

**SEQUENTIX MUSIC SYSTEMS Ltd.**

**P3 SEQUENCER**

**OPERATION MANUAL**

**V3.1.006**

**© 2006**



Introduction.....	6
Getting Started.....	8
Making Connections.....	8
Instant Gratification.....	10
Play Mode.....	10
First Pattern Edit.....	11
Modes of Operation .....	15
Play mode .....	16
Tracks.....	16
Parts.....	17
Part Select .....	17
Parts Are Not Patterns!.....	18
Part Chains .....	19
Changing Mode .....	21
Pattern Edit.....	21
Playlist Edit.....	21
Play Mode Pages and Soft-keys.....	22
Selecting Pages .....	22
Page 1 - Bank / FTS / Tune.....	23
Page 2 - Part Settings .....	30
Page 3 - Part Functions.....	33
Page 5 - MIDI Config.....	35
Page 6 - Bank functions.....	40
Page 7 : Sysex & User Config.....	42
Other Play Mode Features .....	45
Play Mode Knob Use.....	45
Fine Tempo Adjust.....	45
Status LED Displays.....	46
GBar Pos and Playlist Hold .....	46
Pattern Edit.....	47
What's In A Pattern? .....	47
Editing Step Values.....	48
UPPER MODE.....	48
Viewing numeric row values .....	51
STEP MODE.....	52
Assigning and Using Auxiliaries .....	56
The Aux Config Page.....	57
Editing Aux Values.....	61
Summary of Aux Usage .....	63
Pattern Edit Pages and Soft-keys.....	64
Selecting Pages .....	64
Page 1: Timebase and Sync.....	66
Page 2: Direction and Last step .....	68
Page 3: Clear, Copy & Paste .....	70
Page 4: Rotate and Shuffle.....	73
Page 5: The Note Randomizer.....	75

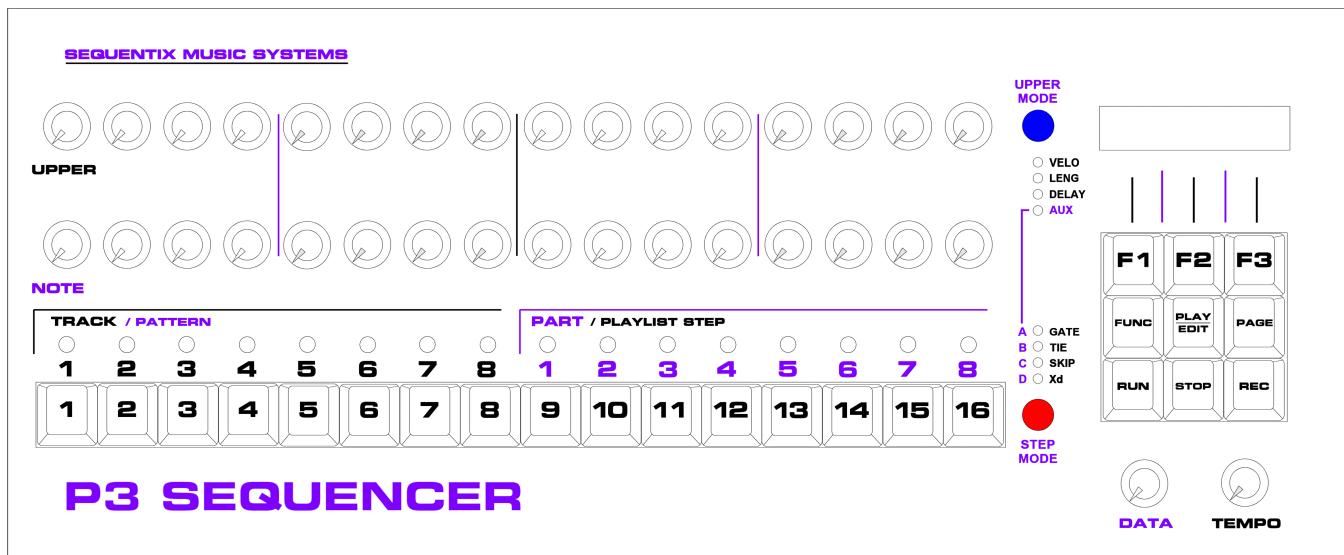
---

Page 6: The Upper Randomizer .....	77
Page 7: Sculpt.....	79
Page 8: Config.....	81
Other Pattern Edit Functions.....	83
Quick Config.....	83
Quick Mute Access.....	83
Leaving Pattern Edit .....	84
Save To.....	84
Playlist Edit .....	86
Why Playlists? .....	86
What's Really In A Part.....	87
What's In A Playlist .....	88
The Default Playlists .....	89
Editing A Playlist .....	90
Changing To Playlist Edit.....	91
Playlist Edit Controls.....	92
Playlist Edit Display .....	92
Playlist Edit LEDs.....	93
Selecting A Different Pattern.....	94
A Longer Playlist.....	95
Leaving Playlist Edit.....	97
Some More Editing.....	98
Playlist Views.....	98
Back To The Playlist.....	102
Step Repeats.....	102
Step Transpose.....	103
Step Sync.....	105
Other Features Of Playlist Edit .....	109
Short-cut Playlist Creation .....	109
GBar Pos and Playlist Hold .....	110
Quick Mute Access.....	110
Real-time Record.....	111
Differences Between Play Mode, Playlist Edit and Pattern Edit .....	111
Pattern Edit.....	111
Play Mode & Playlist Edit .....	111
Record Page 1 -.....	112
Record Page 2 -.....	114
Record Page 3 -.....	115
Use With Caution!.....	116
Leaving Record Mode.....	116
Arpeggio Capture.....	117
Leaving Arpeggio Capture .....	119
Remote Control.....	120
User Interface Controls .....	120
Remote Part Transpose (PXPos) .....	121
Using Auxiliary Events.....	122
Randomizing Pattern Values .....	122

---

How Events Work.....	124
Inter Track Events.....	127
A Quick Grab.....	128
Auxes With Two Values.....	128
Target Values For Grab, Swap and Push .....	131
Transpose By Events.....	132
Aux Notes .....	134
Chord Memory.....	135
Polyphonic Real-time Record.....	135
Accumulator Events.....	136
What Is An Accumulator?.....	136
Putting An Accumulator To Work.....	137
Relative Accumulation.....	139
Accumulator Configuration.....	140
The Other Two.....	144
Auxiliary Value Scaling.....	145
The Accumulator Events.....	146
Accumulator Mask Events.....	147
Knob Mask Events .....	147
Knob Grab Events.....	148
Set Step Value Events.....	149
Repeat Events.....	149
Redirect Aux Events.....	150
Pattern Control.....	151
Global Control.....	151
Appendix A - Preset FTS Scale Definitions.....	153
Appendix B – User Conf Options.....	155
Appendix C - Memory Organisation.....	157
Note on Playlist Edit for 3 Bank Configuration.....	158
Appendix D - Miscellaneous Useful Info .....	159
P3Tools.....	159
Memory Initialisation .....	159
SysEx Firmware Upgrade.....	159
Index .....	161

## Introduction



It's fair to say that the word 'sequencer' has undergone a significant change in meaning since the first devices to be called sequencers appeared, not long after the advent of the first synthesizers.

Most people nowadays think of a sequencer as a piece of computer software which provides multi-track recording and playback of MIDI data, in a manner vaguely similar to a multi-track tape recorder. Many sequencer applications now include audio facilities, and even virtual effects processors and software synthesizers.

However the first sequencers, known retrospectively as analogue sequencers, were altogether different.

An analogue sequencer was a desktop unit or synthesizer module, typically dominated by a bank of knobs or sliders, each of which could be used to adjust the tuning of one step in a sequence.

The sequencer circuitry would pass a voltage output from each knob or slider to a common CV output, stepping through each one in turn under the control of a master clock.

With the CV output driving a synthesizer's oscillators, and the clock output driving its gate, a short, repetitive pattern of notes could be created.

The maximum number of steps in a sequence was the number of knobs on the sequencer.

There was no storage of patterns, other than to leave the knobs in the same positions, and hope the tuning didn't drift.

But ignoring these limitations, the analogue sequencer represented an entirely new kind of instrument - one where the performer could dynamically interact with a pattern of notes without the burden of actually playing them in real time.

The earliest devices, such as Raymond Scott's Circle Machine, were as likely to be used to create abstract sonic effects as musical patterns. They brought forth a new range of rhythmic and melodic ideas, and lead electronic musicians down interesting new paths.

The P3 Sequencer is a modern take on the analogue sequencer, but it takes the concept much further.

P3 is a pattern-based, 8 track hardware MIDI sequencer. Its user interface and primary method of pattern entry are based directly on the classic analogue sequencers.

But it adds full storage and recall of patterns, can simultaneously drive multiple MIDI synths or samplers each playing different patterns, allows chaining of patterns to make longer musical phrases than the basic 16 steps, and also features sophisticated pattern modulation options.

In addition, it allows real-time capture of note input from a MIDI keyboard, so you can use P3 like a more typical hardware MIDI sequencer if you want.

P3 allows real-time control of track mutes and simple pattern and pattern chain selection across multiple tracks, plus analogue style editing of patterns with separate note, velocity, tie, gate, length, gate delay, and a combination of up to four MIDI controllers, extra notes, or programmatic functions, per step.

The unique programmatic functions, called auxiliary events, allow the simple creation of patterns that can dynamically grab values from the patterns on other tracks, randomize step values, alter pattern and global settings, and more - including modulation accumulators that can be configured as LFOs, perform arpeggiation effects or enable complex self-modifying patterns.

A non-destructive Force-to-Scale function provides preset and user-defined scales to keep everything in key (but defeat-able per pattern step if required), and a special pattern sync option allows patterns with different lengths and/or timebases to free-run, or be synced to a global bar.

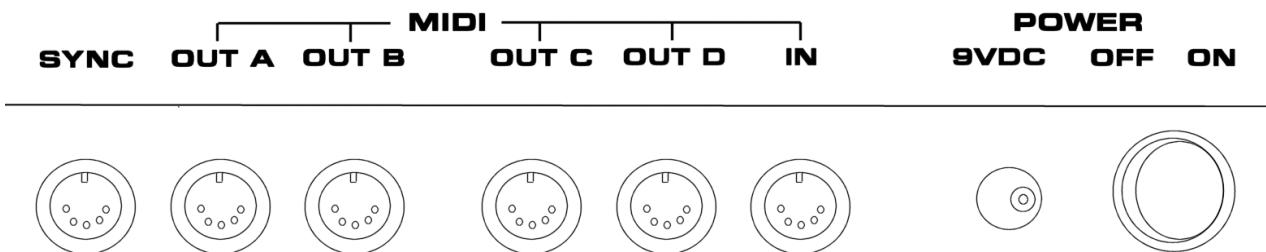
Building on ideas from various vintage and modern analogue sequencers and drum machines, P3 has been designed to provide a direct and intuitive method for creating pattern-based electronic music. That's not to say it will automatically do it all for you - there is a learning curve involved. To get the best from P3 you will need to familiarise yourself with the various modes of operation, pattern and playlist values, track and configuration settings. So, read on...

## Getting Started

### *Making Connections*

Before you can make any music with P3, it must be connected to a power supply and a MIDI synthesizer, sound module or MIDI to CV convertor and analogue synthesizer.

The rear panel connections look like this:



From left to right, the sockets are:

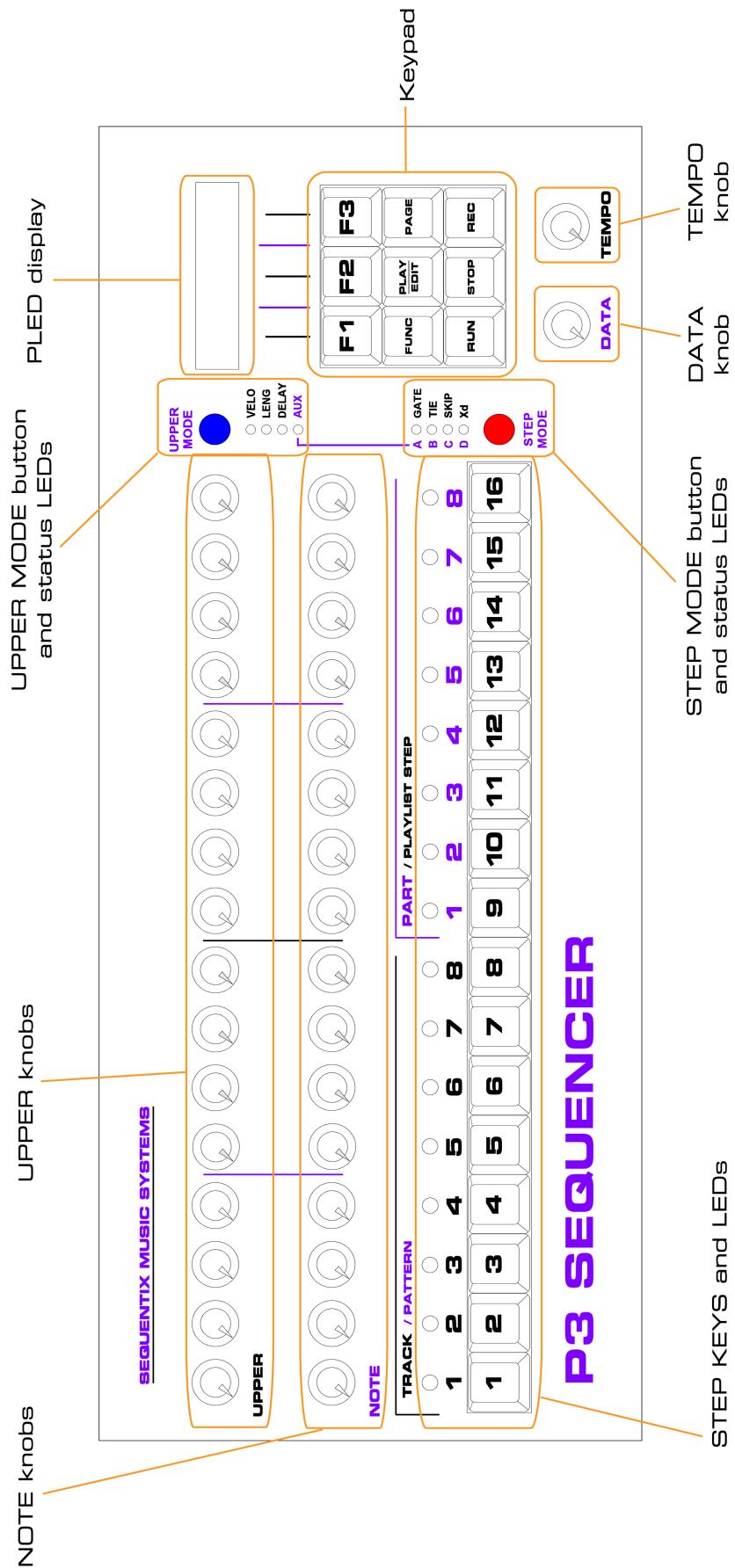
- **SYNC output**, which carries both a clock-only MIDI signal and DIN Sync 24
- four parallel **MIDI outputs** (all carry the same signal – it's a built-in thru-box)
- **MIDI input**
- **Power** supply input

The MIDI and SYNC connections are all standard 5-pin DIN sockets.

The power supply input is a 2.1mm barrel-type connector.

P3 needs a supply of 9 volts DC with centre positive polarity, or 9 volts AC. Current rating must be at least 400mA.

With a power supply and suitable MIDI instrument attached, you can power-on and start making music...



## P3 Sequencer front panel control diagram

## *Instant Gratification*

Before looking at the operation of P3 in depth, let's take a quick look at how to get a basic pattern running and making some noise.

To save time explaining the configuration options, connect a MIDI device to P3 which will respond on MIDI channel 1.

P3 ships with sample patterns in most memory locations, but bank 1 will be empty, and will be selected at first power-on.

Track 1 will be assigned to MIDI channel 1.

## Play Mode

P3 has three different modes of operation, each of which use the front panel controls in a different way. The default mode at power-on is **play mode**.

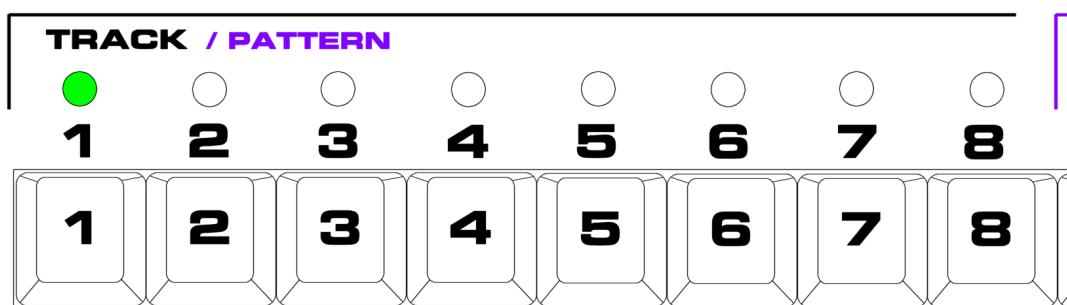
**Play mode** allows the muting and un-muting of **tracks**, selection of **parts**, and access to various control and configuration options.

These will all be described later.

## On the Right Track

All we are concerned about right now is that track 1 is **active**.

In **play mode**, the first eight step keys, as shown below, are used primarily for muting and un-muting tracks:



The LED above track 1 should be green, showing that the track is **active**.

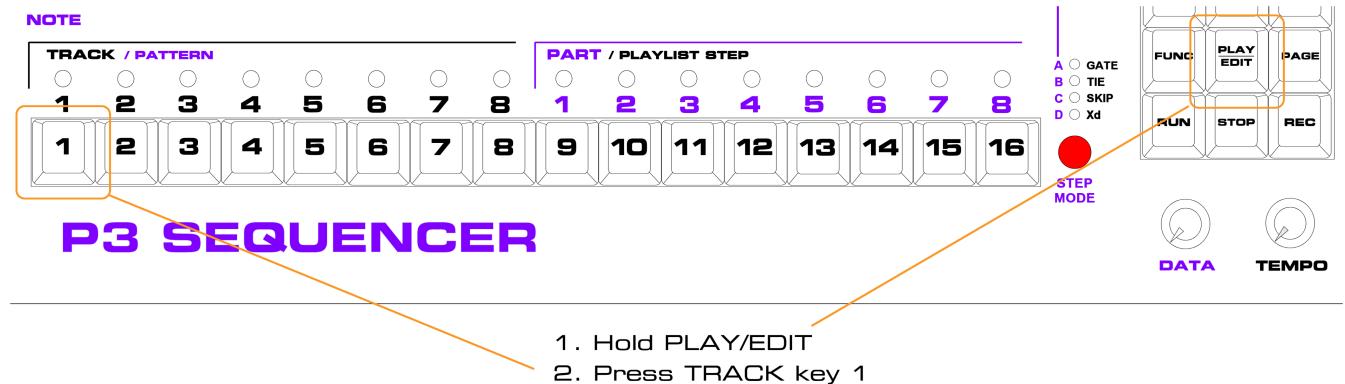
If the LED is off, the track is **muted** – in which case, press **TRACK** key 1 and the LED will turn on.

The status of the other tracks doesn't matter for now.

## First Pattern Edit

To create a pattern, you must switch to **pattern edit** for **track 1** as follows:

- hold the **PLAY/EDIT** key in the centre of the keypad
- press and release **TRACK key 1**



You will now be in **pattern edit**, editing the **pattern on track 1**.

To confirm this, there will be an inverse-video “E” character in the top right of the alphanumeric display. The display should look like this:

<b>16</b>	<b>E</b>
<b>- tbase + sync</b>	

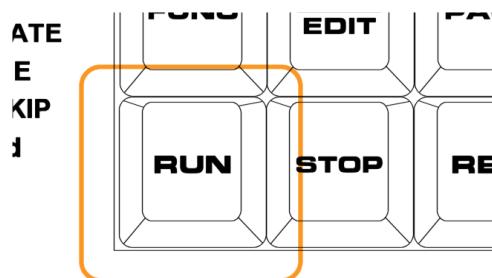
All of the step LEDs will be off.

## On The RUN

P3 is designed so that you never need to stop playback for anything other than bulk memory dump operations.

So let's start P3 running before you enter any pattern data.

Press the **RUN** key, at the bottom left of the keypad:



You should now see a blip of green running across the step LEDs, showing the current playback position in the pattern.

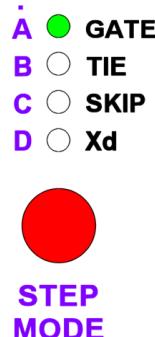
You can adjust the tempo using the **TEMPO** knob at the bottom right of the panel if required, and see the green blip change speed.

## Setting Gates

In **pattern edit**, the 16 step keys are used to toggle status values in a pattern on or off.

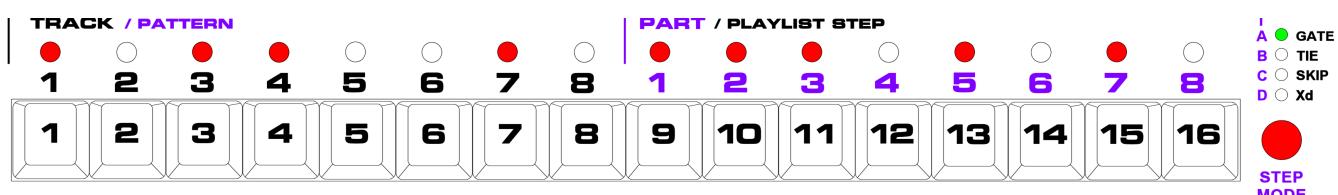
There are several status values for each step in a pattern, but the most important is **gate**.

When you first enter **pattern edit**, the step keys are assigned to **gate**, as shown by the status LED just above the **STEP MODE** button, labelled **GATE**:



**Gate** controls whether a note will be played on each step in the pattern or not.

With P3 running, start pressing some of the step keys, **1** to **16**, to turn steps on and off. An active **gate** status is shown by the LED above the step lighting up red, like this:



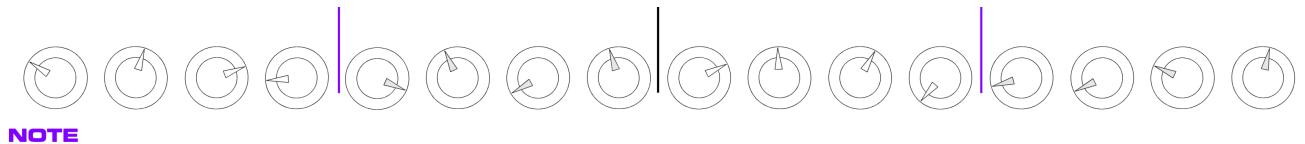
As the green blip passes each step with **gate** active, you should hear a note playing on the attached instrument.

They'll all be the same note so far, but it's a start.

## Setting Note Values

Now let's change some of the **note** values.

This is done using the lower row of 16 knobs, which are labelled **NOTE**:



Simply turn the note knobs above each step to edit the **note** on that step.

The **note** value will jump to the current knob position as soon as you move the knob, and will be shown on the top line of the display, like this:

<b>Step 1 Note C 5</b>
- tbase + sync

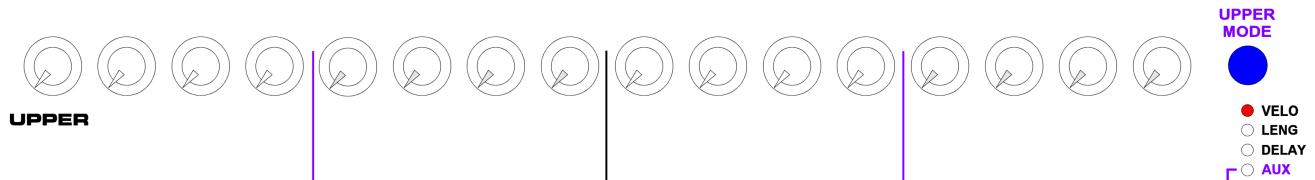
When you stop moving the knob, the display will return to normal after a second or so.

Now that you can choose which steps will play notes by setting gates, and select different notes with the knobs, let's look at the upper row of knobs.

## Setting Velocity Values

On entering **pattern edit**, the **UPPER** knobs are assigned to edit the MIDI **velocity** on each step.

As with the step keys, there are other values the upper knobs can be used to edit. These are selected using the **UPPER MODE** button, with the chosen value shown by one of the four status LEDs below. For now we'll just look at **velocity**.



As you turn the upper knobs, the top line of the display will again show the new values you are setting, like this:

<b>Step 1 Velo 100</b>
- tbase + sync

Go through your pattern, and set a different **velocity** value on each step. You should hear this having an effect on your attached instrument as the pattern plays.

## Saving The Pattern

By now you should have a basic pattern running, can edit note and velocity values, and control the rhythm using the gates.

Once you've had fun with that for a while, let's save the pattern, and get on with looking at the rest of P3.

To exit **pattern edit**, press the **PLAY/EDIT** key once more.

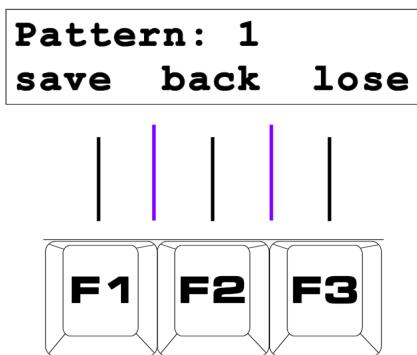
### Soft-keys

Since there are so many functions in P3, it would be impractical (read: very expensive) to provide a dedicated key for each, so a system of **soft-keys** is used.

The soft-keys are labelled **F1**, **F2** and **F3**, and are found on the top row of the keypad, just below the display.

When you exit **pattern edit**, a special page appears that allows you to decide whether to keep your edited pattern or not.

Three soft-key options appear on the bottom line of the display, like this:



This table shows the effect of these options:

Key	Label	Function
F1	save	Store the edited pattern, and return to play mode
F2	back	I didn't mean to press PLAY/EDIT, return to pattern edit
F3	lose	Lose any changes to the edited pattern, and return to play mode

You'll want to save your first masterpiece, so press **F1 – "save"**.

P3 will return to **play mode**, and continue playing your saved pattern.

## *Modes of Operation*

Now that you've had P3 play some notes, and hopefully satisfied the usual urge to get a new bit of equipment doing something as soon as possible, let's look in more depth at the full feature set.

P3 has three modes of operation.

These are:

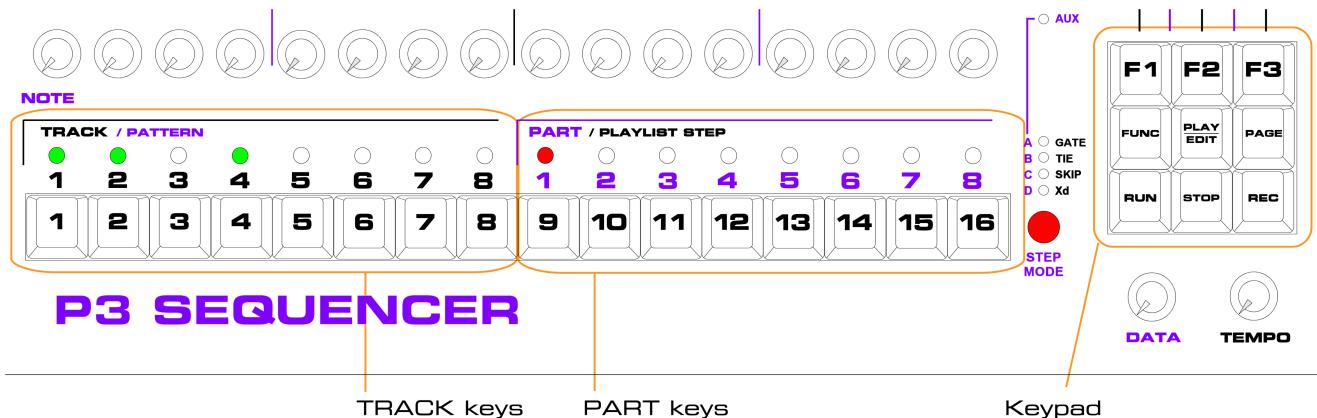
- **Play mode** for control of track mutes, part selection, etc.
- **Pattern edit** for creation and editing of patterns
- **Playlist edit** for selection of patterns, or pattern chains

Play mode is the main operation mode of P3, so let's look at it first.

## Play mode

In each P3 mode, the keys, knobs and LEDs on the P3 panel take on different functions and meanings.

This diagram shows their functions for **play mode**:



### Tracks

The first eight step keys are called the **track keys** in **play mode**.

They are mainly used for muting and un-muting each track.

There are eight tracks in P3, so there are eight track keys, labelled **TRACK 1** to **8** on the panel.

If the track is **active**, the LED above the track key will be **green**.

If the track is **muted**, the LED will be **off**.

When P3 is running, **only active tracks play notes**.

Each track is able to play one pattern at a time, from a set of patterns unique to that track.

Normally there are 8 patterns for each track, a total of 64 patterns per **bank**.

There are six banks, for a total of 384 patterns.

**\*TIP** While there are always 384 patterns in total, it is possible to re-configure the way the patterns are distributed among the banks.

P3 supports an optional 3 bank configuration, where each track gets 16 patterns per bank, and a 12 bank configuration, where each track gets only 4 patterns per bank.

Changing memory configuration is not recommended until you have used P3 for long enough to know that your requirements differ from the standard memory layout.

Refer to the appendix on **Memory Organisation** for details.

## Parts

The patterns that play on each track are controlled by **parts**.

Parts allow you to select different patterns simultaneously across all eight tracks.

Parts also recall the mute settings for the tracks, and some other settings such as Force-to-Scale and bar length.

These more advanced options will be explained later, when we look at the soft-keys used to control them.

## Part Select

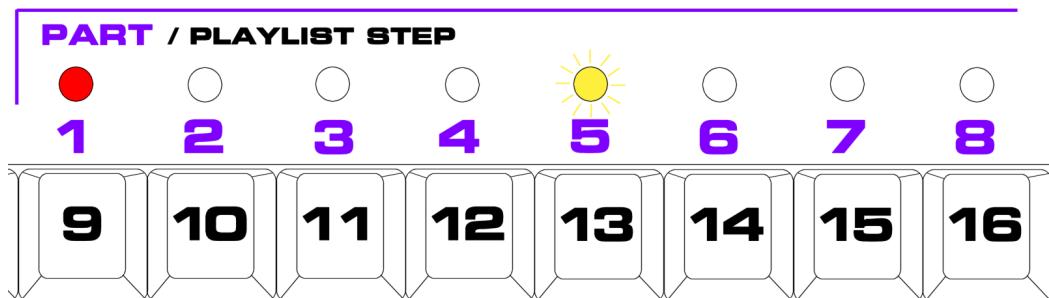
Parts are selected using the **part keys** – the rightmost eight step keys, which are labelled **PART 1** to **8** on the panel.

The currently selected part is indicated by a red LED above the part number.

To select a new part, press one of the other part keys.

If P3 is running when you select a part, the newly selected part LED will flash amber until the current bar completes.

For example, if you are on part 1, and select part 5, the LEDs will look like this:



The change of part is synchronised to the end of the bar, at which time the red LED for part 1 will turn off, and the LED for part 5 will go red.

If P3 is not running, the part change will happen immediately.

## Parts Are Not Patterns!

It is important to note that parts are not the same thing as patterns.

Parts hold a **pattern number**, or a sequence of pattern numbers to be played on each track. These pattern selections are called **playlists**.

In an **initialised bank**, such as bank 1 in a newly supplied P3, each part will play only a single pattern for each track – what we would refer to as a single-step playlist.

The pattern number selected will be the same as the part number.

So a table of the pattern selections across the tracks for each part would look like this:

		Track							
		1	2	3	4	5	6	7	8
Part	1	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2	2
	3	3	3	3	3	3	3	3	3
	4	4	4	4	4	4	4	4	4
	5	5	5	5	5	5	5	5	5
	6	6	6	6	6	6	6	6	6
	7	7	7	7	7	7	7	7	7
	8	8	8	8	8	8	8	8	8

Pattern numbers

This makes it seem very much like parts are just patterns – each part being made up of a pattern for each track.

While you familiarise yourself with P3, it may be helpful to use parts in this way.

In fact, it is possible to edit the **playlist** for each track so that any of the patterns for a track can be used in any part.

One pattern could be re-used for the same track in all eight parts if required.

You can also make a **playlist** up to eight steps in length, so that a series of different patterns can play on a track.

This allows you to create pattern chains longer than the individual pattern length, with different lengths of chain on each track.

The patterns can be repeated for up to 8 bars in each **playlist step** and can be transposed up or down, allowing a single pattern to be used for a progression of different keys or roots.

We'll look at these advanced options in the later section on **playlist edit**.

## Part Chains

Without getting into the complexities of playlist editing, there is a simpler way to chain the same number of patterns together across all tracks - a **part chain**.

A part chain is a multiple selection of parts, where P3 plays each part in the chain in turn, then loops back to the first part at the end.

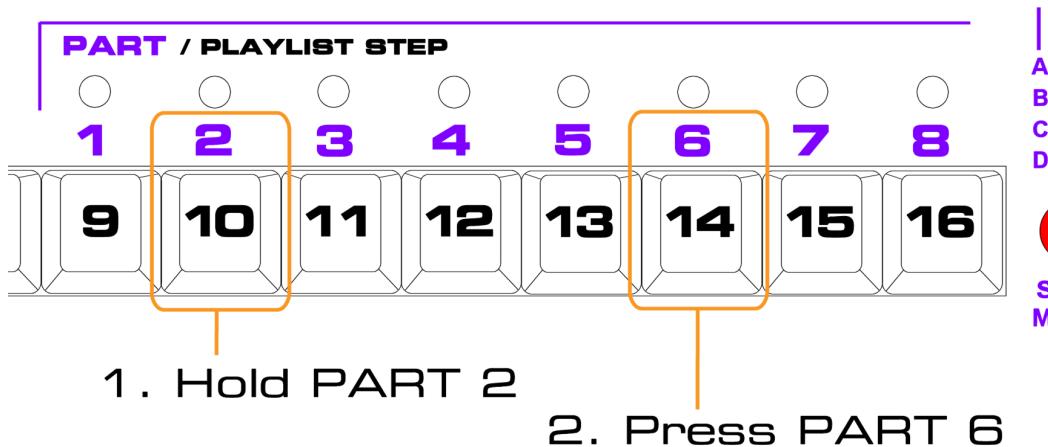
This is similar to having a multi-step playlist on each track, but is considerably easier to set up and understand, though not as flexible.

### Selecting A Part Chain

To select a part chain:

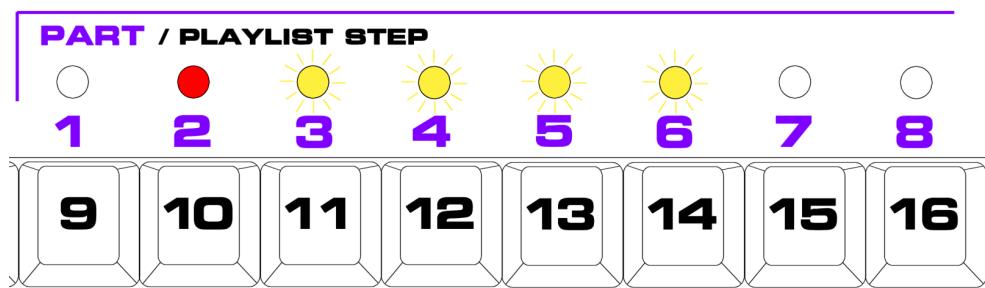
- hold the **PART** key for the first part in the chain
- press the **PART** key for the last part

For example, to create a chain of parts 2 to 6:



When a part chain has been selected, all the part LEDs for the parts in the chain will flash amber, except for the currently playing part, which will show red as normal.

For the example above, the LEDs should look like this:



To clear a pattern chain, just select another single part with one key-press.

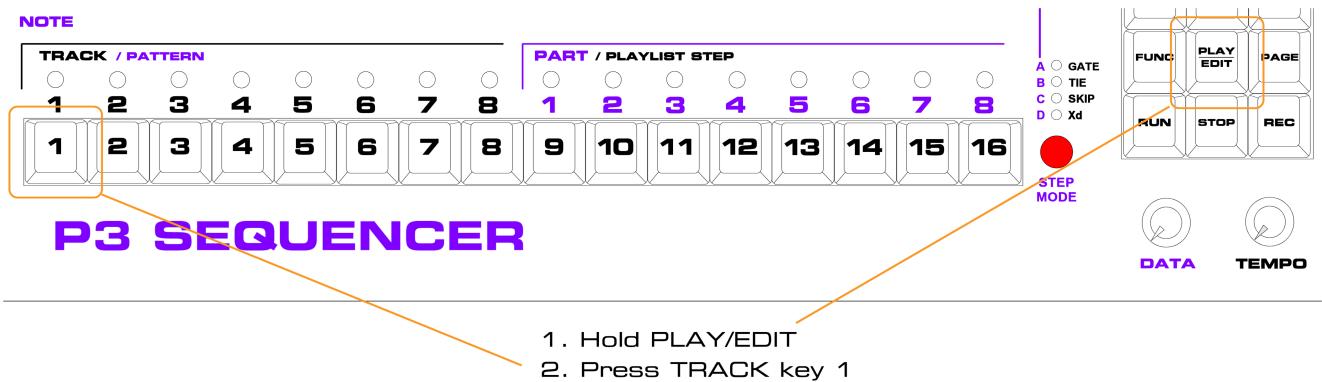
## Changing Mode

### Pattern Edit

You have already seen how to access **pattern edit** from **play mode** in the getting started section, but to re-cap:

- hold the **PLAY/EDIT** key in the keypad
- press the **TRACK** key for the **track** on which you want to edit the **currently playing pattern**

For example, to edit the **current pattern** on **track 1**:



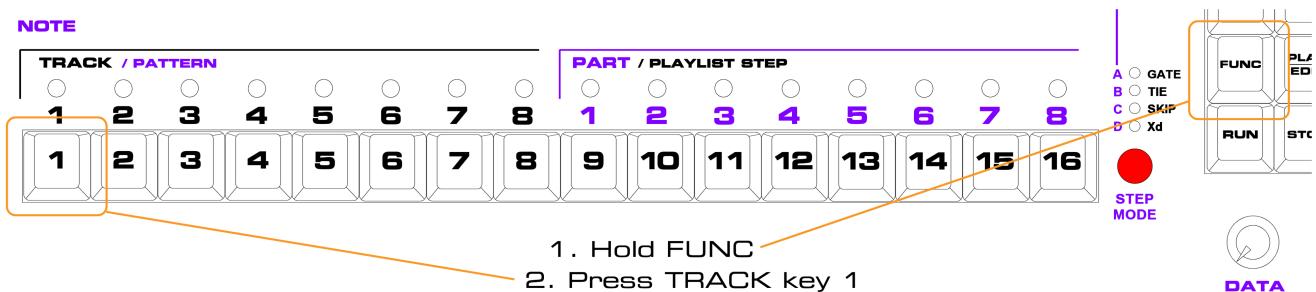
### Playlist Edit

**Playlist edit** is best left until you have a good understanding of **play mode** and **pattern edit**, but we'll need to access it sometime.

To enter **playlist edit**

- hold the **FUNC** key
- press the **TRACK** key for the **track** on which you want to edit the **playlist**

For example, to edit the **playlist** on **track 1**:



The use of **pattern edit** and **playlist edit** will be described in detail in later chapters.

## Play Mode Pages and Soft-keys

Play mode has a number of different display **pages**.

Each **page** shows some different values or text labels on the display and has its own set of **soft-key** assignments for **F1**, **F2** and **F3**.

Some of the pages show information about the current bank or part status, with soft-keys for editing these settings.

Others have soft-keys to access configuration options.

There are six pages in total.

Pages 1, 2 and 3 are for commonly used bank selection and part functions.

Pages 5, 6 and 7 are for less commonly used configuration options.

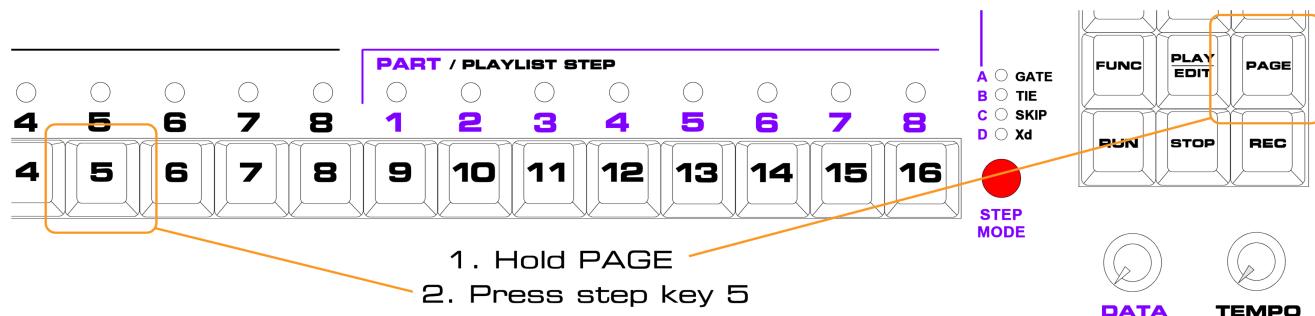
Page 4 is reserved for future expansion, that's why there's a gap.

## Selecting Pages

To select a play mode page:

- hold the **PAGE** key, at the right hand side of the keypad
- press the **step key** for the required page number

For example, to select page 5, (for MIDI config options):



The display will change to show you the page 5 options, and the soft-keys **F1**, **F2** and **F3** take on their new functions.

**\*TIP** You can step through each group of three pages by pressing and releasing the **PAGE** key by itself, or use **FUNC+PAGE** to switch to first page in the other group – handy if your other hand is busy.

Let's look at each **play mode** page, and see what they all do.

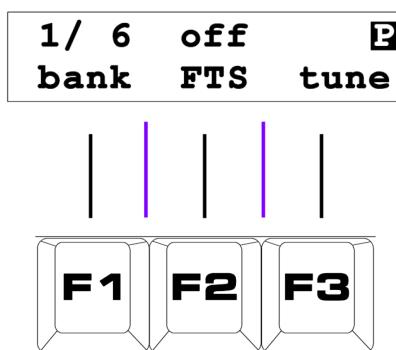
The following table summarises all six pages.

Note the inverse-video "P" in the top right of the display, showing you are in play mode.

<div style="border: 1px solid black; padding: 5px; text-align: center;"> <b>1/ 6 off P</b>  <b>bank FTS tune</b> </div> <p>page 1 - Bank/FTS/Tune</p>	<div style="border: 1px solid black; padding: 5px; text-align: center;"> <b>on off MIDIP</b>  <b>mclk thru chan</b> </div> <p>page 5 - MIDI Config</p>
<div style="border: 1px solid black; padding: 5px; text-align: center;"> <b>16 1 0 P</b>  <b>GBar PCRep PXPos</b> </div> <p>page 2 – Part Settings</p>	<div style="border: 1px solid black; padding: 5px; text-align: center;"> <b>Bank off off P</b>  <b>Init BPgm BTtempo</b> </div> <p>page 6 – Bank Functions</p>
<div style="border: 1px solid black; padding: 5px; text-align: center;"> <b>mute part-copy P</b>  <b>hold all active</b> </div> <p>page 3 – Part Functions</p>	<div style="border: 1px solid black; padding: 5px; text-align: center;"> <b>-- SysX -- User P</b>  <b>send recv Conf</b> </div> <p>page 7 – Sysex/User Config</p>

## Page 1 - Bank / FTS / Tune

Page 1 is the default page when P3 is turned on, or when it returns from another mode. The display will look like this:



Key	Label	Function
F1	bank	Bank Select
F2	FTS	Edit FTS settings
F3	tune	Send reference notes on all active tracks

## F1 - Bank Select

On page 1 the F1 soft-key is labelled “bank”.

The numbers on the top line above “bank”, in the example above “1/ 6”, show the currently selected bank, and the total number of banks available.

Each bank holds a separate set of **patterns** and **parts**, as well as some configuration settings.

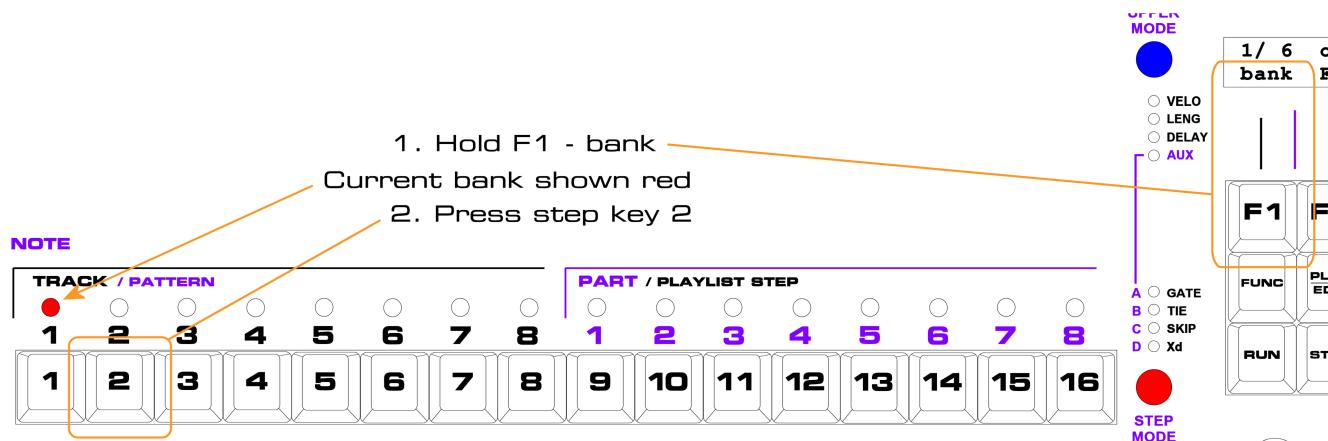
To select another bank:

- hold the “bank” soft-key, F1.

All of the step LEDs will go out, except the one above the current bank number, which will be red

- press the **step key** for the new bank number you want to select

For example, if we want to switch from bank 1 to bank 2:



P3 will switch to the new bank immediately if stopped, or at the end of the next bar while running.

## F2 - Force To Scale

The F2 soft-key on page 1 is labelled “FTS”, for Force-To-Scale.

FTS can 'quantize' all transmitted notes in to a chosen scale (or key signature).

FTS has a number of uses:

- When improvising live edits, and you don't want to hit a bum note
- 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

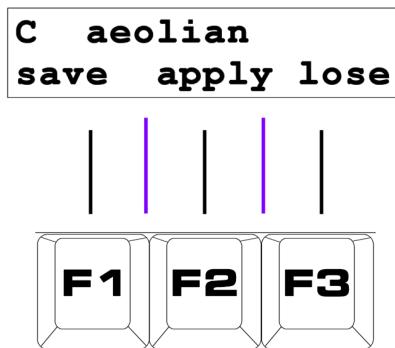
Note that **the pattern itself is not changed** – the notes are corrected **dynamically** as the pattern plays.

\*TIP Individual steps in patterns can be set to ignore FTS and also playlist transposition. This option will be described in the Pattern Edit section later.

## Configuring FTS

To configure the FTS settings press the “FTS” soft-key.

This will enter the FTS config page, which looks something like this:



Key	Label	Function
F1	save	Store the new FTS settings and return to play mode
F2	apply	Temporarily apply the new settings, but remain on this page
F3	lose	Restore the previous settings and return to play mode

The top line of the display shows the current FTS **root note** and **scale**.

The step LEDs will also show the current **root note** - the LED above the step key for the root note will be amber.

## Selecting the FTS Root Note

The step key numbers correspond to the root notes as follows:

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

To change the root note, press the step key for the new root.

## Applying FTS Changes

Changes to the FTS root or scale selection do not take effect immediately - you must first press the “**apply**” soft-key, **F2**.

To indicate that a new selection is pending an asterisk, “\*”, will appear in front of “apply” in the display.

Note that “apply” does not save the settings – it just activates your new selection.

## Sharps or Flats

You may find that the note name for the **root note** is not what you would like. For example, you think of the key as **E flat** rather than **D sharp**.

You can switch the note display format as follows:

- **hold** the **FUNC** key on the keypad
- **press** the **step key** for the FTS root note

This will toggle the display of 'black' note names between sharps and flats.

This setting is stored for each part, and also determines how note names will be shown in **pattern edit**.

## Selecting the FTS Scale

To change the FTS **scale**, use the **DATA** knob to scroll through the available options.

There are 23 preset and 9 user defined scales.

Again, the change will not take effect until "apply" is pressed.

See **Appendix A** for a list of the preset scale definitions.

## Exit FTS Config

To exit the FTS configuration page:

- **press "save", F1**, to store the new FTS settings, and return to play mode
- OR**
- **press "lose", F3**, to restore the previous settings, and return to play mode

The FTS settings are stored for each part.

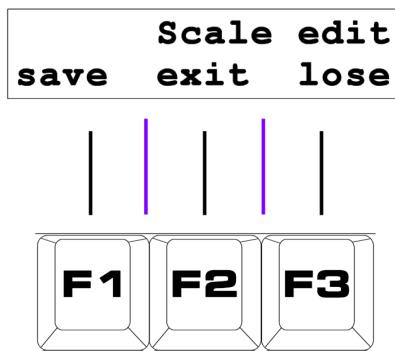
They are recalled whenever a part becomes active.

## FUNC + F2 - FTS Scale Editor

There is a **second function** for the “FTS” soft-key on play page 1.  
It is also used to access the **FTS Scale Edit** page:

- hold the FUNC key
- press the “FTS” soft-key, F2

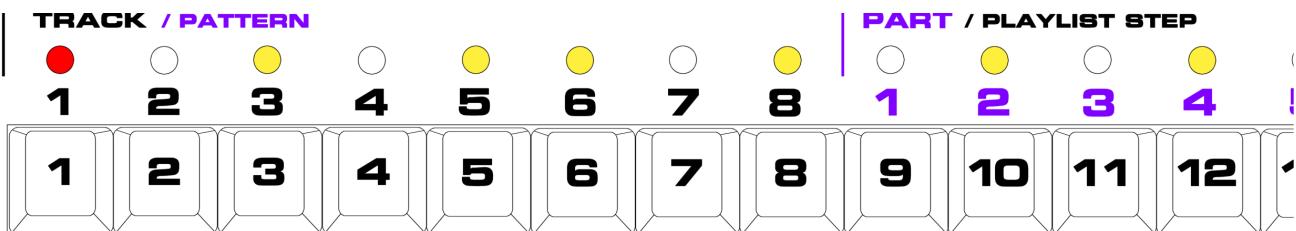
The FTS Scale Edit page looks like this:



Key	Label	Function
F1	save	Save the modified scale and return to play mode
F2	exit	Return to play mode, with the modified scale active, but not saved
F3	lose	Restore the previous scale

In FTS Scale Edit, the step LEDs will show the notes in the scale.

For example, the Major scale will look like this:



The root note, on step 1, is **red** because it **cannot be turned off**.

The other step LEDs from **2 to 12** will be **amber** to indicate if that note is enabled in the scale.  
Each note can be turned on or off using the **step keys**.

The actual notes on each step key vary depending on the FTS root note selected – step key 1 is always the root.

For a root of C, the notes will match those for root note selection:

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

As you toggle each note on or off, the note name will appear in the top left of the display. This makes it easy to see which notes are in the scale, regardless of a change in root note.

## FTS Scale Edits

Changes to the FTS scale apply immediately.

As patterns play, any note that is not in the FTS scale is moved to the nearest note that is.

**\*TIP** FTS looks for the nearest note in alternating directions.

If a note is not in the scale, P3 will test if the note one semitone below is.

If that note is not, it will test the note one semitone above.

It will continue in this way, two below, two above, three below, etc, until it finds a note in the FTS scale.

There are three ways to leave the FTS Scale Edit page, via the soft-keys:

- press “**save**”, F1, to store the scale definition and return to play mode – note that preset scales do not save permanently
- press “**exit**”, F2, to leave the modified scale active, without saving the changes permanently, and return to play mode
- press “**lose**”, F3, to restore the previous scale definition and return to play mode

Remember all scales can be edited, but changes to the preset scales will be lost when a new scale is selected.

For the 9 user defined scales, “**save**” stores the changes permanently.

## F3 - Tune

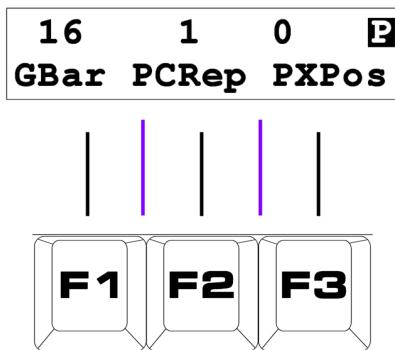
The tune soft-key can be used to send a middle C note-on message on all currently active tracks.

This is useful for testing your MIDI connections, or for tuning unstable analogue synths in variable temperature environments.

Press the soft-key again to send note-offs, if you ever get that minimoog tuned.

## Page 2 - Part Settings

The second play mode page is the **part settings** page, which will look like this:



Key	Label	Function
F1	GBar	Edit Global Bar Length
F2	PCRep	Edit Part Chain Repeat count
F3	PXPos	Edit Part Transpose

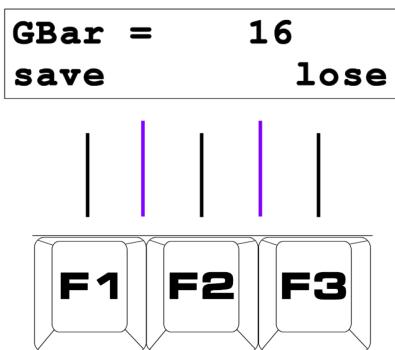
The settings for **Global Bar Length**, **Part Chain Repeats**, and **Part Transpose** are automatically stored for each part, along with the track mutes, just before a new part is selected.

\*TIP Holding FUNC while selecting a new part will switch to the new part **without saving** recently changed settings in the current part.

To edit each of the part values, press the soft-key below it.

This will enter a data entry page, where a new value can be selected using the **DATA** knob.

The value edit page for all three values is similar, and will look something like this:



Key	Label	Function
F1	save	Store the new value and return to play mode
F2	(no function)	
F3	lose	Restore the previous value and return to play mode

If you select page 2 while a **part chain** is active, the part values may change as P3 steps through each part in the chain, reflecting the different settings for each part.

Note that as soon as you choose to edit one of the part settings, the part chain will freeze on the current part until you leave the data entry page.

### F1 - Global Bar (GBar)

The **global bar** setting is the length of one bar of the current part in 16th note steps.

Since the pattern lengths on each track may be different, this value is used by P3 to control when part changes take place.

A useful feature of P3 is that you can automatically re-synchronise any pattern (usually those with unusual lengths) with the start of the global bar if required.

This feature will be fully explained in the later section on playlist edit.

### F2 - Part Chain Repeats (PCRep)

The **part chain repeats** count only takes effect when a part chain is active.

Each part in a chain will repeat the number of times set in its **PCRep** value, before moving on to the next part in the chain.

The total number of steps a part will play for in a chain is therefore the global bar length multiplied by the part chain repeat count.

### F3 - Part Transpose (PXPos)

**Part transpose** is a global transpose of all tracks in a part.

The value is shown in semitones, and is applied to all notes on all tracks except steps where transpose defeat is enabled.

**\*TIP** The PXPos value can be positive or negative.

To make it simpler to find the zero position on the DATA knob, zero is located at the fully anti-clockwise end of rotation, with the value rising to the maximum positive value at the mid-point of the knob.

The furthest negative value is found just after the mid-point, with negative values continuing up to minus 1 at the fully clockwise position. This may take a little getting used to, but it's simpler than hunting for a commonly used value (zero) at the mid-point of a knob.

## Pre-/Post-FTS Transpose

To complicate matters a little, there are two ways part transpose can be applied. These are **pre**- and **post-FTS** (Force to scale).

To toggle the pre-/post-FTS setting:

- hold the **FUNC** key
- press the “**PXPos**” soft-key, F3

A letter “F” will appear in the display after the value of **PXPos** to indicate that pre-FTS transpose is enabled. If there is no “F”, transposition is post-FTS.

There are two reasons why you may want to transpose a part.

It may be because you are moving patterns to a different chord root in the same key, or because you are transposing to a different key.

For example, say you have a pattern that is playing a C major arpeggio.

You want to transpose that pattern, and all the other patterns in your part, to E major. Your FTS setting is currently C Major.

If you use part transpose to shift the part up by 4 semitones, and you have pre-FTS transpose enabled, then your major arpeggio will become minor.

This is because the E, shifted up by 4 semitones, becomes G#.

G#, the major third in an E major arpeggio is not in the C Major scale, so the transposed note will be FTS'd to G – a minor third.

If the part transpose were set to post-FTS, then the arpeggio remains major. This is because E is in C Major, and the FTS happens before the transposition.

By setting part transpose to post-FTS, in effect you are also transposing the FTS root note as well.

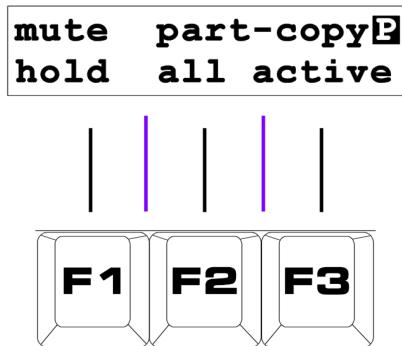
Don't worry too much if this seems complicated.

In practice, you can just try both options, and go with whichever one sounds best.

## Page 3 - Part Functions

This page has three functions for use with parts.

The display looks like this:



Key	Label	Function
F1	mute hold	Copy the current track mutes to a new part
F2	part-copy all	Copy all part data to a new part
F3	part-copy active	Copy common part settings, and active track playlists to a new part

### F1 - Mute Hold

As mentioned previously, each part stores the mute settings for each of P3's eight tracks.

When a new part is selected, the mute status at the time that part was last active is recalled.

The “**mute hold**” function allows you to over-ride this behaviour.

If you select a new part while holding this key, the current track mute status will be carried over to the newly selected part.

### F2 – Part-Copy All

### F3 – Part-Copy Active

The **part-copy** functions allow you to copy all settings from the current part to another part. This includes the **FTS** settings, the values of **GBar**, **PCRep**, **PXPos** and the **playlists**,

There is an important difference between the two options:

- “**all**” copies the playlists for **all eight tracks**
- “**active**” only copies the playlists for **active tracks**

It's important to remember that the **playlists** in each **part** just say which **patterns** should play on each track – they don't hold the **patterns** themselves.

So when you copy a **part**, the destination **part** will play the **same patterns** as the source **part** – the **patterns are not copied**.

If you edit the patterns in a copied part, the altered patterns will play in any other part which uses them.

For this reason, the “part-copy”, “active” function may be more commonly used, as this allows you to create a new part with some tracks playing the same patterns as before, but other tracks playing new patterns over the backing of the copied tracks.

To copy a part:

- **hold** the part-copy soft-key for “all” or “active”
- **press** the **PART** key for the destination part

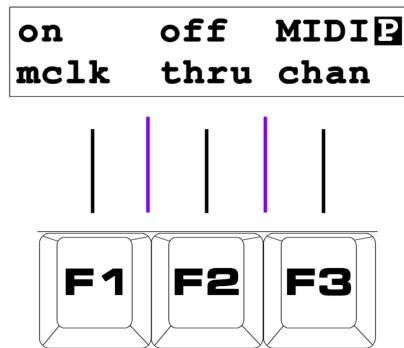
Don't try to select a part chain while copying - it won't work.

As the newly copied part is selected, current part settings will also be saved in the part being copied.

**\*TIP** Holding **FUNC** while performing a mute hold or part copy will copy the current settings to the new part **without saving** recently changed settings in the source part.

## Page 5 - MIDI Config

Page 5 provides access to P3's MIDI configuration, and has the following options:



Key	Label	Function
F1	mclk	Toggle transmission of MIDI clock on the main outputs
F2	Thru Conf	Configure soft-thru and record channel settings
F3	MIDI chan	Configure the MIDI channels for tracks

### F1 - MIDI Clock

The first option, "mclk", toggles the sending of MIDI clock at P3's main MIDI outputs.

If you don't have any slaved devices attached to the main outputs, it's better to turn it off and save bandwidth.

Note that the **SYNC** output is not affected by this setting.

The third option here, "MIDI chan", is the most important, so we'll look at it now and come back to "Thru Conf" later.

## Setting the MIDI Channel for P3 Tracks

The “MIDI chan” soft-key is used to enter the **MIDI channel select** page, where you can configure the MIDI output channel for each of P3’s tracks.

To enter the MIDI channel select page for a track:

- hold the “MIDI chan” key, F3
- press the **TRACK key, 1 to 8**, for the track to configure

All of the step LEDs will go out, except for one in red, and the display will look something like this:

**Select channel..**  
**(trk= 1) cancel**

Only soft-key **F3** is active in this page, to “cancel” the channel selection page.

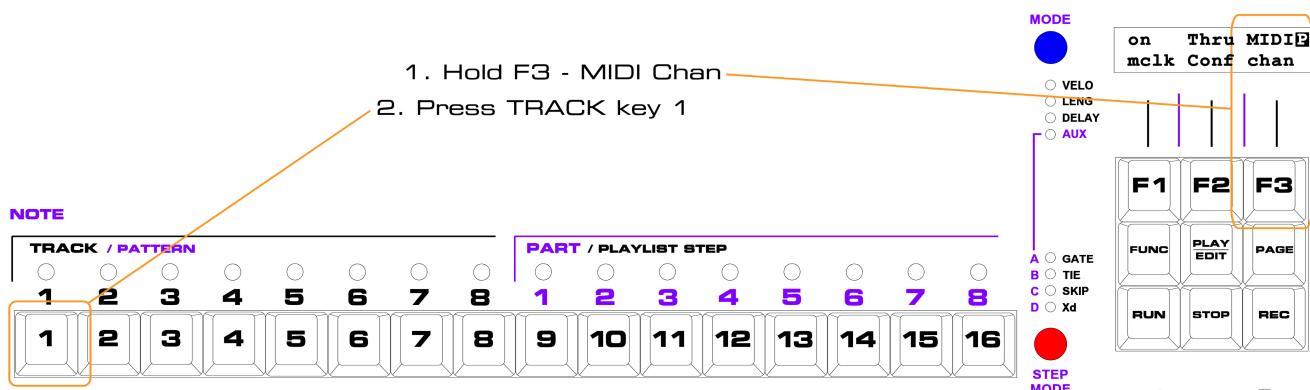
“(trk= n)” on the bottom left of the display confirms the number of the track you are changing.

The LED above the step number of the current MIDI channel for the track will be red.

To change the MIDI channel, press the step key for the desired channel number, **1 to 16**.

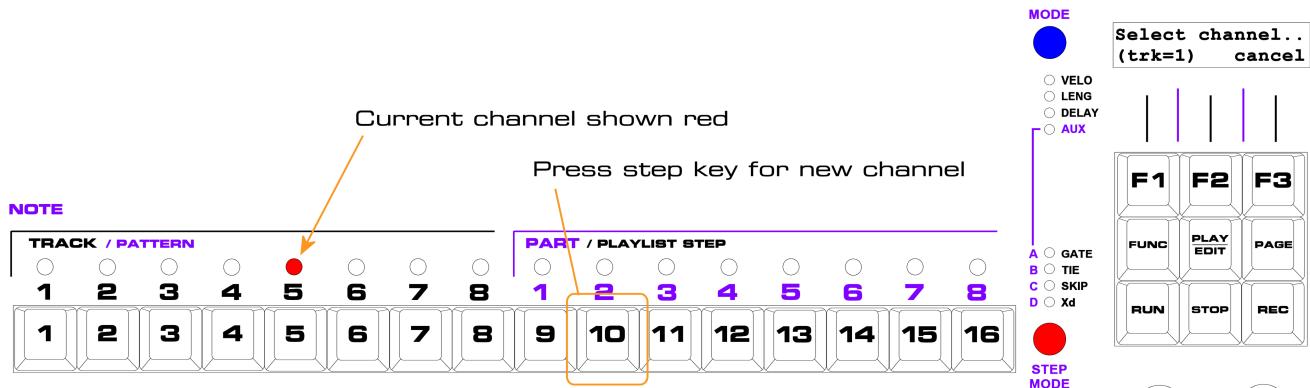
Or, press “cancel”, F3, to leave the MIDI channel selection page without changing the channel.

For example, let’s say **track 1** is assigned to **MIDI channel 5**, and you want to change it to **channel 10**. First, enter the MIDI channel select page like this:



This will bring up the MIDI channel select page for track 1.

In the MIDI channel select page, step LED 5 will be red, showing track 1 is currently on MIDI channel 5:



Press step key 10, and track 1 will be assigned to MIDI channel 10. The step 10 LED will light red briefly. Then P3 will exit the channel select page, and the step LEDs will return to the current track and part status.

## MIDI Channel Scope

MIDI channel selections for each track are global by default, meaning that they are the same for all banks.

However, you can enable different channel settings for each bank, using the “Mchans per bank” user config option.

User config options are described in the section on play mode page 7.

## Special Channels

As well as the MIDI channel settings for each track, there are two other special purpose MIDI channels used by P3.

One is used for sending an optional program change message when a new bank is selected  
Another channel may be used for sending and receiving special P3 remote control messages.

To configure the bank program change channel, use the “MIDI Chan” soft-key with step key 9. The bottom left of the display will show “(pgm)” while you make this selection.

To configure the remote control channel, use the “MIDI Chan” soft-key with step key 10. The bottom left of the display will show “(rem)” while you make this selection.

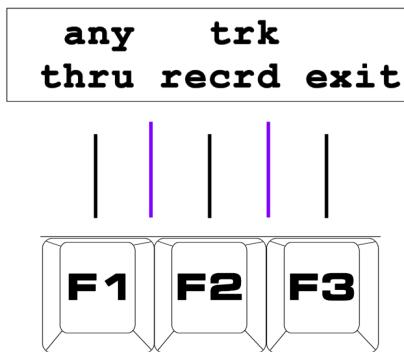
## F2 - MIDI Thru and Record Config

Pressing “**Thru Conf**”, **F2**, brings up the MIDI Soft-Thru and Record channel configuration page.

Here you can configure whether MIDI messages arriving at P3's MIDI input are passed through to the MIDI output, and re-route them to a single channel if required.

You can also configure the MIDI channel for real-time recording input from a keyboard or other MIDI controller.

The “Thru Conf” page will look something like this:



Key	Label	Function
F1	thru	Configure soft-thru channelisation
F2	recrd	Configure the real-time record channel
F3	exit	Return to play mode

### F1 - Soft-Thru Channelisation

The first option, “**thru**”, is the soft-thru channelisation setting. This can be “**any**”, “**off**”, or a MIDI channel number from **1** to **16**.

To select “any” or “off”:

- **press and release** the “thru” soft-key, **F1**, **by itself**.

Repeated presses toggle between these two values.

To select a specific MIDI channel number:

- **hold** the “thru” soft-key, **F1**
- **press the step key** for the MIDI channel to be used

Turning this option “off” means no incoming MIDI messages will be passed.

A setting of “any” means P3 will echo any incoming MIDI messages straight to the MIDI outputs.

Setting a specific MIDI channel number will cause all notes received at the MIDI input to be re-transmitted on the specified channel.

## F2 - Real-Time Record Channel

The second option on this page, “**recre**”, is the MIDI channel to be used for **real-time record** and **arpeggio capture** modes, which are described in later chapters.

This setting can be “**any**”, “**trk**”, or a MIDI channel number from **1** to **16**.

These values are selected using the soft-key in the same way as for the soft-thru options – toggle between “any” and “trk” with single presses, or hold the soft-key and press a step key for a specific MIDI channel.

Setting “**recre**” to “**any**” means all incoming notes will be recorded while record is active, from any MIDI channel.

Setting a specific channel from 1 to 16 means that only notes arriving on that channel will be recorded. All other channels will pass through according to the soft-thru setting.

A setting of “**trk**” means the record input channel will automatically switch to the **output** MIDI channel of the track being recorded upon.

Note that the record channel setting only relates to the MIDI channel on which notes will be **received** during record.

If the record “**mthru**” option (described in the later chapter on real-time record) is active, the recorded notes will be echoed on the MIDI channel assigned to the track being recorded upon.

## Soft-Thru and Record FTS

For soft-thru and MIDI record input, the current Force To Scale settings can also be applied. Hold the **FUNC** key while pressing the “**thru**” or “**recre**” soft-keys to toggle the FTS settings. If FTS is active for either option, the setting is shown prefixed with “**F-**”.

FTS on soft-thru notes will force any notes passed into the current scale.

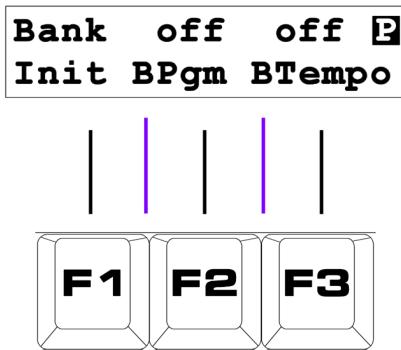
Record FTS is slightly different, and will be explained in the chapter on Real-time record.

## Exit “Thru Conf”

Once you have configured the thru and record channels, press “**exit**”, **F3**, to return to play mode.

## Page 6 - Bank functions

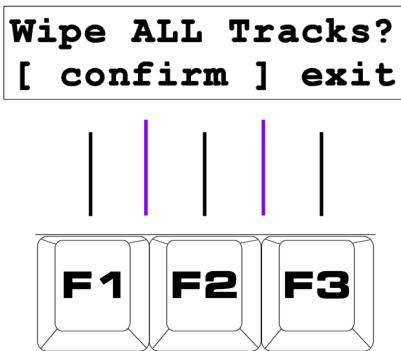
This page provides bank specific options:



Key	Label	Function
F1	Bank Init	Wipe all data in this bank
F2	BPgm	Configure a bank program change number
F3	BTempo	Configure a bank initial tempo

### F1 - Bank Initialise

The “**Bank Init**” function is used to completely wipe all the patterns and parts in the currently selected bank. To ensure it is not used by accident, after pressing the soft-key, the following display will be shown:



**IMPORTANT:** In order to confirm that the bank should be initialised, you need to **press both keys F1 and F2 together** at the same time.

There is no way to get your bank data back unless you made a sysex backup, so **USE WITH CAUTION**.

### Partial Bank Init

There is also a **partial bank init** function which will only erase patterns and playlists for **active tracks**.

If a track is muted when you select **partial bank init**, its pattern data will be left intact, although all common part data will still be affected, so FTS settings may disappear and make it seem like some patterns have changed.

To perform a **partial bank init**:

- hold the **FUNC** key
- press the “Bank Init” soft-key, **F1**.

The confirmation display will read:

**Wipe ACTIVE Trx?**  
[ **confirm** ] **exit**

Partial bank init must be confirmed in the same way as a full Bank Init, with **F1** and **F2**.

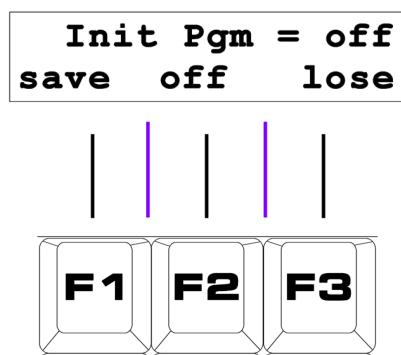
## F2 - Bank Program Change

The “**BPgm**” option allows you to set a program number to be sent in a MIDI program change message when a bank is selected.

The message is sent on the “**pgm**” channel described in the previous section on assigning MIDI channels.

This option is designed for use with multi-timbral synth modules where you are triggering different sounds with a number of tracks on the P3, each on their own MIDI channel, but you need to recall multi-timbral setup patches on the module using a separate global MIDI channel.

Pressing the “**BPgm**” soft-key, **F2**, enters this page:



Key	Label	Function
F1	save	Store the new setting and return to play mode
F2	off	Disable this setting
F3	lose	Restore the previous setting and return to play mode

In this page, use the **DATA** knob to select the required program number, or press the “**off**” soft-key, **F2**, to de-activate the setting for the current bank.

The “**save**” and “**lose**” options store the new value, or recall the original value as usual, and return to the Bank function page.

### F3 - Bank Initial Tempo

The “**BTempo**” option allows you to set a value for tempo that will be recalled when a bank is selected, or the sequencer re-started.

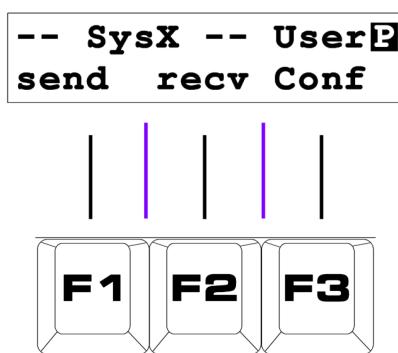
Changing the BTempo value is a similar process as for initial program change number.

\*TIP Bank tempo will normally be over-ridden if the **TEMPO** knob is adjusted. There is a user config option, “Lock BTTempo”, which disables the **TEMPO** knob when a BTTempo value is set.

### Page 7 : Sysex & User Config

The final play mode page provides System Exclusive dumping and uploading of P3's memory, as well as access to configuration options that customise P3's behaviour to the user's preference.

The page display looks like this:



Key	Label	Function
F1	send	Dump all data via System Exclusive
F2	recv	Enter System Exclusive receive mode
F3	User Conf	Enter User Configuration option menu

## F1 - System Exclusive - Send

The “**send**” soft-key, F1, initiates a **complete dump** of all the bank and FTS settings, parts and patterns via MIDI System Exclusive.

Status messages are shown while the dump takes place, and transmission can be aborted before completion using the F3 key.

## F2 - System Exclusive – Receive

The “**recv**” soft-key, F2, puts P3 into **receive mode**, which can be used for two different purposes.

In **receive mode**, any sysex message blocks containing configuration, part or pattern data will be received and stored in main memory after checksum validation.

As blocks can be received in any order, P3 does not know when a complete dump has been received. Status messages indicate reception of each data block.

Any errors detected in a dump will cause an “X” to appear in the bottom line of the display. The last data block in a complete dump will be “pattern 170F”.

To exit receive mode, you must press “exit”, F3.

## P3 Tools

Receive mode can also be used to receive “Dump Request” messages, which originate from a computer based library utility.

In response to these messages, P3 can send any individual pattern, part or configuration block.

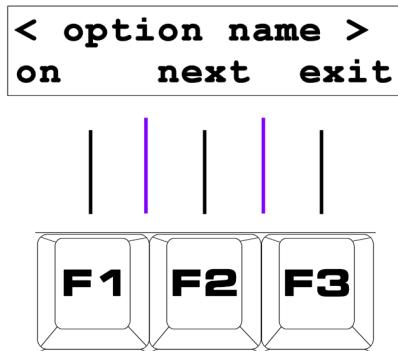
Receive mode is used with the Windows utility “P3 Tools” – an excellent librarian and data management utility written by a P3 owner.

For details of the system exclusive data format, see the separate Sysex manual.

### F3 - User Config Options

The “**User Conf**” soft-key, F3, enters the User Configuration Options menu.

These settings are on/off options that alter the normal behaviour of P3. The options menu looks like this:



Key	Label	Function
F1	on/off	Toggles the current setting on or off
F2	next	Step through the config option list
F3	exit	Return to play mode

You can step though a number of options, with the option description shown on the top line of the display.

The soft-keys in this menu have the following functions:

- **F1** toggles the status of the current config option
- “**next**”, **F2**, advances to the next option in the list
- “**exit**”, **F3**, will exit and return to play mode

Note that changing an option setting takes effect immediately.

See **Appendix B** for a list and description of the **User Conf** options.

## *Other Play Mode Features*

There are a few remaining features of play mode to describe before we move on to pattern edit.

### **Play Mode Knob Use**

Normally in play mode, the NOTE and UPPER rows of knobs have no function.

However, it is possible to assign knobs in **play mode** from **within a pattern** so that a knob can control some aspect of the pattern itself.

This advanced feature will be covered in the later section on Auxiliary Events.

Another possible use of the knobs in play mode is linked with the **User Conf** option, “Enable mixer”.

With this option on, P3 will send MIDI **volume** and **pan** controller messages from the knobs above the **part keys**.

The NOTE knobs will send volume, the UPPER knobs pan.

The messages are sent on the MIDI channels assigned to the same track number as the part number below each pair of knobs – this is to keep the knobs above each track free for use in pattern control.

### **Fine Tempo Adjust**

The P3 **TEMPO** knob adjusts the tempo over a range of 39 to 253 bpm.

The normal step in tempo as the knob is moved is 1 bpm over the middle part of the range, increasing to 2 or 4 bpm at the extremes.

However, the tempo can be adjusted in steps of 0.1 bpm over the full range, using the Fine Tempo adjust control as follows:

- **hold** the **FUNC** key
- **adjust** the **DATA** knob

The display will confirm the exact tempo selected as the knob is adjusted.

## Status LED Displays

In play mode, the otherwise unused **UPPER MODE** and **STEP MODE** status LEDs can be set to display useful information about what is going, or just to look nice.

To cycle through the display modes, press the **STEP MODE** button.

The available modes are:

- **VU meter** – transmitted velocities are shown on each group of LEDs, one set for the odd, and one for the even numbered tracks, in classic VU meter style
- **Beat** – pairs of lit LEDs indicate the current beat position
- **GBar position** – a countdown to the end of the current global bar is shown as a collapsing pattern (really just the number in binary!)
- **Track gates** – each of the eight status LEDs shows the gate status for a track, with track 1 at the top

## GBar Pos and Playlist Hold

Holding the **UPPER MODE** button while in play mode gives you access to the **playlist hold** flags for each track, and also shows the number of steps left until the next **GBar** end.

These flags allow you to suspend the progress of the playlist on any track.

While **UPPER MODE** is held, the track LEDs will normally indicate green for each track.

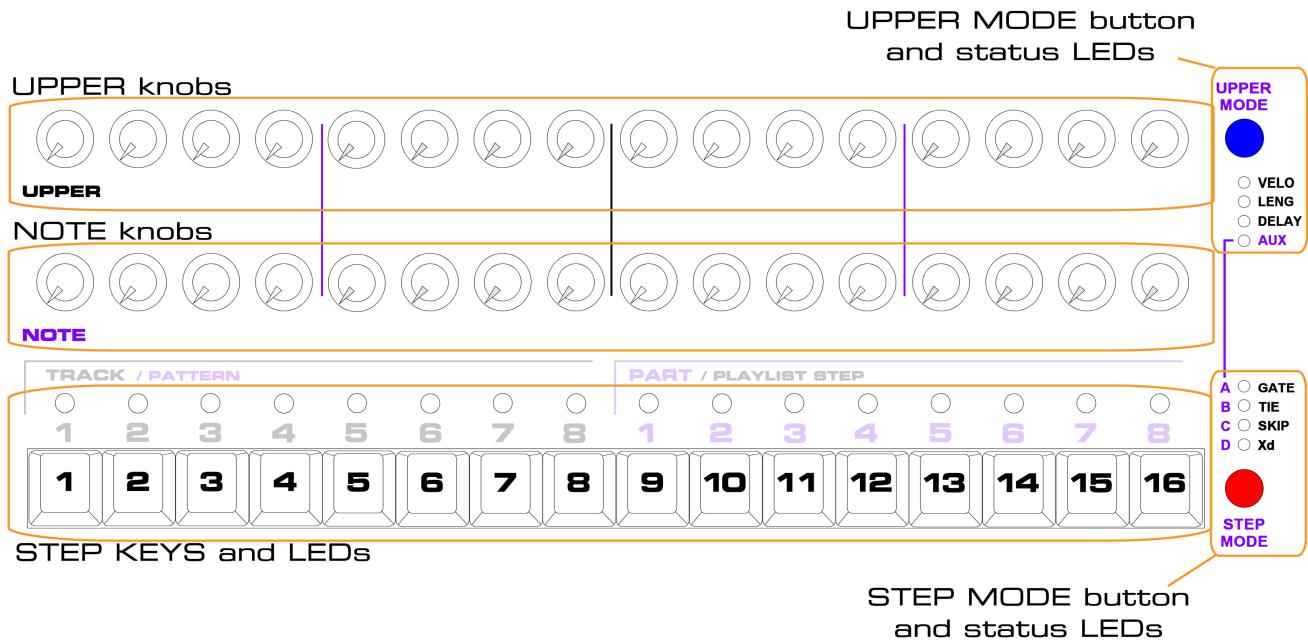
By pressing a **TRACK key**, the advance of the playlist on that track is suspended, and the LED shows red.

Another press of the **TRACK key** will return the track to normal.

## Pattern Edit

You should have had a taste of **pattern edit** already from the getting started section, but let's take the full tour now.

The most important controls in pattern edit are the 16 **step keys**, the 16 **NOTE knobs**, the 16 **UPPER knobs**, and the **STEP MODE** and **UPPER MODE** buttons and status LEDs, shown here:



### *What's In A Pattern?*

Each step in a P3 **pattern** has a number of different **values**, such as the **gate**, **note** and **velocity** values you used in the getting started section.

The set of 16 of each **value** in a **pattern** is called a **row**.

Some of the rows are **numeric** – each step has a range of up to 128 different values, for example the **note** and **velocity** rows.

Other rows are **status** rows – each step can only be **on** or **off**.

For example, the **gate** row is a status row.

There are a number of other non-step values for each pattern, such as the direction, timebase and last step, but these are edited using soft-key controls, which we'll come back to later.

## *Editing Step Values*

Since there are only three rows of controls on P3's front panel, clearly some of the controls have to be shared between the pattern rows.

The **note** row is the most important, so it gets a dedicated set of controls. The **NOTE knobs** **always edit the note row** in a pattern.

The other pattern rows are selected for editing using the **STEP MODE** and **UPPER MODE** buttons.

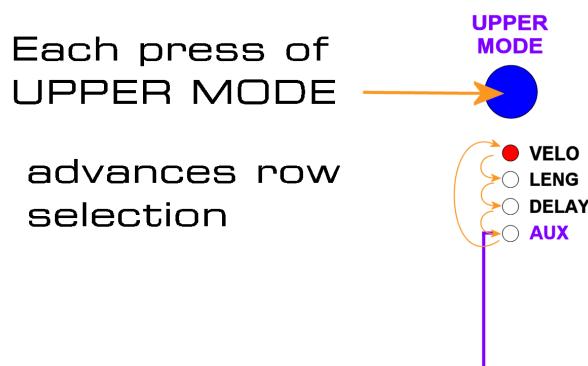
### **UPPER MODE**

You have already used the **UPPER** knobs for editing **velocity** in the getting started section. These knobs can also be assigned to any of the other numeric rows in a P3 pattern.

The **UPPER** knobs are assigned using the **UPPER MODE** button.

Below this button are four status LEDs, which show the current row being edited.

Each press of the **UPPER MODE** button advances to the next row, as shown in this diagram:



The following table shows the name and purpose of each of the rows edited using the **UPPER** knobs:

Row	Purpose
VELO	MIDI velocity value, from 1 to 127
LENG	Gate length, from 1 to 12
DELAY	Gate delay, from 0 to 11
AUX A	The four <b>Auxiliary</b> rows – take numeric values for a variety of different purposes
AUX B	
AUX C	
AUX D	

## Velocity

Velocity is the **default** assignment of the UPPER knobs in pattern edit .  
The status LED for velocity is labelled **VELO**

The range of values is from **1** to **127**.  
The value sets the velocity of any MIDI note message sent from a step.

## Gate Length

Gate length is used to control the “on time” of the MIDI note within each step.  
The status LED for length is labelled **LENG**  
The range is from **1** to **12**.

If you think of each step as lasting for 12 ticks (where a tick is 1/48<sup>th</sup> of a quarter note), the gate length determines how many of those ticks the MIDI note is held on for.

A setting of **1** produces a very short note.  
A setting of **12** means there will be no gap at all between the note on one step and a note on the following step.

Since gate length is a **proportion of the step time**, it varies according to the pattern timebase and tempo.

**\*TIP** Gate length doesn't make much difference on a synth sound that has a significant release time.  
Try out different gate lengths using a sound with very short attack and release times and a high sustain level.  
You'll soon hear the effect it has.

## Gate Delay

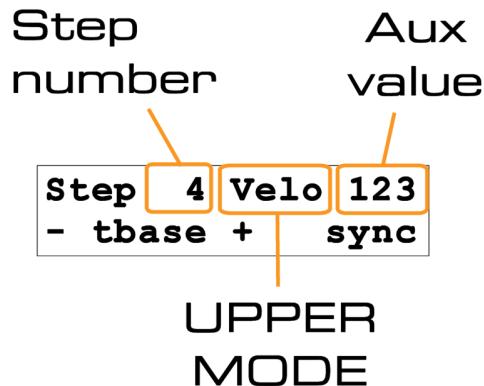
Gate delay allows you to delay the start of a note, so that it occurs sometime after the beat.  
The status LED for delay is labelled **DELAY**  
The range is from **0** to **11**.

As with length, think of each step as being 12 ticks long.  
Normally the note starts on the very first of the 12 ticks.  
If you set a non-zero value for **delay**, then the note will not start until later in the step.

## Display of Changed Values

As you edit each of the **velocity**, **length** or **delay** values, you may want to know exactly what value you are entering.

New values are always shown in the display, along with the step number being edited, like this:



## Auxiliary Rows

You will have seen from the earlier table that each P3 pattern has **four auxiliary rows**. However, there is only one **UPPER MODE** status LED labelled **AUX**.

The explanation for this is that each of the four **numeric auxiliary** rows has a matching **status row**.

Selection of the auxiliary row to edit is made by setting **UPPER MODE** to **AUX**, then using the **STEP MODE** button to select from the four available auxes, A, B, C or D. So we'll come back to the auxiliary rows once we've had a look at the status rows, and **STEP MODE** selection.

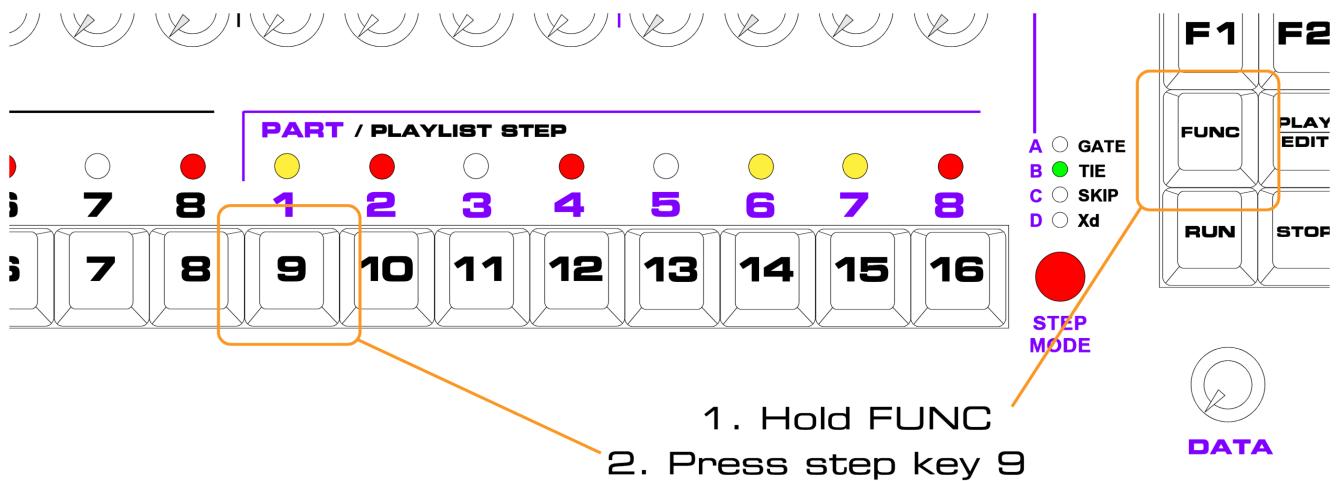
Before that, let's see how you can review the values you have entered for your numeric rows...

## Viewing numeric row values

Since the knobs will not always reflect the stored values in a pattern, and you probably couldn't gauge their position with absolute accuracy if they did, you can view the **numeric values** for each pattern step as follows:

- hold down the FUNCkey
- press the step key for the step you want to inspect

For example, if you want to see what the numeric values on step 9 are:



When you press the step key, the display will briefly show all the numeric values on that step, like this:

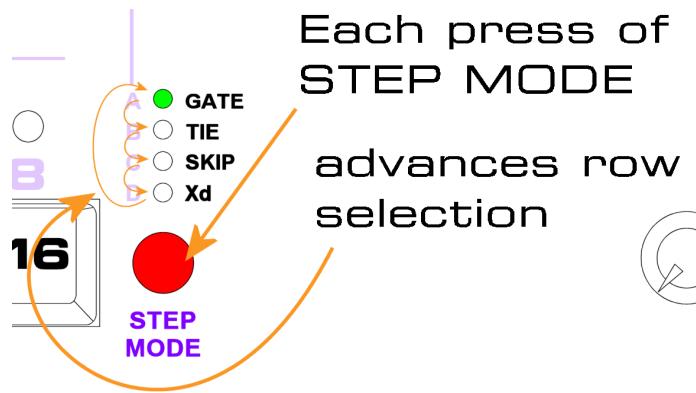
Step Number	Velocity	Note value
Step	V:120	Note
9	I:10	C#5
Gate Length	Gate	Delay

## STEP MODE

You have already used the **step keys** for editing the **gate** status in the getting started section.

The step keys can also be assigned to any of the other **status rows** in a P3 pattern.  
The **step keys** are assigned using the **STEP MODE** button.

Above this button are four status LEDs, which show the current row being edited.  
Each press of the **STEP MODE** button advances to the next row, like this:



Since the status rows have **on / off** values, their current setting can be shown on the LEDs above each step key.

The default assignment for the step keys when you enter pattern edit is the **gate** row.

### STEP LED Colours

P3's step LEDs are **tri-colour** types.

Inside each LED there is a **red** and a **green** element.

Either can be turned on independently, or both can be on at the same time.

Since **gate** is the most important status row:

- **gate status** is always shown in **red** on the step LEDs.
- **All other status values** are shown in **green** on the step LEDs

Whatever status row you select to edit, you can always see what the gate status on each step is.

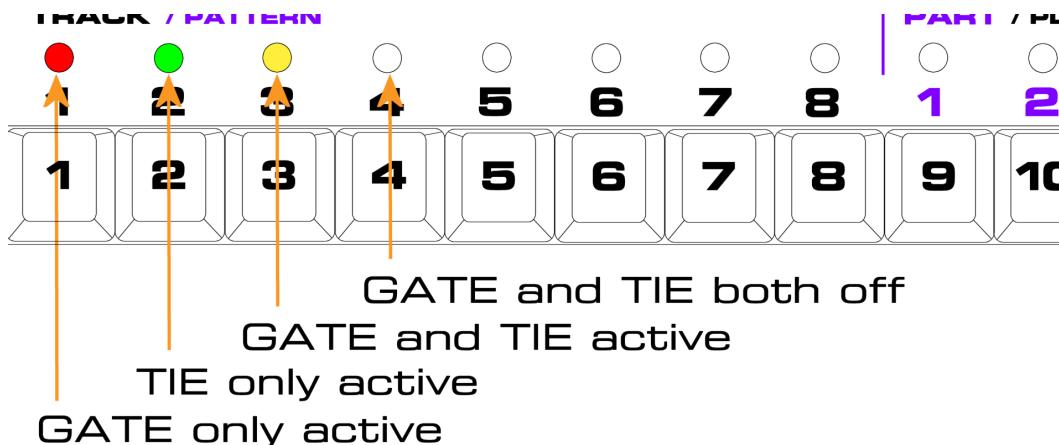
Since gate is shown in red, and other status values may be shown in green, it seems there is a problem if both gate and another status value are active on the same step.

In practice, you just need to remember:

- **green** and **red** together gives **amber**

For example, say you have the **STEP MODE** set to **TIE**.

The LED colours on each step will show the different possible combinations of **gate** and **tie** like this:



To get a feel for how this works, you might like to set every other **gate** on in a pattern.

Then press **STEP MODE** to advance to the **tie** row.

Toggle the **tie** setting off and on for various steps, and you'll see the three different colours for each combination of **tie** and **gate**.

Once you understand how the values will be displayed, let's look at the rest of the status rows.

The full complement of status rows and the purpose of each are shown in the following table:

Row	Purpose
GATE	Controls whether a note is sent on a step
TIE	Either makes the note on a step <b>over-lap a different note</b> on the next step, or <b>stretches the same note</b> over multiple steps
SKIP	Indicates this step should be skipped over during playback
Xd	Disables the effect of any transpose setting on a step
AUX A	The four <b>Auxiliary status rows</b> enable or disable the matching <b>Auxiliary</b> row on each step
AUX B	
AUX C	
AUX C	

## GATE

You should really know what this does by now...  
Notes are only sent on steps with gate turned on.

## TIE

As the name suggests, **tie** joins one step to the next.  
Tie should be set on the first step of two steps to be tied.

If the **same note value** is set on tied steps, only **a single note is played**.  
The note will last from the start of the first step, until the end of the second step set by its length value.

If tie is used on a step where the next step has a **different note value**, then the **first note will over-lap the second note** – that is, the first note will not be released until just after the second note starts.

Note that you can tie as many contiguous steps together as you like.  
Same notes will stretch, and different notes will over-lap.

**\*TIP** If you are using a monophonic synth with an auto- or legato- slide feature, you can use the TIE setting to trigger TB303-style slides.

## SKIP

The **skip** row allows you to selectively **remove** steps from the pattern.

The step values are not lost; the step is simply skipped over during playback.

This can change the rhythmic feel of a pattern, or allow you to make a simple pattern evolve into a more complex one over time as a performance technique.

**\*TIP** Skipping some steps will change the apparent length of the pattern. To alter the rhythmic feel without altering the underlying length of a pattern, use skip with the global bar sync feature – see the playlist edit section for details of this.

## Xd

**Xd** is short for **Transpose Defeat**.

There are a number of ways your pattern might end up being transposed.

In some cases, you may want to transpose some steps of a pattern, but leaves others on fixed notes. The **Xd** row allows you to do this.

## AUX Status Rows

For each of the numeric Auxiliary rows, there is a matching **Auxiliary status** row that enables or disables the auxiliary on each step.

## Assigning and Using Auxiliaries

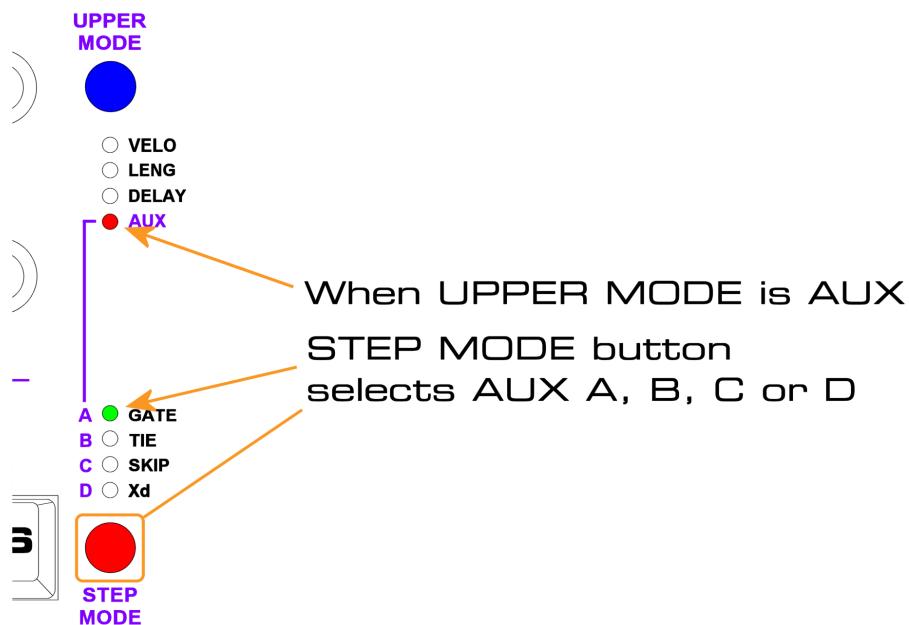
Each of the four auxiliaries (auxes for short) can be freely assigned in each pattern to send **MIDI continuous controllers, after-touch, pitch-bend or program change messages**.

Or they can be assigned to one of P3s “**Auxiliary Events**”.

As we saw earlier, to assign the **UPPER** knobs to edit an auxiliary row, press the **UPPER MODE** button until the **AUX** LED is lit:



The letters for each of the auxes, A, B, C and D are labelled on the left side of the **STEP MODE** LEDs, linked to the **AUX** upper mode status LED as shown:



With **UPPER MODE** set to **AUX**, the **UPPER** knobs edit the **numeric values** for the selected aux, while the **step keys enable or disable the aux** on each step.

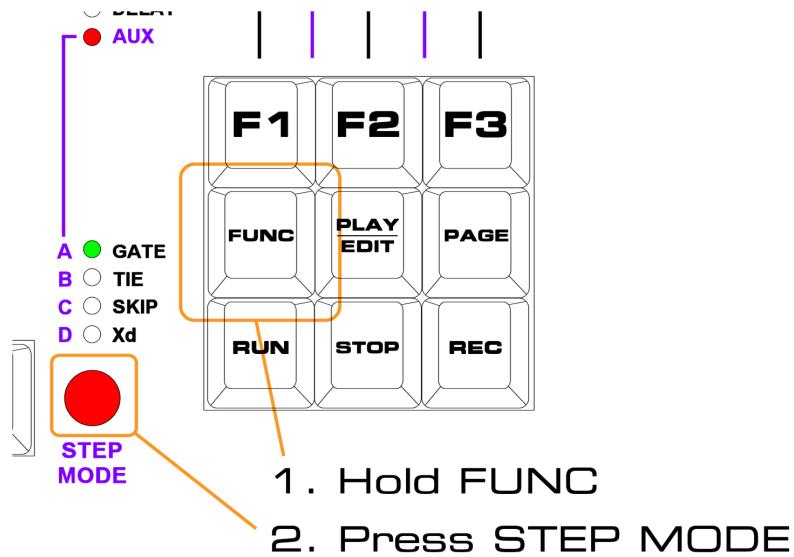
As you press the **STEP MODE** button to cycle through the auxes, the currently assigned controller number or event name briefly appears as an informational message in the display.

## The Aux Config Page

Before using an aux, you will need to assign it to the controller or event you want to use. To do this, you need to enter the **aux config** page.

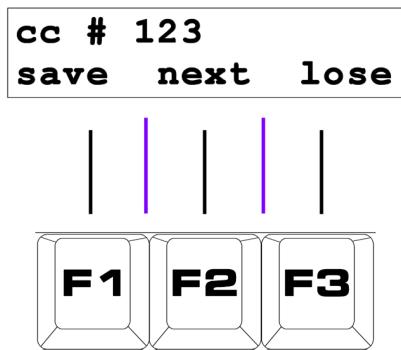
With **UPPER MODE** set to **AUX**:

- hold the **FUNC** key
- press the **STEP MODE** button



**\*TIP** As a shortcut, if you are not already in **AUX** upper mode pressing **FUNC + STEP MODE** will take you there. You will then need to press the key combination one more time to enter the config page.

This **aux config** page looks something like this:



Key	Label	Function
F1	save	Store the new assignment and return to pattern edit
F2	next	Step through the list of available functions
F3	lose	Restore the previous assignment and return

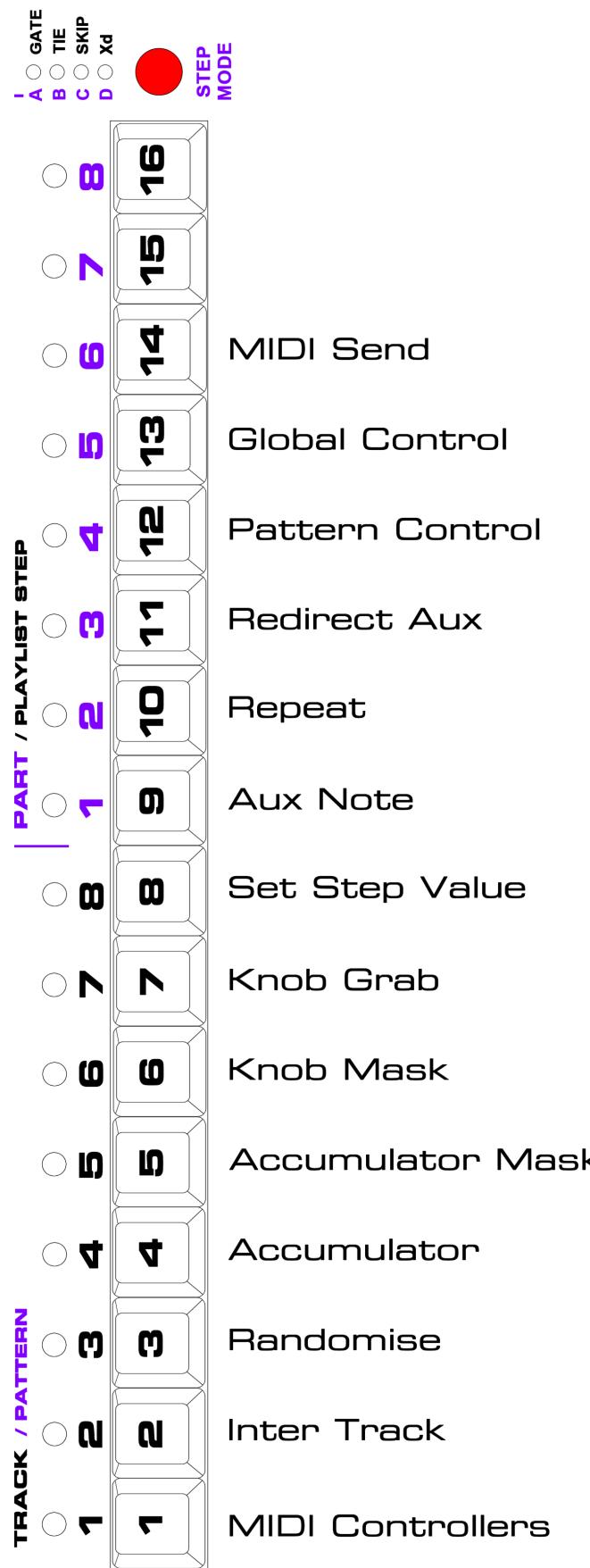
The top line of the **aux config** page shows the **event name** or **controller number** assigned to the current aux.

## Selecting Event Groups

Since there are so many different MIDI controllers and events available, they have been divided into different **groups** by function.

These groups are selected using the **step keys, 1 to 14**.

This chart shows the groups assigned to each of the step keys while in the **aux config** page:



As you select a new group using the **step keys**, the group name will appear as an informational message on the top line of the display.

You will get to know where the ones you like to use are found pretty quickly. Until then, keep this page handy.

The complete list of the events in each group can be found in the later section, **Using Auxiliary Events**.

## Selecting Controllers or Events

To select a controller or event from the **group**, use the **DATA** knob to scroll through the options, or use the “**next**” soft-key, **F2**, to step through the group members one by one.

Once you have found the controller or event you wish to use, press the “**save**” soft-key, **F1**, to store the assignment and exit the **aux config** page.

If you want to assign more than one aux at the same time, you can use the **STEP MODE** key to advance to the next aux while in the aux config page.

Note that this will also save any changed assignment for the current aux.

For example, let's assume you have just entered pattern edit , and want to assign **Aux A** in the pattern to send **MIDI after-touch**:

- switch **UPPER MODE** to **AUX**

The quickest way to do this is to hold **FUNC** and press the **STEP MODE** button, the short-cut mentioned before.

Alternatively, press **UPPER MODE** three times to select **AUX**.

- hold **FUNC** and press **STEP MODE** to enter the **aux config** page
- press **step key 14**, to select the “**MIDI Send**” group

This group contains three MIDI send functions - pitch-bend, after-touch and program change.

- either move the **DATA** knob or press the “**next**” soft-key, **F2**, until “**send midi a/t**” appears in the top line of the display
- press the “**save**” soft-key, **F1**

P3 will return to the normal pattern edit display, and you can set after-touch values to be sent on whichever steps you want using the **UPPER** knobs.

## Editing Aux Values

With the current aux configured for the function you want, you can now set the values for each step using the UPPER knobs, just as for velocity, length and delay.

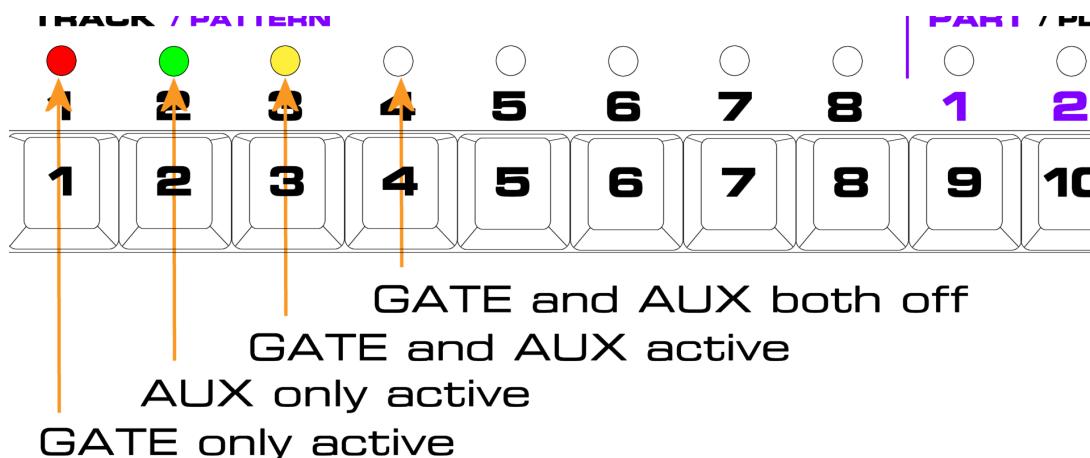
One difference with auxes is that each numeric value also has a corresponding on/off status value too.

**A controller message will only be sent, or an event processed, if the aux status is active.**

**\*TIP** By default, when you edit the value of an aux if the step is not active it will be turned on automatically. This feature can be disabled with the “aux edit sets on” **User Conf** option described in the **play mode** chapter.

As we saw earlier when looking at step LED colours for status rows, the active status for **gate** is always shown on the step LEDs in pattern edit.

This is still the case in aux mode - the step LED colours show the following conditions:

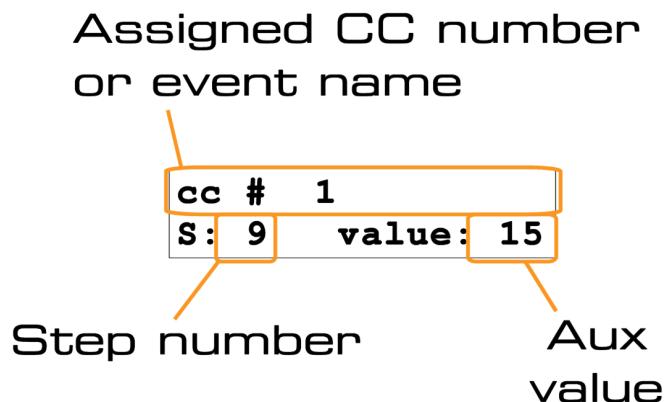


As with the other status rows, each press of a **step key** will toggle the aux status for that step on and off.

## Display of Changed Aux Values

When you edit an aux value using the UPPER knobs, the assigned **controller number** or **event name**, **step number** and **new value** will all be shown briefly in the display.

For a MIDI controller, this display will look something like this:



The display of aux values may vary when an **auxiliary event** has been assigned. This will be explained in the later section, **Using Auxiliary Events**.

## Viewing Aux Values

You can inspect the stored aux values on each step, just as for the note, velocity, gate length and delay display:

- hold the **FUNC** key
- press the **step key** for the step you want to inspect

The display format is the same as for the aux edit confirmation above.

## Summary of Aux Usage

For MIDI controllers, the use of the aux rows is easy to understand.

If the **aux status** is set **on** for a given step, the **numeric value** for that step will be sent as a **MIDI controller message** on the MIDI channel for the track.

The same is true for the MIDI Send functions.

With **auxiliary events**, in most cases extra MIDI data is not generated.

We'll look at what auxiliary events can do in a later chapter.

## Pattern Edit Pages and Soft-keys

**Pattern edit** has a number of different display **pages**.

Each **page** shows some different values or text labels on the display and has its own set of **soft-key** assignments for **F1**, **F2** and **F3**.

Some of the pages show non-step pattern values, such as direction, timebase and last step, with soft-keys for editing them.

Others have soft-keys to access pattern modifying functions and configuration options.

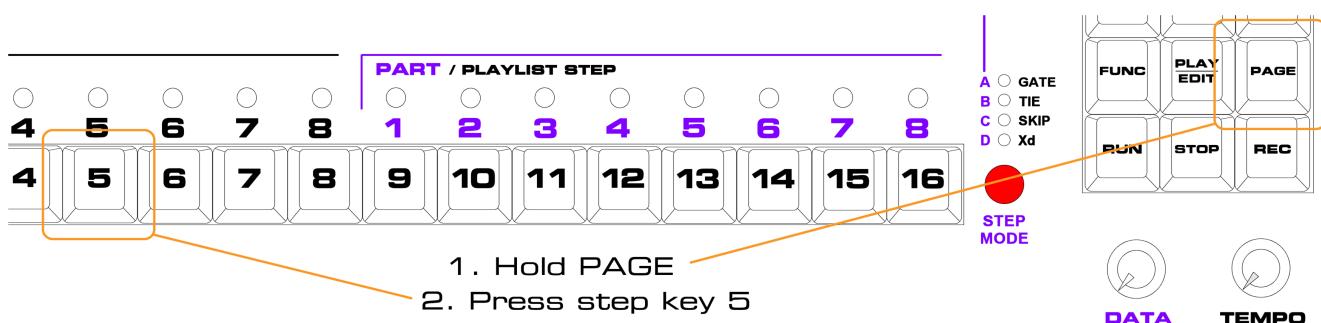
There are eight pages in total.

## Selecting Pages

To select a pattern edit page:

- hold the **PAGE** key, at the right hand side of the keypad
- press the **step key** for the required page number

For example, to select page 5, (for the **Note Randomize** page):



The display will change to show you the page 5 options, and the soft-keys **F1**, **F2** and **F3** take on their new functions.

**\*TIP** You can step through the pattern edit pages in two groups of four by pressing and releasing the **PAGE** key by itself. Use **FUNC+PAGE** to switch to the first page in the other group.

Let's look at each **pattern edit** page, and see what they all do.

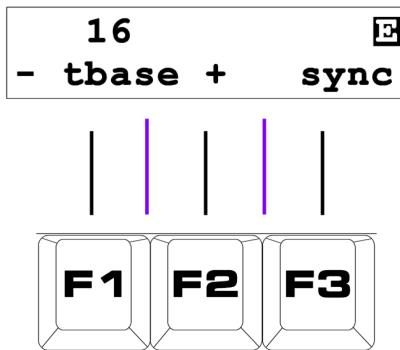
The following table summarises all eight pages.

Note the inverse-video “E” in the top right of the display, showing you are in pattern edit.

<b>16</b> <b>E</b> <b>- tbase + sync</b>	<b>Randomize:</b> <b>E</b> <b>note base range</b>
page 1 – Timebase / Sync	page 5 – Note Randomize
<b>forward</b> <b>E</b> <b>- drctn + last</b>	<b>Randomize:</b> <b>E</b> <b>upper base range</b>
page 2 – Direction / Last	page 6 – Upper Randomize
<b>clr</b> <b>E</b> <b>copy</b> <b>paste</b>	<b>Sculpt:</b> <b>E</b> <b>note upper tap</b>
page 3 – Copy & Paste	page 7 - Sculpt
<b>&lt;- shift -&gt;</b> <b>E</b> <b>Shf</b>	<b>Acc</b> <b>note</b> <b>E</b> <b>Apply</b> <b>conf</b> <b>rng</b> <b>FTS</b>
Page 4 – Rotate / Shuffle	page 8 - Config

## Page 1: Timebase and Sync

The display for page 1 looks like this:



Key	Label	Function
F1	- (dec timebase)	Select previous timebase setting
F2	+ (inc timebase)	Select next timebase setting
F3	sync	Reset pattern to first step immediately

### F1 / F2 - Timebase

The **timebase** setting for each P3 pattern determines the overall length of each step as a subdivision of the tempo. The available timebase values are shown below:

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

The default timebase is **16**.

Each number is essentially the number of steps that will fit in the space of one 4/4 bar at the current tempo.

For example, timebase **16** means each step lasts for one **16<sup>th</sup>** note.

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

Triplets are 2/3rds the length of their non-triplet equivalents, so where a timebase of **16** will fit **16 steps** into a standard bar, **16T** will fit **24 steps** in the same period.

Soft-keys **F1** and **F2** step back and forward through the list of available **timebase** values.

In order to stop patterns getting wildly out of sync as you change timebase, **changes are only applied as the pattern loops** from the last step back to the first.

## F3 - Sync

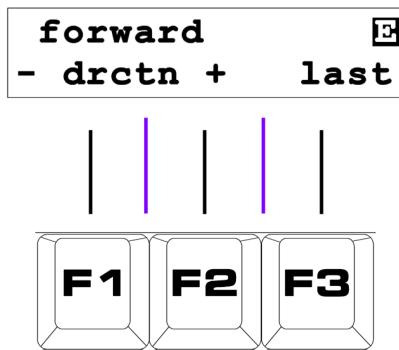
The “**sync**” soft-key instantly resets the pattern being edited to its start position.

This is useful if you get a pattern out of step with the other tracks, perhaps by changing to a triplet timebase or altering the last step.

A **second function** of the “**sync**” soft-key, accessed using **FUNC + F3**, performs a **nudge** of the current pattern - advancing it one extra step forward for each press.

## Page 2: Direction and Last step

The display for page 2 looks like this:



Key	Label	Function
F1	- (dec direction)	Select previous direction setting
F2	+ (inc direction)	Select next direction setting
F3	last	Set the last step in the pattern (pattern length)

### F1 / F2 - Direction

The direction setting for each pattern determines what order the steps will be played in.

There are currently seven directions available, as 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 <b>forward</b> then <b>reverse B</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"> <li>• move to the next step (50% chance)</li> <li>• move to the previous step (25% chance)</li> <li>• repeat the current step (25% chance)</li> </ul>

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.

As with **timebase**, changes of **direction** do not apply until the pattern resets to the first step.

## F3 – Last Step

The “last” soft-key sets the **last step** in a pattern – that is the step after which the pattern will reset to the first step.

Ignoring the effect of **skip**, **last step** determines the **pattern length**.

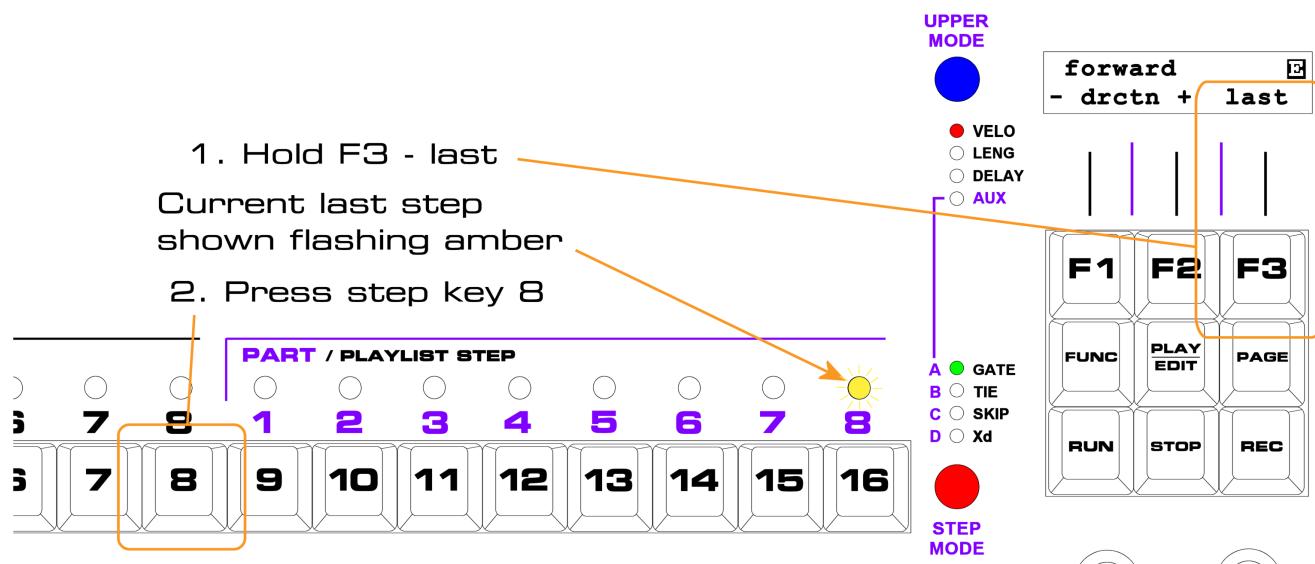
To set the **last step**:

- hold the “last” soft-key, F3

The step LEDs will go out, except for one flashing amber LED indicating the current last step

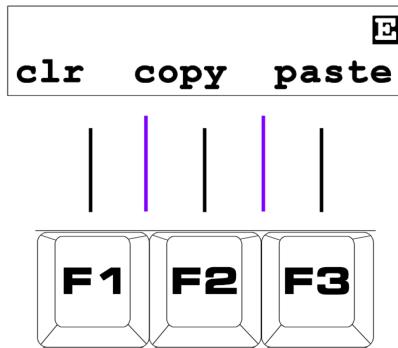
- press the **step key** for the **last step** to be played

For example, if you want to set the last step to **step 8** in a pattern currently set to end on step 16:



## Page 3: Clear, Copy & Paste

The page 3 display shows the following functions



Key	Label	Function
F1	clr	Initialise all pattern steps
F2	copy	Copy steps from the pattern to the clipboard
F3	paste	Paste steps from the clipboard to the pattern

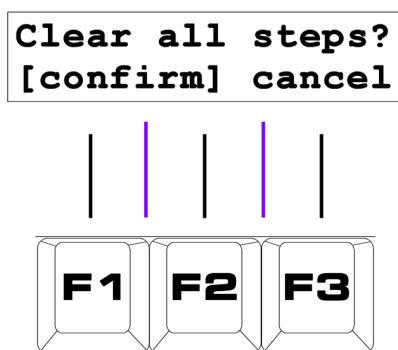
### F1 - Clear

The **clear** soft-key is used to initialise a pattern.

After clearing:

- all steps will have **gate**, **tie** and **skip** turned **off**
- all **note** values will be set according to the **first NOTE knob**
- all **velocity** values will be set according to the **first UPPER knob**
- all **length** values will be set to 5
- all **delay** values will be set to **zero**
- all **aux steps** will be **deactivated**

To use the clear function, press the “clr” soft-key, F1. The display will show:



You must now **press F1 and F2 together** to confirm the operation.

Otherwise, press F3 to cancel.

A **second function** of the “clr” soft-key, accessed by holding FUNC while pressing F1, will **de-activate aux steps only**.

The display will show:

<b>Clear AUX steps?</b>
<b>[confirm] cancel</b>

You must again **press F1 and F2 together** to confirm this operation.

## F2 / F3 – Copy & Paste

**Copy** and **paste** allow you to copy a number of steps in a pattern to another position within the pattern, or to copy some or all of one pattern to another.

### How To Copy

- press the “copy” soft-key, F2, to start the copy process. The display will show:

<b>Select start...</b>
<b>cancel</b>

- select the **first step** to be copied using the **step keys**, or press F3 to cancel

Once the first step is selected, the display will show:

<b>Select end...</b>
<b>cancel</b>

- select the **last step** to be copied using the **step keys**, or press F3 to cancel

When the last step has been selected, the display will briefly show:

<b>Copied nn steps</b>
------------------------

...where “nn” is the number of steps copied to the buffer.

The step LEDs for the steps copied will also briefly turn **flashing amber** before returning to the normal display of step status.

**\*TIP** Note that you can select a **lower numbered** last step to be copied than the first step. This will copy the steps from the pattern to the copy buffer in **reverse order** – so you can reverse small sections of the pattern.

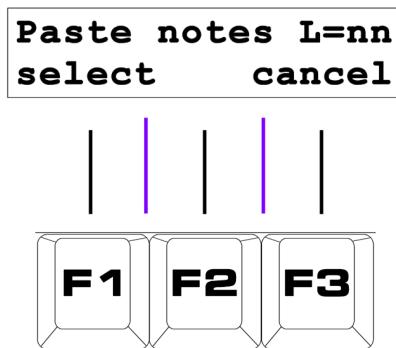
## How To Paste

You can paste the copied data anywhere in the same pattern.

Or you can exit pattern edit, edit a different pattern on the same or different track, and paste the data into the other pattern.

- press the “paste” soft-key, F3, to bring up the **paste page**.

The display will show:



This display tells us two important things.

“**L=nn**” is the **number of steps** that were stored in the copy buffer by the last copy performed.

“**Paste notes**” shows the current **paste filter** setting.

The **paste filter** settings are listed in this table:

Filter	Data Pasted
notes	<b>note, velocity, length, delay, gate, tie, skip</b> and <b>Xd</b> rows
auxes	<b>aux A, B, C and D, numeric and status</b> rows
both	<b>all pattern</b> rows
auxcf	only the <b>aux assignments</b> for the pattern
all	<b>all of the above</b>

- press the “select” soft-key, F1, to select a different **paste filter** if required
- press a **step key** to paste the copied data, starting from that step

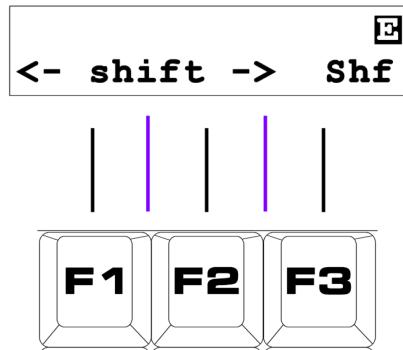
Pasted data will **wrap round** to the start of the pattern if more steps are in the copy buffer than the number from the first paste step to step 16 of the pattern.

Newly pasted steps are shown briefly in **flashing amber**.

If you are pasting only the aux and accumulator config (auxcf), the step number pressed doesn't matter since the settings are global to the whole pattern - but you do need to press a step key to do the paste.

## Page 4: Rotate and Shuffle

The display will show the following functions on this page:



Key	Label	Function
F1	<-	Rotate pattern one step to the left
F2	->	Rotate pattern one step to the right
F3	Shf	Apply classic drum machine "shuffle"

### F1 / F2 - Rotate

The rotate functions move the values for every step in a pattern one step to the left or right.

**F1** rotates to the left, **F2** to the right.

If the **last step** in a pattern is before step 16 step, a modified function can be used – **rotate to last**:

- **hold** the **FUNC** key
- **press** either rotate left, **F1**, or rotate right, **F2**

With **FUNC** held, only the steps in the pattern up to and including the **last step** are moved – the first step wrapping round to the last, or vice versa.

This lets you do interesting things.

For example, you might take a 16 step pattern, set the last step to 8, then use **rotate to last** to move the first 8 steps by 4 steps to the left or right. Then return the last step to step 16.

This would swap steps 1 to 4 with steps 5 to 8, leaving the rest of the pattern intact.

### F3 - Shuffle

The **shuffle** function emulates a feature of classic drum machines where odd numbered steps play exactly on the beat, but even numbered steps are delayed by a certain number of clock ticks.

Shuffle is really just a simple way to set the **delay** value for every step in a pattern:

- **odd steps** have their **delay** set to **zero**
- **even steps** have their **delay** set to a number chosen with the **step keys**

To apply a **shuffle** setting:

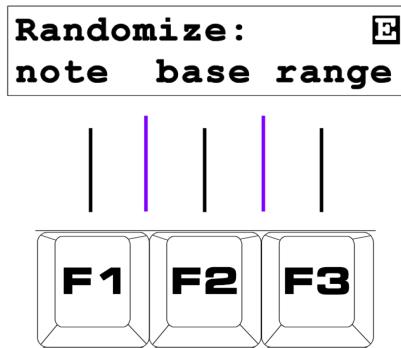
- **hold** the “**Shf**” soft-key, F3
- **press** one of the **step keys** from **1** to **12**

Each step key sets the even step **delay** values to **one less** than the key number.  
Step key **1** will set the **delay** on all steps to **zero**— removing the shuffle effect.

**\*TIP** The Shuffle function is very useful as a quick way of resetting all **delay** values to zero when you've been playing with delay settings manually.

## Page 5: The Note Randomizer

The Note Randomize display looks like this:



Key	Label	Function
F1	note	Randomize the current <b>note</b> value
F2	base	Configure the lowest note that might appear at random
F3	range	Configure the range random notes may cover

### F1 – Note

The **Note Randomize** function lets you enter random **note** values into a pattern as it plays, useful for adding a touch of chaos or just for finding random snippets of melody that might trigger other ideas.

To randomize the note values:

- ensure the sequencer is running
- hold the “**note**” soft-key, **F1**, as the steps you want to randomize play

### F2 – Base Note

The “**base**” option lets you set the **lowest note** that can appear at random in your pattern, as follows:

- press the “**base**” soft-key, **F2**. The display will show:



- use the **DATA** knob to select a new **lowest note**

**THEN**

- press the “**save**” soft-key, **F1**, to store the new selection

**OR**

- press “**lose**”, **F3**, to restore the previous value

## F3 – Range

The “**range**” option lets you set the range of notes above the **lowest note** that can appear randomly in your pattern, as follows:

- press the “range” soft-key, F3. The display will show:

<b>range = nnn</b>
save      lose

- use the **DATA** knob to select a new value from **0** to **127**

**THEN**

- press the “**save**” soft-key, F1, to store the new selection

**OR**

- press “**lose**”, F3, to restore the previous value.

A range of 0 does not impose any limit - the random notes may be anything from the configured **lowest note**, to the highest possible MIDI note.

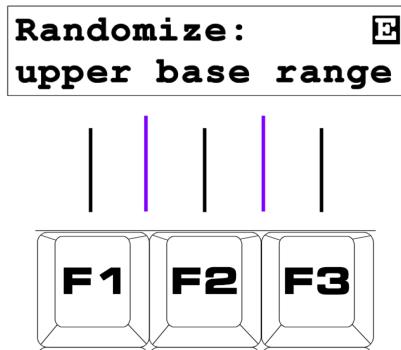
Any other value will give that number of possible different random notes, from the **lowest note** upward.

Note that a range of 1 means only the **lowest note** will be used.

If the **range** is set to be larger than the total number of MIDI notes above the configured **lowest note**, the effective range will be restricted.

## Page 6: The Upper Randomizer

The **Upper Randomize** display looks like this:



Key	Label	Function
F1	upper	Randomize the current <b>upper</b> value
F2	base	Configure the lowest value that might appear at random
F3	range	Configure the range random values may cover

### F1 – Upper

The **Upper Randomize** function lets you enter random values into the row currently selected for editing by **UPPER MODE**.

To randomize the selected upper values:

- ensure the sequencer is running
- **hold** the “**upper**” soft-key, **F1**, as the steps you want to randomize are played

### F2 – Base

The “**base**” option lets you set the **lowest value** that can appear at random in your pattern, as follows:

- **press** the “**base**” soft-key, **F2**. The display will show:

<b>aux</b>	<b>base</b>	=	<b>nnn</b>
<b>save</b>			<b>lose</b>

- use the **DATA** knob to select a new **lowest value**

**THEN**

- **press** the “**save**” soft-key, **F1**, to store the new selection

**OR**

- **press** “**lose**”, **F3**, to restore the previous value.

## F3 – Range

The “**range**” option lets you set the range of values above the **lowest value** that can appear randomly in your pattern, as follows:

- press the “range” soft-key, F3. The display will show:

<b>aux range = nnn</b>
<b>save            lose</b>

- use the **DATA** knob to select a new value from **0** to **127**

**THEN**

- press the “**save**” soft-key, F1, to store the new selection

**OR**

- press “**lose**”, F3, to restore the previous value.

A range of 0 does not impose any limit - the random values may be anything from the configured **lowest value**, to the maximum value of 127.

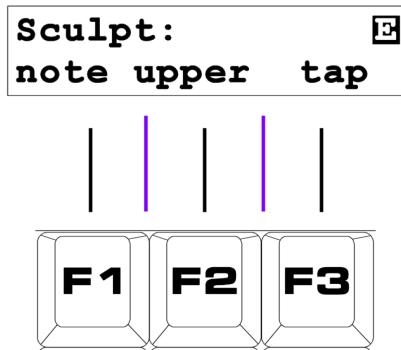
Any other value will give that number of possible different random values, from the **lowest value** upward.

Note that, as with note, range of 1 means only the **lowest value** will be used.

If the **range** is set to be larger than the difference between the **lowest value** and 127, the effective range will be restricted.

## Page 7: Sculpt

This Sculpt page has three functions:



Key	Label	Function
F1	note	Write the current position of the DATA knob to the <b>note</b> value
F2	upper	Write the current position of the DATA knob to the selected <b>UPPER MODE</b> value
F3	tap	Turn on the <b>gate</b> on the current step

### F1 – Sculpt Note

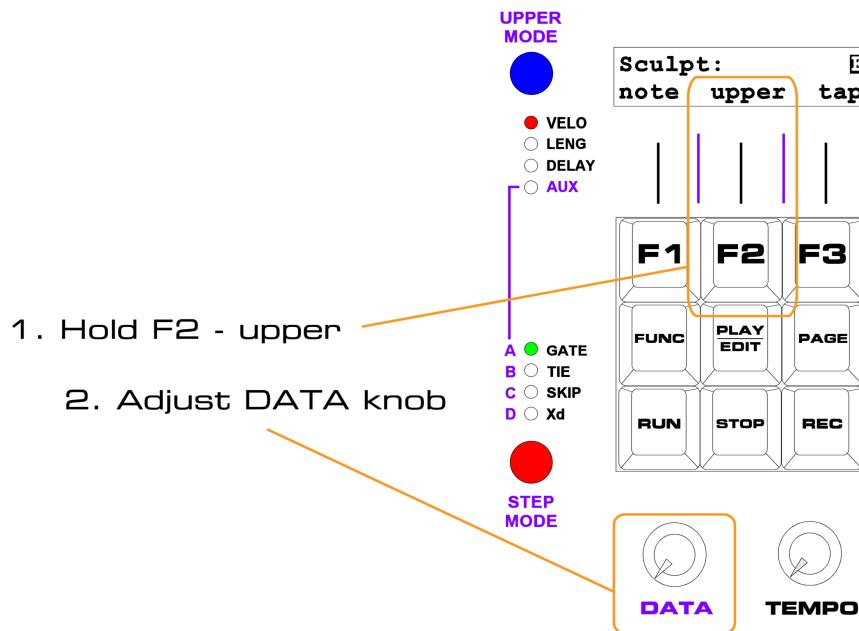
### F2 – Sculpt Upper

The Sculpt functions allow you to record the movement of the **DATA** knob into any of the **numeric values** in a pattern.

For example, if you want the **velocity** in a pattern to rise steadily over the length of the pattern, it could be tedious to go through each step, setting it to a value just a little more than the previous step.

Instead, you can “sculpt” the values as the pattern plays.

In the case of **velocity**, you must ensure UPPER MODE is set to VEL0, then use the “upper” soft-key, F2, as shown:



Actually you don't have to move the **DATA** knob, if it's already in the position you want.

But whatever position the **DATA** knob is in while “upper” is held will be written into the step value at the **start** of each step.

You can leave **DATA** in one place if you want to write the same value to every step.

If you want to see what value will be written before you start sculpting, just move the **DATA** knob without pressing the soft-keys.

The current position will be shown on the display.

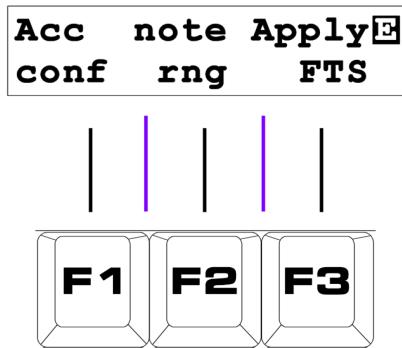
### F3 – Tap

The “**tap**” soft-key, F3, can be used to set **gate** on the current step active as the pattern plays.

This lets you tap in a particular rhythm in real-time, without setting any other values.

## Page 8: Config

This final pattern edit page has three options:



Key	Label	Function
F1	Acc Conf	Configure the <b>Accumulator</b> settings for this pattern
F2	note rng	Configure the range of notes covered by the <b>NOTE</b> knobs
F3	Apply FTS	Apply the current FTS settings to the pattern

### F1 – Accumulator Config

This option is used to configure the accumulator settings for patterns using advanced auxiliary events. These options are described in the later section, **Using Auxiliary Events**.

### F2 – Note Range

This option sets the range over which the **NOTE** edit knobs work.

This setting is held individually for every track – so you can set it up to suit the different instruments you have connected.

Press the **F2** key to set the base note using the **DATA** knob - this will be the note selected when the knobs are turned fully to the left.

Use **FUNC + F2** to edit the span over which the knobs work - there are four ranges of approximately 10, 5, 2 and 1 octaves.

### F3 – Apply FTS

The “**Apply FTS**” function **permanently** applies the current FTS settings to the note values in the pattern.

This is useful if you want to mix different scales across different tracks, so need global FTS turned off for the part.

## Other Pattern Edit Functions

### Quick Config

The Quick Config page has a preset list of combined operations that can be applied to a pattern.

At present, these include options for setting up 1, 2, 3 or 4 of the pattern auxes with the “aux note” events.

This makes it easy to set up a pattern for polyphonic recording.

Refer to the **Using Auxiliary Events** chapter for an explanation of **aux notes**.

To access Quick Config:

- **hold** the **FUNC** key
- **press** the **UPPER MODE** button

From the Quick Config page:

- use the **DATA** knob to select the required operation

**THEN**

- **press** the “**apply**” soft-key, **F1**, to perform the selected operation

**OR**

- **press** the “**cancel**” soft-key, **F3**, to return to pattern edit

### Quick Mute Access

While using pattern edit, if you want to mute or un-mute any of the tracks, you can quickly access the track mutes by holding the **FUNC** key, and pressing **PLAY/EDIT**.

While the **FUNC** key remains held, the step keys **1** to **8** will show the current track mute status, and allow you to toggle them as in play mode.

To remind you which track you are currently editing, that track LED will flash green.

Release the **FUNC** key to return to the normal step key edit function.

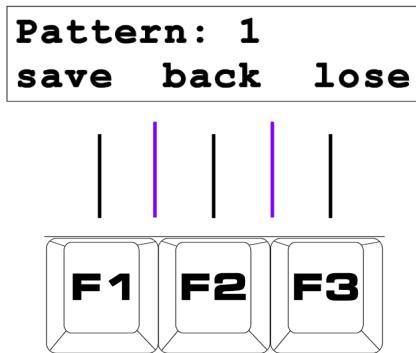
Note that this key combination is not the same as for mute access in playlist edit. Other functions of the keys used in each mode preclude this.

## Leaving Pattern Edit

You already saw how to leave pattern edit in the getting started section, but there were a few details missed out there, so let's revisit it and fill in the blanks.

To exit **pattern edit**, press the **PLAY/EDIT** key once more.

Three soft-key options appear on the bottom line of the display, like this:



Key	Label	Function
F1	save	Store the edited pattern, and return to play mode
F2	back	I didn't mean to press PLAY/EDIT, return to pattern edit
F3	lose	Lose any changes to the edited pattern, and return to play mode

### F1 – Save

If you want to save changes to the pattern you've been editing, press "**save**", **F1**.

### Save To

If you want to save your modified pattern to a **different pattern** location, press one of the **step keys** for the pattern number you want to save it in.

The upper line of the display will show "**Save To:**" and the alternative pattern number selected.

Saving to an alternate location will restore the unedited version of the pattern you were editing to its original location.

## F2 – Back

The “**back**” soft-key, **F2**, returns you to play mode – handy if you pressed **PLAY/EDIT** by accident, or just changed your mind.

## F3 – Lose

The “**lose**” option, **F3**, will restore the pattern as it was **before** you entered pattern edit, and return you to play mode.

## Playlist Edit

In simple terms, **playlists** tell P3 which **patterns** to play on each of the 8 **tracks**.

There is always a **playlist** at work on each **track**, but the **default playlists** are created so that if you don't need to work with **patterns** any longer than the basic 16 steps, then you don't need to know much about **playlists** at all.

It will still be useful to understand what they do.

### *Why Playlists?*

You already saw how to chain a number of **patterns** together across all the tracks, using a **part chain**.

Let's say that you want to chain four patterns together on one track, but only need single patterns on the other tracks.

With a **part chain**, you would need to copy the single patterns to three other locations used by the other **parts** – time consuming and wasteful of space.

Or you may want to transpose the same pattern to a number of different pitches over several bars, but don't want to waste all your parts copying the same pattern multiple times, and part transposing it.

And part transpose will not be any use if you want some patterns to stay at the same pitch while others are transposed.

This is the sort of thing **playlists** are for:

- A **playlist** lets you define a **sequence of patterns** to be played on a **track**.
- Each **pattern** can be used as often as you like in the **playlist**, up to the limit of **8 steps**.
- The **pattern** on each step can be **transposed** over a wide range, and can be **repeated** up to 8 times.
- **Playlists** can also let you synchronise **patterns** to the **global bar** – so you can keep patterns with uneven lengths in step.

## What's Really In A Part?

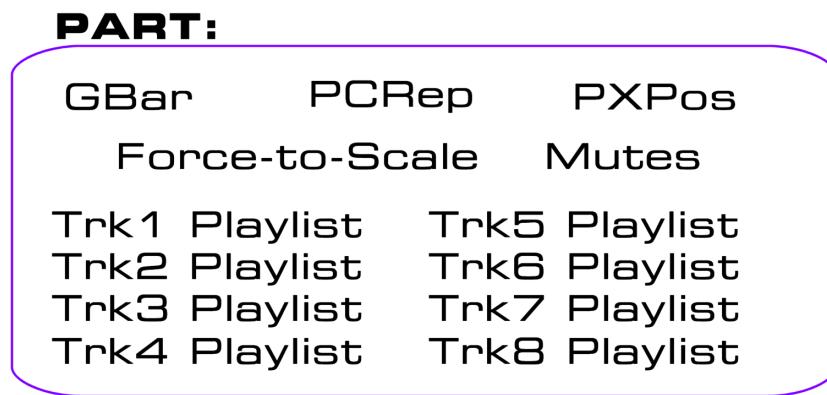
Before we look at the **playlist**, let's take another look at what makes up a **part**.

As you should know by now, there are 8 **parts** in each P3 **bank**.

Each **part** stores some common values – **Force-to-Scale** settings, **Gbar**, **PCReps**, **PXPos**, and the **track mute status** to recall when that **part** is selected.

Each **part** also holds a **playlist for every track**.

This diagram shows what's inside a **part**

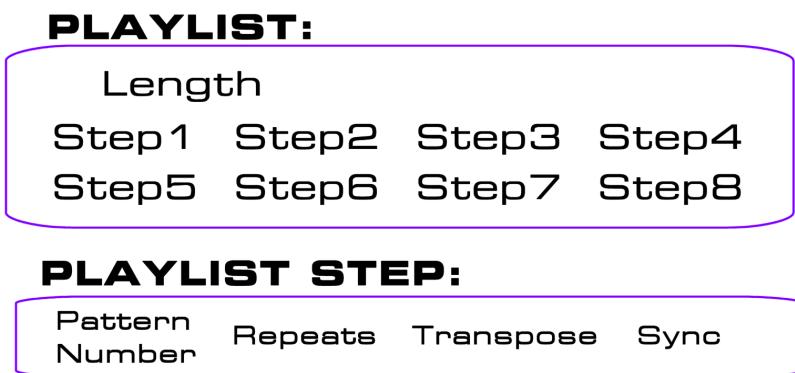


The important point is that **patterns do not belong to a specific part**.

The **playlists** in each **part** can call up any of the available **patterns** on each **track**.

## What's In A Playlist

Time for another diagram:



This diagram shows that a playlist is made up of:

- eight **playlist steps**
- a **length** value

Each **playlist step** holds:

- a **pattern number**
- a **repeats** value
- a **transpose** value
- a **sync** value

So what does this mean?

When you **start P3 running** or select a new **part**, each **track** loads up its own **playlist**, and starts at **playlist step 1**. From there:

- Each **track** plays the **pattern number** in the **current playlist step** for the number of times set in the **repeats** value

**THEN**

- If the **current playlist step** number is less than the **playlist length**, the track moves on to the next **playlist step**

**OR**

- If the **track** has reached the last **playlist step**, it returns to **playlist step 1**

In short, the **playlist** is just a **sequence of patterns**.

## The Default Playlists

When you wipe the entire memory of a P3, or use the **bank init** function to clear a single bank, every part gets a set of **default playlists** that mean you can use P3 without really having to go into **playlist edit** at all.

Let's look at the **default playlist** in each **part** of an **initialised P3 bank**:

- The **playlist length** is **1**
- **Playlist step 1** has:
  - a **pattern number** equal to the **part number**
  - a **repeats** value of **1**
  - a **transpose** value of **0**
  - a **sync** value of **Ptrn**

So, let's say you select **part 4** in an initialised bank:

- The **playlist** on **every track** has a **length of 1 step**, therefore **playlist step 1** will repeat for as long as that **part** is selected
- **Playlist step 1** on every track says, "play **pattern number 4**"

Transpose and sync default values have no effect here, so you can ignore them for now.

In an **initialised bank**:

- if you select **part 1**, then **pattern 1** plays on **every track**
- if you select **part 2**, then **pattern 2** plays on **every track**
- if you select **part 3**, then **pattern 3** plays on **every track**

...and so on.

To a certain extent, the **default playlists** mean you can treat **parts** as if they were just **multi-track patterns**.

This idea only breaks down if you use the **part-copy** functions, but more of that later.

### Important Note On Memory Organisation:

In this chapter it is assumed that you are using P3 in the standard 6 bank, or alternative 12 bank memory configurations. Since there are only 4 patterns per track in 12 bank mode, the default playlists repeat patterns 1 to 4 in the playlists for parts 5 to 8.

For the 3 bank mode, with 16 patterns per track, **playlist edit** behaves in a slightly different manner for pattern selection. These differences are explained in the Appendix – Memory Organisation.

## *Editing A Playlist*

In the rest of this chapter, we'll take a **default playlist**, see how it looks in **playlist edit**, and then gradually extend it.

So that you can follow the example, you'll need to use a **bank** with **default playlists**.

If you just got a new P3, it should default to **bank 1** at power-on, and the bank will be ready to use in this example.

If you have already experimented with playlists before you knew what you were doing, you may want to use the **bank init** function - just to be sure you are starting from the right place.

**Playlists** control which **patterns** play, but if there is nothing in the **patterns**, there won't be anything to hear!

So let's make sure you have a couple of patterns to use with your soon-to-be-created **playlist**.

Follow these steps to create **two patterns** on **track 1**.

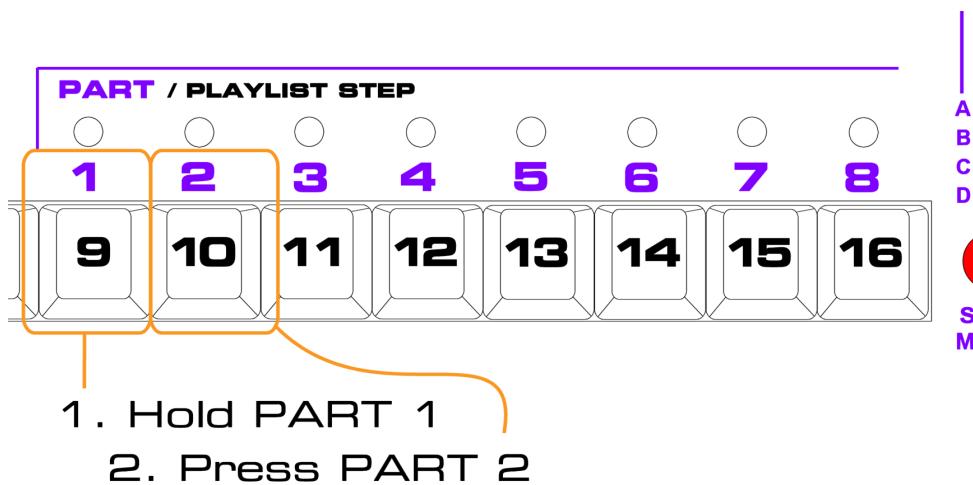
Make sure the patterns have the standard length of 16 steps – we'll be using odd lengths later to look at the **sync** option.

You should be starting from **part 1** in **play mode**

- Enter **pattern edit** on **track 1**
- Create a simple **pattern**, save it – this will be **pattern 1**, then return to **play mode**
- Switch to **part 2**
- Enter **pattern edit** again on **track 1**
- Create another **pattern**, save it – this will be **pattern 2**, then return to **play mode**

You can already play your two new patterns in sequence by using a **part chain** – so let's check that out now.

Set a **part chain** to play **part 1** then **part 2** in turn, like this:



You should hear your first pattern play through once, then your second pattern plays, and this repeats indefinitely.

If not, start again from the top of this section.

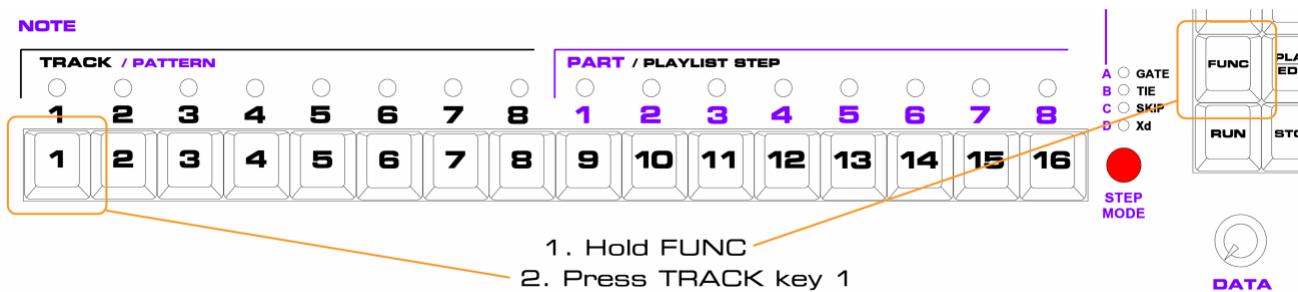
If you're happy that you've got two patterns to play with, **select part 1 by itself**, leave P3 running, and let's get on with **playlist edit**.

## Changing To Playlist Edit

The **play mode** chapter already showed how to get into **playlist edit**, but here's a reminder. From **play mode**:

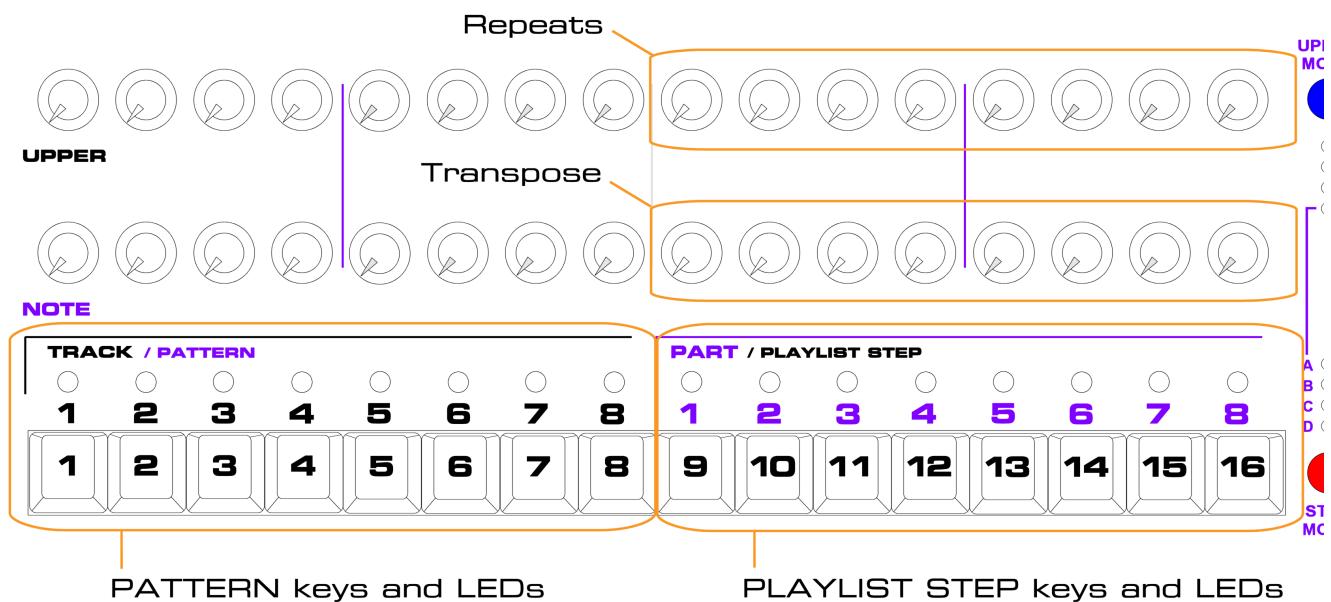
- hold the **FUNC** key
- press the **TRACK** key for the **track** on which you want to edit the playlist

You want to edit the **playlist** on **track 1**:



## Playlist Edit Controls

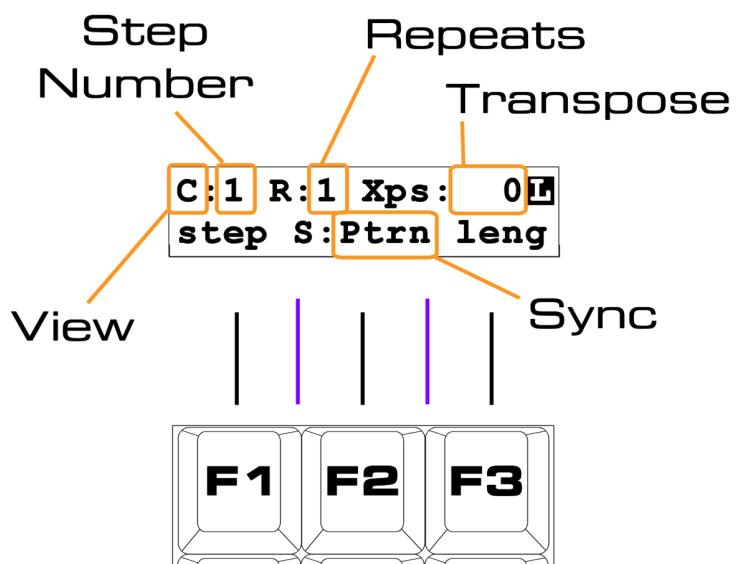
In **playlist edit**, the front panel controls take on the functions shown here:



## Playlist Edit Display

The **playlist edit** display will look like the diagram below.

The labels indicate where the **playlist step** values are shown:



Just ignore the “View” label for now.

The **Step Number** value shows the **current playlist step** – that is, the step playing at any given time.

For the **current playlist step** the **Repeats**, **Transpose** and **Sync** values are shown where indicated.

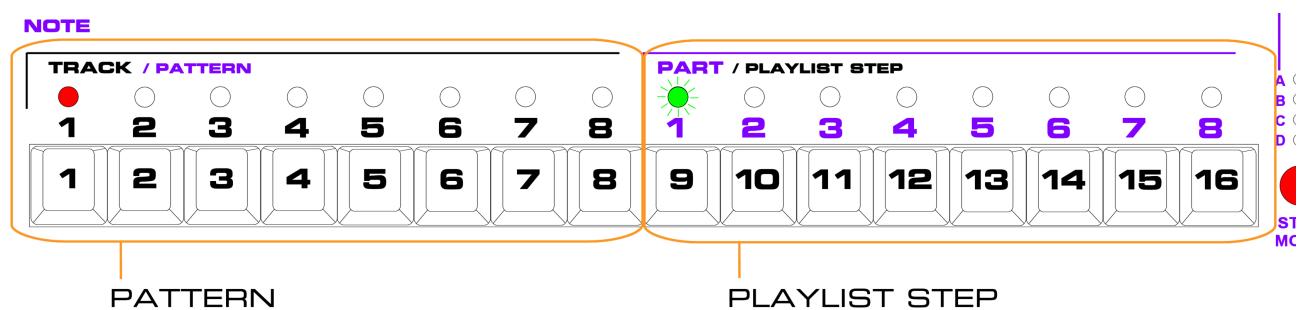
You are looking at the **default playlist** in **part 1**, so:

- **step number** is 1
- **repeats** is 1
- **transpose** is 0
- **sync** is Ptrn

## Playlist Edit LEDs

The step keys and LEDs in **playlist edit** are split into two sets of eight – the **PATTERN** keys and the **PLAYLIST STEP** keys.

Right now, the LEDs will look like this:



In **playlist edit**, the **pattern number** for the current step is shown by a **red** LED above the **PATTERN** keys.

The **red** LED here shows **pattern 1** is playing on this step.

**One or more** of the LEDs above the **PLAYLIST STEP** keys may be **green**, and one of these green LEDs will **flash**:

- The **number of green** LEDs shows the **length** of the **playlist**
- The **flashing green** LED shows the **current playlist step**

The **flashing green** LED here shows you're currently on **playlist step 1**.

The lack of any other green LEDs shows that **step 1** is the **only step** used in this playlist – the **length** is **1**.

## Selecting A Different Pattern

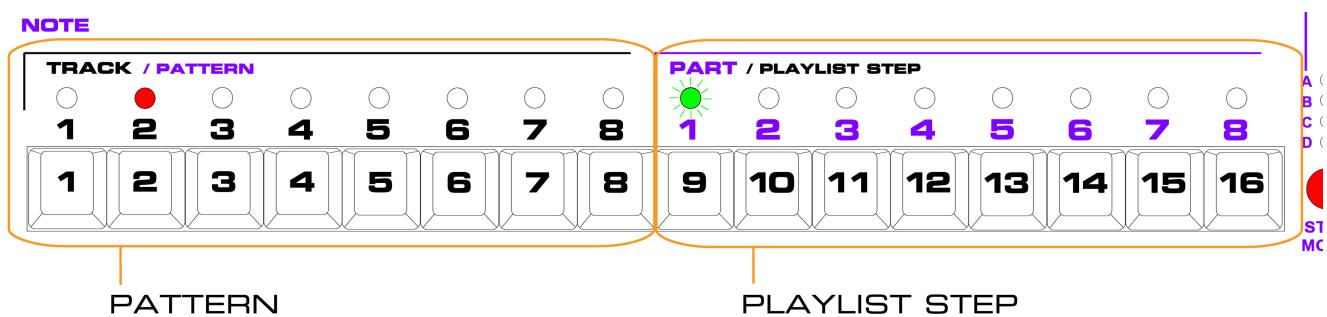
If you followed everything correctly so far, you are in **playlist edit**, looking at the **playlist** for **track 1**, in **part 1**.

This **playlist** has a **length** of **1 step**, and plays only **pattern 1**.

Let's change the **pattern** on this step to **pattern 2**:

- press **PATTERN** key 2

The LEDs should now look like this:



The **red LED** above **PATTERN** key 2 shows that **pattern 2** is now selected for this step.

Note that P3 won't switch to the **pattern 2** until **pattern 1** has finished its current pass. Then you will hear **pattern 2** start playing.

You have edited your first **playlist**.

To change **playlist step 1** back to play **pattern 1**:

- press **PATTERN** key 1

The **red LED** will move from **PATTERN** key 2 to **PATTERN** key 1.

As before, **pattern 2** will continue playing until its last step, at which point the track will switch back to **pattern 1**.

## A Longer Playlist

Now you can change the **pattern** on a single **playlist step**, let's try something a little more adventurous.

Let's chain patterns 1 and 2 together so they play in turn on this track.

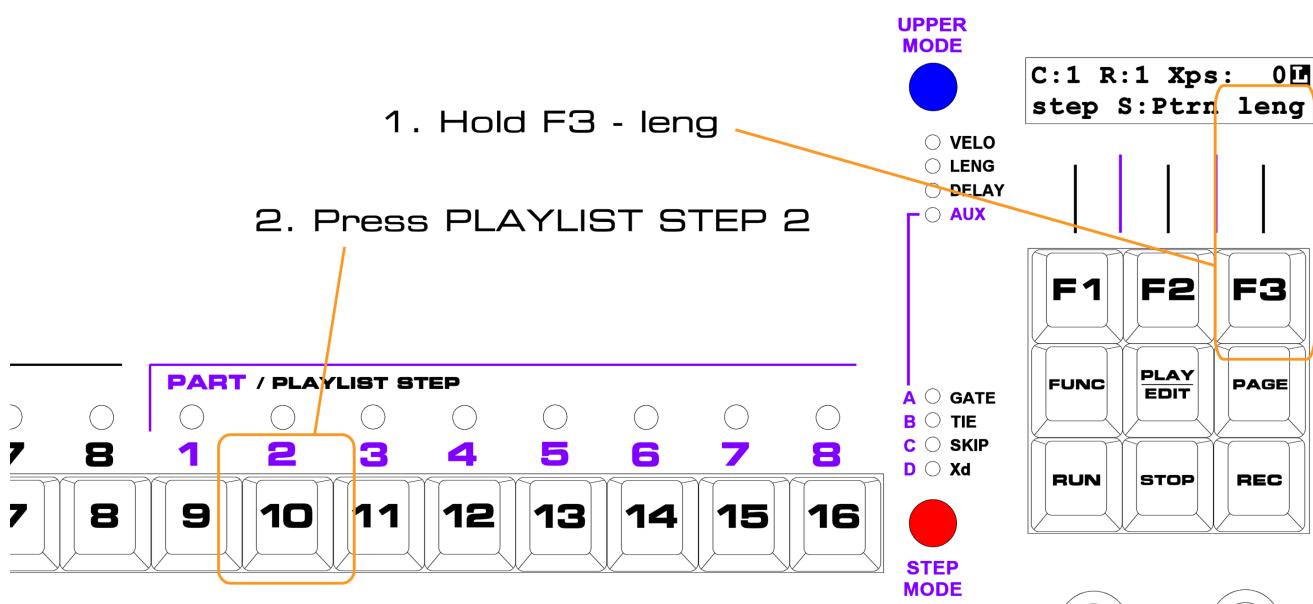
To do this, you need to:

- set the playlist **length** to 2 steps
- set **playlist step 1** to play **pattern 1**
- set **playlist step 2** to play **pattern 2**

To **set the length** of a playlist:

- **hold** the "leng" soft-key, F3
- **press** the **PLAYLIST STEP** key for the **last step** in the playlist

In our example, you'll use **PLAYLIST STEP** key 2 to set the **length** to 2, like this:



As soon as you set the **length** to 2, the **PLAYLIST STEP 2** LED will light up **green**.

**Remember** – the **number of green LEDs** above the **PLAYLIST STEPS** indicates the **length** of the playlist

You will also see the green LED above **PLAYLIST STEP 1** continue to flash until the end of **pattern 1**, which is playing on that step.

As soon as pattern 1 ends, **PLAYLIST STEP 2** will start **flashing green** instead.

With a **length** of 2, playlist steps 1 and 2 play in turn.

You should hear your two patterns play in turn, just as they did in the **part chain**.

As the playlist advances, the “current step” in the display will change from 1 to 2, like this:



You will also see the red PATTERN LED switches from **pattern 1** while **playlist step 1** plays, to **pattern 2** while **playlist step 2** plays.

How did it know to play **pattern 2**?

Well, remember you started out using **part 1** on **track 1** for the previous playlist edit example.

You edited the **playlist** to find **pattern 1** was on **playlist step 1** – the default playlist for a “blank” **part 1**.

The reason you have found **pattern 2** on **playlist step 2** is because a **default playlist**, even though its length is only set to 1 step, still has a pattern number on all the unused steps.

Playlist step 1 has the same pattern number as the part number.

Each following playlist step has the next pattern number up.

So an **initialised playlist** in **part 1** looks like this:

- Length = 1
- Playlist Step 1 = pattern 1
- Playlist Step 2 = pattern 2
- Playlist Step 3 = pattern 3

...and so on.

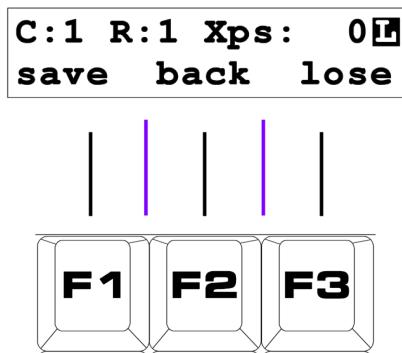
You have changed the playlist **length** to 2, and you can see your **playlist** is playing 2 steps, with a **different pattern** on each step.

## Leaving Playlist Edit

Having hopefully proved how easy it is to edit a **playlist**, let's look quickly at how you **save the changes** and return to **play mode**.

To exit **playlist edit**, press the **FUNC** key once more.

Just as with **pattern edit**, there is a confirmation page giving you the following options on the soft-keys:



Key	Label	Function
F1	save	Store the edited playlist, and return to play mode
F2	back	I didn't mean to press <b>FUNC</b> , return to playlist edit
F3	lose	Lose any changes to the playlist, and return to play mode

So, press “**save**”, **F1**, and your **playlist edit** is done.

You have edited the **playlist** on **track 1, part 1**, and set the **length** to **2**.

Step 1 is playing **pattern 1**.

Step 2 is playing **pattern 2**.

Back in **play mode**, you can hear that your **track** is playing **two patterns in a chain**, without the need for a **part chain**.

## Some More Editing

Let's continue with our **playlist** example, by returning to **playlist edit** for the same **track**.

Hold **FUNC** and press **TRACK** key 1.

With P3 still running, you should see the **PATTERN** LEDs switching from **pattern 1** to **pattern 2**, and back.

The green **PLAYLIST STEP** LEDs show the **length** is 2, and the flashing step will show, along with the moving red **PATTERN** LED, that **pattern 1** is on **step 1** and **pattern 2** is on **step 2**.

Now that the playlist length is 2, the step you are looking at changes as the playlist advances.

So if we want to change the pattern number in a playlist longer than 1 step, you may have a problem.

You change the pattern on a step by pressing the **PATTERN** key for the pattern number you want.

But what if the playlist advances just as you reach for the **PATTERN** key?

You could end up selecting a different pattern for the following step.

## Playlist Views

To solve this problem, **playlist edit** has two different ways to **view** the **playlist** you are editing.

The **view** is indicated by the letter in the top left of the display:

- C stands for **Chase**
- E stands for **Edit**

### Chase View

The default is **Chase view**.

In **Chase view**:

- the display shows the values of **step number**, **repeats**, **transpose** and **sync** for the **current playlist step**
- the **PATTERN** LEDs show the **pattern number** on the **current playlist step**
- selecting a new pattern using the **PATTERN** keys, selects a new pattern for the **current playlist step**

As the **playlist** advances, so does the step you are looking at.

This makes **Chase view** better suited to looking at what's in your playlist rather than trying to edit it – unless you're quick.

As a visual aid to show that you are in **Chase view**, aside from the "C" in the display, one each of the **UPPER MODE** and **STEP MODE** status LEDs light up and "chase" downwards by one step on each beat.

## Edit View

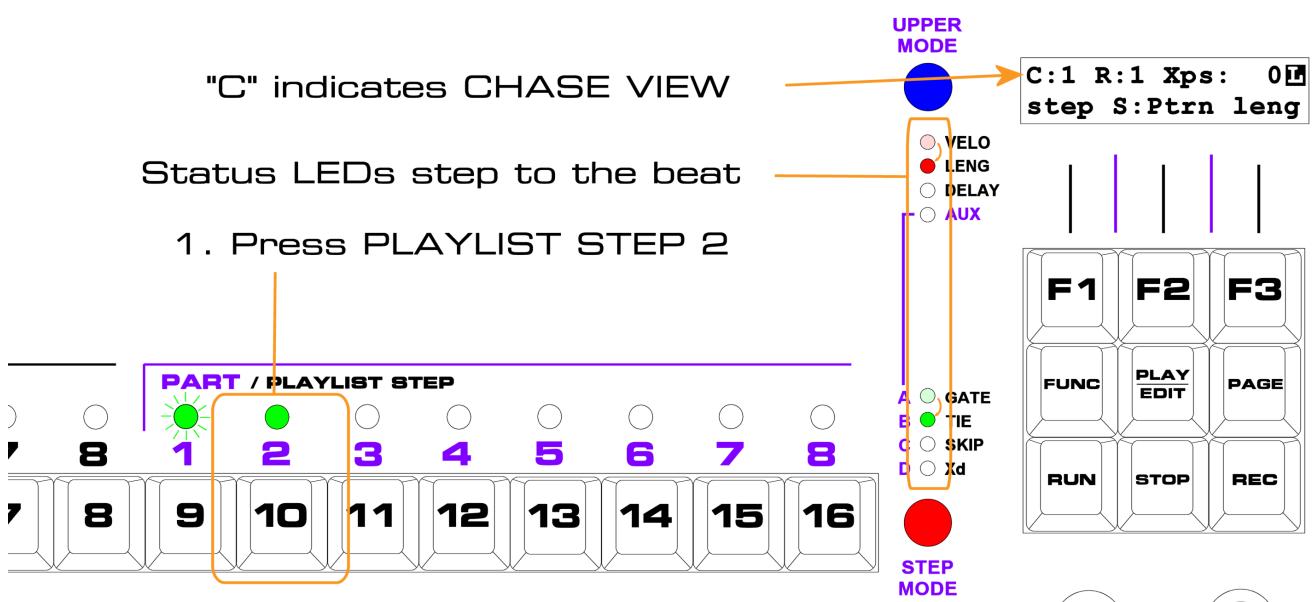
In **Edit view** you can **select any playlist step directly** in order to edit it, or just to view the values of **repeats**, **transpose** and **sync** without having to wait for that step to come round.

The step you chose to edit will **remain selected**, even though the **playlist continues to play as normal**.

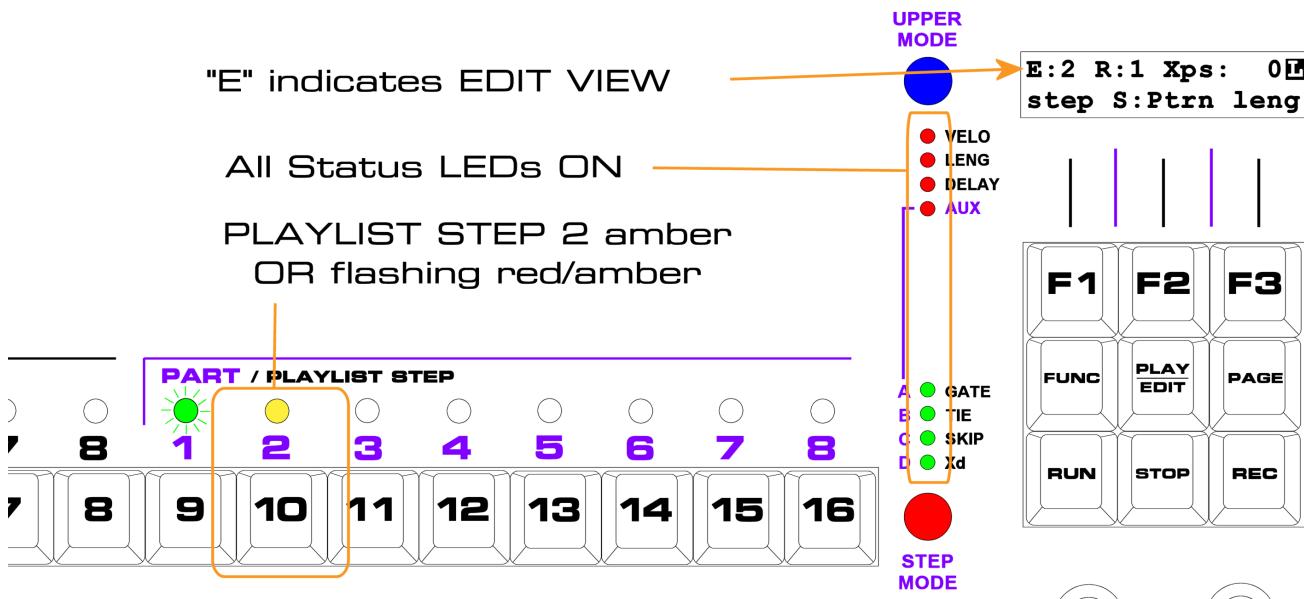
To select **Edit view** all you need to do is press the **PLAYLIST STEP** key for the step you want to edit.

Let's continue with our example by using **Edit view** to look at step 2 in the **playlist**.

Select **Edit view** for **playlist step 2** as follows:



As soon as you press **PLAYLIST STEP key 2**, the display switches to **Edit view**, which will look like this:



The things that show you are now in **Edit view** are:

- “E” in the top left of the display
- **UPPER / STEP** status LEDs **all on**
- the **step number** after “E:” will be your chosen step, 2
- the LED above the **PLAYLIST STEP** key 2 will be **amber OR flashing red/amber**

The display shows “E:2” to confirm that you are **Editing step 2**.

The colour of the **PLAYLIST STEP** LED also shows this, but probably needs a little more explanation.

### LED Colours Revisited

When you select a playlist step for **Edit view**, the **red** element of the **PLAYLIST STEP** LED will turn on.

Since the **playlist** in our example has a **length** of **2 steps**, the **green** element of the LED is **already on**.

As you saw when we looked at the **status rows** in **pattern edit**

- **red + green = amber**

That's why the LED will be **amber**.

In **Edit view**, the **current playlist step** continues to **flash** just as in **Chase view**.

So when **step 2** becomes the **current playlist step**, the LED will **flash green**.

The **red** element **stays on** because this is the **step being edited**.

The result is that **when the edited step** is the **current playlist step**, the LED will **flash between red and amber**.

At this point, try selecting some other **playlist steps** for editing.

If you select a step from **3** to **8**, you will see that the LED is **solid red**.

The playlist **length** is still only **2**, so none of the LEDs from **3** to **8** have their green element on. And they never become the **current playlist step**.

To summarise the various colours of **PLAYLIST STEP** LEDs:

LED Colour	Meaning
<b>green</b>	this step is within the current playlist <b>length</b>
<b>flashing green</b>	this step is the <b>current playlist step</b> being played
<b>amber</b>	this step is within the current playlist <b>length</b> , and is <b>selected for editing</b> in Edit view
<b>flashing red / amber</b>	this step is the <b>current playlist step</b> being played, and is <b>selected for editing</b> in Edit view
<b>red</b>	this step is NOT in the current playlist length, but has been <b>selected for editing</b>

**\*TIP** Being able to select steps for editing even though they are not currently part of the playlist, means you can set up the patterns on these steps exactly as you want **before** you increase the playlist length to include them.

## Back To The Playlist...

Now let's use **edit view** to look at the rest of the settings on a **playlist step**.

We'll use **playlist step 2**, so if you have selected another step while playing with **edit view**, you'll need to switch back to **step 2** now.

**\*TIP** There's an alternative method for selecting a **playlist step** in **edit view** – if you move the **DATA** knob in **playlist edit**, P3 will switch to **edit view** for the step at the current knob position, or scroll to other steps if already in **edit view**. The 8 steps are spread evenly around the **DATA** knob.

## Step Repeats

Any step in a **playlist** can be made to repeat the **pattern** it plays up to 8 times, using the **repeats** value.

With P3 still running, your **playlist** should be playing **pattern 1** then **pattern 2** in turn, once through each pattern.

The **UPPER** knobs above each **PLAYLIST STEP** key are used to adjust the **repeats** value on each step.

Turn the **repeats** knob above **PLAYLIST STEP 2** now, until the top line of the display looks like this:

<b>Step 2 Reps 4</b>
<b>step S:Ptrn leng</b>

You have set the **repeats** value on **playlist step 2** to a value of **4**.

The display will return to normal after a second.

Then you'll see the change of value confirmed on the **Edit view** display for **step 2**:

<b>E:2 R:4 Xps: 0</b>
<b>step S:Ptrn leng</b>

Listen to your patterns playing and watch the **flashing green** LED for the **current playlist step**. You should see that **pattern 2**, on **playlist step 2**, **repeats 4 times**.

Note that although you are in **edit view**, you can adjust the **repeats** value on any **playlist step** with the **UPPER** knob above that step.

Try this out by using the knob above **PLAYLIST STEP 1** to set **repeats** to **3**, and then use the knob above **PLAYLIST STEP 2** to set **repeats** back to **1**.

You should hear **pattern 1** play **3 times** then **pattern 2** plays **once**, before the playlist repeats itself.

**Repeats** can be used to build a playlist that lasts over many bars.

The maximum value of **repeats** is 8, so with 8 steps in a playlist, the total time a playlist takes to run through once can be up to **64 bars**.

## Step Transpose

You can **transpose** the **pattern** played on any **playlist step** over a wide range.

The **same pattern** can be selected on a number of **playlist steps**, and **transposed by different amounts**.

Let's work through an example of that now.

At the moment you should have a playlist with a length of 2 steps, pattern 1 on step 1, and pattern 2 on step 2.

Follow these steps:

- use the **repeats** knob above **PLAYLIST STEP 1** to set **repeats** back to **1** on **step 1**
- use **F1 + PLAYLIST STEP key 2** to select **edit view** for **step 2**
- press **PATTERN** key **1** to select **pattern 1** on **step 2**
- use **F1 + PLAYLIST STEP key 3** to select **edit view** for **step 3** – note that this step is not yet playing since the length is only 2
- press **PATTERN** key **1** to select **pattern 1** on **step 3**
- use **F1 + PLAYLIST STEP key 4** to select **edit view** for **step 4**
- press **PATTERN** key **2** to select **pattern 2** on **step 4**
- use **F3 + PLAYLIST STEP key 4** to set the **playlist length** to **4**

You should now have a playlist that plays the same series of patterns as before.

But instead of having a **2 step playlist** like this:

<b>Step</b>	<b>1</b>	<b>2</b>
<b>Pattern</b>	<b>1</b>	<b>2</b>
<b>Repeats</b>	<b>3</b>	<b>1</b>

...you should have a **4 step playlist**, like this:

Step	1	2	3	4
Pattern	1	1	1	2
Repeats	1	1	1	1

To confirm that this is what you have, go through each step in **edit view** and check the values.

Or you might want to switch back to **chase view**, so you can watch the values change as the playlist advances.

If you haven't already found how to do this by accident, to switch back to **chase view**:

- press and release the "step" soft-key, F1, by itself

If the **playlist** is as it should be, let's try **transposing** pattern 1 on a couple of steps.

In **playlist edit**, the **NOTE** knobs above each **PLAYLIST STEP** are used to set the **transpose** values.

Turn the **transpose** knob above **PLAYLIST STEP 1** now, until the top line of the display looks like this:

**Step 1 Xpos +12**  
**step S:Ptrn leng**

Note that the **zero** position is in the **centre** of the travel for the transpose knobs.

You have set the **transpose** value on **playlist step 1** to a value of **12**.

Listen to the pattern playback, and you should hear **pattern 1** now plays an octave higher on **playlist step 1**.

When the **playlist** advances to step 2, **pattern 1** plays again, but this time at its normal pitch.

Now set the **transpose** on **step 2** to a value of **+7**, just as before.

You should hear **pattern 1** running through three different pitches before **pattern 2** comes along.

## Step Sync

The **sync** value on each **playlist step** takes one of two values.

The default is **Ptrn**, which is short for “**pattern**”.

The other option is **GBar**, which is short for “**global bar**”.

You should remember the **global bar** setting from the discussion of **parts**.

To re-cap, each **part** has a **global bar length**, which is used to determine how long one bar lasts in 16<sup>th</sup> notes, since the patterns may have different lengths and timebases.

Changes to a new **part** are synchronised to the end of the **global bar**.

The **sync** value on each **playlist step** allows you to synchronise individual **patterns** to the **global bar** too.

Let's look at an example of **global bar sync** in action, to see what this means.

If you've been keeping up with the example so far, you should have a playlist that looks like this:

Step	1	2	3	4
Pattern	1	1	1	2
Repeats	1	1	1	1
Transpose	+12	+7	0	0

It may be a little different, but as long as you have the right patterns on the 4 steps, it'll be fine.

Since you are working with a default **part**, the **global bar length** will be **16**, which should also be the same length as your **patterns**.

So you can experiment with the **sync** setting, you'll need to change the length of **pattern 1**.

## Accessing Pattern Edit from Playlist Edit

At this point you may be thinking you'll need to exit **playlist edit** so you can edit pattern 1 from **play mode**.

But you can access **pattern edit** directly from **playlist edit**.

You may remember that from **play mode**, you can only edit the currently playing pattern on a track, using a combination of **PLAY/EDIT** and the **TRACK** key.

From **playlist edit**, you choose the **pattern** to edit **directly** by **pattern number**.

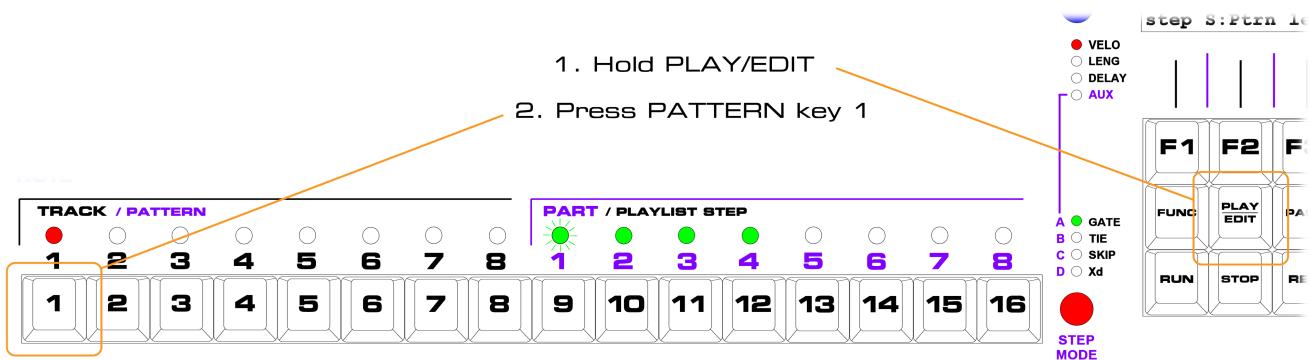
This means you can be sure which pattern you're going to edit, even if a playlist is stepping through different patterns.

From **play mode**, you always have to enter **pattern edit** while the **pattern** you want is playing. It also means you can edit a **pattern** that isn't currently playing.

To enter **pattern edit** from **playlist edit**

- hold the **PLAY/EDIT** key
- press the **PATTERN** key for the **pattern number** you wish to edit

In this case you want to edit **pattern 1**, like this:



In **pattern edit**, set the **last step** of the pattern to **step 6**.

If you don't know how to do that, revisit the chapter on **pattern edit**.

Save the pattern, and as you leave **pattern edit**, you will return to **playlist edit**.

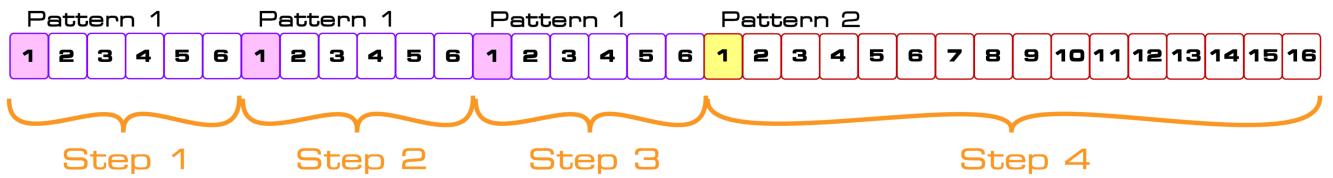
**\*TIP** When you are in **pattern edit** for a pattern that is **not currently playing** the "blip" that normally runs across the step LEDs is replaced by a "blanking blip" – each step LED goes out briefly on the current position in the pattern that is actually playing.

When you see the normal blip re-appear, you know the pattern you are editing is live.

## Step Sync, continued...

Back in **playlist edit**, and with **pattern 1** only playing for **6 steps**, your **playlist** should sound a little different now.

This diagram should help visualise what's going on:



**Pattern 1** plays for **6 steps** on each of **playlist steps 1, 2 and 3**.

**Pattern 2** plays for its full **16 steps** length on **playlist step 4**.

This is what happens if the **default** setting of **sync – pattern**, is used on every **playlist step**.

Now select **playlist step 1** in **edit view**.

The **sync** value is shown above soft-key **F2**, and this key is used to toggle the value.

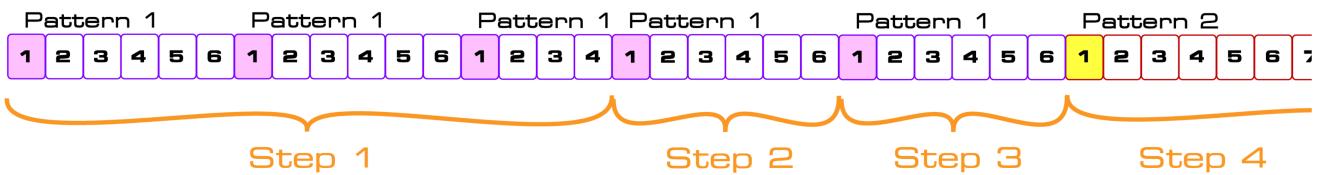
Press **F2** now, and the display should change to this:

```
E:1 R:1 Xps:+12█
step S:GBar leng
```

The **sync** value on **step 1** is now set to **GBar**, for **global bar**.

This means that **playlist step 1** will play for the **full length of one global bar**, regardless of the length of the **pattern**.

The result is this:



**Playlist step 1** will play **pattern 1** twice for its full length of **6 steps**, then again for its first **4 steps**, before the playlist moves on to **step 2**.

The **total number of steps** that **pattern 1** plays for is **16 steps** – the **global bar length**.

As you can see, **playlist steps 2 and 3**, which still have a **sync** value of **Ptn**, play as before for the **6 step** length of **pattern 1**.

But on **playlist step 1**, **pattern 1** is effectively stretched back to **16 steps** long.

This would still apply even if the **playlist** were only 1 step long.

Set the length back to 1 now, and listen to the pattern.

**Pattern 1** repeats in a regular **6 step, 6 step, 4 step** pattern.

The **start** of the **pattern** is synchronised with the **start** of the **global bar**.

There are a number of uses for this feature.

You can use **GBar sync** to automatically repeat short **patterns** up to the **global bar length** without having to work out how many times you would need to repeat it to fit.

If you have a **pattern** with an unusual length, you can force it to fit into a predictable length that matches, or divides equally into, the other patterns.

A particularly interesting use of **GBar sync** is where you are editing a pattern, and **skipping** some pattern steps.

With the pattern locked to the global bar length, it keeps in time with the other patterns, but the rhythm can change dramatically as you add and remove skip steps.

You could try this out now with **pattern 1** – make sure the playlist length is set to 1, then go into pattern edit.

Experiment with different last steps, and skipping.

## **GBar Sync and Repeats**

If you use the **GBar sync** setting on a **playlist step** in conjunction with **repeats**, you'll find that the total length of the playlist step is equal to the global bar length multiplied by the repeat value.

The **pattern** itself will play for its own length until the **total length** of the **playlist step** has passed – it will not be reset on each **global bar** start.

This is useful for setting up a pattern with an odd length to shift against the other patterns over time, yet still be brought back into line after a regular number of bars have elapsed.

## Other Features Of Playlist Edit

### Short-cut Playlist Creation

You may often find that you want to create a playlist that simply chains 2 or more patterns together, with no repeats and no transposition.

If you are working from a default playlist, you can often do this just by setting the length required, but if you need to use a different starting pattern number, or you already changed some of the other playlist steps, there is a short-cut method to setting up a simple playlist with sequential patterns:

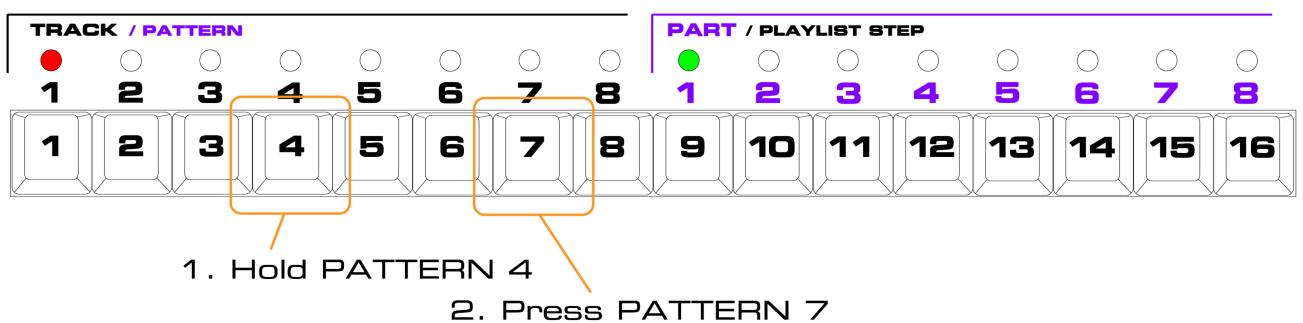
- hold the **pattern** key for the pattern you want the playlist to **start** on
- press the **pattern** key for the pattern you want to **end** on

When you do this, the playlist will be configured with the set of patterns you selected, in order, with repeats set to 1, transpose set to 0, and sync set to "Ptn".

For example, let's say you want to create a playlist with a **length** of **4** steps, like this:

Step	1	2	3	4
Pattern	4	5	6	7
Repeats	1	1	1	1
Transpose	0	0	0	0
Sync	Ptn	Ptn	Ptn	Ptn

You can do this in one step, like this:



As soon as you release the pattern keys, you will see the green PLAYLIST STEP LEDs indicate the length has been set to 4.

Step through the playlist using **edit view**, and you'll confirm that patterns are where you wanted them.

## GBar Pos and Playlist Hold

In common with **play mode**, holding the **UPPER MODE** button while in **playlist edit** gives you access to the **playlist hold** flags for each track, and the **GBar** countdown.

These flags allow you to suspend the progress of the playlist on any track.

While **UPPER MODE** is held, the track LEDs will normally indicate green for each track.

By pressing a **track key**, the advance of the playlist on that track is suspended, and the LED shows red. Another press of the **track key** will return the track to normal.

## Quick Mute Access

While using **playlist edit**, if you need to mute or un-mute any of the tracks, you can quickly access the track mutes by holding the **STEP MODE** key.

While it is held, step keys 1 to 8 will show the current track mute status, and allow you to toggle them as in **play mode**.

Release the **STEP MODE** key to return to the normal **playlist edit** functions.

## Real-time Record

P3 can record notes into any pattern from an attached MIDI keyboard or other controller. Record is available in **play mode**, **playlist edit** and **pattern edit**.

Recording is restricted to one note per pattern step, so you need to ensure the timebase on the patterns you record into will be high enough to capture what you want to play.

**\*TIP** If you have assigned one or more of the auxes in a pattern to **aux note** events, record mode will record into your pattern **polyphonically**, with the lowest held note being written to the main note value in the pattern and any other notes being written into available auxes.

### *Differences Between Play Mode, Playlist Edit and Pattern Edit*

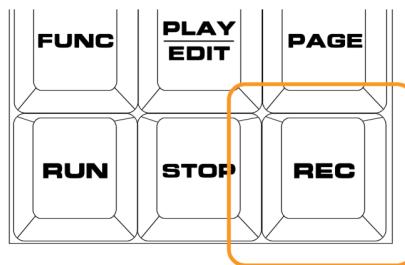
#### Pattern Edit

In **pattern edit**, notes will be recorded **only** into the pattern **currently being edited**.

If there is a playlist on your track that is cycling through a number of patterns, input notes will record only on the edited pattern.

They will record onto the pattern at the current step location, even if another pattern is selected.

To activate **record** in pattern edit, simply press the **REC** key:



#### Play Mode & Playlist Edit

In **play mode** and **playlist edit**, notes will be recorded into any pattern currently playing on the track selected.

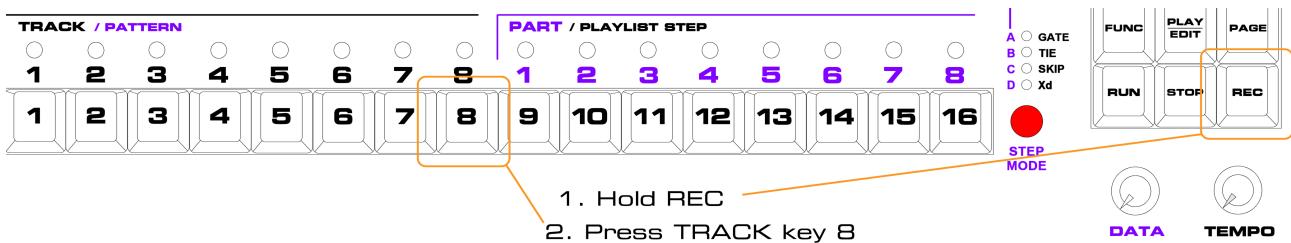
If you have a **playlist** running through a number of patterns, record will enable you to write notes into all the patterns in turn.

This is very useful for recording longer patterns of up to eight bars, or longer if you use slower timebase settings.

To activate record in **play mode**:

- **hold** down the REC key
- **press** the TRACK key for the track you want to record on

For example, to activate record on track 8:



To activate record in **playlist edit** just press the **REC** key.

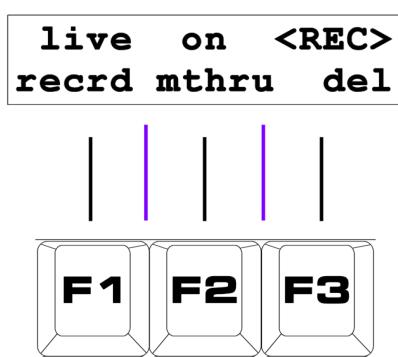
This will activate recording on the current **track** in playlist edit.

Once record is active, the normal soft-keys and pages are replaced by the record pages.

In play mode and playlist edit there is one more record page than in pattern edit – page 3 provides sculpt functions for play mode record that are already available in pattern edit page 7.

### Record Page 1 –

The first record page looks like this:



Key	Label	Function
F1	recrd	Toggles <b>live</b> / <b>ready</b> option (default is <b>live</b> )
F2	mthru	Toggles pass-through on track MIDI output channel of record input
F3	del	Delete note on current step in pattern

## F1 – Record Ready / Live

This option enables or disables the recording of notes while record mode is active.

Press the “**re cwd**” soft-key, **F1**, to toggle the setting between “**live**” and “**ready**”.

In **pattern edit**, you can always restore the pattern you are working on by exiting to **play mode** and choosing to “**lose**” changes on the way.

So this option defaults to “**live**” for **pattern edit** record.

In **play mode**, there is no way to recover what was previously in a pattern once you have over-written it.

So this option defaults to “**ready**” in **play mode** and **playlist edit**.

This means input notes will not be recorded into the pattern, but you can audition what you are going to record.

You must switch to “**live**” before any notes will be recorded, or before using the “**del**” soft-key.

## F2 – MIDI Thru

When record is activated, incoming notes to be recorded are received on the record MIDI channel configured using the “**Thru conf**” option on play mode page 7.

While record is active, notes on the record channel will not be passed through according to the normal soft-thru settings.

Instead, if you enable the “**mthru**” option, notes arriving on the **record channel** will be passed through on the **output MIDI channel** of the **track being recorded**.

This means you can automatically re-route the notes from your controller keyboard to whichever instrument you are going to record a pattern for.

## F3 – Delete

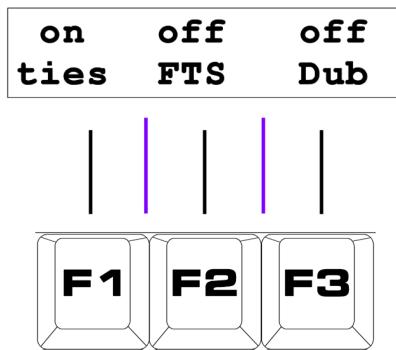
The “**del**” soft-key can be used to clear the current step as the pattern plays.

If you hold the key, all steps passed while the key is held will be cleared.

This lets you wipe out all or part of a bad take without having to clear the gate, tie and possibly aux enable settings on each step.

## Record Page 2 –

The second record page looks like this:



Key	Label	Function
F1	ties	Toggles setting of <b>tie</b> where recorded notes overlap steps
F2	FTS	Toggles Force-To-Scale of recorded notes
F3	Dub	Selects special record modes

### F1 – Ties

Soft-key **F1** toggles the record ties option.

With the “ties” option enabled, when a recorded note is held over more than one pattern step, or if the note overlaps the next note, the steps will have the **tie** setting turned on.

With “ties” 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.

### F2 - FTS

The “FTS” option selects whether notes being recorded into a pattern should be forced to the current scale before they are written to the pattern.

Note that even if FTS is turned off, notes will still be FTS’d according to the current settings. The difference is that they will be recorded into the pattern with their original note values.

## F3 - Dub

The Dub option provides two special recording modes.

Pressing the “**Dub**” soft-key cycles through three possible options:

- **note**
- **gaps**
- **off**

### *Note Dub*

With Note Dub active, the status of existing pattern steps is left unchanged, but the note and velocity values are over-written by any new note held on the MIDI master keyboard.

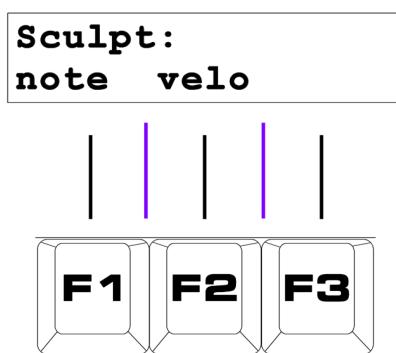
### *Gaps Dub*

Gaps Dub allows new notes to be recorded only into spaces in the existing pattern. If there is already a note on a step, it will be left intact.

This is useful for building up patterns of increasing complexity while maintaining the notes already present.

## Record Page 3 –

The third record page, only available while recording in **play mode** or **playlist edit**, looks like this:



Key	Label	Function
F1	note	Write the current position of the <b>DATA</b> knob to the <b>note</b> value
F2	upper	Write the current position of the <b>DATA</b> knob to the selected <b>UPPER MODE</b> value
F3		(no function)

The record Sculpt page allows you to perform the same sculpting functions that are available in pattern edit while recording in play mode.

The advantage here is that since you can record into multiple patterns as they are played back by a playlist, you can sculpt the data in a chain of patterns too.

Sculpt operates in just the same way as it does in pattern edit.

One subtle difference is that, since the UPPER MODE status LEDs are not used the same way in play mode, the soft-key label is used to show which value will be sculpted.

You still select from the possible options using the UPPER MODE button, just as in pattern edit.

## Use With Caution!

The danger in **play mode record** is that there is no “undo” function available.

Once you have over-written your patterns by recording in new notes, or by using the sculpt function to alter existing notes or other values, there is no way to restore what was there before.

## *Leaving Record Mode*

To exit pattern edit record, just press **REC** again.

## Arpeggio Capture

Arpeggio capture provides an alternative method of generating patterns using an arpeggiator function similar to that featured on many synthesizers.

Unlike normal record, it is available only in **pattern edit**.

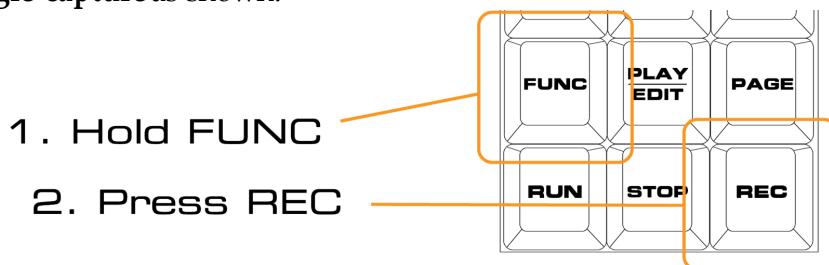
Notes received on the MIDI record channel are written into each pattern step in turn, regardless of which step in the pattern is playing.

The position of the next step to be written is automatically incremented as each note is entered, and the length of the pattern increased to include newly added steps.

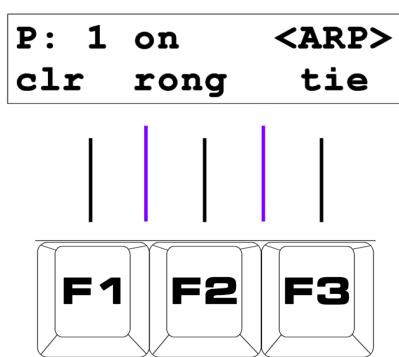
For example, if you play a 4 note chord, each of the 4 notes will be written into 4 contiguous steps of the pattern, and the length of the pattern will be increased to include these new steps.

Starting from step 1, this would give you a 4 note repeating arpeggio of the chord played.

You activate **arpeggio capture** as shown:



There is only a single page display, which looks like this:



Key	Label	Function
F1	clr	Reset to a zero length pattern (last step 1, step 1 skipped)
F2	rong	Toggle “reset on new group” option
F3	tie	Copy and tie the most recent note to the following step

The number on the top line of the display, next to the letter “P” shows the step number that the next note to arrive at the input will be written to.

On entering arpeggio capture mode, this will be set to the step after the last step of the pattern, unless the pattern is 16 steps long, in which case it will be set to step 1.

There are three soft-key functions in arpeggio capture.

### F1 – Clear

The “**clear**” soft-key, **F1**, resets the length of the pattern to **1 step**, sets **step 1** to **skip**, and resets the “next step” to be captured to step 1.

This has the effect of creating a pattern which does not play any notes – ready for a new arpeggio to be captured.

### F2 – Reset On New Group

The “**rong**” option is short for “**reset on new group**”.

This option is toggled on and off using soft-key **F2**.

With this option on, the pattern is reset when the first new note arrives after all previous notes have been released.

In effect, this automatically clears the pattern as if you pressed **F1** just before playing each new group of notes.

This makes arpeggio capture behave more like a traditional arpeggiator.

If you play a chord, each note in the chord is written into a pattern step, and played in a repeating pattern.

When you release one chord, and then play another, the pattern is cleared before being filled with the notes from the new chord.

The order the notes are played in determines their order in the capture arpeggio - so by controlling the order you press the notes in the chord entered, you can control the nature of the resulting arpeggio.

**\*TIP** You can create up, down, up & down or random direction arpeggios by altering the direction setting on the patterns.

By using aux events to add note offsets using the accumulator, you can achieve multiple octave range effects, as well as more complex arpeggio modulations.

With the “**rong**” option turned off, arpeggio capture can be used as a step-time input mode. You can enter the pattern as a series of notes from the MIDI keyboard.

### F3 - Tie

The “tie” soft-key is really only useful where “rong” is turned off for step-time input. Pressing tie will cause the most recently entered note to be copied to the next step, and the two steps tied for a longer note.

### *Leaving Arpeggio Capture*

To exit arpeggio capture mode, just press **REC** again.

## Remote Control

P3 provides a number of remote control options.

To activate these, you must enable the reception of remote control messages on a special MIDI channel. This is done using the “Enable P3 RemCon RX” **User Conf** option.

The remote control channel is selected in a similar way to the track and record input channels - see the section on MIDI channel selection for the “**rem**” channel.

### **User Interface Controls**

The User Interface Controls are a set of MIDI controller messages which allow you to access P3 user interface functions remotely.

For example, if you are a MIDI guitarist, you could configure a MIDI foot controller to give you control over bank selection, track status, track selection for real-time record and record options.

In this way you can use P3 as a “hands-free MIDI looper”.

The table below shows the MIDI controllers available for UI control:

CC #	Remote Function	Data Format
12	select/exit record track	0 - 7
13	toggle record options	0 = live/ready; 1= Dub; 2 = FTS; 3 = ties
14	enable track	0 - 7
15	mute track	0 - 7
16	toggle track	0 - 7
17	bank select	0 - 11
18	multi-function track select	0 - 7

For each of the controls that take a data value from 0 to 7, that corresponds to the *track number minus 1*. The bank select control data byte corresponds to the required *bank number minus 1*.

For example, to mute track 5, assuming a remote MIDI channel setting of channel 1, the MIDI string would be:

**B0 0F 04**

B0      MIDI CC on channel 1  
 0F      hex for controller number 15  
 04      track number 5; 5 – 1 = 4

Controller number 13, for “toggle record options”, toggles the record options in the same way as each of the record mode soft-keys shown.

For example, to toggle between “live” and “ready”, again assuming a remote MIDI channel setting of channel 1, you would send the following MIDI string:

**B0 0D 00**

- B0     MIDI CC on channel 1
- 0D     hex for controller 13
- 00     toggle live/ready option

### ***Remote Part Transpose (PXPos)***

With remote control receive enabled, incoming **note** messages on the remote MIDI channel will temporarily over-ride the current **PXPos** setting.

This allows you to remotely transpose all tracks, relative to middle C.

While remote PXPos is active, the current PXPos value shown on **play mode** page 2 is replaced by the remote value followed by an asterisk, “\*”.

Changing part, or re-starting P3, will recall the stored PXPos value for the current part.

## Using Auxiliary 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.

In essence, an auxiliary event is like a controller message that is sent internally within P3. Each event can modify some internal setting or value within the P3's sequence 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, track mutes, or part selection.
- 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 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	64
Length	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
Gate	on															

It has to be said, this is a pretty boring pattern.

## ...Made Slightly More Interesting

Now, let's assign **aux A** in your pattern to the event "**Rndmz note**" (short for "randomize note").

To do this:

- change **UPPER MODE** to **AUX**
- ensure **aux A** is selected as the current **STEP MODE**
- enter the **Aux Config** page using **FUNC + STEP MODE**
- press **step key 3** to select the **Randomize** event group

When you select an event group, the group name is shown for a couple of seconds, and then the display will show the selected event within that group.

In the case of the Randomize group the event you want, "Rndmz note", is the first event in the group, so it will be selected by default.

If you wanted another event in the group, you could either scroll through the group members by using the **DATA** knob, or step through them one at a time using the "next" soft-key, **F2**.

- to confirm the assignment of the "Rndmz note" event to **aux A**, press the "save" soft-key, **F1**

Now you can enter some values for the event.

So you can hear the effect right away, start P3 running if it isn't already.

- using **UPPER** knob **1**, enter a value of **12** for "Rndmz note" on **step 1**

Adjusting the knob should turn the step LED for step 1 **amber** – indicating that both **gate** (red) and **aux A** enable (green) are active on the step.

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.

If you look at the stored note value on step 1, using **FUNC + step key 1**, you will see that the note is still set to C4.

## 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 inside P3:

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”, P3 generates a random number 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 should 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.

This is once place 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**, **Xd** and the **aux enables**, the randomize events are a little different.

Go back to your example pattern, and assign **aux B** to the event ‘**Rnd mask gate**’, as follows:

- change **UPPER MODE** to **AUX** (if it isn’t already)
- ensure **aux B** is selected as the current **STEP MODE**
- enter the **Aux Config** page by holding **FUNC + STEP MODE**
- press **step key 3** to select the “Randomize” event group
- move the **DATA** knob until “Rnd mask gate” is shown on the display
- press “**save**”, **F1**, to confirm the assignment
- using **UPPER** knob **1**, enter 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”, P3 generates a random number 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 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.

It may be a good idea to clear out the existing patterns using **bank init**, just to ensure you don't have any unexpected aux configuration in there.

*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	64
Length	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
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	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
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 P3 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 perform the following steps:

- change **UPPER MODE** to **AUX** (if it isn't already)
- ensure **aux A** is selected as the current **STEP MODE**
- enter the **Aux Config** page using **FUNC + STEP MODE**
- press **step key 2** to select the “Inter Track” event group

As with the Randomize group, the event you want to use in this example, “grab”, is the first event in the group, so it will be selected by default.

- to confirm the assignment of the “grab” event to **aux A**, press the “save” soft-key, **F1**

## Auxes With Two Values

If you now try to enter a value for your newly assigned aux using the upper knobs, you may be in for a surprise.

When you use the UPPER knobs to edit a value for **grab**, **swap** or **push**, the display will show something like this:



This is different to the normal display when you edit an aux value.

For one thing, the event name is only “grab”, but “note” also appears on the top line.

And where you would normally expect to see “Value:” on the bottom line, it says “Track:”.

The Inter Track events have **two different values** for each step.

The first is the **target value** - the pattern value (note, velocity, length, etc.), you are going to be moving around.

Virtually all of the values can be used, as well as some combinations, so we'll not list them yet.

The second value sets the **target track**.

As you move the **UPPER** knob, you will find it only adjusts the number for "Track:" between 1 and 8, and that the word "note" remains on the top line.

This is because the knobs normally only edit the **target track**.

To change the **target value**:

- **hold** the **FUNC** key
- move the **UPPER** knob

While the **FUNC** key is held, the **UPPER** knobs allow you to scroll through all the possible **target values** on the top line of the display.

The **target track** numbers will continue to change too, but you have much finer control over those once you release **FUNC** to select a specific track.

Let's get back to the example pattern...

- adjust the **UPPER** knob on **step 1** to show "Track: 2" in the display

If the display shows anything other than "grab **note**" on the top line:

- **hold** the **FUNC** key
- adjust **UPPER** knob **1** until you see "grab **note**"
- release the **FUNC** key
- adjust the knob again to "Track: 2"

The display should confirm you have entered the correct values like this:

<b>grab note</b>
<b>S: 1 Track: 2</b>

Repeat this on each of the three other steps, **5, 9 and 13**.

Now, as your previously boring pattern plays, you will hear different notes play on steps 1, 5, 9 and 13.

This table shows what is happening:

<b>Step</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>
Pattern A Note	C4	C4	C4	C4	C4	C4	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 note, 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 note, 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 MIDI channel, 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+P (current note value plus any playlist 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)

**\*TIP** To quickly set all the steps in a pattern to the same value, so they all grab note from the same track for example, use the Sculpt upper function described in the pattern edit chapter.

## 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+p**

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 P

The “**xpose by trk n+p**” event is much the same as “**xpose by trk n**”.

The “**+p**” is there because the **note value** used from the **target track** is taken after any **playlist transpose** is applied.

If you’re familiar with playlist transpose, try editing the playlist on track 2 to transpose pattern B, and see what the effect is.

## 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 using the NOTE knob
- 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 using the UPPER knob
- set the **aux B** value on step 1 to G5 using the UPPER knob
- set the **aux C** value on step 1 to B5 using the UPPER knob

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 using the NOTE knob
- 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 using the UPPER knob
- set the **aux B** value on step 1 to 7 using the UPPER knob
- set the **aux C** value on step 1 to 11 using the UPPER knob

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

As mentioned in the chapter on real-time record, and worth repeating here, if you have configured some of the auxes in a 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 part transpose (PXPos) or a playlist step 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 P3 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.

## 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**.

<i>Example pattern:</i>				
Step	1	2	3	4
Note	C5	E5	G5	C6
Velocity	64	64	64	64
Length	6	6	6	6
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:

- change **UPPER MODE** to **AUX** (if it isn't already)
- ensure **aux A** is selected as the current **STEP MODE**
- enter the **Aux Config** page using **FUNC + STEP MODE**
- press **step key 4** to select the "Accumulator" event group

The first event in the group is "**offset note abs**", which will be selected by default.  
That's the first one we'll use, so just press "save", **F1**, to confirm the assignment.

- use the **UPPER** knob on **step 3** to enter a value of **5** for "offset note abs"

The next time the pattern passes through **step 3**, you will hear the pitch of **all following notes** in the pattern rise.

If you stop P3, then restart 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 P3 is started**, 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**.

In the case of **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 P3 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** as follows:

- enter the **Aux Config** page using **FUNC + STEP MODE**

Confirm that the top line of the display shows “offset note abs” – which should be the current assignment for **aux A**.

- press the “next” soft-key, **F2**, to select the “**offset note rel**” event
- press “**save**”, **F1**, to confirm this new assignment
- use the **UPPER** knob on **step 1** to set a value of **1** for “**offset note rel**”

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.

- 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:

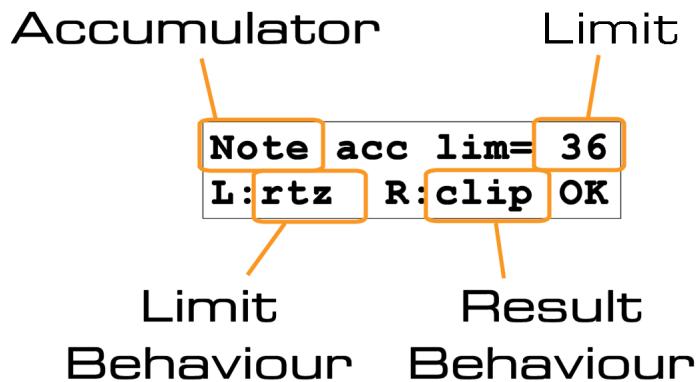
- limit the maximum value an accumulator can reach
- tell it how to behave when it reaches that value
- tell it how to behave when adding the accumulator value to one of the pattern values gives a result that is out of range for that value

The **accumulator configuration** pages are accessed from **pattern edit** using the “**Acc Conf**” soft-key, **F1**, on pattern edit page 8.

Go into pattern edit, then:

- hold the **PAGE** key
- press step key 8
- release the **PAGE** key
- press “**Acc Conf**”, **F1**

This should present you with a display that looks like this:



This shows you:

- these settings are for the **Note** accumulator
- the **limit** for the accumulator is **36**
- the **limit behaviour** for the accumulator is **rtz**
- the **result behaviour** is **clip**

So what do these settings mean?

Well hopefully you get the bit about it being the note accumulator.

## Accumulator Limit

The **limit** setting is the **maximum value** that the accumulator can hold.

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**.

## Limit Behaviour

The **limit behaviour** 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 **limit behaviour**:

- **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**.

These are just another on/off value, like the gate, tie or skip values in a pattern.

When an accumulator is cleared its **reverse switch** is also turned **off**, so they normally have no effect.

But if you configure the **limit behaviour** 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 **limit behaviour**, press the soft-key below it on the display – **F1**.

Each press of **F1** advances to the next option, so if you press F1 twice, it will step from **rtz** to **clip** to **rvtz**.

Once “**rvtz**” is shown in the display, press the “**OK**” soft-key, **F3**.

This table shows what happens now:

<b>Step</b>	1	2	3	4	1	2	3	4	1	2	3	4	<b>1</b>	2	3	4
<b>Stored Note</b>	C5	E5	G5	C6	C5	E5	G5	C6	C5	E5	G5	C6	<b>C5</b>	E5	G5	C6
<b>aux A</b>	12	off	off	off	12	off	off	off	12	off	off	off	<b>12</b>	off	off	off
<b>Note Acc</b>	12	12	12	12	24	24	24	24	36	36	36	36	<b>24</b>	24	24	24
<b>Reverse sw.</b>	off	<b>on</b>	on	on	on											
<b>Played Note</b>	C6	E6	G6	C7	C7	E7	G7	C8	C8	E8	G8	C9	<b>C7</b>	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**.

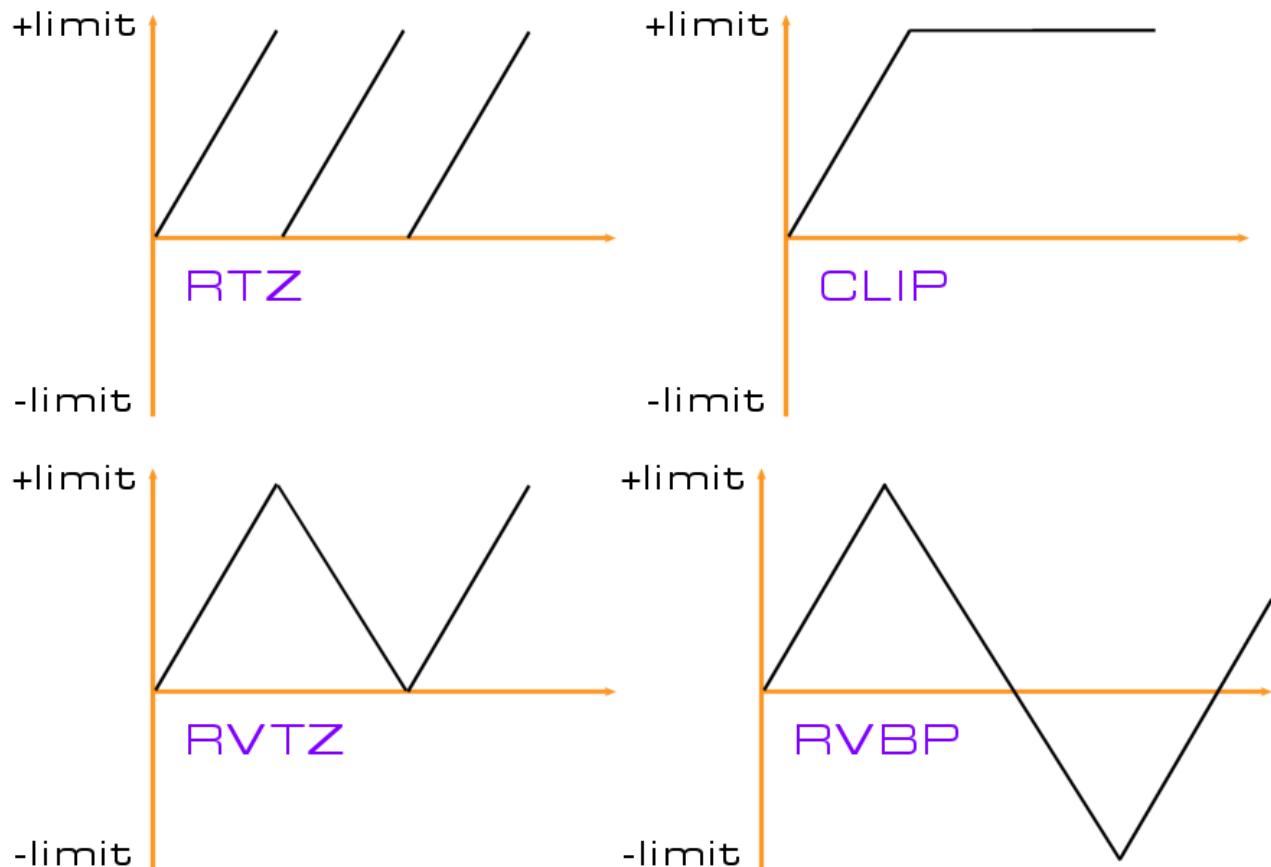
With the **limit behaviour** set to **reverse to zero**, when an offset event attempts to move the accumulator below zero, the **reverse switch** is turned **off**, and the accumulator starts to rise again.

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.

### Result Behaviour

The last setting for accumulator config, **result behaviour**, 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.

## The Other Two

The velocity and aux D accumulators both behave in the same way as the note accumulator.

They have their own set of the same configuration values, and these are accessed by using the **PAGE** key on the **Accumulator Config** page.

Press **PAGE** once and you will see the **velocity** accumulator settings, once more and you will see the **aux D** settings.

One further press of **PAGE** takes you to an extra page with the last two accumulator configuration settings.

The page looks like this:

<b>RoPS</b>	<b>xDacD</b>	
<b>on</b>	<b>off</b>	<b>OK</b>

These last two settings are stored for **every pattern**.

The “**RoPS**” setting is short for “**Reset on Pattern Select**”, and is toggled with soft-key **F1**. 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 chained together in a playlist where you want accumulator modulation to continue.

The “**xDacD**” setting is short for “**auX D Accumulator Disconnect**”, and is toggled with soft-key **F2**. 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

You may notice when you use the aux D accumulator to change the values of aux events that sometimes adding an offset of 1 seems to have no effect on the event assigned to aux D.

This can happen because the accumulator values used by events, where the event has a smaller range of values than the maximum possible for an aux, are scaled down.

For example, the event “**xpose by trk n**” takes a value from 1 to 8.

If this event were assigned to aux D, and the accumulator used to offset the event value, the **full range** of the accumulator – 0 to 127, will be scaled across the range of the event – 1 to 8.

So in order to offset by 1 track for the **xpose...** event, you must offset the accumulator by 16.

The range of 128 accumulator values divided by the 8 possible track values gives 16 steps per value.

By scaling events in this way, you can use the accumulator to make very gradual changes to events with reduced value ranges.

## The Accumulator Events

For each of the three accumulators, there is an absolute and relative offset event:

<b>offset note abs</b>	set note accumulator value	range: 0 to 127
<b>offset note rel</b>	add to note accumulator value	range: 0 to 63 / -64 to -1
<b>offset velo abs</b>	set velocity accumulator value	range: 0 to 127
<b>offset velo rel</b>	add to velocity accumulator value	range: 0 to 63 / -64 to -1
<b>offset aux D abs</b>	set aux D accumulator value	range: 0 to 127
<b>offset aux D rel</b>	add to aux D accumulator value	range: 0 to 63 / -64 to -1

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.

<b>add dAcc to leng</b>	add aux D accumulator (scaled) to length	no value
<b>add dAcc to dlay</b>	add aux D accumulator (scaled) 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:

<b>Mask gate,dAcc&gt;n</b>	defeat gate bit unless xD accumulator > value	range: 0 - 127
<b>Mask gate,n&gt;dAcc</b>	defeat gate bit unless value > xD accumulator	range: 0 - 127
<b>Mask tie, dAcc&gt;n</b>	defeat tie bit unless xD accumulator > value	range: 0 - 127
<b>Mask tie, n&gt;dAcc</b>	defeat tie bit unless value > xD accumulator	range: 0 - 127
<b>Mask xB, dAcc&gt;n</b>	defeat aux B unless xD accumulator > value	range: 0 - 127
<b>Mask xB, n&gt;dAcc</b>	defeat aux B unless value > xD accumulator	range: 0 - 127
<b>Mask xC, dAcc&gt;n</b>	defeat aux C unless xD accumulator > value	range: 0 - 127
<b>Mask xC, n&gt;dAcc</b>	defeat aux C unless value > xD accumulator	range: 0 - 127
<b>mute trk, dAcc&gt;n</b>	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 “**Knob Mask**” event group use the current positions of the **front panel knobs** to control masking of gate, tie, the aux statuses and accumulator defeats.

The knobs are only active for control of these events in **play mode**, but they use a special **pass-through** mode so that the current values for the knob positions will be held even if you enter **pattern edit** or **playlist edit** and move the knobs.

If the knob has moved from its most recent position when last in play mode, moving the knob when you have returned to play mode will not change the current value, but will instead show you what the current value is on the display.

You must **pass the knob through** this value in order for it to become active again.

After any change of bank, the stored knob positions will all be reset to zero, in order to give you a known starting point for the knob controlled events in the patterns of each bank.

The knobs used by these events are assigned automatically.

The first event from this group assigned to a pattern aux will use the **NOTE** knob above the **TRACK** key the pattern is on.

Any other event to be assigned will use the **UPPER** knob above the **TRACK** key.

So there can be only two knobs used per pattern, though a number of auxes can mask different things based on the **UPPER** knob position.

For each status value that can be masked, there are three events.

The first de-activates the status unless the **knob position** is **greater than** the **aux value**.

The second de-activates the status unless the **knob position** is **within plus or minus 3** of the **aux value**.

The third de-activates the status unless the **knob position** is **less than** the **aux value**.

The knob controlled mask events are:

<b>Mask gate,Kn&gt;n</b>	defeat gate unless knob is > n	range: 0 - 127
<b>Mask gate,Kn~n</b>	defeat gate unless knob is near n	range: 0 - 127
<b>Mask gate,Kn&lt;n</b>	defeat gate unless knob is < n	range: 0 - 127
<b>Mask tie,Kn&gt;n</b>	defeat tie unless knob is > n	range: 0 - 127
<b>Mask tie,Kn~n</b>	defeat tie unless knob is near n	range: 0 - 127
<b>Mask tie,Kn&lt;n</b>	defeat tie unless knob is < n	range: 0 - 127
<b>Mask auxB,Kn&gt;n</b>	defeat aux B unless knob is > n	range: 0 - 127
<b>Mask auxB,Kn~n</b>	defeat aux B unless knob is near n	range: 0 - 127
<b>Mask auxB,Kn&lt;n</b>	defeat aux B unless knob is < n	range: 0 - 127
<b>Mask auxC,Kn&gt;n</b>	defeat aux C unless knob is > n	range: 0 - 127
<b>Mask auxC,Kn~n</b>	defeat aux C unless knob is near n	range: 0 - 127
<b>Mask auxC,Kn&lt;n</b>	defeat aux C unless knob is < n	range: 0 - 127
<b>Mask auxD,Kn&gt;n</b>	defeat aux D unless knob is > n	range: 0 - 127
<b>Mask auxD,Kn~n</b>	defeat aux D unless knob is near n	range: 0 - 127
<b>Mask auxD,Kn&lt;n</b>	defeat aux D unless knob is < n	range: 0 - 127
<b>Mask nAcc,Kn&gt;n</b>	defeat note accumulator unless knob is > n	range: 0 - 127
<b>Mask nAcc,Kn~n</b>	defeat note accumulator unless knob is near n	range: 0 - 127
<b>Mask nAcc,Kn&lt;n</b>	defeat note accumulator unless knob is < n	range: 0 - 127
<b>Mask vAcc,Kn&gt;n</b>	defeat velo accumulator unless knob is > n	range: 0 - 127
<b>Mask vAcc,Kn~n</b>	defeat velo accumulator unless knob is near n	range: 0 - 127
<b>Mask vAcc,Kn&lt;n</b>	defeat velo accumulator unless knob is < n	range: 0 - 127
<b>Mask dAcc,Kn&gt;n</b>	defeat aux D accumulator unless knob is > n	range: 0 - 127
<b>Mask dAcc,Kn~n</b>	defeat aux D accumulator unless knob is near n	range: 0 - 127
<b>Mask dAcc,Kn&lt;n</b>	defeat aux D accumulator unless knob is < n	range: 0 - 127

## ***Knob Grab Events***

These events take the current value from any play mode knob, and use it to over-write the pattern values.

The knobs for these events are not automatically assigned – the aux value is used to specify exactly which knob should be assigned.

The knobs are numbered from 1 to 32 - the **NOTE** knobs are numbers **1 to 16**, and the **UPPER** knobs are numbers **17 to 32**.

<b>knob n to velo</b>	replace length value with current value of knob n
<b>knob n to leng</b>	replace velocity value with current value of knob n
<b>knob n to auxC</b>	replace aux C value with current value of knob n
<b>knob n to auxD</b>	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:

<b>set note</b>	
<b>set velocity</b>	
<b>set length</b>	
<b>set delay</b>	
<b>xpose by n</b>	adds a set value to the note value
<b>add n to xC</b>	adds a set value to the aux C value
<b>add n to xD</b>	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.

<b>rep*2, note + n</b>	repeat the note twice, with a note offset added to the repeat
<b>rep*2, velo + n</b>	repeat the note twice, with a velocity offset added to the repeat
<b>rep*3, note + n</b>	repeat the note 3 times, with a note offset added to each repeat
<b>rep*3, velo + n</b>	repeat the note 3 times, with a velocity offset added to each repeat
<b>rep*4, note + n</b>	repeat the note 4 times, with a note offset added to each repeat
<b>rep*4, velo + n</b>	repeat the note 4 times, with a velocity offset added to each repeat

The value range for all these events is +/- 63.

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.

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.

<b>rep*N by length</b>	sets note to repeat n times, with length of note controlling repeat time
<b>set rep*n note+</b>	set repeat note offset for rep*n
<b>set rep*n velo+</b>	set repeat velocity offset for rep*n

## 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.

<b>aux B = cc #n</b>	temporarily re-assign aux B to a MIDI controller	range: 0 - 127
<b>aux B = event n</b>	temporarily re-assign aux B to an aux event	range: aux events
<b>aux C = cc #n</b>	temporarily re-assign aux C to a MIDI controller	range: 0 - 127
<b>aux C = event n</b>	temporarily re-assign aux C to an aux event	range: aux events
<b>aux D = cc #n</b>	temporarily re-assign aux D to a MIDI controller	range: 0 - 127
<b>aux D = event n</b>	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.

<b>set lstep abs</b>	sets the last pattern step	range: 1 - 16
<b>set lstep rel</b>	adds a given value to the last step	range: +/- 15
<b>set norm tbase</b>	set tbase to one of the standard settings (1, 2, 4, 8, 16, 32, 64)	
<b>set trplt tbase</b>	set tbase to one of the triplet settings (2T, 4T, 8T, 16T, 32T, 64T)	
<b>set direction</b>	set direction (always reset to stored value at the end of the pattern)	

## *Global Control*

This group allows you to modify global settings.

<b>mute trk n</b>	deactivate track n	range: 1 - 8
<b>un-mute trk n</b>	activate track n	range: 1 - 8
<b>select part</b>		
<b>set pchain length</b>		
<b>set tempo</b>	sets the global tempo, temporarily over-riding the tempo knob	



## Appendix A - Preset FTS Scale Definitions

FTS Scale Preset Definitions:

No.	Name	Short name	Notes
1		off	all
2	major	maj	CDEFGAB
3	aeolian	aoe	CDD#FGG#A#
4	harm. minor	har	CDD#FGG#B
5	mel. minor	mel	CDD#FGAB
6	blues 1	blu	CDD#FF#GAA#
7	pentatonic	pen	CDEGA
8	mixolydian	mix	CC#EF#GAA#
9	lyd dim	lyd	CDD#F#GAB
10	blues 2	bl2	CDD#EGA
11	major aug	aug	CD#EGG#B
12	arabian	arb	CDEFF#G#A#
13	balinese	bli	CDD#GG#
14	hungarian 1	hng	CDD#F#GG#B
15	oriental	ori	CC#EFF#G#A#
16	raga-todi	rga	CC#D#F#GG#A#
17	chinese	chi	CEF#GB
18	hungarian 2	hn2	CD#EF#GAA#
19	japanese 1	jap	CC#FGG#
20	japanese 2	jp2	CDFGG#
21	persian	per	CC#EF#G#B
22	diminished	dim	CDD#FF#G#AB
23	whole tone	who	CDEF#G#A#



## Appendix B – User Conf Options

This table lists the purpose of each of the User Conf options accessed from **play mode** page 7:

Quick ptn select	With this option enabled, in <b>playlist edit</b> , when you select a new pattern on a playlist with a length of 1, P3 will automatically save the playlist and return to play mode. This is useful for users who don't take advantage of the playlist features, and just want to select different patterns quickly.
Hold chain mutes	When a <b>part chain</b> is selected, this option will cause the current track mute status to be copied to the following parts as the chain advances. The initial status will be recalled from the first part selected. Note that the existing mute status for the parts is over-written with this option active, even when you subsequently select one of the parts by itself.
Aux edit sets on	With this option enabled, in <b>pattern edit</b> , when an auxiliary value is changed using the upper row of knobs, the status for the edited step will be set to 'active'.
Send bpgm on run	Normally the 'bpgm' program change message is only sent when a new bank is selected. With this option on, it will be re-sent whenever the sequencer is started. This may cause you to lose edits on patches on some synths.
Lock btempo	If you have specified an initial tempo for a bank, with this option active, the tempo knob will be disabled to prevent accidental changes of tempo.
Start on cont	With this option active, a MIDI Continue message is treated as a MIDI Start.
Redir CC on REC	In record mode, incoming MIDI notes on the record channel are re-routed to the MIDI channel of the track being recorded. Controller messages still pass according to the normal soft-thru setting. With this option is enabled, controllers also pass on the record track channel (if record <b>mthru</b> is enabled).
Mchans per bank	With this option enabled, MIDI channels are still configured as normal, but will be remembered uniquely for each bank.
MIDI thru FTS	This option chooses whether notes passed by soft-thru will have the current FTS settings applied. It can also be changed from the <b>thru conf</b> page.
Auto save edits	With this option enabled, the normal save/lose/back options are not shown in pattern edit - the pattern is simply saved immediately.
Enable mixer	This option enables sending of MIDI volume and pan messages from the knobs above the PART keys 1 to 8, only in <b>play mode</b> . The MIDI channels used are those assigned to each of the 8 tracks.
Note numbers	This option will cause notes to be displayed as MIDI note numbers rather than note names in <b>pattern edit</b> .
P3 RemCon TX	This enables the sending of P3 remote control messages.
P3 RemCon RX	This enables the reception of P3 remote control messages.
No ANO on stop	This option disables the sending of MIDI all-notes-off messages normally sent whenever P3 is stopped.
Sticky keys	This option enables "stickiness" of the modifier keys FUNC, PLAY/EDIT and REC in play mode, so you can more easily use P3 one-handed.
Global Xd Ignore	This option disables the Xd row in all patterns
Middle C = C4	This option alters the display of MIDI note octaves from the MIDI standard of C5 to the traditional standard of C4. Notes in octave -1 are displayed with a minus sign only, since there are only three characters for note names.



## Appendix C - Memory Organisation

In order to provide flexibility in use, the allocation of P3's available pattern storage can be configured by the user in one of three ways.

These are:

- 3 banks of 8 tracks, with 16 patterns for each track
- 6 banks of 8 tracks, with 8 patterns for each track
- 12 banks of 8 tracks, with 4 patterns for each track

These options allow you to choose the balance between the number of banks, and the number of patterns per bank.

This choice will depend on the style of music you are making. Typically a 'bank' will be used to hold all the parts and patterns for one piece of music. The more complex your pieces are in terms of number of different patterns, the fewer banks will be available.

The memory organisation can be changed by holding down the **PAGE** key **during power-on**.

After the initial LED cycle, this will present the following menu:

**memory layout:**  
**3x16    6x8    12x4**

The desired option is selected by pressing the soft-key - **F1**, **F2** or **F3** below it.

Changing the memory layout is **non-destructive** - existing patterns are re-distributed among each track cross the banks.

For example, in 3 x 16 mode, patterns 1 to 8 on track 1, bank 1, will remain as patterns 1 to 8 on track 1, bank 1 if you switch to 6 x 8 mode.

But patterns 9 to 16 on track 1, bank 1 from 3 x 16 mode, will become patterns 1 to 8 on track 1, bank 4 in 6 x 8 mode.

The extra parts and playlists that become available in the 6 and 12 bank modes are hidden from the modes with lower numbers of banks, but their data is not lost if you switch.

This means you can change the memory layout at any time, without losing data - although you may have to copy and paste some patterns around.

## Note on Playlist Edit for 3 Bank Configuration

Since the 3 bank memory configuration gives 16 patterns for each track, and playlist edit normally only assigns 8 of the step keys for pattern selection, there are some slight differences in playlist edit to accommodate this.

**PLAYLIST STEP** selection for **edit view** must be performed by:

- **holding** the “step” soft-key, **F1**
- pressing the **PLAYLIST STEP** key for the step to be edited

This use of the “step” key for playlist step selection means that **all 16 step keys can be used for pattern selection** on the **current playlist step**.

Since the **pattern number** must be displayed using the red LEDs, the step currently being edited is displayed **ONLY** by the number in the top left of the alphanumeric display.

A red LED above the **PLAYLIST STEP** keys indicates the **pattern number** on the current step.

## Appendix D - Miscellaneous Useful Info

### P3Tools

P3Tools is an independently written utility that provides extensive pattern management facilities for P3, using MIDI sysex transfer.

This tool is highly recommended if you have a Windows-based PC.

It will require the installation of the Microsoft .NET framework, and is available for download via the P3 Yahoo group (see below).

### Memory Initialisation

A special key combination can be used during power-on to perform system initialisation. This will **clear all data memory**, and **reset the configuration options to factory defaults**.

Hold **F1** and **step key 7**, then power-on P3.

Continue holding the keys until the display shows:

<b>Wiping ALL data!</b>
<b>[confirm] cancel</b>

Press both **F1** and **F2** to confirm you want to completely wipe the sequencer memory, or **F3** to cancel the initialisation process.

### SysEx Firmware Upgrade

P3 has a Flash ROM based CPU which supports firmware upgrades via MIDI System Exclusive dump.

In order to load a new firmware image, you must enter **Firmware Upgrade Mode** – a utility provided by the CPU boot-loader program, which is entirely separate from the main program.

This is accessed by holding **FUNC** and **PAGE** together **while powering on P3**.

Each time P3 is powered-on, it performs a firmware image check to verify the contents of program memory. It will automatically enter firmware upgrade mode if any error is found.

On entry to firmware upgrade mode the display will show something like this:

<b>v3.2 CHK = ABCD</b>
<b>boot flash</b>

The top line shows the checksum value for the current ROM image, and the bottom line the two soft-key options.

If you press the “boot” soft-key, **F1**, P3 will continue booting up as normal.

If you press the “flash” soft-key, **F3**, P3 will prepare to receive a System Exclusive dump of a new firmware image.

Firmware images are available for download as sysex dump files (**.SYX** extension) and standard MIDI files (**.MID** extension).

The files can be played back by any MIDI utility that supports these file types.

The **.SYX** files require a delay between sysex blocks of around 100ms.

This delay is encoded in the **.MID** file.

The **P3Tools** program includes a firmware upgrade utility that takes advantage of hand-shaking messages returned by P3 to optimise the transfer process and provide error detection, and is highly recommended.

When a complete dump has been transmitted, P3 will indicate whether the upload was successful or not, and show the new image checksum. You may then boot as normal.

Where possible, new firmware versions will not require the backup and restore of pattern data, but in cases where the internal storage of pattern data has been modified, this may be required.

It is generally regarded as good practice to do a data backup before a firmware upgrade anyway.

Details of firmware releases are provided via the Sequentix website, and the P3 Analogue Sequencer ‘Yahoo!’ group.

**<http://www.sequentix.com>**

**<http://launch.groups.yahoo.com/group/analogue-sequencer/>**

# Index

- after-touch, 56
- auxiliary events, 122
- bank initialise, 40
- bank program change channel, 37
- bank tempo, 42
- Berlin School, 149
- chord memory, 135
- Chris Franke trill, 149
- default page, 23
- default playlist, 86, 93
- display pages, 22
- Force To Scale, 25
- global bar length (GBar), 31
- global bar sync, 105
- hands-free, 120
- initialise pattern, 70
- memory organisation, 157
- MIDI clock, 35
- MIDI output channel, 36
- MIDI soft-thru channel, 38
- MIDI volume and pan, 45
- mute hold, 33
- note display format, 27
- note knobs, 13
- note quantize, 25
- nudge, 67
- numeric rows, 47
- part chain, 19, 155
- part chain repeats (PCRep), 31
- part copy, 33
- part keys, 17
- part transpose (PXPos), 31
- parts, 17
- pattern length, 69
- pitch-bend, 56
- play mode controls, 16
- playlist edit, 21
- playlist hold, 110
- polyphonic record, 135
- post-FTS transpose, 32
- pre-FTS transpose, 32
- program change, 56
- record knob movement, 79
- record MIDI channel, 38, 113
- remote control channel, 37
- reverse copy, 71
- setting note, 13
- setting velocity, 13
- soft-keys, 14
- special purpose MIDI channels, 37
- status rows, 47
- step-time input, 118
- System Exclusive transfer, 42
- test MIDI connections, 29
- tick, 49
- track keys, 16
- tri-colour LEDs, 52
- triplets, 66
- user configuration, 44
- view step data, 51
- VU meter, 46
- zero position, 31