

DROID blue-6

and DROID Forge 1.6

Release notes

more free RAM, less patch SIZE, more SPEED!



These are the release notes for the DROID blue-6 firmware for your DROID master and for the DROID Forge 1.6. Please always update both at the same time.

Contents

Hints for updating.....	2
General improvements.....	3
Adapt Your Patches?.....	4
New features.....	5
Bug fixes.....	6
DROID Forge 1.6.....	7
Motor Fader Performance Sequencer (MFPS).....	9

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, 4 and 5 and update directly from blue-2 to blue-6, if you like. Please read the release notes of the skipped releases. 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 continue with upgrading.**
- 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.

Hint: The new Forge has a new and easy firmware upgrade feature. You can use this, if your master has at least blue-2. You find it in the menu “File / Upgrade firmware of master”.

If you use any of the new patch compression options, make sure that the firmware of your master modules matches the one that is supported by the Forge (as displayed in “About DROID Forge”).

If you are using the Moto Kit with the standard patch and no Droid Forge, unpack the contents of motokit-1.6.zip to the SD card and press the button of the MASTER18.

GENERAL IMPROVEMENTS

- **RAM:** The biggest bang in this new firmware is that it saves lots of RAM. This means that you can put even more stuff into your patches. Some RAM is simply saved by updating the Forge and your master's firmware to the new version. But the most impact has the new option "Detect and share duplicate values for inputs" in the Preferences dialog. Here is a comparison of the RAM sizes of the Moto Kit MFPS default patch:

	blue-5	blue-6	blue-6 with shared values
RAM used by patch	103,074	94,980	82,160
free RAM	5,201	14,147	26,967
free %	5%	13%	25%

- **SIZE:** The second optimization is that of patch size. You might have run into the limit of 64,000 bytes for the total patch (not counting spaces and comments). There is already a global option in the Forge that renames long patch cable names into short ones, which saves lots of space.

Here is the new next optimization: every input and output name now has an abbreviation. For example instead of button2 = B1.2 you can now write b2 = B1.2. This is done automatically by the Forge if you enable the option "Use abbreviated parameter names" in the preferences. In the MFPS default patch this reduces the patch size from 42,527 to 25,761 bytes!

- **SPEED:** The third optimization is less visible but all the more important and was the most work for us. The execution times of the patches has been vastly reduced by more than 30 internal optimizations in the firmware. For example the Moto Kit default patch used to have a loop time of approx 2.5 ms in blue-5 and now needs just 0.8 ms for one cycle. That is a speedup of about 300! Shorter loop cycles mean more accurate timing, less jitter, less latency. If you are curious about the loop time of your patch: it is now displayed in the status dump (see manual).

ADAPT YOUR PATCHES?

The good news is: In most cases you don't need to change anything in your patches after updating to this version. There are two situations, however, in which this could be necessary. Both have to do with a speed optimization that is called "UI Slowdown". This optimization utilizes the fact that pressing a button or turning a pot needs a really long time when compared to other internal stuff that runs on a micro second resolution. A few number of milliseconds of delay after pressing a button is usually not noticable.

So in blue-6, circuits that are meant as a part of the "user interface" are now executed at a slower rate (with the benefit that all other circuits are executed on a *higher* rate). Specifically this means that these circuits are just executed in every 8th loop cycle. This affects `[button]`, `[buttongroup]`, `[pot]`, `[motorfader]`, `[faderbank]`, `[fadermatrix]` and the user interface part of `[motoquencer]` (its actual sequencing runs at full speed, of course). The circuits dealing with encoders are excluded from this optimization for the while but will follow later.

As long as your patches are built in a "normal" way, this does not have any noticable negative effects. The two situations where your patch could go "weird", anyway, are these:

Situation 1: You use UI circuits like `[button]` for non-UI tasks like negating signals or building "high speed" oscillators. Consider this example:

```
[button]
  button = 1 - _LED
  led = _LED
  states = 1
  output = 01
```

In blue-5 this creates an oscillator that changes its state every second loop cycle. In blue-6 this is running at a much lower speed. Even worse, when other non-button-circuits depend on a direct immediate effect, weird things can happen. So the learning is: don't use `[button]` or `[pot]` for such unusual tasks. Checkout `[math]` or `[flipflop]`. They do a better job. Also the upper example completely floods your "saved state" file, since all these "button presses" are saved to your SD card!

Situation 2: Incomplete selects can be another sources of weirdness. When several circuits share one LED, you must make sure that at any time *exactly one* circuit is controlling this LED. This is nothing new. But in blue-5 the effect of not doing this was that the circuit that was later in your patch usually won the battle for the LED and overrode what the previous circuits did. In blue-6 this later circuit might not be executed in the same loop cycle as the previous one and so both circuits control the LED from time to time - resulting in flicker. The learning is: Check your selects. Make sure that at most one circuit is controlling an LED at any time.

But: If you don't want to deal with all these details, you can switch of the UI slowdown optimization by a new parameter in the `[droid]` circuit like this:

```
[droid]
  uislowdown = 0
```

if you do this, these "weird" behaviours should go away.

NEW FEATURES

- Abbreviations for parameters: every input and output now has an abbreviation. Instead of `button2 = B1.2` you can write `b2 = B1.2`. If you write your patches manually, look at the cheat sheet, which is contained in the ZIP file of the Droid firmware in the folder “manual”. There is a new column with the abbreviations. In most cases the abbreviation is one or two characters, rarely three.
- Massive speed update by optimizing the code at more than 30 places. This reduces the jitter and latency and improves the timing precision of your patch.
- Status dump: Added the information about the average loop cycle duration (this is the time in which all circuits, inputs and outputs are computed / sampled).
- **[ifequal]**: New circuit for a simple comparison. This does a subset of what **[compare]** does, but executes faster. The circuit was introduced to speed up the Motor Fader Performance Sequencer (MFPS) as it uses a lot of compare circuits.
- **[buttongroup]**: New trigger output “extrapress”. This sends a trigger when you press a button but the selection does not change. The feature is used by the MFPS now: if you hit the track button of the already selected track, it toggles the page.
- **[buttongroup]**: The input “startbutton” now supports the new value -1. This sets the maximum number of buttons to “on” at start as a trigger to “clear”. It makes only sense if you have `maxactive > 1`, of course.
- **[motoquencer]**, **[encoquencer]**:
 - In the button mode “skip”, the non-skipped active gates are now shown in cyan.
 - In “Repeats” mode the LED below the fader/encoder now is red for the number 4, 8, 12, 16 and yellow for 2, 6, 10 and 14, while you move the encoder fader.
- **[motorfader]**: You can now give “ledvalue” a negative value. This enabled a special mode where the LED is automatically highlighted (more brightness) when the fader is in the position of its start value (= default value). This feature is used in the MFPS. In version 1.5 it had been built with lots of extra circuits. These are now no longer necessary, so RAM and CPU is saved. See the manual pages for the “motorfader” circuit for details.
- **[droid]**: New input “uislowdown”. You can set this to zero, if you want to disable that part of the speed optimizations that slows down the execution of UI circuits like **[button]**, **[buttongroup]** and **[pot]**.
- **[algoquencer]**: the input “pattern” can now be combined with `nextpattern`, `prevpattern` and `reroll`. Up to blue-5 these three inputs were ignored when “pattern” was used.

BUG FIXES

- In rare situations the state of some circuits were not correctly saved. This has been fixed.
- MASTER18: The values of the output registers in the status dump were mixed up (e.g. O8 was displaying the value of O7). This has been fixed.
- MASTER18: When you did a patch upload via USB twice in a short time – specifically if you did the second one before the first one actually was complete, a garbled patch would be uploaded and would not work. This happened for example when you pressed the “Upload” icon for a larger patch twice in a row. This has been fixed.
- [**miditrough**]: The parameter “totrs” was completely disfunctional and ignored. It had no effect. The only chance of forwarding MIDI to TRS was *not* using the parameters “tousb” nor “totrs”. It was then falling back to the default to sending the events to the first TRS output. This has been fixed. Found independently by [@supergregg](#) and [@Tim Riot](#).

DROID FORGE 1.6

New features

- New option “Detect and share duplicate values for inputs” in the Preferences. This option enables a new RAM optimization, which makes use of the fact that in a typical patch many input values appear over and over again. Now the code shares internal data structures. This saves substantial RAM. See the chapter “GENERAL IMPROVEMENTS” above for a comparison.
- New option “Use abbreviated parameter names” in the preferences. This option is off per default. Enable it if your patch size exceeds the limit of 64,000 bytes and you will get a substantial reduction of the size.
- New menu entry “Export compressed patch”. It saves the contents of that “droid.ini” to a file that is deployed with “Save to SD” or “Activate” and comprises all compressions.
- Removed patch compression option “remove empty lines”. They are now always removed. Previously this was optional so that any error messages by the master (blink codes) would show the correct error line. With the maturity of the Forge in these days we assume that Forge generated patches don't generate errors anymore, so we got rid of this option.
Hint: If you **do** run into a situation where your master an error in a specific line (which should never happen), you can use the new option “Export compressed patch” and open the exported file with a text editor to check the line number there.
- New convenience function “Upgrade firmware of master” in the File menu: It copies the firmware files for the MASTER and MASTER18 to the SD card. The nice side-effect: the firmware of the master modules is now contained in the Forge (as well as the manual), so in most cases you don't need to download the ZIP file of the firmware anymore. An upgrade of the Forge is sufficient.
- New “Preferences” setting “Always save master's firmware to SD card”. If it is enabled, a “Save to SD” will always also copy the firmware files for the MASTER and MASTER18, so they will automatically upgrade if necessary.

Patch generators

- You can now specify a path to the Python3 interpreter that should be used to run patch generators (suggested by [@eising](#)). You do this in the Preferences . This is a feature for more experienced users who have more than one Python installation or use an installation with non-standard paths.
- Ignore subdirectories in DROID Patches/Generators. They might be part of Python modules. In 1.5 any subdirectory generated an error (found by [@eising](#)).
- When you open the dialog of a patch generator, it will now check if all presets are complete. This is important for self-written patch generators. Incomplete presets can lead to weird behaviour. Now you are warned and can fix this.

Bug fixes

- beta2: When loading patches [algoquencer] pitch (output) would be converted into pitch1 (input). This has been fixed. Found by [@Glass](#).
- beta2: Fixed crash when loading patches that contained [clocktool] with “inputpitch” (Found by [@jpizzo](#)).
- beta2: Fixed left-over formatting stuff in documentation of “harmonicshift”.

- Fix dragging of encoders. When you specify an encoder with E1.1 (rather than just 1), you can drag & drop it like buttons and pots. But the button within the encoder was not dragged, also not its label. This has been fixed. Found by [@Effiksmusic](#).
- Fix the setting "Do not treat unknown parameters as errors". It had no effect. Probably never had. Sorry for this. Found by [@Effiksmusic](#).
- Fix editing source code of patch section: It would lose the first comment line of the first circuit.

MOTOR FADER PERFORMANCE SEQUENCER (MFPS)

These changes affect the patch that is generated with the patch generator “Motor Fader Performance Sequencer”.

If you have a Moto Kit and use the patch that has been shipped with it, you can get an updated patch on our [downloads page](#) or in the #droid-motokit discord channel. There is a ZIP file for the Moto Kit. Copy both master18.fw and droid.ini to the SD card of your MASTER18. This updates your master and loads the new sequencer patch.

You can also use the Forge and create the patch yourself with the the patch generator “Motor Fader Performance Sequencer” and load the default preset. Don't forget to update the Firmware of your MASTER18, as well.

Bug fixes

- The sequencer is now consistently named “Motor Fader Performance Sequencer”, with the abbreviation MFPS.
- Fixed a patch problem, when you change the number of tracks to 1.
- Fixed a patch problem, when there are more functions mapped into the performance menu than you have faders. This is for example the case if you change the number of tracks to 5 in the default patch.
- Fixed the octave switch during tuning mode. It was applied twice, so it switched **two** octaves up and down per step. It's now just one as it should.
- If there is no saved state on the SD card, the sequencer's clock now starts “running”, not “stopped”.

New Features

- Increase the maximum “swing” value a little bit.
- As requested by many users, the fader “SWING” now also affects external clocks. This is a bit unlogical since this fader is contained in the menu for the internal clock and also is only present if you have enabled the internal clock. Nevertheless it's obviously very useful. It is still possible to supply a “swung” clock from an external source instead.
Fun fact: It's possible to use an already swung external clock and still apply swing from the internal clock fader. Depending on the phase of the swing these two swing settings either add up or cancel out.
- A press on the track button of the already selected track now does the same as the PAGE button: it toggles the current page (steps 1-8 or 9-16). This was suggested by [Mylar Melodies](#).
- In the REP mode (skip/repeats), the touch buttons (which select steps to be skipped) now show enabled steps in cyan, if they are not skipped. And while you move the fader, the LED below it is red for the numbers 4, 8, 12, 16 and yellow for 2, 6, 10 and 14. This helps dialing in specific values.
- If hold hold CTRL while pressing one of the interval buttons ROOT, 3RD, ... 13TH, only that button stays active, all others are switched off. That way by holding the CTRL button, you can now “perform melodies” with these interval buttons. Also it allows you to quickly switch off all buttons in order to make a completely new selection. Try it out!
- Constant length: The CTRL button has got yet another *magic* function: When you hold it while changing the number of repeats of a step (button REP), the total number of repeats in your sequence stays constant, if possible. So if you hold CTRL and move a fader to reduce the number of repeats on a step, the next step(s) will add repeats and their faders move accordingly. It's really magic - try yourself...