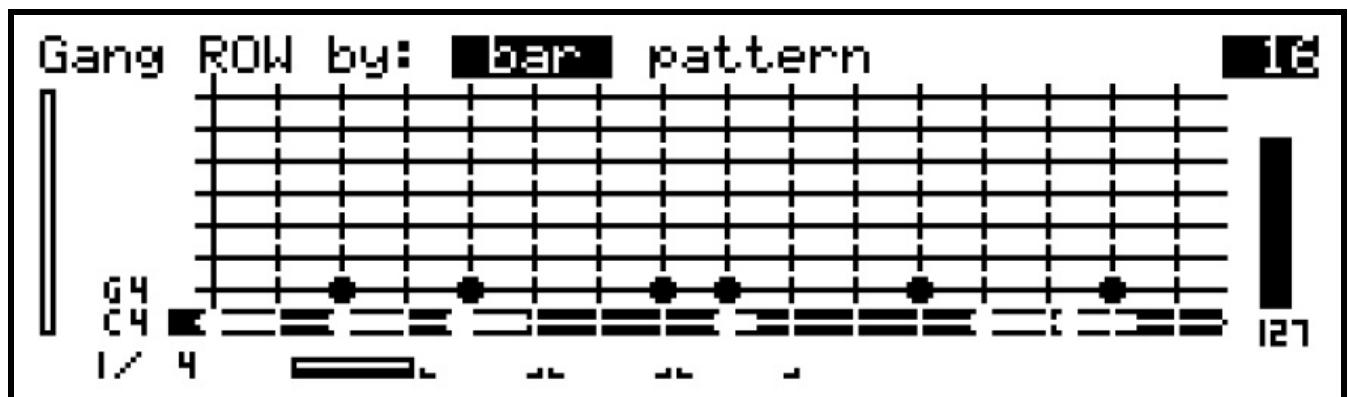
CK edit offers the ability to select multiple notes for editing at the same time, in a similar way to the GANG function provided in P3 edit.

Note that in P3 edit, the gang function selects which steps are to be edited in parallel. When ganging notes in CK edit, it is the individual note events which are marked as ganged. This allows some of the notes on a single step to be ganged and their values edited together, without un-ganged notes on the same step being affected.

Let's gang some of the notes in our example pattern.

Return to grid view, and select the row for note C 4.

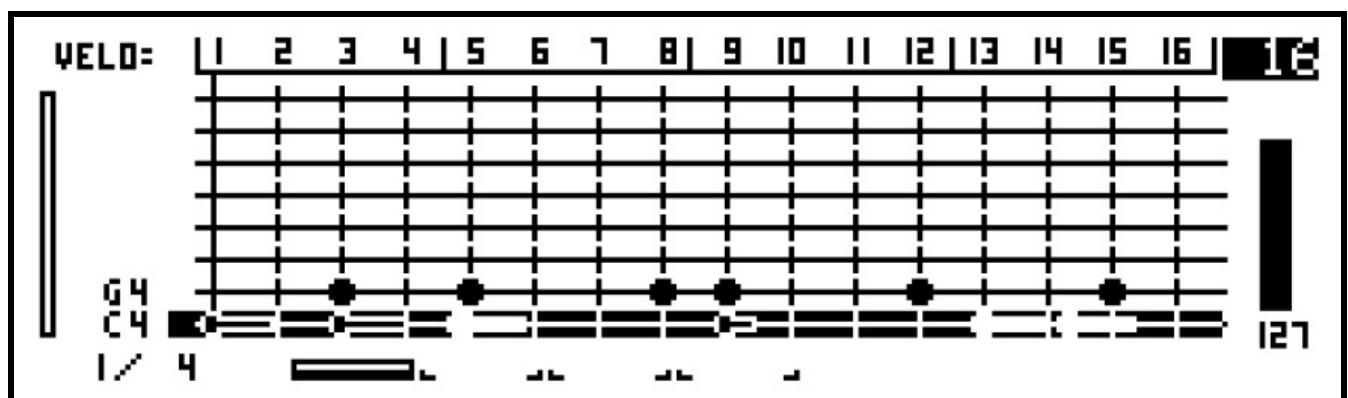
Then press and hold the GANG key. The display will change to look like this:



The top line of the display shows the text, "Gang ROW by", which we'll come back to. With GANG held, press the step keys for steps 1, 3 and 9. Then release the GANG key.

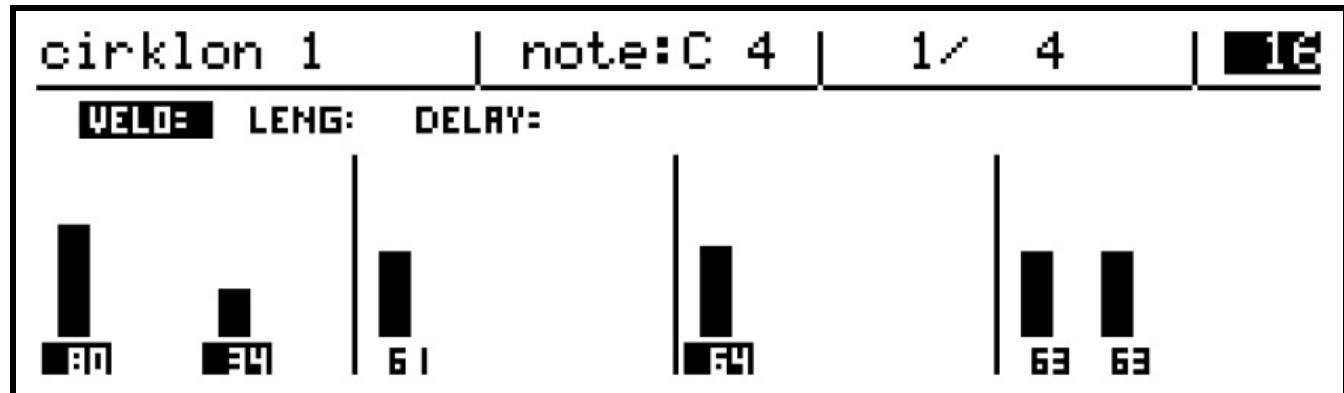
The GANG key will remain lit, and the encoder LEDs above steps 1, 3 and 9 should also be lit in green, or amber for a ganged step which was already the current edit step.

The blobs on the display showing the newly ganged notes will become hollow:



If you now turn one of the encoders for any of the ganged notes to alter its velocity value, the display will switch to row view as normal, but you will see the velocity value for all the ganged notes change together, by the same relative amount.

The number values shown below the bars will also be highlighted to indicate which steps are ganged:



It's possible to gang notes on different rows at the same time.

While holding the GANG key, turn the ROW encoder to select a different note row then gang some of the notes on that row using the step keys as before.

While holding the GANG key, pressing the step key for a note which is already ganged will remove that note from the selection.

GANGing by Step or Row

To quickly gang all the notes on a single step, hold GANG and press the **encoder** for that step.

To gang every note on the current row, hold GANG then press the ROW encoder.

The GANG + ROW function can select all the notes in the current bar, or for the entire length of the pattern. The scope for selection is shown at the top of the display while the GANG key is held, and can be changed using the VALUE encoder.

The current setting (*bar* or *pattern*) is highlighted on the display while GANG is held.

Clearing the Ganged Selection

When the gang function is active, pressing and releasing the GANG key alone will turn off gang mode. However, by default the notes will retain their ganged marker.

You can re-enter gang mode by pressing GANG again, and the previously ganged notes will already be selected.

If you wish to clear all ganged note selections, hold the GANG key and press DELETE. This will un-mark all notes in the pattern, and exit GANG mode.

Depending on your typical use of the gang function, you may find that you always want to have the gang selection cleared when you exit gang mode. This can be enabled with the user config option, “*ck edit, auto clear gang*”, described in the play mode menu section.

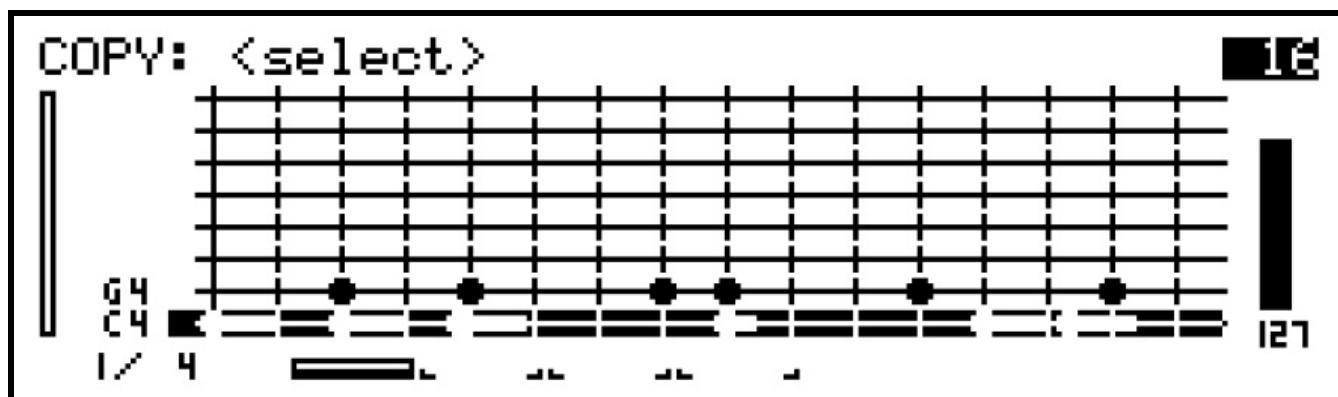
Copying, Deleting and Inserting Notes

The COPY function, as in P3 edit, allows you to copy part or all of a pattern to a clipboard. The copied notes can then be inserted (pasted) elsewhere in the pattern, or into another CK pattern. Note that the clipboards are separate for P3 or CK pattern types and copy/insert operations will not convert between them.

The basic copy function copies all the notes in a chosen range of steps.

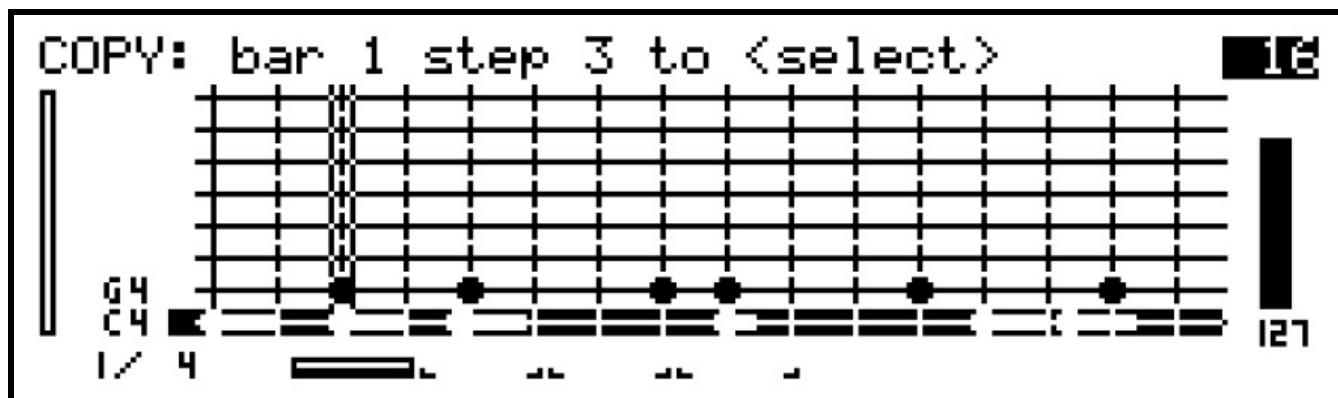
To start a copy operation, press the COPY key.

The key LED should flash, and the top line of the display will show this text:



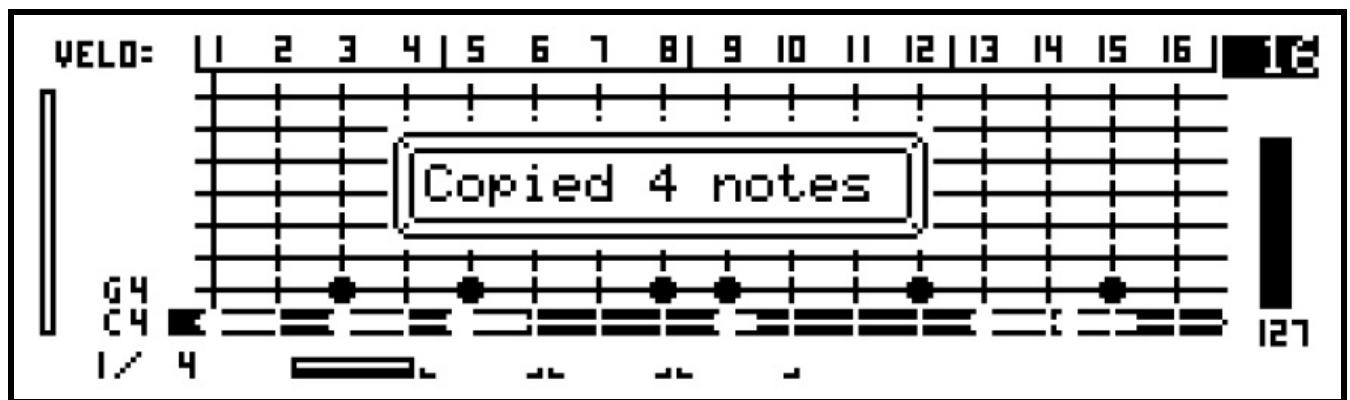
As an example, let's copy the notes from step 3 to step 5.

Press step key 3 to select the start of the copy range. The display will change to show:



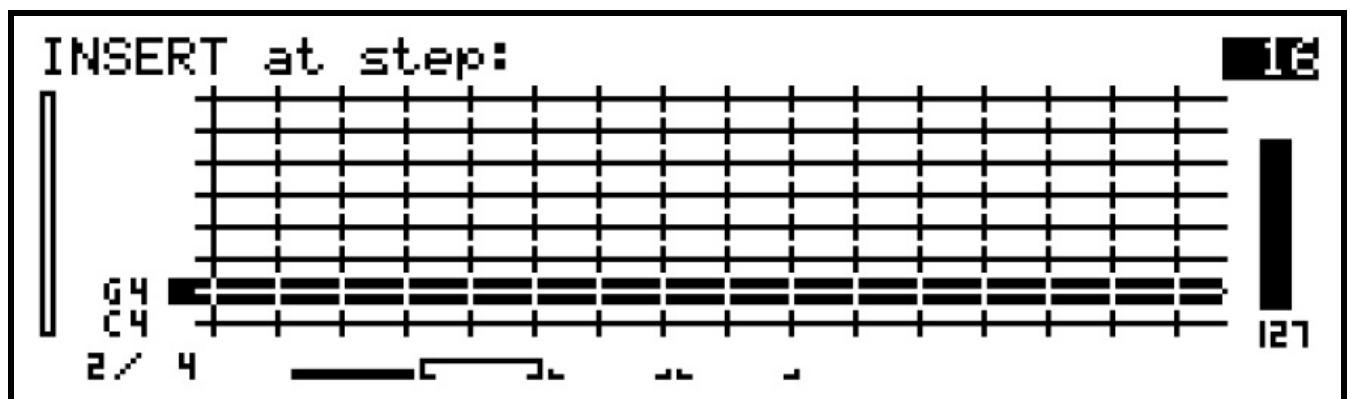
The vertical marker on the grid shows the start position for the copy.

Now press step key 5. The display will show:

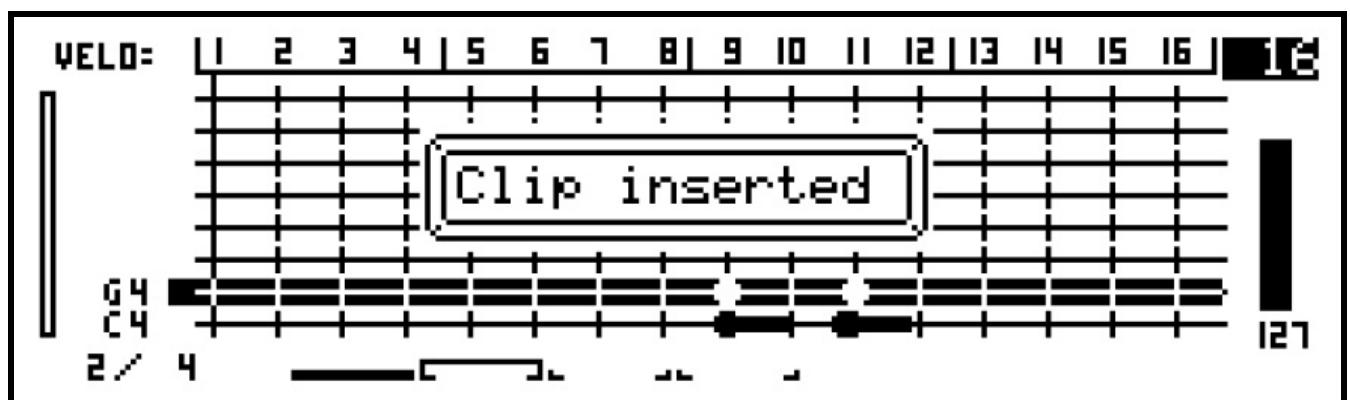


The four notes from steps 3 and 5 are now copied to the clipboard.

Now turn the BAR encoder to select bar 2, to insert the copied notes elsewhere in the pattern. Press the INSERT key. The LED will flash, and the top of the display will show:



Now press step key 9. The display will confirm the notes have been inserted, and the new notes will appear on the grid:



The relative position of the copied notes to the first step of the area that was copied determines where the notes will be positioned when they are INSERTed.

Copying by Bar or Ganged Selection

You may also use the COPY function to copy all the notes in the current visible bar, or all the ganged notes in a pattern.

To copy all the notes in a bar, press the COPY key followed by the BAR encoder.

This popup message should appear:



In this case, the note positions are copied relative to the start of the bar.

When notes have been selected using the gang function, you may copy **only the ganged notes**.

To do this, while gang is active, press the COPY key, then the GANG key.

A popup message should appear similar to this:



The gang function will remain active.

The position for ganged notes copied to the clipboard will be relative to the step position of the earliest note in the ganged selection.

When you INSERT a clip copied from ganged notes, any current ganged selection will be cleared, and the newly inserted notes will appear in the ganged state.

Cutting / Deleting Notes

It is possible to copy notes to the clipboard and **remove them from the pattern** in one operation, using the DELETE function.

Select the range of notes to delete as for the COPY function – press the DELETE key to start, then select the start and end step, or press BAR to delete all notes in the current view bar, or GANG to delete all currently ganged notes.

When selection is complete, the notes will be copied to the clipboard and removed from the pattern.

For some users, it may be preferable to have the option to directly DELETE notes, without affecting the contents of the clipboard. If you would prefer this workflow, you can enable it with the user config option – “*ck edit, COPY+DELETE = CUT*”

When this option is active, pressing DELETE will invoke a direct delete function for the notes selected. They will be removed, and not copied to the clipboard.

To CUT the notes to the clipboard, hold the COPY and DELETE keys together. The top line of the display will confirm which function is active as you select the range of steps.

Ripple Edit

For CK patterns, you can perform ripple edits by holding SHIFT as you press the INSERT or DELETE keys. Ripple edits will add or remove space from the pattern according the clip size, or chosen delete area.

For a ripple INSERT of copied events, blank space is first inserted into pattern (according to the clip length) by moving all following events later in time.

For a ripple DELETE, after the selected events are removed from pattern, any following events are moved forward, so it appears the deleted section has been removed from the total pattern length.

Normally, ripple edits will not alter the overall BAR length of the pattern.

The user config option, “*ck edit, RIPPLE bar length*”, can be enabled to make all ripple operations add or remove bars from the pattern length according to the clip size, rounded to the nearest whole bar.

You can insert a single empty bar into a pattern before the current bar.

Hold SHIFT and press INSERT, then press the BAR encoder.

This function will also increase the pattern bar length.

Refer to the “User Configuration” section of the chapter “Play Mode Menu”, if you don’t yet know how to change the user config options.

Fixing and Labelling Rows

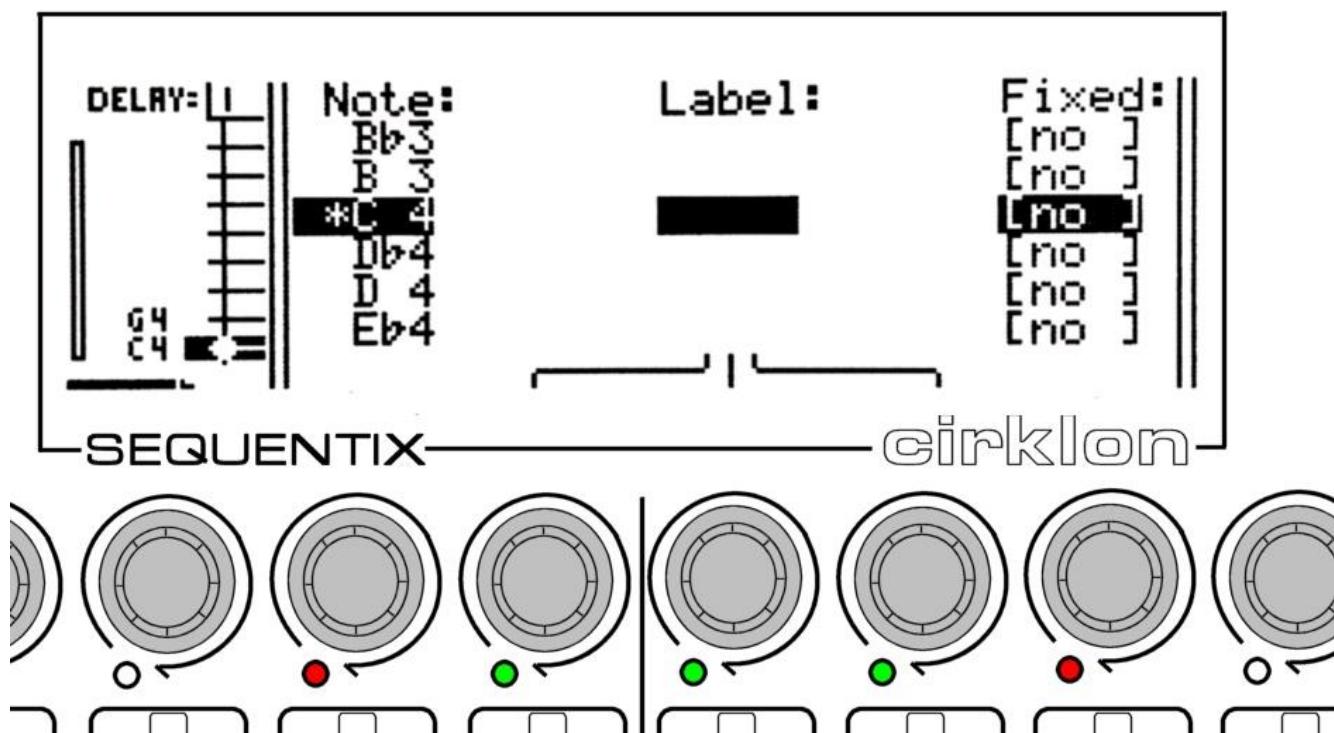
So far we have only added two rows to the example pattern. You could continue adding as many more notes as you need, up to the 128 note limit imposed by MIDI, but the grid view only has space to show eight rows.

Once more than eight rows have been added, the scroll bar on the left hand side of the grid view comes into use. The box represents the current view position within all the rows of the pattern, and will move up and down a thin centre line to indicate where you are within the full range of rows.

As grid edit is aimed mainly at percussion programming, a couple of problems may become apparent. It would be tedious to have to add the same set of note rows to the grid every time you create a new pattern. And you may prefer to see which drum instrument is assigned to each note, rather than the note name itself.

The further two columns in the “add/edit rows” page, “Label” and “Fixed”, address these issues.

For example, let's say the C 4 note in use in this pattern is mapped to a bass drum sound on the instrument assigned to the track. Press MENU, then “add/edit rows”, and use the VALUE encoder to select note C 4 in the list. The display should look like this:



Encoders 8, 9 and 10 are active here for setting a 3 character label to replace the note name. Characters A to Z, 0 to 9, <flat>, <sharp>, backslash and dash can be used.

The “Fixed” option marks this note so it always appears on a row of the drum grid in an empty pattern, or a pattern where the note is not present and the row hasn’t yet been manually added.

Note that you can scroll through the list and edit the labels for multiple notes at one time. Pressing encoder 7 will add a note to the grid without leaving this list. An asterisk next to the note name shows notes that have already been added.

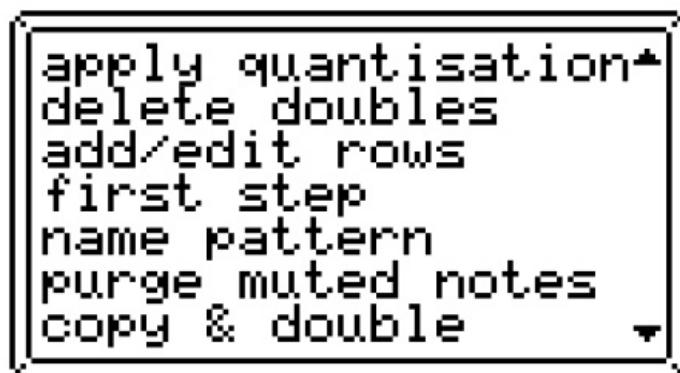
Note also that the labels and fixed options are stored as part of the **instrument definition** for the instrument assigned to the track.

Use the encoders to enter a label for the note C 4, then press ENTER.

On returning to the grid view, the entered label will appear in place of C 4, in the grid and row views. To remove the label, rotate all three characters back to a space, and the original note name will be shown again.

Other Edit Menu Functions

There are a number of other functions to be found on the edit menu:



Apply Quantisation

This function moves the start-times of all events in the pattern to the nearest step position, according to the current view timebase setting.

Quantisation is most likely to be used in conjunction with real-time recording, where note events are not quite “on the beat”.

Remember that patterns may also be left un-quantised, with real-time quantisation applied using the “quant%” track control value.

Delete Doubles

Using real-time record or the clipboard insert function, it's possible to end up with more than one of the same note event occurring at exactly the same time.

The **delete doubles** function checks for duplicate events, and removes all but the first for any given time.

Note that this function is automatically triggered whenever the **apply quantisation** function is used.

First Step and Pattern Rotate

This function allows you to move the start time of all the events in a pattern, such that the notes on a chosen step move to the first step of the pattern, just as on the P3 edit menu. This operation cannot be performed on a pattern set to infinite bar length.

It is also possible to 'rotate' all events in a pattern, one step at a time, by holding the SHIFT key while turning the BAR encoder. The movement of notes should be clearly visible on the grid view.

Note that when the GANG function is active, using SHIFT + BAR to rotate notes will move only the ganged notes.

Name Pattern

The function allows you to edit the name of your pattern using text input mode.

Purge Muted Notes

As described earlier, when notes are turned off by pressing the step keys, the note events are not removed, but left in a muted state.

This function clears out all muted notes, freeing the memory used and tidying up the un-filled bars from the row view.

Copy & Double

This function appends a copy of the entire pattern after the last bar of the pattern, doubling the bar length as it does so.

Continuous Controller Editing

CK patterns may also contain MIDI Continuous Controller events.

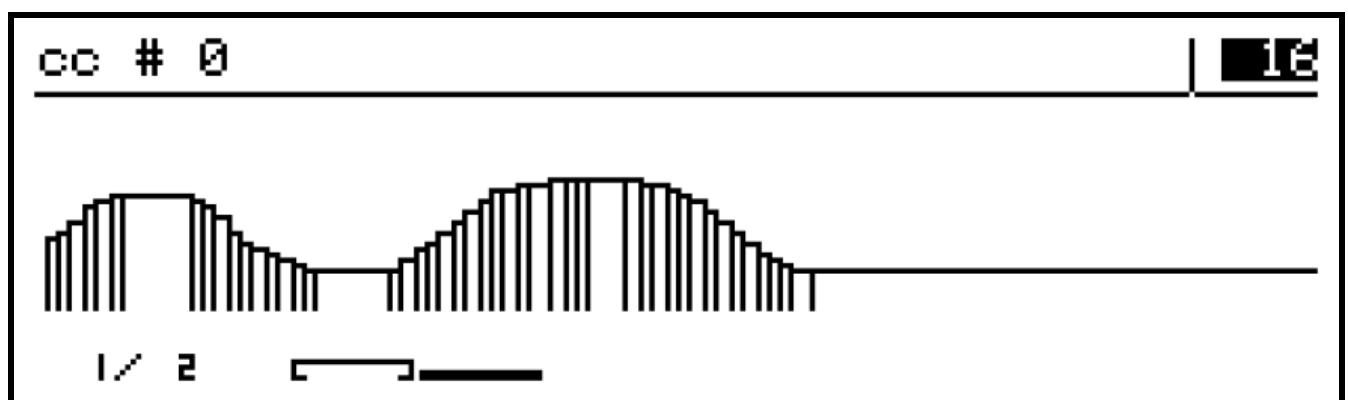
A single event can be entered and edited manually for each tbase step.

Or in real-time record mode (see later chapter), streams of CC events can be captured in track or edit modes using the front panel knobs, or from any CC messages passed by MIDI soft-thru to the assigned instrument.

CC events can be viewed, added and deleted in CK edit using the CC view.

To access this view, hold SHIFT and press the ROW encoder.

The display should look something like this:



The top line shows the current CC number, or the label assigned to the CC if one has been created for a MIDI track value.

Each CC event in the visible bar (or sub-bar) will be shown as a vertical line with its height corresponding to the controller value, and a horizontal line continuing until the next event.

When no CCs are yet present in a pattern, turning the ROW encoder will scroll through all 128 possible CC numbers, from 0 to 127.

If one or more CC events are already present in a pattern, the ROW encoder will scroll through only the used CC numbers. To access unused controllers, you must hold the SHIFT key as you turn ROW. This allows quicker access to the CCs in use, while still allowing an unused CC to be easily selected for addition to the pattern.

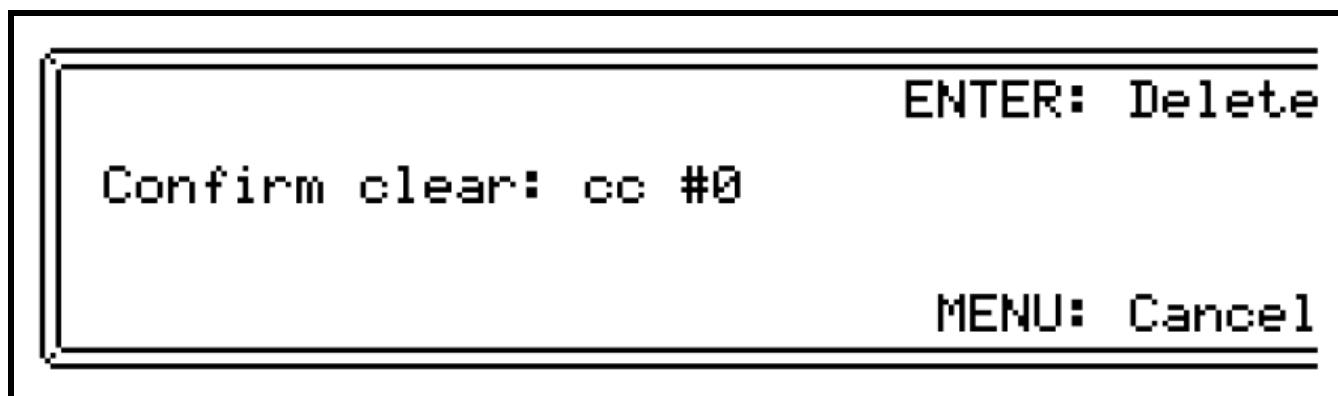
When CC events have been real-time recorded, there will often be a number of CC events in the same step period.

To delete all the events for the current CC number in a single step, hold DELETE and press the step key.

To delete all the events in the current bar, hold DELETE and press the BAR encoder.

To delete all events for the current CC in the entire pattern, hold DELETE and press the PATTERN key.

This function requires confirmation as shown in the screen below:



A single CC event can be added in an empty step by turning the encoder for that step.

A new value will be created, starting from zero.

Or, if an event was already present in that step, its value will be edited.

Note that this will only alter the first event in a step.

Temporarily increasing the view timebase will allow finer control over the placing of CC events in time, but for smooth CC curves, real-time capture from a knob or external control is often preferable.

If real-time record is active, and the record filter has CC recording enabled, new CC events can be entered using front panel knob A.

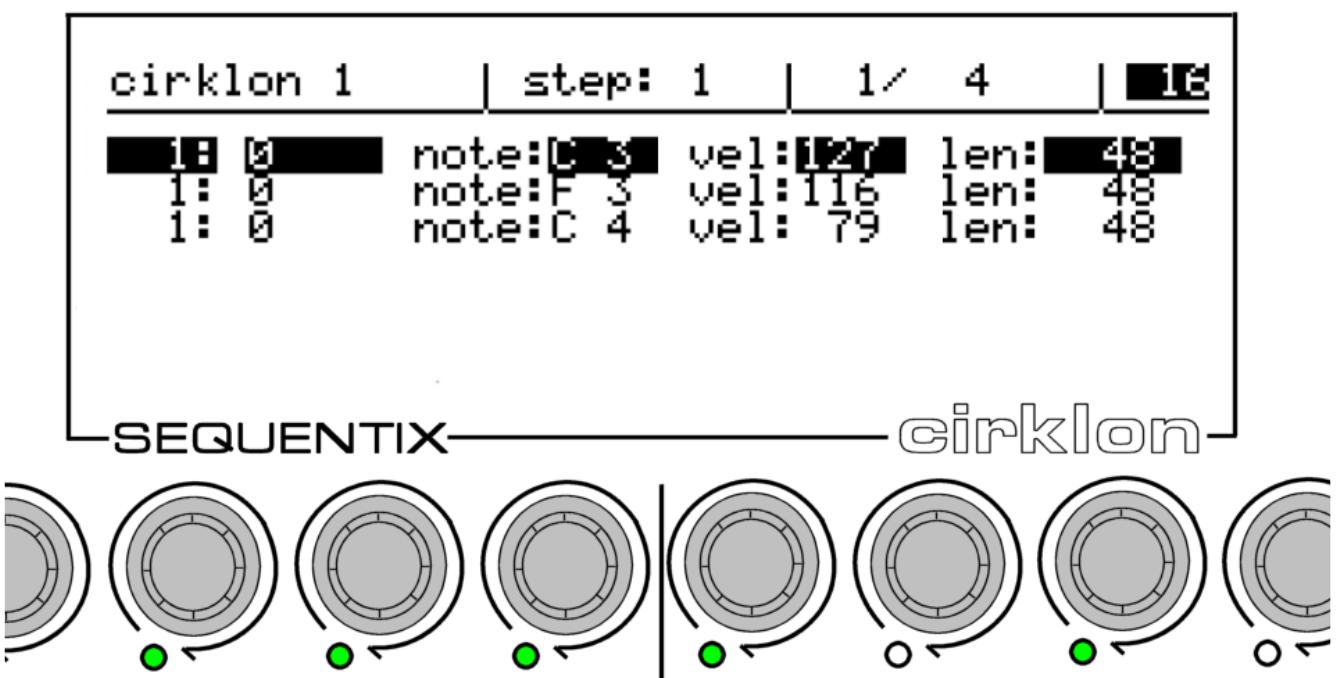
To prevent clashing between existing and new CC events, as soon as a new value has been entered using knob A, following events will be automatically deleted, unless you exit record mode, stop playback, or once the pattern loops.

11. CK Event List Edit

Since drum grid edit is optimised for creation and editing of percussion patterns, and does not include the means to edit other event types than notes and CCs, a low-level event list view is provided.

You can jump straight to event list edit by holding SHIFT as you press the PATTERN key to edit a CK pattern. Once in CK pattern edit, you can switch back and forth between drum grid and event list edit by pressing the PATTERN key.

A typical view in event list edit may look like this:



The top line of the display shows the current pattern name, step view position, bar position and bar length, and the view timebase setting.

Event values are edited using the encoders below each of the values shown on the display. The first two values for all events are the step position and delay value (ticks offset from the step), edited using encoders 6 and 7.

The range of the delay value depends on the view timebase setting.

For example, timebase 16 has a step period of 48 ticks, so the tick offset can be -23 to + 24.

Unlike in grid view, if the delay value is edited beyond the limits of the current step, the step value will change, and the step view position will follow such that the same event continues to be selected for editing in its new step location. You can freely move an event around in the pattern by editing its step or tick values.

The other lit encoders edit the remaining event values.

For a note event, encoder 8 edits the note value, encoder 9 the velocity value, and encoder 11 the note length.

Other event types have fewer values, but are edited in the same way.

You can scroll through the event list using the ROW encoder. Scrolling beyond the list for the current step position will advance the view to the adjacent step.

The step keys can be used to select the step view position directly within the current bar, but they can not be used to modify events.

As in drum grid edit, the grid timebase setting is adjusted using the VALUE encoder, and the bar length with a combination of the LAST key and the BAR encoder.

The selected event can be deleted by pressing the DELETE key.

A new event can be inserted using the INSERT key.

By default, note events are inserted. To select a different event type for insertion, hold the SHIFT key and press INSERT. A popup message will show the newly selected event type, like this:



Use the INSERT key as before to insert the new event type in the current step period.

Event list edit would be a laborious way to create a pattern from scratch, but can be very useful for making minor changes to patterns recorded by real-time input.

12. Real- and Step-time record

It is possible to record in real-time into both P3 and CK patterns from an attached MIDI keyboard or other controller. The differences between the two pattern types mean there are some differences in the record options depending on which type is being recorded into.

In both cases, real-time record is enabled by pressing the record key. The record mode settings are found on a special menu, which will appear while record is active after pressing the MENU key.

The current edit track selection determines which pattern will be recorded.

Also, if the FILL pattern field is selected on the TRACK page, recording will apply to the fill pattern.

CK Pattern Record

CK patterns can record un-quantised note, controller, channel pressure (aftertouch), pitch-bend, poly pressure and program change messages.

A record filter option allows you to limit which event types will be recorded.

Recording is possible from the play pages, and pattern edit modes. The same record menu will appear in both, as shown here:



The "RECORD" option can be switched between "live" and "ready".

The "ready" mode does not record any events into the pattern – it is there so you can audition what you plan to record before you go for a take.

Quantised Recording

Pressing encoder 5 toggles between the options RECORD and Q-RCRD (as shown above encoder 6), which corresponds to 'natural' or 'quantised' recording.

The “mthru” option allows you to switch off soft-thru of notes as they are being recorded. The soft-thru behaviour in record is also affected by the “record enable” options in the MIDI Config page – the *thru* option must be enabled at the port level for soft-thru to work. See the MIDI Configuration section for details.

The “click” function provides a metronome reference by sending MIDI notes. You can configure exactly which notes are sent, their velocity, timing and output MIDI channel using the click config page.

This page is accessed by holding SHIFT while pressing encoder 8 under the “click” option:



Note and velocity A are sent on the first beat of each bar.

Note and velocity B are sent on subsequent beats. The number and spacing of the beats is set by the “tbase” option.

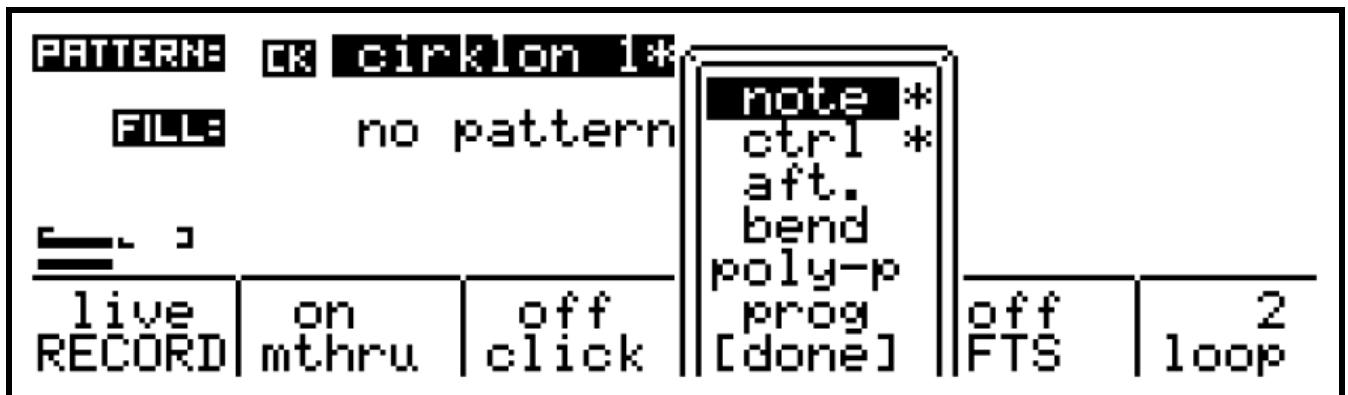
Finally, the MIDI option sets the port number and channel on which the metronome signal should be sent.

The next record menu option is “filter”.

This is an **inclusion** filter – only events which are **enabled** in the filter list will be recorded.

The text above the “filter” label gives a shorthand indication of which events will be recorded. Turning the encoder will pop-up a more detailed list of the event types.

You can scroll through the list, and press the encoder to enable or disable recording of each event type:



Scroll down to [done] or press the MENU key to hide the filter selection list.

When enabled, the record FTS option will apply the current Force-to-scale settings to incoming notes as they are recorded into the pattern.

Finally, the “loop” option allows you to adjust the bar length of a CK pattern without having to go into pattern edit.

Two further things to note about CK pattern record:

If you enable record mode while the sequencer is not running, it will automatically start on receiving the first incoming note.

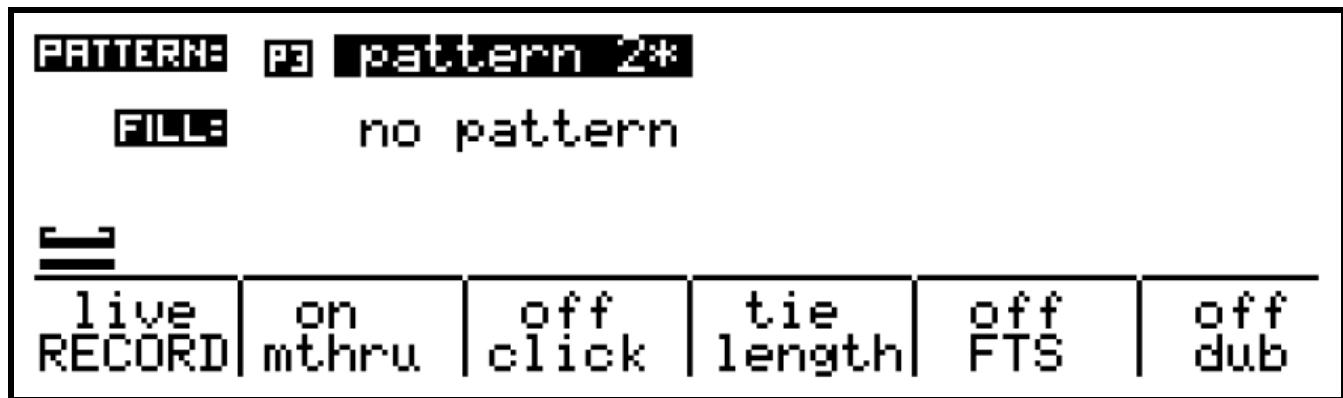
While record is active, the DELETE key can be used to mute existing notes in the pattern as they play.

CK ‘Tap’ Note Entry

While record mode is active in the CK pattern edit mode, you can use the SCULPT key to enter new notes on the current row at the current playback position. The velocity of the new notes is set by the position of knob A and their length is set by the duration of the keypress.

P3 Record

The record menu options for P3 patterns are as shown:



The "RECORD", "mthru" and "click" options are the same as for CK recording.

Once again, encoder 5 toggles between the options RECORD and Q-RCRD, which corresponds to 'natural' or 'quantised' recording.

It means that even the ordinarily tight/quantised P3 patterns can capture the nuances of live timing – i.e. the RECORD option writes each step with a STEP DELAY, ranging from -47 to +47 ticks. Those who prefer the traditional tight quantise should choose Q-RCRD.

The "length" option allows you to choose how notes which are held for longer than one step of the pattern are handled.

There are three options here, *off*, *tie* and *long*.

off - holding a single note over multiple steps will record a repeating pattern of single notes on every step, and overlapped notes will be recorded with a gap between them.

tie - when a note is held over more than one pattern step, each step it passes will be set to that note, with the tie flag turned on. On playback, this will give you a continuous note across the recorded steps. If a new note overlaps the previous note, the previous step will have the tie setting turned on so the notes also overlap on playback.

long - when a note is held over more than one pattern step, the note value is written only to the current step when the note is first played, but the length value will be increased for as long as the note is held, or until the starting step loops back round.

The "dub" option changes the way notes will be recorded into a pattern which already has notes active on some steps. It can be set to *off*, *note* or *gaps*.

off - existing notes are ignored and will be over-written

note - the gate flags are left unchanged, but the note and velocity values in each step are over-written by any note held on the MIDI master keyboard.

gaps - new notes will only be recorded only into spaces in the existing pattern. If there is already a note on a step, it will be left intact.

If aux note events have been configured for a P3 pattern, keyboard input can be recorded polyphonically into them.

P3 Step-time record (Arp capture)

Step-time record for P3 patterns is accessible only in P3 pattern edit.

It is accessed by holding SHIFT while pressing the record key.

The record LED will light and flash in this mode.

Step-time record allows you to enter notes from a MIDI keyboard into consecutive steps of a pattern, with the input position advancing after each note is entered.

Notes can be entered with the sequencer stopped or running, and the timing of input notes has no effect on their recorded position.

Ties and rests can be entered manually from menu soft-keys.

The step-time record menu looks like this:



There are three different modes for step-time input, selected by pressing encoder 6 – the MODE option on the menu. Turn the encoder then press to select one of the other options.

A selection menu will appear:



The three choices are *step*, *poly* and *arpeg*.

step - notes will enter each step, starting from the current input position, in the order they were received. The previous notes do not need to be released for each following note to be recorded, which makes it possible to enter a group of notes by playing a chord, which will then play back as an arpeggio.

poly - notes will enter each step, as with step mode, but if multiple notes are played together, **and the pattern has been configured with aux rows assigned to play extra notes**, the chord played will be entered in the current step. All notes must be released before input will advance to the next step. This allows polyphonic step-time input of chords or single notes.

arpeg - this mode is the same as the monophonic step mode. However, whenever a new note or chord is played after previously recorded notes have been released, the input position is reset to the first step in the pattern, and the last-step value reset so that only the newly entered notes will be played. The result is that this mode behaves in a similar fashion to a non-programmable arpeggiator.

The step record input position is shown at the right and centre of the display. This shows the bar and step number that the next note received will be recorded into. You can manually force this position to any point in the pattern by selecting that step with its step encoder. Note that you must hide the record menu to do this, by pressing the MENU key. When the menu is hidden, step-time record will remain active.

Returning to the other menu options...

The “mthru” option determines whether incoming notes will be soft-thru’d or not, as with the normal record mode.

The “clear” option resets the last step of the bar to step 1, sets step 1 to skip, and resets the next step to be captured to bar 1, step 1.

This has the effect of creating a silent pattern, ready for new notes to be entered from the beginning.

As notes are entered, the last step value for the bar will be increased to include the new notes if required, and extra bars may also be added.

The precise order the notes are played in determines their order in the captured arpeggio - so by controlling the order you press the notes in the chord entered, you can control the nature of the resulting arpeggio.

Or you could use some of the other P3 direction modes for a more random or varied arpeggio pattern.

The “tie” function will cause the most recently entered note to be copied to the next step, and the two steps tied for a longer note.

To exit step-time record, press the record key, or exit from P3 pattern edit mode.

13. P3 Patterns & Aux events

Having seen earlier how to assign an **auxiliary event** to one of the auxes in a pattern and enter values for it, let's look at what you can actually do with the many **events** in P3 patterns.

Aux events first appeared in the Sequentix P3 Sequencer.

The implementation in Cirklon is very similar, though subtly different in some ways.

The same can be said of this section of the manual, so it is worth reading through in case you miss the differences.

In essence, an auxiliary event is like a controller message that is sent internally to modify some setting or value within the P3 pattern playback engine:

- They can alter the current values for each playing pattern – note, velocity, length, delay, aux and status values.
- They can change pattern values like timebase, direction and last step.
- They can change global values, such as the current tempo and track mutes.
- They can also access normally “hidden” values that affect patterns in unusual ways.

Some of the events are more difficult to understand than others, so let's look at an example of the sort of thing you can do quite easily, before looking at some of the trickier stuff.

Randomizing Pattern Values

The **aux events** are divided into a number of different **event groups**, one of which is called “**Randomize**”.

The Randomize group allows you to add a random amount to any of the numeric pattern values.

Or you can randomly mask the pattern status values, which just means that you can randomly turn off active status values on some steps.

A Boring Pattern...

Let's work through an example of applying some randomization to pattern values.

The table below shows a simple pattern.

If you are familiar with P3 pattern edit, you should be able to enter it quickly.

If you're not too sure how to go about entering the pattern, you may not be quite ready for this chapter.

Example pattern:

Step	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Note	C4															
Velocity	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	
Length	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	
Gate	on															

It has to be said, this is a pretty boring pattern.

Now, let's assign **aux A** in your pattern to the event "**Rndmz note**" (short for "randomize note").

To do this:

- use the **ROW** encoder to select **aux A**, if not already selected
- press, hold and turn the **ROW** encoder to access the aux selection list
- press, hold and turn the **ROW** encoder again to access the aux group list
- scroll to the *Randomize* group, and select it by pressing **ROW** once more
- the "*Rndmz note*" event is first in the group, so just press **ROW** to select it

Now you can enter some values for the event.

So you can hear the effect right away, start Cirklon running if it isn't already.

In a new pattern, the aux values in every step will be zero.

Turn encoder 1 clockwise to increase the value on step 1 to 12.

Normally, editing an aux value will also activate the aux on that step.

If you have turned off the "aux edit sets flag" user config option, you may need to turn on the step manually by pressing the step key.

Listen to the pattern play now, and **each time step 1 plays**, it will play a **random note**, somewhere **between C4 and C5**.

The rest of the notes in the pattern will still play the same note – C4.

The note value for step 1 also remains at C4.

It is only the transmitted note value that is being changed by the randomize event.

How Events Work

The “Rndmz note” event, like all events that modify pattern values, changes the value of note **dynamically**.

The stored value in the pattern is not changed.

When a pattern step with the “Rndmz note” event enabled is played, the following happens:

1. the **note** value is read from the pattern
2. the **gate** status is read from the pattern
3. the **aux A** value is read from the pattern
4. because **aux A** is assigned to “Rndmz note”, a random number is generated between zero and the stored value for aux A, and this random number is added to the **note** value
5. if **gate** is on, the **note** is sent

Some extra steps are not described there to keep it simple – obviously the velocity, length, and so on are also read from the pattern, but you get the idea.

You can enable and set a different value for aux A on whichever steps you like, and a different random number will be generated and added to note on each step.

Try setting aux A on some other steps – some with a small value of only 1 or 2, others with larger values.

You will hear the different amounts of random variation on each of the steps you modify as the pattern plays.

It may seem a little uncontrolled, and indeed, out of tune.

You might find the Force-To-Scale feature useful – you can randomize note values, but use FTS to ensure they play in key.

Random Masks

There are events like “Rndmz note” for each of the other numeric pattern values – velocity, length, delay and auxes B, C and D.

All will add a random number between 0 and the aux value to the chosen pattern value.

For the status values, **gate**, **tie**, **X** and the **aux flags**, the randomize events are a little different.

In the example pattern, assign **aux B** to the event “Rnd mask gate”, as follows:

- use the **ROW** encoder to select **aux B**
 - press, hold and turn **ROW** to access the aux selection list
 - press, hold and turn **ROW** again to access the group list
 - select the *Randomize* group
 - select the “*Rnd mask gate*” event
-
- use step encoder 1 to set a value of **75%** for “*Rnd mask gate*” on **step 1**

As you listen to your pattern now, you will hear that sometimes the note on step 1 plays, and sometimes it is missed out.

If you keep a count of how often it plays and how often it doesn’t, you should find that **on average** the note plays three out of every four times.

There is not a regular pattern – the value sets the **probability** that a step **with gate already active** will **remain active** when the mask event is applied.

When a pattern step with the “*Rnd mask gate*” event enabled is played:

1. the **note** value is read from the pattern
2. the **gate** status is read from the pattern
3. the **aux B** value is read from the pattern
4. because aux B is assigned to “*Rnd mask gate*”, a random number is generated between 1 and 100. If the **stored value** for aux B is **less than the random number**, the **gate status is turned off**
5. if **gate** is on, the **note** is sent

Try setting the value for “*Rnd mask gate*” on some other steps.

Only a step with gate already active will play – masking will not turn on a step that is off.

The value for the event is the **probability that the gate will be active** on each step:

- a low value means a low probability of the gate being active
- a high value means a high probability of the gate being active

For example, if you set a value of 25% on a “*Rnd mask ...*” event, the status value on that step will be active once in every four times that the step plays.

Using random masking like this, you can create patterns that subtly change over time, sometimes throwing in an extra note here and there.

Experiment...

Try assigning some of the other randomize events to your pattern.

Bear in mind that you will need to trigger a sound that makes changes to the pattern values audible – there's not much point randomizing the length of a note if the sound has a long release time that will mask the change.

Things like subtle randomization of velocity can add a degree of interest to an otherwise static pattern.

Inter Track Events

The next group of events we'll look at are a little more complicated to understand than the Randomize group, but they are also capable of much more interesting effects.

The group name “**Inter Track**” refers to the fact these events cause interaction **between different tracks**.

The first three events in the group are for **moving values** between two different tracks. The two tracks involved are referred to as:

- **current track** – the track with the pattern that is actually triggering the event
- **target track** – the track specified by the event as the track to exchange values with

The first three events are:

- **grab** - replaces a value on the **current track** with a value from the **target track**
- **swap** - swaps a value between the **current track** and the **target track**
- **push** - replaces a value on the **target track** with a value from the **current track**

Let's work through an example to explain what these events can do.

Before you can move data between tracks, you will need P3 patterns to work with on two tracks, so let's set them up now.

Set up **example pattern A** on **track 1**, and **example pattern B** on **track 2**, as shown below.

Example pattern A:

Step	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Note	C4															
Velocity	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	
Length	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	
Gate	on															

Example pattern B:

Step	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Note	C5	D5	E5	F5	G5	A5	B5	C6	D6	E6	F6	G6	A6	B6	C7	D7
Velocity	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64
Length	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
Gate	on															

Pattern A is the same boring, repeating-note pattern we used before.
Pattern B is an equally boring rising scale.

A Quick Grab

To demonstrate an **Inter Track** event, let's use the first event in the group – “grab”.

Having entered both of the example patterns above:

- start Cirklon running, if it isn't already
- ensure **track 1** is **active** – it will be playing **pattern A**
- **mute track 2** – it will be playing **pattern B**, but **no notes will be sent**

The reason you have muted track 2 is that it will be easier to hear what is going to happen to **pattern A** without **pattern B** also making a noise.

Enter pattern edit for track 1 (pattern A), and use the **ROW** encoder as before to select the aux A row, and assign the *grab* event from the *Inter Track* group.

Auxes With Double Values

Unlike other auxes, the values in each step for the *grab*, *push* and *swap* events are split in two parts.

First is the **target value** which will be grabbed, swapped or pushed.

The second part is the **target track** that the event will affect.

The target value can be any of the row values in a P3 pattern, and some special cases, which will be listed in full below.

The target track can be from **1** to **16**.

Higher numbered tracks cannot be specified, though the inter track events can be used in patterns on higher tracks, so long as they target tracks in the first 16.

As you turn the encoders to edit the event values on each step, they will step through each track.

By holding in the step encoder and turning (as normal for bigger steps), the **target value** will be changed.

Use the step encoders to set steps 1, 5, 9 and 13 to a value of “note trk: 2”.

Now as your previously boring pattern plays, you will hear different notes play only on these steps.

This table shows what is happening:

Step	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Pattern A Note	C4															
Aux A enabled	*				*				*				*			
Pattern B Note	C5	D5	E5	F5	G5	A5	B5	C6	D6	E6	F6	G6	A6	B6	C7	D7
Track 1 Plays	C5	C4	C4	C4	G5	C4	C4	C4	D6	C4	C4	C4	A6	C4	C4	C4

On each step where **aux A** is **enabled**, and set to **grab** the note from track 2, the **current note value** from the pattern playing on **track 2 replaces the current note value** from the pattern on **track 1**.

This gives a number of possibilities for different effects.

If you set every step in pattern A to grab the notes from track 2, track 1 would play **all the same notes** as track 2.

Only the note values are being grabbed, so if you were to turn off some of the gates in pattern A, it would double-up some of the notes in pattern B on the track 1 instrument, not play all of them.

This is one use for note grabbing.

Another interesting use is to grab notes from a pattern with a different length.

Return to the example grab patterns, go into **pattern edit** for **pattern B** on **track 2**, and change the **last step** to step **15**.

As pattern A plays, it will still grab notes from pattern B.

But since the pattern lengths do not match, the notes that are grabbed will shift around on each pass of the pattern.

With just a few notes grabbed, track 1 will play a pattern that mostly repeats, but chosen steps will run through a changing sequence of notes over a much longer period.

For another effect, edit pattern B and set the **direction** to **random**.

The notes grabbed by track 1 are different every time – but always randomly chosen from the selection of notes in the pattern on track 2.

This can be a much more selective way of randomising notes in a pattern than using the basic “Rndmz note” event.

Experimentation is the key here again.

As you play with these events, you should soon realise some of the apparently complex modulations of patterns that can be achieved quite simply.

Target Values For Grab, Swap and Push

Having looked at the grab event, the **swap** and **push** events should not be any great mystery.

Push does the same thing as **grab**, but in the **opposite direction** – the **target value** on the **current track** replaces the value on the **target track**.

Swap essentially does a **grab** and a **push** at the same time – the **target values** on the **current track** and **target track** are swapped.

You can experiment with the differences between these events quickly by re-assigning the auxes in your example patterns. The pairs of step values for one event will work for the others.

Here is the full list of **target values** that can be acted on:

- note
 - velo
 - leng
 - dlay
 - gate
 - tie
 - X
 - all (all pattern values are replaced)
 - nte+A (current note value plus any accumulator offset)
 - nte+B (current note value plus any bar transpose)
 - vel+A (current velocity value plus any accumulator offset)
 - aux B
 - aux C
 - aux D
 - xD+A (current aux D value plus any accumulator offset)

Transpose By Events

Continuing with the **Inter Track** event group, the remaining events in this group are:

- **xpose by trk n**
- **xpose by trk n+B**

These two events allow you to **transpose** the **note value** on the **current track**.

The transposition is performed relative to middle C, which means that if the **note value** on the **target track** is middle C, the note on the **current track** is **not transposed**.

To demonstrate these events, you can use pattern A and pattern B from the grab example again.

Instead of assigning aux A to “grab”, assign it to “xpose by trk n”.

If you set **aux A** active on steps **1, 5, 9** and **13**, with a value of **2**, you will get the following result:

Step	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Pattern A Note	C4															
Aux A enabled	*				*				*				*			
Pattern B Note	C5	D5	E5	F5	G5	A5	B5	C6	D6	E6	F6	G6	A6	B6	C7	D7
Track 1 Plays	C4	C4	C4	C4	G4	C4	C4	C4	D5	C4	C4	C4	A5	C4	C4	C4

On step 1, track 1 plays the original note from the pattern.

This is because the note value on track 2 is C5 – middle C, so there is no transpose.

On step 5, track 1 plays G4. This is the original note - C4, transposed up by 7 semitones.

The transpose is the difference between the note on track 2 – G5, and middle C – C5.

Check the results on the last two enabled steps and you’ll see the same rule applies.

To demonstrate a more practical use of the transpose event:

- enable **aux A** with a value of **2** on every step of **pattern A**
- change the **timebase** on **pattern B** to **1**

This will show a useful feature of the way aux events work.

With the timebase of pattern B set to 1, pattern B will only play 1 step in the same time it takes for pattern A to play all 16.

But the note value from pattern B that is used to transpose pattern A remains current for as long as it takes the next step to play.

The result is **pattern A** will play for its **whole length, transposed** by the **note value on step 1 of pattern B**.

Then **pattern A** will play for a second time, **transposed** by the **note value on step 2 of pattern B**.

And this will continue for **all 16 steps of pattern B** - giving you a means to sequence 16 repeats of pattern A, each one with a different transposition.

Plus B

The “**xpose by trk n+B**” event is much the same as “**xpose by trk n**”.

The “**+B**” is there because the **note value** used from the **target track** is taken after any **bar transpose** is applied.

Aux Notes

The “**Aux note**” event group is different to the others since the events do not change any values.

Each aux that is assigned to an **aux note** event can play one **extra note** on each pattern step, in addition to the main note value.

If you assign all four auxes to aux notes, your pattern will be **5 note polyphonic**.

There are four events in the Aux note group:

- **aux note abs**
- **aux note rel**
- **aux note abs xF**
- **aux note rel xF**

In these events, “**abs**” means **absolute**, and “**rel**” means **relative**.

For the **absolute** events, you enter the extra note as a normal **note name**.

The **relative** events take a **number** value that is an offset from the main pattern note.

The “**xF**” versions of the events send the extra notes without applying any Force-To-Scale settings.

Note that no aux notes play unless the step gate is on.

For example, let’s say you want to send a C major 7 chord on step 1 of a pattern using four notes:

- set the **note** value on step 1 to C5
- assign **aux A** to the event “**aux note abs**”
- assign **aux B** to the event “**aux note abs**”
- assign **aux C** to the event “**aux note abs**”
- set the **aux A** value on step 1 to E5
- set the **aux B** value on step 1 to G5
- set the **aux C** value on step 1 to B5

When the pattern step plays, the four note chord C5, E5, G5, B5 will be sent.

All notes share the same **velocity** value.

Let's do the same thing, using the "**aux note rel**" event:

- set the **note** value on step 1 to C5
- assign **aux A** to the event "**aux note rel**"
- assign **aux B** to the event "**aux note rel**"
- assign **aux C** to the event "**aux note rel**"
- set the **aux A** value on step 1 to 4
- set the **aux B** value on step 1 to 7
- set the **aux C** value on step 1 to 11

In the case of **relative** events, the note sent by each aux is the main note shifted by the aux value.

In this example, with a main note of C5:

- C5 + 4 semitones = E5
- C5 + 7 semitones = G5
- C5 + 11 semitones = B5

...and you have the same notes as you set for the "aux note abs" event.

The difference with the **relative** events is that you can now transpose all the notes in the chord just by changing the main note value on the step.

Chord Memory

It is possible to use the **aux note rel** events to recreate the **chord memory** function found on early polyphonic synthesizers.

This means you can set up the aux values in advance to create a given chord shape on every step of a pattern.

Then you can set the root note of your chord using the normal note value, and activate the gate.

Polyphonic Real-time Record

If you have configured some of the auxes in a P3 pattern with any of the aux note events, you can record polyphonic patterns directly from an attached keyboard.

The lowest note played in any step will be written to the main note value, and up to 4 more notes will be written into the auxes.

Accumulator Events

All the events looked at so far have had one thing in common – their effects only apply to the individual steps that they are enabled on.

The “**Accumulator**” event group provides another set of events which can alter pattern values, but they do so in a way that continues to affect the pattern even once it moves on to other steps.

These events are able to make lasting changes to the way a pattern plays because they give you access to some hidden values that affect pattern playback.

These hidden values are the **accumulators**.

What Is An Accumulator?

According to the dictionary, an **accumulator** is simply something that **accumulates** – that is, it increases gradually in quantity or number.

In the ways that the accumulators are typically used in P3, this is exactly what they do.

There are three accumulators for each track:

- **note accumulator**
- **velocity accumulator**
- **aux D accumulator**

What they do is very simple.

The three accumulators are each named for one of the pattern values – **note**, **velocity** or **aux D**.

Whenever the **note** and **velocity** values are used to send a MIDI note:

- the number held in the **note accumulator** is **added** to the **note value**
- the number held in the **velocity accumulator** is **added** to the **velocity value**

So you can really think of the **note accumulator** as just another **transpose** setting – with the same effect as scene or bar level transpose.

The **velocity accumulator** is similarly just a “transpose”, or a **numerical offset**, for the **velocity**.

When the aux D value is used to send a MIDI controller, or if aux D is used for an aux event:

- the number held in the **aux D accumulator** is **added** to the **aux D value**

Like the other accumulators, it's just a numerical offset that applies to the aux D value.

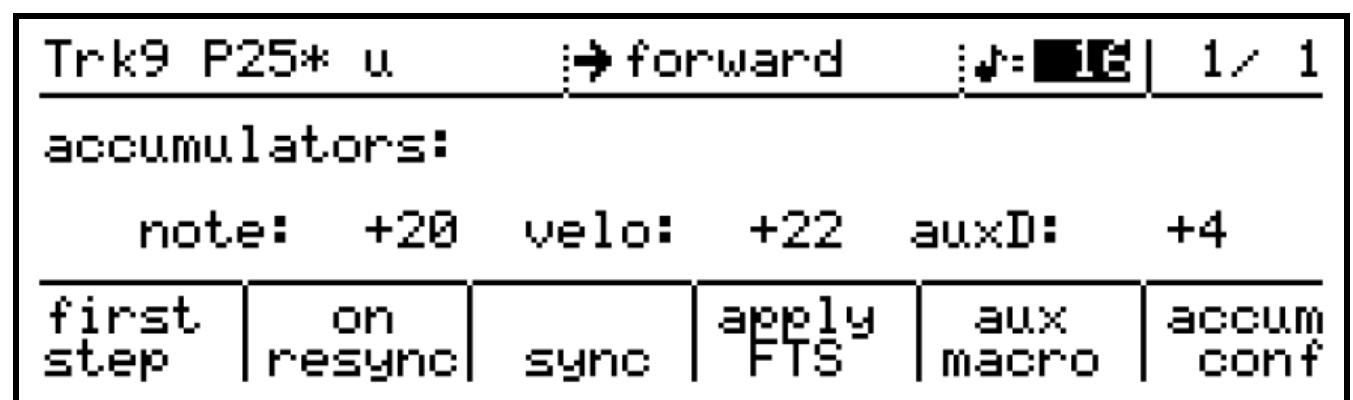
The important thing to realise is that **the accumulator values are added to the pattern values on every step of the pattern**.

Normally, the accumulators have no effect.

This is because, whenever Cirklon is **started**, or a **new pattern is selected** on a track, the **accumulators are reset to zero**.

So unless you choose to change the values in the accumulators, they remain at zero, and have no effect.

As a helpful guide to what's going on, you can see the accumulator values at any time for the current track by pressing the MENU button from within pattern edit:



Putting An Accumulator To Work

Let's look at an example to get an accumulator doing something useful.
We'll start with the note accumulator.

Here's an example pattern to work with.

It is only 4 steps long, so make sure you set the **last step** to 4.

Example pattern:

Step	1	2	3	4
Note	C5	E5	G5	C6
Velocity	64	64	64	64
Length	24	24	24	24
Gate	on	on	on	on

Enter the pattern, and start it playing, and you should hear a simple rising arpeggio.

Let's configure aux A with an event that will affect the note accumulator.

In pattern edit for the example pattern, use ROW to select aux A.

The press, hold and turn ROW twice to access the group list, and select the "Accumulator" event group, then select the first event in the group - "**offset note abs**".

Using step encoder 3, enter a value of 5 for the event.

The next time the pattern passes through **step 3**, you will hear the pitch of **all following notes** in the pattern rise by 5 semitones.

From first starting the pattern playing, the result is like this:

Step	1	2	3	4	1	2	3	4	1	2	3	4
Stored Note	C5	E5	G5	C6	C5	E5	G5	C6	C5	E5	G5	C6
aux A	off	off	5	off	off	off	5	off	off	off	5	off
Played Note	C5	E5	C6	F6	F5	A5	C6	F6	F5	A5	C6	F6

For the **first two steps after starting**, the pattern plays the notes just as you'd expect. After the first time step 3 is reached, the notes all rise in pitch by 5 semitones.

What has happened is simple.

The "offset note abs" event puts the **aux value** into the **note accumulator**.

At step 3 in the example pattern, a value of **5** is put into the **note accumulator**.

This value will remain in the accumulator until a new pattern is selected, or Cirklon is restarted.

All notes played by a pattern have the **current note accumulator** value added to them. So **all the notes** in the example pattern are **raised by 5 semitones** after step 3.

It doesn't matter that the pattern reaches its last step and repeats – the accumulator value will remain.

As an experiment, with the pattern still playing, change the value for **aux A** in **step 1**.

You will hear the pattern continue with the same notes until it passes **step 1**, at which point a new value is put into the note accumulator, so the following steps will all be transposed by the new amount.

That is, until the pattern reaches step 3 again – where the number 5 will be written back in.

Turn off **aux A** on **step 3**, and only the value from **step 1** will apply to the notes.

This shows that the current accumulator value:

- applies to every step in the pattern
- only changes when an enabled aux event puts a new value into the accumulator

If you've followed this so far, you may be thinking that it's handy you could transpose small sections of a pattern with a couple of "offset note abs" events, but it's not exactly earth shattering.

And you could always achieve the same thing by editing the notes.

The real power of the accumulators is only apparent when you use them in the way that their name suggests – increasing gradually in quantity or number.

Relative Accumulation

Let's go back to the example pattern.

Turn off any enabled **aux A** steps you have set, and re-assign **aux A** to the "**offset note rel**" event.

Then set the value on step 1 of 1.

Each time the pattern plays, the four notes will rise in pitch by 1 semitone compared to the last pass of the pattern.

Eventually, the pitch offset will disappear and the notes will play at their stored pitches. But on the next pass, they will start to rise again.

The reason for this behaviour is because the "offset note rel" event is **relative**.

The **aux value is added to the current accumulator value**.

For the example pattern, this is the result:

Step	1	2	3	4	1	2	3	4	1	2	3	4
Stored Note	C5	E5	G5	C6	C5	E5	G5	C6	C5	E5	G5	C6
aux A	1	off	off	off	1	off	off	off	1	off	off	off
Note Accumulator	1	1	1	1	2	2	2	2	3	3	3	3
Played Note	C#5	F5	G#5	C#6	D5	F#5	A5	D6	D#5	G5	A#5	D#6

Each time the “offset note rel” event on **step 1** occurs, the **note accumulator** value is increased by **1**.

The **note accumulator** value is **added** to the **note** on **every step**, so each pass of the four steps rises in pitch by 1 semitone compared to the last pass.

Eventually, a **limit** is reached, at which point the note accumulator value is **reset to zero**. But the pattern continues as before.

Now change the value for **aux A** on **step 1** to **12**

Your pattern of 4 notes will now rise by an octave on each pass, and you’ll find that the accumulator is reset when it reaches 3 octaves of transposition – but that is only a default setting.

Accumulator Configuration

To give you more control over the things they can do, each of the accumulators has some configuration settings.

These settings let you:

- limit the maximum value an accumulator can reach
- tell it how to behave when it reaches that value
- tell it how to behave when adding the accumulator value to one of the pattern values gives a result that is out of range for that value

The **accumulator configuration** page is accessed from the pattern edit menu. Press the MENU key, then encoder 11, under the “accum conf” label.

This should present you with a display that looks like this:

Pattern accumulator settings:					
note	36	rtz	R:clip		
velo	127	rtz	R:clip	on	off
aux D	127	rtz	R:clip	on	off
accum	limit	mode	out	RoPS	XdAcD

Encoder 6, labelled “accum”, is used to select which accumulators settings are to be edited.

The **limit** setting is the **maximum value** that the accumulator can hold.

Adjust the limit by turning encoder 7.

If you experimented with the step values on the “offset note rel” event you may have found that you can set a **negative** offset – transposing notes downwards.

If you start from zero and apply negative offsets, then the accumulator value can become negative.

For this reason the **limit** value for the accumulator is **absolute** – that is, the limit is for positive and negative numbers.

For example, if you set **limit** to **10**, **positive** offsets will be able to raise the value to **+10** and **negative** offsets will be able to lower the value to **-10** (minus 10).

In the current example pattern, when the note accumulator reaches 3 octaves of transpose, it is cleared.

The **default limit of 36** gives you 3 octaves range on the note accumulator – the limit is given in semitones.

You can set the limit anywhere between **0** and **127**.

The **mode** setting tells the accumulator what to do when it **exceeds** its limit.

The default is **rtz** – which stands for “**return to zero**”.

So when the example pattern tries to increase the **note accumulator** value beyond its limit of **36**, the **accumulator value is reset to zero**.

There are four possible settings for **mode**:

- **rtz** - when the value exceeds the limit, the accumulator is **reset to zero**
- **clip** - the value is restricted to the limit, and will go no further
- **rvtz** - reverse to zero
- **rvbp** - reverse bi-polar

Return to zero was explained above, and you can hear its effect in the example pattern.

The **clip** behaviour is the simplest – the accumulator value won't exceed the limit, it just sticks there once the limit is reached.

The **reverse** options will need a little more explanation.

Each **accumulator** has a **reverse switch**.

When an accumulator is cleared its **reverse switch** is also turned **off**, so they normally have no effect.

But if you configure the **mode** to either "**rvtz**" or "**rvbp**", when the accumulator reaches its limit, the **reverse switch** is **inverted** – that is, if the switch is off, it is turned on; or if it is on, it is turned off.

The effect of turning **on** the **reverse switch** is that any **relative offset** to the accumulator is **subtracted** from the accumulator value, rather than **added** to it - the effect of a **relative offset** is **reversed**.

Let's look at what this does to the example pattern.

You have a four note arpeggio, with "offset note rel" assigned to aux A.
Aux A is adding **12** to the **note accumulator** every time **step 1** is passed.

With the **default** accumulator config, the **note accumulator** rises to **36** (3 octaves up), then is **reset to zero**.

To change the **mode**, press the soft-key below it on the display – encoder 8.

Each press advances to the next option, so if you press encoder 8 twice, it will step from **rtz** to **clip** to **rvtz**.

This table shows what happens now:

Step	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Stored Note	C5	E5	G5	C6	C5	E5	G5	C6	C5	E5	G5	C6	C5	E5	G5	C6
aux A	12	off	off	off	12	off	off	off	12	off	off	off	12	off	off	off
Note Acc	12	12	12	12	24	24	24	24	36	36	36	36	24	24	24	24
Reverse sw.	off	on	on	on	on											
Played Note	C6	E6	G6	C7	C7	E7	G7	C8	C8	E8	G8	C9	C7	E7	G7	C8

When the offset event on the step shown in **bold** attempts to push the accumulator beyond its limit, the reverse switch is turned on, and the offset value is subtracted from the accumulator.

The notes start shifting downward by an octave each time.

This will continue until the accumulator value reaches **zero**.

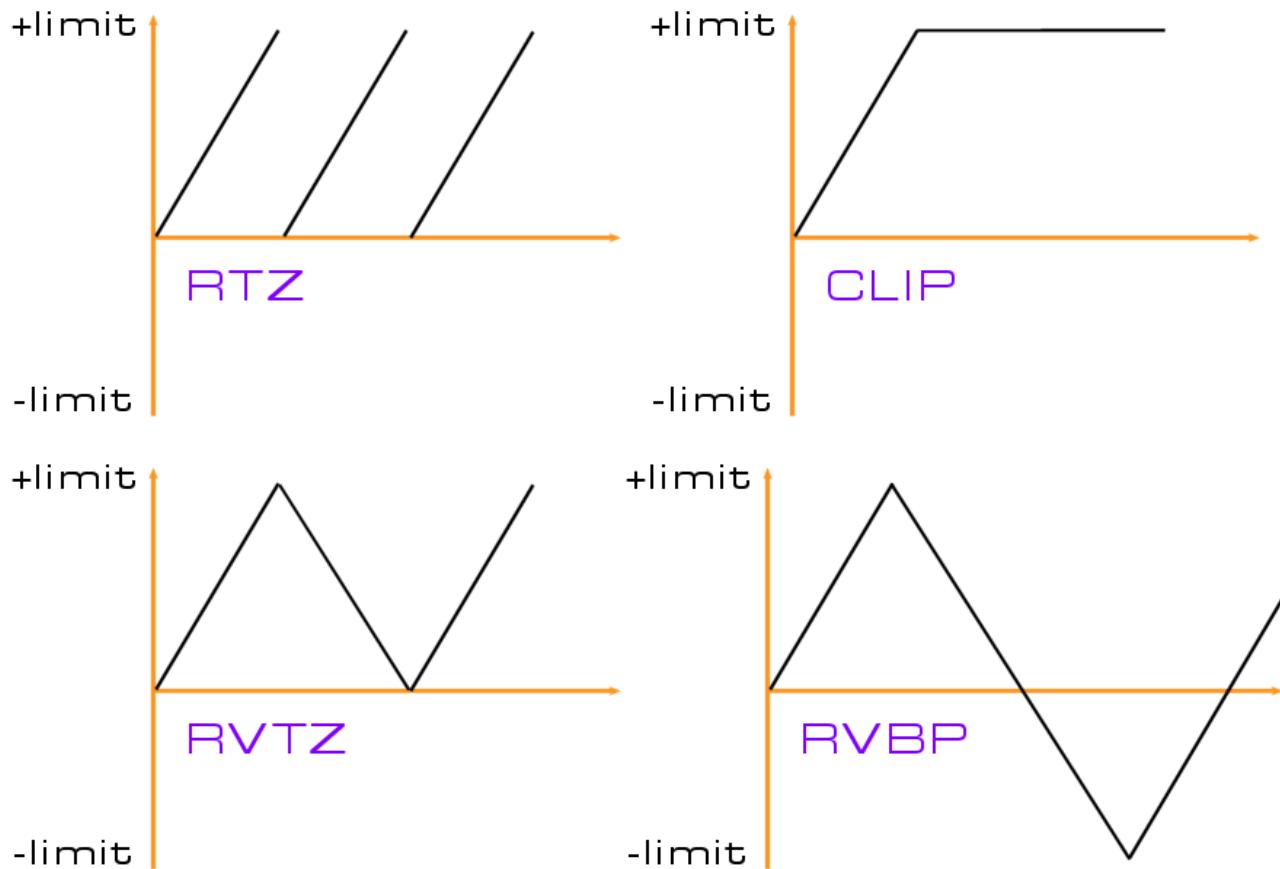
With the **mode** set to **reverse to zero**, when an offset event attempts to move the accumulator below zero, the **reverse switch** is turned **off**, and the accumulator starts to rise again.

The alternative reverse option, **reverse bi-polar**, would continue through zero.

It will allow the accumulator to go all the way to the **negative limit**, before the reverse switch is turned off again.

These behaviours may be simpler to understand with a diagram.

If you have a pattern with an offset event which adds 1 to the accumulator on each pass, the accumulator value will change as shown for each of the four **limit behaviours**:



Experimentation, as ever, is the way to learn how to use these best.

The last setting for accumulator config, **out**, determines what happens when the **result of adding the accumulator value to the pattern value is out of range**.

There are two options:

- **clip** - the result is not allowed to exceed the maximum or minimum possible value
- **wrap** - the result is folded back round to the opposite end of the value range

The **clip** option is the default.

You need to be careful when using the wrap option, since you can end up with very high or very low values suddenly changing to the opposite extreme.

With note or velocity values, this might give you a surprise.

These remaining two settings on the accumulator config page are pattern level settings.

The “**RoPS**” setting is short for “**Reset on Pattern Select**”, and is toggled with encoder 10. This setting is **on** by **default**, and means that all three accumulator values will be reset when the pattern is selected.

If **RoPS** is turned off, when the pattern is selected, any accumulator values changed by the previous pattern will be carried over.

This is useful for patterns that will be consecutively where you want accumulator modulation to continue.

The “**xDAcD**” setting is short for “**auX D Accumulator Disconnect**”, and is toggled with encoder 11.

This setting, **off** by default, allows you to break the normal link between the aux D accumulator and the aux D value – with **xDAcD** set **on** the value of **aux D is not changed** regardless of the aux D accumulator value.

This setting is provided so you can continue to use aux D for controllers or other events, while using the aux D accumulator for the “**Accumulator Mask**” events.

Auxiliary Value Scaling – No More

One significant difference between the behaviour of aux events on P3 and Cirklon is that the aux value scaling that applied in P3 no longer applies.

So let's move along...

The Accumulator Events

For each of the three accumulators, there is an absolute and relative offset event:

offset note abs	set note accumulator value	range: 0 to 127
offset note rel	add to note accumulator value	range: -127 to +127
offset velo abs	set velocity accumulator value	range: 0 to 127
offset velo rel	add to velocity accumulator value	range: -127 to +127
offset aux D abs	set aux D accumulator value	range: 0 to 127
offset aux D rel	add to aux D accumulator value	range: -127 to +127

Note that the relative events have a wider range of values than on the P3 Sequencer.

There are another set of events which take no value, but simply prevent each accumulator from affecting specific steps on which the event is set active.

The accumulators are not reset, and any offset events on the same step will still apply.

The accumulator defeat events are:

note acc defeat
velo acc defeat
auxD acc defeat

You may also want to dynamically modify the accumulator limits - overriding the values set for the pattern. This can be done using these events:

set note acc lim
set velo acc lim
set auxD acc lim

The last two events in the group allow the aux D accumulator to be used to modify the length and delay settings for each step.

add dAcc to leng	add aux D accumulator to length	no value
add dAcc to dlay	add aux D accumulator to delay	no value

Accumulator Mask Events

The “**Accumulator Mask**” event group use the value of the **aux D accumulator** as a **threshold** to control variations to a pattern:

Mask gate,dAcc>n	defeat gate bit unless xD accumulator > value	range: 0 - 127
Mask gate,n>dAcc	defeat gate bit unless value > xD accumulator	range: 0 - 127
Mask tie, dAcc>n	defeat tie bit unless xD accumulator > value	range: 0 - 127
Mask tie, n>dAcc	defeat tie bit unless value > xD accumulator	range: 0 - 127
Mask xB, dAcc>n	defeat aux B unless xD accumulator > value	range: 0 - 127
Mask xB, n>dAcc	defeat aux B unless value > xD accumulator	range: 0 - 127
Mask xC, dAcc>n	defeat aux C unless xD accumulator > value	range: 0 - 127
Mask xC, n>dAcc	defeat aux C unless value > xD accumulator	range: 0 - 127
mute trk, dAcc>n	deactivate track when xD accumulator > value, and reset accumulator to 0	

These events are used along with an “**offset aux D rel**” event.

For example, if you set up an offset event such that aux D increases by 1 each time the pattern loops, you can set certain steps not to be activated until the pattern has played for up to 127 times.

This allows for very gradual modification of a pattern as it plays.

Knob Mask Events

The events in the “**Knob Mask**” group use the current position of the front panel knobs A and B, or the **track control values**, **knob1** and **knob2**, to control masking of gate, tie, the aux statuses and accumulator defeats.

Refer to the chapter on **track values** for an explanation of how to assign and edit the knob1 and knob2 values for each track.

The knob source for each event is selected at the same time as you assign the event to an aux.

For each status value that can be masked, there are three events.

The first de-activates the status unless the **knob value is greater than the aux value**.

The second de-activates the status unless the **knob value is within plus or minus 3 of the aux value**.

The third de-activates the status unless the **knob value is less than the aux value**.

The knob controlled mask events are:

Mask gate,Kn>n	defeat gate unless knob is > n	range: 0 - 127
Mask gate,Kn~n	defeat gate unless knob is near n	range: 0 - 127
Mask gate,Kn<n	defeat gate unless knob is < n	range: 0 - 127
Mask tie,Kn>n	defeat tie unless knob is > n	range: 0 - 127
Mask tie,Kn~n	defeat tie unless knob is near n	range: 0 - 127
Mask tie,Kn<n	defeat tie unless knob is < n	range: 0 - 127
Mask auxB,Kn>n	defeat aux B unless knob is > n	range: 0 - 127
Mask auxB,Kn~n	defeat aux B unless knob is near n	range: 0 - 127
Mask auxB,Kn<n	defeat aux B unless knob is < n	range: 0 - 127
Mask auxC,Kn>n	defeat aux C unless knob is > n	range: 0 - 127
Mask auxC,Kn~n	defeat aux C unless knob is near n	range: 0 - 127
Mask auxC,Kn<n	defeat aux C unless knob is < n	range: 0 - 127
Mask auxD,Kn>n	defeat aux D unless knob is > n	range: 0 - 127
Mask auxD,Kn~n	defeat aux D unless knob is near n	range: 0 - 127
Mask auxD,Kn<n	defeat aux D unless knob is < n	range: 0 - 127
Mask nAcc,Kn>n	defeat note accumulator unless knob is > n	range: 0 - 127
Mask nAcc,Kn~n	defeat note accumulator unless knob is near n	range: 0 - 127
Mask nAcc,Kn<n	defeat note accumulator unless knob is < n	range: 0 - 127
Mask vAcc,Kn>n	defeat velo accumulator unless knob is > n	range: 0 - 127
Mask vAcc,Kn~n	defeat velo accumulator unless knob is near n	range: 0 - 127
Mask vAcc,Kn<n	defeat velo accumulator unless knob is < n	range: 0 - 127
Mask dAcc,Kn>n	defeat aux D accumulator unless knob is > n	range: 0 - 127
Mask dAcc,Kn~n	defeat aux D accumulator unless knob is near n	range: 0 - 127
Mask dAcc,Kn<n	defeat aux D accumulator unless knob is < n	range: 0 - 127

Knob Grab Events

These events take the current value from either knob value, and use it to over-write the pattern values.

The knobs for these events are not automatically assigned – the aux value is used to select from front panel knobs A and B, or track control values knob1 or knob2, so a different knob source can be assigned on different steps.

knob n to velo	replace length value with current value of knob n
knob n to leng	replace velocity value with current value of knob n
knob n to auxC	replace aux C value with current value of knob n
knob n to auxD	replace aux D value with current value of knob n

Set Step Value Events

The “**Set Step Value**” event group mainly has events to set a pattern value on the step to a new value.

This may not seem like much use but it is very useful in combination with one of the mask events.

For example, you might use it to randomly switch a step between two specified notes, or by using the aux D accumulator and a mask event, you can perform pattern ‘morphs’, where the notes of the pattern gradually switch from the normal note values to those set in a ‘set note’ aux.

The events of this type are:

set note	
set velocity	
set length	
set delay	
xpose by n	adds a set value to the note value
add n to xC	adds a set value to the aux C value
add n to xD	adds a set value to the aux D value

This group also contains:

set midi chan

...this allows you to re-assign a single step to any MIDI channel of your choice. Great fun when used with another event to randomize the MIDI channel value!

Repeat Events

The “**Repeat**” event group used to create repeats of notes within the same step.

They are useful where you may want the occasional double speed note without having to use a faster tbase setting, or for those ‘Berlin School’ grace note effects.

The first repeat note events all divide the step into equal parts for the main note and the repeat notes.

So, a **rep*2** will divide the step in two equal parts, and each note will play for a proportion of their half of the step based on the **length** value for that step. **rep*3** will divide the step in three equal parts, and **rep*4** into four.

rep*2, note + n	repeat the note twice, with a note offset added to the repeat
rep*2, velo + n	repeat the note twice, with a velocity offset added to the repeat
rep*3, note + n	repeat the note 3 times, with a note offset added to each repeat
rep*3, velo + n	repeat the note 3 times, with a velocity offset added to each repeat
rep*4, note + n	repeat the note 4 times, with a note offset added to each repeat
rep*4, velo + n	repeat the note 4 times, with a velocity offset added to each repeat

The value range for all these events is +/- 127.

The note and velocity offsets are added to the note or velocity value on each repeat.

So if the primary note is a C, and you set **rep*4, note + n** on a step with a value of 1, the four notes played would be C, C#, D, D#.

These four notes would each be a quarter of the length of the normal note length for the pattern.

If you want to have both a velocity and note offset on your repeat notes on the same step, you can combine these events on more than one aux.

For each of the repeat events, there is an equivalent **prepeat** event.

These behave in almost the same way, except that when a prepeat event is active on a step, a negative delay value is applied to the step such that the repeating notes begin **before** the normal step time.

The next repeat event is a bit different - it doesn't equally divide the step among the repeat notes.

It simply repeats the note after the **length** setting for the step has elapsed.

If the length is set too long, you may get no repeats. If it is set very short, you can get very rapid streams of notes.

Since this event takes a value for the number of repeats, another couple of events are provided to configure note and velocity offsets if required.

rep*n by length	sets note to repeat n times, with length of note controlling repeat time
set rep*n note+	set repeat note offset for rep*n
set rep*n velo+	set repeat velocity offset for rep*n

One event new to Cirklon allows you to retain the length value as normal, by specifying the repeat time in terms of the standard P3 tbase settings – **rep*n tbase**

This event must be used in conjunction with **rep*n by length**.

Finally, the **step rep*n** event will cause the whole step to be played repeatedly before playback advances to the next step. Note this will increase the effective length of the bar. Each step can be repeated up to 256 times.

Redirect Aux Events

The “**Redirect Aux**” event group allow you to temporarily re-assign aux B, C or D to a different MIDI controller or aux event.

There is no such event for aux A as the auxes are processed in order - you must re-direct one aux from an earlier one. Since A is the first aux, it could not be re-directed.

aux B = cc #n	temporarily re-assign aux B to a MIDI controller	range: 0 - 127
aux B = event n	temporarily re-assign aux B to an aux event	range: aux events
aux C = cc #n	temporarily re-assign aux C to a MIDI controller	range: 0 - 127
aux C = event n	temporarily re-assign aux C to an aux event	range: aux events
aux D = cc #n	temporarily re-assign aux D to a MIDI controller	range: 0 - 127
aux D = event n	temporarily re-assign aux D to an aux event	range: aux events

auxes to trk n...

This event re-directs all following auxes so that they apply to the specified **target track** instead of the **current track**.

This allows you to cascade more than four aux events to one track, or send more than the maximum four MIDI CCs.

Pattern Control

This event group allows you to modify pattern settings.

set lstep abs	sets the last pattern step	range: 1 - 16
set lstep rel	adds a given value to the last step	range: +/- 15
set norm tbase	set tbase to one of the standard settings (1, 2, 4, 8, 16, 32, 64)	
set trplt tbase	set tbase to one of the triplet settings (2T, 4T, 8T, 16T, 32T, 64T)	
set direction	set direction (always reset to stored value at the end of the pattern)	

Global Control

This group allows you to modify global settings.

mute trk n	deactivate track n	range: 1 - 32
un-mute trk n	activate track n	range: 1 - 32
set tempo	sets the global tempo, temporarily over-riding the tempo knob	
set scene xpos	sets the scene transpose value, over-riding the current setting.	

14. Loading and Saving Data

All the song data and instrument definitions you create, as well as your configuration settings, are held in Cirklon's internal RAM memory.

Although internal RAM is battery-backed and non-volatile, the battery will require to be replaced after some years use, so it's a good idea to take regular backups of your data.

As you create more songs, and start to fill up the internal space, you may want to archive older songs to free-up space in RAM for new projects, while still being able to retrieve the old song at a later time.

Or you may want to save some or all of your instrument definitions to share.

Cirklon has a number of options to transfer data between internal RAM memory, and its internal flash storage (a micro-SD card on the CPU board) or an SD-card in the rear panel slot. Cirklon 2 can additionally make use of USB storage devices in the host port.

To make a FULL BACKUP of all your data for disaster recovery, the "Save raw SYX" option described below is the recommended approach.

It is also possible to backup and restore a complete image of the internal RAM via MIDI system exclusive transfer, but this is very slow for large amounts of data.

To save individual SONGS, instrument definitions or your configuration settings, there are special file formats for each. These files are based on the JSON data interchange format - a plain-text, human readable format. Details of the schema for these files will be released in a separate document.

The various data management options can be found on the play mode menu, under the "Card/Sysex" sub-menu. From the SONG, SCENE or TRACK pages, press MENU, then encoder 10. This display should appear:



Wipe Songs

The “wipe songs” option is not a load/save function as such. It allows you to quickly remove all SONG data from memory, while leaving all instrument definitions and configuration settings in place.

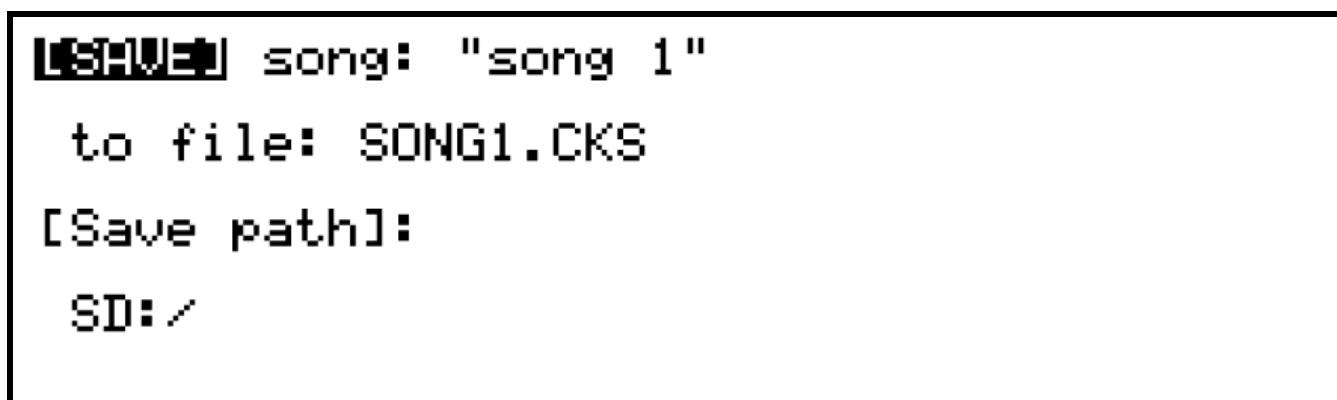
Obviously you should be careful to ensure that you have saved any important song data before using this option. A confirmation screen will appear, and you must simultaneously press encoders 6 and 8 together to proceed with the removal of all song data.

Card SAVE

Pressing “card SAVE” brings up a list of save options:



Upon selecting the first option, “SAVE Current Song”, the display should be something like this:



This page shows the current song name along with the filename and destination folder it will be saved to.

The filename is based on the song name, with any unsupported characters removed.

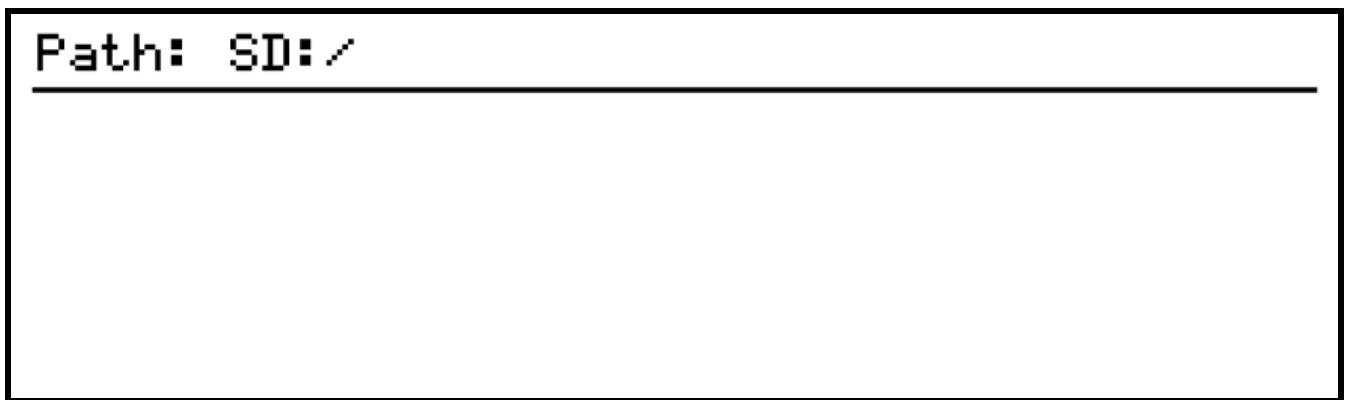
With [SAVE] highlighted, you can save the song to the filename and folder shown by pressing ENTER.

If a file with that name already exists in the current folder, you will be prompted to replace the existing file, or cancel the save operation. If you choose to cancel, you will be returned to the save song page, where you should alter the filename to make it unique, or press MENU once more to abort the save operation.

To alter the filename or destination, scroll and select either value on the song save page. For the filename, the text input mode display will appear:
(the file extension is hard-coded to .CKS)



If you select the [Save path]: option, the folder browser page will appear:



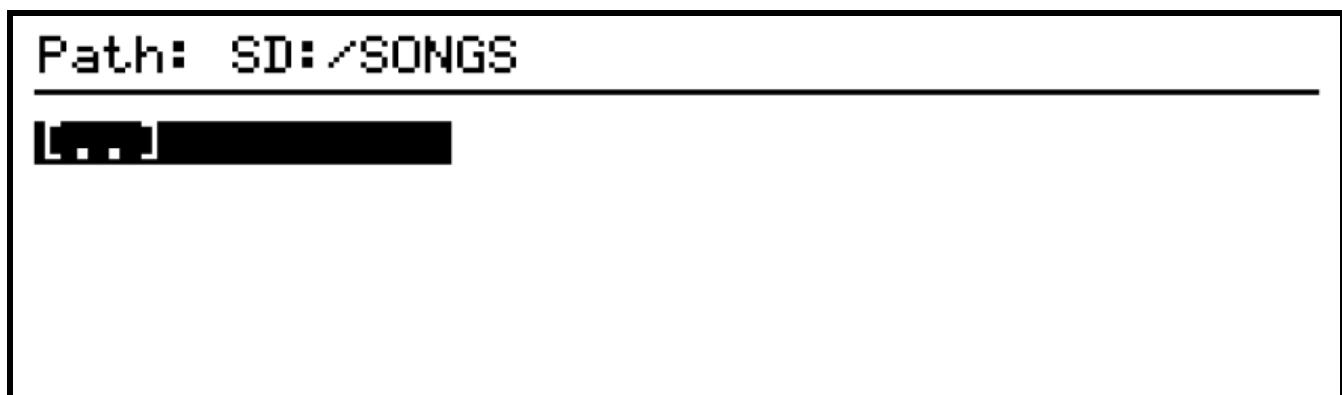
The top line shows the current path. The SD: label shows you are looking at a card in the rear panel slot. If the internal flash storage is selected, the label IN: will appear.
The single backslash shows you are looking at the root folder.

In this case, there are no other folders on the SD card.

To create a new folder, press the INSERT key, and enter the name for the new folder in the usual way:

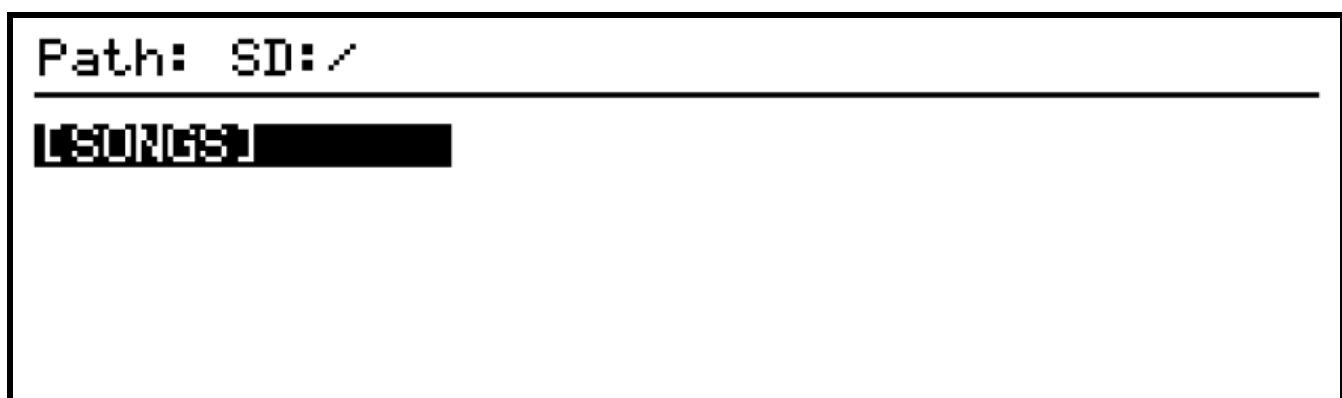


Once the new folder name is entered, you will return to the folder browser page, with the new folder selected as the current path:



With a folder structure on the card, you can navigate by scrolling and selecting the folders names here. The [...] label represents the parent of the current folder.

Selecting it here will return to the root folder of the card, showing the newly created SONGS folder as the only other choice:



With the desired path selected, press MENU to return to the save song page.

SAVE All Songs

The “SAVE All Songs” option writes a **single CKS file**, containing **all the songs currently in memory**. Note that it does not save each song to a file of its own. However, the song data is stored in the file in the same format as a single song file, and it is possible to extract an individual song from an “all songs” file using a text editor.

The multiple-songs file is intended as a way to back up a set of songs which are likely to be re-loaded together, such as a live set.

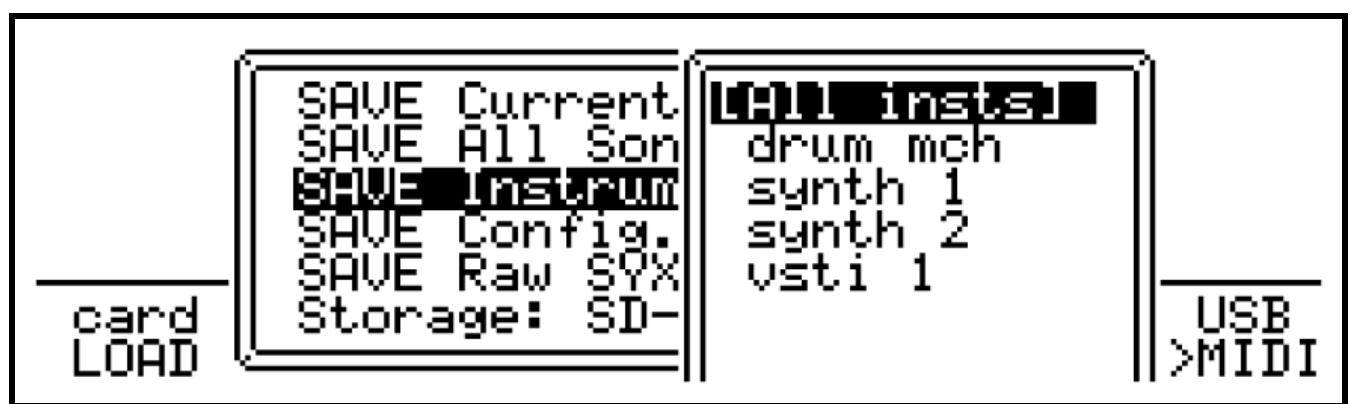
The songs will be written to the file in the order that they are currently stored in memory, and this order will be restored when file is re-loaded after performing a wipe-songs operation.

SAVE Instrument...

This option is used to save either a single instrument definition, or all definitions in memory, to a file with the extension CKI.

Instrument definitions include the MIDI settings for the instrument as found on the instrument definition page, as well as the track values configuration, including any CC labels, and the drum grid configuration, including the note row setup, note labels and fixed-row settings.

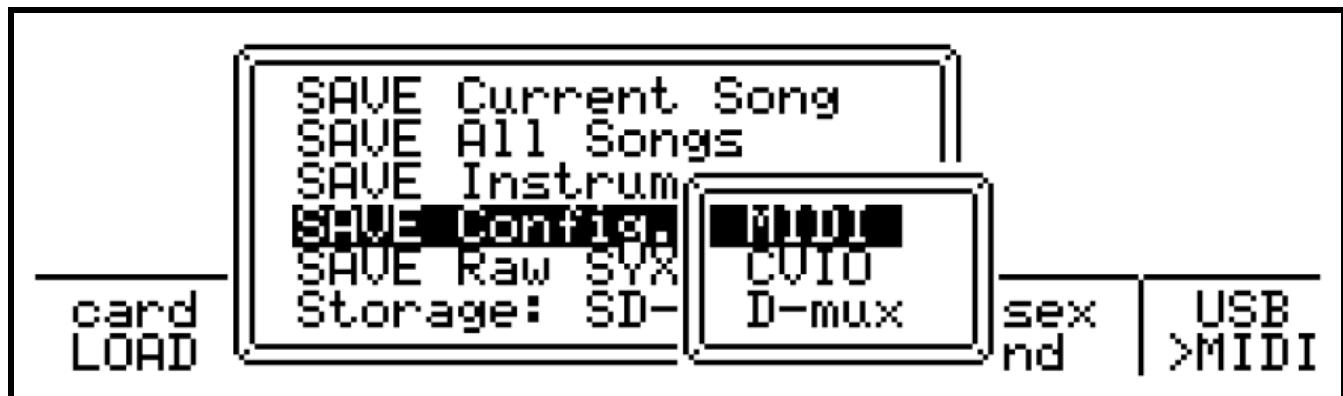
On selecting this option, you then select the instrument to save, with the option to save all of them at the top:



After selecting the instrument to save, a filename and path page similar to the song save page allows you to alter the filename and save path if required.

SAVE Config..

This option allows you to save your MIDI port, CVIO or Drum-mux configurations to a CKC file. The three sub-options appear as below:



A page where you can alter the filename and save path follows selection of the configuration to be saved.

SAVE Raw SYX Dump (Full Backup)

The “Save Raw SYX Dump” option will write a single file in binary sysex format, containing a complete dump of all data in memory, including songs, instrument definitions, and configuration options.

This method is best for making a full backup that can be used to restore your Cirklon to the exact state it was in at the time the file was saved.

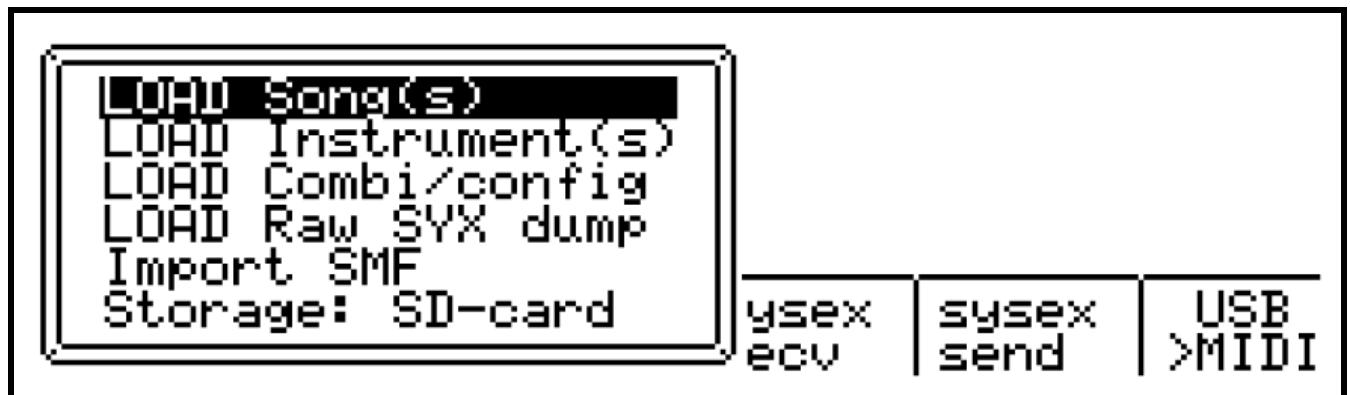
Storage

The final option on the save menu, “Storage:” allows you to switch between the internal flash storage and the SD-Card slot. On Cirklon 2, the further option of “USB stick” will appear if a suitable device is present in the USB host socket.

The menu option shows the currently active storage device.
Press ENTER on this option to cycle between the available devices.

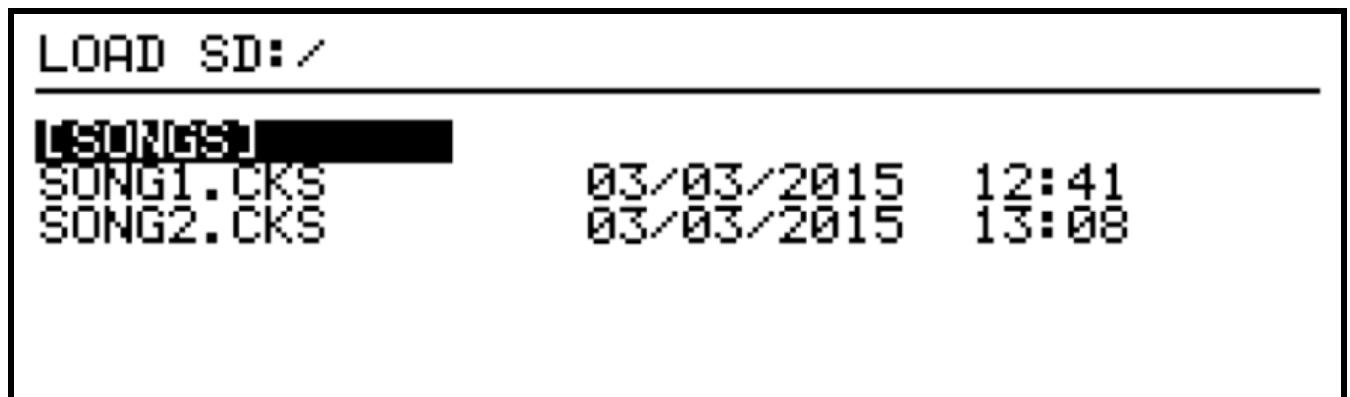
Card LOAD

“Card LOAD” has the following options:



The “LOAD Song(s)” option can be used to load a song saved individually, or multiple songs which have been saved in a single file using the “SAVE All Songs” option.

When you select this option, the file browser page appears:



The top line shows the current folder path.

Below is a list of sub-folders and .CKS (song) files in the current folder.

Folder names are shown in square brackets.

You can navigate to another folder by selecting it, or select the [...] label for the parent folder.

If a song with the same name as any of the songs to be loaded already exists in memory, you will see a display like this:



To replace the song in memory with the file version, press ENTER

To load the file version of the song with a new name, leaving the existing song in place, press the SAVE key.

To skip loading this song, press MENU.

If you are loading a file containing multiple songs, after pressing SKIP, the load operation will scan through the file for any further songs, and continue loading.

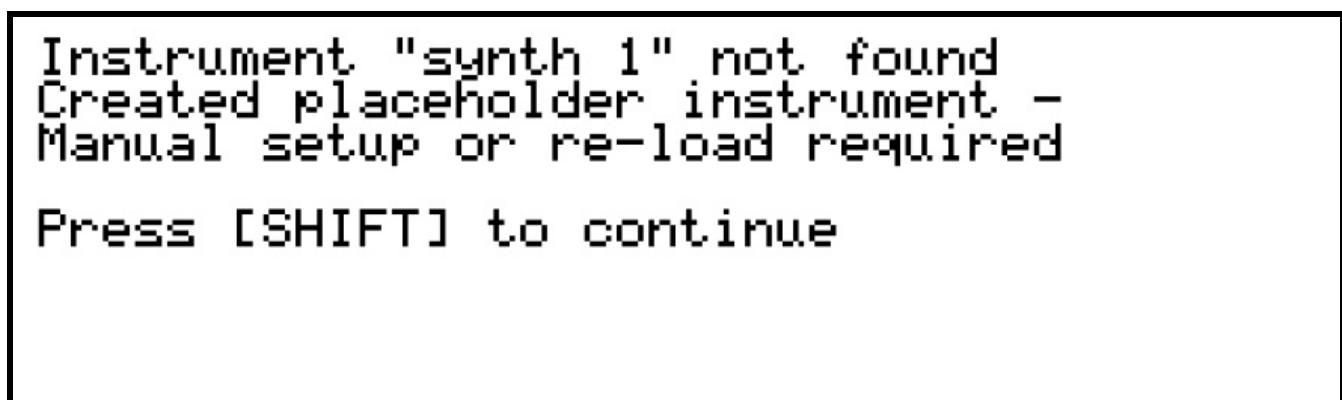
The replace/rename/skip prompt may appear again if any of those are also found in memory.

When re-loading songs, the instrument definitions are NOT included in the song files.

Any instrument assignments in the song data are referred to by name.

As the song is loaded, the instruments used are looked up in memory to check that they exist.

If any instruments used have been deleted or renamed, you may see the following display:



In order to continue loading the song, a dummy instrument is created with default settings – a ‘placeholder’.

Once the load operation is complete, you will need to resolve any missing instruments.

If the instrument has simply been renamed, you can delete the placeholder instrument from the instrument list, as described in section 2.

When you delete any used instrument definition, you will be prompted to select another instrument with which to replace the assignments of the placeholder instrument in the newly loaded song – at this point you can select the renamed version of the ‘missing’ instrument.

If the missing instrument definition was deleted from memory, but you have a backup of it in an instrument file, you can re-load the definition from that file, and the placeholder instrument will be replaced with the correct definition.

If you don’t have a copy of the instrument definition, you will need to manually re-create the settings for that instrument.

The “LOAD Instrument(s)” option is used to restore saved instrument definitions.

As with song loading, you will be prompted to replace, load under a new name, or skip any instruments which are already found to be loaded in memory.

The “LOAD Combi/config” option is primarily intended to be used to reload the configuration files for MIDI, CVIO or Dmux settings saved using the “SAVE Config” option.

Note that the file selection list for this option will show ALL files with an extension beginning with .CK

You may select song or instrument data files here, and they will load successfully.

Finally, the “LOAD Raw SYX” option will re-load a full memory dump from a SYX file.

*** Note that the entire contents of memory will be cleared before a SYX file is loaded**, even if loading of the file is not successful.

A confirmation page will appear to make sure this doesn’t happen by accident:

ALL existing data will be lost.
Are you SURE?

[Continue]

Abort

To confirm the re-loading of SYX data, press encoders 6 & 7 together. The chosen file will be read into memory, all the songs and configuration data reloaded, and you will be returned to the first SONG in memory.

Import SMF

This option allows you to import the contents of a standard MIDI file, with the .MID extension.

When selecting this option, you must then choose whether to merge the data into the existing song, or create a new song:



Each track in the MIDI file will be converted to a CK type pattern.

Note that instruments will not be assigned to the tracks in a new song – you will need to manually assign appropriate instruments for the patterns being imported.

MIDI Sysex

Due to the amount of data that can be stored in Cirklon's memory, sysex backup can be a time-consuming process. The use of flash storage is strongly recommended.

If you must use sysex, dumps can be sent and received either via MIDI port 1, or USB MIDI port 1.

Switch between serial or USB MIDI by pressing encoder 11 at the main Card/Sysex menu. The arrow indicates the current selection.

With suitable connections made to the device that is to receive a sysex dump, and the device waiting for the dump to arrive, press encoder 10, "sysex send" to start the dump.

The process for receiving and saving the dump will depend greatly on the target device, so is not described here.

To re-load a previous dump, press encoder 9, “sysex recv”.
Then start transmission of the dump at the source device.
All being well, the previously saved data should re-load.

Factory Reset / Memory Initialisation

If you need to completely re-initialise Cirklon’s memory, you can do this as follows.
With the power off, hold down the SONG key.
Turn the power on, continuing to hold down the SONG key until you see the confirmation
page with the text “Wiping ALL data”

At this page, press encoders 6 and 8 together to confirm the full reset.

All song and instrument data will be removed, and all configuration settings returned to
default values.

15. Firmware update

Cirklon supports the loading of operating system updates via MIDI or MMC/ SD-card. Due to the much greater speed of SD-card, it is strongly recommended to use this method.

The update process is carried out in a special firmware update mode, which is entirely separate from the main sequencer program.

It is not possible for a failed OS update to leave Cirklon in an unusable state. You will always be able to access update mode to try again if there is a problem with an OS update.

Before updating the firmware, **you should always backup your current data**, as described in the previous section. The single SYX file backup is the best method to use.

In some cases, the data in memory will be unaffected by the update process, but this is not always true.

In very rare instances, it may be required to use the individual file formats for backup and restore of data between major version changes - this would be clearly indicated on the download file page.

Firmware update mode is accessed by holding down step key 1 as you power Cirklon on. This will bring up this display:



Firmware update images can be found on the download page at the Sequentix website:
<https://www.sequentix.com/cirklon-downloads>

Beta versions which may include new features still under test can be found via the forum, in the OS updates section:

<http://forum.sequentix.com>

Updates will be in the form of a .SYX file.

Different files are used for each model, Cirklon and Cirklon 2

For the **original Cirklon**, the file name will be **CIRKOS.SYX**

For **Cirklon 2**, the file name will be **CK2OS.SYX**

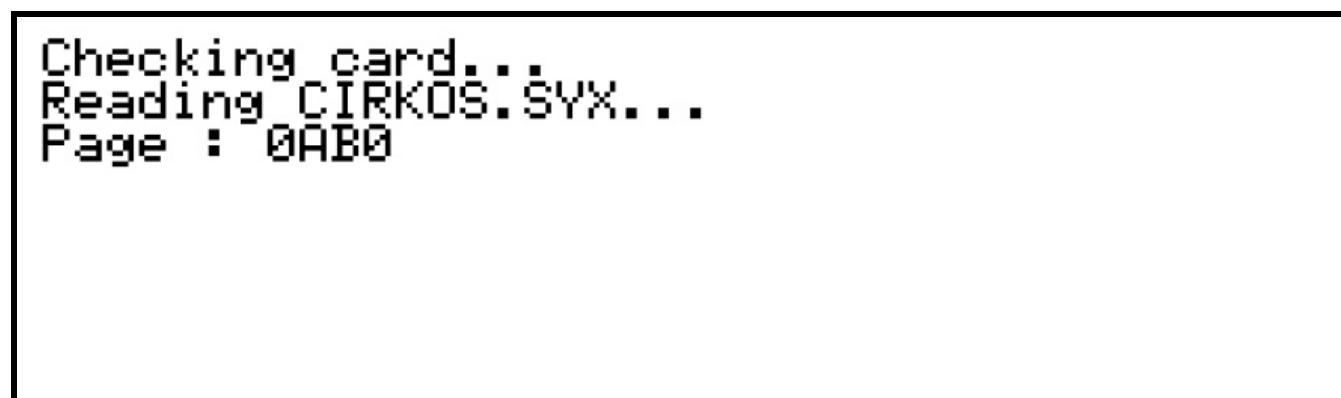
These names are hard-coded into firmware update mode to maximise space available for the main code.

Updates downloaded from the web will be contained in a ZIP file, named according to the build or version number.

You should extract the SYX file from that ZIP, and copy it to your SD card.

With the card containing CIRKOS.SYX for Cirklon, or CK2OS.SYX for Cirklon 2 in the root folder, inserted in the rear panel card slot, press encoder 8.

The display should show the progress as the file is loaded into the program flash memory inside your Cirklon CPU:



Once the update is complete, Cirklon will boot straight into the new version of the OS.

A check of all data in memory is performed on every boot. If there are no problems indicated, it may be OK to start using your existing data, but the recommended approach is to re-load the backed-up data.