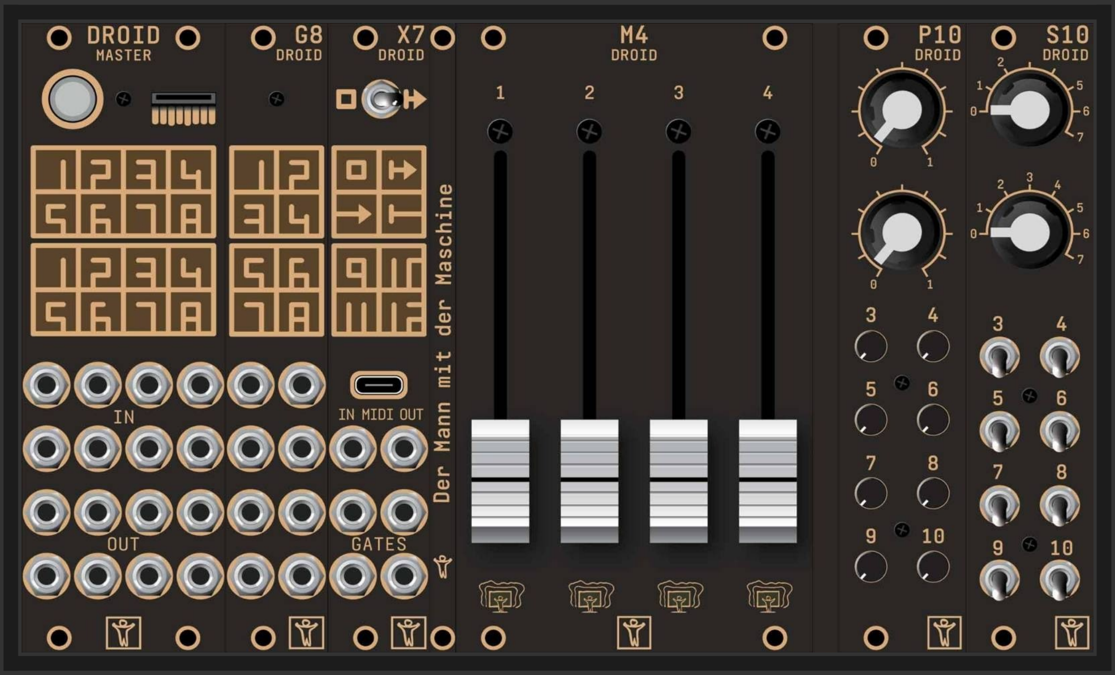
DROID blue-3

Release notes



These are the release notes of the DROID blue-3 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

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:

- The chronological order of the firmwares for the DROID master is **green-8 -> blue-1 -> blue-2 -> blue-3.** Do not skip any of these versions.
- 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-3
- There is also a new firmware for the X7. If you have an X7, install the firmware as described in the user manual.
- There is **no** new firmware for the M4. It is the same as in blue-2.

GENERAL IMPROVEMENTS

- Your patches now need substantially less RAM. Your mileage may vary, but as a rough estimate the memory needed is just 60% as opposed to blue-2. This is due to an internal restructuration so that inputs and outputs of circuits now need just 2 bytes each if they are not patched.
- Your DROID master now supports up to four G8 expanders. The new version 2 of the G8 has an output header for chaining to the next G8. So you can create a chain of up to four G8s. The last G8 can be a version 1 without the extra output header. So if you want a second G8 on your master, get a version 2 G8 and put it between the master and your existing version 1 G8. You address the jacks with a dot notation, e.g. G3.7 for the seventh gate on the third G8. The user manual and the Forge have been updated.
- Support for the upcoming P8S8 controller. As soon as the P8S8 is available, it will work with blue-3.

NEW CIRCUITS

- [`delay`]: New CV delay for delaying continous CVs and also gate patterns and integer numbers.
- [`recorder`]: New circuit for recording and playback of CVs, gate patterns und the movement of integer numbers.
- [`gatetool`]: New circuit for operating on gates, triggers and edges and changing gate lengths.
- [`once`]: New simple circuit for sending exactly *one* trigger after the start of the Droid master.
- [`select`]: new simple circuit for just copying if select is high.
- [`flipflop`]: Simple and fast 1-bit flip flop (as suggested by @nandoll)

NEW FEATURES

- All circuits that allow user interaction now support a consistent way of saving their state to the SD card and of loading a default state at the start or via a trigger. These are [`algoquencer`], [`button`], [`buttongroup`], [`calibrator`], [`faderbank`], [`fadermatrix`], [`matrixmixer`], [`motoquencer`], [`motorfader`], [`notebuttons`], [`nudge`] and [`pot`]. All these circuits now support the following new inputs:
    - "clear": resets the circuit into its default state. Some circuit have means for specifying what that default state should be. This is often done with "startvalue" or a similar input. For example [`notebuttons`] has now an input called "startnote" that sets the note to select on a "clear" trigger or a Droid start without a loaded state. **Note: on some circuits existing inputs have been renamed to match this new scheme. See below for details.**
    - "clearall": the same, but also all presets are overwritten with that default state
    - "dontsave": If you set this to 1, the saving and loading of the circuits state (and presets) to and from the SD card is disabled. When your Droid starts, the circuit start in its default state.
- [`droid`]: New trigger inputs "clear" and "clearall" for globally triggering these inputs at those circuits that support them.
- The X1 register now also supports integer numbers in the range 2 … 1000 and -1000 … -2. See the updated manual for details.
- When you have multiple patches on the SD card, like droid11.ini, droid12.ini and so on, each patch file now has its own separate state file. That means that switching between the patches now also correctly switches between separate saved states. So e.g. the current button of a buttongroup in droid11.ini will be remembered and restored when you switch to that patch.
- The master's button is now lit while it is reloading the patch.
- M4: All circuits that use the motor faders can now have more than 25 notches. You can set the number of notches to up to 201 (allowing to select values in the range 0 ⋯ 200). When the number of notches is more than 25, however, the force feedback is turned off (as the notches would be too small).
- M4: The setting “m4notchpower = 0” in the [`droid`] circuit now turns off the force feedback completely.
- [`lfo`]: "taptempo" can now be combined with "hz".
- [`motoquencer`]:
    1. New input "luckyscope" to rule which range of the sequence is affected by the "Lucky" operations. The questions here is how start/end and the currently selected page are taken into account.
    2. New trigger input "clearskips" to remove the "skip" state from all steps
    3. New trigger input "clearrepeats" to set the repeats to 1 for each step
    4. New input "defaultcv" for setting a CV to be set when "clear" is triggered or the motoquencer starts without any saved state.
- [`button`]:
    1. The output "longpress" is not longer a trigger output but reflects the current state of the momentary button. That means that now "longpress" gets 1 after you hold the button 1.5 seconds and *stays* 1

until you release it. This allows new ways to use long presses - namely selecting something while you hold it long enough and deselecting that immediately after releasing it.

2. New output "shortpress". This new output emits a trigger whenever the button is pressed. If "longpress" is used at the same time, the trigger "shortpress" is delayed until the button is released and only sent if the button was hold less than 1.5 seconds (this solves an issue reported by @oedoen).
3. New input "longpresstime" for setting the time for a long button press (default is still 1.5 seconds).

- [`buttongroup`]:
  1. New input "longpresstime" for setting the time for a long button press (default is still 1.5 seconds).
  2. New input "startbutton" for selecting one buttons that should be active on a Droid start when there is no saved state, or on a trigger to "clear" or "clearall".
  3. Improve behaviour when multiple buttons are pressed at once, to match your most probable expectations as what should happen. This is useful in situations where “maxactive” is greater than 1 and you want to enable, disable or toggle several buttons at once. The impact of a burst of presses is delayed for 25 ms in order to see the complete set of “simultanous” presses and toggle the buttons more intelligently with that knowledge.
  4. Improve behaviour in situations like where “minactive = 4” and “maxactive = 4” and the number of buttons is 7. Whenever you press an unselected button, now the button is switched off that was switched on the longest time ago. Such a configuration is useful for chord voicings.
  5. New trigger output “selectionchanged” that triggers whenever any of the buttons' states has changed. In case of button burst handling, the trigger comes as soon as everything is settled and thus maybe delayed by up to 25 ms.
  6. It's no allowed to omit the button inputs completely. Together with “maxactive = 0” this is ideal for making LEDs dark in situations where some buttons are not used by any circuit.
- [`math`]: Treat omitted inputs as neutral, not as 0. So for example a "product" with just one input will copy that input, not set it to 0.0. This is useful when you want to temporarily disable the line with the second input in your patch.
- [`fadermatrix`]: The new inputs “startvalue1” … “startvalue4” allow to set a start value for each column.
- [`algoquencer`]:
  1. If there is no saved state, all the steps now start *empty*, not all set as it was before. Seems more useful to me. If you lose your state (e.g. you start without the SD card), it's better not to play any beats per default rather than hit the drums on all clock beats.
  2. Added new output "lengthoutput", which outputs the currently selected pattern length (as suggested by @trickyflemming)
- [`notebuttons`]:
  1. Has now 16 presets.
  2. New input "startnote", which specifies the note to set on a Droid start when there is no saved state, or on a trigger to "clear" or "clearall".
  3. New output "gate" for creating a CV/gate keyboard (suggested by @effiksmusic)
- [`calibrator`]: Has now 4 presets
- [`nudge`]: Has now 16 presets

- [`midiin`]: You can now configure which “running” state the Droid should assume at system start. Previously it assumed the MIDI state to be stopped at the beginning and switched to “running” as soon as the first channel event of any type was received. Now you can choose the start state between “stopped”, “running” and “auto”. This is done with the new parameter “initialrunning”. Auto is similar to the current behaviour but just starts at “note on” events, so things like CCs would not simulate a start anymore.

- [`midiout`]:

  1. Now allows to send both to USB and TRS at the same time (use the new parameter "trs": set “usb = 1” and “trs = 1”)

  2. The “select” input now works slightly different. Previously unselecting the circuit simply plugged the MIDI output stream. That way any change in input gates etc. could get lost when they happened while the circuit was not selected. Now if unselected, the circuit simply does nothing and waits until it's selected again. So it will recognize any changes at the inputs as opposed to the last time it was selected and creates the necessary MIDI events to match the current situation.

  3. New trigger input “updateccs” that forces an instant update of all CCs (useful if the receiver has lost memory of the current CC states, e.g. because of powerloss etc).

- [`quantizer`]: New output "changed", which triggers every time the output value changes (same as "notechange" in [`minifonion`]) (suggested by @kylesignalsounds)

- [`switch`]: Increased the number of inputs and outputs from 8 to 16. Due to the general improvement in RAM usage and one specific further optimisation in [`switch`], The RAM usage of one switch circuit went down from a fixed 288 bytes to now 96 bytes plus 12 bytes per used input plus 4 bytes per used output. So a switch with four inputs and one output now needs just 148 bytes.

BUG FIXES

- Add missing X1 register to the status dump output.
- Add missing switches of S10 to status dump output
- Fix permanent reboot cycle in case of some broken invalid patch files.
- Fix permanent reboot cycle in case of a broken SD card (or another situation where you can write a patch file to the SD card but that can never be read by the master).
- Avoid permanent reboot if a constant in a Droid patch is longer than 15 characters (e.g. 5.3333333333333333)
- Fix possible crash on invalid negative controller number.
- Fix detection for double click on button right after reboot.
- Entering the USB stick mode via the X7 would never succeed if your patch created constant changes that need to be saved to the SD card. This has been fixed.
- Fix invalid free memory computation where you have 4294967291 Bytes of free RAM.
- Fix improbable but possible error in saving the state to the SD card (states of ciruits might be altered by reboot).
- Fix problem that lead to a patch taking literally or almost forever to load after pressing the button (found by @nandoll)
- M4: Fix colors of the RX.Y registers. The green and blue channels had been swapped. Now the colors exactly match those of the normal RX registers of the LED on the master.
- [`cvlooper`]: Fix the “gate is always high” bug found by @tmp and @oedoen. This bug appeared if you used one of I1 through I8 for the gate input and used an external gate signal with a slight DC offset that made the input have a non-zero voltage even if it was considered to be low. Even 1-2 mV were sufficient here. Temporary workaround in blue-2: use a gate input from the G8 instead or use a [compare] circuit on the input and compare the input against a threshold of e.g. 1V. In blue-3 this is now done correctly and the usual 1V threshold for gate signals is applied.
- [`clock`]: Fix strange effects when delay is used. The clock would still run even if the input clock was stopped. Or it would be delayed for minutes (found by @DeepSea).
- [`spring`]: Fix useless default value of "speed". That is now 0 (= normal speed). In fact the circuit was almost useless if "speed” was not patched, as the default of 1 (= 10V) meant a speed of 1024.
- [`slew`]: Fix reboot or reboot cycle when the CV for the exponential slew is super small (near to 0, e.g. 0.000000000000000001) (found by @supergreg)
- [`button`]: Do a bit more aggressive debouncing when “longpress” is used. That makes the detection of long button presses (and the suppression of the short presses) more reliable.
- [`lfo`]:

  1. Make timing more precise when the output of one LFO is used as taptempo input for a second one. They now keep the phase relation and run with the same frequency (found by @rss).

    2. Improve overall precision of timing (found by @Michael Gerdau). In blue-2 the LFO was running about 0.36% too slow. E.g. when you specified “hz = 100”, it actual frequency was about 99.63 Hz. This has been fixed and now it is 100.00 HZ (+/- 0.005 Hz).

    3. Improve the precision of taptempo input a lot. This also affects the taptempo inputs of other circuits. The error was about 25µs per input clock tick (found by @jpizzo, @rss and others).

- [`motorfader`]: Fix undesired alteration of a virtual fader value when a fader is not selected by any circuit and then selected, again (bug found by @tmp). Furthermore unselected

- [`motoquencer`]:

    1. Fix unsaved position of end in some situtations

    2. Fix situation where Metric Saver did not correctly resync

    3. Fixed “luckyamount = 0” for luckyfaders / luckycvs. Here 0 meant essentially maximum. Now the faders are correctly reset to the bottom position.

    4. Fixed missing notes in recording if recordmode = 1 (record without ties)

    5. Autoreset now counts correctly after a reset. It was one step too early the first time after each external set (found by @oedeon).

    6. Fix bug where the “gate off” of an inactive step was ignored when the previous step was on and the gate length was set to 1.0. In that case the gate output would last forever.

    7. Fix bug where a changing "cvbase" or "cvrange" would permanently alter your melody. Now it's like it should be: changes in these inputs change the range of the output pitches, but if you change back these inputs, the melody goes back to its original state.

    8. Fix sitation with very small “cvrange” where the pitch CV went outside the allowed range due to quantization.

- [`fadermatrix`]: "ledcolor" is now correctly just active when the circuit is selected and does not affect other circuits, like another fadermatrix or motoquencer anymore.

- [`algoquencer`]:

    1. **Finally the bug that could lead to a total system freeze in rare situations has been fixed!** The reason for this bug was that the circuit tried to place random beats into a bar **forever**. A lot of things needed to come together to trigger this bug and so your system could run for hours before the actual freeze. Thanks for @Loaf of Heresy and @jpizzo for reporting this and especially to @Mandroid, who was able to send my a Droid patch plus state file with that I was able to reproduce the problem reliably.

    2. The "lengthbutton" length was not updated on a preset switch when the algoquencer was not selected. The length is now correctly switched (found by @jpizzo and @spacejam).

    3. The “pattern” input now definitely sets the random seed for all random choices - if used. So by using that input you can now deterministically select something like a builtin “preset” for every random choice (dejavu needs to be 1 to avoid true random).

- [`pot`]:

    1. Fix the impact of "slope" for "lefthalf", "righthalf", "lefthalfinv", "righthalfinf" and "bipolar"

    2. Make the colors of the "ledgauge" match those of the R-registers (suggested by @reheller)

- [`midiout`]: Fix situation when you send just one CC, and no other CCs or notes. Here - due to MIDI head compression - if the receiver loses the first byte (or is rebooted), it would never catch up again. Now we make sure that at least every 250 ms a full MIDI event is being sent.
- [`sequencer`]: Fix impact of “gatelength”. Instead of being relative to one clock cycle as it should be, it was relative to just one 1 ms and the clock cycle was ignored. Now “gatelength = 0.5” sets the gatelength correctly to the duration of half a clock cycle.

CLEANUPS THAT NEED YOUR ATTENTION

In blue-3 we have made some cleanups that were necessary to make all the circuits more consistent, remove useless waste of memory or make other important repairs. **Some of these cleanups need your attention! You might need to adapt your patches in order to make them work on blue-3.** We apologize for the inconvenience, but believe that these changes absolutely make sense and make things easier and more straightforward for everybody in the future.

- The three circuits [togglebutton], [fourstatebutton] and [switchedpot] have been superseeded by [button] and [pot] long ago. This is the last firmware where they still work with one restriction: **These three circuits do not save their state to the SD card anymore!** This means that your patches still work but you will lose your button and pot states of these circuits when you switch off your system or reload your patch. In the next firmware these circuit will be removed completely. Forever.
    1. **Your task: Replace [togglebutton] and [fourstatebutton] with [button].** This is simple and can be done by renaming the circuits, because [button] can do both.
    2. **Your task: Replace [switchedpot] by [pot].** This is more work since you need to create one [pot] circuit for each output of your former [switchedpot] circuit. Use the “select” inputs for what you have done with “switch” in the [switchedpot].
- The registers for overriding the eight LEDs on the X7 are now R49 … R56 (they used to be R24 … R32, but since there are now up to 4 G8s, the numbers have shifted, even if you don't have or use any G8).
  Your task: If you are using R24 … R32, rename them to R49 … R56.
- [copy]: Removed "minimum" and "maximum" and "inverted". This saves memory and makes the circuit faster. You can simply use [math] if you need “minimum” or “maximum”. “inverted” is mostly 1 - X and can easily be done directly in the input.
  Your task: If you are using “minimum”, “maximum” or “inverted” from the [copy] circuit, replace it by [math] or by 1 - X.
- Circuits that have a “startvalue” input: Using this input does not prevent the state from being saved, anymore. Use the new “dontsave = 1” if you need that.
  Your task: if you depend on a state not being saved, use “dontsave = 1” instead of or in addition to “startvalue”.
- Some circuits had a "reset" input that was not for setting any clock to 0 or similar but loaded the default state back into the circuit (buttons, sequences, etc.) This is now done by the new trigger input “clear” instead.
  Your task: If you get a problem that your patch does not load because of “reset” not being defined, replace it by “clear”.
- In the [pot] circuit the input “resetvalue” has been renamed to “startvalue”.
  Your task: rename “resetvalue” to “startvalue” in your [pot]-circuits.

Here are the details about the circuits that are affected by the clear / reset / startvalue-changes:

[button]: The input "reset" has been replaced by "clear". It sets the button to 0 or whatever the value of "startvalue" is. "startvalue" does not any longer prevent saving the state to disk. use "dontsave=1" for that.

[pot]: The input "reset" has been replaced by "clear". And "resetvalue" has been renamed to "startvalue".

[`algoquencer`]: The trigger input "clear" now resets *all* settings of the algoquencer to its defaults: the steps, accents and alternate page, the pattern length and so on. But there is the new input "clearpage", which does what the old "clear" input did: clearing all beats from the current page.

[`matrixmixer`]: The trigger input "reset" has been renamed to "clear" in order to be consistent with the other circuits. Also the use of "startvalue" does not disable saving the SD anymore.

[`nudge`]: Same: please use "clear" instead of "reset" now. Also the input "persist" has been removed. Use "dontsave" instead.

[`motorfader`] and [`faderbank`]: Use now "clear" instead of "reset". "startvalue" has no influence on the saving anymore. This is done with "dontsave". "resetvalue" has been dropped. use "startvalue" instead.

[`calibrator`]: "reset" has been replaced by "clear".

X7

If you are using the Droid X7 expander, you will find some changes here. These require a firmware upgrade of the X7. The new firmware version is **orange-1012**. Please consult the Droid manual for how to do that.

The old orange-911 firmware is still compatible, so there is no immediate need for the update. I recommend it if you have experienced the bug with sporadically lost MIDI data on the TRS output.

Here are the changes:

- The MIDI TRS output could lose one byte of data sporadically (maybe after minutes, hours or even never). Sometimes that did not have any audible effect. Sometimes, however, a “note off” got lost and you got a stuck note (as @eising experienced). This has been fixed. The fix does not address any issues in MIDI over USB.
- The next hardware revision of the X7 (Rev 1.5.1) supports using USB-C to C cables (small modern plug on both sides). The transition from the previous revision (Rev 1.3) to 1.5.1 happens roughly during 2023. There is no other difference between these two hardwares. Just and solely the support of USB-C to USB-C cables. *We apologize for the fact that the X7 of the first revision does not allow the usage of these cables, just USB-A to USB-C!*

  Note: even if you X7 is already has hardware 1.5.1 ,it might still not work if your module has been shipped in 2023.

USER MANUAL

As always in a release, the user manual also has been updated. Most things are clarifications, bug fixes and of course the chapters about the new circuits or new features.

This time there are some new general chapters that help you creating cool and versatile patches. Try and read them, this might ease your life ;-)

- New chapter 3.1 about overlaying buttons and pots with multiple functions (using select and selectat)
- New chapter 3.2 about presets
- New chapter 3.3 with details of how taptempo inputs behave
- Rework of chapter about the controllers, including a new trouble shooting section.
- New chapter about the R2M/R2C controller bridge modules.
- Add missing documentation for the [`polytool`] circuit.

DER MANN MIT
DER MASCHINE

DROID FORGE

- Support for the new firmware blue-3
- Support for the new upcoming P8S8 controller
- Support for the new G8 with up to four G8s on one master
- Preferences: new options for disabling the regular check for the existance of the X7 and/or the SD card. Switching off this fixes stability problems encountered on some Windows systems.
  Note: because these new options are necessary to make the Forge usable on some Windows systems, they are active per default on Windows. This means that your "Activate!" and "Save to SD" icons now are always active even if there is no X7 connected or new SD card present.
- Make section names work that start with dashes, e.g "---My Section" (found by @Dysonant)
- In the source code editor the keyboard shortcuts for increasing and decreasing the font size now work (usually Cmd/Ctrl + and - and 0).
- In Windows Ctrl-= is now an additional shortcut for "Zoom in", so you don't need the shift key any more on an English keyboard (found by @ImaginedWorld).
- Make background in "text mode" white, when light theme is selected, to improve readability (found by @milgaus).
- Added links to helpful Youtube videos to the "Help" menu.
- New "Views" option to mirror the icons for the cable plugs (change left and right). The mirrored orientation seems to be more natural for some users.
- Making a register label empty with Cmd-L and removing the text now correctly removes the label and does not create an empty label anymore.
- New command "Duplicate circuit" with short cut Cmd-D (as suggested by @MSMS).
- New icon for "New patch", because the old one was the same as for "New parameter" (found by @MSMS).
- Fix bug where a mouse click in the "Add circuit" dialog selects the wrong circuit if the list is scrolled (found by @greg).
- "Paste smartly" now does **not** increase the number of buttons, pots, leds and other things on controllers if the copied circuit contains a "select" parameter. Numbers in cables and input/output registers are still dealt with. The assumption behind this is, that the buttons and LEDs in question are overlayed via "select" and thus to be reused by the copied circuit (suggested by @eising).
- Fixed a glitch found by @trevormeier: "If you edit a field and use the arrow key to finish entry and move to the next field, in the future whenever you press enter to finish entry, the cursor/active field will move in the same direction as the last time you used the arrow key." Pressing enter now always leaves the cursor in the same cell.