

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From 2020, the original Cirklon model has been replaced by Cirklon 2.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

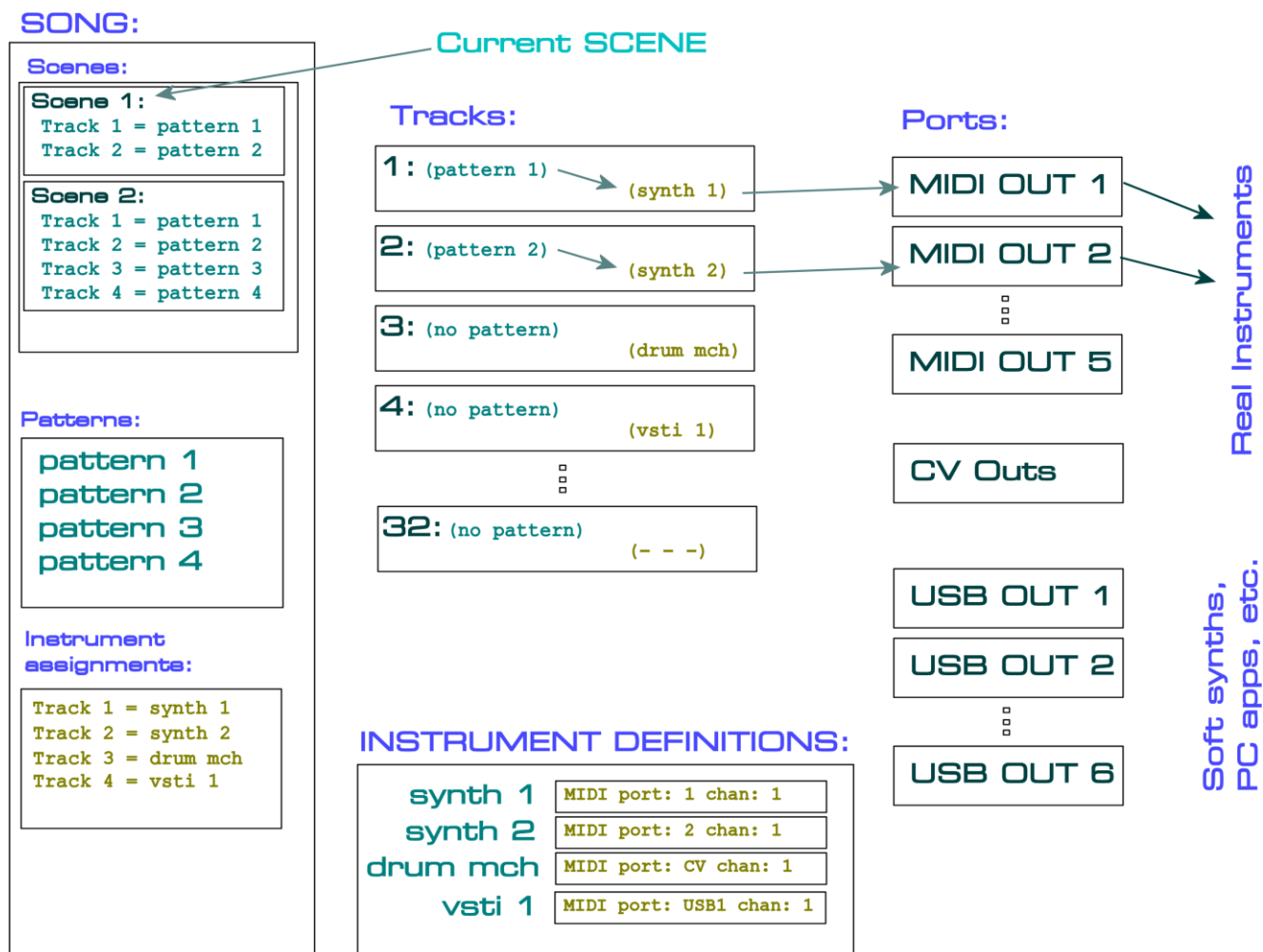
The instrument assignments determine which instrument will play the pattern on each track.

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...