

# Drumkid Eurorack manual

## Introduction

Drumkid is a drum machine module which uses random numbers to create unpredictable rhythms. Essentially, you start with a standard drum beat and then use Drumkid's knobs to adjust how much randomness to add.

You start with a pre-programmed rhythm using four samples, then manipulate this rhythm using the knobs and control voltage (CV) inputs to add hits to (or remove hits from) the original rhythm.

This manual consists of a tutorial to guide you through the basic workflow of the module, followed by a detailed description of each parameter or feature. I have also included some information on how to hack/modify the firmware code or even the hardware.

For help with any aspect of Drumkid, you can join the Bradshaw Instruments discord server here: <https://bradshawinstruments.com/discord> (<https://bradshawinstruments.com/discord>)

## Tutorial

The easiest way to start with Drumkid is to play a simple beat without any random additions, then gradually add random elements to get a feel for how the module's algorithm works.

1. Connect a patch lead to "out 1" so you can hear the output
2. Set the knobs as follows:
  - Chance 50%
  - Zoom 0%
  - Vel 100%
  - Vel range 0%
  - Cluster 0%
  - Crop 0%
  - Crush 0%
  - Pitch 50%
  - Magnet 0%
  - Swing 50%
  - Drop rand 50%
  - Drop all 50%
3. Press "Play"
4. Select a preset beat by pressing "beat", then using the +/- buttons (beat #2 is a good starting point)

5. Slowly decrease the "chance" knob to zero and notice how the beat is gradually silenced as the probability of each hit happening is reduced
6. Slowly increase the "chance" knob back to 50% and beyond. As you increase past 50%, notice how more and more random hits are added to the beat, until at 100% you hear every sample being triggered over and over (100% probability).
7. Set the "chance" back to around 2 o'clock (75%), so you can hear that the full original beat plus some random hits, and now try adjusting the "zoom" knob. As you increase the zoom, the beat will become busier as hits are triggered at smaller note values (8th-notes, 16th-notes, 32nd-notes, etc). These first two knobs, chance and zoom, are the heart of the machine, and I would recommend playing around with them for a while to get a feel of how they affect the beat.
8. Slowly reduce the "vel" (velocity) knob from 100% to 0% - the extra random hits will become quieter.
9. Set "vel" to 50%, and "vel range" to 100%. The hits should now have a full range of volumes, from quiet to loud. Try playing with these knobs to adjust the relative volume of the random hits.
10. With both "chance" and "zoom" at around 2 o'clock (75%), try slowly adjusting the "magnet" knob to see how it reduces the chaotic nature of the randomness.
11. Try out the remaining knobs to see what effect they have. If you get stuck, set the knobs back to the values shown in step 2.
12. Try connecting CV signals, one at a time, to the four CV inputs, to get a feel for what can be done with these.

## Parameters (knobs)

### Chance

Determines the chance/probability of a hit occurring on a given step. At 0%, no hits will be generated. At 50%, all hits from the programmed rhythm will be heard (but nothing else). At 100%, every possible hit will be generated.

### Zoom

Determines which steps of the rhythm are eligible to have extra random hits. At 0%, hits will only be generated on quarter notes. As you increase the zoom knob, you introduce the possibility of hits on 8th notes, 16th notes, 32nd notes, etc, all the way until 128th notes. Note that these values will be different if the tuplet mode is anything other than "straight" (e.g. triplet).

### Cluster

Similar to the "chance" parameter, but designed to create repeated hits or rolls. If a random hit was generated on the most recent eligible step, "cluster" is the probability of another hit being generated on this step (unless "chance" is higher). For instance, if chance is set quite low, only occasionally generating hits, and then cluster is set to 90%, you will get occasional but quite long series of repeated hits. If you set cluster to 100%, you will keep hearing repeated hits continually.

## **Crop**

This crops the sample to create a staccato effect. At 0%, the entire sample is heard. As you increase the crop knob, the sample will become shorter and shorter until you only hear a tiny pop at 100%.

## **Crush**

This reduces the bit depth from 16 bits (CD quality) at 0% down to one bit at 100%, creating a harsh, crunchy, distorted sound.

## **Pitch**

This changes the playback speed of the samples. At 50% (12 o'clock), the playback speed is 100% (normal), and there is a central "dead zone" at this value to make it easier to lock the pitch here. Above 50%, the playback speed increases, resulting in a higher pitch. Below 50%, the playback speed decreases, resulting in a lower pitch. As you keep reducing the pitch knob towards 0%, the sample playback is actually reversed, and then increases in pitch again, so that 0% results in fast, reversed sample playback.

## **Velocity (vel)**

This determines the velocity (volume) of the randomly generated hits. It is used in conjunction with "vel range", so if "vel range" is 0% and "vel" is "75%", all randomly generated hits will be at exactly 75%. However, if "vel range" is 50% and "vel" is 50%, the randomly generated hits will have random velocities in the range 25% to 75%.

## **Velocity (vel) range**

The range of velocities for randomly generated hits (see above).

## **Magnet**

This knob affects whether hits are generated at completely random points within the bar, or whether they are more likely to occur on larger-value subdivisions, e.g. quarter notes. At 0%, no adjustment is made to the way random notes are generated, but as you increase the magnet knob, the likelihood of a note being generated becomes proportional to the note's

subdivision, so a quarter-note hit is more likely than an eighth-note hit, which in turn is more likely than a 16th-note hit, etc. In practice, this knob creates order out of chaos, turning a messy set of random hits into something more rhythmic. Note that as "chance" approaches 100%, the efficacy of the "magnet" effect is gradually reduced to zero - it was designed this way so that the behaviours of "chance" and "magnet" were not fighting each other at high probabilities. This is perhaps the most subtle control on Drumkid - I would recommend keeping it at 0% in normal use, and increasing it when you want to create a groove from something that is otherwise too messy.

## **Swing**

Allows you to swing certain hits. At 50% (12 o'clock), hits are generated as normal, with a dead zone to make it easier to keep this setting locked. As you turn the knob to the left (towards 0%), any off-beat 8th-notes are increasingly swung, while turning the knob to the right (towards 100%) swings any off-beat 16th-notes. When either of these swing modes is in use, random hits of a lower subdivision than the swung hits (e.g. 32nd notes when in 16th-note mode) are muted. The swing knob is not available when tuplet modes (e.g. triplet) are activated in the settings. If you are not getting the desired behaviour while adjusting the various knobs which generate random hits, check that swing is set to 50% (off).

## **Drop all**

This knob drops (or mutes) certain channels. At 50% (12 o'clock), all hits are audible. As you turn the knob to the left, you first mute the snare, then also the hi-hat, then the tom, and finally the kick. Turning the knob to the right first mutes the kick, then the tom, then the snare, then finally the hi-hat. At 0% and 100%, all random hits are muted. This knob essentially allows you to create "breakdown" sections in a performance by progressively (or suddenly) eliminating certain channels.

## **Drop rand**

Similar to the "drop all" knob (see above), but only applies to randomly generated hits, not hits from the programmed beat.

## **General usage/workflow**

All current settings (e.g. tempo, time signature, tuplet mode, etc) are saved automatically.

## **Inputs and outputs**

### **Clock in**

Receives a clock input (a series of short pulses) to allow the module to be synchronised to an external signal. More details in the "synchronisation" section.

## **Clock out**

Outputs a clock signal (a series of pulses). The number of pulses sent per quarter note (PPQN) can be set in the menu, as can the length of the pulses.

## **Trigger outputs (kick/snare/hat/tom out)**

Each of the four samples has its own trigger output. This outputs a short pulse every time a given sample is played by the module. The pulse length can be adjusted in the settings menu. Note that if you only want to use the trigger output and not generate any audio for a given sample channel, you can disable individual sample audio outputs in the settings menu.

## **CV inputs (chance/vel/pitch/zoom in)**

The four CV inputs all work in the same way: the CV signal is added to the relevant potentiometer value. A positive CV signal will increase the value of a given parameter, while a negative signal will reduce it. For instance, if you have the velocity knob set at 50% and a sine wave LFO signal (with a range of -5V to 5V) connected to the "vel in" CV input, the velocity parameter will swing from 0% to 100%. If the CV signal would cause the parameter to go beyond the 0% to 100% range, it is clamped, except for "pitch in" because it's fun to have some extra range and there's no real reason not to. CV signals beyond the range -5V to 5V are clamped.

## **Out 1**

Audio output at 44.1kHz. You can use the menu to adjust which channels are muted for each output, and whether the crush effect is active for each output.

## **Out 2**

Second audio output, see above.

## **Display and indicators lights**

### **Main display**

The four-digit display is normally used to show which quarter-note of the rhythm is currently being played. Pressing buttons will cause the display to show other relevant information. You can always return to the main display by pressing "back".

### **Error**

The "error" light is lit when there has been some sort of issue, usually related to loading samples from the SD card. The main display will give an error code when something goes wrong (see error code section). The error light will remain lit until the error is no longer actively causing an issue.

## Pulse

This light blinks once per quarter note when a rhythm is playing.

## Clock in

This blinks every time a clock pulse is received.

## Clock out

This blinks every time a clock pulse is sent.

## Buttons

### Play

This button seems self explanatory but has different behaviours depending on whether you are using internal or external clock. When using internal clock, pressing "play" once starts the rhythm playing from the start of the bar, and pressing it again stops the beat. When using external clock, the beat will always continue to play whenever a clock signal is received, but pressing "play" will reset the rhythm to the start of the bar.

### Tap

Tapping this button repeatedly will set a new tempo (each tap is a quarter note). The first tap will reset the rhythm to the start of the bar, and subsequent taps will fine-tune the tempo by averaging the past few taps. This button only works in internal clock mode.

### Live (edit)

This button puts the module into "live edit" mode, where you can play a beat using the four buttons below the main display (+/-/✓/×), corresponding to kick/snare/hi-hat/tom by default. In this mode, all random hits are muted, and a metronome sound is played. When you trigger a sample using one of the four buttons, it is added to the current rhythm. When you add a hit, **the velocity of that hit is determined by the current value of the "vel" knob**. If you make a mistake, you can clear the whole current beat by pressing "clear", or clear a specific sample channel by holding "clear" and pressing one of the four buttons below the main display. When you are ready to save your beat, press "save", use the +/- buttons to choose a slot (the existing beat in that slot will be overwritten), then confirm by pressing "✓". Beats are

not saved automatically. To revert to how the beat was before you started editing, press the "beat" button, go to a different beat and then return to the beat you were editing (i.e. just press "+" then "-").

## Step (edit)

This button puts the module into "step edit" mode. Press the button repeatedly to cycle through the four sample channels. Once you have selected the desired channel (e.g. snare), use the +/- buttons to select a step within the rhythm, then press "✓" to add a hit or "x" to remove one. **The velocity of the hit is determined by the current value of the "vel" knob.** The number of steps available is dependent on both the time signature and the tuplet mode - for example, in "straight" mode with a 4/4 time signature, there will be 32 steps, but in "triple" mode with a 3/4 time signature there will only be 18 steps. If you make a mistake, you can clear the whole current beat by pressing "clear", or clear a specific sample channel by holding "clear" and pressing one of the four buttons below the main display. When you are ready to save your beat, press "save", use the +/- buttons to choose a slot (the existing beat in that slot will be overwritten), then confirm by pressing "✓". Beats are not saved automatically. To revert to how the beat was before you started editing, press the "beat" button, go to a different beat and then return to the beat you were editing (i.e. just press "+" then "-").

## A/B

These buttons currently not operational. They are reserved for future firmware updates or user modification (see "hacking" section).

## Save

When in live edit mode, this button is used to save an edited beat (see the live/step edit sections of the manual for details).

## Beat

When you press this button, the current beat number will be shown on the display. Use the +/- buttons to select a new beat.

## Tempo

To manually adjust the tempo (in internal clock mode), press the tempo button then use the +/- buttons to change the tempo by whole BPM values, or the ✓/x buttons to make fine changes (0.1BPM). In external clock mode, you can press the tempo button to view the current calculated tempo.

## Tuplet

This button allows you to select one of four different tuplet modes using the +/- buttons: straight, triplet, quintuplet, or septuplet. The tuplet mode determines which steps of the pattern are eligible for random hits, as well as which steps of the pattern are available in live edit or step edit mode.

## Kit

When an SD card is inserted, this button allows you to select the current set of four samples (a "kit"), using the +/- buttons. Press the "✓" button to confirm your choice and load the samples, which might take a couple of seconds, during which audio output will be interrupted. Once the kit has been loaded, it will stay in the module's memory permanently until you choose another kit, meaning you don't need to leave the SD card inserted. See below for example file/folder layout - there should be a folder called "samples" on the root of the SD card, containing a number of subfolders, each with the name of the "kit" (set of four samples). Each subfolder/kit should contain four WAV files with the filenames "1.wav", "2.wav", "3.wav", and "4.wav". The files must be in WAV format. The files can be stereo, but will be converted to mono when loaded. The sample rate can be 44100Hz, 22050Hz, or 11025Hz (you can use other sample rates, but they will be played back at 44100Hz). The maximum combined audio data size for each kit is 420kB (after any conversion to mono). An error light and message will be displayed if your total file size is too large. If you want to use longer samples (for cymbals etc), you can try reducing the sample rate of these files.

Example file layout on the SD card:

- (SD card root)
  - samples
    - 808
      - 1.wav
      - 2.wav
      - 3.wav
      - 4.wav
    - 909
      - 1.wav
      - 2.wav
      - 3.wav
      - 4.wav
    - jazz
      - 1.wav
      - 2.wav

- 3.wav
- 4.wav
  
- rock
  - 1.wav
  - 2.wav
  - 3.wav
  - 4.wav

## **T.sig (time signature)**

This button allows you to select a time signature using the +/- buttons.

## **Clear**

In live edit or step edit mode, this button clears/deletes hits from the pattern. See the live edit or step edit sections of the manual for details.

## **Back**

This button exits you from whatever state the display is currently in and returns you to the main display.

## **Menu**

Pressing the "menu" button will enter a mode where you can adjust various settings. You can scroll through the different settings using the +/- buttons, then choose a setting to adjust using the "✓" button. Once you have selected a particular setting, you can then adjust/confirm it using the +/-/✓/x buttons, or go back using the "back" button. The following settings can be adjusted from the menu:

- Clock (internal or external)
- Out 1 (activate or deactivate individual channels on output 1)
- Out 2 (activate or deactivate individual channels on output 1)
- Crush (choose which outputs use the crush effect)
- Pulse length (length of sync/trigger output pulses in milliseconds)
- PPQN out (how many pulses per quarter note are transmitted by the clock output)
- PPQN in (how many pulses per quarter note are expected by the clock input)
- Factory reset (resets everything to factory settings - all data will be lost!!)

## **Synchronisation (clock)**

The module can either use its own internal clock or it can be synchronised to an external clock signal. Here's an overview of each mode:

## Internal clock

In this mode, you just press "play" to start the beat, which will then play at the current specified tempo until you stop the beat by pressing "play" again. You can set the tempo manually with the "tempo" button or by pressing the "tap" button repeatedly. Since the module can work with very small subdivisions (128th-notes), the ideal situation is to use the internal clock, which will be more precise, although this will depend on your preferred workflow.

## External clock

In this mode, the next step of the beat will play every time a pulse is received. If you use an input value of 1 PPQN (pulses per quarter note, set in the settings menu), then every pulse received will play the next quarter note in the rhythm. If the rhythm contains hits that need to be triggered in between these pulses (e.g. 8th-notes), the module will attempt to calculate when to do this automatically. In external clock mode, you can reset the rhythm to the beginning of the bar by pressing "play".

## Error codes

The error codes below correspond to those that will be shown on the display if something goes wrong, and are included to help identify any potential problems:

- er 1: the SD card is missing
- er 2: can't open a file on the SD card
- er 3: can't close a file on the SD card
- er 4: can't mount the SD card
- er 5: the total sample size is too large

## Troubleshooting

- I can't hear anything, or certain samples are missing
  - Make sure that "drop all" (and possibly "drop rand") are in the desired positions (50% or 12 o'clock will mean nothing is muted)
  - Make sure that all desired channels are enabled in the "out 1" or "out 2" section of the settings menu
- I'm trying to add hits in either live or step edit mode, but I can't hear them
  - Check that the "vel" knob is not set to zero (this determines the velocity of the hit)
  - Make sure you haven't accidentally exited live edit mode (you will hear a metronome if you are in live edit mode)
- The timing sounds weird when I'm trying to edit a beat

- Make sure you're in the correct tuple mode for what you're trying to do, otherwise the quantisation will be wrong
- There is a weird click at the end of a sample, or it cuts off too soon
  - Check the "crop" is set all the way to zero

## Updating firmware

If there is a future firmware update (to introduce new features or fix bugs), this can be loaded onto the module by following these steps:

1. Turn off the power to your Eurorack system
2. Unscrew the module so you can access the circuit board
3. On the rear circuit board, find a device labelled "Pico 2"
4. Connect one end of a USB micro cable to the Pico (but don't plug it into your computer yet)
5. While holding down the "BOOTSEL" button on the Pico, plug the other end of the USB cable into your computer, then release the BOOTSEL button
6. The Pico should now appear as a mass storage device on your computer (if not, unplug the cable and try the previous step again)
7. Drag and drop the new firmware (a ".uf2" file) onto the mass storage device
8. Once the file is successfully transferred, remove the USB cable from the Pico and screw it back into your case
9. Turn on your Eurorack system and check that the module is working as intended

## Hacking

Drumkid is a hackable project - it is designed so that anyone with the right skills can modify it, or use the design as the basis for their own projects. The C++ code, Kicad schematic/PCB files, and other related material, are all available on Github at [github.com/mattybrad/drumkid2](https://github.com/mattybrad/drumkid2) (<https://github.com/mattybrad/drumkid2>). What follows is an overview of this module's design:

- Based around a Raspberry Pi Pico 2 running C++ code
- Audio generated by a PCM5102 DAC
- 5V power for the Pico comes from an switching regulator
- 3.3V power for the DAC comes from a separate regulator (cleaner power than using the Pico's 3.3V source)
- TL072 op-amps used to amplify audio outputs to Eurorack level
- TL072 op-amps also used to amplify trigger outputs to Eurorack level (>5V)
- MCP6002 op-amps are used to bring the CV inputs to the correct voltage range for the Pico to read
- SD card socket running in SPI mode
- SIPO shift registers (daisy-chained) control the 4-digit 7-segment display and LEDs

- PISO shift registers (daisy-chained) read the buttons
- Multiplexers read the potentiometers and CV inputs
- Two separate, stacked PCBs to make the module more compact
  - The bottom PCB contains the Pico, DAC, and power regulation
  - The top PCB contains the user-facing components (buttons, pots, sockets, etc)
  - Where possible, I've used SMD components to save me from having to do too much manual soldering, but a lot of the through-hole components are still manually soldered
  - A third "PCB" is used as the module's faceplate, although it has no electrical connections - it just makes sense to get the faceplate manufactured in the same place as the PCBs (usually JLCPCB, for reference)

The easiest way to start modifying this module is by editing the C++ code and uploading your own custom firmware. I won't describe in detail how to write/build in C++ for the Pico, since there are many guides available on the web. For reference, I use Visual Studio Code on Windows, with an official Raspberry Pi debug probe for uploading code and debugging. If you don't require debugging capabilities, you could instead make your code edits, build a new .uf2 file and upload it via USB as described in the "updating firmware" section of this manual.

If you want to try hacking/modifying the hardware, take a look at the Kicad project relating to your PCB version (printed on the PCB).

If you have any questions about modifying the firmware or hardware, there is a Discord server here: <https://bradshawinstruments.com/discord> (<https://bradshawinstruments.com/discord>)