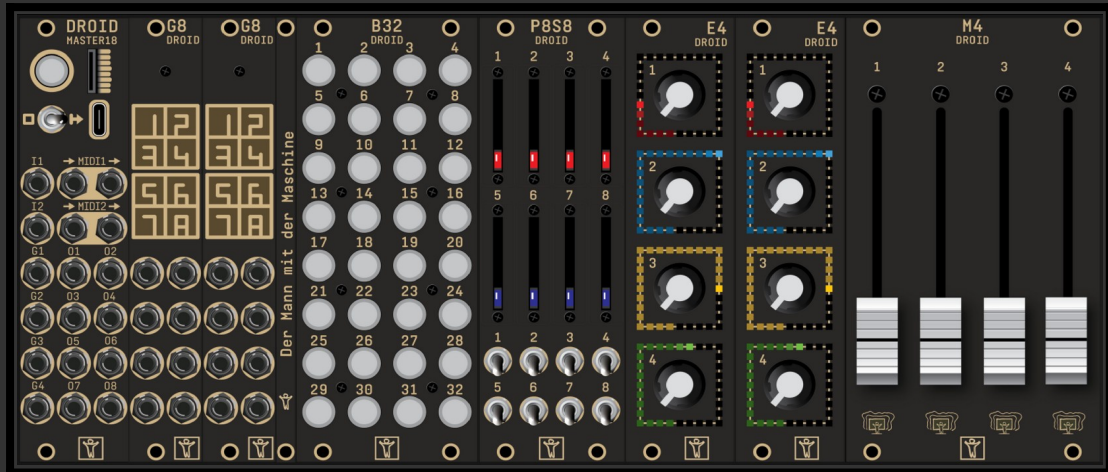


DROID blue-5

Release notes

MASTER18, Patch generators, Motor fader sequencer



These are the release notes for the DROID blue-5 firmware for your DROID master. Here you will learn about all the new features and also about a lot of bug fixes.

Contents

Hints for updating.....	2
General improvemets.....	3
New circuits.....	4
New features.....	5
Bug fixes.....	6
DROID Forge 1.5.....	7

HINTS FOR UPDATING

In the firmware ZIP file there is a subdirectory called **manual** with the current user manual for Droid. Please carefully read the chapter about how to update your firmware. In addition here are some hints:

- If you are still on the old **green-8** firmware, you must first update to **blue-1**, then to **blue-2**.
- If you are one **blue-1**, you must first upgrade to **blue-2**.
- You may skip blue-3 and update directly from blue-2 to blue-4, if you like. Please read the release notes of blue-3 in this case. There are some things you might need to adapt in your patches.
- **If you start the firmware upgrade and all LEDs of the master flash magenta, your firmware file is not accepted. In this case:**
 - **Put the firmware file ("droid.fw") from your current firmware onto the SD card (that from blue-2)**
 - **Do an update of the bootloader as described in the user manual.**
 - **After that upgrade to blue-4.**
- There is **no** new firmware for the X7 in this release. It is the same as in blue-3.
- There is **no** new firmware for the M4 in this release. It is the same as in blue-2.
- There is **no** new firmware for the E4 in this release. It is the same as in blue-4.

GENERAL IMPROVEMETS



- Support for the new MASTER18. This new master module is an interesting alternative for the classic MASTER. The MASTER18 has the same internal architecture (same CPU, same amount of RAM, etc). In order save HP and money, it does neither have CV inputs nor LEDs. But it has MIDI and USB support right built in so you don't need an X7. It even has two TRS MIDI ports with in and out each.

The MASTER18 is ideal for situations where you need a good MIDI integration and where you don't need CV inputs (maybe your DROID creates all the CV itself, anyway).

Here are the most important facts:

- Just 6 HP (instead of 8 for a MASTER or 12 for MASTER + X7)
- No CV inputs and no LEDs (sorry guys, there was no space)
- Two gate / trigger inputs I1 and I2 and four gate outputs G1 ... G4
- Eight high precision CV outputs (same as on the the MASTER)
- Two TRS MIDI inputs and outputs (with DIN adapters)
- USB port for MIDI and fast SD card access.
- You can still use the X7 with the MASTER18 and get a total of 10 MIDI ports!
- Builtin VCO tuner
- Sinfonion Link: Attach the MASTER18 to your Sinfonion as a listener and follow the current root / scale (degree), transposition, clock, reset and more information.
- **Cheaper price, especially if you compare with the MASTER + X7 combination!**

For the MASTER18 there is a special firmware version. So blue-5 contains one firmware for the MASTER and another for the MASTER18.

NEW CIRCUITS

- **[detune]**: New circuit for disharmonic detuning of up to eight voices
- **[case]**: New circuit for selecting one out of up to 16 inputs based on 16 related conditions.
- **[multicompare]**: New circuit for making decision based upon eight different reference values
- **[sinfonionlink]**: New circuit for syncing with the Sinfonion (MASTER18 only)
- **[vcotuner]**: New circuit for measuring a frequency and tuning a VCO (MASTER18 only)
- **[outputcalibrator]**: New circuit for calibrating the CV outputs on the MASTER18 (rarely needed).
- **[unusedfaders]**: New circuit for efficiently disabling unused motor faders.

NEW FEATURES

- **[buttongroup]**: Allow “startbutton = 0”, for unsetting all buttons in a situation where minactive = 0.
- **[arpeggio]**, **[chords]**, **[motoquencer]**, **[encoquencer]**: new inputs “noteshift” and “selectnoteshift”. These do the same as in **[minifonion]**: A diatonic transposition within the current scale.
- **[arpeggio]**, **[chords]**, **[motoquencer]**, **[encoquencer]**, **[minifonion]**:
 - New input “harmonicshift”, which reduces the harmonic complexity by disabling certain scale notes.
 - The “degree” input now supports all **108 scales** of the Sinfonion (not just the first 12). The “degree” input does not wrap around anymore at 12. You find a table of the scales in the user manual.
- **[motorfader]**: New output “button” for easy access to the touch button while the motorfader is selected.
- **[motoquencer]**, **[encoquencer]**:
 - New input “invert” for musical inversion of melody
 - New input “buttoncolor” for user a defined color of the gate buttons.
 - New input “taptempo” for supplying a steady clock for computing correct gate lengths in situations where the clock input is be stopped and restarted.
 - New input “repeatshift”. This introduces a diatonic transposition (linke “noteshift”) for every repetition of a step (if repeats > 1). This is like a miniature arpeggiator.
 - New input “ratchetshift”. Similar here: diatonic transposition for each sub note of a step when you use ratcheting. Another kind of mini arp.
 - New input “accumulatorrange”. This enables a pitch accumulator, which can alter the pitch of a note by -2, -1, +1 or +2 steps within the selected scale for each repetition of the sequence.
- **[arpeggio]**: New input “autoreset” for limiting the cycle length.
- Ship a simpler example patch for new masters (which is in the firmware package in the directory patches and is called start.ini)
- **[midithrough]**: Support broadcasting to all output ports or to all ports of a certain type (TRS or USB).
- **[quantizer]**: New option “direction” for choosing between rounding down, to the nearest value or up. The new default is now “nearest” (it was down). Suggested by @SLOfizz.
- **[encoquencer]**:
 - New option “nume4s”, which is used when zorder = 1 or zorder = 3. This allows you to use an arbitrary NxM rectangle of you giant E4 encoder matrix for the encoquencer. Indirectly suggested by @Indignant.AI.
 - Make the enabled gates better visible. Now instead of one single LED at the bottom of the LED ring, three LEDs are used for the gate. The gap (unused LEDs) left and right of that was increased to two LEDs. So now the gate uses the complete lower bar of the ring – except the left and right corner.

BUG FIXES

- **[midiin]**: Fix detection of "active" when active sensing is used. Now "active" correctly goes to 0 at latest 350 ms after unplugging the cable. Previously it stuck to 1 forever.
- **[pot]**:
 - The LED gauge is not longer shown after a change in select, now only when the virtual value has changed. This avoids confusion when lots of buttons, pots, faders and encoders are selected at the same time.
 - Fix newly introduced problem where it takes about 40 seconds after a restart until the pots would react to any movement (when they are overloaded with select). Found by [@eising](#) and [@effiksmusic](#).
- **[encoder]**, **[encoderbank]**: fix tiny oscillation around the point defined by "snapt0". Now the value eventually lands exactly on the "snapt0" value.
- **[encoquencer]**, **[motoquencer]**:
 - Limit the maximum value of "cvrange" to 1. Strange things would happen, if this value was higher. Also limit "cvbase" to the range -1 ... 1.
 - If **all** steps of your sequence are set to "skip", the last step was repeated over – even if skipped. Now the sequencer is silence when all steps are skipped.
- **[motoquencer]**:
 - Fix weird sometimes sticking fader when the note range is too large.
 - Fixed the description and example of the pitch accumulator in the manual.
- **[encoquencer]**: If the number of allowed notes would be > 63 (for example due to a large value of cvrange), the encoder would not allow to move beyond the first couple of notes. This was due to an internal rounding error and has been fixed.
- **[switch]**: Fixed bug appearing when the number of used outputs is not a power of 2 and the offset is negative (found by [@Jean](#)).
- **[arpeggio]**: Fix first note played after an external reset when that reset is not (within 5 ms) at the same time as the next clock signal. In that case the arpeggio always started with the second note. Also fixed some examples in the manual. When you set root = 2 it does **not** set the first played note to D unless you also set the pitch range to start at D. Both issued found be [@kIngvrhltnss](#).
- **[encoquencer]**, **[motoquencer]**:
 - Fix: The gatelength input was sampled just at a clock tick and ignored during the duration of the step. For that reason a modulation of the gate length just one Droid loop cycle after the clock was not effective for the current step. So when you added a linked sequencer for modulating the gatelength input, there was on offset of one step. The gate length setting of step 3 affected step 4 etc. This has been fixed.
 - In compose mode now the linked sequencer outputs are updated, as well. So when you have a linked sequencer modulating the velocity of the gate length, these parameters are used when changing a step in compose mode.

DROID FORGE 1.5

New features

- Support for the new MASTER18
- Patch Generators (see below)
- New option *View / Show RAM used by circuits*. If turned on, in the title bar of each circuit the RAM usage of this circuit is displayed as a number of bytes. This helps optimizing your patch when you are over 100% RAM usage and are looking for circuits to remove or replace by more RAM efficient ones.
- New RAM analysis function in menu *File* called “Analyse patch RAM usage”. It creates a list of all types of used circuits with their individual counts and RAM usage. This helps you to identify those circuits that use up the most memory.
- The Forge now double checks if patches have been saved correctly to disk. After saving a patch the contents of the saved file is checked and you get a warning if it is incomplete, missing or otherwise garbled. This check is also done for the automatic backup files and when saving a patch to the SD card.
- The Forge now checks for situations where you use an output register (such as L1.1 or O7) only as in *input*. Remember: using outputs as input can make sense if at some other place in your patch it is used as an output as well. Then you access that output value by reading from the same register. But if nobody writes into L1.1 or O7, reading from there always results in the value 0. And that’s a strong indication that you have garbled something up in your patch. The Forge now helps you to find such issues.

So if you get some new patch problems after updating the Forge to 1.5, check if you really write into all those output registers that you read from.

Bug Fixes

- Fix selection of values from a table via the “Table” button in the edit dialog of a cell. For example for “color” in “encoderbank”, every number was listed twice, all names were missing except “magenta” and numbers like “0.8” would be converted into “0.8000000943949”. All of this has been fixed.
- Fix automatic increment of trailing numbers in patch cables names in outputs when doing “expand parameter array”. They were increment by 3 instead of by 1.
- Avoid duplicate warning for losing unsaved changes when opening a new file while the current patch is modified.
- Automatically convert typed spaces into underscores when typing patch cable names directly in a cell (spaces are not allowed here).
- Fix weird behaviour when pasting multiple columns, which were copied from an input parameter, to an output parameter. Sometimes the second column would get copied instead of the first one. This has been fixed.
- Fix crash when “Save section as patch” was called from the right-mouse-menu on the section title. This crash happened mainly on Windows but rarely also on Mac. It did not happen when you called this option from the main menu. Found by @Raymond.

Patch generators

The most important new feature of the Droid Forge is the introduction of the **Patch Generators**.

Building complex patches for Droid can be quite challenging, especially if you are just at the beginning of your journey. So what people have suggested since the beginning was a good collection of ready-to-use patches. While this idea sounds appealing, it's actually not as ideal as it seems. Everybody has as a different set of modules and controllers, and also different needs. And creating a patch variant for every possible situation would vastly multiply the number of needed patches.

As an example take a patch that creates a performance sequencer based on M4 motor fader controllers. There are so many possible configurations:

- How many tracks should be provided?
- Should the output be via MIDI or CV/Gate?
- What parameters per step should there be?
- Shall we use two or four M4 controllers?
- Should it rely on a G8 expander for gate output?
- What type of master module should it use?
- ... and so on.

It's obvious that creating one dedicated patch for each variant like one for "Sequencer with three tracks for two M4s, MIDI output, no G8, using MASTER18, extra CV for velocity" would not be feasible.

So we had to find a better solution. And here it is: the **Patch Generators**. A patch generator is a program or script that creates a ready-to-run patch based on choices you make in a dialog in the Forge. For example in this dialog you would select the number of tracks, the configuration of modules to use and so on. Then you press OK and get a patch that you can upload to your master. Since it's a normal patch, you can edit it before this. For example you could rearrange the buttons by dragging and dropping them with the mouse.

As a start the Forge comes with one important first patch generator: that for a performance sequencer with motor faders. Look for the new submenu "Patch Generator" in the File menu of the Forge. Enable the generators and then in this menu you get a list of all available patch types to create. In the manual there is a chapter about the Patch Generators and the patches they create.