The M8 Companion

Version 8.15

September 11, 2026

(a flânerie by Prabhakar Ragde and Brian Simpson)

This is a tutorial introduction to the Dirtywave M8, a portable music device. It is a work in progress, though now chapters 1 through 8 have complete drafts, at least. Chapter 9 only has a few notes for future work, so is transitional.

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

1 Introduction

The Dirtywave M8 is a device for creating multi-tracked music, combining synthesized sound and sample manipulation. It does a particularly good job of managing the tradeoffs between portability and usability. We are not prone to hyperbole, and we will adopt a measured tone through much of this flânerie, but don't mistake this reserve for judgment. Each of us has many electronic musical devices, but the M8 is special. At times, it is the only such device we will have with us (not counting our phones). This work is an attempt to give back to the creator and to the community that has grown up around this device. If you're on the fence about acquiring one, do not hesitate (if you can afford it). If you have one in your hands as you read this, you're in for a treat.

The first model (M:01) units shipped in October 2021, and the second model (M:02) started shipping in May 2024, with production ongoing.





(images from dirtywave.com)

M:02 (on the right) has a larger, flush-set screen, a fully aluminum case, a larger battery, a built-in microphone, and USB-C instead of micro-USB. But M:01 is a very capable device, in no way obsolete (though no longer manufactured); it continues to be fully supported and will receive firmware upgrades at the same time as M:02. The two models have the same capabilities except for those that result from the physical differences (such as being able to record from the microphone).

The M8 looks more like a gaming device than a musical device, and this is no accident. The main hardware inspiration is the Nintendo Game Boy (1989-2003).



(image from Wikimedia, by Evan Amos)

The navigation cross on the Game Boy corresponds to an inverted-T of keyswitches on the M8, as is common on computer keyboards, and the four play buttons of the Game Boy correspond to four more M8 keyswitches, two labelled SHIFT and PLAY (below), and two labelled OPT and EDIT (above and to the right).

The popularity of video games in the 1980s spawned the musical genre known as chiptune, emulating the low-fidelity music that accompanied games. The Little Sound DJ (LSDJ) software (2000-present) was developed as a purely musical application for the Game Boy. LSDJ is the main software inspiration for the M8 user interface, but the M8 is capable of far more sophistication than typically exhibited in chiptune. Neither of us have any particular interest in chiptune, but we were both drawn to the M8 for its general musical possibilities.

Dirtywave was a one-person corporation for the creation of these models, that person being Timothy ("Tim") Lamb. (Musician and long-time core user mikey303 joined the corporation at the beginning of 2025.) Tim was active in the chiptune community when LSDJ was released, and wrote several software synthesizers. M8 was originally planned as a hardware MIDI controller using an LSDJ-style interface, but quickly moved beyond that. Besides the one hardware revision to date, there have been several major firmware revisions, each offering new features. Many devices succumb to feature bloat, where new features seem stuffed into corners, or interfere with the use of previous features. This is definitely not the case with M8; the additions seem carefully thought out and elegantly implemented. Users are quite happy with the level of support, and the speed with which bugs are fixed. Small preorder batches have been periodically announced, and they sell out quickly. There is a thriving user community centred around a Discord server.

The PDF manual for M8, available on the Dirtywave website, is well written, and should definitely be consulted alongside this document. But Tim prioritizes making new units, fixing bugs, and updating the firmware, so the manual can lag behind (when we acquired our M8s, it was a full software version behind) and it never contained all the details (there is some on-device help not duplicated in the manual). Other written materials are harder to come by. There are a few on the Discord server, along with pinned messages and what can be found through a search of historic threads, as well as many introduction and tutorial videos on social media platforms (some quite good).

Our goal is to provide a structured path to gaining experience on the M8. We don't expect to be comprehensive,

but we will try to collect some of the best folklore techniques. Hopefully, there is something here of value for those more familiar with the M8, or with electronic music in general.

The M:02 is about 92×129×16mm in size, with the keys rising another 5mm above the front panel. It weighs about 290 grams. It has a 3.5" IPS TFT display (480×320), which has touchscreen capability, though this is only put to modest use. (Tim does not feel it is fast or accurate enough to rely on it more.) Battery life is about 12 hours under normal circumstances (screen brightness is the main factor). It has two rudimentary speakers (the grilles are visible at lower left and right), and the microphone port is just below the lower left corner of the screen.

On the top side are four 3.5mm TRS ports, two for audio in/out and two for MIDI in/out, as well as a single USB-C port, used for charging and power, and for USB audio/MIDI in/out. The M8 is a USB MIDI client but not a USB MIDI host, so one cannot connect USB client keyboards to it without a host being involved (typically a computer or phone, though standalone host boxes are available).

On the right side is the power button (soft on/off), as well as a micro-SD card slot. As of firmware 6.5 (Jan 2025), the M8 can be mounted as a hard drive on a computer, tablet, or smartphone. But for backups and large file transfers, it's better to physically eject the SD card and put it into a standard reader.

The computing engine in the M8 is a Teensy, which is a 32-bit microprocessor chip (M:01 uses a Teensy 4.0 and M:02 a Micromod). For performance reasons, there is no operating system running, the way there is on your phone or computer. This enables seamless project changes without any awkward silences. Another unusual feature is that samples (recorded on the device, or imported) are streamed on playback directly from the SD card. Most samplers move samples from a drive to random-access memory (RAM), meaning some limitation on the length or number of samples. On the M8, the only limitation is the capacity of the storage card.

The M8's user interface is in the **music tracker** paradigm, which originated on the Commodore Amiga (an early personal computer) in 1987. Current popular software trackers on major computer platforms include Renoise and SunVox. There are fewer hardware trackers, but we can mention Polyend Tracker (in various versions) and, in the Eurorack module format, Nerdseq from XOR Electronics.

A tracker displays a grid of note information with time running vertically down the screen, and the information associated with a single note running horizontally (in a single row of the grid). You can see the grid in the M:01 photo above.

Time is further structured into discrete segments, as in sheet music. Sixteen sixteenth notes are collected into a **phrase** (which may be cut short); related phrases are collected into a **chain** of user-defined length; and an arbitrary number of chains are strung into one **track** of a **song** (one song per project). Eight tracks can play simultaneously.

There are 128 user-defined instruments available in each song (an arbitrary number can be stored globally for quick loading). Each instrument can use one of four synthesizer types, or a sampler type, which can record and playback samples (there is an editor with visual waveform display). There is also a MIDI type for sequencing external devices such as other synths, with an internal loopback option. A user-defined instrument stores the chosen type, plus settings for all associated parameters affecting the sound, and modulations (automated changes to those parameters over time). You can see the definition of an instrument in the M:02 photo above. Most types are monophonic (single voice), though there is a supersaw type that can play user-defined chords, and an option of internal MIDI to distribute the notes of MIDI chords over a set of other tracks, with allocation handled by the device.

Part of the information associated with a single note in a phrase is the instrument used to voice it. It is thus possible to have a phrase with a different instrument playing on each of the sixteen possible notes. This allows for considerable flexibility already, but even more is provided by having each note issue up to three **commands** that do useful things. We'll defer discussion of further mechanisms to later chapters. There are also three send effects (originally chorus, delay, reverb, but recent firmware permits a phaser, flanger, or comb filter as alternative to chorus) with amounts set per instrument (changeable per note via commands) and a per-instrument limiter (really more of a clipper/waveshaper) that can be used for various flavours of overdrive. Recent firmware revisions also added a global three-band parametric equalizer and end-of-chain compressor, as well as a per-instrument equalizer and a pool of equalizers that can be shared by multiple instruments.

The M8 also can send display information to its USB host. The website m8.run (hosted by Tim) will, when loaded in a WebMIDI-capable browser (such as Chrome), display a copy of the M8's screen in the browser window (which is presumably much larger). That's how we made the screen captures in this document. In fact, there is a version of the firmware called "headless" that runs on a standalone Teensy, displays in this fashion, and uses the computer keyboard

for buttons. It's intended for use at work or another location where you'll go regularly, but don't want to always carry your M8 back and forth. Some people have used headless to prepare while waiting for the next batch to go on sale, but the button placement and thus muscle memory is different, so this is of limited value.

The small number of buttons (and their lack of sensitivity to impact force) means that it's not going to be possible to, in real time just on the M8, play a melody, finger-drum, or tweak several parameters at once. For that sort of thing, you will need to hook up a MIDI controller with keys, pads, and knobs. But the information-packed displays in the various views on the M8, and the ease of switching among them, makes it well suited to designing full songs starting with one or more simple melodic ideas and elaborating on them. It's not surprising that it sells out so quickly, or that there is a fiercely loyal user community.

In the next two chapters, we'll start with an overview of the organization of the M8, and how to navigate among its various screen views. Then we'll give you audio of a brief, simple song (designed for didactic purposes, but not without its own charms) and show you how to construct it from scratch on the M8. Along the way, you'll learn more about key commands and configuration, and start to develop some muscle memory. There are natural places to digress and let you explore features on your own, if you wish. Subsequent chapters will delve into more advanced topics, and give you more opportunities to practice efficiently.

2 Prelude

The M8 is designed to be held in both hands with keys pressed using both thumbs, but that's not the only way it can be used. You could hold it in one hand and poke at it with one or two fingers of the other hand. You could put it flat on a table and use one or several fingers on either or both hands. You could put it in a stand that tilts it at an angle. It's up to you.

If you have bought a new M:02, it will start up in the M:02 demo song. There are several demos in the factory content, and one in particular has been chosen. The M:01 demo song is also available. If you have bought a used M:02 or M:01, and you are lucky, the previous owner has arranged for it to start up in the M:02 or M:01 demo song. If not, it may start up in the last song the previous owner played or worked on. We will assume only that your device starts up in some song.

2.1 M8 organization

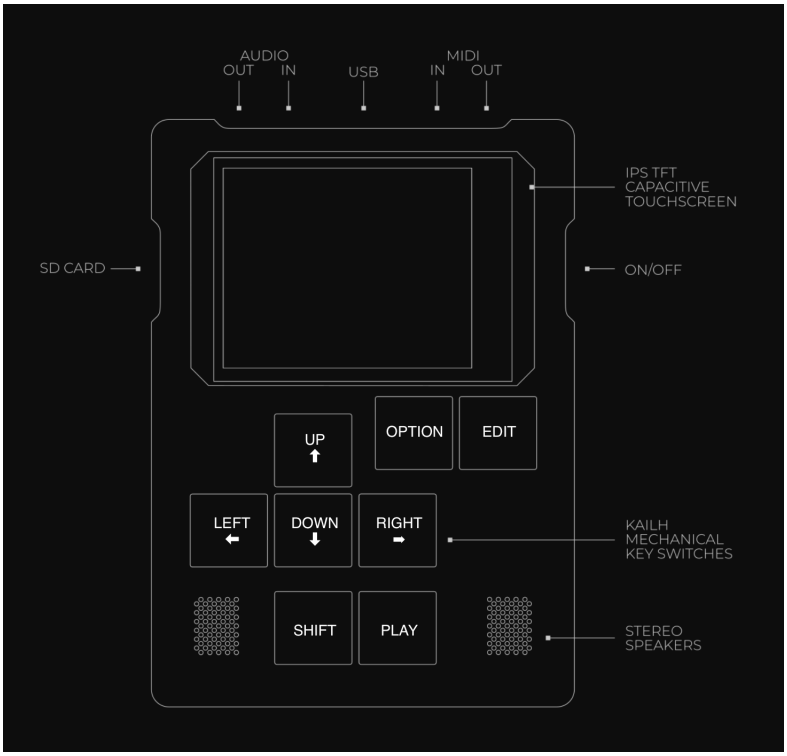
The M8's SD card has a typical file system on it and uses conventions from desktop computers. When you mount it on your own computer, you will see everything, but when you are saving or loading on the M8, the view is often restricted to a relevant zone.

There are files and folders (directories). There is a root folder that contains everything. One of the subfolders in the root folder is called `Songs`, and on a fresh device, it will have a subfolder called `Demos` containing several factory demo songs. The convention is to represent a path to this subfolder from the root folder as `/Songs/Demos`. This terminology is also used on Unix, and therefore macOS and Linux. On Windows, the slash is a backslash, but otherwise it's the same idea. The other file convention that M8 borrows from Unix is that when you are in a subfolder, the enclosing folder is called `/..`, and so you select that to move up a level. Sometimes, a family tree metaphor is handy, where the enclosing folder is the **parent**, and the subfolders are **children**.

2.2 Basic navigation

There are only eight keys on the M8, so you should expect that there are many key combinations, and many cases where the meaning of a key or key combo is situation-dependent. The M8 has been designed so that, for the most part, a key combo in one situation may not act identically to the same combo in another situation, but it will act similarly, and the similarity will be logical. We will point out these similarities, but you should also watch for them, as they are helpful in the learning process.

We will adopt the convention for keys and key combos from the M8 manual, so the up key is called [UP]. We'll see how key combos are succinctly described when we get to them shortly.



(outline image from Dirtywave.com, key labels added)

You will not be surprised to hear that the arrow keys are used for navigation on the M8. But there are several types of navigation, due to the fact that not all the information would fit on a single screen. (Even in a computer app on a larger screen, not all information is displayed at all times.)

The different screens on the M8 are called **views**. The arrow keys, by themselves, move the **cursor** (the four flashing rectangle corners that indicate a specific place on the screen) within a view. Tapping the [DOWN] key will move the cursor to the next spot below; holding it will act like repeated tapping and keep the cursor moving down (this is behaviour you may have experienced with the spacebar on computers, tablets, and phones). To move from one view to another, we have to do something else.

Of the other four keys, the [PLAY] key is the most straightforward. Tapping it by itself causes music to start or stop (it is a toggle). But exactly what music plays depends on the view and even sometimes where the cursor is, as we'll see.

The three remaining keys are **modifiers**. They are sometimes used on their own, but more often in combination with the arrow keys or each other, to provide new meanings.

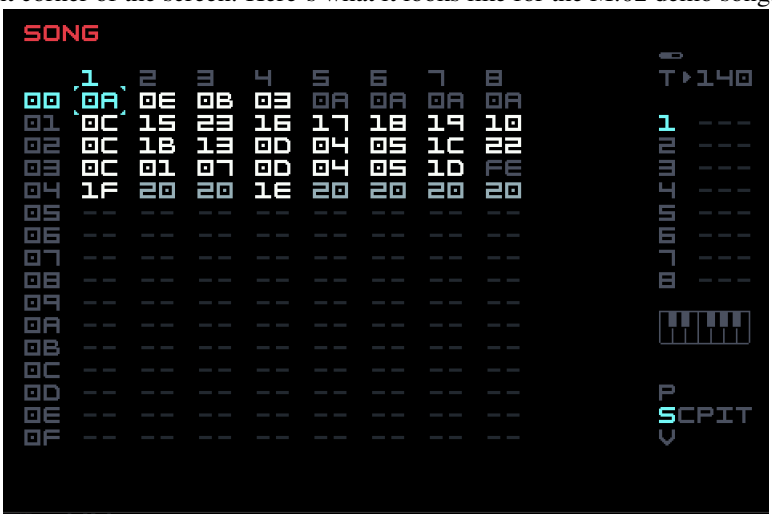
Of the three modifier keys, the one to the left of [PLAY], namely [SHIFT], tends to signify a smaller change in meaning. On a computer keyboard, it is used to capitalize letters, by holding the shift key and tapping the chosen letter. [SHIFT] tends to be used similarly on the M8; you press it first and then, while holding it down, tap or hold another key.

Used in conjunction with the arrow keys, [SHIFT] signifies a change in view. If you hold [SHIFT] and then tap [RIGHT], you move from one view to the view arranged to its right (we'll explain the arrangement of views and what each one shows, below). Again, following the convention in the manual, we'll describe this as [SHIFT]+[RIGHT].

The leftmost modifier key on the top, [OPTION], tends to signify a larger change in meaning. On a computer keyboard, it is used to choose alternate characters. When it makes sense, [OPTION] is generally used to mean "no" or "leave". Finally, the rightmost modifier key on the top, [EDIT], not surprisingly, is used to change state or stored values. When it makes sense, [EDIT] is generally used to mean "yes" or "enter". Remembering these general principles makes it easier to learn and quickly use various key combos.

2.3 Overview of views

The M8 will start up in some song, and in Song View. (Feel free to play the song, if you haven't already.) We'll explain what you're seeing shortly, but for now, we want to point out the little mini-map made of letters at the bottom right corner of the screen. Here's what it looks like for the M:02 demo song.



The S is brighter to indicate Song View. The name of the view will be at the top left of the screen, but the mini-map is useful to know how to navigate to other views.

The horizontal arm of the cross, `SCPIT`, represents five views. You can navigate to the right with [SHIFT]+[RIGHT], and back towards the left with [SHIFT]+[LEFT], and as you do, you will pass through other views, with most of the screen changing entirely. On the mini-map, the position of the highlight will change. The other views, in left-to-right order, are Chain View, Phrase View, Instrument View, and Table View.

There is one hidden screen to the left of Song View, and that is Live Mode. It is actually a variation on Song View. We're not going to talk about it until later, but if you accidentally find yourself in it, you need to know that to get back to Song View, you go left again, not right as you might expect. The help text at the top of the screen will remind you of this. It's like a little one-way cul-de-sac loop that you don't need to worry about for now.

The vertical arm of the cross in Song Mode has a `P` above the `S` representing Song Mode, and a `V` below. Above is Project View, reachable by [SHIFT]+[UP]. That's where we'll go in a moment to create a new empty project, but first we should mention that the `V` represents Mixer View, and you can go there to adjust the output volume, as described in the manual, if it is too loud in your headphones or earbuds.

As you navigate along the horizontal arm, what is shown on the vertical arm may change. This is also the case for navigation along the vertical arm at times. For example, in Mixer View, another letter appears below the `V`. In other words, the map is not fixed or complete; it changes depending on where you are.

2.4 Hexadecimal notation

If you look at the Song View of the demo song, or the photo of it above, you will notice that some entries have a curious mix of numbers and letters. For example, under the cursor, you see 0A, and at the bottom of that column, you see 1F.

These are in fact numerical values, represented in hexadecimal notation ("hex" for short). Our familiar decimal notation (base ten) uses the digits 0 through 9. Hexadecimal (base sixteen) adds six more digits, using A through F for those. A represents a digit value of ten, B a digit value of eleven, and so on up to F with a digit value of fifteen.

Using these extra digit values makes the representation of most numbers shorter. It's also handy for working with powers of two (one, two, four, eight, and so on) which are prominent in computer science. So hexadecimal was used a lot on early computers, including the Commodore Amiga. But powers of two also are useful in music (four beats to a bar, half notes and eighth notes, etc.), and the M8 still has a small screen, so hexadecimal is a good choice. You will quickly get used to it.

The numbers zero through nine are 00 through 09 in two-digit hexadecimal notation. But ten is 0A, and so on up to fifteen as 0F. Sixteen is 10 (one times sixteen, plus zero). Thirty-two is 20 (two times sixteen, plus zero). The entry 1F we pointed out above represents thirty-one.

You will rarely have to translate hexadecimal notation into decimal. For example, it suffices to know that changing the right digit to one greater or smaller is a small increment or decrement in the represented value. But changing the left digit to one greater or smaller is a larger increment or decrement (you can probably figure out exactly how much, but in many circumstances, you don't need to). All the entries in the song view are labels of chains, and there's no need to think of them as numbers at all.

Five is half of ten, so with two decimal digits, one can represent zero through ninety-nine, and the closest whole number to halfway is fifty (50 in decimal). Similarly, eight is half of sixteen, so with two hexadecimal digits, one can represent zero through two hundred and fifty-five (FF in hexadecimal), and halfway is 80 in hexadecimal, because eight is half of sixteen. So we can reason about a useful value (halfway through a range) without doing a full translation into decimal. One-quarter of the digit range falls between 2 and 3 in decimal, but in hexadecimal, it falls exactly on 4. Similarly, three-quarters of the digit range in hexadecimal is C. You'll have lots of reasons to remember 0, 4, 8, C soon enough.

We sometimes want to represent negative numbers. A parameter controlling some aspect of the device could be **unipolar** (taking on non-negative values, from zero up to some maximum value) or it could be **bipolar** (with zero in the middle or near the middle of a range that includes both positive and negative values). There is a convention for representing negative numbers with only two hexadecimal digits. As before, 00 represents zero, 01 represents one, and so on, up to 7F, which represents one hundred and twenty-seven.

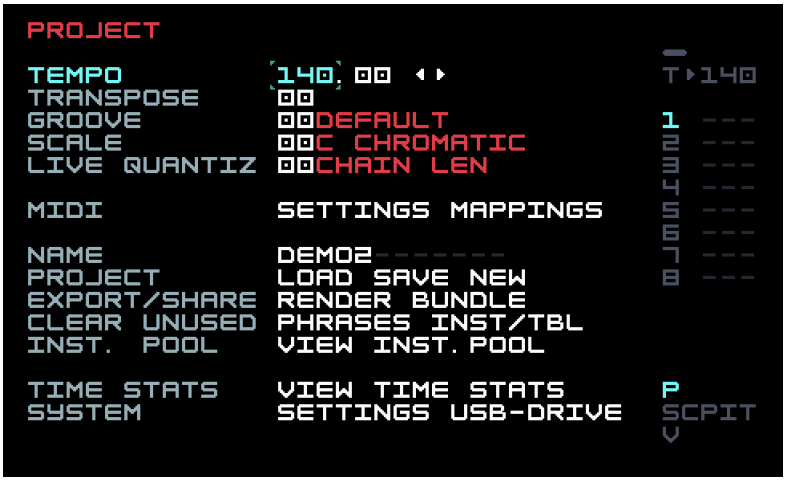
But for a bipolar range, FF represents negative one, FE represents negative two, and so on back to 80, which represents negative one hundred and twenty-eight. It's even harder to translate a negative number represented in hexadecimal into decimal, so except for the ones close to zero, don't try. There's a full table in an appendix of the M8 manual.

These two conventions mean that when looking at a two-digit hex value, one has to know whether it is referring to a unipolar or bipolar range. This is usually easy to figure out. Negative volume makes no sense, so that would be unipolar. But a **modulation** (change) in volume could be negative, so that would be bipolar.

Not everything on the M8 uses hexadecimal. For example, the tempo or speed of a song (often expressed in beats per minute, or **BPM**) is shown in decimal notation, because there's space in that view, and powers of two don't really come into play with a tempo like 138 BPM. Usually, when hexadecimal is used to number things (like the rows in Song View, whose labels are in the left column), the numbering starts at 00 (for the first row). But when regular decimal is used, the numbering starts at 1 (like the eight tracks, numbered 1 through 8 in Song View).

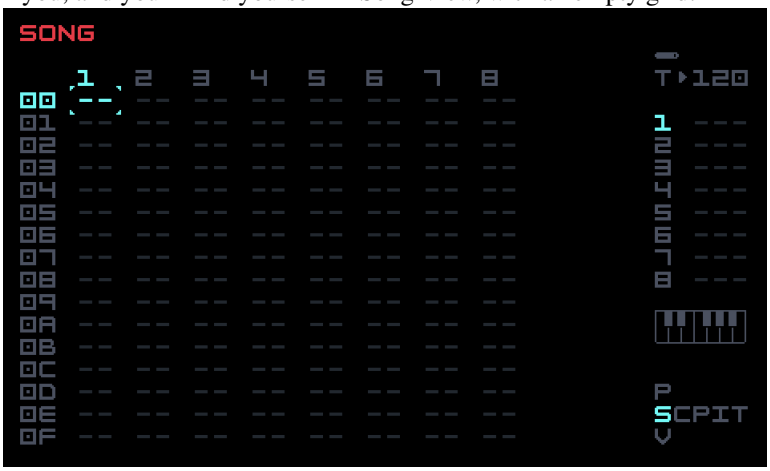
2.5 Creating a new project

If you're still in Song View on a demo song, [SHIFT]+[UP] will get you to Project View. If you've moved elsewhere, you'll have to navigate a bit more. Here's what Project View looks like for the M:02 demo song.



The left column has row labels, and the rest of each row has useful information or actions. Using just [DOWN], navigate down to the row labelled PROJECT. The row label for the row that the cursor is on will be highlighted in blue, and what the cursor surrounds will also be highlighted. When you reach the PROJECT row, the cursor will be on LOAD; use [RIGHT] to navigate to NEW. (We won't always spell out navigation so labouriously.)

With the cursor on NEW, tap [EDIT] to "enter" or select this action. The M8 doesn't need any more information from you, and you'll find yourself in Song View, with an empty grid:



But where is this song located? Go back to Project View, and you'll see that the information is different from the demo song.



The main thing to notice here is that the NAME row is empty. This is a nameless project, so far. If you try to navigate to SAVE in the PROJECT row, and select it, you will see a line of text on the bottom of the screen telling you to enter a name before saving. So navigate up to the NAME row. The cursor will surround a single hyphen (an indication of emptiness), and if you hold [EDIT], the Project View will disappear and you will see a screen for selecting a character. Keep holding [EDIT]!



You can choose your own name a character at a time, but at this point, the simplest thing to do is to navigate down to RANDOM (still holding [EDIT]) and let go of [EDIT] when your cursor is on RANDOM (as it is in the above screenshot). You will be returned to Project View, with the name field filled in by a random name made from two random words.

If you got tired or accidentally let go of [EDIT] too early, you will probably see a single character in the name field (the one your cursor was on when you let go) with the cursor on a hyphen after it. To delete that single character, navigate to it, and use [EDIT]+[OPTION], which is commonly used for ”delete”. You can also use this technique to delete the random name, character by character. Because [EDIT] by itself has a special meaning while working with names, you might want to hold [OPTION] first and then press [EDIT] to do deletion. Deleting the first character this way has a special behaviour: the rest of the name shifts one position left to fill the gap. You can change individual characters in a name by editing, or change blank spaces at the end into characters.

[EDIT]+[OPTION] works as delete in many contexts. For example, it can be used in the load screen to delete whole songs. Some people use a sort of thumb roll to hit both keys. Tim aims one finger at the crack between the two keys. You can work out your own favourite method.

Now that your new song has a name, you can navigate to SAVE in Project View, and select it with [EDIT]. A screen will appear where you can select a save directory (folder). The default will be `/SONGS`, and that will be fine (you can move it later, if you wish, or more accurately, you can save it elsewhere and then delete the earlier version).

To cancel without saving at this point, tap [OPTION] (remember this has a meaning of ”no” or ”leave”). But to save, tap [EDIT] (meaning ”yes”).

Your song will be saved as something like `RANDOMNAME.M8S`. If you need to reload it later, or you want to go back to the demo song, you can select LOAD in Project View. But if you instead navigate down to Song View, you will be ready to start creating your first song.

If you’ve done any amount of work on a computer, you know the importance of periodically saving your work. The M8 has an autosave mechanism, but it can fail. There is no shortcut for saving a project while in other views. You must come back to Project View and save it. This becomes very quick after the first few times you do it. In what follows, we won’t be reminding you to save your work. You need to remember, and develop your own rhythm for saving. As of firmware 6.2, saving a new version of a song will move the previous version to the `/BACKUP` folder. This can be used to fix things if you somehow mess up and save a bad version, but you should remember to visit this folder periodically and clear it out.

2.6 Chains and phrases

You have a blank musical canvas in front of you. Tapping [PLAY] at this point will have no effect.



Recall that we said in the introduction that a track is a sequence of chains, a chain is a sequence of phrases, and a phrase holds sixteen possible notes.

The leftmost column holds row labels. The eight columns to its right are the sequence of chains for each track, with the track number at the top. The screen shows sixteen rows, labelled 00 through 0F, but if you navigate down to the bottom, it will scroll to reveal more rows. Rows can go up to FF, so there can be as many as 256 of them. For our first song, we’re not going to need to scroll.

Let’s put a chain reference in the first row of track 1, that is, row 00 (you’ll have to get used to ”first” meaning 00 sometimes). Navigate up there and tap [EDIT]. The entry 00 will appear. This is a reference to chain zero, which we’ll call chain 00 from now on. In general, tapping [EDIT] on an empty cell (with ”–” in it) will insert the last edited or deleted value.

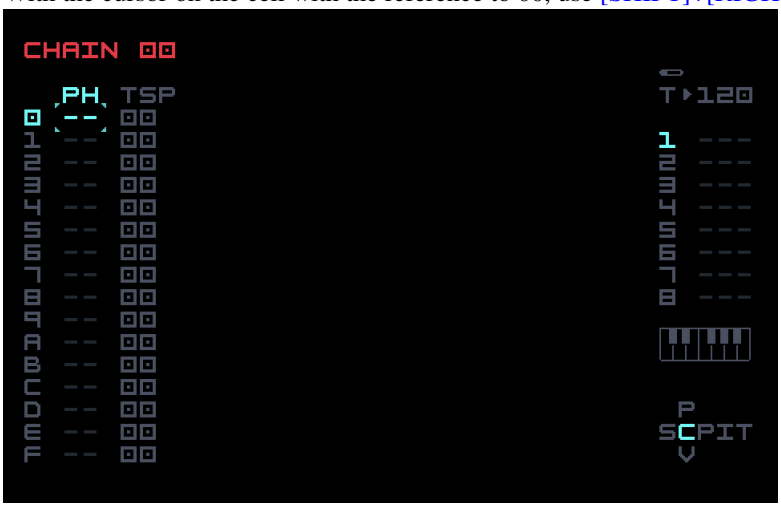
What if we want this entry to refer to another chain? To edit values, hold [EDIT] and use the arrow keys, possibly repeatedly. [LEFT] and [RIGHT] will result in a small increment or decrement respectively, in this case of 01. [UP] and [DOWN] will result in a larger increment or decrement respectively, in this case of 10 (in hex, that is, the left digit alone will change). Try changing the entry to something other than 00, and then delete it using [EDIT]+[OPTION]. Tapping [EDIT] will restore the value you deleted. Take it back to 00.

There is no general undo mechanism on the M8. What we just saw is a way of undoing a mistaken deletion. In some cases, large changes will be preceded with a warning, but at times you will have to recreate something that you messed up.

Before we work with chain 00, let’s take a brief look at the right side of the Song View screen. You already know about the mini-map at bottom right. At top right is a battery indicator and a tempo indicator (120 BPM in this case).

In between is a display of what notes are playing on each track, as well as a visual keyboard indication for the track whose column the cursor is on.

With the cursor on the cell with the reference to 00, use [SHIFT]+[RIGHT] to navigate to Chain View.

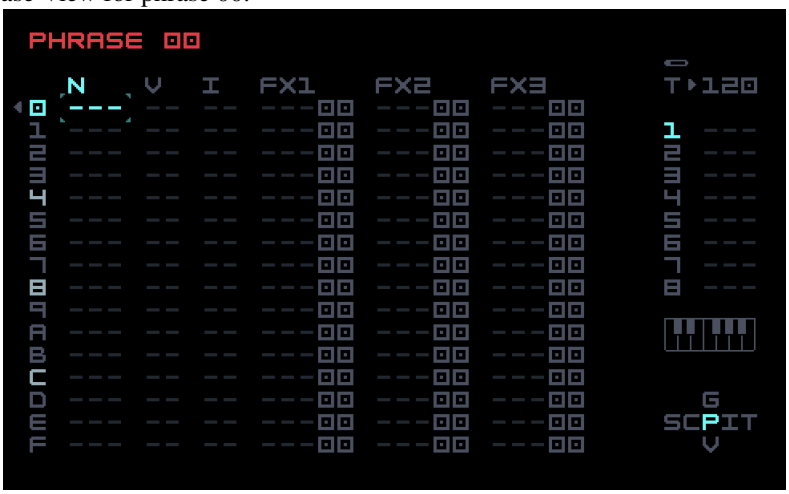


A chain is a mid-level musical organizational tool. Intuitively, it is a chunk representing a musical idea. Since a phrase has 16 note slots, and can be thought of as one bar (space for four quarter-notes in 4/4 time), a chain might be four bars or eight bars, as these are common musical groupings. But there are 16 rows in total, so a chain can contain up to 16 phrases. We're going to use four phrases in each chain for this song. The remaining twelve slots can be left alone, as the playback mechanism takes an empty slot as an indication to move to the next chain.

There are two columns. The one labelled PH contains references to phrases. The one labelled TSP contains a value used to transpose a phrase (move every note up by the specified amount). With your cursor on the first row, PH column, tap [EDIT] to add a reference in this cell to phrase 00.

Chains can contain any number of phrases from zero to sixteen, and, as we'll see, there are ways to end a phrase early or hop around in one. Each track has an independent playhead (position in its sequence of notes). But to tame the potential chaos, it can be advisable to make all of your chains the same length, by padding with empty phrases, which will have the effect of keeping the playheads synchronized. We'll do that throughout this tutorial. If you navigate away from the top corner, you'll see that the 00 is greyed out. That's because phrase 00 is empty. In general, empty phrases, regardless of their label, are greyed out in Chain View, as a useful visual aid.

We said that we were going to use four phrases per chain, so, using what you've learned, add a reference to phrase 00 in rows 01, 02, and 03. Then, with your cursor on the last thing you added, use [SHIFT]+[RIGHT] to move to Phrase View for phrase 00.



Now [PLAY] will do something! There still won't be any sound, but you will see a playhead indicator moving down just to the right of the row labels. When it reaches the bottom, it wraps around to the top and continues as before. To get sound, there must be at least one note in the phrase, and an assigned instrument.

At this point, we could describe note information and how to set up instruments, and you could go off and make your own music. You can still do that, but to add a little more structure to the process in this document, we're going to use a short song we designed as your first M8 song, and show you how to create it on your own device. You can hear that song at the start of the next chapter.

We haven't described all the possible views yet. More will be discussed in the proper context. But there is one view that isn't so easy to get to, has very limited and specific contexts in which it is relevant, and the manual doesn't discuss it. It's called the **boot menu**, and to see it, you have to start with the M8 off, and hold [SHIFT]+[OPTION] while pressing the power button.

This puts the M8 in a special state where various options can be changed. You should visit this at some point, just to familiarize yourself with what is possible. The options that might be immediately useful to you as a beginner affect key delay (the time before a key press is registered) and repeat rate (the time before a key held down is considered to be a repeat press, for example, when navigating down through a phrase by holding [DOWN] instead of pressing it repeatedly). If you've been having trouble with double hits, or you find repeats too slow, this is the place to make adjustments.

This is also the only place you will find a battery charge percentage listed (though it is only an estimation) which might provide reassurance if you are worried about the device charging properly. To leave the boot menu, you can just turn off the power, but if you have made any changes you want to retain, you should save them first.

2.7 To keep in mind

- Start internalizing the sense of [EDIT] as "yes" and [OPTION] as "no", and the use of [SHIFT] for changing between views. While there are a lot of key combos, there's good internal consistency, which helps to learn them.
- The key to muscle memory is practice. The more you use your M8, the quicker it becomes natural. Facility also seems to return quickly when you've been away from your device for a while.
- Hexadecimal can be intimidating, but once again, it will become natural with a bit of practice. You rarely need to figure out what a two-digit hex number is in decimal. More often, you will be moving values up and down without reference to specific magnitudes. In some situations, the two digits work independently to specify different things.

3 A first song

Here's the audio of the song you'll create. You can listen to it in your browser, or download it. It has a campy, retro feel, like the theme song for a spy thriller TV show that doesn't take itself too seriously. It is nowhere near as interesting as the M8 demo songs, or other demos of M8 you may have watched on social media, but it is short and simple enough for one to recreate it without too much effort, yet complex enough to hold our attention and to give a feel for what creating something longer and more complicated would be like. It also introduces several important techniques, certainly not all of them, and some others that are relatively minor in the grand scheme of things, but serve us well in the immediate context.

Download link: FIRST.mp3

Creating this just from listening to the sound could be a daunting task even for experienced M8 users. But you don't have to do that. We'll explain how to create it, step by step, and at the end, you'll have a full song on your machine that sounds like this. An M8 project file is available for download at the end of the chapter, but you will have it (or something equivalent to it) if you follow the instructions, and you'll understand it a lot more if you work through them.

3.1 Creating chains and phrases

The end of the previous chapter left you with one chain, 00, containing four phrases, all references to phrase 00. We're going to leave phrase 00 empty, making chain 00 four empty phrases (which still take up four bars). As discussed earlier, this will be useful for padding. Some people organize their labelling (for example, having all chains and phrases for track 3 have left digit 3) but you can decide later whether you want to do this.

Something you need to think about sooner is how to organize tracks. You have eight tracks, but you cannot label them; they are numbered 1 through 8. You just have to remember what each track is used for. Some sort of default can make this easier (lead melody and supporting elements to the left, percussion to the right, for example). For this first song, we'll tell you how to use tracks.

We'll start with the kick drum track, which will be track 6 (the idea is that tracks 6, 7, 8 will be for percussion). Navigate over to track 6 in the first row in Song Mode, which should be an empty cell. Don't do this, but you know that if you tap [EDIT], you will get the last chain label you entered or deleted, namely 00. You want a new chain to work with, and you can get that by double-tapping [EDIT], which in the manual is called [EDIT]+[EDIT]. That will get you 01. This key combination generally gets you a new object, but the mechanism is pretty simple; it just keeps counters and increments them. Consequently, if you interrupt the flow of this lesson to go off and try things on your own (and we certainly encourage you to do this) the numbering you get when you return to the lesson might not be the same as ours. It doesn't matter what labels are used, as long as they are used consistently.

With your cursor still on the chain label 01 that you just created, use [SHIFT]+[RIGHT] to navigate to Chain View for this chain. Chain 01 will be empty, with your cursor on the first row. Double-tapping [EDIT] will fill that cell with a reference to an unused phrase, phrase 01 in this case. We're going to make this chain four phrases long eventually, but we don't need to do that right away. Instead, use [SHIFT]+[RIGHT] to navigate to Phrase View for this phrase.

Your cursor will be in the leftmost cell of the first column, the one labelled N. N stands for "note". If you tap [EDIT], C-4 will appear. This is "middle C", the fourth occurrence of C on a full piano keyboard. The little keyboard visualization to the right of this view shows a single octave on such a keyboard, starting with C. The keys that would be white on a piano (but are black in this reverse-video view) are, from left to right, C, D, E, F, G, A, B. The two black keys at top left are C# and D#; the three at top right are F#, G#, and A#.

With the cursor remaining on this cell, [EDIT]+[RIGHT] will change it to C#-4 (displayed as `C#4` on the M8, to fit a three-character column), that is, raise it by one semitone. [EDIT]+[LEFT] will lower it back down to C-4 again. [EDIT]+[UP] will raise it by an octave to C-5, and [EDIT]+[DOWN] will lower it again. These key combos have the same small/large effect in many other settings, and are important to memorize.

The next cell to the right, in the V column, was automatically filled in with 64. V can be thought of as "volume", but since there are so many types of volume, this is commonly referred to as **velocity** (a reference to the speed at which one strikes a piano key or drum, which in turn affects the volume of the sound). The velocity range is 00-7F (so that negative automation can turn it all the way off).

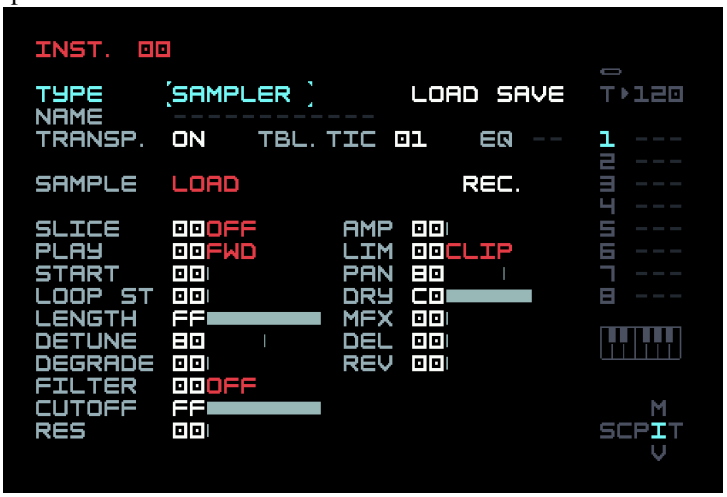
Though we have added a note to the phrase, playing it will still produce no sound, because we have not specified an instrument. This is done in the third column, the one labelled I at the top.

Navigate to the I column of the first row and tap [EDIT] to add a reference to new instrument 00. Then tap [SHIFT]+[RIGHT] to navigate to Instrument View for this instrument. The screen will look like this:

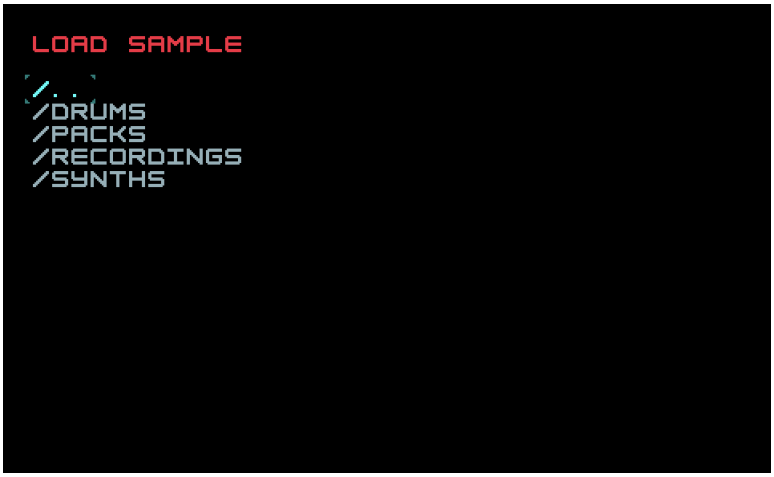


To the right of the cursor in the first row, you will see the words LOAD and SAVE. LOAD will let you browse instrument presets. There are a lot in the factory content, but we're going to make our own. SAVE can save the current instrument as a preset so that it can be used in other songs, but you have to name it first. Within a song, we refer to an instrument by its number, so we won't bother naming our instruments for now. The numbered instruments in a song are saved with the project. If an instrument preset is loaded, it is a copy, and the same thing is true if an instrument is saved as a preset. That is, the saved instrument presets and their possible copies in various projects are independent; a later change in one will not be reflected in the other.

With the cursor where it is, on NONE to begin with, you can use [EDIT]+[RIGHT] to start navigating through instrument types. The cursor won't move, but the field under it will change to show different text, and the screen below will reflect the ways to interact with the chosen type. [EDIT]+[LEFT] will back up in the list of types. Find the Sampler instrument.

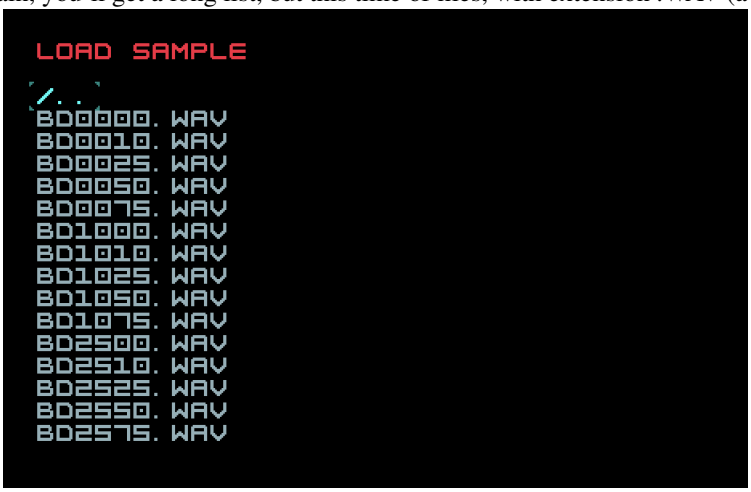


We will need to load a sample. This is done on the line labelled SAMPLE, with choices LOAD and REC. REC will record a sample, but we're going to use one of the factory samples. So navigate to LOAD and tap [EDIT]. You will see a browsing screen that looks like this:



It doesn't say so, but you are in the SAMPLES folder. Remember that /.. (which the cursor is on) means "parent" or "enclosing folder"; if you choose that, you will be at the root folder of the SD card's file system. It's still only showing you things relevant to samples. Your song project lives in this folder, but you don't see it in this view. With your cursor on SAMPLES, tap [EDIT] to descend back into it. (At any point, you can tap [OPTION] to say "no" and go back to Instrument View without loading a sample. This would be useful if you change your mind about replacing a sample already loaded into the instrument.) Then go into DRUMS. You will see folders called HITS and SLICES. Go into HITS. This contains a number of folders, one called DCM8, and the rest starting with TR. These are all samples of various drum machines. There are no acoustic drum samples in the factory content, as far as we know.

The Roland TR-808 is an iconic drum machine that has been used on many songs that you've heard, whether or not you knew it at the time. Choose that folder. You will see a long list of subfolders (SD at the bottom of the list is off the screen). These are abbreviations for types of percussion; SD means "snare drum". But choose BD, the bass drum. Again, you'll get a long list, but this time of files, with extension .WAV (a type of audio file).



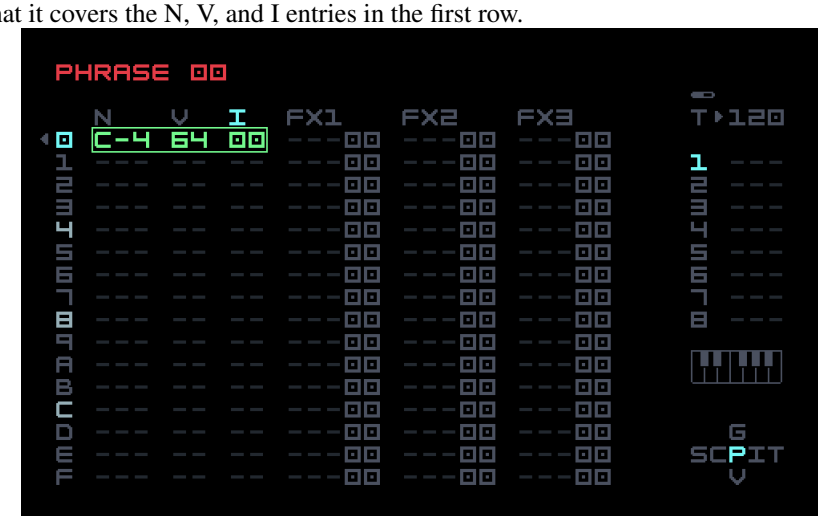
These are the bass drum samples. There are a lot (again the list goes off the screen) because there were two parameters controlling "tone" and "decay" on the TR-808, and it has been sampled at various settings. You can put the cursor on one and hold [PLAY], and you will hear the sample playing (this works for instrument browsing as well). Choose BD2525.WAV, and tap [EDIT] to confirm your choice. You will be back at Instrument View with your cursor on the name of the sample you just chose. Tapping [EDIT] again will return you to the folder where that sample lives, if you want to make another choice. But right now, you don't.

Navigate back to Phrase View. Now, finally, you can hear some sound when you press [PLAY]. Doing this in Phrase View will loop only the phrase being viewed, and play only the sounds in that phrase. Pressing [SHIFT]+[PLAY] will play all tracks of the song, starting from this phrase, while keeping you in Phrase View. Right now this is the only

sound in the whole song, and this phrase is the first one on its track, so this doesn't make a difference yet. The note value of this sound, C-4, happens to play the sample exactly as recorded (no alteration of its speed to change the pitch). So we'll keep using that note.

Before we start on the bass drum sounds for our target song, let's take a short diversion to learn some more useful editing commands while making a "four on the floor" pattern. This is just four equally-spaced hits in one bar, so in rows 0, 4, 8, and C. These row labels are even highlighted to point this out. You know enough to be able to put in the other three hits using just arrow keys and [EDIT], but we will explore some other methods of achieving the same result.

With your cursor on that single C-4, tap [SHIFT]+[OPTION]. The entry will be surrounded by a green rectangle. This is **selection mode**. You can extend or shrink the selection in either direction with the arrow keys. Extend it so that it covers the N, V, and I entries in the first row.



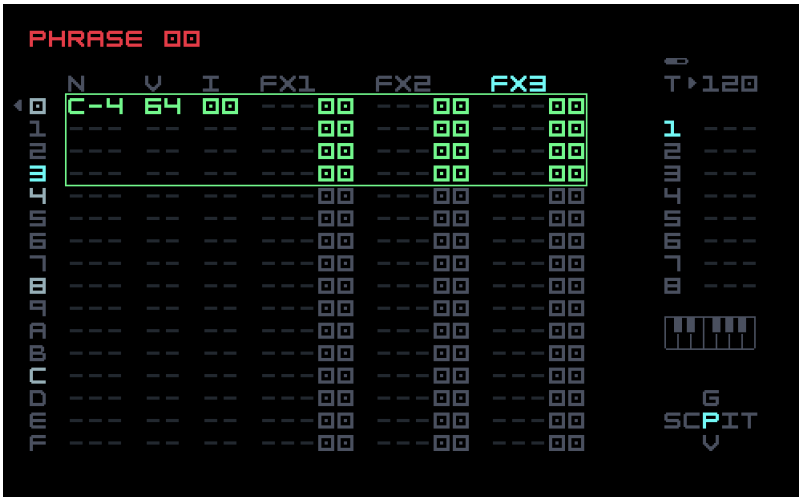
Then, to leave selection mode, tap [OPTION]. The green rectangle will disappear, but the selection (the three cells in the first row) will have been copied into an internal clipboard. (A message at the bottom reminds you of this.)

To paste the contents of that clipboard, you can use [SHIFT]+[EDIT]. So to create the second hit, navigate to the first cell in row 4, and tap [SHIFT]+[EDIT]. You now know a faster way of creating rows 8 and C, but we can do even better.

Let's get rid of the row 4 we just created. You know how to select it, but here's a faster way. Hold [SHIFT] and keep holding it. Tap [OPTION], and you will go into selection mode. Tap [OPTION] again, and the whole column will be selected! That's cool, but we don't want to select the whole column. Tapping [OPTION] once more will select the whole row, including the FX cells we haven't talked about yet. That's what we want, but tapping [OPTION] once more would select the entire set of cells, and once more just leaves selection mode without doing anything (remember this, because it preserves your clipboard as well).

But with the whole row selected, let go of [SHIFT], and tap [EDIT]+[OPTION], the delete combo. The whole row will be deleted. Actually, it will be cut, which means removed from the screen but copied into the internal clipboard. So here's another place where undo is available to us. If we have deleted a selection, we can just paste it back if we change our minds.

Now we just have row 0 as we want it. Go back up to row 0, use your new fast method to select the whole row while holding [SHIFT] down, then let go of [SHIFT]. Tap the [DOWN] key three times, and the selection will extend to the first four rows.



Tap [OPTION] to leave selection mode (and copy those rows to the clipboard). Where is your cursor? On the first cell of row 4, exactly where you can paste the clipboard (with [SHIFT]+[EDIT]) to fill in row 4 and empty rows 5 through 7. If you did this, the cursor would then be on row 8, ready for another paste. So just two more pastes will finish the job. But don't do this. Instead, go up and delete row 4, then move your cursor to the note on row 0.

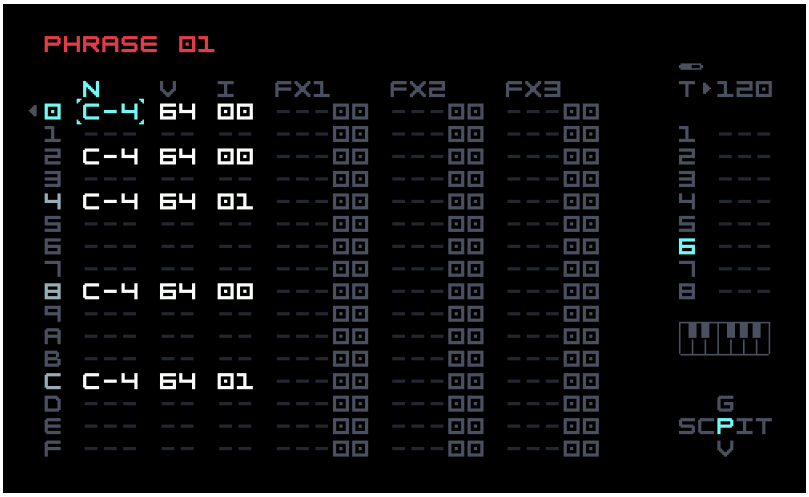
Select the whole column, then hold [OPTION] (keep holding it) and tap [LEFT]. All the notes will get filled in, with the V and I values from the first row. We don't want all of them filled in, though; we want every fourth row. Tap [LEFT] again, and every second row will be filled in. Once more will fill in every third row, and once more will get us to every fourth row. (Don't do this, but if you tap [RIGHT], you get a random selection filled in, and tapping [RIGHT] again changes to another random selection.)

You can now leave selection mode by tapping [OPTION]. Our song doesn't use four on the floor, so we need to get rid of most of this. In the case of a fill operation, the clipboard contains the previous contents of what was filled over. So if we do a paste, we'll be back to just row 0 filled in. This ends our mini-lesson on selection mode.

The first bar of our drum pattern has kicks on 0, 2, and 8. Put those in and tap [PLAY] to listen to them. It also has snare hits on 4 and C. Since those don't overlap, we could put them in the same track, or we could put them on track 7. Some advantages of using one track are that we save a track for something else, and all the information is on one screen. Some disadvantages are that it might be too much information to take in at a glance, we might want simultaneous hits later on, and we can't mute kick or snare separately while performing, only together. For now, let's put them in the same track. We'll leave track 7 open if you want to move them later.

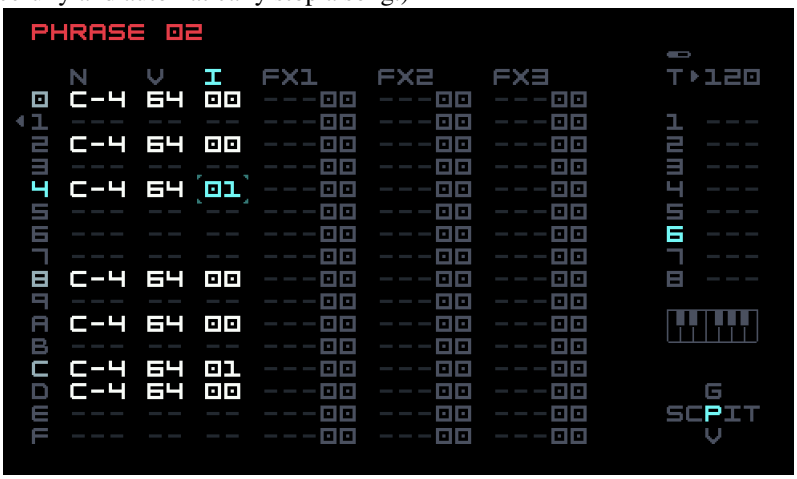
Move the cursor to the N column in row 4 and tap [EDIT], which will create a new kick hit here. We want a snare hit, and we don't have a snare instrument yet. You could go over to the I column and tap [EDIT]+[EDIT] to get a reference to instrument 01, and then set that up from scratch. But instead, we're going to make a copy of instrument 00 and put it in instrument 01. This is called **cloning**. The key combination for this (which works in many other contexts, as we'll see) is holding [SHIFT] and tapping [OPTION], then while still holding [SHIFT], tapping [EDIT]. The manual uses [SHIFT]+[OPTION, then EDIT] to refer to this action. This is moderately complicated, but it doesn't get much worse than this, so if you can internalize the cloning action, you can consider that a milestone achievement.

Go to the I column in row 4, cursor on 00, and clone it into 01. Then navigate to Instrument View with [SHIFT]+[RIGHT], and you will be in instrument 01 set up as a bass drum sample. Change the sample to a snare drum sample (go to the name of the sample, tap [EDIT] to bring up the sample browser, go up to the parent folder and then down into the SD folder, which won't be visible on the screen until you scroll down to it). Choose SD0050.WAV. Now go back to Phrase View and add the snare hit on row C. Listen to the result.



Cloning is useful for making a copy with slight variations. Because we didn't do much to set up our bass drum beyond choose a sample, it didn't save us much time in making the snare. But in general, an instrument has many specific settings for parameters, and cloning will copy all of those, saving more time.

Cloning also works for chains and phrases. We're going to use it to fill in the other phrases in our four-bar drum pattern. The third phrase is the same as the first, but the second is a variation, and the fourth is another variation. Go back to Chain View, navigate to the second phrase, tap [EDIT] to get a reference to 01, and clone it into 02 with [SHIFT]+[OPTION, then EDIT]. Then navigate to Phrase View, and you will be in phrase 02, which will look exactly like phrase 01. The variation here is kick hits on A and D, so add those, and listen to the result with [PLAY]. You can navigate among phrases in a chain without leaving Phrase View, using [OPTION]+[UP] and [OPTION]+[DOWN]. At this point, the whole song is a two-bar drum pattern, and [SHIFT]+[PLAY] will loop it. (We'll later learn how to gracefully and automatically stop a song.)

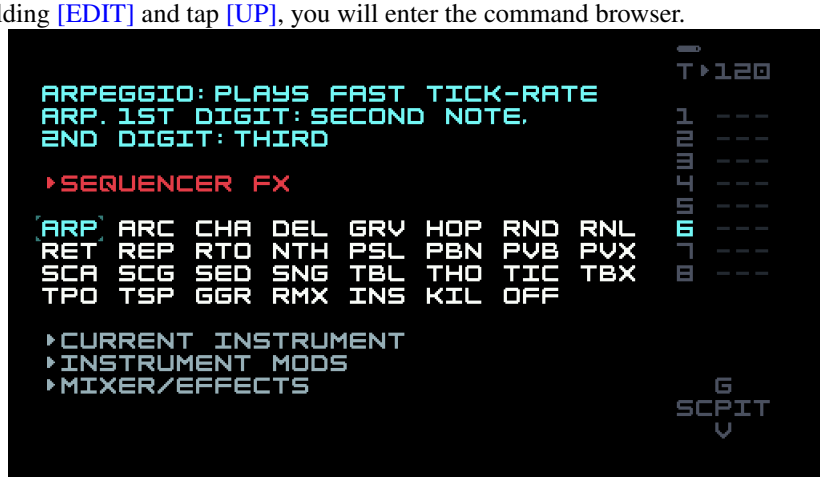


To put in the third and fourth bars, navigate to Chain View and row 2, enter 01 there, then go to row 3, enter 02, and clone it into 03. The difference between phrase 02 and what phrase 03 should be is this: the kick on 8 moves down to 9, and the kick on D moves to E. There's a new snare on 7.

But that isn't everything. There are extra snare hits in the audio, and they seem to be between steps. This is achieved with the help of the three pairs of columns that take up the right half of Phrase View. These are labelled FX1, FX2, FX3. FX is short for "effects", but to avoid confusion, we will always use the abbreviation FX in this context, and never for the send effects such as delay or reverb. Each column holds a **command** and an associated value.

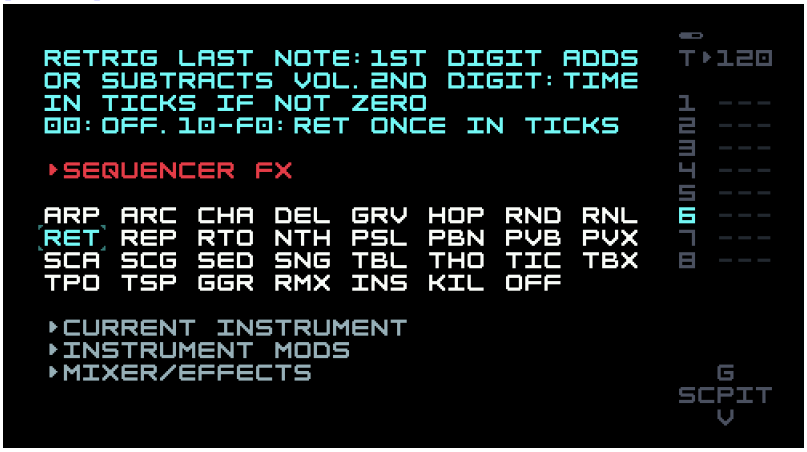
The commands for the extra quick hit need to be on row C. Move the cursor to the FX1 column in this row, so that it surrounds the three hyphens next to the greyed-out 00. Hold [EDIT]. That adds a default command (ARP in this case) and a default value next to it (00, not greyed-out any more). This is not the command we want, so if you keep

holding [EDIT] and tap [UP], you will enter the command browser.



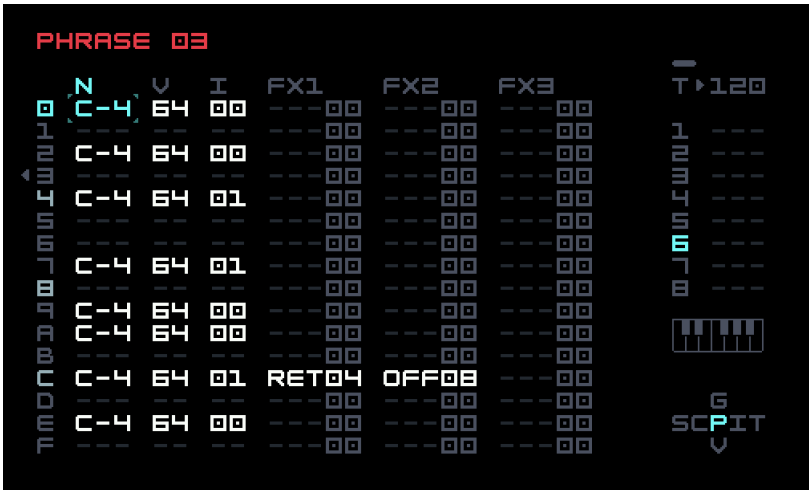
As of firmware 4.0.0, the commands are organized into four pages, because there are so many of them. The command we want, RET, is on the first page, just below ARP. But the cursor is on ARP in the top left corner, and you can see help text for the command, telling us in the first word that ARP stands for "arpeggio". We're not going to use this command yet, but you can see the word "tick" in the description.

A **tick** is an even smaller unit of time than a sixteenth-note step. There are six ticks per step. Still holding [EDIT], tap [DOWN] to move the cursor onto RET.



The help text tells us that RET stands for "retrigger". The first digit adjusts the volume change on each retrigger; we'll leave that at 0. The second digit specifies the number of ticks between retriggers, and we'll change that. Letting go of [EDIT] will leave the command browser and go back to Phrase View, with the chosen command filled in. We used RET 04 for the audio, but try other values. You will notice a problem: even with our value, there are way too many extra hits.

The reason for this has to do with how the length of notes is calculated. Each note in our phrase lasts until the next note entry. This didn't matter before, because we are playing one-shot samples to completion. To alter this, we can use the OFF command. Putting OFF 08 into the FX2 columns of row A will mean that the snare hit in this step will become two hits, the first starting on the beat and lasting for four ticks, the second lasting for four more ticks. It's also possible to put an OFF command (with no associated value, that is, it takes effect immediately) into the note column by using the delete key combo on an empty slot. Here's what the final view of phrase 03 looks like.



Another way of achieving this effect would be to put a hit on row B, but delay it with the DEL command. One can do a lot with FX commands, and even more power is provided by the table mechanism, which is a very general extension of these ideas that we'll discuss in a subsequent chapter.

Let's quickly add a hi-hat track. Navigate back to Song View, move to track 8 in the first row (we'll leave track 7 for future independent snares or other uses), and add a reference to new chain 02. Navigate to Chain View, add a reference to new phrase 04, and put in three more identical references to preserve our uniform four-bar chains.

Then navigate to Phrase View, add a reference to new instrument 03, and navigate to Instrument View to set it up. Choose Sampler type, and fill the sample slot with a 606 hi-hat with medium decay (sample name 606MHAT.WAV). Go back to Phrase View, set the velocity of the first note to 24, and then fill the remaining rows with the same values. In these two brief paragraphs, we haven't mentioned any keystrokes, so this is a good test of how well you've learned the navigation and editing commands mentioned above.

The hi-hat track can sound a bit robotic by itself, but it's a subtle element in the final track, so it won't be noticeable. There are ways to make it more interesting. One could put in a more varied pattern, or use probability to drop out the occasional hit or vary the velocity of each hit. One could also delay some hits slightly. We'll see ways of doing all of these things shortly.

We've seen that tapping [PLAY] when in Phrase View will loop just that phrase. In Chain View and Song View, it will start playing from the cursor position (and loop the chain in Chain View). There are also key combos to mute and solo the current track which you can learn about in the M8 manual. One thing that is not mentioned in the manual yet is that when in selection mode, tapping [PLAY] will loop the selection for just this track, and [SHIFT]+[PLAY] will loop the selection but play all tracks. There are thus many ways to focus in on a small part of a larger work under development and listen to just that part while tweaking it. In looping a restricted region, you need to keep in mind that any effect whose scope extends outside that region could sound different. This is obvious for, say, a note that would keep playing outside the region. Other changes might be more subtle and possibly confusing.

We haven't completed our song! We only have the drum pattern, and it's only four bars long and not very sophisticated (still, we will leave it like this). Before we add the melodic elements, let's learn a bit more about instruments.

3.2 General parameters in Instrument View

All the instrument types, except MIDI out, have a section at the top in Instrument View with information specific to the type, and two columns of parameters below, with most of the left column also specific to the type.

The common area at the bottom of the left column describes a multi-mode filter. You choose a filter type and specify values of two parameters, cutoff frequency and resonance. Resonance is a property of hardware filters that allows control of the amount of a volume boost at the cutoff frequency. This can be musically pleasing, especially if either or both of the parameters are modulated, so it is emulated here in software.

The filter types available are lowpass (cutting off higher frequencies), highpass (cutting off lower frequencies), bandpass (cutting off both), bandstop (cutting out frequencies around the cutoff, so the opposite of bandpass), and

LP>HP (like bandpass, but resonance controls the HP cutoff frequency, so the width of the range of preserved frequencies can be adjusted; this is called a base-width filter on Elektron devices). They all have musical purposes that we'll explain as they arise.

The common right column in Instrument View is for effects. The signal path can be thought of as two amplifiers with a limiter in between, after which selected amounts of the result are sent to a single global "mod effect", delay, and reverb. The mod effect defaults to chorus, but optionally could be a phaser, flanger, or comb filter.

The AMP parameter allows a boost of the signal before the limiter. This seems paradoxical, but it is used to create distortion or overdrive. The simplest form of limiting is clipping, which is like shaving off the top of the waveform when it exceeds a set value. The other types of limiter are described in the M8 manual; they can sound considerably different, and it is worth experimenting with them early on to know what is available. The DRY parameter then allows the signal thus distorted to be lowered again.

There is also a PAN parameter here that allows placement of the monophonic sound in the stereo field, and three parameters for the send amounts to the global effects. The parameters of the global effects can be set in Effects Settings View, which is under Mixer View.

3.3 Synth instrument types

The M8 manual has detailed descriptions of parameters for all synthesizer types, but it assumes some prior experience with synthesis. We'll give a more introductory, high-level overview of the types, and leave the details to the manual. Your own experimentation will give you a sense of where and why you might use each type. You can adjust parameters while a song is playing, and the oscilloscope at the top of several views (including Instrument View) helps to understand what is going on.

Wavsynth provides a deliberately degraded "8-bit" sound to emulate the music and sound effects of early handheld games. That is, the waveform is coarse in resolution, despite the much better resolution possible on the M8. Not surprisingly, wavsynth can be used for chiptune, though this is not its only use. Bit reduction is often prized as an effect in other synths, though here the amount of bit reduction is fixed and cannot be varied or modulated. Wavsynth provides several basic waveforms, and the 4.0 firmware release added many more wavetables (within the shape formerly known as "overflow") that remain undocumented but available for exploration.

Macrosynth is a port of the open-source code for the Eurorack module Braids, by Mutable Instruments. MI had a considerable influence on the Eurorack community, not only for the variety and usability of their modules, but because nearly all of the software is open-source, so it shows up in many places, including some very expensive synthesizers.

Macrosynth provides no fewer than 44 engines, ranging from basic waveforms to variations on them (such as a swarm of saw waves), through to physical models of plucked, blown, bowed, and struck instruments, various types of percussion, and various types of noise. The prefix "macro-" means "large" in Greek, and its use here is used to refer to the design of the interface, where each engine has two parameters, generically called Timbre and Color, that under the hood control more specialized parameters of the given model in a fashion that is not documented and not open to change by the user. The tradeoff is that the user does not have to understand the effects of cumulatively hundreds of parameters, at the cost of some loss of control. This is a common approach in many Mutable Instruments modules, and their popularity is a testimonial to the effectiveness of the approach. Macrosynth will reward extended study.

FM Synth also rewards extended study, but here all the parameters are available. FM stands for "frequency modulation", meaning one oscillator (repeating waveform) controlling the frequency of another. In this context, the oscillators are called **operators**. The mathematics of sound treats repeating waveforms as the sum of sine waves, a sine looking like a triangle made curvy rather than straight. This is discussed in detail in a sister flânerie, The Sound of Math, but all you really need to know is that FM is easiest to control when the oscillators are sines, and sounds best when the frequencies of controller (modulator) and controlled (carrier) are related by small integer multiples or ratios.

FM Synth provides four operators, and "algorithms" that hook them up in various carrier-modulator combinations. You don't have to stick to sines; other basic waveforms are available, and one algorithm is just the four operators with no cross-modulation, meaning you can use it for limited polyphony (multiple voices forming a chord). This is by far the deepest of the instrument types, and we could write a whole flânerie just on it (though here we will stick to a single later chapter), but you shouldn't be intimidated by the potential complexity. It is possible to use it in simple ways (just one operator, or one carrier and one modulator) to get very nice sounds, and there are recipes and presets for more

complex sounds. FM excels at glassy or bell-like sounds, but is also capable of reasonable approximations to many physical instruments. It will work well for long, evolving sounds ("pads").

One of the best FM synth devices available is the Elektron Digitone (the Digitone II was recently released). Simon "Ess" Mattison, who worked on the Digitone, and now makes software synths under the label Fors, is a big fan of the M8, and advised Tim on the design of FM Synth (though it also draws from other sources, such as classic Yamaha devices). FM Synth can't do everything the Digitone can, but it provides a respectable chunk of it.

Hypersynth is the most recent synthesizer type, being added in firmware release 3.0. It is a "supersaw", offering six saw wave pairs, where each pair can be detuned slightly with a parameter (providing natural variation over time), and pairs can be offset in pitch (forming chords) with user-defined intervals (sixteen such chords can be defined per instrument). This is the easiest way to add polyphony in M8, but the sonic variation is limited. Saw waves are rich in harmonic overtones, resulting in a "buzzy" sound, and several of them can sound like a swarm of bees. The filter options are definitely helpful here, as restricting the frequency range can result in a much more versatile sound. In firmware 6.6, a few more waveform options were added, not as many as Macrosynth or even FM Synth offers, but ones that work well with the detuning and chord capabilities.

3.4 The sampler instrument type

Just as FM Synth provides a respectable chunk of the Elektron Digitone, the Sampler instrument type brings to the M8 a good portion of the Elektron Digitakt, a highly-regarded sample-based device. Sampler does not have all the bells and whistles of Digitakt or other popular samplers. It does not have dozens of "mangling" effects, and one cannot "timestretch", or play a sample slower while preserving its original pitch, though there are low-fi techniques to approximate this that can be effective. But it has a good set of capabilities, and some say it has the best sampling workflow available.

A sample can be played forward or backward, one-shot or looped, pitched up or down (by changing the speed, so a sample pitched one octave higher plays in half the time), and looped without reset on a new note, so a very short sample can be used as a waveform to define an oscillator. One can speak or sing briefly into the microphone and extract several oscillators from the sample to build a song whose sounds do not betray their origin. In a process generally known as **resampling**, a selection in Song View can be rendered to a WAV file that can immediately be used as a sample. There is a Render View for more complex rendering.

The start and length of a sample can be modulated. Sample choice is fixed per instrument (cannot be modulated), but several samples can be combined into one (there are external editors and browser-based tools to facilitate this) and then sliced on the M8. One can slice a sample at regular intervals, or one can manipulate slice points, trim, and perform other actions in a sample editor on the M8. The editor allows arbitrary zoom-in at any time. On the Digitakt, zoom-in is only available at sampling time, and never for imported samples. Digitakt users would kill for this facility. Once sliced, slices can be selected for playback using note values (this is handy for a multisample of an instrument) or addressed using commands (to make breakbeats from drum loops, or to build kits from different sounds).

The Sampler instrument type offers so much sonic capability that even if the M8 had no synth types, it would still be a sought-after device. Many samples are available on the Internet free or at low cost. Your challenge will be to not go wild and fill up your device with huge folders that you have to scroll through and preview. There is some factory content to get you started, both percussive and melodic. Note that this content differs on M:01 and M:02; there are no included drum loops for you to slice on M:02, for example.

We'll go into more detail on the Sampler instrument in chapter 5.

3.5 Adding the bass

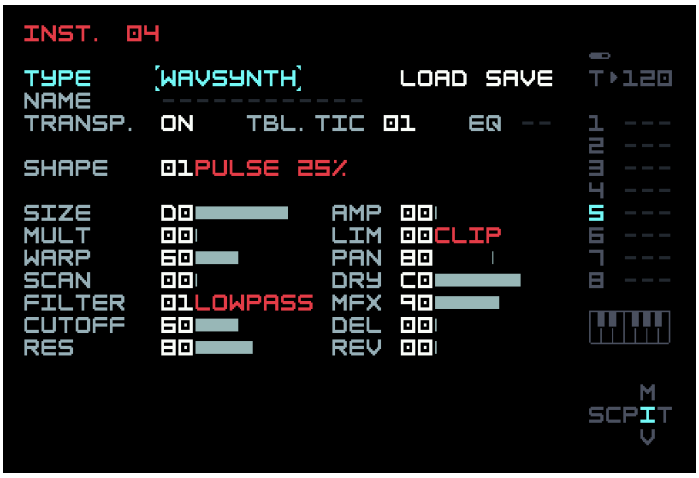
Since the song player has probably scrolled off your screen, and you might want to refer to it as we work on the song some more, here's another copy of it.

We'll use the Wavsynth on track 5 for the bass. Yes, the Wavsynth has a lo-fi sound, but we're going to be using a pulse wave, which only has two levels, so Wavsynth sounds as good as a more precise synth engine would.

If you count carefully from the beginning of the song, you will hear that the first three notes of the bass come in at the end of the fourth bar. They are really a leadup to the second four-bar chunk. This is informally called a **pickup**

(the formal term is **anacrusis**). We will need a whole chain just for these three notes, and another one for the four-bar chunk that follows them.

In Song View, put a reference to a new chain 03 on row 00 of track 5. Navigate to Chain View, and put in three empty phrases (phrase 00), followed by a reference to a new phrase 05 in row 3. With your cursor on the 05, navigate to Phrase View. The last three eighth notes of this bar are rows A, C, E. Put a note on row A (it doesn't matter what it is, we will change it later), move to the instrument on this row, and make it a reference to new instrument 04. Then navigate to Instrument View for this instrument. Choose Wavsynth, and set the Shape, Size, Warp, and MFX (chorus) parameters as shown in this view.



These are arbitrary choices, so you can change them to your taste. You can preview the sound with [EDIT]+[PLAY]. But it won't sound like it does in the final song, because that has modulation on the velocity and the filter frequency cutoff. Set the Filter, Cutoff, and Res (resonance) parameters as shown, and then navigate upward to the Instrument Modulation View, which we haven't previously seen.



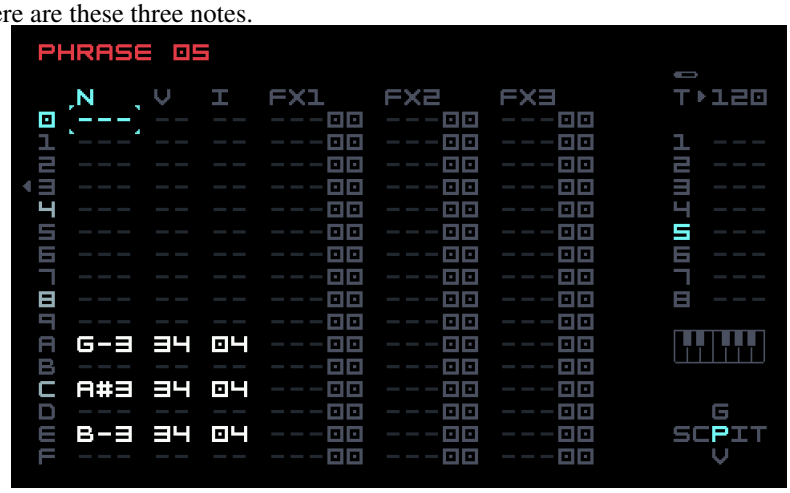
With earlier firmware, each instrument had two LFOs (low-frequency oscillators; imagine twisting a knob back and forth slowly) and two envelopes (roughly, a contour that rises from zero and falls back down). But more recent firmware generalizes this to four modulators, each of which could either be an LFO or an envelope (and there are further choices within each of those categories). However, the defaults are still like the old two-and-two arrangement. Since we are going to use two envelopes, we might as well use the defaults for mod sources 1 and 2.

The envelope modulators default to AHD, which stands for "attack, hold, decay". This is an envelope that rises to its maximum value at a rate set by the ATK parameter, holds for the time specified by the HLD parameter, and then decays at the rate set by the DEC parameter. We are going to keep HLD at zero, so it is just a AD envelope. You can see the required settings for mod source 1, modulating volume, but try other settings to hear what they sound like (the preview key combo [EDIT]+[PLAY] works here as well). With this modulation active, the sound will no longer drone

on at the same volume until you lift your fingers (with preview) or until another note is played (in a phrase). The sound will diminish as we would expect from an acoustic or electric bass.

Mod source 2 will affect filter cutoff, which mutes the higher frequencies of the sound before lower ones. Putting an envelope on this (with the possibility of different parameter values than for the volume envelope) shapes the sound further, with the added dimension of the moving resonance boost. Once again, try various combinations, and go back to Instrument View and change the cutoff and resonance there to see how all these parameters interact. The ones in the screenshot are an arbitrary choice that sounded good. All of these can be changed on a per-note basis with FX commands.

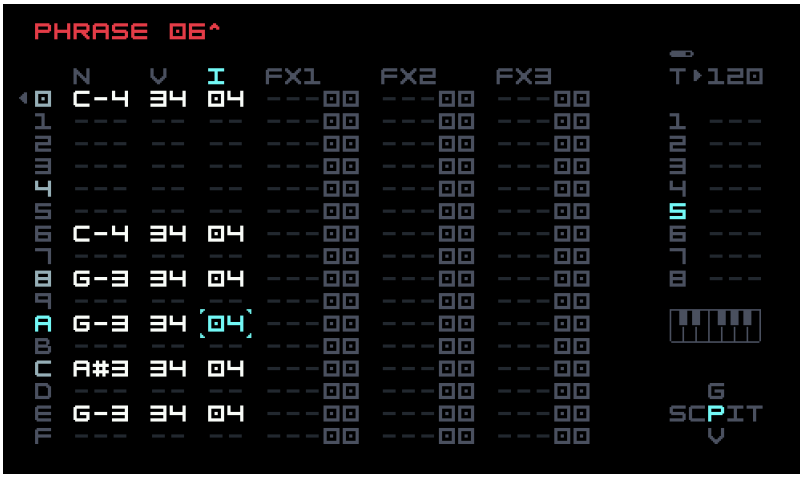
Now go back to Phrase View, and remember phrase 05 is for the three note pickup in the fourth bar of chain 03. Here are these three notes.



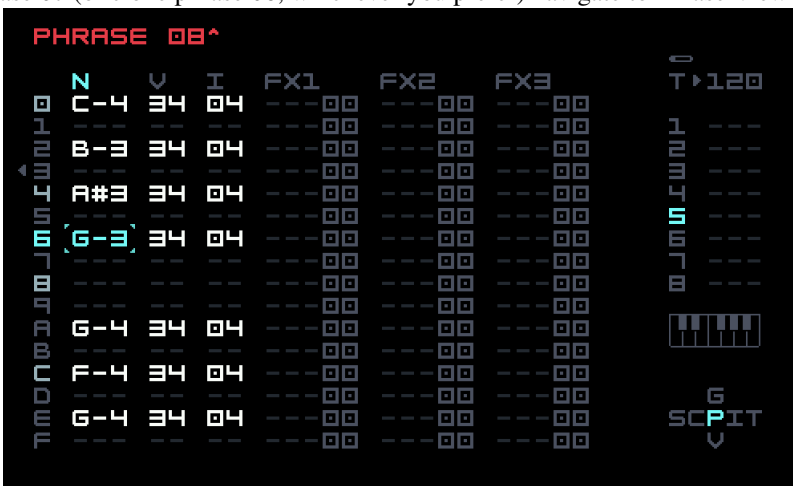
If you don't have the ability to take a melody in your head and translate it into note names and octaves (we're not very good at this), you will have to sculpt your melodies on the M8 making liberal use of the ability to loop a phrase or to start it playing at the cursor position. It is not the same as picking out a melody on keys, but it can be almost as fast, if you're not proficient on a musical keyboard. If you have the time and inclination, you can try to listen to the recording of the song, and see if you can recreate the bassline by yourself (or do better).

Recall that in selection mode, holding [OPTION] and then pressing [LEFT] cycles through various fill modes. If you instead press [UP], the notes in the section will be "randomized upward", and similarly for [DOWN], while [RIGHT] will randomize both the triggers (rows on which notes appear) and the note values. If you don't have a specific melody in mind, you could use this random choice as a starting point, tweaking it into shape. But we didn't use this method here.

Once you have the pickup set in phrase 05 in chain 03, go back to Song View and put a reference to a new chain 04 in row 01 of the bass track 5. Here a one-bar phrase starts which is used a lot in the bassline. Navigating to Chain View for chain 04, add a reference to new phrase 06 in row 0, and navigate to Phrase View for it. Here is what it looks like.



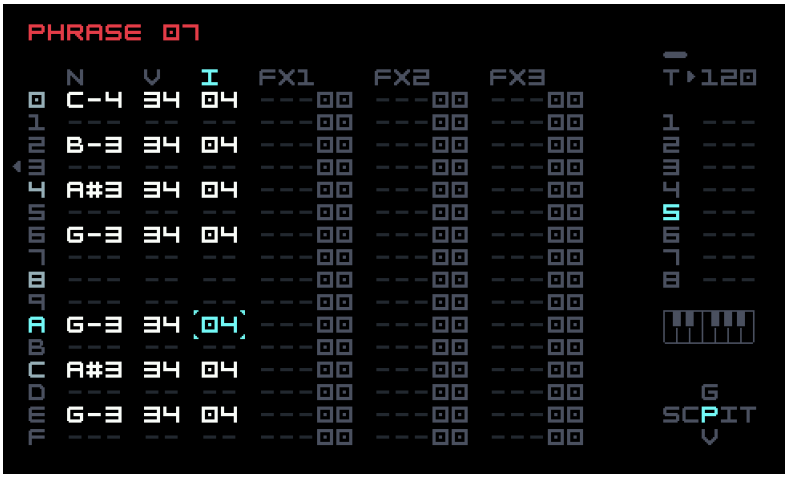
This phrase is used in the first three bars of chain 04 (so you should add references to it in rows 1 and 2), but the fourth bar starts differently, and ends with a new pickup. So add a reference in the fourth row of chain 04 to new phrase 07 (or clone phrase 06, whichever you prefer) navigate to Phrase View for phrase 07, and set it up.



The percussion tracks 6 and 8 repeat their four-bar chain, so we could just leave them as is. But, to maintain the discipline of having the playheads for each track all on the same row in Song View while the song is playing, we should extend them downward by repeating their chain references. This will also let us end the song cleanly, once we set out to do that.

So far, we have two rows of our song completed (rows 00 and 01), using three tracks and four chains, and since we are using chains with four rows, we have eight bars of the song done. Now we will work on the third row of the song (row 02), that is, the ninth through twelfth bars.

The pad comes in at the beginning of the third song row. A pad is a long sustained sound, often with gentle attack and decay. We will set the pad up shortly, but we are still working on the bass, and in these four bars that correspond to row 02, it sounds the same for the first three, but has yet another pickup in the fourth. That similarity makes it easy to set up. Put a reference to chain 04 in song row 02 of bass track 5, clone it into a copied chain 05, navigate to Chain View for the new chain, clone a new fourth bar phrase 08 in it, navigate to Phrase View, and change the last three notes.



Since we have nearly completed all the bass variations, we can extend them to the full length of the song, almost. To do this, navigate back to Song View, and copy the second row (row 01), then paste it onto rows 03, 04, 05, 06. The actual row 05 is different in the final song, but we will change it later, after we've worked on the pad and pluck.

We won't use them now, but there are two extended versions of cloning you should keep in mind. If, in Song View, you use the clone action [SHIFT]+[OPTION, then EDIT] on a chain reference, it will create a new chain which will contain all the phrase references of the cloned chain. But if you make the ending tap of [EDIT] a double tap, it will do a "deep clone", which clones all the phrases as well. This is useful if you want to make changes in those phrases without it affecting the chain that was cloned. As of firmware 6.6.1, making the ending tap a triple tap will do what we might call "deep deep clone", which clones the chain, all phrases within it, and all instruments referenced by those phrases, for even more independence from the cloned chain.

We are now ready to work on the end of the song, at the beginning of song row 07. This is a single bass note, slightly louder, and with more sustain. So go down to row 07 in the bass track 5 and add a reference to a new chain. By mistake, we numbered this 07 (there is no chain 06 in our arrangement). Within the new chain (whether you number it 06 or 07), we want a single new phrase 09 in the first row (no others are needed). Navigate to Phrase View for this phrase, and put in that single note, a C-4, velocity 44. To let it ring out, we change the envelope decay with an FX command, DE170 in the FX1 column (that is, changing the decay of envelope 1 to 70). Then, to stop this track to end the song, put HOPFF in the FX1 column of the last row (row F).

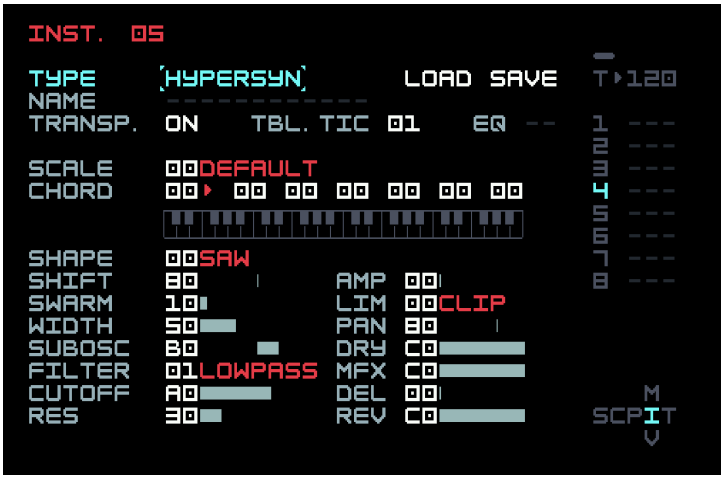
We want to similarly stop the percussion tracks, but we don't need a final note for them. So in one of them, create a reference to a new chain 08, and in it, a reference to a new phrase 0A, and in that, a HOPFF in the FX column of the first row. Then use that chain in song row 07 of tracks 6 and 8.

3.6 Adding the pad and pluck

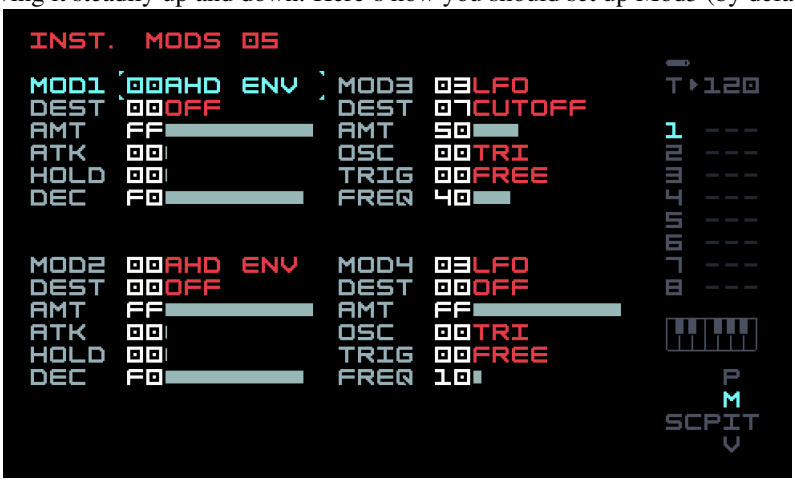
In this song, the pad is extremely simple: it is a single note that starts low, builds in volume to a plateau, and then persists at that level to just before the very last bass note that ends the song. We'll use track 4 for the pad. In Song View, put a reference to new chain 09 in row 02 of track 4. Navigate to Chain View for that chain, and put a single new phrase 0B in the first row. Navigate to Phrase View for that phrase, and put a single C-4 note on its first row. Add a reference to new instrument 05, and navigate to Instrument View for it.

We'll use Hypersynth as our synth type. Although Hypersynth has chord capability, we won't be using it here; the six oscillators will sound in unison, with slight detuning using the Swarm parameter, and a bit of spread in the stereo field using the Width parameter. We'll also add a bit of suboscillator for support. We designed this song before waveform choice was added to Hypersynth, so it uses the default SAW waveform.

For the generic synth parameters, we will use the lowpass filter to tame the sound a bit, and add not only chorus as before, but also reverb. Here is the full Instrument View.



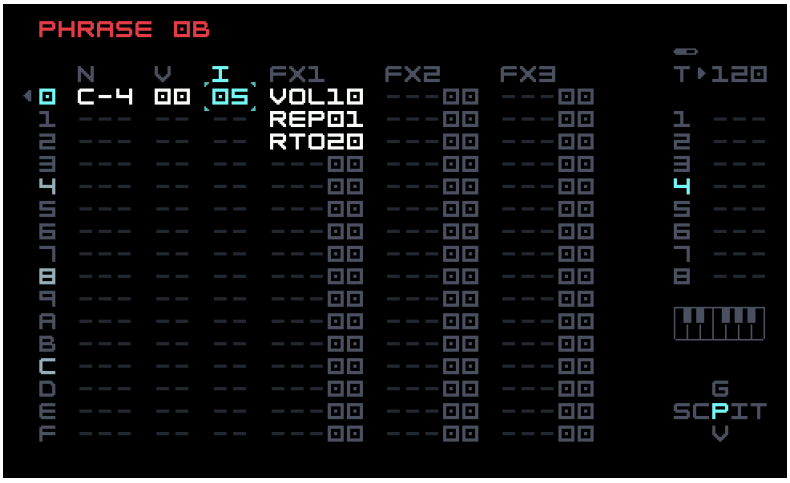
There is some evolution in the pad over the length of the song. Part of this is achieved by an LFO on filter cutoff, moving it steadily up and down. Here's how you should set up Mod3 (by default an LFO).



When you press [EDIT] to change the frequency of the LFO, note that at the bottom of the screen, there is a translation of the 2-digit hex amount into a time period. There are three more modulators left unused; you could put them on Swarm and Width, and add another one to modulate pitch slightly for a vibrato effect. But we didn't.

The slow volume build is a challenge for the envelope shapes available. We won't use another of the four instrument modulators to achieve this. Instead, we'll use some FX commands in the phrase. Go back to phrase view, set the velocity to 00 in the V column of row 0, and add the VOL command in the FX1 column, setting the amount to 10.

That seems silly at first, because we could just set the V column to 10. But we're going to modify that value by putting a REP command just below the VOL command, in the FX1 column of row 1. This will repeat the previous command at each step, with an optional change. REP01 will increment the argument of the VOL command at a rate of one per step. The change is smooth, not stepped, which is just what we want. The repetitions are cancelled by another command in the same column or REP00. But to hold the volume at the plateau level, we use the RTO ("repeat to") command. Putting RTO20 just below the REP01 (the FX1 column of row 2) will stop the repetitions when the increment reaches 20, for a total volume of 30 (adding in the result of the VOL command itself). Here's the final view of this phrase.

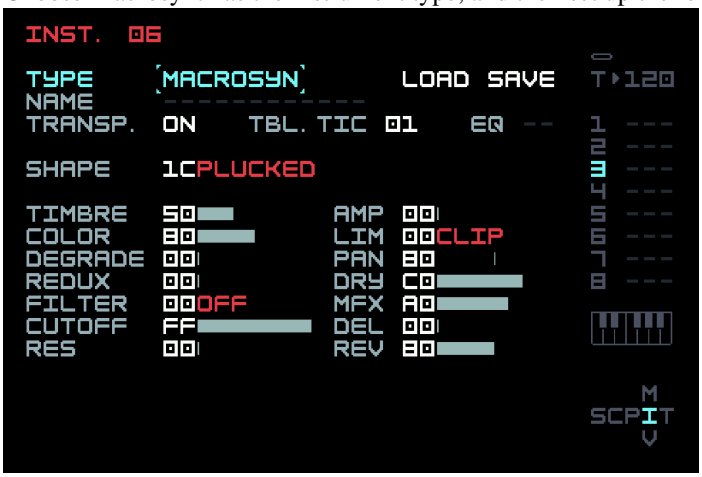


Stopping the playhead for the pad in song row 07 will not kill the sound. That has to be done explicitly with a KIL command. So back in Song View, create a reference to a new chain 0A in row 07 of track 4, navigate to Chain View for that and add a reference to new phrase OC on row 0, navigate to Phrase View for that, and put KIL00 and HOP00 commands in rows 0 and 1 of the FX1 column.

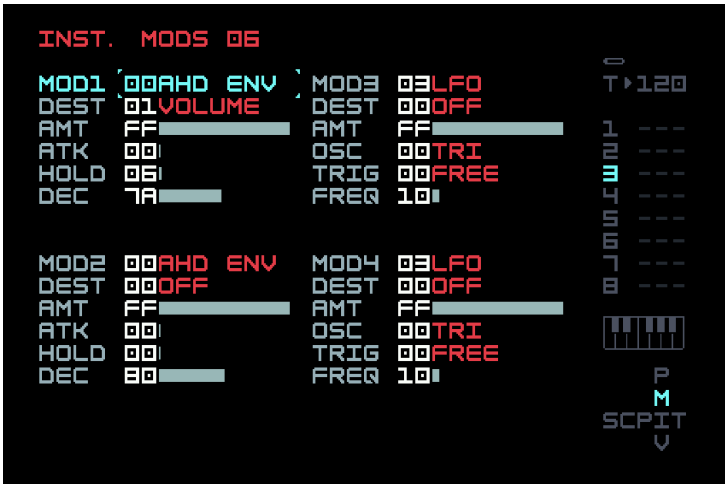
The pluck is the main melody. We're calling it that because it's less generic than "lead", and because we are actually using a physical synthesis model simulating the sound of a plucked string. This is one of the models available in Macrosynth, called "Plucked".

The pluck starts four bars after the pad, in song row 03. We'll use track 3 for this sound. Navigate to that spot in Song View, add a reference to new chain 0C, and navigate to Phrase View. There are three notes in this bar, on rows 2, 4, 6, and the first one is conveniently a C-4. So add that note, velocity 40, and a reference to new instrument 06, then navigate to Instrument View.

Choose Macrosynth as the instrument type, and then set up the remaining parameters like this.

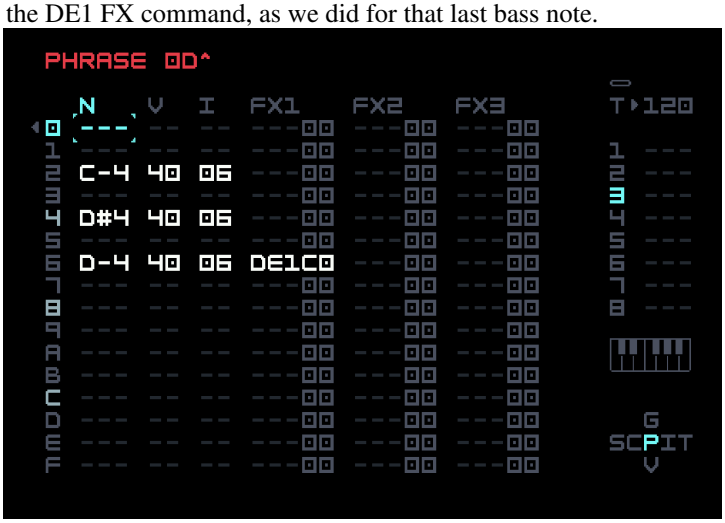


The important ones are the shape (plucked model), the timbre and color (these change the sound, try some other values), and the chorus (MFX) and reverb settings. We put an envelope on the volume to make the pluck shorter, and you know that is done in Instrument Modulation View.

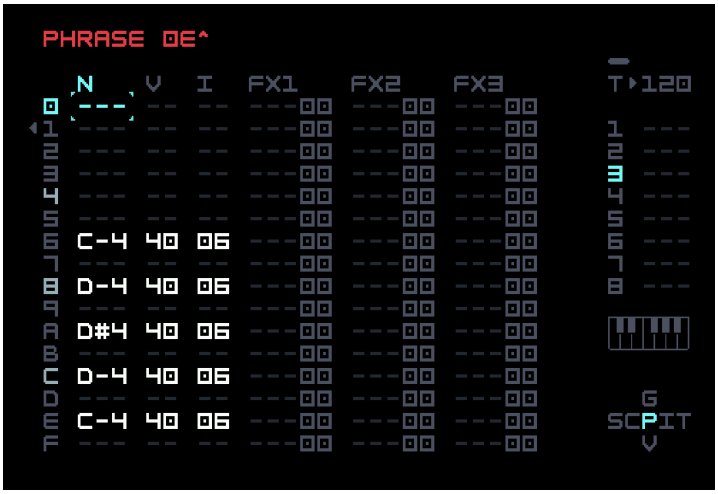


You will notice, when doing sound previews, that consecutive notes don't sound identical, even if all the settings are the same. The engine has a certain amount of variation built into it. We won't always be this lucky, and we'll usually have to program our own modulations to achieve this.

Navigate back to Phrase View and put in the other two notes. We want the third one to ring out longer, so we'll use the DE1 FX command, as we did for that last bass note.

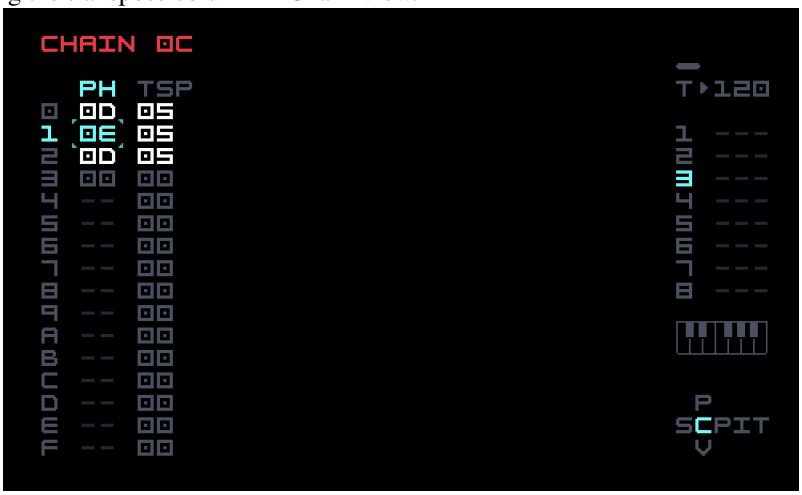


Back to Chain View for chain 0B. The third bar is the same as the first, but the second bar is different. Add references to phrase 0D in the second and third rows. The fourth row is silent, so use the empty phrase 00. Then go back to the second row and clone phrase 0D into phrase 0E. Navigate to that and set up the five-note pattern in this second bar.

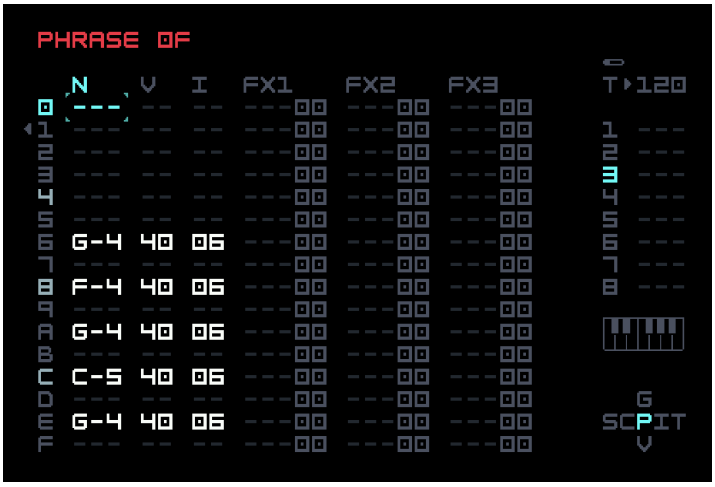


That completes the work on chain 0B. Navigate back to Song View. In the pluck track 3, chain 0B is in row 03. It repeats in row 04, and again with a slight variation in row 06. But row 05 has a larger variation. Here we are close to the classic AABA song pattern, though that uses eight-bar chunks, and we are using four-bar chunks for simplicity.

Add references to chain 0B in rows 04, 05, and 06, and then clone the one in row 05 into new chain 0C. The B section (sometimes called the ”bridge”) would normally be something more ambitious, but again for the sake of simplicity and learning, we have chosen something very simple: we just transpose the previous chain 0B up by five semitones (a fourth). This would be a nuisance to do by cloning phrases and adjusting each note, but it is very easy using the transpose column in Chain View.

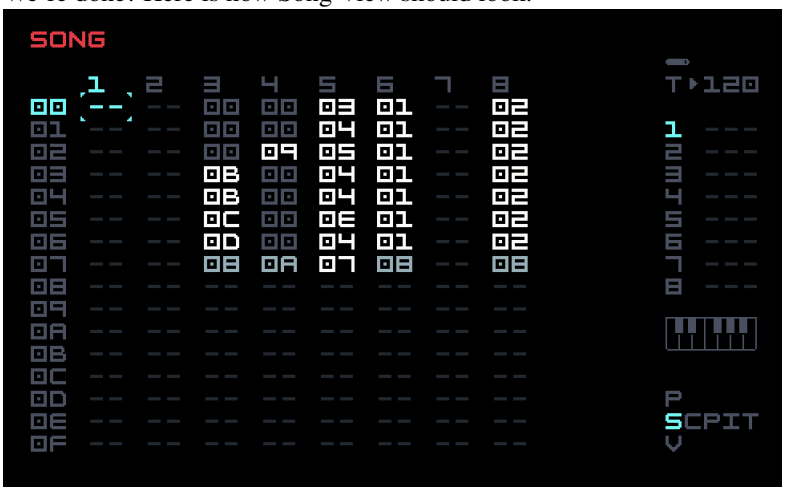


For the slight variation on the A section in row 06 of Song View, clone the reference to 0B into new chain 0D, and navigate to Chain View. The variation is in the second bar, so clone the reference to phrase 0D in row 1 to new phrase 0F, navigate to Phrase View, and make the changes.



Back in Song View, track 3 can be stopped with phrase 08 as with the drums. Because row 05 has the pluck transposed, we should do the same with the bass in this row. So navigate to track 5, clone the reference to chain 04 into new chain 0E, navigate to Chain View, and add the transpositions (up five semitones). We chose not to transpose the pad. If you want to do that, it's not quite as simple, because the pad is a single note triggered in row 02, and there is no trigger in row 05 to transpose. You would need a new chain in this row with a phrase that uses the PIT ("pitch") command in an FX row, and another one when the note needs to be transposed down again.

We're done! Here is how Song View should look.



Download link for project file: First.m8s

This file does not contain the factory samples used. If for some reason these are not on your M8, you can substitute any available samples of a similar nature, since the precise sound is not so important.

This song is too short and simple. But expanding it into something more respectable can be done by just doing more of what you already did to create it. You could make the percussion tracks more varied and interesting, and maybe add some additional percussive sounds in the unused track 7. The pad could easily be made more interesting, and there are unused modulators in the bass and percussion tracks as well. The B section could be rewritten to diverge more from the A section, or you could go further, beyond AABA form, to a more complex melodic exploration. The idea of cloning a phrase or chain to make a variation on it is the key to further development of this sort. Tracks 1 and 2 could be used for ornament, vocal samples, or whatever sounds you care to fit in. You may also be more inspired to do this kind of work on something that you create yourself from scratch, and you now have the tools to let you do that.

In future chapters, we'll look further into the more general sound engines of M8, as well as some major features we haven't covered yet.

3.7 To keep in mind

- Don't try to memorize all the transformations possible in selection mode, or their key combos. Just roughly remember what's possible, and return to the manual or this tutorial as needed, at first. Some of them may become part of your regular routine, while others may be used infrequently, or not at all.
- Cloning is an important developmental tool. You may have had the experience with other electronic music devices of being stuck in a four-bar or eight-bar loop. M8 makes it quick to copy and tweak to create variations, and arrange the results in a larger musical structure.
- It's worth taking a little of your M8 time to explore the various instrument options. We'll say a little more about them in upcoming chapters, but neither this tutorial nor the manual can really cover all the details.

4 Dub techno with the FM Synth

The FM Synth is the deepest synthesis engine on the M8. It is possible to use it just with preset instruments, from the factory content or the user community. One can also tweak these experimentally, changing parameters to see what effect they have on the sound. At the opposite extreme, one can learn enough of the theory of how FM works in general to be able to hear a sound and have a sense of what configuration would at least get into the vicinity. We're going to steer between these two extremes. A little bit of demystification can help to guide preset tweaking or experimentation.

FM stands for "frequency modulation". Most people have seen or heard examples of this, though they may not realize it. When a violinist enriches their tone by a careful, rhythmic wiggling of the left-hand finger holding a string to the fretless fingerboard, it changes the length of the string and thus the pitch being played, slightly. This is known as **vibrato**. When an electric guitarist wiggles their "whammy bar" with their right hand, it changes the tension of the strings and again the pitch. Speeding this up past the ability of humans is essentially what the FM Synth does, with surprisingly complex results.

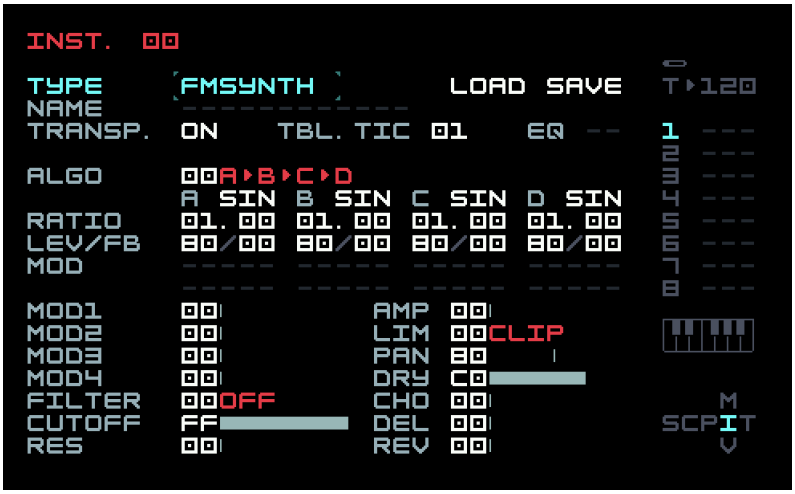
Both the left-hand wiggling of the violinist and the vibration of the string caused by the movement of the bow in their right hand are examples of physical oscillation. The string vibration causes, by means of waves of air pressure, a similar vibration in our eardrums, which our ears interpret as sound. The faster the vibration, the higher the perceived pitch. The frequency of the vibration is measured in cycles per second, a unit known as Hertz (Hz) after the German scientist. For example, an A-6 note is 440 Hz on the M8. We won't need to use this precision very often, but you may also see this terminology used elsewhere. Note names, in particular octave numberings, are not standardized across devices. In scientific pitch notation, 440 Hz is A_4.

FM, then, is the use of (usually rapid) changes in pitch, for sonic effect. Its use in electronic music was popularized by the analog modular synthesizers of Robert Moog and Don Buchla in the 1960's, and in the digital realm by the work of John Chowning (at Stanford) in the 1970's, licensed to Yamaha for use in the DX7 and its successors. The patents granted to Chowning and Yamaha ensured a period of market dominance, but following their expiry, many other manufacturers incorporated these ideas into their devices, because a lot of successful performers had also taken to them.

We haven't mentioned it yet, but you've probably noticed a visual waveform display or **oscilloscope** at the top of some views, notably when previewing an instrument or playing a song. This is more than eye candy; it can be useful in sound design, and we'll use it right now as we reproduce the vibrato effect using the FM Synth.

4.1 A first look at the FM Synth

Create a new project, add a new chain in Song View, a new phrase in that chain, and a single C-4 note in that phrase, then navigate to Instrument View and choose FM Synth as the type.



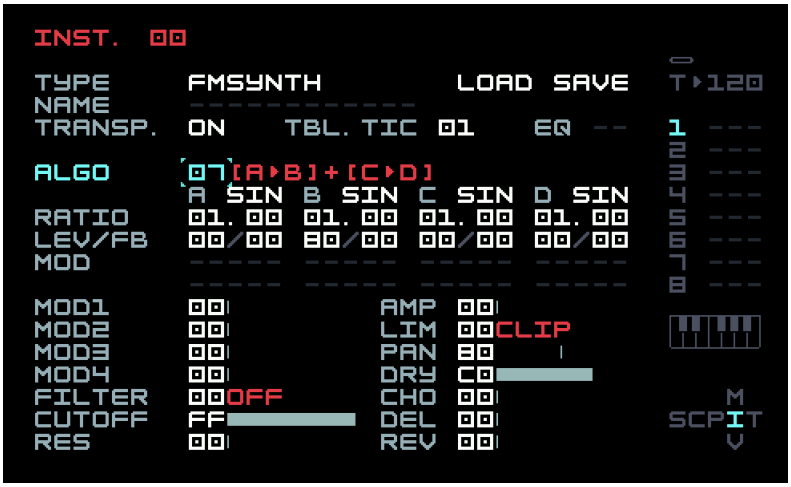
Digital FM synthesizers can be quite complicated. Tim has made a number of ingenious choices to respect the physical and computational limits of the M8. There are still a lot of parameters, and we'll approach them slowly. An oscillator is often called an **operator** in FM synthesis. It can be further characterized as a **carrier** (the oscillator being modulated) or a **modulator** (which modulates the frequency of a carrier). Going back to the analogy of the violinist, the string is the carrier and the wiggling left-hand finger is the modulator.

FM Synth has four operators, called A, B, C, and D (4-op FM). This is relatively modest; the DX7 offered 6-op FM, and some of its successors even more. But it's not a deficiency. The Yamaha DX11 used 4-op FM also, as does the currently-available Yamaha Reface DX, and the Elektron Digitone (both models). One can do quite a lot with four operators.

To cut down on configuration complexity, most digital FM synths have a set number of operator arrangements, known as **algorithms**. There are twelve on the M8 (compare with eight on the Digitone and DX11, and thirty-two on the DX7). These are quite similar to the twelve on the Yamaha Reface DX, though the operator implementation there is a bit different. The algorithms are described with a brief linear text string on the M8, and with two-dimensional diagrams in the M8 manual. Because all these machines differ in the set of algorithms, the set of modulators, and the ways they can be applied, it is generally not possible to directly port a patch from one to another. One can often trim and adjust and get close, though.

The default algorithm 00 is described as A>B>C>D. What this means is that operator D is the carrier, and C modulates D. But B in turn modulates C, so C is in a sense both modulator and carrier. It's as if B were someone grabbing the left wrist of a violinist and shaking it. (Please don't try this at home!) Finally, A modulates B. This algorithm is not what we want to start with, so change it to algorithm 07, described as [A>B]+[C>D]. This means that B and D are carriers, and their sounds are mixed together. A modulates B and C modulates D, respectively.

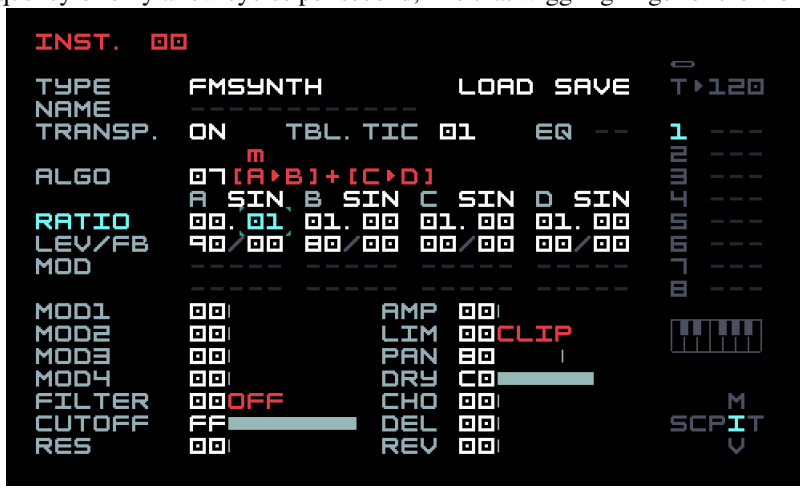
The row after the algorithm choice and description sets the waveforms for the individual operators, and the row after that is labelled Ratio. We'll get to these in a moment. The next row is labelled LVL/FB. LVL means "level" and FB means "feedback". The level is simply the amplitude or volume of the operator. We only want to hear operator B at first, so set the levels of the others to 00. The screen should look like this.



Now you're ready to listen to the result, either by holding [EDIT]+[PLAY] or by tapping [PLAY] and letting the note you put in the phrase play continuously with periodic retriggering. You should hear a low, simple sound, and see a simple waveform on the oscilloscope at the top.

The waveform is a sine wave. You may remember sines from high-school trigonometry; they are useful in computations involving angles and distance. Mathematically, any periodic waveform (repeating exactly) can be represented as the sum of sine waves of different frequencies. This makes them useful in discussing complex sounds; we can talk about the frequency **spectrum** of a sound, the set of frequencies and levels of the sine components of that sound. Practically, FM is easiest to control when the operators are sines, though the other waveforms are there because they have their purposes.

The row labelled Ratio contains four numbers, one for each operator, though each number has separate two-digit fields before and after the decimal point. Ratios are expressed in regular decimal, not hexadecimal. A ratio is just a multiplier of the note frequency. An operator with ratio 1.0 has the same frequency as the note; a ratio of 2.0 means twice the frequency of the note. We want the modulator to be really slow. So take the left field of A's ratio down to 00 (with [EDIT]+[DOWN]) and the right field up to 01 (with [EDIT]+[RIGHT]). You'll notice, as you do this, that the M8 adds a m/c annotation above the algorithm description, to help you see what you're changing. A ratio of 0.01 is a frequency of only a few cycles per second, like that wiggling finger of the violinist. Your screen should look like this:



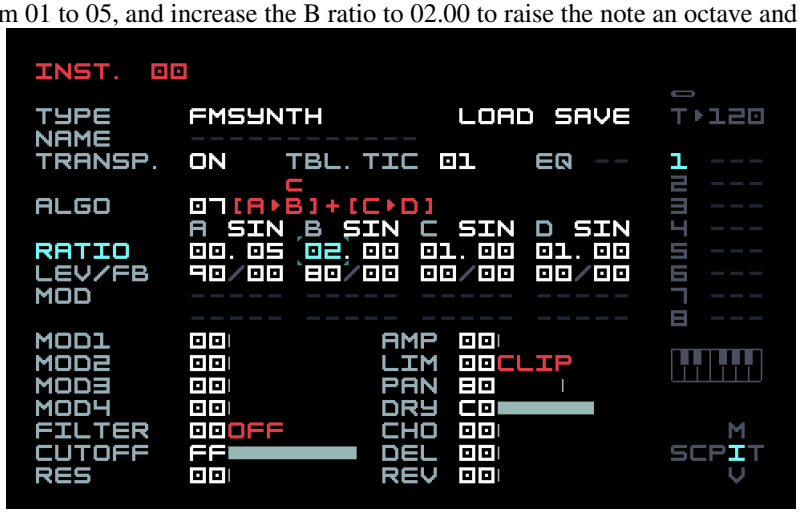
Now listen to the result. It won't sound any different. But the waveform will be rocking back and forth slowly. That might make you frown. Surely if the frequency were being changed, the waveform should become narrower, then wider. But its shape is constant, with only its position changing. This setting of parameters reveals a core truth about the FM Synth: it does not implement frequency modulation, but **phase modulation**. This is a good thing, as it turns out, and the DX7 and Digitone work the same way. You can skip over the explanation that follows without any real

effect on your use of the M8, but you might find it interesting.

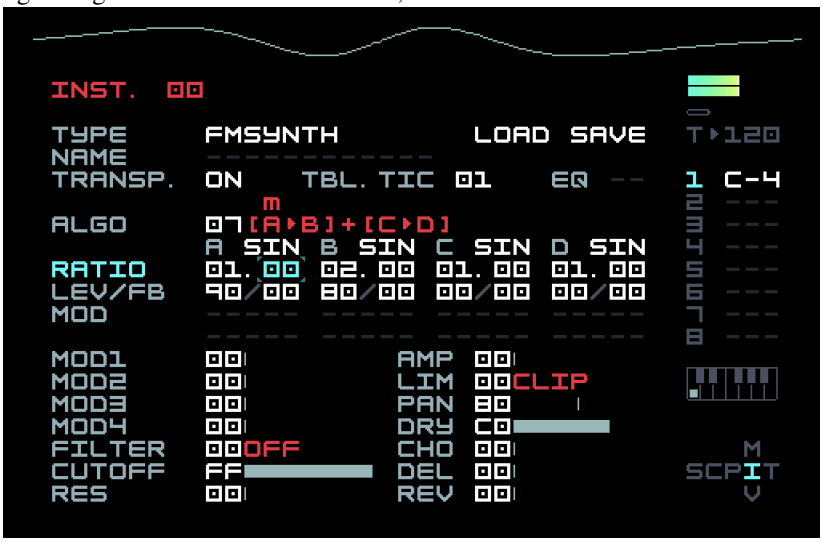
You may have worked a lot of problems in high-school calculus that involved cars or rockets. There, you would have learned that speed is the derivative (instantaneous rate of change) of position, and acceleration is the derivative of speed. Going the other way, position is the integral (continuous sum) of speed and speed is the integral of acceleration. Analogously, frequency is the derivative of phase (position in the waveform), and phase is the integral of frequency.

Quantities tend to be continuous in the analog realm, but discrete in the digital realm, so integrals become sums and derivatives become differences. Phase modulation is computationally simple and efficient digitally. It is also gentler. Phase modulation is like pushing and pulling on the bumper of a moving car, while frequency modulation is like tapping rhythmically on the accelerator or brake. (Definitely don't try this at home!) Ultimately what matters is what is musically useful and diverse, and phase modulation definitely has those qualities. Because of the relationship between phase and frequency, anything that can be done with phase modulation can be done with frequency modulation, and vice versa, but the modulator for one might be relatively simple compared to the modulator for the other, in order to achieve the same effect. End of digressionary explanation.

We can get closer to the vibrato effect by increasing the speed of the modulation. Take the right field of the A ratio from 01 to 05, and increase the B ratio to 02.00 to raise the note an octave and make it easier to hear.



This isn't actually the best way to add vibrato. You can put an LFO on pitch, and there's also a specific FX command, PVB (pitch vibrato). We used the idea more as a gateway into FM synthesis. Remember we said that FM was vibrato sped up past human ability? You can do that now, by taking the left field of the A ratio from 00 to 01, and taking the right field down from 05 to 00, so the A ratio is 01.00.



Previewing this, you will hear and see a waveform slightly more complex than a sine. You now have a 2-op FM implementation to play with, and you can do quite a lot with just that (as we will see). Try various ratios and levels for the A operator. Stick to whole-number ratios at first (change only the left field), but then try a small change in the right field, by .01 or .02. Many FM synths have a "detune" parameter to achieve the latter effect. Then try some fractional ratios. You'll find that the "big jump" edit operations make a change of .25 in the right field of a ratio. Those sounds are still more complex but fairly tonal. Moving away from whole numbers can result in sounds that are wooden, metallic, or just noisy (possibly with more modulation involved). Raising the level of the A operator from zero makes the sound more complex, but not in a linear fashion like turning up a volume knob. You can get a grasp on these phenomena with the help of a little more musical math.

We already mentioned that doubling the frequency of a sine wave increases its perceived pitch by an octave. A-6 is 440 Hz, and A-7 is 880 Hz. These sound good together; they are harmonious. Doubling the frequency of A-7 twice (multiplying it by four) gets us to A-8 at 1760 Hz. What if we triple the frequency of A-6?

The perceived pitch of this is pretty close to E-7, which is seven semitones or a fifth above A-6. The reason it isn't exact is due to a musical compromise called **equal temperament**, which lets musical pieces move through more than one key without requiring musicians to retune on the fly. The point is that it still sounds good. In fact, all the multiples of 440 Hz sound good with that A-6. The sine waves with those frequencies are known as **harmonics** of the **fundamental** frequency (the smallest one). Adding these sine waves together at various levels will result in different waveforms with different **timbres** (this word refers to the qualities of the sound other than pitch and volume). For example, a sawtooth wave has all harmonics, dropping off in level proportional to the inverse of the multiplier.

The final piece of the puzzle is the mathematical analysis of FM with pure sine carrier and modulator. This shows that if the carrier frequency is C and the modulator is M, then harmonics are added at C+M, C+2M, C+3M... but also at C-M, C-2M, C-3M, and so on. For example, if we have a carrier at 440Hz and a modulator at 220Hz, harmonics will be added at 660Hz, 880Hz, and so on, but also at 220Hz (C-M). Generally speaking, as the level of modulation is raised above zero, the level of the added harmonics goes up, and closer ones go up more than ones further away. But it's not always that simple. Sometimes raising a level can make a particular harmonic decrease. The mathematics is well-understood by experts, but it doesn't lead to simple rules of thumb or guidelines. A certain amount of experimentation is inevitable. The level of the fundamental frequency can also be reduced, and it's good to be aware of this possibility, so you can compensate if necessary. (This can happen because negative values of the C-kM frequencies show up as positive values but phase-inverted, so they will cancel out harmonics that they overlay.)

The math in the previous paragraph explains why the user interface of the FM synth steers you towards whole numbers or nice fractions. That makes it more likely that the sound is harmonious and responds well to note pitch information, though sometimes you want to dive into those in-between regions for particular kinds of sounds.

There are lots of parameters on the FM Synth that one might like to modulate. Desktop synths often handle this with a two-dimensional "modulation matrix" to select multiple destinations for each modulation source. Tim has squeezed a limited version of this into the FM Synth. As you know, there are four modulators available in Instrument Modulation View, which can be various types of envelopes or LFOs. Instead of sending a modulator directly to an FM operator parameter, there is another level of indirection, which we'll call "operator modulators" or just **op-mods** (in the M8 manual or on the screen, they are called just "mod", a potential source of confusion). This will let one instrument modulator control several FM operator parameters.

In addition to the destination choices for the instrument modulators that we've already seen, such as volume and filter cutoff, the instrument modulator can be chosen to be the source of one of the four op-mods. Then, in Instrument View, in the two rows labelled MOD, just below LEV/FB, each of the four operators has two **op-mod slots** (again, our invented terminology) available for our use. One op-mod slot can direct any of the four op-mods to four FM-specific destinations on the operator: level, feedback, ratio, and pitch (altered by semitones). The instrument-specific left side below has four controls for the base amount of each op-mod that is applied (the same amount for any op-mod slot that uses that op-mod). An instrument modulator assigned to an op-mod then has the same effect on it as if it were assigned to the parameters we've already used instrument modulators on. We'll make use of these op-mods in what follows.

4.2 Percussive elements

Dub techno is a genre that brings together elements of classic techno, notably the speed and repetitive nature of the rhythm and melodies, and elements of classic dub, with its heavier bass and use of delay and reverb. We've chosen it for two reasons. First of all, it uses a diverse array of sounds, letting us discuss various aspects of the FM Synth, but none of the types of sound we need are so complicated or specific as to create difficulties for us. It also takes from dub an emphasis on slower, more minimal changes over time, often improvised by the performer. That makes it easier to perform on the M8, since no rapid knob twists or flicking of faders is needed.

In contrast to the previous chapter, we won't design a full song, only a basic groove consisting of one four-bar chain per track. Track 1 will use chain 01, and so on up to track 8 using chain 08. You could fill this out into a song with a proper beginning and ending, using ideas from the previous chapter and ones you'll learn in this one, and include all the automation for slow changes so all you need to do is tap [PLAY]. But this lets us discuss, in a later section, how we might instead set things up for more of an improvised performance.

Here's audio of the groove repeated four times, with just an abrupt beginning and ending. The repetitions are not identical due to the use of FX commands on some tracks, which will be explained in due course. Listening to it at length using the project file will make the variations more clear.

Audio download link: FM.mp3

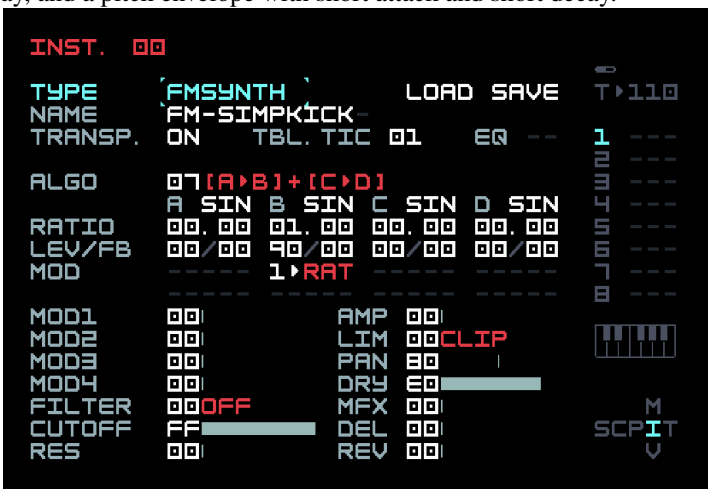
Project file link: FM.m8s

You don't have to stick to this groove at all. We will explain how we created it, but along the way, there will be plenty of opportunity to vary the sounds and the rhythms. You can do that the first time through, if you're eager, or you can build it as we did and then tweak it afterwards to make it more interesting or diverse.

There are many factory instruments on the M8, including some using FM Synth, and more available through the user community. One can also try, after a bit of experience, to adapt presets from other musical devices that use FM. In doing this, you might find that the presets are not necessarily implemented in the simplest fashion. The designer hit upon something that sounded good and saved it. There might be parameters or even whole operators that look like they are doing something, but which have no significant effect on the final sound. In what follows, we have tried to remove these extraneous elements that can creep in through experimentation, and present simple, clean implementations.

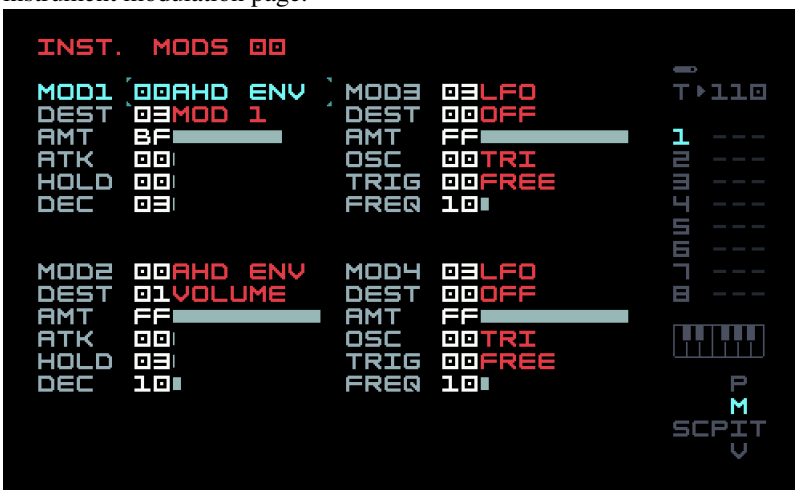
Create a new project, and in it make an empty chain 00 with four empty phrases 00. This is for possible future use; our instructions won't use this chain. Then, in track 1, put a reference to new chain 01, add a single phrase 01, move to the phrase, add a single note, set its instrument to 00, and its type to FM Synth. We've slowed the tempo to 110 bpm (in Project View), which is slower than most techno but faster than most dub.

The first four tracks contain percussive elements: a kick drum, a snare, a hi-hat, and a bell. The kick drum is the simplest of the sounds. One can make a kick drum out of just about any sound source with the right kinds of modulation. There are three important elements: a low sound, a volume envelope with a short attack and modest decay, and a pitch envelope with short attack and short decay.

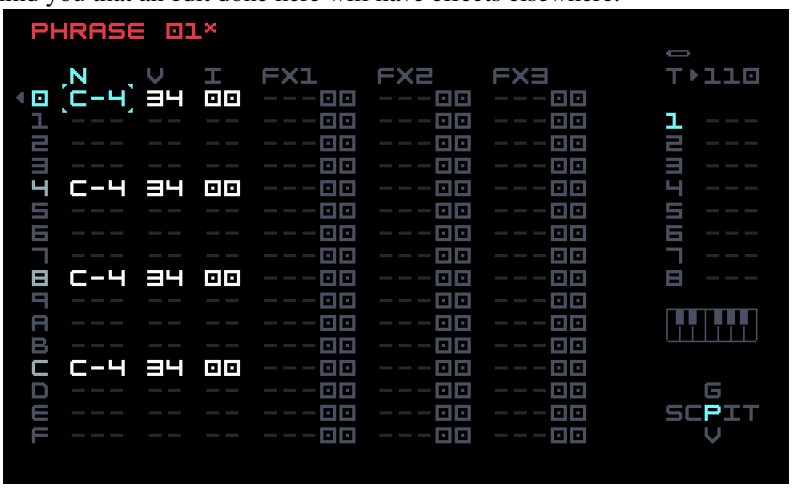


You can see that we're using the 2-op setup from the previous section, but the modulator A has been turned off. So

we are only using the single carrier B as a sine wave! The volume envelope is handled as we've seen in the previous chapter, but the pitch envelope uses an op-mod on operator B's ratio, mostly to illustrate the use of op-mods. Here's the instrument modulation page.



Instrument mod 1 is assigned to op-mod 1 for the ratio envelope, and its AHD envelope has a shorter decay than instrument mod 2, which implements the volume envelope. Go back to phrase 01 and put in the standard "four on the floor" techno beat, and then back to chain 01 to add three more references to phrase 01 to make up the four bars. You will see an X appear after the title of Phrase 01 in Phrase View. That means that this phrase is used more than once in the current chain. A caret (^) means that this phrase is used in a different chain. These little visual indications help remind you that an edit done here will have effects elsewhere.



Even with this simple an instrument, there are a number of parameters that you can use to affect the sound. Reducing the amount of the pitch envelope as set in Instrument Modulation View will make it sound more like a dull thump. Experiment with the two decay times. You could turn modulator A on (that is, actually use FM), or even use the other 2-op stack of operators C and D to layer another sound in, with the help of the two remaining instrument modulators. You may also be tempted to vary the beat a little with an alternate phrase in the chain.

You will often see a kick drum enhanced by using a highpass filter with resonance turned up, and the cutoff frequency tuned to the oscillator frequency. Because we wanted a more muted kick, we have not done this. Some people insist that a kick drum should be tuned to the key a song is in; others think this is far too fussy. Ours is in tune, but only because we're being lazy and going back to our old favourite C minor (signalled by our use of D# and A#, really E♭ and B♭).

For the snare drum, set up a single chain 02 on track 2, with a single phrase 02 containing a single note using instrument 01. (This initial setup will be the same for all tracks, with only the numbers changing in parallel, so we'll

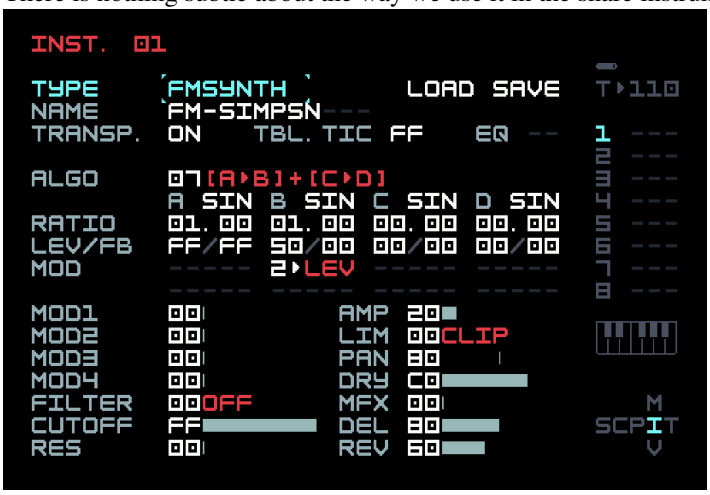
stop repeating this.)

The snare uses a parameter that we haven't discussed: the feedback (FB). This is a computationally simple idea originally investigated in this context by Norio Tomisawa, the Yamaha engineer who authored several patents extending John Chowning's ideas for use in the DX7. A controlled amount of the oscillator's output is fed back into its own modulation. We can see the practical effect of this with the help of M8's oscilloscope.

Set up the 2-op FM stack using just operators A and B on algorithm 7 as in the previous section, with both operators at ratio 1.0, but turn modulator A off, and set the level of carrier B to 50. Then start auditioning the sound while turning up the feedback of carrier B in larger steps, 00 to 10 to 20 and so on. You will see and hear that higher harmonics are added in, and the waveform starts looking like a saw wave. Take the feedback of carrier B back down to 00. Next, set the level of modulator A to A0, and repeat the process, but this time add feedback to the modulator. You will hear something similar at first, but partway through a bit of instability will develop along the slope of the displayed waveform, and then as you continue, the sound becomes noise, whose character alters somewhat with further increments.

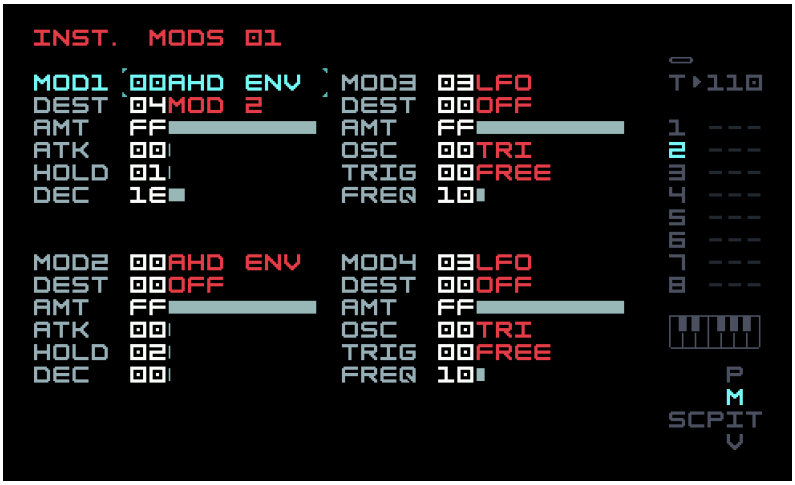
In a nutshell, this is the use of feedback: to morph from harmonically simple sine to harmonically rich saw, and also as a source of noise. Unlike the DX7, the M8 offers several choices of operator waveform, including saw and various types of noise. But it can be useful to use feedback in a subtle fashion.

There is nothing subtle about the way we use it in the snare instrument, though.



The feedback and level are at the maximum value of FF on modulator A, so it is producing a lot of noise, and that in turn makes carrier B noisy. On a real snare drum, the noisy tail of the sound is produced by wires running across the bottom drum membrane, which is not struck directly, but set into motion by air moving within the body of the drum when the top membrane is struck with a drumstick.

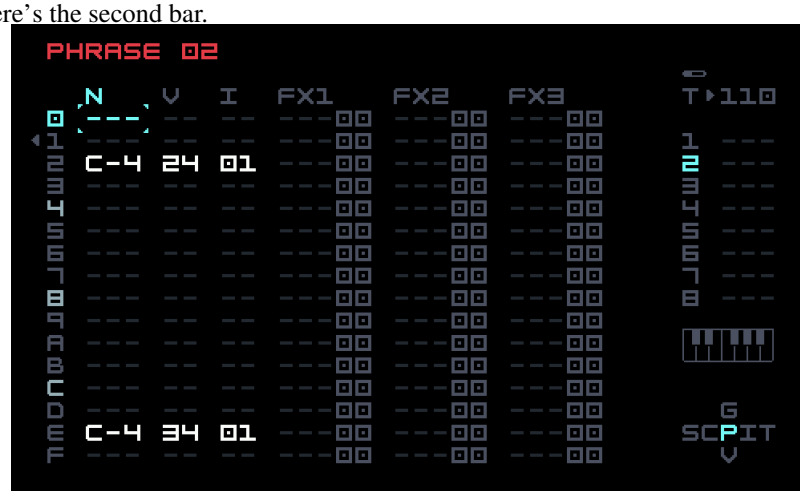
There is a volume envelope as with the kick drum, but this time we have used an op-mod to carrier B's level to achieve a similar effect, just to demonstrate that there are often several ways to do something on the M8. We used op-mod 2 just to emphasize that there is no direct connection between the numbering of instrument mods and op-mods, except for the connections that we specifically set up.



The sharp transient simulating the initial impact of the drumstick is achieved here by clipping. If you turn down the level of carrier B, it will go away. We've chosen to accentuate it a bit by increasing the AMP setting in the right effects column to 20. You can experiment with changing this value, and also by choosing a different limiter from the several choices available. We've also added some reverb and, for the first time, delay.

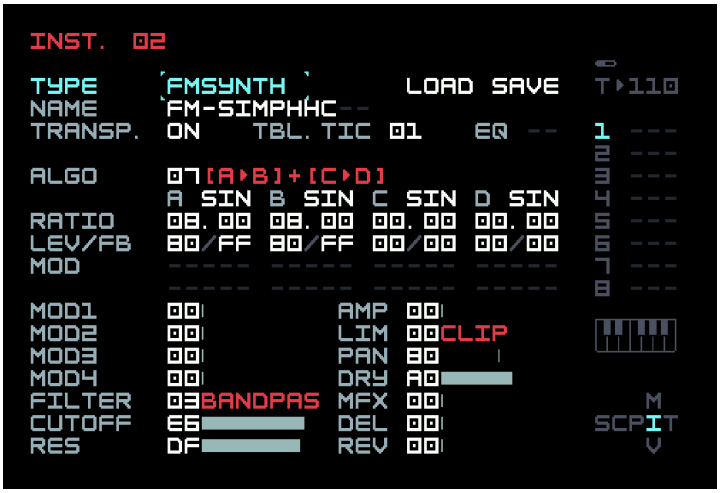
You might choose to design your own transient using the idle C-D stack, add a pitch envelope as with the kick drum (more subtle, though), or do something more sophisticated with the noise tail.

The snare in our groove is not as rhythmically insistent as in the song from the previous chapter, or the kick drum here. We've used sparser hits, with some velocity variation, over the last three bars of the chain (the first one is empty). Here's the second bar.

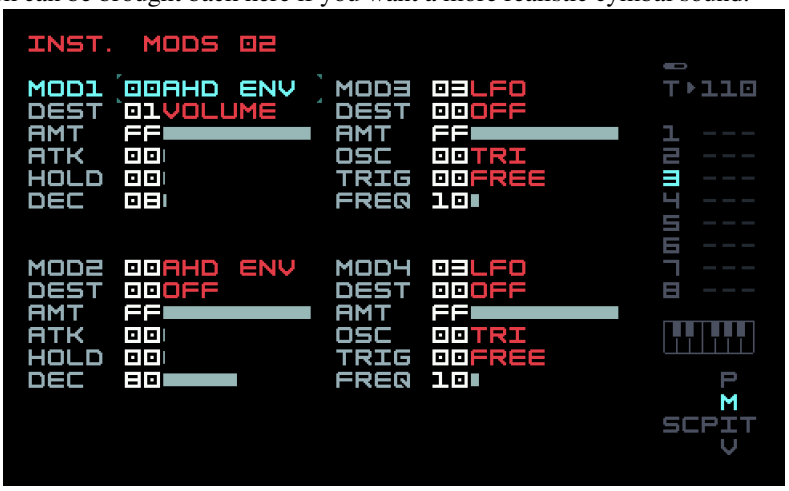


The third bar has a V-34 hit on row 3 and a V-24 hit on row B; the fourth bar has a V-34 hit on row 6 and a V-24 hit on row 9. These were chosen by ear, somewhat arbitrarily; feel free to vary them to taste.

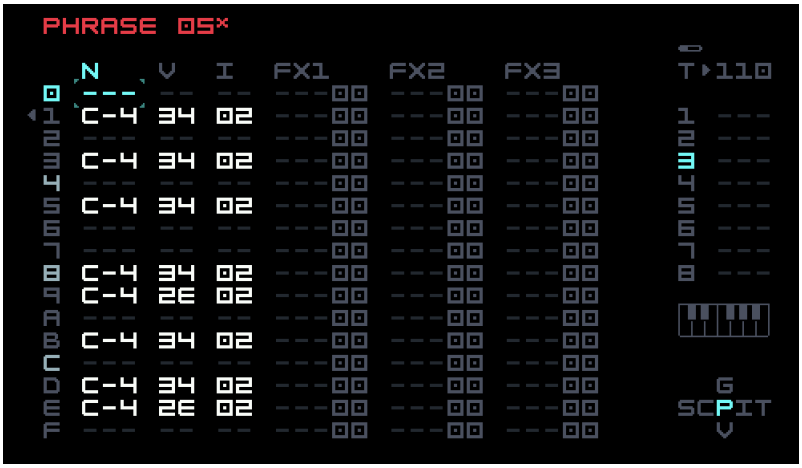
Set up the instrument for the hi-hat on track 3. A closed hi-hat is a simple variation on the snare: a higher noise (shaped by filtering) and a short volume envelope. Lengthening the decay a bit gets one an open hi-hat. Again, there's not much use of FM, though we'll use feedback to make the noise. A real hi-hat is more sophisticated, but this gets close enough for much electronic music.



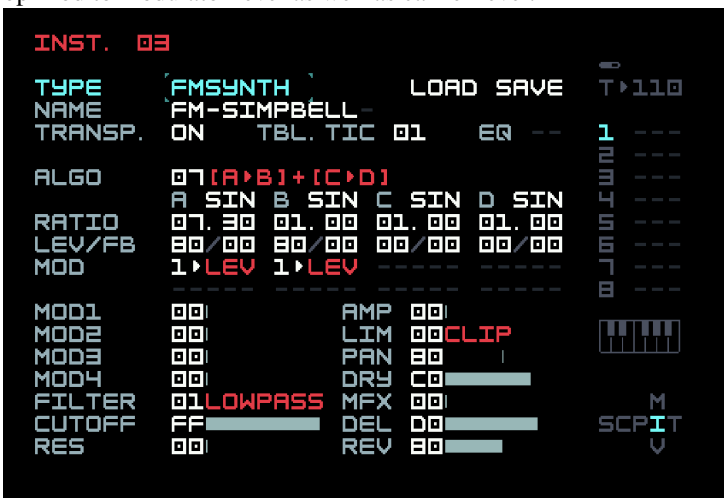
Note the use of a bandpass filter to focus the noise in a narrow high range. We've gone back to having the instrument modulator affect volume directly, rather than using an op-mod on carrier level. You can play with the parameters, but there is nothing much going on here. The technique used for making more metallic sounds on the next track can be brought back here if you want a more realistic cymbal sound.



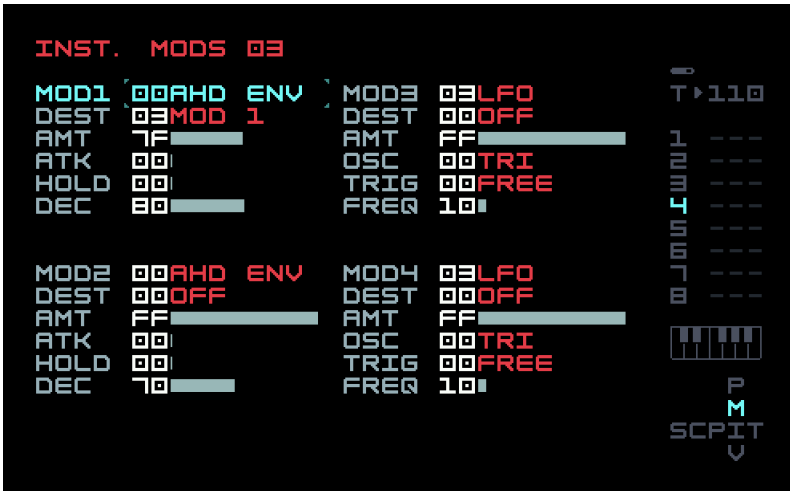
In our first song, we just put a hi-hat hit on each note of a phrase. Here, the phrase is slightly more sophisticated, with some rhythmic and velocity variations, but it's still just one phrase repeated four times in a single chain. You might choose to vary it more, or wait until we cover some probabilistic techniques in the next section, and bring those back for some random variation in velocity and timing.



The bell on track 4 finally demonstrates one of FM's advantages. This sound, as well as many metallic, woody, and percussive sounds, has an inharmonic frequency spectrum, which FM can produce with ratios that are not nicely related. Academic papers are usually written in bloodless prose, but it is hard not to hear the glee in Chowning's voice from this single sentence in his original 1973 paper: "Bell-like sounds can be produced by making the change of the index directly proportional to the amplitude envelope." Index is Chowning's term for the ratio of the modulator level to the modulator frequency (this plays a role in the mathematical analysis). So we can take the snare drum, turn down the feedback, set the modulator ratio to something that isn't a whole number or small-denominator fraction, and send the op-mod to modulator level as well as carrier level.



Here we see the advantage of using the op-mod as a volume envelope; we can reuse it as the modulator level envelope, leaving three instrument modulators free for other uses.



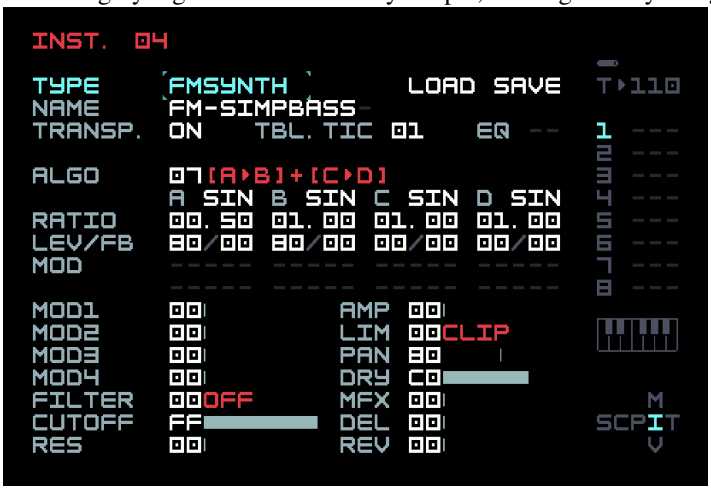
The phrases in the single chain on track 4 are quite simple. There are nonzero phrases on bars 3 and 4. Bar 3 has a single C-4 note, with velocity 44, on row 8. Bar 4 has a single G-4 note, with velocity 4B, on row 6.

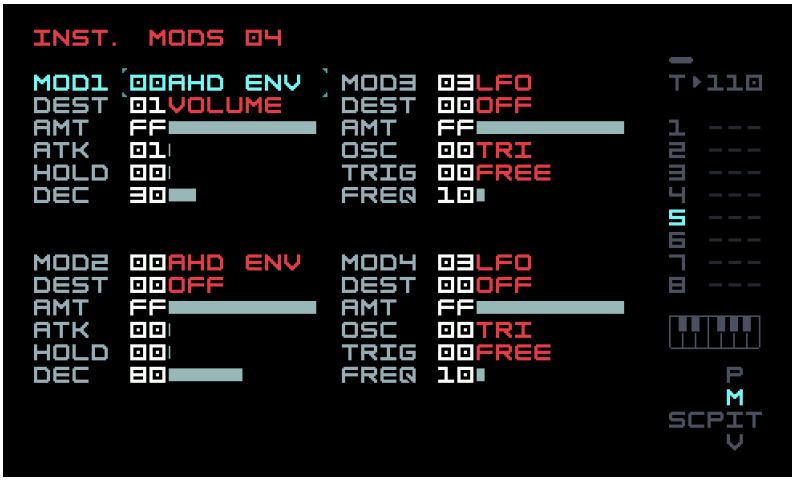
We could write a whole chapter on inharmonic FM sounds, but there are lots of tutorials on this available on the Internet, so let's move on.

4.3 Melodic elements

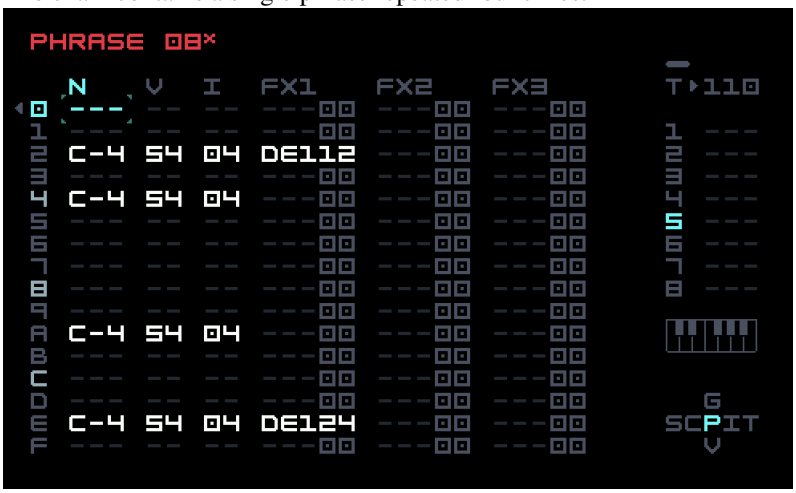
The four melodic sounds in tracks 5 through 8 are bass, dub stab, pad, texture. The adjectives used here are not all that accurate; arguably the bass is more rhythmic than melodic, and the bell from the previous section can probably be tuned better than the texture. We just needed to break up the development a bit.

In the analog world, bass is often enhanced with a "sub" that can be inexpensively generated as a square wave with half or quarter frequency of the main oscillator (one octave or two octaves down). The FM equivalent is a modulator with ratio 0.50 or 0.25, which adds in harmonics like a sub, but of course with sine waves. This works well, and FM basses are highly regarded. Ours is fairly simple, once again only using two operators.





The chain contains a single phrase repeated four times.

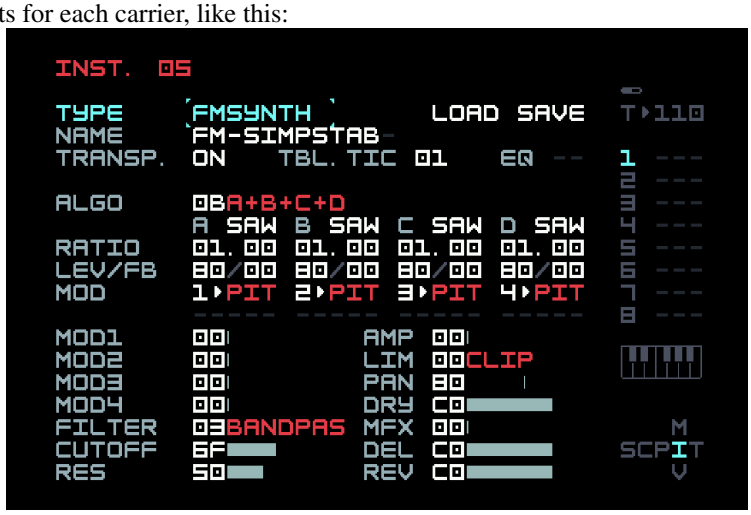


In listening to this, we noticed some periodic clicking. This was caused by the relatively long decay envelope. Sometimes a note following too closely would retrigger the envelope, causing a discontinuity that was audible as a click. Our solution was to use an FX command to shorten some of the delay envelopes enough to eliminate the clicking. You can hear the effect we strove to eliminate by deleting the DE1 commands.

Another source of clicking is an out-of-phase carrier (so it is at a nonzero value when a note is triggered) that is quickly activated by an envelope with attack 00. If you can't eliminate the click with a softer attack, or you don't want to use a softer attack (you are designing a percussive sound, for example), the SNC ("FM sync phase position") FX command can help. This is really the only FX command specific to the FM Synth that isn't connected to an obvious parameter. Setting this to zero could eliminate a click. Conversely, you might want to use it with a nonzero value to introduce a click, to accentuate the transient of a drum sound.

A dub techno stab is a short chord, usually minor, that is processed through filters, delay, and reverb. This is a fairly specific sound, but the techniques introduced here have more general applicability. We use algorithm 0B, described as A+B+C+D. In other words, we have four carriers, none of which can be modulators. Once again, there is no actual FM involved. We set the waveforms to SAW, since we're going to filter the result anyway, and this gives us harmonic richness. The bass design in the previous chapter used a similar technique. We used a pulse wave, which is harmonically rich in a different way, and passed it through a modulated filter to sculpt those harmonics over time. This is known as **subtractive synthesis**, and it's what we're doing here as well, only with four simultaneous notes instead of just one. The DX7 famously had no filter; harmonics were added by FM, not taken away. But M8, by offering the same filter options in every sound engine, and a choice of waveforms in the FM Synth, blurs the line between subtractive synthesis and FM synthesis (as does the Digitone in a similar fashion).

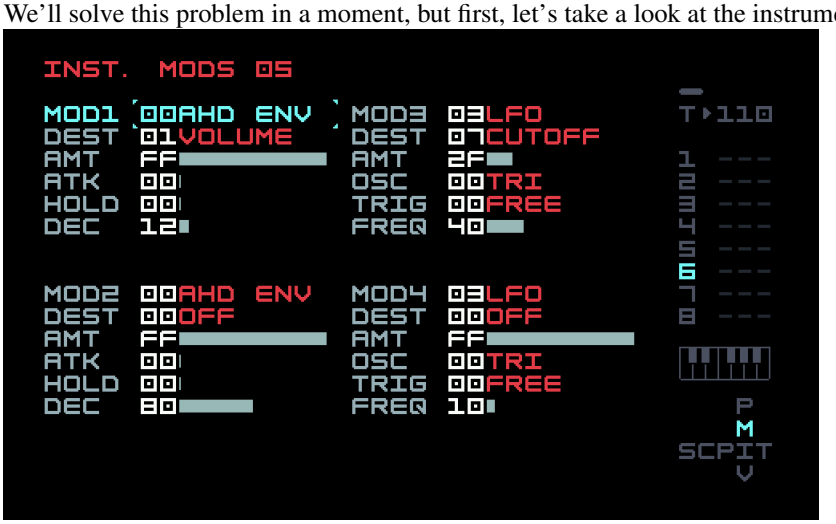
But how do we set the pitches of the four notes of the chord? The base note will come from the phrase. We could look up the ratios for the others on the Internet. We already know, for example, that 1.5 will work for a fifth. There's an easier way, though, which is to use an op-mod aimed at PIT (pitch), measured in semitones. We could use op-mod slots for each carrier, like this:



With this setup, changing the MOD2 amount to 03 will make the interval between carriers A and B three semitones, or a minor third. Similarly, MOD3 at 07 is a fifth, completing a minor triad. Since we have one more note to use, let's be bold and add a ninth by setting MOD4 to 0D (14 semitones).

But if we do this, the instrument will always play a minor ninth chord. That could be fine, but it would be nice to have more flexibility. We can specify these settings using FX commands in the phrase, so if we wanted, this chord instrument can be used for any four-note chord. And that's what we'll do. However, there's an additional complication. Three FX commands use up the whole row, and we want to make the stabs not trigger at times, randomly. We don't have the command space to do that.

We'll solve this problem in a moment, but first, let's take a look at the instrument modulators.



Besides the unsurprising volume envelope, there is a slow LFO on filter cutoff. Since we are using a bandpass filter, this will considerably alter the sound of stabs over time.

Now, about that squeeze on FX command space. Since we only need one more command for the randomness, one solution would be to nail the fifth down in the instrument, freeing up one slot. We wouldn't have a general chord instrument, but it could do major and minor chords, with an added seventh, ninth, or whatever. But it couldn't do diminished chords, which lower the fifth by a semitone, without using up that third FX command slot.

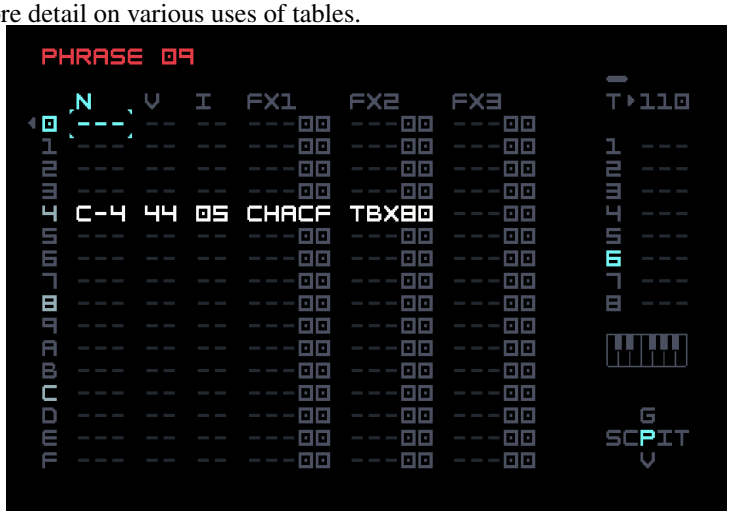
The more general solution is to use a table, which extends the use of FX commands. A table can be thought of as

a parallel sequencer running at tick speed (remember the default is six ticks to one step in a phrase). Each instrument has an associated table which starts running with each note, but there are 128 instruments and 256 tables, so tables 80 through FF are not associated with any instrument and can be used with a TBX (auxiliary table) command, allowing for selective use of a table, or use of different tables with the same instrument. We'll put the chord specification into the first unassociated table, leaving two slots free in the note's row in the phrase, for further modification there.

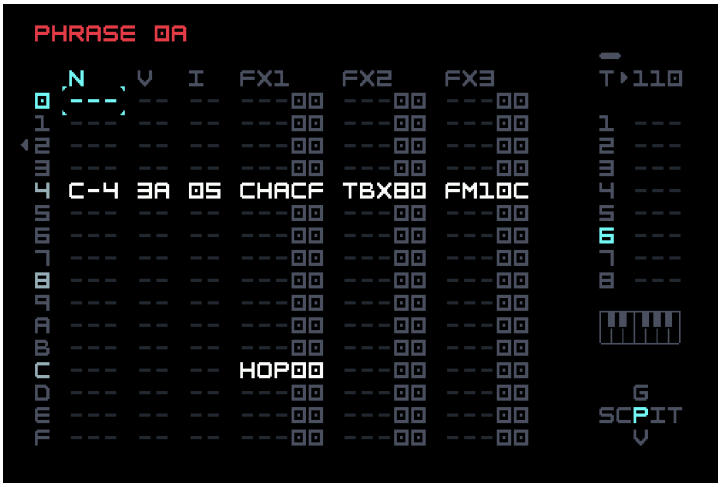
The table associated with an instrument is at the end of the minimap, in Table View, and you have probably accidentally encountered it by now. It's a little trickier to get to the unassociated tables. You have to put a reference to the table in a TBX command, and then, with your cursor on that value, use [SHIFT]+[RIGHT] to move to Table View for that table.



The THO01 (table hop to line 01) in line 01 of the table has the effect of stopping the table, so the FM commands remain in force for the duration of the note. In other words, we're not really using the sequencing capability of this table. It's just a place to put more FX commands to take effect as the note is triggered. In a later chapter, we'll go into more detail on various uses of tables.



In the first phrase of the chain, we've used TBX to run the auxiliary table, but also CHA CF for the randomness. The CHA command has separate measures of randomness that apply to the left and to the right of the command, specified by the left and right digits of the value. F means that the TBX command is always run but C means that the note to the left will trigger on average 80% of the time. The help text at the bottom explains the effects of the chosen digits when you edit the value.



The third phrase of the chain (the second and fourth are empty phrases) looks similar, but has two additional commands. One, on the same row as the note, raises the root note of the chord by an octave (12 semitones), in effect playing the first inversion of the chord. This is not as noticeable as the LFO on the bandpass filter; we just did it for demonstration purposes.

The other addition is a HOP 00 command on row C. HOP in a phrase will jump on the next step to the specified row of the next phrase in the chain, or, if the first digit is nonzero and the second specifies a position above the command, it will loop that many times. We've used it here to cut the chain short by one beat, so the dub stabs won't show up at the same point in each four-bar chunk. This is a general way to implement **polymeters**, or loops of different length playing in parallel, which can be very effective in increasing the sense of variation. The downside is that if we extend the song with multiple rows, this track will no longer have its playhead synchronized to the others. If you do this extension, you might want to reconsider this choice. We tried this with the bell, but it didn't work as well. The bell sounds better in lockstep with the other elements.

While we are in Phrase View, this might be a good time to mention that we have added **swing**. Musically, this consists of slightly lengthening some notes, or delaying the following notes. If you imagine a piano playing a steady stream of eighth notes, and then you slightly lengthen every other one but shorten the following note so the tempo does not change, you will have swing.

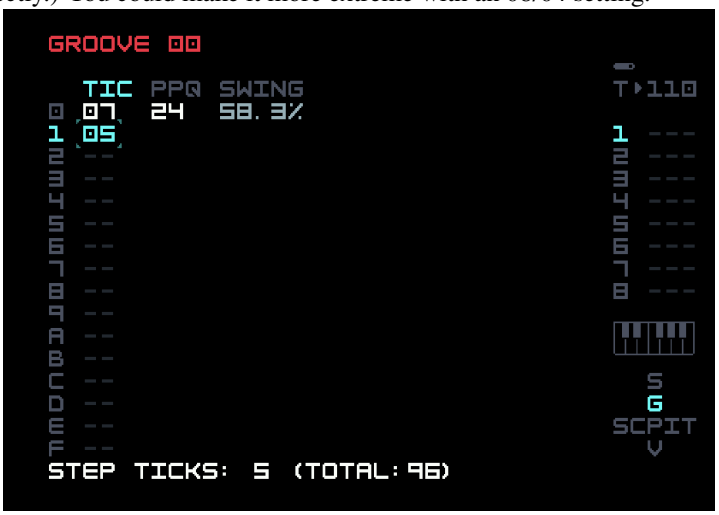
The implementation in most electronic music devices is somewhat curious. Because they are based on sixteen steps in a bar, the swing setting will lengthen steps 0, 2, 4, and so on, while shortening steps 1, 3, 5, and so on. They are swung sixteenths. But if you work in eighth notes, as we have mostly been doing, this will have no real effect. Rather than a single setting like this, M8 provides the more general mechanism of **groove tables**, which can be used for other purposes as well.

The default groove table is 00, and you can get to it from Phrase View by navigating upward (you can see the G in the mini-map).



Each row in the groove table represents a number of ticks in a step at the default PPQ setting of 24. This setting (and the computation of swing percentage) was added in firmware 6.5; PPQ is short for PPQN (pulses per quarter note), a standard unit of measure of time relative to a tempo. We'll keep it at 24 for now, and discuss other options soon. The sequencer progresses through the table, one row for each step, and loops back to the beginning on encountering an empty row.

The reason there are two 06 entries instead of just one for the default six ticks per step is to facilitate a quick application of the kind of swing discussed just above. With the cursor on the first row, [EDIT]+[UP] will increase the first row to 07. To keep the groove table aligned with the phrase, we use [EDIT]+[DOWN] to decrease the second row to 05. With each of these edit steps, a calculation of total ticks per phrase flashes at the bottom of the screen, to help you with the math. This results in swung sixteenths, within the tick resolution of the M8, with a swing percentage of 58.8%. (Many other devices, lacking a general mechanism like groove tables, set swing by altering the percentage directly.) You could make it more extreme with an 08/04 setting.



This won't do much to our sounds so far. To get swung eighths (with a noticeable effect on the bass), we used the following groove table.



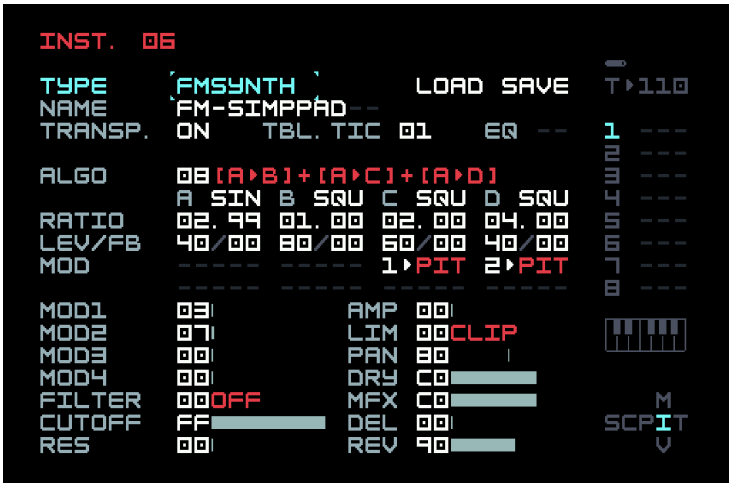
Each track can have its own groove table, set with the GRV FX command. A groove table with a single row containing 03 will play at twice the speed set by the BPM; similarly, 0C will play at half speed. The M8 manual has an appendix with some other groove examples. This is the way to get out of 4/4 time, if you want that.

The change between the first two groove tables above is fairly noticeable, and one might want to be even more subtle. This is where the PPQ parameter comes in. The other settings keep doubling the resolution, from 24 PPQN to 48, 96, and 192. This allows more fine adjustments. For example, with PPQ at 48, we could even out the treatment of sixteenth notes within the swung eighths above, if we wanted. 07 and 06 in the first two rows (at PPQ 24) correspond to 14 and 12 at PPQ 48, and the 05 in the fourth row similarly corresponds to 10. These could be replaced by, for example, 13, 12, 12, 11 respectively, creating an even more subtle swing. The computed swing percentages reduce the amount of mental math required.

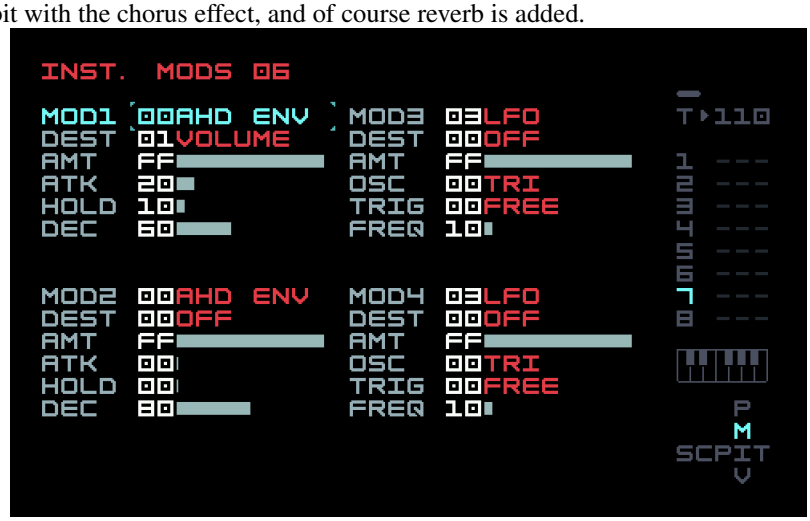
Don't forget that the DEL command will delay any individual note by some number of ticks. The microtiming MTT command (introduced in firmware 6.5) allows fine adjustment of timing both backwards and forwards, with a resolution of 1/48 step (192 PPQN, regardless of groove table setting) and a maximum change of one step. (One cannot do a backward shift on the first step of a phrase, though.)

This non-FM use of algorithm 0B (A+B+C+D) is the easiest and most flexible way to get four-voice polyphony on the M8. This capability was expanded considerably in firmware 5.0.0 (released February 2025), which permits operator waveforms to be chosen from the wavetables added to Wavsynth in firmware 4.0.0. Wavsynth has a SCAN parameter which scans through the wavetables; in FM Synth, the FB (feedback) parameter is repurposed to do this. Firmware 5.0.0 also lets the width of the IMP ("impulse") waveform be varied by the FB parameter, permitting the classic and attractive technique known as **pulse width modulation** (which is also available in Wavsynth and Macrosynth).

Track 7 has a pad. We used Hypersynth for this in the previous chapter, and we could do something similar with algorithm 0B, using finer detuning rather than spreading the carriers out into a chord. You might find yourself using this instead of Hypersynth on occasion, since FM Synth has more control over waveforms. But we're supposed to be demonstrating uses of frequency modulation, so let's go with algorithm 08, which has one modulator in common for three carriers.



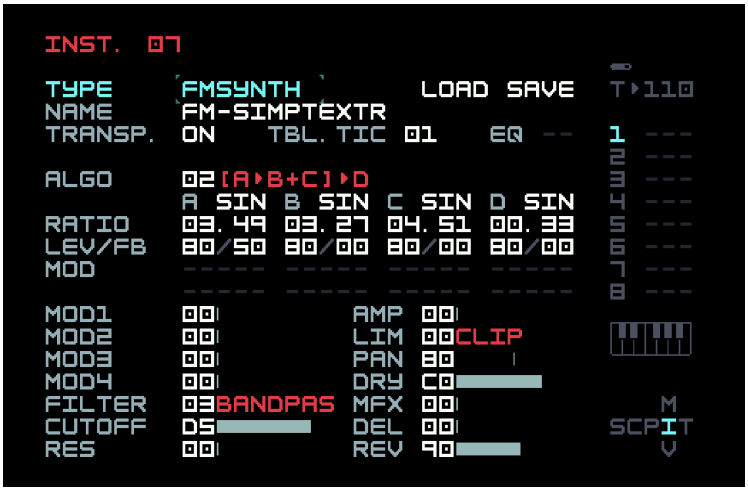
The carriers are tuned to a minor chord and then spread across three octaves, and the modulator is slightly detuned. Detuning the carriers didn't seem to work as well, but you may find a setting more to your taste. The sound is thickened a bit with the chorus effect, and of course reverb is added.



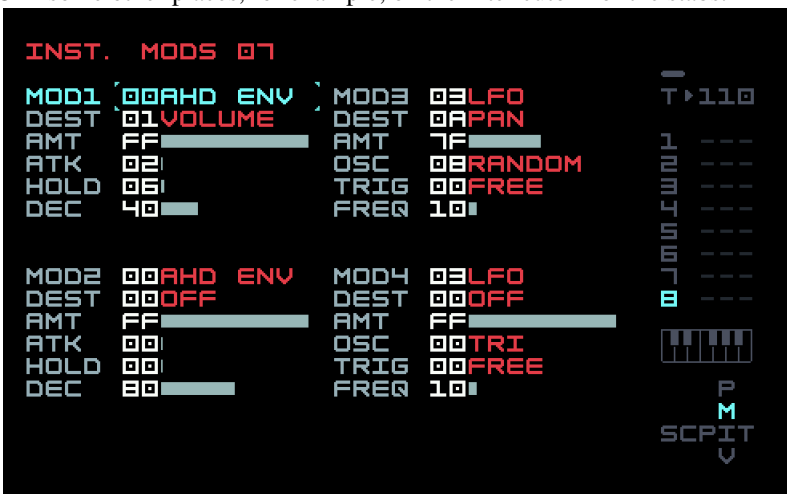
Our previous pad played continuously, with a slow volume rise and plateau created by FX commands. Here we use a slow AHD envelope. There is only one triggering note, in the fourth bar of the chain, a C-5 in the first row with velocity 18 and a CHA 80 command in the FX1 column, so the pad sounds with 53% probability.

This pad sound is the first one we have designed with both FM and a non-sine carrier (three squares in this case). Generally speaking, the more complex the carrier waveform (even with 2-op FM), the noisier the result, and this is even more the case with non-sine modulators. One interesting use of a non-sine carrier arises by setting its ratio to 0. This means that the carrier does not oscillate at all on its own; any change in it is due to the phase modulation from the modulator. If we view the waveform as the graph of a function, this applies that function to the modulator, and the result is what we hear as the carrier's voice. For example, if the carrier is a saw wave, it is the graph of the function $y = x$, so the modulator is passed through unchanged. But choosing another waveform for the carrier will have an effect. Setting the carrier ratio to 0 makes it into a **waveshaper**. Using some of the wavetable options introduced in firmware 5.0.0 can be quite effective.

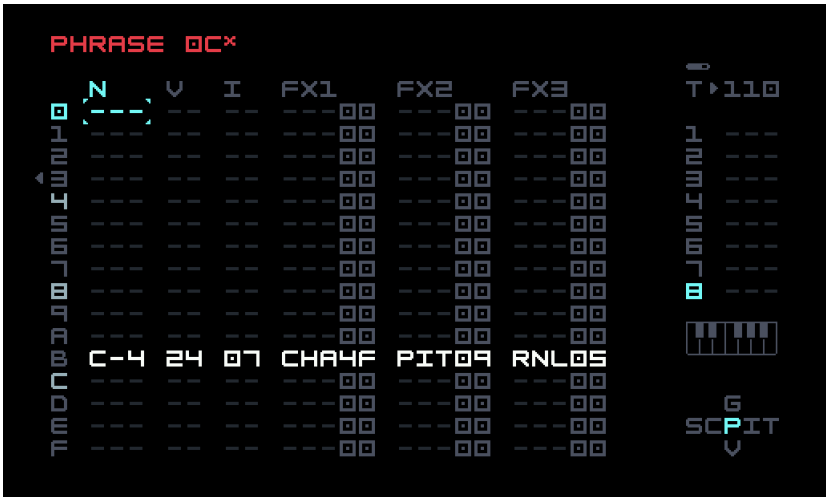
Finally, on track 8, we have a texture, which would fit into the factory content in the folder labelled FX, in this case meaning sound effects that don't fit neatly into the other more musical categories. We have a free hand to design this, and since it's supposed to be something unusual, let's go for algorithm 02, which has two parallel modulators (one a stack of two operators) and only one carrier. We just changed values until it sounded interesting, like an insect flying by on a peaceful summer day.



Besides the usual volume envelope, there is an LFO on pan (left-right position in the stereo field), with the LFO type set to random. We haven't done anything with panning so far; our instruments have been in the centre, with some stereo imaging provided by the ping-pong delay and the reverb. We could go back to the other tracks and reposition or automate panning for some of them. Generally, bass and kick should be centred. Similarly, we could use a random LFO in some other places, for example, on the filter cutoff for the stabs.



The chain for this last track has one phrase repeated four times. The single entry uses a CHA command to have the note trigger with 25% probability, as well as a combination of a PIT command (changing pitch in semitones) and a RNL (randomize to left) to vary the pitch with each trigger, so the sound is not identical each time.



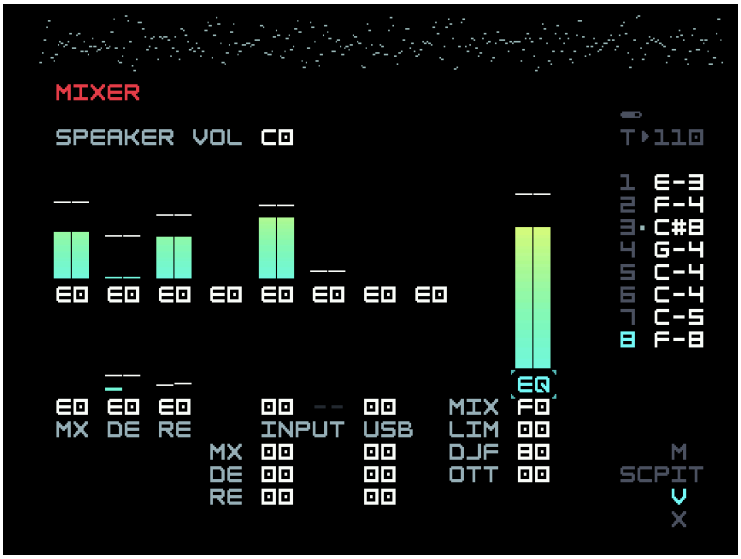
That wraps up our sound design for our dub techno groove! There's a lot more that could be done to it, but we'll confine ourselves to some brief remarks in the next section on how one might improvise using what we have, or something only slightly more elaborate. As a bonus, since we now have eight full tracks, hit [PLAY] in Song Mode, then press all four direction buttons simultaneously. You will see oscilloscope displays for all tracks! This is an "Easter egg", an undocumented feature (though it might show up in the next revision of the manual). Press the four buttons again to get back to the usual Song View.

As for the FM Synth, we've only scratched the surface, rarely using more than one modulator or two instrument modulators. There are more algorithms to explore, more operator waveforms, and more LFO and envelope instrument modulators to be applied to parameters. You have a solid basis for experimentation, and hopefully some ways of understanding the effects of what you are doing, so you can vary them sensibly. You can now start to deconstruct FM Synth instruments from the factory content or community sources. Be sure to try to simplify them; you might be surprised at how far you can go. A bigger challenge, but one that might be within your grasp, is to translate recipes for other FM devices for use on the M8.

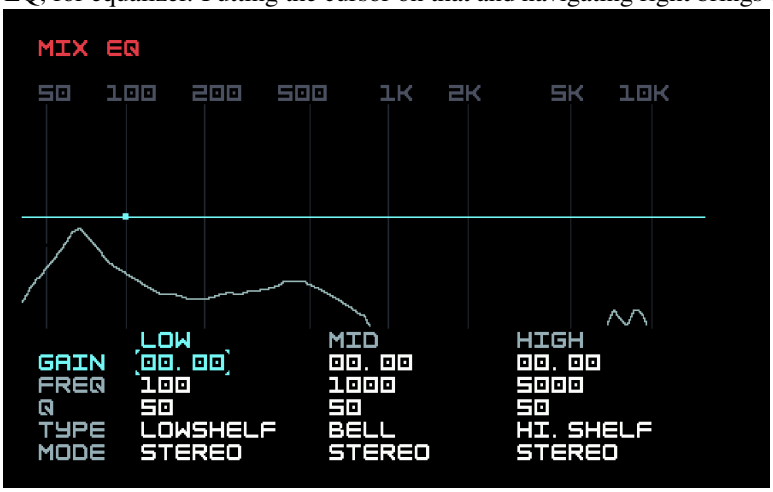
4.4 Performance options

Everything in this section applies to the M8 in general, not just the FM Synth, and by "performance" we don't mean that you have to have an audience, only that you've pressed [PLAY] or are about to. You can start the groove going and then navigate to places where you can make real-time changes that affect the sound (filter cutoff for an instrument, for example). If you want to navigate during performance, you'll need to do it quickly. We've been stressing navigation along the mini-map because it's conceptually easy and usually visible, but there are shortcuts. For example, in Instrument View, there are key combos to move to the next instrument (next highest number) or the previous one. In Phrase View, there are key combos to move to the next or previous phrase in the chain. These are documented in the manual and you should start learning them, if you haven't already, because they will save you time even while composing.

There are some places where more global change is possible. One useful view is Mixer View, which is a V at the bottom of the mini-map and accessible from the main spine. At the top is a global volume setting (we mentioned this very early on) and below it an animation of the levels of each track, with a master stereo animation to the right. Below each track animation is an individual volume setting for each track (defaulting to E0).

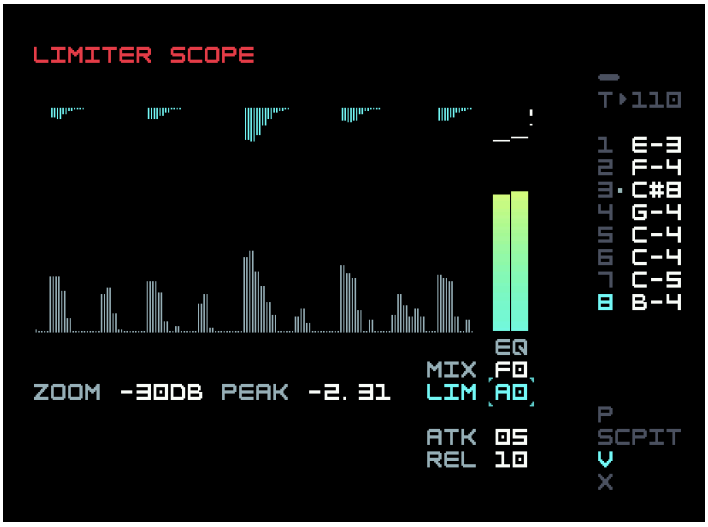


At bottom centre are controls for input level and effect sends for the physical jack and USB audio. At bottom left are global controls for the effect sends (the sum of all the sends from instruments and the above two sources). There is a master animation to the right. Below it, the first setting is for mix volume (used for renders). But above that, you see EQ, for equalizer. Putting the cursor on that and navigating right brings one to the mix equalizer.



This is a three-band parametric equalizer, added in firmware 4.0.0, with a lot of configuration options. You should consult the manual for details. We'll discuss the general use of EQs in the Sampler chapter, but they allow one to emphasize or reduce various parts of the frequency spectrum. A similar equalizer is available for each instrument (as of firmware 5.0.0), though several instruments can share one if desired. These can be used to fit instruments into the mix without clashing. As of firmware 6.6, the waveform oscilloscope view normally at the top was replaced by a spectrum view further down, another feature that was long thought too computationally intense for M8. While you should primarily use your ears to adjust the mix this way, the spectrum view can assist.

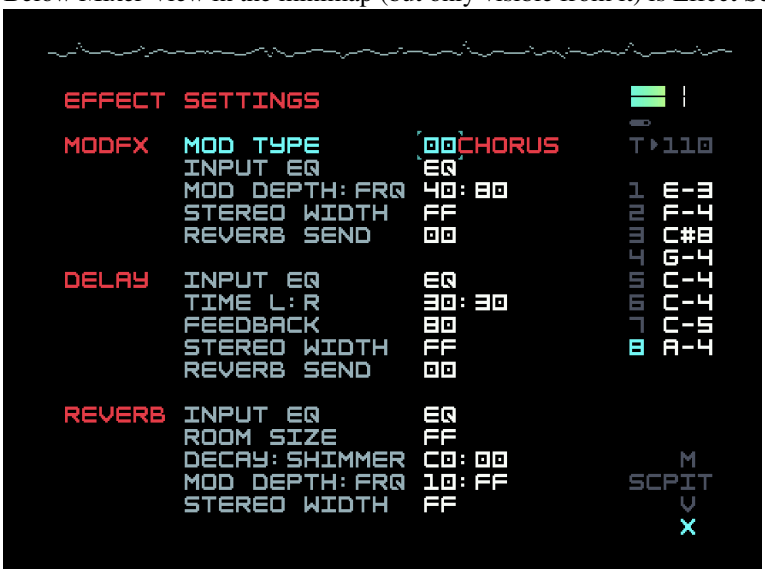
Returning to the mixer view, the next setting is for a mix limiter. The limiter available for each instrument is primarily used for adding various flavours of distortion, but this one functions more like an end-of-chain compressor, reducing the dynamic range of the mix by automatically adjusting the volume. If you're familiar with compressors, this one lacks a ratio parameter and a sidechain input. The effect can be subtle or pronounced, and it is worth investigating with the aid of the manual.



Next in the bottom right column of Mixer View is the setting for a master DJ filter. A DJ filter typically moves from lowpass to highpass, though here several options are possible. 80, the middle setting, has no effect. This filter can be used to good effect with our groove.

While the cursor is on an individual track volume, there are key combos to mute or solo that track, or to clear all mutes and solos (once again we refer you to the manual for details). These key combos work on the single track implicitly selected in Phrase and Chain View, and again on the track the cursor is on in Song View and Live Mode, so it is a good idea to memorize them if you plan to do this sort of thing. Soloing can be useful even during composition, to quickly add and remove context while working on a track.

Below Mixer View in the minimap (but only visible from it) is Effect Setting View (an X on the mini-map).



You can look up what all of these settings are in the M8 manual. The EQ for each effect is a more recent addition. Here, for example, one can keep low frequency sounds from reaching the reverb, which can cause muddiness in the final output.

These settings can be automated, but we're not sure how effectively they can be manipulated in real time on the M8. The delay feedback cannot be taken above 100% to grow and threaten to swamp the sound before it is dialed back, and there is no playable filter in the feedback loop to alter the character of the repeats. These classic dub sounds (maybe less important in dub techno) are not easy to achieve on the M8. (Hint: tables, discussed more in a later chapter, can help.)

We've already discussed how, in Song View, one can start the song playing on any row. There's also a key combo to cue another row, which will start it when the current row has finished playing. This becomes more interesting when we consider the role of empty rows. Previously, we've thought of an empty row as causing the song to start again at the beginning. But if there are non-empty rows below an empty row forming an "island", the island will loop in the same way. So one can create isolated regions and then jump from one to another by a planned action in real time. This is useful not only in performance, but in development, where one can work on a multi-row chunk of a song in isolation from the rest.

Live Mode, which you've probably accidentally invoked also, goes even further, allowing each individual track to be cued for performance on any nonempty row, independently from the other tracks. Even with our little one-row groove, we could create an additional one-row island of variations of each track, and then mix and match top and bottom versions at will.

We'll discuss using other devices alongside the M8 in a later chapter, including MIDI controllers and keyboards. These can significantly expand the possibilities for performance. In the next chapter, we'll shift gears to examine sample creation, manipulation, and playback. There is some overlap with synthesis, but also new challenges and possibilities. We'll also continue to introduce techniques applicable in more general contexts.

4.5 To keep in mind

- You don't need to fully understand the mechanics of the FM Synth in order to make effective use of it. The overview you get by working through this chapter is a good start, along with more experience in the effects of parameters and modulations.
- Simpler algorithms like 1>2+3>4 or even 1+2+3+4 can yield good results.
- A little change can make a big difference when using the FM Synth. Keep your modulations subtle, and if you are stacking operators, use slow sines for the upper ones, at low levels.

5 Crafting beats with the Sampler

A sample is a digital recording of a sound. The Sampler, at heart, plays back such sounds. This is quite different from the synthesizer instrument types such as the FM Synth, which synthesize or create their waveform in real time through computation. The Sampler provides a number of ways to alter the playback of recordings in real time, arguably making it as powerful as the FM Synth, albeit in a very different way.

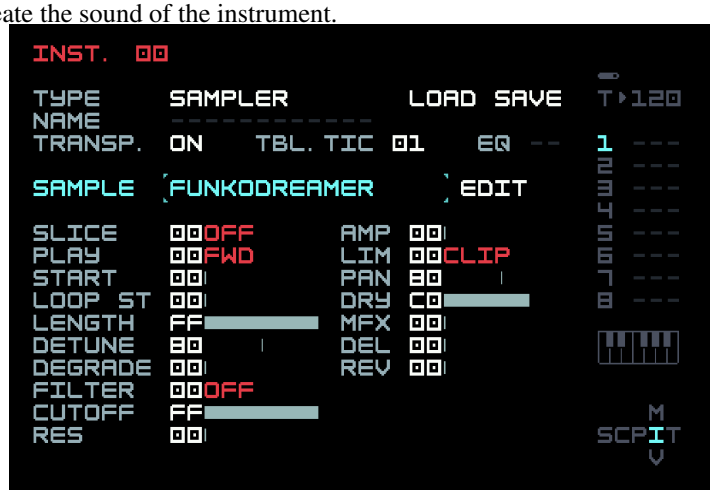
If you have paid any attention at all to popular music in the last several decades (or even earlier), you have heard samples used extensively. Whole new genres of music exclusively or dominantly use samples. If you've used samplers before, you'll recognize many of the features that M8 offers, right away. If you're brand new to such manipulations, don't worry; you've picked a great device to start with.

We said in chapter 1 that the M8 streams samples directly from its mass storage device (the SD card), while many other devices that manipulate samples put them into RAM (fast but volatile memory). One consequence of this is that those other devices have a limit of how long a sample can be (from a few seconds to a minute or so) while the only limit on M8 samples is the size of the memory card. The price to be paid for this is that only a few SD cards have high enough specifications to be usable with the M8. One such is included with it new, but if you have bought one used, or want to use more than one card, check the list maintained on the Dirtywave website. Another issue is that heavy use of samples (across many tracks, changing frequently) can result in glitches or dropouts. This is rare, but be warned, and don't expect miracles.

An analog waveform is continuous, but a digital representation of it is discrete in two dimensions, time and amplitude. The resolution for each of these two dimensions can vary in the WAV format used by M8 for samples. One common choice is 44.1kHz, 16-bit, as used in compact discs. This means 44100 values per second, each one represented by 16 bits (binary digits) with a total of 2 to the power 16 or 16656 possibilities, used to represent points in the range of the waveform (symmetric about zero). (Annoyingly, these values are also called samples; if we need to disambiguate, we'll refer to them as *little samples*.)

Audiophiles argue endlessly over whether 16-bit audio is good enough, but M8's internal audio is 24-bit (as of firmware 6.5), which is definitely good enough. M8 can handle WAV files up to 32-bit (integer, not float), but there are some consequences beyond larger file sizes which we'll discuss as they arise. In the other direction, we've seen that Wavsynth deliberately uses 8-bit resolution for its low-fidelity sound. M8 provides limited ways to change the resolution and sample rate of a sample; computer software typically has more options for this.

You can think of the M8 Sampler as consisting of two parts: the Instrument View, and the Sample Editor. The Instrument View is the first thing you see when you load up the Sampler. Like the other instruments on the M8, the Sampler's Instrument View is a single screen that covers all the essentials of the real-time playback manipulations that create the sound of the instrument.



The Sample Editor goes one layer deeper than the Instrument View. It does not manage playback; rather, it prepares a sample surgically before that. Here you can precisely slice your sample, loop and crossfade regions, change sample rate, normalize the waveform, and much more. This is also where you record samples directly into the M8, whether you're using the audio inputs or, on the M:02, the built-in microphone.



It would be impossible to cover everything the Sampler can do in a short chapter. Here, we'll focus on the basics of sound-shaping and beatmaking, with the goal of helping you achieve musical results quickly. With that in mind, we'll mostly stay in Instrument View, though we'll allude to features in the Sample Editor when applicable, and discuss it briefly at the end. Though we focus on drum samples and loops, the Sampler can also be used to good effect with melodic or atmospheric material.

First, let's talk about perhaps the most important and overlooked part of working with samples.

5.1 The art of sample selection

The M8 ships with a nice assortment of samples to get you started — drums, synth stabs and pads, odd textures. You can be happy for a long time playing with these sounds and seeing how much music you can squeeze out of them. If you bought your M8 used and it's missing the factory samples, email `support@dirtywave.com` and they should be able to help you out.

At some point, you may decide that you want to load some of your own samples onto the M8. The process is straightforward, provided that they are in WAV format (with file extension `.wav`). Eject the SD card from the device and mount it on your computer via a card reader or adapter. Find the "Samples" folder, open it, and drop in your samples. Easy enough. It's even easier using firmware 6.5 or later, since USB Drive mode can be used to avoid physical manipulation of the SD card by mounting the M8 directly on your computer, tablet, or smartphone. However, this is best for small transfers, not backups or transferring large numbers of files. To use it, just connect M8 to the USB host and select USB Drive mode in Project View.

If you're thinking about loading your entire sample collection onto the M8, you might want to consider a couple of things. First, it's generally easier to work with a small, well-curated set of samples than a massive stockpile. The process of sifting through sounds can be time-consuming and uninspiring — perhaps you've experienced the tedium of previewing one snare sample after another, painstakingly sorting through subtle sonic variations until you've forgotten the musical idea you were pursuing in the first place. Sample-based music-making works best when you know what sounds you like and how to access them quickly.

The second consideration here is a practical one. The M8, by default, does not sort large folders of files. They display in an order dependent on the file system of the SD card. If you have a giant folder of kick drum samples that you're familiar with, you may be surprised when you open the same folder on the M8 and see your samples appear out of order as far as you're concerned. Fortunately, there's a quick key combination to sort directories, which is [SHIFT]+[OPTION], but it might take a second or two to process depending on how many files you're sorting. But you still have to navigate a large folder with the limited user interface of the M8, where you can't as easily scroll, point and click. It's only the display that is sorted, and you'll have to do this again on your next visit. There is also a limit of the number of files that M8 will display in any folder (1000).

You might think that organizing your samples into a large tree of folders, with long, descriptive filenames that include the source and BPM, would enable quicker navigation. But there is also a limit of 128 characters on the file path, which includes the file name plus all the names of enclosing folders. So you'll find yourself taking out vowels and rendering everything unreadable when you return to the folder later.

As you're thinking about which samples you might want to add to your device, have a look at the section of the official M8 manual titled "The microSD Card". Here you'll find directions for formatting your SD card and some tips for avoiding "CPU TOO BUSY" errors. In general, you'll get the best performance using 16-bit mono samples, though this isn't a strict requirement.

If you're looking for some new sounds for your M8, we can recommend Legowelt's free collection of sampled synths and drum machines. There's a ton of high-quality material there to explore. For a wider range of noises and field recordings, try Freesound, where you can search and filter according to sound category and licensing level. You'll find a nearly endless supply of recordings available under a Creative Commons license, free to use in whatever way you like.

5.2 Loading and using samples

Now that you've got your samples ready, let's make some music with them.

As we did at the beginning of chapter 4 with the FM Synth, create a new project, add a new chain in Song View, a new phrase in that chain, and a single C-4 note in that phrase, then navigate to Instrument View and choose Sampler as the type. At the beginning of chapter 3, we used Sampler twice, to load a kick sample and a snare sample into two different Sampler instruments. A Sampler instrument can only hold one sample, though we'll see creative ways to get around that later on. One sample may be loaded into several Sampler instruments for different treatments.

The process is the same for any sample: navigate to Load, tap [EDIT], and navigate in the file system to the sample. You can preview a sample by tapping [PLAY] while the cursor is on it, without it being selected; tap [EDIT] to select

it and return to Instrument View, or [OPTION] to get out of the load process without loading anything. If you return to Phrase View and tap [PLAY], you'll hear your chosen sample repeating once per phrase.

You can repeat this process again within the same phrase — create a new instrument by navigating to the "I" column and double-tapping [EDIT], then choose the Sampler, assign a sample, return to your phrase and drop in some notes. As we did in chapter 3, you can also clone this instrument into a new instrument and change the sample, saving a bit of time and effort. But you should keep in mind that each track on the M8 is monophonic. That means if you trigger a sample and then trigger another one in the same track, the new sample will cut off the earlier one if it is still playing. This was not a problem in chapter 3 because the samples were very short, but in general this kind of overlap can happen.

This may seem like a limitation, but it can be used as a creative tool. For example, you can think of each track as its own "Choke Group" or "Mute Group" (interchangeable terms used by Ableton Live and the Akai MPC respectively), a collection of sounds that cut each other off so only one can play at a time. The quintessential use-case for a Mute Group is the grouping of open and closed hi-hat samples as a way to emulate the behavior of a real hi-hat: you hit an open hi-hat and it rings out, then cuts off abruptly on the next hit when it's closed.

Another creative way to make use of the M8's monophonic tracks is to explore the technique of Linear Drumming, where a complete drum beat is played across a drum kit but no hits occur simultaneously, just a discrete sequence of kick, hi-hat, snare etc. This is a cool approach for creating interesting grooves and feels, and also helps create a little space in your mix by giving each drum hit its moment to shine. There are human players who use this technique with physical drum kits, even though they have two drumsticks and one or two kick pedals that can all be used simultaneously.

Of course, if you find the monophonic nature of individual M8 tracks a bit stifling, you can always slide over to the next track and continue layering your pattern there — more on that later.

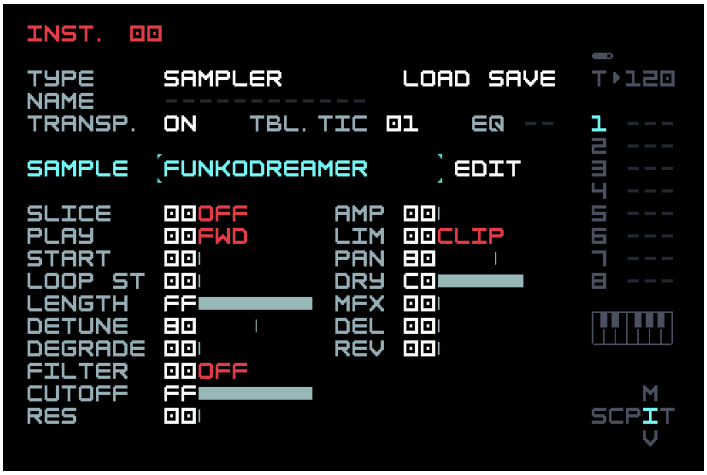
There's something uniquely joyful about working with samples, because it's easy to end up with unexpected results and stumble upon happy accidents. With fairly basic techniques, you can transform the sound of a sample into something that sounds completely different. Let's walk through some of the sample-tweaking tools available on the M8.

First, while we're still in Phrase View, let's focus on a couple of very important parameters right in front of us, ones we're already familiar with from earlier chapters: Note (the N column) and Volume (the V column). Note will typically change the pitch of your sample, which is done by changing its playback speed. Doubling the playback speed increases the pitch by an octave (and also cuts the total playing time in half). Changing the pitch/speed of a sample can be transformational all on its own — for instance, if you pitch a sample down, it will play back at a slower speed, which often reveals interesting audio artifacts or new dimensions to the sound. If you've worked with samples on a computer, you may be familiar with techniques to change just one of pitch or playback speed without affecting the other. These are computationally intensive, and can't be done in real time on the M8.

The bit resolution of the sample can make a difference while changing pitch. Higher resolution places restrictions on how much a sample can be pitched up. A sample with 32-bit resolution cannot be pitched up at all. For maximum flexibility while retaining fidelity, it's best to convert it to 16-bit resolution.

Similarly, adjusting Volume can dramatically change how you perceive your sample as part of a pattern. Our ears tend to pick out different elements in a song depending on how loud or soft they are in the context of a mix, so it's important to listen closely and get your sounds balanced the way you like. You can create lively, dynamic patterns by carefully setting the Volume value on each step of your phrase, even if you're working with simple, static sample sources.

Let's return to the Instrument View screen and go through some of the parameters specific to the Sampler. These all have corresponding FX commands to adjust an instrument on the fly.



PLAY lets you change the sample playback direction and loop the sample. Besides the obvious choices on these (forward/backward and loop/no-loop, or one-shot), there are more advanced ones we'll discuss later. One common technique is to reverse the playback direction on a one-shot sample like a crash cymbal, giving you a satisfying "whoosh" sound that fades in dramatically.

START changes the start point of the sample. At 00, the sample plays from the very beginning, and as you increase this value, you nudge the start point incrementally forward. This can be useful if you want to trim a bit of noise or silence at the beginning of a sample (though you can do this more permanently with the Sample Editor discussed below). It can also be handy if you want to tame a harsh transient — for instance, a drum sample can sound quite different when you start playback after the initial attack.

LENGTH sets the sample length. Like the Start parameter, this can be useful for trimming noise or silence off the end of your sample. If you set your Play direction to reverse, Length will determine when your sample starts (since your sample will start playing back from the endpoint).

LOOP START sets a point that playback jumps back to, when in loop mode and the end is reached (or the length is used up). In reverse mode, playback jumps to this point when the beginning is reached.

DETUNE is a useful companion parameter to the Note you set in Phrase View. With Detune, you can adjust the pitch up or down and apply that change to all instances of your instrument. For example, if you create a melody with your sample using Notes in Phrase View, but later find it's slightly out of tune with another element in your song, you could use Detune to nudge the pitch of your sample up or down while preserving the Notes that make up your melody. Detune allows fine changes (as little as 1/16th of a semitone) but as a consequence has limited range. It will also subtly affect the playback time.

DEGRADE is sample rate reduction. This effect can be pretty extreme and make your sample unrecognizable if turned up too high, but when applied more subtly, can add a nice layer of crunch that brings otherwise sterile samples to life.

The controls in the right column, for filter and effects, are the same as with the synth engines we've discussed. Finally, nestled in the upper right of the screen just below LOAD and SAVE is the all-important EQ effect, which we discussed briefly in chapter 4. Each instrument on the M8 includes a three-band parametric equalizer, which is extremely useful, especially when applied to samples, which may have quite a wide harmonic range. You can use this EQ in a variety of ways, either as a mixing tool or for sound design. For example, if you're working with a sample of a full drum beat but only want to hear the cymbals, you could use the EQ to isolate the high frequencies (you could also use a high-pass Filter to do this — on the M8, there are often multiple ways to accomplish your goals).

To be more precise, the M8 does not strictly have an EQ setting per instrument, but instead has an EQ slot per instrument where you an assign any of 128 saved states. That means you can create EQ "templates" that can be saved and re-used across instruments. For instance, you might find it beneficial to create an EQ slot that simply rolls off some low frequencies, and use that as your starting point when working on any sounds that aren't your kick or bass. That way you get a head start on keeping your mix clean in the low end and avoiding muddy bass buildup.

5.3 Slicing made easy

Let's examine one more useful tool on the sampler's Instrument View.

At the top of the left-side column is the SLICE parameter. This is a handy feature for quickly chopping your sample up into equal-sized slices. This is not the only way to slice samples on the M8 — in the more advanced Sample Editor, you can set up to 128 slice markers with extreme precision, and store these in the sample itself. Still, the Instrument View's Slice parameter is a great way to quickly chop up a long sample, whether you use it with clear intention or as a way to stumble into happy accidents.

Let's play with it. Create a new phrase and a new Sampler instrument. If you have a long sample at hand, you can use it, but we'll provide one for you. One common use of slicing is to slice up drum breaks from old R&B, jazz, or funk records. This can be problematic, because in most cases the material is not in the public domain, and permission has not been obtained. There are drum breaks available that have been cleared, but they're not as compelling and iconic. User Sacco Belmonte on the Dirtywave Discord comes to our rescue. Sacco has recreated some classic drum breaks using drum software on a computer, and made these "cover version" recordings available for free use. Here's a break we can use (right-click or control-click on the player to download the WAV file). You may recognize it, or be able to guess its inspiration from the file name.

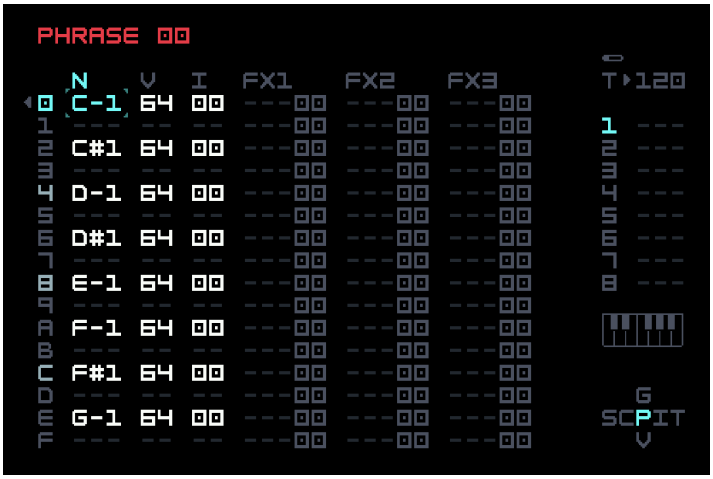
Put the file somewhere in the Samples folder on the M8's SD card, and then load it into the Sampler instrument that you just created. While in Instrument view, hover your cursor over SLICE, then hold [EDIT] and tap [RIGHT] once. The first option you see is 01 FILE which allows you to play back your sample according to custom slice markers that you can set in the Sample Editor. We're going to skip this option for now. (Sacco has actually already done this for us, because slice markers can be stored as metadata in a WAV file, but we'll pretend this hasn't happened for now.)

Keep holding [EDIT] and tap [RIGHT] again and you'll start to see two sets of values appear. The left side shows the hex value for the SLICE parameter, and the right side (shown in parenthesis) is a translation of the hex value that indicates how many slices are created. While continuing to hold [EDIT], tap [RIGHT] until you reach "10 (016)", that is, hex value 10 or decimal value 16. At this setting, you've chopped your sample into 16 equal slices, which is a fairly common approach (think of the 16 pads on an MPC), especially when working with samples that have been trimmed to a musically-useful length. For instance, if your sample is exactly 2 bars long (as this one is), then slicing it into 16 equal parts will give you a slice at each eighth note.

Now that you've sliced your sample, switch over to Phrase View. Navigate to the N column, and tap [EDIT] to insert a note in the first row. By default, the M8 will populate the note C-4 in the N column, as well as the default volume of 64 in the V column (while the I column displays the number of the instrument containing the sample we just sliced). If you tap [PLAY] at this point, you'll notice that you don't hear any audio playing back. What's going on?

When you use the Slice parameter, your slices are mapped to notes starting at C-1. Since we sliced our sample into 16 parts, we'll have slices available across 16 notes from C-1 to D#2, which is not enough to reach C-4. That's why C-4 in the N column gives us no audio back. Using [EDIT]+[DOWN] will decrease this value by octaves, so three such taps gets it to C-1. Tapping [PLAY] now should let you hear the first slice play back in your phrase.

Put the eight notes from C-1 to G-1 on every other row in the phrase.



Tapping [PLAY] gives you a reasonable rendition of the first bar of the sample. It won't sound quite right, because the BPM of your project (which is probably the default 120) won't match the BPM of the sample. The REPITCH option for the PLAY parameter provides a way to adjust this, but we'll just live with it. Since the second bar of the sample is very similar to the first, hearing just one bar is good enough for this brief example, but if you like, you can create a second phrase that uses the notes from G#-1 to D#-2 to play the whole sample. You then can proceed to rearrange the notes, duplicate some of them, move them to previously unused rows, and use retrigger (RET) and the other FX commands you've learned about, as well as the new ones for adjusting Sampler parameters. That's the basic idea of chopping.

What if you want to change the pitch of your sample slices? You can't use the N (note) column, since we're already using that parameter to reference the slices themselves. Instead, you can use an FX command like PIT, which allows you to set a relative pitch change on a given step. You can also make use of the DETUNE parameter in Instrument View if you want to alter the pitch of all slices played back.

Modulation is what brings electronic music to life, and the M8 offers a variety of creative ways to get your parameters moving around in either predictable or chaotic ways. You've seen Instrument Modulation View in chapters 3 and 4, and all of the modulators we saw there are also useful with Sampler, even though the instrument-specific parameters are quite different.

A simple AHD (attack-hold-decay) envelope on volume can work wonders on a sample. For instance, if your sample has a harsh click or pop at the beginning, you might add a bit of ATK (attack time) to smooth it out. Or, if your sample is long but you want to carve out a short percussion hit from it, dial back the DEC (decay time) to isolate a very small portion of the sound. These parameters can be used in tandem with the Start and Length controls on the Instrument View screen to precisely trim and sculpt your samples into exactly what you want them to be. Slicing and looping can also introduce clicks and pops, if the transitions cause abrupt changes in waveform amplitude. Sometimes work in the Sample Editor is needed to fix these.

One frequently-used technique is to assign an LFO to a parameter, then set the OSC of the LFO to Random and TRIG to Hold. With this configuration, your LFO will generate a new random value every time a note is triggered, and will hold that value until the next note. In practice, this means that you can add a little (or a lot) of unpredictable variation every time your sound is played back, with the amount of variation determined by the AMT parameter. At low AMT levels, this can be an excellent way way to "humanize" the feel of one-shot drum samples. Suppose you're working on a drum beat and you want to add some hi-hats. With a bit of Random LFO on something like Filter or Volume, each hit of the hi-hat will sound a little different, just like it does when playing a real drum kit.

5.4 The long and the short of samples

Combining the features of long sample capability and slices may have already given you an idea to overcome the limitation of one sample per instrument. Why not glue multiple samples into one long one, and then slice them up again inside M8 (or place slice markers as part of the glueing process)? In fact, we don't even need slices for this technique, because we can reduce the LEN parameter and modulate START. But slices certainly make it easier. The

long glued-together sample is called a **sample chain**. You can find sample chains of various sorts (drumkits, full ranges of acoustic instruments such as piano) in the same places you can find one-shot samples. User `brian3kb` on the Dirtywave Discord and the Elektronauts forum has created an excellent tool called Digichain that runs in the browser (even offline, once loaded) and constructs, deconstructs, or manipulates sample chains.

At the other end of the spectrum, very short samples can also be quite useful. Consider a sample that is just a single cycle of a repeating waveform, like one peak of a triangle wave, or one tooth of a saw wave. Looping that sample will create a sound like a triangle or saw oscillator.

A sample is not treated quite the same as an oscillator, though. When it is triggered, it plays from the start (or end, or a selected point). An analog oscillator is free-running, and the digital oscillators in M8 synth engines are designed to mimic that behavior. The OSC options for the PLAY direction in the Sampler instrument allow looped samples to be free-running as well. This means that our single-cycle waveform (SCW) sample will act like a digital triangle or saw wave. Of course, we already have these, but other SCWs can expand the sonic possibilities. If you don't use the OSC options, retriggering might cause a pop or click.

We can combine the long and short ideas, building sample chains of SCWs. If the SCWs are related (for example, different pulse widths of a square wave), then manipulating slice choice is like manipulating that parameter on an oscillator. This resembles the use of wavetables (discussed more in Chapter 6). SCWs and sample chains of them are also available on the Internet.

5.5 A note on mixing

Once you have some samples playing across a few tracks, you may want to work on your mix. What's the best way to balance your levels? How do you keep your mix clean when combining a variety of different sounds? We've already talked about using the EQ and Filter to carve out unwanted frequencies and avoid muddy bass buildup. Now let's focus on volume.

There are several points along the signal chain where you can make decisions about volume, beginning at the Instrument level. This process is known as **gain-staging**, gain being a synonym for volume adjustment. We've discussed the AMP, LIM, and PAN parameters in Instrument View in earlier chapters. As you increase amplification, your sample will play back at a louder volume, but its timbre will also change according to the limiter setting you've selected. Play around with these settings until you've dialed in a tone that you like, and don't worry too much at this point about whether your sound is too loud or soft — you'll have more opportunities to control the volume as we continue down the signal path.

Although panning with PAN doesn't directly impact volume, it can change the perceived loudness of a sound in the context of a mix. Keep this parameter in mind if your mix starts to get too crowded — sometimes a bit of separation across the stereo field makes a dramatic difference.

Under PAN is the dry signal level (DRY), followed by the three effect sends MFX, DEL, and REV. In the context of mixing, it's important to remember that these parameters control the amount of signal sent to separate send tracks. In other words, as you turn up your delay send, you're increasing the volume of your signal on another track elsewhere in your mix, so the overall result will be louder. Thankfully, the M8 provides a DRY parameter to keep things in balance. As you turn up your FX sends, you may want to dial back your DRY level accordingly. You can also use DRY to lower the volume while you're experimenting with AMP and LIM, allowing you to focus more on the tone than managing the sound's loudness.

If we navigate out of Instrument View and into Phrase View, we have another way to control volume that's tied to the particular phrase we're editing. The "V" column stands for Volume, and it can be changed on a per-step basis. We talked about using Volume as a sample-sculpting tool for creating dynamic phrases, but it's just as relevant as part of the mixing process.

Finally, if you navigate to Mixer View, you have a dedicated volume control for each of your eight tracks. Changing a track's volume at the Mixer level will affect all of the phrases it contains, so your use of this parameter will depend on how you're using the M8's eight tracks. For instance, if you're writing a song where each track represents a specific instrument (drums on track 1, bass on track 2, synth pads on track 3, etc.) then you may find the Mixer useful as a final stage for balancing your levels and getting the mix just right. On the other hand, if you're using your M8 project to build a live set where each track contains many different kinds of sounds or entire song stems, then you may want

to balance your mix at the phrase level and treat the Mixer as a control surface for fading tracks in and out during your performance.

There are even more ways to control volume. You could return to the Instrument View and apply a filter to reduce a sound's perceived loudness (as well as its timbre). You could animate any of the parameters we mentioned by adding modulators or FX commands throughout your phrases. If you're using the master limiter or the OTT compressor, that will certainly have implications for your mix. You have no shortage of mixing tools on the M8, for both creative sound design and problem-solving.

Here's a brief checklist you can use when you want to work on your mix:

- Is one of your samples too loud or too quiet every time it plays? Go into the Instrument View and adjust the DRY level, or play with AMP and LIM settings to dial in the right combination of volume and tone.
- Are you hearing volume issues on specific steps within a phrase? Go into Phrase View and adjust the "V" column values for any problematic notes.
- Do you want to balance the levels between entire tracks? Or are you preparing a live set where you fade your tracks in and out of the mix as you perform? Drop down into the Mixer view and control your track levels from there.

5.6 Layering and resampling

One of the most effective ways to transform a sample into something else is to layer it with another sample.

Suppose you have a snare drum sample that sounds good but doesn't cut through the mix the way you'd like — you could try layering it with another snare sample that has more punch, or perhaps mix in a clap or percussion hit. Maybe you're working with a synth pad sample that gives you a nice chord but sounds kind of plain — you could liven it up by layering in a bit of noise or a field recording that adds texture. Sometimes the combination of two (or more) things results in a whole new thing that's much more interesting.

So how do you layer samples on the M8? We've already touched on the fact that each track on the M8 is monophonic, so if you want to hear two or more sounds at the same time, you'll need to use multiple tracks. And since the M8 only gives you eight tracks to work with, you need to manage your track usage wisely — if you're trying to write a full song, you probably don't want to spend multiple tracks dialing in the perfect snare drum sound. This may seem like a frustrating conundrum: How do you balance the desire to layer sounds with the practicality of track limits?

Fortunately, the M8 has the perfect solution. It's a feature that allows you to combine many sounds into one, without losing your creative flow. The official manual calls it "Selection to Sample", but you could also think of it as resampling or performing a Quick Render.

Here's how it works: While you're in Song View, use [SHIFT]+[OPTION] to enter selection mode. Then navigate the cursor to highlight a selection of phrases that you want to combine into one sample — you could pick just a couple phrases, or an entire row, or multiple rows. Once you've made your selection, double-tap [EDIT]. Song View temporarily disappears from the screen, and you'll see the "Record selection to sample?" dialog. There are three options: Quick Render, Real Time, and External Only. You want Quick Render, which will already have the cursor on it, so tap [EDIT] one more time to select it, and watch the bottom of the screen as the M8 works its magic.

When it's finished, you'll have a new Sampler instrument added to your Instrument Pool, with a new sample assigned that contains the mix of whatever you selected for Quick Render. If you selected a row of phrases that are all 16 steps long, then the resulting sample will be 16 steps long. If you selected multiple rows or phrases of different lengths, the resulting sample will be as long as it would take to play through the phrases in one full pass before they start repeating. In any case, you'll have a perfectly mixed, perfectly trimmed sample that includes everything in your selection. Brilliant.

It's hard to overstate just how convenient this feature is. The M8 includes a more robust Render process that has its own screen and additional options to tweak. It's very useful, but requires you to stop your music-making and dive into the menu. With Quick Render, you never have to leave your song. All it takes is a few taps of [EDIT] to consolidate multiple tracks into a new sample — the M8 automatically names the file after your current project, and automatically assigns the sample to a new instrument, so you're ready to start using it right away.

To start playing with the new sample, just go into a phrase, slide over to the Instrument column, and tap [EDIT]. The new Instrument containing your Quick Render sample should be the default Instrument that appears, but if there's any doubt, check the name at the bottom of your screen to see if it matches your Project name.

Of course, since your Quick Render sample is now part of a Sampler instrument, it's fair game for further manipulation. Let's say you build up a layered percussion loop over five tracks, then select the phrases and create a Quick Render of the full ensemble. Great. Then you create a new phrase, drop in your new instrument, place a note in the sequence, and tap [PLAY]. There's your percussion loop, just as you heard it before. But why stop there? Now you can move over to Instrument View and start playing with parameters. Maybe you decide to slice this new sample into 16 slices and rearrange them across your phrase. Maybe you add an LFO that randomly modulates DEGRADE. Maybe you spend some time manipulating the sound and building new phrases, only to execute another Quick Render and start the process all over again. Layers upon layers, uninterrupted sample-tweaking bliss. The M8 is truly special.

One thing to keep in mind: In order to use this workflow, you need to have at least one slot free in your Instrument Pool (since the result of the Quick Render process is the creation of a new Instrument). The M8 allows for 128 Instruments per project so you have plenty of space to work with, but if you're doing a lot of layering and quick rendering, you may want to review your Instrument Pool from time to time and clear any slots you don't need anymore.

If you have a few samples on the M8 that you want to quickly make into a sample chain, you can use Quick Render for this. Just set up a phrase that plays each of them, suitably spaced. Of course, you will want to use an external tool like Digichain for more complicated situations.

Most samplers are equipped with some version of resampling capabilities, but for us, the M8's Quick Render feature is by far the best in class. It's fast and easy, but more importantly, it's totally seamless as part of a music-making workflow. It's something you can do without thinking, yet it dramatically expands the capabilities of the M8 and the potential for where your musical ideas might go. And, again, it's as simple as: Select your phrases, double-tap [EDIT] to open the dialog, tap [EDIT] to Quick Render, done.

The existence of Quick Render implies something a bit slower, and the reason for this is to allow more control and customization. Render View can be reached from Project View, and lets you select tracks to render that don't happen to be adjacent, decide for each send effect whether it is used or not, and other useful options. This is more suited for export from M8.

We've talked about resampling, but what about sampling itself? We can record samples on the M8, from the analog audio or USB audio inputs, or from the microphone on the M:02. Chapter 8 goes into more detail on using M8 with external instruments or sounds, and the topic is best handled there. If you can't wait, here's a nice 11-minute video from Jay Maculate that covers the process well. We'll link the video in chapter 8 also to remind you, if you have the patience to wait.

Jay Maculate – How to Record Samples on the Dirtywave M8

The recording options live at the top of the Sample Editor screen.

5.7 The Sample Editor

Editing a sample in the Sample Editor is a bit more intricate than the techniques we've been describing so far in this tutorial. It's about manipulating a lot of information (the entire waveform, and more) in a dynamic fashion. Jay Maculate also has a good video of about the same length as the one linked above, but covering the use of the Sample Editor. We're not going to be able to improve on the combination of this video and the pages on the Sample Editor in the M8 manual, using just static text and screenshots. So we'll content ourselves with some commentary on the uses of the tool.

Jay Maculate – How to Edit Samples on the Dirtywave M8

If you've used an audio editor on a computer before, perhaps Audacity, or the ones built into DAWs (digital audio workstations) like Ableton or Logic Pro, you'll be familiar with the basic idea, which is to define a time-based region of the waveform and then perform some action on it. But M8's Sample Editor is not nearly as full-featured as those. M8 has a small screen and only eight buttons for manipulation. Consequently, it doesn't try to do too much, and tries to do what it does well.



On a computer, you would define the region by clicking in the waveform display and dragging. You can actually do something similar using the touchscreen on M8 (we'll explain how below) but the screen is small, so it's really only good for getting in the approximate vicinity and not for fine work. In the Sample Editor, the region is defined precisely by the two fields in the row labelled SELECT. These show the start and end position of the selected region in number of little samples, where the little sample numbered 0 (displayed as 00000000) is the very first one, displayed at far left.

With your cursor in one of these fields, you can change the value it contains with the usual key combinations, [EDIT]+[LEFT] and [EDIT]+[RIGHT] for small increments, and [EDIT]+[UP] and [EDIT]+[DOWN] for large increments. When you are making small increments, the screen zooms in to show more detail. There is only one level of zoom, and this is the only way to get it. You can hold your chosen key combo and it will repeat, so you don't have to tap constantly. There is one more level of increment available, which is tempo-based. [OPTION]+[LEFT] and [OPTION]+[RIGHT] will move by one step (in a phrase); [OPTION]+[UP] and [OPTION]+[DOWN] will move by four steps. You can also hold [EDIT] and tap or drag on the touchscreen to change the value under the cursor (try it and see if it suits you).

We talked about how slice markers can be stored as metadata in a WAV file. It's possible to define and store a loop region in the same way. This can be much more precise than using the LOOP START and LENGTH parameters in the Sampler instrument, but it can't be modulated on the fly. If you have having trouble with pops or clicks while looping, you can use this precision to choose start and end points with zero or very low amplitude. When a loop region is defined, it will override LOOP START and LENGTH (once a looping play mode is chosen).

There are also two fields for slice markers, one for the slice number (starting at zero, displayed as 00), and one for the start position. The slice markers are displayed on the waveform (if you open up the sample we provided above, you'll see the slice markers that Sacco put in). The end position is the start of the next slice, or the end of the sample. Among the processes that can be applied to a region, there are some quick slicing options. You can apply even slicing into a given number of slices, as you can in the Sample instrument, but you can also slice by transients (rapid increases in amplitude, like at the beginning of drum hits) or by silence. The slice markers obtained in these ways can be adjusted manually.

Most of the processes that can be applied are straightforward. You'll notice there is an UNDO option, which is rare on the M8, but it is only one level, so don't be hasty. At any point, you can preview the sample, selected region, or slice that you are working on by holding down [PLAY]. Fade in and fade out are good for taming pops or clicks at the beginning or end of a sample, and loop crossfade (XFADE) can soften the transition from the end of a loop to the beginning (which can be jarring even if no pop or click is created).

Finally, at the bottom there are options to save (a new sample, whose name you will be prompted for) or overwrite (the sample you loaded). It's usually best to create a new sample; you might change your mind about an edit, and you can clean up unused samples later.

5.8 Bundling

Project files do not contain samples. For archiving, or for sharing a song with others, it is best to create a bundle. Project View has a line labelled Export/Share which has an option to do this, and the new bundle will be a folder in the Bundles folder of the SD card. That folder will contain the project file as well as two subfolders, one called Samples and one called Instruments.

There is one significant difference between a bundle and the project files we have created so far. The Sampler instruments we built use what are called **absolute references** to samples, that is, a description of where to find the sample that starts from the very top of the filesystem tree. These references can become obsolete if we move samples around or rename them, and we would have to reload each sample into its respective instrument.

In a bundle, the Sampler instruments use **relative references**, assuming there is a subfolder called Samples in the same folder as the project file, and pointing into that. This means that a bundle can be moved anywhere on your M8 and it will still work. More importantly, it can be zipped up and sent to someone else, and they can put it anywhere on their M8, and it will work. They don't have to put the samples into their main Samples folder, and possibly deal with name clashes or organizational issues. You should keep in mind, though, that a bundle is a complete, self-contained copy of a song, and if you make any subsequent changes to the song, they will not be reflected in the bundle.

Many M8 users who share audio of songs on the Internet (some on the Dirtywave Discord, some on streaming services) include the bundles, and you can load them into your M8 and learn from them.

5.9 To keep in mind

- Samples and synths complement each other nicely, as we saw with our very first song. If you're primarily a synth person, keep in mind that field recordings or foley sounds (perhaps created with the M8's own microphone) can add complexity that is hard to match with synthesis. If you're primarily a samples person, don't forget that the space of parameter modulations is much larger for synths, perhaps most evident in longer sounds, like pads or drones.
- The ease of sampling into M8, and resampling internally using Quick Render, are superpowers that other devices wish they had. Spend some time with these; they could become important parts of your toolkit.
- M8 is a very capable device, but it is small and computationally limited. There's no shame in taking a sample out to a computer for more detailed work or things that are easier to achieve there, before returning it to the M8. USB Drive Mode can speed up this process.

6 More about instruments

The FM Synth and Sampler are the sound creation powerhouses of the M8. But they each have their limitations. Sound design is not quick or straightforward on the FM Synth. A small change to a parameter can have a large effect, so it can be difficult to modulate over a range of related, pleasing sounds. While it has a large range, having only four operators and limited modulators means that some sounds are just not possible. The Sampler, on the other hand, can play any sound that has been recorded. But one has to find that recording, either by searching on the Internet or through one's own sprawling and poorly-organized collection. The set of transformations that can be applied to a sample is limited.

Each of these two powerful instruments compensates in part for the deficiencies of the other, but there are still circumstances where the remaining instrument types can be useful, for a particular sound, or for the speed and ease of deploying them. In this chapter, we'll discuss the remaining sound generation types: Wavsynth, Macrosynth, and Hypersynth. We saw each of these briefly in chapter 3, but we'll look at them in more detail. We won't go as deep as we did in chapters 4 and 5, because these instruments are more specialized, and the M8 manual does a good job of covering them. We'll confine ourselves to overviews and some tips on use.

There are two instrument types that do not directly generate sound on the M8, but are used to control other instruments. The MIDI Out instrument can send sequencing information to another device over a cable, using the global MIDI standard. The External Instrument is a variation on MIDI Out for use when you want to bring the sound made by that other device back into the M8 to send to the effects (chorus/phaser/flanger, delay, reverb) and mix with

the sounds on other M8 tracks. MIDI generated by another device or controller (such as a keyboard) can also be used to sequence an M8 track using one of the sound engines above.

Before we get into the details of these instruments, we'll discuss some general topics that apply to all of them.

6.1 Instrument modulators

In the previous chapters, we mostly used AHD (Attack-Hold-Decay) envelopes, because each of these stages have duration measured in ticks, so we didn't have to worry about turning notes off in the phrase. We also used some synth models that had an envelope built in (such as the pluck sound in the first song, from Macrosynth) and some samples whose sounds had specific duration (such as the percussion in the first song).

The ADSR (Attack-Decay-Sustain-Release) adds a fourth parameter, sustain, which is not a duration, but a level. The envelope amplitude falls to this level during the decay phase and stays there for the duration of the note. When the note ends, the release phase starts, and the amplitude falls to zero. There is no release phase if the note ends because a new note is triggered, only if the OFF command is used. The diagrams in the M8 manual may be of use in visualizing this. ADSR envelopes were included in Moog synthesizers from the mid '60's, and are common on polyphonic keyboard synthesizers. If one plays a piano note and holds the key down, the amplitude is somewhat like ADSR, but it will decay slowly during the sustain phase. Some synthesizers (such as the DX series, but not M8) have additional parameters to implement this.

The drum envelope was added by Tim when he was working on his FM drums pack. It is reminiscent of a diagram in Chowning's 1973 paper showing how a drum sound can be made from a modulator-carrier pair by adjusting the index (modulator ratio in M8 lingo) with an envelope whose amplitude starts off in the middle, swells, and then decays to zero. Tim has added a sharp attack and partial decay phase with both duration and final level controlled by the Peak parameter; the Body parameter for the swell duration; and the Decay parameter for the duration of the final decay. These durations are measured in milliseconds. The drum envelope thus has aspects of both AHD and ADSR envelopes, and its primary use is on the ratio of FM modulator operators. But, of course, it can be used to modulate all the other destinations as well, and on instruments other than FM Synth.

The trig envelope is like AHD, but it is bipolar, and one chooses an instrument or track as the trigger source (usually a different one, but that's not necessary). The primary use case is for "ducking". This is an effect often caused by the use of a compressor, which limits the dynamic range of sounds. When there is a sharp increase in volume, such as caused by a kick drum hit, it can cause the compressor to turn down the volume, which sounds like all the already-playing instruments "ducking" to let the kick drum sound. This is highly prized as a dancefloor rhythm. Absent a compressor, one can simulate the effect to some extent using trig envelopes with negative amount to turn down the volume of other sound sources on kick hits. Again, since it is a general effect, you can think of other uses. (M8 did not have a compressor when the trig envelope was added, but in the 6.2.0 firmware release, the OTT compressor was added to Mixer View.)

More complex envelopes can be created using tables, as we'll explain in a later chapter. The same is true for LFOs.

Each envelope shape is a different instrument modulator type. In contrast, there is only one LFO type, but several different shapes and behaviours can be configured within it. Once again, there are diagrams in the M8 manual to help you. We've mostly been using the triangle shape, which is a smooth back-and-forth with abrupt changes at the extremes. The sine shape slows to a halt at the extremes, which can make a subtle difference in some circumstances. There are other shapes familiar to us from audio-rate oscillators: ramp up and down (like a saw wave), and square up and down (differing on whether it starts at zero or finishes there). Then there are exponential down and up shapes, and finally two random shapes. "Random" is stepped; a pseudo-random source is sampled and held for the duration of the cycle. But "Drunk" is continuous, like an inebriated person trying to walk home.

There are four LFO behaviours, two looping and two "one-shot" (that is, the shape is only traversed once). The two looping behaviours differ in whether or not the LFO resets to the start of its shape on each trig. In Free mode, it does not; in Retrig mode, it does. The two one-shot behaviours differ on what value is held after the shape is traversed. In Once mode, the final held value is the start value; in Hold mode, it is the last value of the shape.

LFOs are unipolar except when applied to two specific destinations, Pitch and Pan. We described ten shapes above, but there are actually twenty choices for the shape setting, because the first ten have periods measured in steps, but they repeat with the second ten having periods measured in ticks. A "T" is added to the end of the shape to remind

you of this.

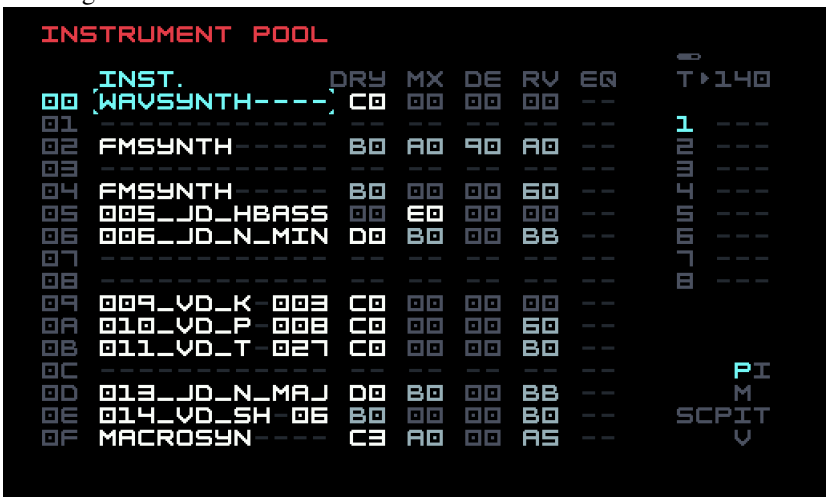
The final instrument modulation type is Tracking. This lets one create a numerical value from note frequency or velocity that can be applied to the standard set of destinations. There are actually three choices, the third one disconnecting the velocity from affecting volume like it usually does. These choices are made with the Src (source) setting. Two additional parameters, lval and hval, specify the points in the source to map to zero and max amount respectively. That is, you can have the tracking apply to only a subrange of the source, and if lval is greater than hval, the modulator will decrease as the source increases.

The most common destination for this modulation is probably filter cutoff. If you are applying, say, a lowpass filter with some resonance, and not using tracking, then as you play higher notes, the resonance will no longer have as much effect, and the volume will start to drop off. What you really want is for the filter cutoff to track note frequency, so that the effect on the higher notes remains the same. User *impbox* on the Discord has figured out the magic values to keep note and filter aligned: cutoff 20, tracking amount 82, lval 00, hval 54. You might have to do some experimentation to find similar values if you are putting this feature to another purpose.

Another useful destination is modulator amount on the FM Synth. Higher notes don't need as many even higher harmonics added to them, and to properly simulate some real-world instruments, the FM modulation needs to be reduced. This can be done with a tracking modulator whose source is frequency (an example of where you want lval to be greater than hval). Using this kind of modulator with velocity as the source and filter cutoff as destination can give a similar sort of damping effect where notes of higher frequencies are less complex (a piano works this way, for example).

One modulation destination that is worth highlighting is another modulator, known colloquially as a "mod mod". A common use is to apply an LFO to the frequency of another LFO to turn it into an even more complex modulation. There wasn't really space in the user interface to allow specifying any other modulator as the destination, so if this type of destination is chosen on modulator 1, then it can only affect modulator 2. Similarly, modulator 2 can only affect modulator 3, and so on in a circle, so modulator 4 can only affect modulator 1. If the destination is Mod Amt, then the amount of the destination modulator is reduced by a multiplicative factor specified by the mod mod. The destination Mod Rate will affect all time parameters on the destination modulator, again in a multiplicative fashion. Finally, there is Mod Both. If you only want to affect one parameter, such as decay but not attack, you will need to use FX commands in a phrase or table. One example of using the multiplicative factor is to create a really small modulation, such as a Detune amount for an FM carrier that is greater than zero but less than 0.01.

While we are talking about things visible in Instrument View, it may be a good time to note that navigating up a screen gets us to Instrument Pool View, introduced in firmware 4.0.0. It is visible in the mini-map as a P above the M representing Instrument Modulation View. Here is the Instrument Pool View for the demo song for M:02.



You can see that an I has appeared in the mini-map to the left of the highlit P representing the current position, meaning you can quickly get to Instrument View for an instrument listed just by putting the cursor on it and navigating right. Navigating left from there will return you to Instrument Pool View. The view also lets one control all dry volumes, as well as modfx, delay, and reverb sends, and change or quickly access the assigned EQs for each instrument.

There are button combos to reorder instruments in the table, and M8 takes care of renumbering all instrument references elsewhere, if you change the number of an instrument this way. There are also two other ways to get to Instrument Pool View. For details, and more useful shortcuts, we refer you to the M8 manual.

6.2 Scales and transposition

We have already seen a few places where transposition (moving all notes by a set amount) is made easy on M8. We used it for an easy variation in our first song, because there is a transposition column in Phrase View, and we've also seen that there is a similar column in tables, though we haven't used it yet (see chapter 7).

If you don't know any music theory, this is not the place to learn it, and if you do, you will likely find something annoying (hopefully not everything) in the brief overview that follows. But we'll persevere.

We described in chapter 3 the mini-keyboard display visible in several views in terms of a single octave on a piano keyboard, the lower row being the white keys (with names like C, D, E, F, G, A, B) and the upper row being the black keys (names modified like C-# or C-sharp on the M8, but also called D♭ or D-flat). We also alluded to our songs in chapters 3 and 4 being in the scale of C minor, because of a few sharps/flats we used.

If we stick to the white keys, and furthermore make C our **root note**, the place from which we depart and return to, then we are using the scale of C major. But if we do the same thing, but make A the root note, we are using the scale of A minor. The difference can be heard by playing the seven note sequence C, D, E, F, E, D, C (in C major), and then the seven note sequence A, B, C, D, C, B, A (in A minor). Major scales sound more cheery and sunny; minor scales more dark and gloomy. Of course, this is not a firm distinction; one can write a lovely uplifting melody in a minor key, or something miserable in a major key. The difference between major and minor scales is in the placement of the one-semitone jumps as opposed to the two-semitone jumps. (This discussion is complicated by multiple variations on minor scales, at least one of which varies going up versus going down, but never mind about that now.) To get to the C minor scale we used earlier, we just move the root from A to C; that is, we transpose every note up three semitones.

If we take a melody written in C major (using only white keys) and transpose it up one semitone, we potentially use all five black keys and two white keys within an octave. (This is the key of C# major.) When we used transpositions for variations in chapter 3, we were careful in our choice of how much, so we only needed to transpose the pluck and bass, and nothing else. If we had transposed our two tracks up one semitone, it would have sounded bad, unless we transposed everything, in which case it would just sound a bit odd, and only in the transition. Some other transpositions (of only some tracks) sound reasonable and yet others sound bad again. We could smooth things out a bit if we insist that a transposition respect the scale; in our example, that we move any notes that land on black keys to the nearest white key (with some systematic way of choosing when there are two options). In a nutshell, this is a feature that M8's extensive scale settings provide.

But first, transpositions. In addition to the transposition columns we have seen in chains and tables, there is a global transpose in Project View, with an option in Track View to exclude a particular track from this. Global transpose is intended for use in performance, and an example of something you might wish to exclude is a percussion track. We have also seen a very local transposition with the PIT command, which we used to build the chord stab in chapter 4. All of these (and others to be described later) will respect the scale options we will now describe.

When in Groove View (which, remember, is above Phrase View), the mini-map shows an S above the G, which is Scale View. Navigating up gets us to Scale View.



This looks complicated, but the important thing to notice is at the bottom, where there is a name field and options to load and save. While you can make your own scales (and we'll explain how), there are 92 factory scales provided, including variations on major, minor, and pentatonic (based on the five black keys in an octave), and the seven Greek modes. These provided scales aren't built in, any more than the factory instruments; they can be deleted, modified, or tweaked and variations saved.

Sixteen different scales can be available in a project. The default scale (global across the project) is scale 00, and in a new project this defaults to C chromatic, that is, all twelve familiar semitones are available. This is the scale we've been using this far. There is a row in Project View where the global scale can be set, and Scale View for that scale can be reached from there also.

The top row, labelled "Key", specifies the root note of the scale. Changing it also changes the top row of the 12-semitone table that follows. That table has note labels in the first column, and an enable/disable switch for each note in the second column. To build a C major scale (instead of just loading it from the factory scales), we could switch off all the sharp (#) notes. The last column is an offset in semitones, with separately-editable fields before and after the decimal point (as we had with FM ratios). By setting the offset in the C row to 02.00, we could make a C entry in a phrase play as D, if we wanted to. It's probably more useful to edit the field after the decimal point, if you want to do microtonal work. At the bottom, just above the name field, is an option to change the global tuning, with the value of A-6 given in Hertz (default 440).

We wrote our previous songs in the key of C minor, even though we didn't specify this (we were using the default C chromatic scale). We just avoided using the notes that weren't in C minor, and you can do the same. The value of using an M8 scale is more evident when doing transpositions or altering pitch with randomization. While sticking to a major or minor scale will not guarantee consonance, it makes more pleasant sounds more likely. You could guarantee consonance by building a "scale" that only has notes of a pleasant chord, or even just one note (quantization to octaves). Scales can be set on a per-track basis by using the FX command SCA. The value provided will specify both the scale and the key. The SCG command will set the global scale.

6.3 Wavsynth

Although the amplitude resolution of M8 is 16 bits (so 2 to the power of 16 or 65536 different values are possible), Wavsynth is an 8-bit synthesizer, meaning it deliberately only uses 2 to the power of 8 or 256 different values. You can see the effect of this using the oscilloscope, if you listen to a waveform like saw. It looks like a staircase rather than a ramp. Those vertical edges add more higher harmonics to the sound, making it seem brighter, as if coming out of a small, inexpensive speaker. These higher harmonics can be filtered away, but you will not reclaim clean waveforms, for the most part.

We deliberately chose Wavsynth for our first melodic sound in our first song in chapter 3, though we carefully chose a waveform which looks the same in either resolution. It's true that Wavsynth is primarily intended for chip-tune. However, it's entirely possible that you could find a use for it in other contexts, especially in a mix with other

instruments.

Wavsynth is simple enough that you can explore it on your own, with the help of the M8 manual. One thing you may find puzzling is the inclusion of three extra filter options. These are essentially 8-bit filters, as they alter the waveform in an approximation of what a filter would do, while conforming to the 8-bit restriction. Consider them as just another way to alter the sound.

Added in firmware 4.0.0 are a number of shapes which are **wavetables**. A wavetable is a series of related waveforms, like frames of a movie. The idea is to modulate the sound by changing position in the table. The Scan parameter is the way to do this. Earlier firmware only offered the Overflow shape which just used the contents of whatever was next in the firmware (with mostly noisy effects). If the filter is off or set to one of the WAV filters, the oscilloscope is also off, and shows the static waveform. This is useful for on-the-fly selection without having to look at the graphic depictions in the manual.

6.4 Macrosynth

We've already mentioned that Macrosynth uses code from the Eurorack module Braids by Mutable Instruments. This is a good choice for a number of reasons, not only because it is a reuse of open-source code that was very popular in another synth community, but because a Eurorack module also has to cope with user interface limitations due to size, and code for digital Eurorack modules also has to respect the modest processing chips used in their builds.

The original Braids module was close to the size of the M8 (13 x 8cm), with a 4-digit seven-segment display, one push encoder for editing, six knobs for control, five input jacks and one output jack. We've already seen the "macro" approach, where one parameter controls several others, used with the op-mods in FM Synth, but there we had some control over configuration. With Braids and Macrosynth, for better or for worse, we have only two major controls (Timbre and Color) and no say in how they affect each model.

This also leads to a certain vagueness in the documentation, because we don't know what those hidden model parameters being controlled are or what they do, and part of the point is to not worry about things like that. Neither the official Mutable Instruments documentation nor the unofficial illustrated guides and cheat sheets created by the Eurorack community are much more illuminating than the M8 manual, which has a main section on Macrosynth plus a description of each model in an appendix. The bottom line is that you have to try each one and see what it can do for you.

The 48 models (chosen by the Shape parameter) can be divided into a number of categories: virtual analog (VA), digital, vocal, additive, FM, physical modelling, percussive, wavetables, noise.

"Virtual analog" is a term used to refer to software emulations of classic analog synthesizers, usually using subtractive synthesis, where a restricted class of waveforms is available for later processing with voltage-controlled amplifiers and filters. It's worth noting that the VA options here cover most of what Wavsynth can do, but in full 16-bit fidelity (and what isn't covered here is covered further down). The VA model range from 00CSaw to OF Saw Comb.

The "Digital" models aren't trying to simulate analog, but rather simulate classic digital, somewhat reminiscent of Wavsynth. The models here include 10 Toy and 11 - 14 Digital Filters, which resemble the emulated filters of Wavsynth, but again in full fidelity.

The "Vocal" models use a technique called formant synthesis, based on the idea that three bandpass filters tuned to specific frequencies can sound like elongated vowels. This is not lifelike speech, but something vaguely reminiscent of warmup exercises a choir might do. The models are 15 Vosim through 17 Vowel FOF.

There is one "Additive" model, 18 Harmonics. Subtractive synthesis takes a harmonic-rich waveform like saw and subtracts harmonics using filters. Additive synthesis adds harmonics to the fundamental sine wave, a technique that is expensive in the analog realm but feasible digitally. With only two parameters to control the added harmonics, the possibilities are limited, but it's worth noting that, in contrast to FM Synth offering a fixed choice of operator waveforms, the Elektron Digitone has a parameter that morphs through additive waveforms in a manner similar to this model.

There are three models that use two-operator FM, from 19 FM to 1B Chaotic Feedback FM, and you may enjoy contrasting these with FM Synth.

Physical modelling is probably the most compelling of the categories offered by Macrosynth, though one has to keep in mind that the computational resources are limited, so one will not get startling fidelity, but rather something

that gets reasonably close to the real thing. We've already put 1C Plucked to good effect in the lead of our first song in chapter 3. The other models are 1D Bowed, 1E Blown, 1F Fluted, 20 Struck Bell, and 21 Struck Drum. You may enjoy comparing these last two to FM Synth, but there are no equivalents for the others.

There are three models in the "Percussive" category, and they are not included in the previous paragraph because they do not emulate physical instruments but electronic ones, in this case the Roland TR-808. The models are 22 Kick, 23 Cymbal, and 24 Snare, and you might enjoy comparing these to the samples from the original included in the M8 factory content. These can be tweaked in ways that a sample cannot (hence the number of samples included in the content), so they may be more amenable to modulation or automation.

The next category is "Wavetables", and at least some of these have been ported to the 8-bit Wavsynth by Tim, based on the brief labels seen on the M8's screen. They range from 25 Wavetables to 28 Wav Paraphonic. This last model is one of the few places where chords can be built on M8 in a single track (the others are the multi-carrier FM Synth models, used for example in the dub stab in chapter 4, and Hypersynth below).

Finally, Macrosynth offers several types of noise, from 29 Filtered Noise through to 2F Morse Noise. Putting these through a resonant filter, distortion via a limiter option, and send effects can be rewarding, especially with lots of modulation.

That's a lot to explore, so take your time doing so.

6.5 Hypersynth

Hypersynth was the last synthesis engine to be added to M8. This was possible mostly because the saw waveform is very simple to compute, and algorithms managing collections of closely-related saws have been developed. The range of sounds that Hypersynth can make is limited, but they are very compelling sounds. You've already heard it in the pad in the first song in chapter 3.

Hypersynth offers twelve saw voices, in six pairs. Each pair can be tuned to a separate note, as we'll explain. The two voices in each pair can be simultaneously detuned using the Swarm parameter, and the voices can be spread in the stereo field using the Width parameter. This sounds quite nice. At some point, the detuning becomes dissonant, but if you keep increasing it, it approaches a consonant interval again. There is a sub-oscillator available, either two octaves below the root note (the one specified in the phrase) or one octave below, and the choice and amount is controlled by the Subosc parameter.

Each of the pairs can be tuned to a different note using the six fields in the Chord row. Hypersynth is scale-aware, that is, one can specify a scale from the project pool, or use the global scale. The note choices are relative to the scale. For example, if the scale is the default C chromatic and a C-4 note is used in the phrase, then to play an E-4, one would use 04 (four semitones) in a chord field. But if the scale is C major, to play an E-4, one would use 02 (two notes up, C - D - E).

Six notes in a chord is a lot, and you might not want to use them all at once. The Shift parameter provides an alternate approach, cross-fading between the first three pairs and the second three. So one could put a different triad (three-note chord) on each, and then switch from one to the other, or smoothly cross-fade (if the chords permit this mixing). You can also clear any voice (using [EDIT]+[OPTION]) and it will not sound.

Each instrument of Hypersynth type can store sixteen chords, and there is an FX command (CRD) to choose among them in a phrase or table. It's a bit of work to set up the chords, but copy/paste works well on them, and you can look for community instruments already set up, though since one can't name chords, you will have to decipher them from the numerical information shown, unless the source has additional documentation.

There were many requests, since the introduction of Hypersynth, to add more waveforms. But since the supersaw algorithms tend to be specific to saws, it did not seem possible. However, Tim worked out a way, and in firmware 6.6, a number of alternate waveforms were introduced, including some variations on saw, square, triangle, and sine.

Although we have now described all the opportunities for polyphony within an M8 instrument, it is still possible to coordinate multiple instruments, as we'll describe in chapter 9.

7 Working with tables

Tables have a reputation for being difficult to grasp, but this doesn't have to be the case. Possibly they are just underdocumented. When people are given a powerful, general mechanism, some of them try to take it as far as they can go, and the results can look really complicated in demo videos. But there are many simple uses of tables. LSDJ, the Game Boy software that was the inspiration for M8, does not have an equivalent of Instrument Modulation View for envelopes and LFOs; there, modulation is done with tables (some instruments have envelopes in recent versions). We'll show you how to use tables to get extra modulators and LFOs in M8, or more complex ones.

We saw tables briefly in chapter 4, where we used one to have more than three FX commands on a step (to specify a chord for the dub stab). There, we described a table as "a parallel sequencer running at tick speed", and described how each instrument has an associated table, plus there are 128 unassociated tables. The instrument table will be started with every note trigger. We instead used an unassociated table, which we started in parallel with a TBX command on a given note. That allowed us to build different chords on different notes. There is also a TBL command to change the table of the current instrument.

When we wanted to time drum hits between steps in the first song in chapter 3, we talked about there being six ticks to a step, and about FX commands that timed events in ticks. The default tick rate for an instrument table is 01, meaning six rows of the instrument table will fire for each step of the phrase. That is, the table sequencer is moving six times as fast as the note sequencer. But this can be changed with a setting visible at top right in Instrument View. When the table is changed with TBL, the new table uses this tick rate also.

The default tick rate for a table started in parallel with TBX is 06, that is, the table sequencer runs at the same speed as the note sequencer. We didn't notice this in chapter 4, because we only used two rows of the table, and the second row immediately jumped back to the first using a THO command (to hold the pitch we chose for our chord for the duration of the note). It didn't matter what speed the table was running at.

The tick rate can be changed with an FX command. TIC in a phrase will set the tick rate of the associated instrument table. There are a few commands that act differently in phrases and tables, usually for good reasons, and we will try to point this out when we use one. TIC is one of these. In a table, it will set the tick rate of that column of the table. Yes, the three columns in a table can run at different rates! One usually wants to do this right away, but the first row is in high demand for other commands to be run at the same time as the note triggers. So there is a special way to set the tick rate for a column: put the TIC command in the last row. THO jumped all playheads to a specified row; we can move just one playhead for one column with a HOP command. HOP works differently in tables than in phrases, and this is true for a few other commands, which we'll point out as we go. THO is another example; when used in a phrase, it starts the instrument table on the specified row, or an auxiliary table if one is specified with TBX on the same phrase row.

There are five special tick rates. TIC00 sets the table to step per triggered note. That is, if there are two notes in a phrase, then the table will move one row exactly when each note triggers. There is a special case within this special case: TIC00 in row F of a column in a table started with TBX will step only when a TBX for that table is triggered in a phrase. This allows for very slow movement through a table, if desired. On the other end of the time scale, TICFF sets the table to run at 200 Hz, that is, one row every five milliseconds. This is the way to get "microtiming", or resolution better than 1/6 of a step, but there is only one choice of fast speed. Someone called it "the speed of light on the M8". TICFE is a recent addition, that isn't really a time-based rate at all; rather, it changes the behaviour of a table to something called a "note map", which we'll talk about later on. Similarly, TICFD is a "velocity map", and TICFC is an "octave map".

That summarizes the basic tools we need to work with tables. Now we need to see how to combine them with other FX commands to achieve our desired results. It's important to keep in mind that there are two types of commands: absolute and relative. Absolute commands just set a parameter to the given value. Relative commands change the value that the parameter already has by a given increment/decrement. If you forget this, your tables might not do what you expect.

7.1 Arpeggios and their variations

If you are familiar with arpeggiators on other synths, it might help to think of a table as a generalized arpeggiator. If you're not, the metaphor will not help, but you're about to learn what an arpeggiator is. Either way, you've probably

seen the ARP command, because it's the default inserted when you first edit a command field, but that is a very specific kind of arp used in early game devices as a chord substitute.

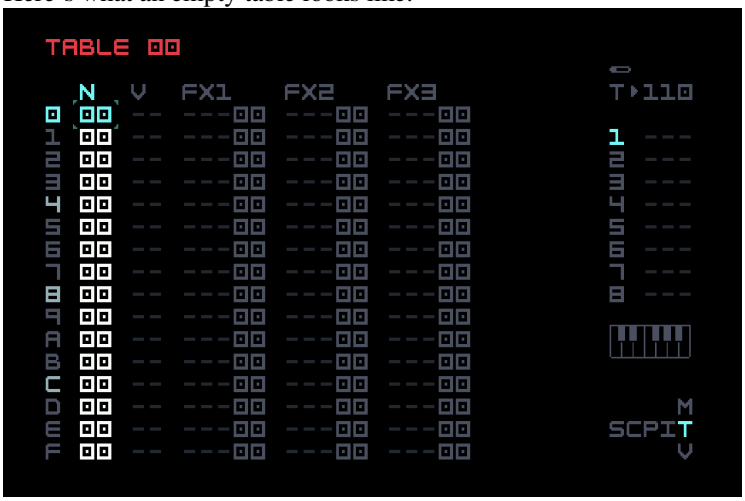
An arpeggio is a chord, but played one note at a time instead of all together. Some keyboard synthesizers have an arpeggiator feature, and when it is engaged, the behaviour of the keyboard changes. One can hold down several notes, and they will play one at a time, and then repeat. The order and speed of the sounds can be configured. Holding down notes is a lot easier than playing them over and over.

We can't use a table to build an arpeggiator that works like on a hardware synth, because a table only gets information about the note that triggered it. There's no way to send it additional information. Instead, we'll use the N column in the table, which transposes the triggering note. We can build a specific chord this way. To change chords, we will have to change tables, or split the table into sections and use THO to land in the relevant section.

If you want to play along, create a new, empty project, with one chain containing one phrase, and that phrase containing a single C-6 note in row 0 with an instrument of your choice. Set that instrument up with a simple, short sound, say FM Synth in its default configuration, with a short volume envelope.

It's easiest to use the instrument table, but more likely in practice that you would use an unassociated table for this task, so you could be selective about when the arpeggio plays, and also change the chord (as discussed briefly above) from time to time. In the FX1 column for the single note in Phrase View, add a TBX80, and use that to navigate to table 80.

Here's what an empty table looks like.



The N column is an offset from the note in the phrase that triggers the table (similar to the transposition column in Chain View). It respects the current scale, so it counts in semitones if the default chromatic scale is used, but in general it counts in scale steps.

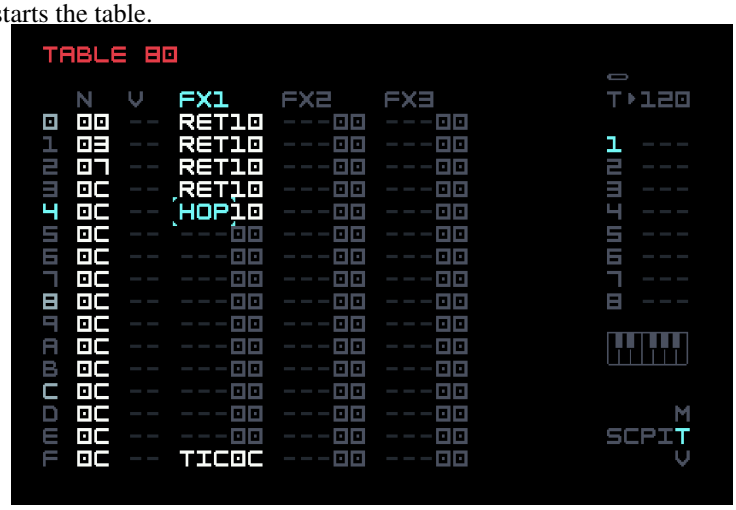
The V column is a multiplier for the velocity from the phrase note. 00 is completely off, not surprisingly, but 7F is 100%, and that is the default (which only explicitly appears when you try to edit this field). The help text at the bottom will tell you the percentage in decimal, so you don't have to do the math.

The playhead for the note and velocity columns moves at the same rate as the first FX column. The default tick rate for a table started with TBX is six, so the table will play at the same rate as the phrase. To slow things down a bit, both so it sounds better musically and so we can see what the playheads are doing, put a TIC0C in the last row of the FX1 column. That's a tick rate of twelve, so the table rows have duration equal to eighth notes.

If you add note values of 00, 03, 07, and 0C in rows 0 through 3 of the N column, that spells out a 4-note minor chord. Listening to the result, you will notice that two things are wrong. First of all, the volume envelope is only triggered with the first note in each bar, and the other notes are heard as the envelope decays. Second of all, the four notes only play once per bar; they don't repeat, as an arpeggiator would.

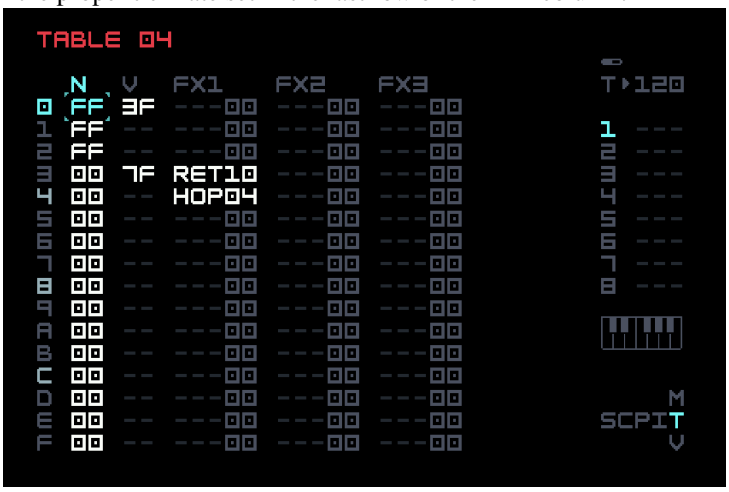
The first behaviour is actually desired in some situations, and it is called **legato**. You can have it happen in a phrase by using the PIT command in an FX column to alter the pitch of the previously-triggered but still-playing note, without retriggering the envelope. But in our case, we want to retrigger the envelope, and we can do that by adding RET10 in rows 0 through 3 of the FX1 column.

To have the arpeggio repeat, we can put a HOP00 command in row 4 of the FX1 column. Now the playhead for that column, on reaching row 4, immediately jumps back to row 0. That's why we needed the RET10 in row 0, not to play the note the first time, but for all the repeats. If we had more nonempty phrases or chains in our song, the arpeggio would play until another note or an OFF command is encountered on this track. You can see the playheads on the other empty FX columns make it through the first half of the table before the phrase repeats and its first note restarts the table.



HOP is one of those FX commands that acts differently in phrases and tables. In a phrase, the second digit of its value specifies the row in the next phrase to jump to, and the first digit is ignored. In a table, the second digit specifies the row in the table to jump to, and the first digit is either 0 (meaning it always happens), or a non-zero value (in which case it happens that many times, allowing short loops). You can see how playhead motion can get pretty complicated with clever programming.

Other ornaments and note modifications can be handled in a similar fashion. For example, grace notes. A grace note is a short related note played to delay the specified note slightly. When played on a drum, it is called a flam. We can use three ticks in a table moving at tick rate 01 to play a 32nd note. In the example below, the grace note is one semitone below and softer. Notice how we have to specify the transposition for all three ticks. We've been lazy and used an instrument table, but in practice this is also an occasional ornament, and should be put in an unassociated table with the proper tick rate set in the last row of the FX1 column.



Here we use HOP04 in row 4 to stall the playhead, because we don't want the effect to repeat, unlike with the arpeggiator.

These techniques can be iterated to create all sorts of repeatable note and rhythmic variations. By using the CHA (chance) command in a phrase to make the starting of an auxiliar table a probabilistic event, and then further using

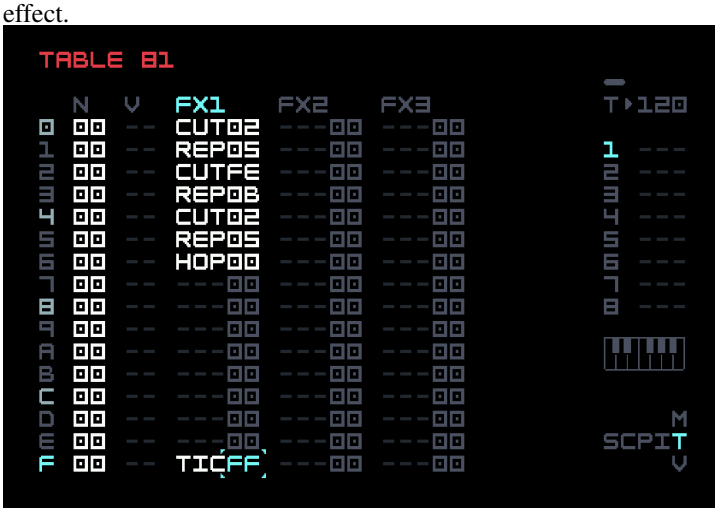
CHA in the table (different behaviour) to do the same for individual rows or FX commands, one can build generative songs with more structure.

7.2 LFOs and Envelopes

When one gets enthusiastic about sound design, it is easy to run out of modulators. Tables can be an additional source of modulation. They can't do everything that instrument modulators can do. Specifically, they are "steppy"; values only change at the column tick rate, and with the two-hex-digit range, so the graph of the modulation might resemble a Wavsynth waveform. Often this is not noticeable (for example, if the modulation is on volume) and tables can be used to build additional modulation shapes.

To perform the modulation, we use a relative command affecting a parameter. For example, CUT for the filter cutoff. CUT 10 will increase the filter cutoff by 10 (hex). Such a change will persist until the next change, or a retrigger. We'll combine this with the REP (repeat) command. In a table, REP repeats the previous row of the same column for the specified number of repeats (this is different behaviour from its use in phrases; there, a change value is specified, and the result is smoothed). CUT 10 followed by REP 02 will have the effect of raising the filter cutoff by 10, then twice again on subsequent column steps for a total increase of 30, which will be held until the next change. Instead of REP 02, we could also use RTO 20, because RTO in tables can be used directly after a relative command (no intervening REP needed, as in phrases) to specify an additional total increment.

Let's design an approximation to a triangle LFO with a period of 24 table steps. To get a total rise of 12 (decimal), we can CUT 02 and then REP 05, for a total of six steps. CUT FE (a drop of 2) followed by REP 0B (eleven more such drops) gives us a drop of 24 from the height of 12, reaching negative 12. Then another CUT 02 and REP 05 gets us back to zero, and we can repeat with a HOP00. Here's the table. We've set the tick rate to 200Hz so you can hear the effect.



If we increase the magnitude of the cutoff changes to 03 and FD, we get a range from +18 to -18. But we can't get a range between 12 and 18, unless we compromise on the shape of the LFO. If we change the last REP05 to REP04, the cutoff change will not get back to zero before the table repeats. That makes it no longer an oscillator, but a sort of jagged envelope, like a saw with a tapering blade.

REP works perfectly well when the row above is empty; it just holds at that table row for the specified number of steps. The same thing is true if the row above is an absolute command or another REP. To convert an LFO to an envelope, we only need to use the grace note technique of hopping to the same row to stop the table instead of letting it repeat. For example, we can build an AHD envelope using a positive CUT possibly followed by a REP (for the attack), then an empty row and REP (for the hold), and finally a negative CUT and possibly a REP (for the decay). Again, only certain shapes are possible, but also it is easier to make bipolar or inverted envelopes.

We cannot approximate ADSR envelopes so easily, because the sustain phase relies on the implementation of OFF in the M8 firmware initiating the release phase, and we don't have a way of being notified in a table of an OFF phrase event. What we can do is have a sustain phase where we specify both level and duration. We can also have the sustain

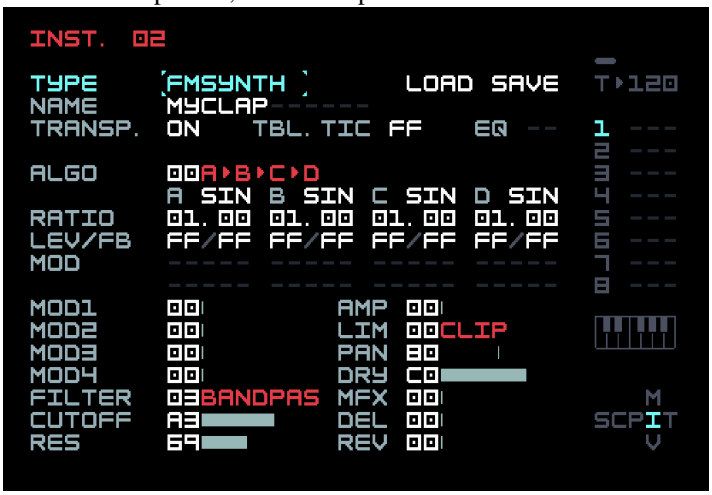
phase decay as it does for a piano, which isn't possible for an instrument modulator.

A certain amount of math (elementary, but still) is needed to make sure a complex LFO or envelope shape returns to zero, or to know where it is at when the table repeats. One of the M8 community members, *twwi*, has created a very nice table generator website that does the basic math and graphs the result. Handy as it is, you shouldn't use it without being able to understand the table it gives you, because if the result doesn't sound quite right, you should know how to fix it. If you are using a browser that supports WebUSB, you can even use this website to write the table back to your M8, rather than copying it labouriously.

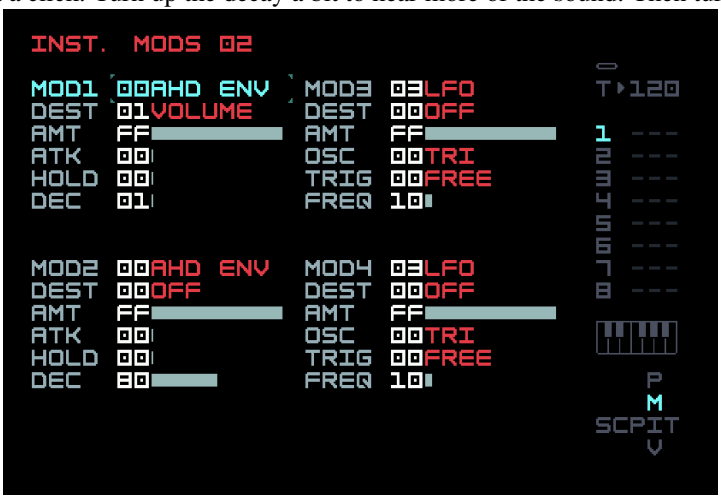
7.3 Sound design

As our use of tables gets more complicated and less methodical, we are moving into the area of general sound design. We're going to show just one example, a deconstructed version of the clap preset in the factory content (in the Drums instrument folder). Our version is slightly modified to make it easier to explain. Tim's version has artifacts in it from the experimental phase of his design process. For example, there are REP01 commands in it to shift a column's timing. These can be edited more quickly than moving part of a column up or down. After the explanation of our version, you can go back to the original, and it will make more sense.

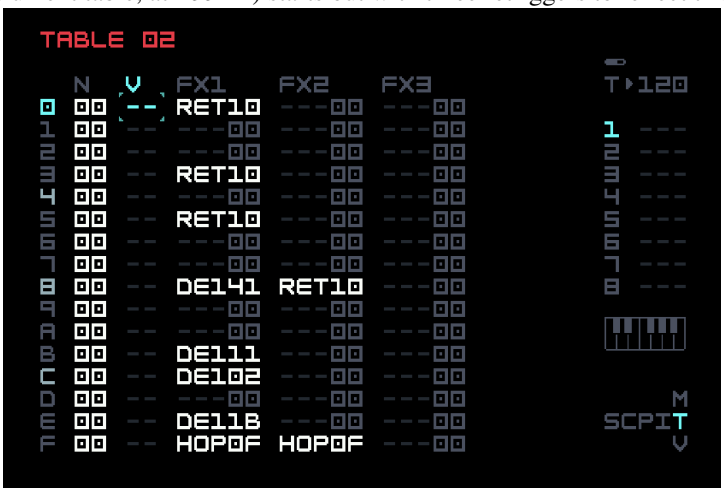
The clap is based on a fairly simple synth patch, using the default FM Synth configuration, but with all levels and feedback turned up to FF, to create a particular kind of noise.



This is tamed considerably with a single instrument modulator that is just a short envelope. The resulting sound is just a click. Turn up the decay a bit to hear more of the sound. Then turn it back down.



A transient plus noise is pretty much how we designed our FM snare in chapter 4, and if we play enough with the transient and the effects, we can probably get close to something that sounds like one person's clap. But what about several people trying to clap together? They aren't going to get the timing precisely right. So the table (running as an instrument table, at 200 Hz) starts out with three retriggers to reflect this slight inconsistency.



The table ends with a HOP0F, so we know it will only run once. What happens after the first three retriggers is a fourth retrigger, in row A of the FX2 column, but it is accompanied by a change in the decay envelope (the DE1 command that does this is absolute). This is done to introduce the diffuse noise that would result from the reverberations of the room, but with more control than just sending to the reverb effect.

The help text says that DE141 results in decay of 10 steps and 5 ticks, but this does not take into account the table speed. The default tick rate for an instrument table is 1, and the default number of ticks per phrase step is 6, so this means 65 ticks (41 in hexadecimal). But the table is running at 200 Hz, or one table step every 5 milliseconds, so the decay time of the envelope is being set to 65 times 5 or 325 milliseconds. The precise time is not as important as the sound it initiates, especially as the envelope decay is modified thrice more, in rows B, C and E. Tim probably arrived at these values through experimentation.

You should slow the table tick rate down to something from 01 to 06, and listen to the separate events. Then restore the 200 Hz tick rate, and alter the timing of the first three retriggers, and the values of the three DE1 commands, to see how they affect the sound of the clap. The table length of 16 is a constraint, but you can push that a little by using REP commands early on, as Tim does (you can put a REP right below a RET and it acts as if an empty row were above it; it does not repeat any retriggers).

This example is more of an inspiration, and a glimpse of possibilities, than a specific technique that one can generalize. The point is that a table offers much more scope for precise timing and alteration of sound parameters than a phrase does.

7.4 Alternate table maps

If the tick rate of a table is set to FE, it behaves very differently. The note name and octave are used as an index into the table, so a C-0 note sends the playhead to row 0, C#-0 to row 1, and so on up to D#-1 mapping to row B. Then E-1 will map to row 0, E#-1 to row 1, and so on. The playhead stays on that row for the duration of the note. Although this means only three FX commands can be used, one of them could start an auxiliary table. It is possible to reduce the range of the mapping in each column independently by putting HOP00 after the last desired row. For example, if HOP00 is in row B of a column, then D-1 would map to row A but D#-1 would map to row 0.

In the live video meetup introducing this feature and others in firmware 4.0, Tim said that he didn't have a particular use case in mind, that it was more of an experimental feature that users could run with. The most obvious use is with an external MIDI keyboard. One could map each note in an octave to a different percussion sound (the INS command could be used to change instruments), or map all the white keys to the notes of a particular scale to make improvisation easier, while the black keys trigger sound effects. In general, one can use this feature for live control using a MIDI

controller that can send notes.

But there are also internal uses. Since note value indexes slices in a sample, one could use the table to set parameters differently for sixteen slices (the seventeenth, if there is one, would have the same settings as the first, or you could wrap around more quickly). Adding a TICFE instrument table to the dub stab chord we built using FM Synth in chapter 4 would let us play any of sixteen chords with just the choice of phrase note, leaving the three FX columns in the phrase available for use. We could position each chord using the transposition column in the table.

Firmware 6.0 enables two more of these ”special tick rates”. A table at tick rate FD becomes a **velocity map**, that is, the table row chosen depends on the velocity of the associated note. This can be used in a manner similar to velocity tracking, but it is more general, since a full auxiliary table can be triggered for each velocity range. Similarly, a table at tick rate FC becomes an **octave map**. This could be used, for example, to create a **keyboard split**, where different regions of an attached MIDI keyboard trigger different instruments. One common use of this is to play bass with the left hand and lead with the right.

Note, velocity, and octave maps are only implemented for instrument tables, so put away your dreams of cascading them! Furthermore, they operate on the note and velocity as specified in those phrase columns (or in an incoming MIDI note) and not on any modifications due to FX commands.

7.5 Other stuff?

8 Interacting with other devices

You have probably used headphones or earbuds with your M8, plugging them into the 1/8” (3.5mm) audio in socket on the top left of the M8. The plug probably had two black rings on it, forming three metal regions that are connected to three wires inside the cable carrying left and right audio signals. This is called a TRS plug (tip, ring, sleeve). Some listening devices with microphones have a fourth region on the plug formed by a third black ring; these are TRRS plugs.

In the Sampler chapter, we saw how to use the Sampler instrument to sample from the audio in M8 socket, also on the top left of the M8. This also takes a 3.5mm TRS plug. If you sample from the headphone out of a computer or phone, you probably will use a stereo audio cable with 3.5mm TRS plugs on both ends (when we say stereo audio cable from now on, this will be the default). On the right side of the top of the M8 are two sockets of the same size, labelled MIDI in and out. If you have not used these yet, you might be surprised to learn that the same stereo audio cable can be used with the MIDI ports. But the cable does not carry analog (continuous) audio information in this case; rather, it carries digital (discrete) information using the MIDI standard, which can be thought of as instructions to a musical device (e.g. note pitch and velocity).

Between the two sets of 3.5mm sockets, on the top of the M8, is the USB socket (USB-C for M:02 and USB-micro for M:01). You have almost certainly used this to charge your M8. A USB cable can carry power, but some USB cables can also carry information, and this port can be used to send both audio and MIDI information between the M8 and other devices.

The first challenge, then, is to get the right kind of cables necessary to hook the M8 up to other devices that you want to use with it. But you also have to understand something about the nature of the interaction, and how to configure both the M8 and the other devices so that they can effectively communicate with each other.

In this chapter, we’ll go through a number of scenarios for such interactions, and try to alert you to potential issues.

8.1 Using an external display

One of the simplest interactions possible is to use the screen of a laptop or computer monitor as an external display for the M8. Besides offering an almost certainly larger view, this can be used for screen captures (still) or screen recording (video). That’s how we made all the examples of the M8 screens in this document.

For this, you will need a USB cable capable of carrying information, with one end suitable for plugging into the M8, and the other end into your computer. Hubs and adapters can be used. When you make this connection, nothing will happen on either side. Make sure your M8 is turned on.

You then need to use a Web browser to connect to the site m8.run (maintained by Tim). The catch is that not just any Web browser will do. You need one that supports the WebUSB standard. At the time of writing, this is Chrome, or a browser that uses the Chromium toolkit (such as Opera or Edge). The m8.run website will pop up a message if accessed by a browser without this support.

If you are using the right browser and open this site with M8 connected and powered up, it should display the screen. There is a three-line "hamburger" menu in the top left with various options, including a brief help popup. You can get M8's audio to play through your computer's chosen output device (usually speakers or headphones). To do this, make sure M8 is set as your computer's input source or microphone, and make sure that the browser page is authorized to use the microphone (click the little circular button on the left of the URL bar into which you typed "m8.run"). Finally, after doing all this, you might need to reload the browser page, and click in it. You might also want to turn off M8's own speaker (in the Mixer View page, [SHIFT]+[DOWN] from Song View).

If you prefer an app, there is one written in C, called m8c, and available on GitHub. There is no particular advantage to using this with the M8, except that it might be made to work on devices for which a Chromium browser is not available. This is the case for iOS, the operating system for Apple iPhones and iPads, on which all browsers are just "skins" on top of Safari, which does not support WebUSB. A port of m8c for iOS is available through the App Store.

8.2 USB audio options

USB has evolved over time, resulting in a number of different connectors, but also many digital protocols with different capabilities and speeds. As you are aware, it also provides power: 5V at 500mA commonly, but more recent protocols can negotiate higher voltages and power delivery. Devices that use USB fall into three broad categories: hosts (typically computers, tablets, and phones), peripherals (or clients), and hubs (that increase the number and type of connections).

It's important to understand that M8 falls into the peripheral or client category. It is not a host. This means it must be connected to a host (or a hub, in the right way) and cannot be connected directly to another peripheral. This is the case for both audio and MIDI (to be discussed below) Most musical devices that are not computers will not be hosts. If you are unsure about a particular device, look in its manual, or in the technical specifications on the manufacturer's website. If these mostly talk about connecting the device to a computer, it is probably not a host.

Although analog audio cables don't have a direction marked on them, they typically carry information in one direction, from a source of audio (like M8's audio out port) to a user or consumer of audio (like the audio in port on a powered speaker). In contrast, a single USB cable can carry information in both directions. We see this with MIDI information when using m8.run. Clearly information is flowing from M8 to the computer in order to animate the display. But the computer keyboard or mouse can be used to affect the M8 at the same time. We will see a more complex example of this in the section below on External Control Surfaces.

This section is primarily about USB audio. USB audio information also can travel in both directions between a host and client. The simpler direction, in terms of options, is USB audio into the M8. You will have to set up the host to make sure it is transmitting USB audio out. Like analog audio through M8's audio in port, USB audio can be recorded in the Sampler instrument (stereo, mono left or mono right, or mixed with the analog audio). USB audio can be managed in Mixer View, with a volume control into the mix, plus sends to the built-in modfx, delay, and reverb.

Until firmware 6.0, USB audio out from the M8 was a mirror of the analog audio out. While this is a pretty good setup much of the time, there are now more options. The options are selected by the "USB AUDIO OUT" line in the System Settings View. "MIX" is the previous behaviour. There is now a page of diagrams in the M8 manual for some of the new options, such as the ability to send the analog audio in straight to USB audio out.

Chief among these other options is the ability to take an individual track (only one, but "pre-fader", that is, before the mixer page) and send it on USB audio out. What this permits is the use of software on the host to process a given track and then sending it back via USB audio (or even analog audio) into the M8 to be mixed (with some work in Mixer View) with the rest of the tracks. This is known as an **insert effect** (since the returning audio can still be sent to the built-in effects, or used in an External Instrument, as described below).

Other options for USB AUDIO OUT allow for the USB audio out stereo channel to contain all the sends to one of the built-in effects, from all of the M8 tracks. Using the same idea as above, the returned audio can be configured to

replace or augment one of modfx, delay, or reverb. From a portability standpoint, this can be done with a single cable and a phone (loaded with suitable FX apps) that one probably already has along on the go.

Unfortunately, it is not as easy to use conventional guitar pedals (which mostly have 1/4" TS mono inputs and outputs) in this fashion with the M8, even if they have USB audio in/out, because they are unlikely to be USB audio hosts. They can still be used as end-of-chain effects, processing the full analog audio mix from M8 (which may only be one track, the others muted), but the result can only be returned for sampling and not for live mixing with other tracks.

Firmware 6.5 added one more line in Systems Setting View, which is "USB AUDIO MODE", with default "STEREO". This innocuous line heralds something that was long requested but thought too computationally intensive, even by Tim. But he figured it out. The other, new option is "MULTICHANNEL", which when selected has the M8 send a number of individual audio channels via USB: two for the main stereo mix, two for the stereo output of each M8 track, plus two each for the modfx, delay, and reverb sends, for a total of 24 channels. This lets an audio host capable of handling multichannel USB audio in process each of these tracks as wished, playing the result, recording it, or sending it back to M8 (there are still only two channels for stereo USB audio in on M8).

This might take more work to set up in the USB audio host than the scenario made possible by firmware 6.0 (which remains possible), but it clearly has much greater scope. For example, all of the three M8 effect sends can now be processed by FX apps on the USB audio host, mixed into a single stereo signal on that host, and the result returned to M8 to be added to the main stereo mix there or used in other ways that audio in can be used on M8. Another use would be to record stems in real time in a DAW (possibly with further real-time manipulations and effects there) instead of stems having to be rendered on M8 and transferred. As with other M8 features that might stress M8's CPU, keep the total computational load of your song in mind.

8.3 MIDI Settings View

The MIDI 1.0 standard was first published in 1985. It has been extended in various ways over the years, and MIDI 2.0 came out a few years ago, but relatively few devices support the newer standard so far. We're not going to try to be comprehensive in our discussion of MIDI, but rather to give some illustrations of the kinds of things you can do with MIDI on the M8. Your further explorations will involve research into your own specific devices, and of course direct experience.

The published standard starts out by specifying the cabling, using cables with 5-pin DIN connectors on either end. These had already been standardized for various electrical uses. The standard only uses two pins for MIDI out and three pins for MIDI in and thru. The plugs have a circular metal shield about 13mm in diameter and five pins grouped on one side of the circle. When manufacturers started making smaller and thinner musical devices, they hit upon the idea of using stereo audio cables, with their narrower plugs. Such cables only have three wires, so there are really only two choices of how to implement the standard, but the manufacturers, in their rush to market, did not try to standardize. As a result, there are two different implementations of MIDI using 3.5mm TRS plugs, Type A and Type B.

Most manufacturers use Type A; a few switched to Type A when it later became part of the MIDI standard. Others stick with Type B. Often a 3.5mm TRS to 5-pin DIN adapter is included with a device that has the narrower ports. However, these adapters are rarely labelled with their type! It doesn't hurt a device to use the wrong type, but it won't work. Furthermore, an adapter converting between these two types would, in audio terms, switch left and right stereo channels. Since this is not something people tend to want to do, it's not easy to find these adapters. As expected, this is a frequent source of MIDI issues, so be warned. The site minimidi.world has a lot of information on TRS MIDI, including a list of devices with their types.

MIDI messages come in several broad categories. The first one is **synchronization**. **Clock** messages are used to ensure that different devices stay "on the beat" and make sounds at the same time. Typically there should be at most one source of MIDI clock in a system, and some number of listeners. Clock messages are sent out at 24 to the quarter note, or 24 ppqn. This is exactly the tick rate of the M8, and this is not a coincidence. Tim designed the M8 this way so as to minimize issues with MIDI synchronization.

Another type of synchronization involves the built-in sequencers on devices. You usually want to start and stop these at the same time, but not always. Sometimes a device will have a sequencer, but you will choose not to use it, preferring instead to send note sequencing messages from another device. The start and stop instructions are called

transport messages. For each device, then, you have to answer the questions: do you want it to send clock? receive clock? respond to transport messages, or not?

On the M8, this and much more can be set up in MIDI Settings View, which is accessible from Project View. For synchronization, there are two lines in this view labelled SYNC IN and SYNC OUT. Each has a number of options, including OFF, CLOCK, TRANSPORT, and CLOCK+TRANSPORT. There are a couple of other options for which we refer you to the M8 manual. We'll discuss a few of the other lines in this view in this chapter.

MIDI has a concept of channels (1 through 16). Synchronization messages are not specific to a channel, but note messages are. A channel may be associated with a whole device (for example, if it is monophonic) or a track on a device. A note message has a pitch (standard Western chromatic scale) and a velocity, both encoded in the range 0-127. There is a note-off message, but the usual way to turn off a note is by sending a note message with the same pitch but velocity 0. There are also messages to send mod wheel and pitch bend information, since these were and are common controls on synth keyboards.

Program change (PC) messages can be used to select presets on the receiving device. These could be sounds (like instruments on the M8) or patterns (sequences). On some devices, these are linked (unlike on the M8). The precise way in which a receiving device responds to a PC message depends on the device.

There is even more variation in the interpretation of control change (CC) messages. These are used to select and change parameters on the receiving device. Some devices will have a fixed set of CC messages they respond to; with others (such as the M8) the mapping can be altered.

Finally, there is a category of system messages, of which the ones you are most likely to encounter are System Exclusive (SysEx) messages, which basically form an ad-hoc way of transmitting arbitrary information over a MIDI channel. For some devices, this is the way to do firmware updates; for others, SysEx might be used to backup or receive presets, or to perform various control functions specific to the device. We will encounter a use of SysEx in the section below on external control surfaces.

The most general and direct way to use an M8 track to produce MIDI messages is by using a MIDI Out instrument.

8.4 MIDI Out instrument

In this document, we've been able to assume that you either have an M8, or are seriously researching a future purchase. We can't make any assumptions about other devices you might have or are contemplating with which you might want to use M8's MIDI capabilities. This makes getting into specifics about interaction a bit tricky.

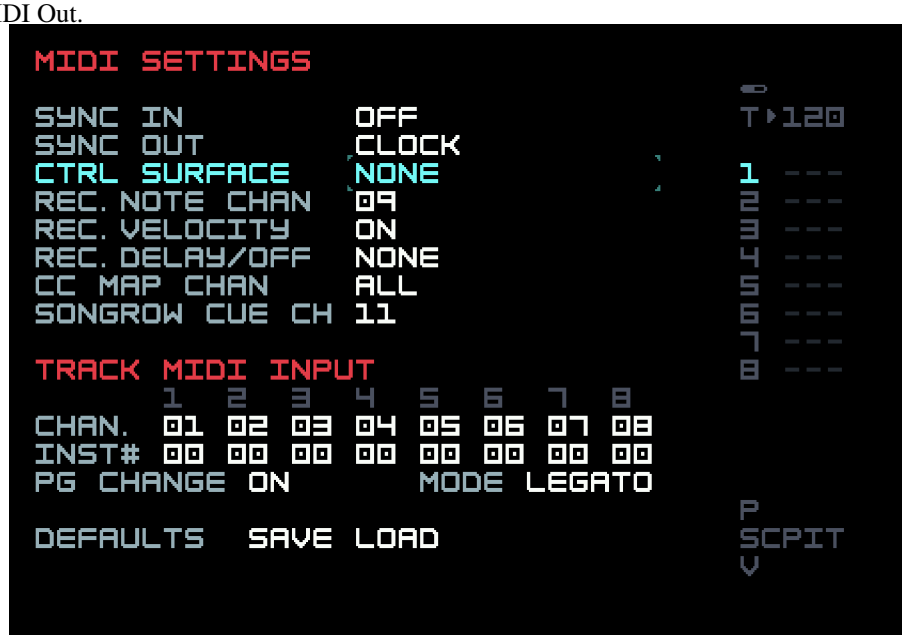
We've chosen a representative device to use in our examples, without getting too far into details which might be significantly different for your devices. Our choice is the Roland S-1 Tweak Synthesizer.



There are several reasons why this is a good choice for our purposes. First of all, it's a lovely device in its own right, the most capable of several that Roland has released in the AIRA Compact line. It recreates the sound of the legendary Roland SH-101, an analog monophonic synth that can be heard on many hits of the '90's. The S-1 is digital, but uses a technology developed by Roland that models the original analog circuits, and by all accounts does a pretty good job of that. It is much smaller and much cheaper (introduced at around $200), has a built-in battery that can be charged via USB, and has four-voice polyphony (four notes may overlap; they all use the same sound settings, but have independent envelopes).

The S-1 is often touted as a good companion for the M8, as it is similarly portable, and offers features that M8 doesn't: easy four-note polyphony, with lots of knobs to tweak the sound, and a two-octave button keyboard (pretty small buttons, though). It uses 3.5mm TRS audio ports and Type A MIDI ports, so standard stereo audio cables can be used for all connections with the M8. Finally, it is not only a good receiver for MIDI produced by M8, but a reasonable source of MIDI to affect M8, so we can use it in later sections to discuss that.

To have the M8 use MIDI to play the S-1, we need only run a stereo audio cable from M8's MIDI out port to S-1's MIDI in port. To set things up on the M8's side, we use the MIDI Settings page, and set up an instrument of type MIDI Out.



Here we have made sure that the M8 will send clock but not receive it, and that transport is off so the S-1's sequencer will not be started. None of the other settings are relevant.

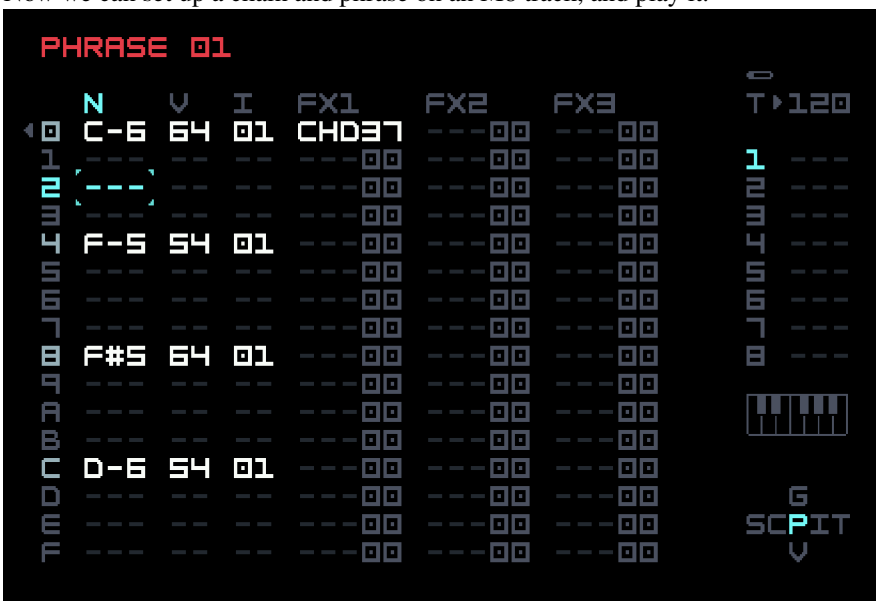
Both S-1 and M8 can handle MIDI over USB. However, both are USB peripherals, not hosts, so we can't connect them directly to each other with a USB cable.



In Instrument View, we set this instrument up so that MIDI is sent out on the MIDI Out port and not USB (just in case you want to charge M8 while doing this). We have to choose a MIDI channel to send on, and we choose channel 1. We don't need to touch any of the other settings.

On the S-1 side, we have to set it up to receive the required information. Since we aren't trying to sell you an S-1 (or discourage you from buying one), we'll spare you the details. Suffice it to say that running through menus with a single four-character seven-segment display that wouldn't have been out of place in a calculator from the 1970s, and a single encoder plus overloaded keyboard buttons to navigate and edit can be fairly frustrating. There is a menu item to choose the single transmit/receive channel that S-1 uses (so we need to make sure channel 1 is selected); there is a menu item for synchronization (with an "auto" setting, which will use transmitted clocks but otherwise default to the S-1's internal clock).

Now we can set up a chain and phrase on an M8 track, and play it.



Here we have four notes with four different pitches and some velocity variation. To demonstrate polyphony, we have added a use of the CHD (chord) FX command, which will add two notes to the specified root note. The two digits indicate an offset using the scale in effect. Since we are just using the default chromatic scale, the chord specified is

root, three semitones above, and seven semitones above, so a minor triad. We could have added just one note with the ADD command, and using all three FX lanes will let us build a seven-note chord (though S-1 will only play four of them). Tables will allow even more polyphony, if the receiver permits it.

CHD and ADD specify simultaneous notes. If we instead want overlapping notes, we could put in a note with nothing in the instrument column, which will use the previous instrument but not stop the previous note.

This does what we want. S-1 is silent until we press play on the M8, and then it plays the chord and three individual notes, which can be heard at S-1's MIX OUT jack with headphones. One can twist the knobs to alter the sound on S-1 while M8 is playing it. S-1 has an arpeggiator, and if this is turned on, the chord sounds as three sequential sixteenth notes (their order determined by the arp settings on S-1). If the note on the second beat were not present, the arpeggio would repeat until a new note is played on M8.

S-1 has 64 presets, containing both sound settings and patterns, but since we're not using the S-1's sequencer, a program change will just change the sound. We can use the MPG FX command to choose a new preset on the S-1.

S-1 has a fixed set of CC assignments. For example, CC 74 is filter cutoff. (This is actually a common use of CC 74. The MIDI standard defines a few CCs and makes suggestions for others, but many are left undefined.) If we wanted to control that from the M8, we could set up one of the ten CC slots (say slot A) in the MIDI Out instrument to send 74. We would then use the CCA FX command to send a new value when we wanted. We could also, in Instrument Modulation View, assign an envelope or LFO to CCA. But the control values are discrete (range 0-127), and in certain cases may sound "steppy", unlike the more smooth way that envelopes and LFOs are handled internally. This would be noticeable on pitch, and perhaps on filter cutoff under certain circumstances, but not on most parameters.

We could, instead of using headphones with S-1, send audio from S-1 back into M8 using a stereo audio cable. If we are going to do that, though, it is more convenient to use the External Instrument type, discussed in the next section.

At this point, we're going to remind you that audio coming into M8 (whether by an audio cable, USB cable, or the M:02 microphone) can also be sampled, that is, recorded into a new sample using a Sampler instrument. This can be as simple as starting recording on the M8, starting audio production, ending audio production, stopping recording on the M8, and then trimming out the dead space in the Sample Editor. But there are many more possibilities, including synchronizing recording, recording to a fixed size, starting recording automatically when audio reaches a chosen threshold level, having a metronome playing, and even a "count in" bar if you're playing something live. In chapter 5, we mentioned this useful video covering the subject, and you now know enough about external interaction to appreciate it fully.

Jay Maculate – How to Record Samples on the Dirtywave M8

8.5 External Instrument

External Instrument offers special handling of the common case where audio from another device controlled by MIDI from M8 is being fed back into M8 (via the analog inputs or USB audio in) to combine with other audio tracks there. The advantage offered by External Instrument is that, in contrast to the usual handling of audio into M8, here the audio goes through the filter and amp stages familiar to you from other M8 instruments like Macrosynth, as well as the M8's send effects (which normally external audio can be sent to).

The disadvantage of External Instrument, relative to just using MIDI Out and audio in separately, is that there are only four CC slots available instead of ten, because space is needed in Instrument View for the usual filter and amp controls. Also, the MIDI Out instrument permits default values for the CCs, to be sent on program change to initialize parameters that are not stored in presets on the receiving device; there isn't space for these default values in External Instrument.

Quick rendering is not available, because the MIDI instructions have to be played at tempo to make the receiving device respond properly. However, real-time rendering is available, using the same key combo (select and then double-tap [EDIT]). The dialog offers a third choice, where the first External Instrument found in the selection is recorded in real time. Furthermore, the audio mix can still be monitored, and any tweaks to the receiving device (for example, twisting knobs on the S-1 to alter the timbre) would of course be recorded. In System Settings, there is an option for a count-in metronome to facilitate real-time recording.

Between what we've written here and what is in the M8 manual, you should be able to figure out External Instrument for the S-1, or for the MIDI receiving device of your choice.

8.6 Using MIDI to affect M8 tracks

The S-1 also transmits MIDI (on the single channel chosen) so we can use it to discuss receiving MIDI on the M8. If we connect S-1's MIDI out port to M8's MIDI in port using a stereo audio cable, M8 will receive the MIDI messages from S-1. But how do we tell M8 what to do with them?

The most obvious source of MIDI on S-1 is the button keyboard, and of course using it results in MIDI note messages. This miniature keyboard is not velocity-sensitive like fancier keyboards; the velocity sent is chosen by a menu item on S-1. We can have these affect M8 by using the Track MIDI Input table near the bottom of MIDI Settings View. This has an entry for each track on M8, allowing one to select a MIDI channel for the track and an instrument. That instrument will be used to play a MIDI note message on that track. This will happen whether or not M8's sequencer is running. If it is running, the sounds from MIDI note messages will be interspersed with the notes programmed on that track, which of course can use any instrument. The default setting is track 1 uses channel 1, and so on.

If program change is turned on in MIDI Settings View, a program change message on a MIDI channel can be used to change the instrument associated with tracks using that channel. A given MIDI channel may be used by several tracks. The MODE setting here offers varying interpretations of this setup. Choosing MONO means that each track plays its instrument at the same time. LEGATO is like this, but will not retrigger notes (only change pitch) if the notes overlap. POLY instructs M8 to use different tracks to play overlapping notes, thus achieving a polyphonic sound. Of course, there have to be enough tracks available, otherwise some notes will be cut off.

S-1 only transmits on a single MIDI channel, but some keyboard controllers will allow splits so different regions of the keyboard transmit on different channels, and a multitrack music device with MIDI out capability or sequencing software on a computer, table or phone will be capable of transmitting on multiple channels simultaneously. These can be used to turn M8 into an eight-voice music source with instrument selection, driven not by its internal sequencer but externally via MIDI.

A more pragmatic use of a MIDI keyboard with M8 is to facilitate note entry into phrases. To this end, MIDI Settings View allows one to choose a MIDI channel to be the Rec Note Channel. The default is channel 9 (the first one not used by defaults for tracks). This can be changed, or we can change our keyboard to use this channel. If M8 is in Phrase View with the cursor in the note column, and a MIDI note message is received on the rec note channel, then the note will be entered at the cursor.

If you have some facility with keys, you might want to play notes into M8 in real time. You can do this by tapping [PLAY] in Phrase View. The playhead will start moving, any notes already inserted will play as expected, and note messages will result in note insertions. Velocity information is also recorded (and can be varied with a velocity-sensitive keyboard). If you don't play precisely on the beat, DEL commands will be used to approximate your timing. But remember that the tick resolution is six ticks per quarter note, so some quantization will take place. Sometimes this will erase a bit of sloppiness on your part, and sometimes it will magnify it.

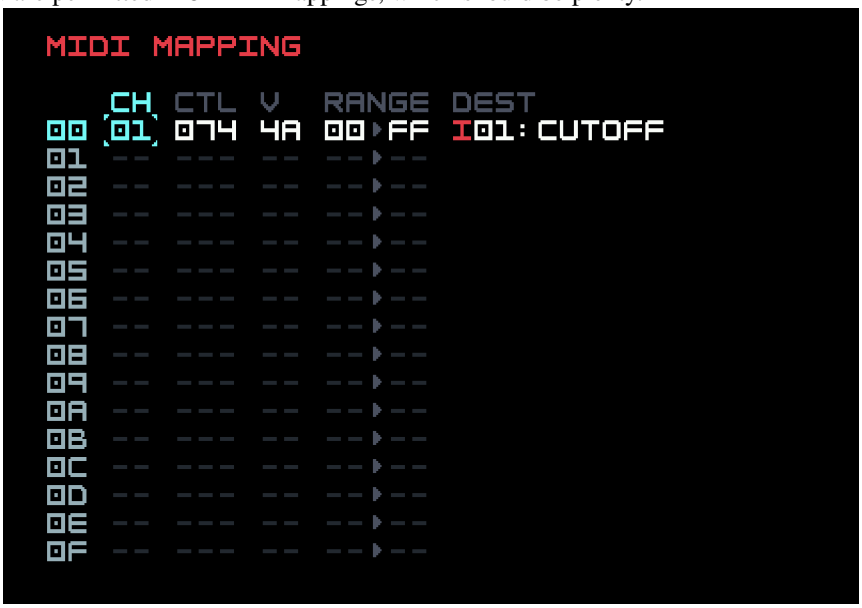
If you wish to record into several phrases, set them up as empty, and then use [SHIFT]+[PLAY] in Phrase View to play the whole song. The playhead will continue past the phrase being viewed, and recording will happen in another phrase; you just won't be able to see it. Unfortunately, metronome count-in is not available for this. If you have a spare track, you can set this up yourself.

You are likely to decide that a take isn't good enough, and if there are too many edits for you to salvage it, you might want to start over. There isn't an undo operation here, but a workaround is to drop into Mixer View ([SHIFT]+[DOWN]), and tap [SHIFT]+[OPTION] to save a snapshot of the empty phrases. If you later want to erase a whole take, you can return to Mixer View and tap [SHIFT]+[EDIT] to restore the empty snapshot.

There is a list, at the end of the S-1's manual, of which MIDI CCs are recognized or transmitted. Most devices with fixed MIDI CC implementations will have such a list somewhere. M8 does not have one, because its response to CCs is configurable. This is known as **MIDI learn**. We saw that CC 74 would control filter cutoff on the S-1. There is also a knob for this, and if one turns the physical knob on S-1, it will transmit a CC 74 message with the knob value. We could use this value to control a filter on M8, but we could also use it to control something entirely different.

But let's stick with a filter for now. We can set up instrument 01 on M8 to be FMSynth. In Instrument View, we can move the cursor down to filter cutoff. To have S-1's knob affect this on M8, we only need hold [OPTION], and turn the knob on S-1 to send a CC 74. The words ASSIGNED MIDIMAP appear at the bottom of M8's screen to let us know that learning has taken place.

The list of MIDI mappings set up so far can be seen in MIDI Mappings View, also accessible from Project View. You are permitted 128 MIDI mappings, which should be plenty.



The V column contains the last received value. If you turn the filter cutoff knob on S-1, you will see this value change. The change is also visible in the filter cutoff display in Instrument View.

MIDI learn can be a fraught or messy process on many devices; it's nice to have it be so straightforward on M8. But it still can take some time, if one wants to use many knobs and buttons. This has to be done at least once, but one can save a project template to reuse the mappings for a specific device in future.

Some editing is possible in MIDI Mappings View. You can edit the channel and CC number. You can specify a range to which the value associated with the CC (normally in the range 0-127) will be mapped, for more subtle effects on the chosen M8 parameter. A parameter can only be affected by one CC mapping, but one CC can be mapped to several parameters. If we do this with CC 74, the filter cutoff knob on S-1 becomes a **macro** knob, which can have complex effects on M8. This is quite useful both in the studio and for live performance.

Sooner or later, it occurs to one to try to run a cable from the MIDI out port of a device back to the MIDI in port on the same device. This is known as **MIDI loopback**, and the point is to try to gain more control, for example, to implement macro knobs on a device that does not directly support them. For some devices, such as S-1, this is pointless, or results in unacceptable feedback loops. But for others, such as Elektron boxes, it is a useful, if somewhat awkward technique.

We don't have to do this on M8, because Tim has made the rare but commendable decision to allow internal MIDI routing. That is, a track on M8 can receive MIDI from a MIDI Out instrument on another track, just by setting this up in the relevant menus. M8 has no knobs, so we can't get macro knobs this way, but we can immediately adapt the polyphony idea above to have one MIDI Out instrument play several other M8 tracks to create polyphony that is more general than what we have previously seen. Another use would be to send an envelope or LFO (albeit steppy) to a parameter that is not normally available as a destination of an instrument modulator.

All we have to do to use internal MIDI is, in Instrument View for a MIDI Out instrument, set the Port to Internal. Previously we set this to MIDI, avoiding USB, and both MIDI+USB is also possible. Internal is the fourth setting. Then any MIDI message sent by this instrument is available (only) internally on the chosen channel, which can affect other tracks by an appropriate setting in the Track MIDI Input table in MIDI Settings View.

It's not immediately obvious how to use MIDI learn to have a track affected by MIDI CC sent internally, since

there is no knob to turn on the internal track, unlike with an external MIDI controller. Here's where the touchscreen comes in handy. Either the vertical or horizontal touchscreen axis can be assigned to control any parameter, in the same way as MIDI learn (hold [OPTION] and stroke in the right direction). Then that mapping can be edited in MIDI Mappings View to instead use CC from an internal MIDI track.

There are many MIDI keyboard controllers available, as well as controllers with various numbers of configurable knobs, buttons, faders, and pads. Just remember that if you want direct connectivity, you need one with a physical MIDI output, either DIN or TRS (and maybe you need a type converter). If you already have a computer, tablet, or phone involved in your system (or you have one of the rare devices that acts as a USB MIDI host), then you may be able to use that to route MIDI to/from the M8 over USB.

8.7 External control surfaces

The third line in MIDI Settings View (accessible from Project View) allows one to specify a control surface. At present, there is only one choice supported, the Novation Launchpad Pro mk3 (LPP). If any other possibilities arise, we will document them here. Although there are several Launchpad models, for the time being, it has to be exactly this model to use this mode (even mk2 will not work). Tim added some support in firmware 5.0 for use in his own live performances, and expanded it in 6.0 in response to user enthusiasm. This mode does not need configuration, unlike the various uses detailed above, though this necessarily means it cannot be customized, either. The best video demo of the use of an LPP, at the time of writing, is Tim's overview of the 6.0 firmware, available on Dirtywave's YouTube channel. Most of what follows in this section is taken from that video, with additional clarification from Tim and others on the Dirtywave Discord.



(image from Novation.com)

The LPP is a popular MIDI controller, with an 8x8 grid of velocity-sensitive pads (with multicolour LEDs beneath) surrounded by control buttons (with fixed labels, but also some lighting options). On release, it was primarily intended for use with the digital audio workstation Ableton Live, and this is reflected in the permanent labelling of the control buttons. But many other software products provide support for controlling them via LPP, and LPP is fully programmable via MIDI SysEx messages. This, combined with its 1/8" TRS MIDI in/out matching M8's, is what makes it so suitable for additional control of M8 (and explains why many M8 owners already owned one for other reasons). Although the control modes are aimed at live performance, they can also be useful in song development, as we'll see. The M8 continues to function exactly as it would if LPP were not attached, that is, all the button combos and views on M8 work the same way. Sometimes you will need to look at M8's screen for information, as LPP does not have a screen.

The first thing one must do with LPP in order to use it with M8 is to update LPP's firmware to the latest version using Novation's Components software. Do this even if you bought it new in a sealed box, otherwise it may not respond as you expect. LPP requires USB power; M8 cannot provide this. You will need two 1/8" TRS cables (plugs on both ends). One goes from M8 MIDI out to LPP MIDI in, and the other from LPP MIDI out 1 to M8 MIDI in. Once that is done, you can turn on M8, and select LPP mode if you have not done so already. M8 will send the necessary programming commands, and the pad lighting on the LPP will change.



This photo (taken in dimmer light so the pad colours are visible) is showing Song View for the demo2 project. Relevant control buttons are lit. There are three relevant modes on the LPP: Session, Note, and Sequencer (plus a fun hidden mode we'll describe later). You can see that the Session control button in the top row is lit up in blue to indicate that that mode is active (this view is just after turning M8 on), but Note and Sequencer are lit white to show that they can be selected. Session mode was the only one available in the 5.0 firmware; the rest were added in 6.0.

8.7.1 Session mode and global actions

In Session mode, the pads mirror the Song View grid, so each column of pads represents one track. But you can scroll the LPP view independently of the view on M8, using the up/down triangle buttons at the top of the left control

column. Since there are no labels on the LPP pads, colours are used to provide some partial information. Unused phrases are unlit; empty phrases are dull pink; nonempty phrases are white. If you hit [PLAY] on M8, the top row of pads on LPP will turn green, indicating the playhead position for each track. But you can also hit Play on LPP (right-pointing triangle, second from bottom on the left control column) and the same thing happens.

Certain actions on LPP are global, that is, they work in all three modes. These include the Mute and Solo buttons on the bottom control row. When Mute is selected, it will light red (white when unselected), and this permits muting/unmuting of each of the eight tracks, using the track selection buttons at the bottom of each pad column (with horizontal lines on them, white for playing, red for muted). When Solo is selected, a track selection button will solo the corresponding track (tap the track button again to bring in the other tracks). For both Mute and Solo, pressing them again will cause them to flash, and in this mode, pressing and holding a track selection button will have the desired effect, but only as long as you hold it (allowing more rapid interactions).

Another global action allows saving of a snapshot, by holding Shift (top left) and pressing Record Arm (leftmost on bottom control row). The Record Arm button will stay lit to indicate a saved snapshot, and you can revert to that at any time by pressing Record Arm again. You can see how this would be useful in live performance, to improvise and then return to standard play.

In Session mode, the pads on the right control column, lit and marked with the $>$ character, can be used to start playing on a given row. You can switch into M8's Live Mode by holding Shift and tapping the left-triangle button just to its right. The Novation logo at top right will light up to tell you on LPP that you are in Live Mode on M8; repeating the action goes back to Song Mode on M8. While in Live Mode, individual chains can be cued on LPP using the pads.

The Projects button (rightmost on top control row) will take you into the Load Projects View on M8. You can either use the M8's buttons (since you have to look at its screen anyway), or use the LPP navigation pads at top left. Tapping Shift on LPP selects the new project under the cursor. Note that there is no "do you want to save the old project?" dialog as there would be in M8 navigation; LPP mode assumes you have messed up the old project with live improvisation and don't want to save your spontaneous changes; you really just want to start the next song in your performance.

8.7.2 A slight digression for Note mode

There is an editing submode of Session mode, but we'll first describe another, simpler mode, namely Note mode. In Note mode, each pad of the LPP plays a distinct note. Lower rows play lower notes, and upper rows play higher notes. It isn't nearly as easy to use as a proper keyboard, but it does have the advantage of respecting the global or track scale, so every pad plays a note in scale. The root notes in each octave are lit white.

But which instrument is playing? The track selection buttons still work in this mode, but as you know, M8 has no notion of a default instrument on a track. The instrument that is playing is the last one selected on the chosen track. Holding Shift and pressing one of the Up/Down buttons below it will pull up the Instrument Pool View on M8, after which the buttons can be used to move around and select another instrument. If you want to add to the pool, the Projects button will bring up the Load Instrument dialog on M8, which can be navigated like the Load Project dialog above (arrow keys to move, Shift to select, tap Projects again to exit without selecting). These two instrument-focussed interactions also work in Sequencer mode, as described below.

8.7.3 More on Session mode

Back to Session mode. The bottom button on the left control column (just below the play button) has a circle on it, often used as the symbol for recording. It also says "Capture MIDI" below that circle. We'll call this the Record button. If you tap this while in Session mode, it enables editing on the LPP, and the button will light red to remind you of this. Tapping it again leaves the editing submode.

While Record is lit, the cursor position is indicated on LPP by a blue pad, and chain references identical to the one under the cursor will be pulsing (so you know where your edits will have an effect). By holding Record, you can change the note at the cursor in the current phrase using the keypads,

You will notice the Clear and Duplicate buttons on the left control column are also lit. To clear a chain, hold Clear and tap the pad you want. To duplicate a chain, hold Duplicate and tap the pad to copy (that pad will be lit red), then tap once elsewhere to clone into that spot or double-tap to deep-clone. Holding Duplicate and using the $>$ buttons on

the right control column will copy and paste entire rows. If, in Note mode, you hold Record and press Play, you can live-record into the currently-playing phrase using the keypads. But this is probably better done in Sequencer mode. Double-tapping a pad will go into Sequencer mode focussed on that chain, or you can just tap the Sequencer button to do so for the chain under the cursor.

8.7.4 Sequencer mode

Sequencer mode is the most complex. The 8x8 grid of pads is divided into three regions. The bottom four rows work as keypads, as in Note mode. The top right 4×4 grid represents the 16 phrase slots in the displayed chain (identical phrase references will pulse), and the top right 4×4 grid represents the 16 note slots in the displayed phrase. The up and down buttons scroll the keypads to reach higher or lower notes. The left and right buttons move within the same row to select different tracks. To loop play of a single phrase, hold the phrase pad and press the play button.

Tapping the Record button to engage edit submode opens up more possibilities. In this mode, one can hold a note pad in the top left 4×4 grid to edit it. The $>$ buttons on the right let one change velocity in a coarse fashion (only 8 values possible), and the up and down buttons allow for quick octave shifts of the given note. Holding Record and tapping in the keypad section changes the note under the cursor. Remember that the pads are velocity-sensitive, so both pitch and velocity are recorded.

When Shift is held, the left/right buttons will rotate notes within the phrase. When a phrase pad is held, the up/down buttons will change the transposition of the phrase (as specified in Chain View on M8). The uses of the Clear and Duplicate buttons as detailed above for chains in Session mode also works for phrases in Sequencer mode (hold Duplicate, single-tap to copy, single-tap to paste, double-tap to clone). And remember that the instrument commands described above in Session mode and the live-record command mentioned for Note mode also work here!

8.7.5 Just a bit more...

Finally, the fun hidden mode is Beat Repeat mode. You enter it by holding Shift and pressing Session (which will turn red to remind you). In this mode, the bottom row turns blue, and the two rows above it represent the 16 slots of a phrase. You can hold just one pad to select a single slot, or two pads to select the range between and including them. In either case, while the playhead proceeds normally through phrases and chains, only the selected slots will loop on playback. That is, if slots 2-4 are selected, then for each phrase, the notes 2-3-4-2-3-4 will repeat for 16 steps, at the same rate as if the mode had not been engaged. You can, of course, change the selection on the fly.

The blue row represents tracks, and tapping a pad can exclude or include tracks from being affected. Between this, and being able to easily change the selection, there is a considerable range of possibilities. For any given song, it will take some experimentation to find the sweet spots, but the result will be nice and fairly easy improvisational variation.

That concludes our description of the LPP modes for M8 control. You may be concerned with the fact that the TRS MIDI jacks on M8 are entirely used up when it is connected to LPP. LPP has a second MIDI out port, which can be configured to mirror its MIDI in. The communications between M8 and LPP use MIDI channels 1-3, so if you want to use M8 to control other devices by MIDI, you can use this second out port on LPP, making sure that M8 is sending to those other devices on MIDI channels 4 and higher. Getting MIDI into M8 from other devices while LPP is connected is a bit trickier. You will have to use something with MIDI merge capabilities on the path from LPP MIDI out to M8 MIDI in, to merge in the MIDI information from those other devices.

9 To be completed

9.1 Other chapters?

Perhaps an ”Intermediate techniques” chapter, but what might it contain?

Techniques for polyphony

9.2 Things to cover somewhere

What NTH actually means and how to use it.

SNC (for hypersynth), incl >80 (inverts half the oscs, gives pulse)
RND, RNL (briefly in ch 4), SED (random seed).
NXT, ERR

9.3 Undocumented FX Commands (add somewhere)

Instruments:
FMP in FM Synth DONE (also any non-obvious commands in other synths)
Sequencer: ARC RTO GGR (all 4.0) INS (omitted?)
Instrument Mods: LS4 (omitted?)
Mixer/Effects: VIN (IVO renamed?) IRE (mistakenly IRV in 3.0 man) VI2 (renamed IV2?) DJC DJR DJT

10 Sources and acknowledgements

Thanks to Tim Lamb for creating such a great device, for being open and transparent about its design and use, and for consistently engaging with its user community. Thanks to that community, on the Dirtywave Discord and a smaller group on Elektronauts, for being friendly and helpful, with many quick responses to questions, useful tips, short videos of techniques, and longer video or audio of songs. We'd especially like to thank those who gave us feedback on early drafts of this document.

We learned a lot from longer tutorials or walkthroughs by Red Means Recording (Jeremy Blake), EZBOT (Matthew Piecora), and mikey303 (Mike ?). The Discord meetups and overviews of major firmware updates (videos available on Dirtywave's YouTube channel) were a rich source of information.

Dave Mech's courses for the Elektron Digitone I and Digitakt I provided many practical examples and tips for sound design and composition that could be translated to the M8 context, as well as being excellent examples of how to structure a learning experience in electronic music.