MBPV: The Matrixbrute Patch Viewer

v. 0.5 – 2022-03-12 (beta)

Github: https://github.com/MatrixbrutePatchViewer/MBPV **Contact**: matrixbrutepv@hotmail.com

Important notice:

- This software is written for **my own use**, it's **not** associated with Arturia in any way. This is an early version, but most functionality is in place.
- It is **not** to be considered as "professionally released software", and **no guarantee in any way** is given. **Use at your own risk.**
- It is shared for free in the hope that it can be useful for other Matrixbrute owners. You're not allowed to sell this software or charge for it.
- The program will only **read** the Matrixbrute files exported by MCC (no editing/writing is possible during normal operation).
- It can't connect to the Matrixbrute directly.
- All data formats are decoded by users (not based on any Arturia documentation), so info shown in a preset/patch might be inaccurate, missing or wrong – and might stop working after a firmware update.

- **You should always keep backups of your important files on another drive, and also off site (cloud / another location).**

The Matrixbrute Patch Viewer (MBPV for short) is a program to view many of the parameters a preset is made of, including sequences, mod matrix, presets and panel settings. This gives a better overview of a preset than looking at the hardware synth.

MBPV can only open files exported from **Arturia Midi Control Center** (MCC) for the **Matrixbrute** (not Polybrute!) of type **Project** - 256 presets (.mbprojz), **Bank** – 16 presets (.mbbz) and **one preset** (.mbpz).

MBPV is NOT an editor (nothing can be changed or saved) / real time viewer (it's not currently possible to connect it to the MB)

Contents

Installation and running on Windows

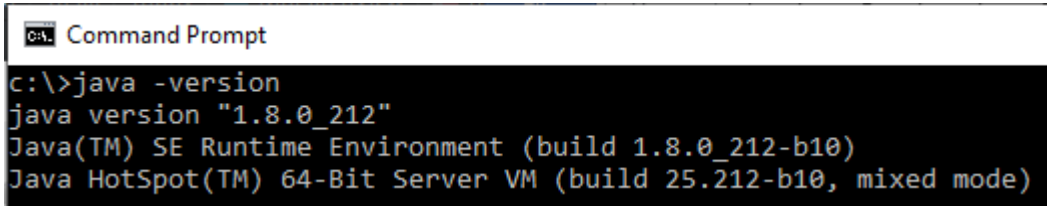
This application is written in java using OpenJDK version 11, and to run it you need java to be available on your machine in version 11 or higher. Java can be downloaded from: https://jdk.java.net/java-se-ri/11 or as a direct link (windows 64bit): https://download.java.net/openjdk/jdk11/ri/openjdk-11+28_windows-x64_bin.zip

Check if you have java already:

- Open a command prompt (**win key+R**, then type **cmd**), then in the **Command prompt** window that opened, type **java -version**
- If you get ***'java' is not recognized as an internal or external command*** java is not installed on your machine.
- If java is installed, check that the version returned is **11** or greater (java 8 is reported as version 1.8, and it's common to have this version installed already)

The image to the right shows I have Java 8 (version 1.8) installed as default.
To use MBPV this output (or higher than 11) is required:

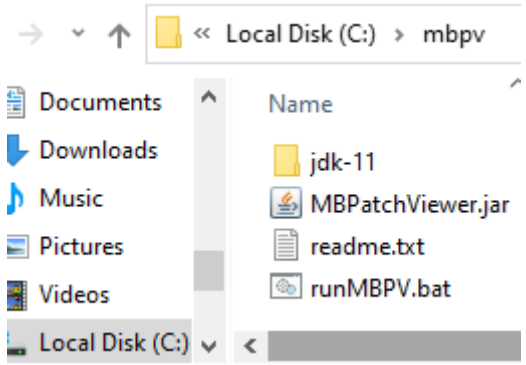
```
openjdk version "11.0.2" 2019-01-15
OpenJDK Runtime Environment 18.9 (build 11.0.2+9)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.2+9, mixed mode)
```



Example installation

Use this if you don't have java at all, or have a lower version a default that you don't want to disturb – this way java 11 will only be used for MBPV:

Make a folder **c:\mbpv** (or wherever you like, just substitute with your actual folder below).
Unzip the downloaded java zip into that folder, to get this structure: **c:\mbpv\jdk-11**
When done correctly you should find java.exe here **c:\mbpv\jdk-11\bin\java.exe** and you can (from the command prompt) type this to check the version (should be `11` as shown above): **c:\mbpv\jdk-11\bin\java.exe -version**



You can also put **MBpatchViewer.jar** in that folder, and make/edit the **runMBPV.bat** containing the following to start the application with a double click on a shortcut on your desktop, or on the bat file in windows explorer.
Replace **c:\mbpv** with your actual install folder for java and the jar-file:

```
start c:\mbpv\jdk-11\bin\javaw -jar c:\mbpv\MBPatchViewer.jar
```

(The **start** command makes the command prompt window disappear when the application has started)

Installing and running on Mac (not tested!)

Everything is written in java and should also work on Mac, but it's not tested. As for windows, it needs java 11+ and starting it should be similar as on windows (but with "mac"-specific paths etc). Script to start might be something like this (but java might need a full path prefix if the needed version is not in the path):

```
java -jar MBPatchViewer.jar
```

For options (as described in the **Configuring the application** chapter below), the -D switches should work the same way, but the path must be mac-specific.

Configuring the application

It is possible to configure some startup values for the program, those are given as **-Dkey=value** switches. If **value** contains spaces it must be surrounded with "".

The available keys are the following (all are optional, and ignored if syntax is wrong). **Must** be written exactly as below, with same case:

- **path :** set the folder where your MB files is located. **Note**: use / not \ in path, and "" around if the path contains spaces!
- **exportPath :** set the folder where you want exports saved. Same **note** as for path.
- **file :** sets the filename you want to open when starting MBPV, should be only the filename without path (like **Factory_2.0.mbprojz**). Require "" if spaces in name.
- **zoom**, **zoomPanel** and **zoomMatrix :** Sets the zoom level for **both**, just **panel** or just **matrix** window. Legal values 50-200 in steps of 10 (50,60,70…190,200)

Example using **runMBPV.bat** custom configs - see the included runMBPV.bat for more info. Replace `c:\mbpv` with your actual path (must be done in runMBPV.bat):

Ex1: This will start MBPV with both windows zoomed to 120%, and opening the file **My MB.mbrojz** located in **C:\My Patches\Matrixbrute** (note: / not \ in the bat file!)

```
start c:\mbpv\jdk-11\bin\javaw -Dzoom=120 -Dpath="C:/My Patches/Matrixbrute" -Dfile="My MB.mbrojz" -jar c:\mbpv\MBPatchViewer.jar
```

Ex2: This will start with the Panel window (and all other future windows!) zoomed to 150%, but the matrix window at 120%

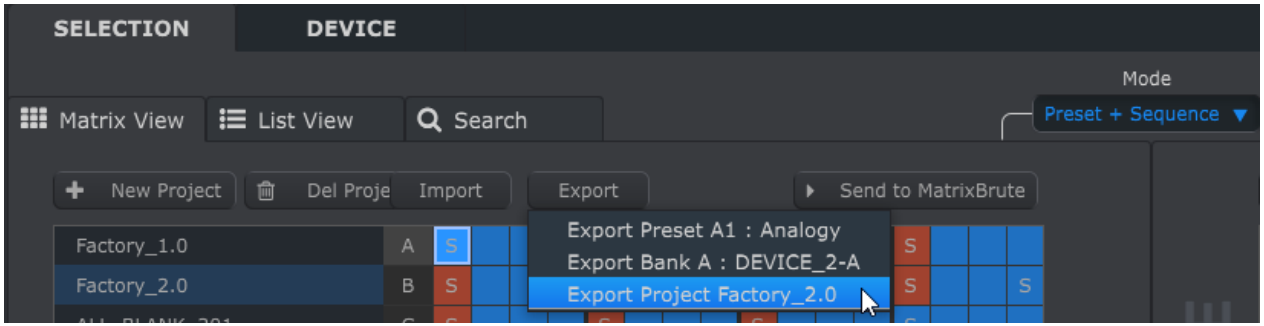
```
start c:\mbpv\jdk-11\bin\javaw -Dzoom=150 -DzoomMatrix=120 -jar c:\mbpv\MBPatchViewer.jar
```

Ex3: This will start specifying folder for project files and exported files, and saves you from finding the folder manually each time you restart MBPV

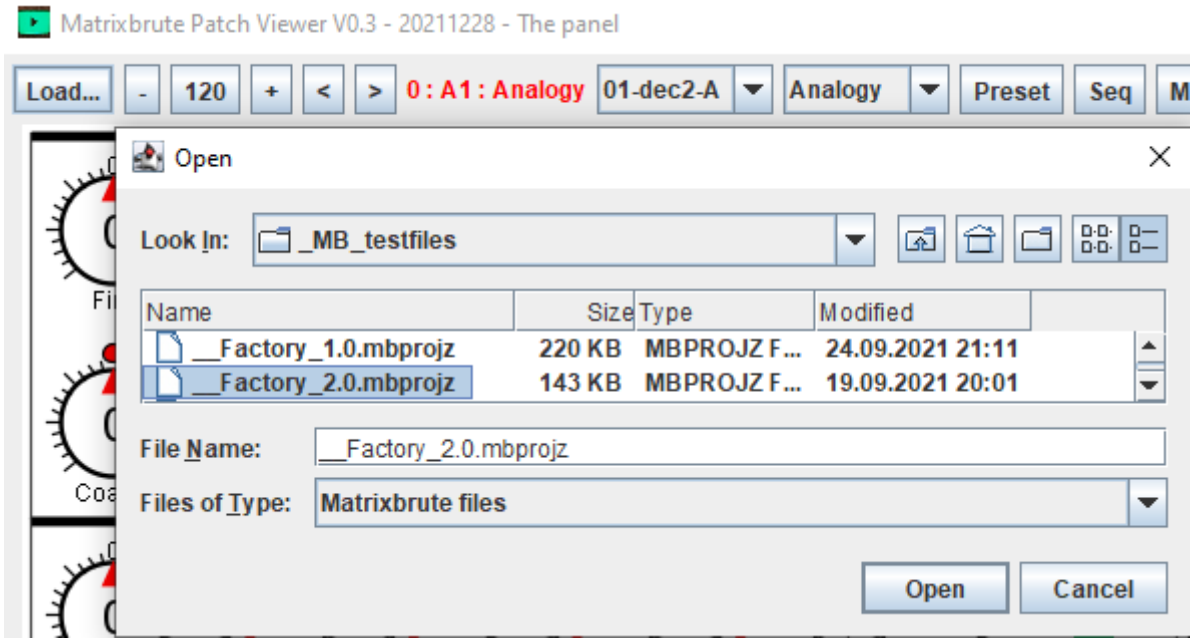
```
start c:\mbpv\jdk-11\bin\javaw -DexportPath="C:/mbpv/exports" -Dpath=" C:/mbpv/projects" -jar c:\mbpv\MBPatchViewer.jar
```

Getting the Matrixbrute data into MBPV

1. Export your project / bank / preset from **Arturia Midi Control Center** (MCC) as shown in this picture. Remember the folder you save it to.



2. Load the exported file into Matrixbrute Patch Viewer, using the **Load…** button



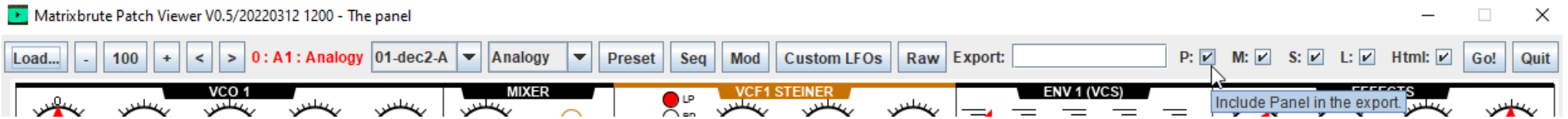
3. **Tip**: You can configure the default folder MBPV will show when clicking "Load" using the **–Dpath** option, see the **Configuration the application** chapter above.

General information

The application contains 3 windows, at the moment only one instance of each can be open at any time. Quit the application by closing the **Panel** window.

The panel window represents the front panel of the Matrixbrute. The matrix windows represents the matrix, and contains 4 views corresponding to what the matrix can show. Finally the "Raw" windows can dump & diff preset data in a raw format (numeric). Mostly useful for debugging/decoding, but can also be used to see the difference in multiple patches.

The panel and matrix window both have the same toolbar. All buttons have tooltips, so rest the pointer over to see more info.

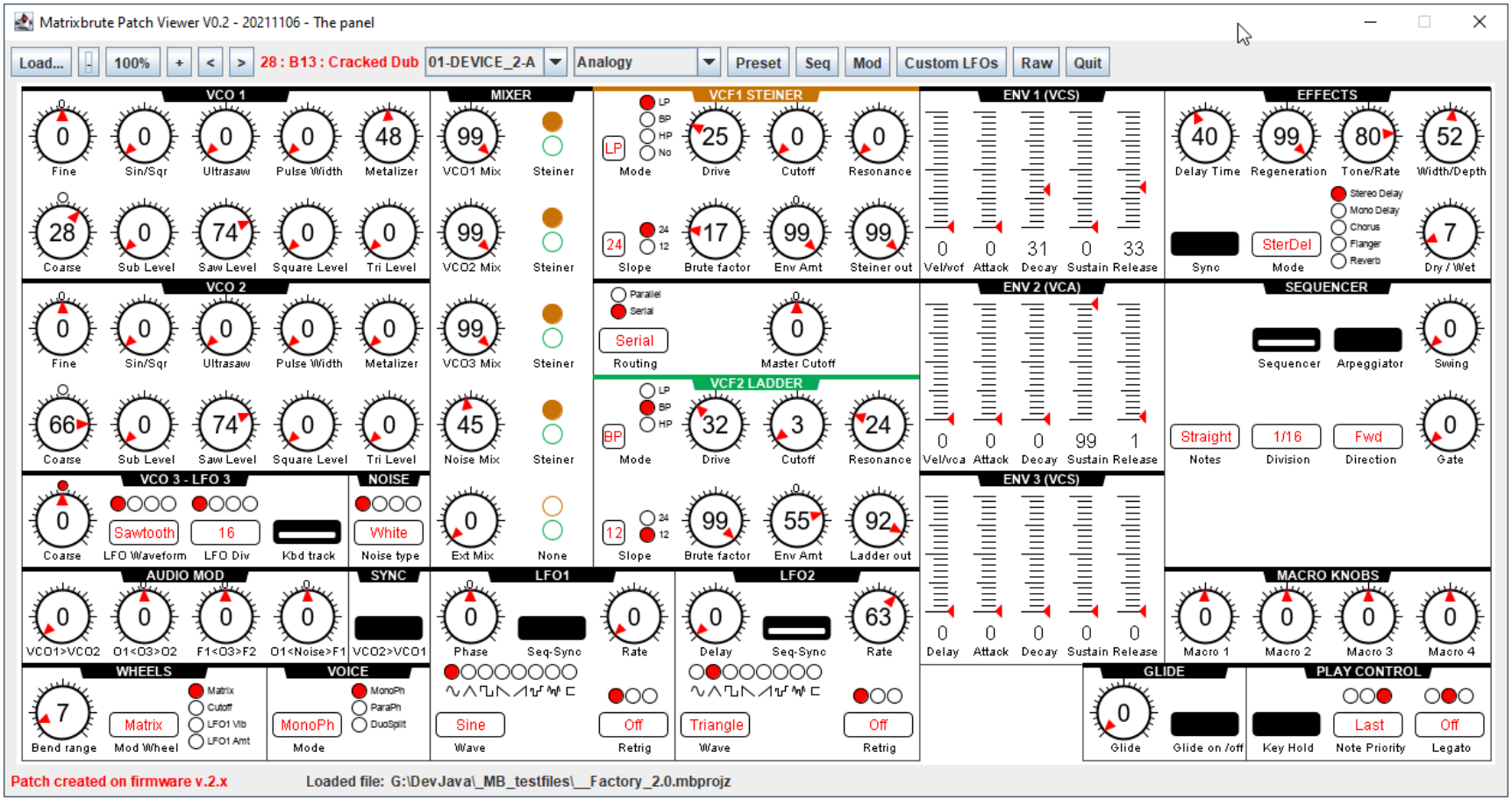


- **Load…** : Opens a dialog for picking the file to open. Default location can be configured with the –Dpath option when starting. See configure-chapter.
- **-** (minus) : zoom down
- **100**: The current zoom level (50-200%). Click to reset to 100%
- **+** (plus): zoom up
- **<** and **>** : choose previous/next patch
- **1:A2:Bass of Steel** : The preset number (0-255), preset slot (A2) and preset name
- **Dropdown** to select the bank
- **Dropdown** to select the preset within the chosen bank
- **Panel** : not present in this picture, but in the Matrix window it will set focus to the Panel window
- **Preset**: Open matrix window in **Preset** mode (listing all preset, which also can be selected)
- **Seq**: Open matrix window in **Sequencer** mode (also shows Matrix Apreggiator when relevant)
- **Mod**: Open matrix window in **Mod Matrix** mode
- **Custom LFOs**: Open matrix window in **Custom LFO** mode
- **Raw**: Opens the window for dump & diff presets (numerically)
- **Export –> Go!** : Export to images and HTML, see chapter **Export to images and HTML** for more information.

The bottom of the window contains the filename currently loaded, and in the lower left corner info about the firmware version (FW) the patch was created on (or "empty patch" if it's no data). A patch created on FW1, then imported to the Matrixbrute will remain in FW1 version even if exported back into MCC. Only **saving** the patch on the Matrixbrute with a higher firmware will update it to FW2 version.

The Panel window

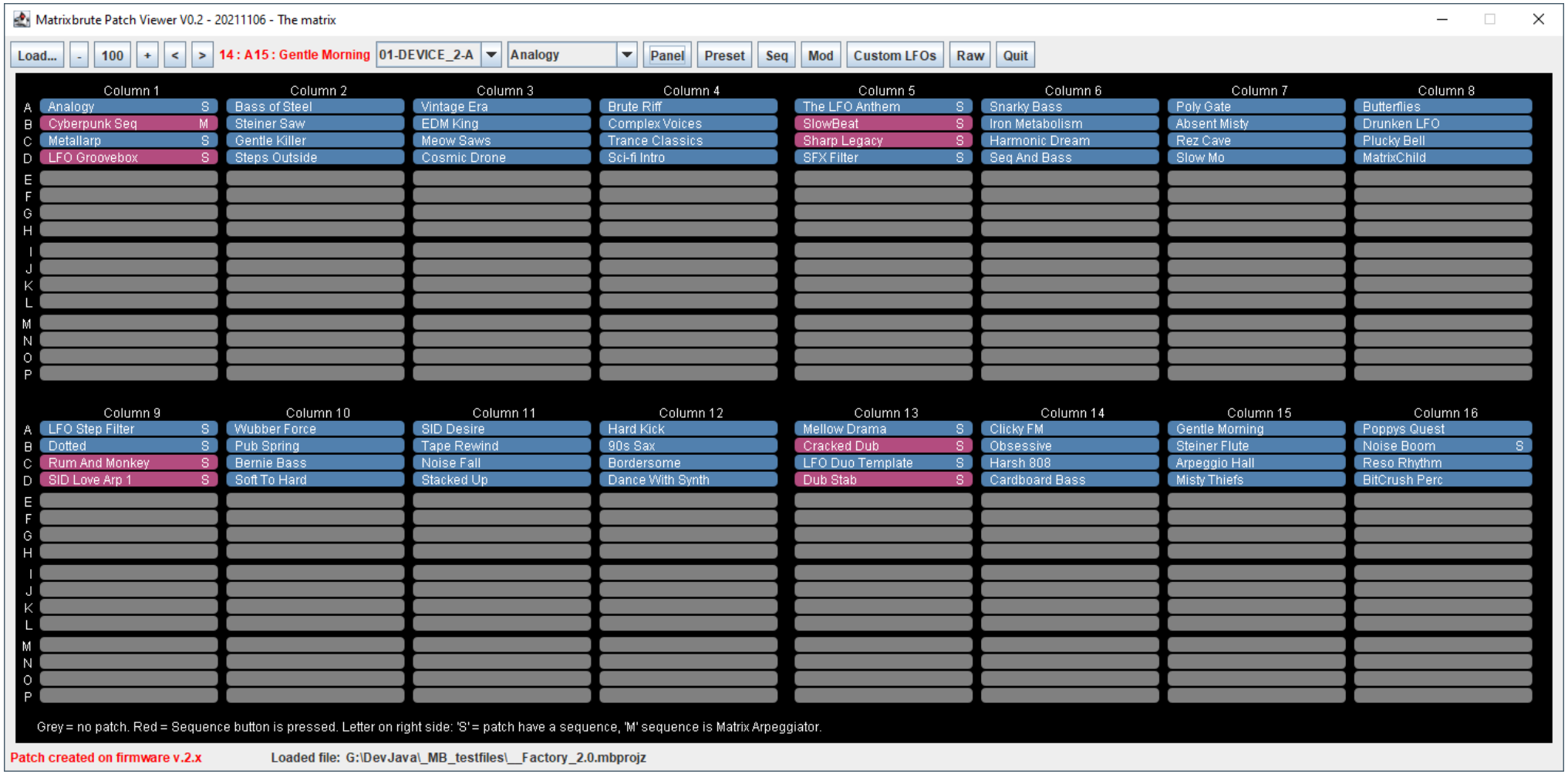
This window always contain the front panel, and the standard menu bar for setting zoom, selecting patch and opening the matrix window in different modes.



- The layout is based on the layout on the Matrixbrute, but the controls to the left of the keyboard is placed in bottom right (Glide, Play control and Macro knobs)
- LFO-section have one extra wave LED-light "c" (for **custom**), but to be consistent with the Matrixbrute all lamps is lit when custom is chosen.
- Push buttons (like Sequencer and Arpeggiator) are black if **off**, and have a white line if **on**. The sequencer button in the pic above is on.
- Rotary knobs have a number inside for the value (normally 0-99 or -99 to 99 depending on the knob type). A red arrow also indicate the position of the knob
- Parameters with multiple values (like Noise Type) have LED-lights matching the actual layout, and also the value written inside the push button.
- This view is on purpose not using actual images of knobs & buttons, but tries to be as clear as possible to read from a distance when sitting at the MB.
- **Note:** The sequencer swing and gate knob does not work in the current version.

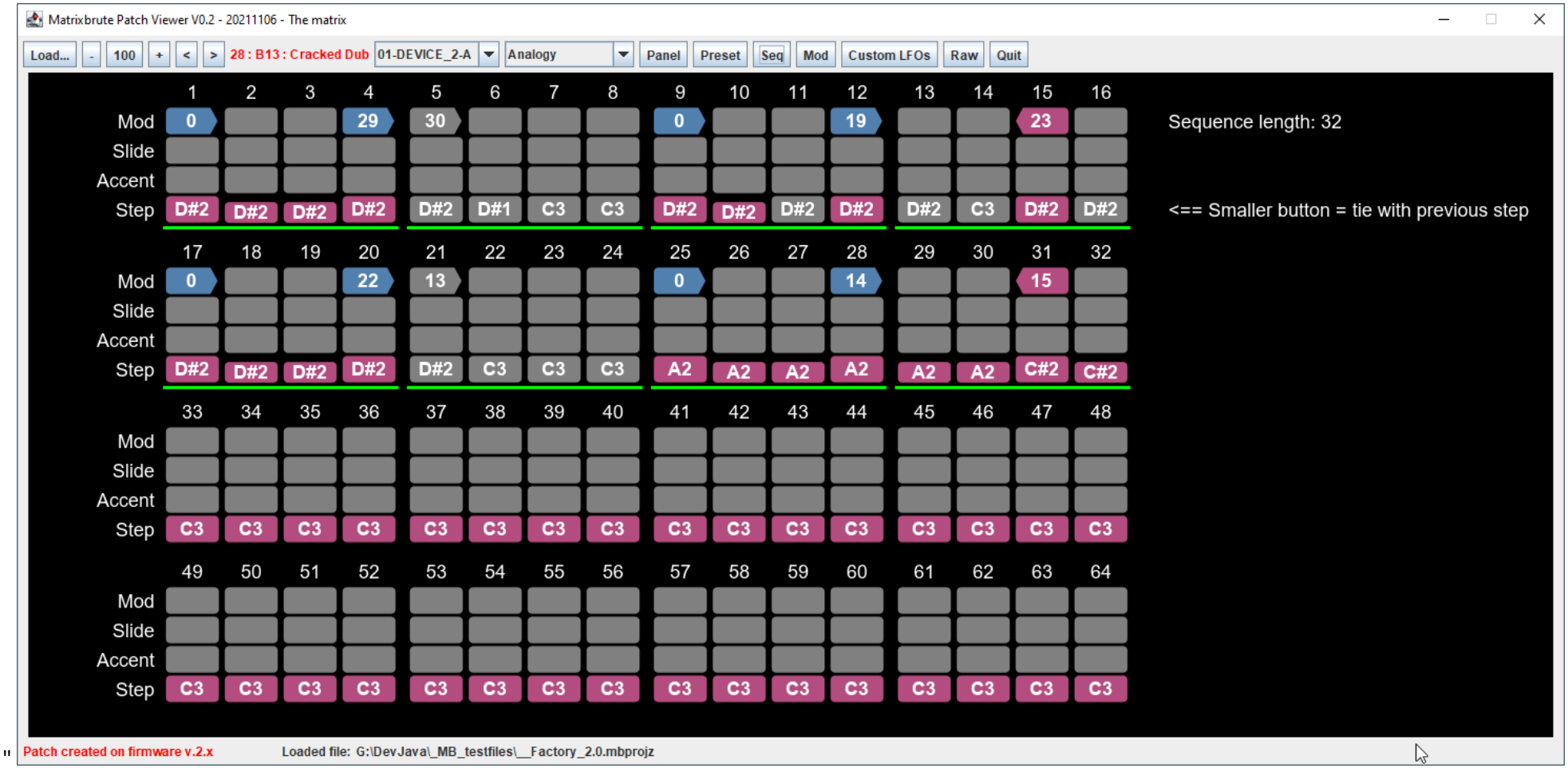
The Matrix window - Preset view ("Preset" button)

This view lists all presets with name and indication of sequence or not (see short description in the bottom of the window). Due to space limitations the left 8 columns are shown as 8x16 fields at the upper part, and the right 8 columns in the lower part. Click any preset to select that one, and the panel window will be update with it.



The Matrix window – Sequence view ("Seq" button) – Sequencer mode

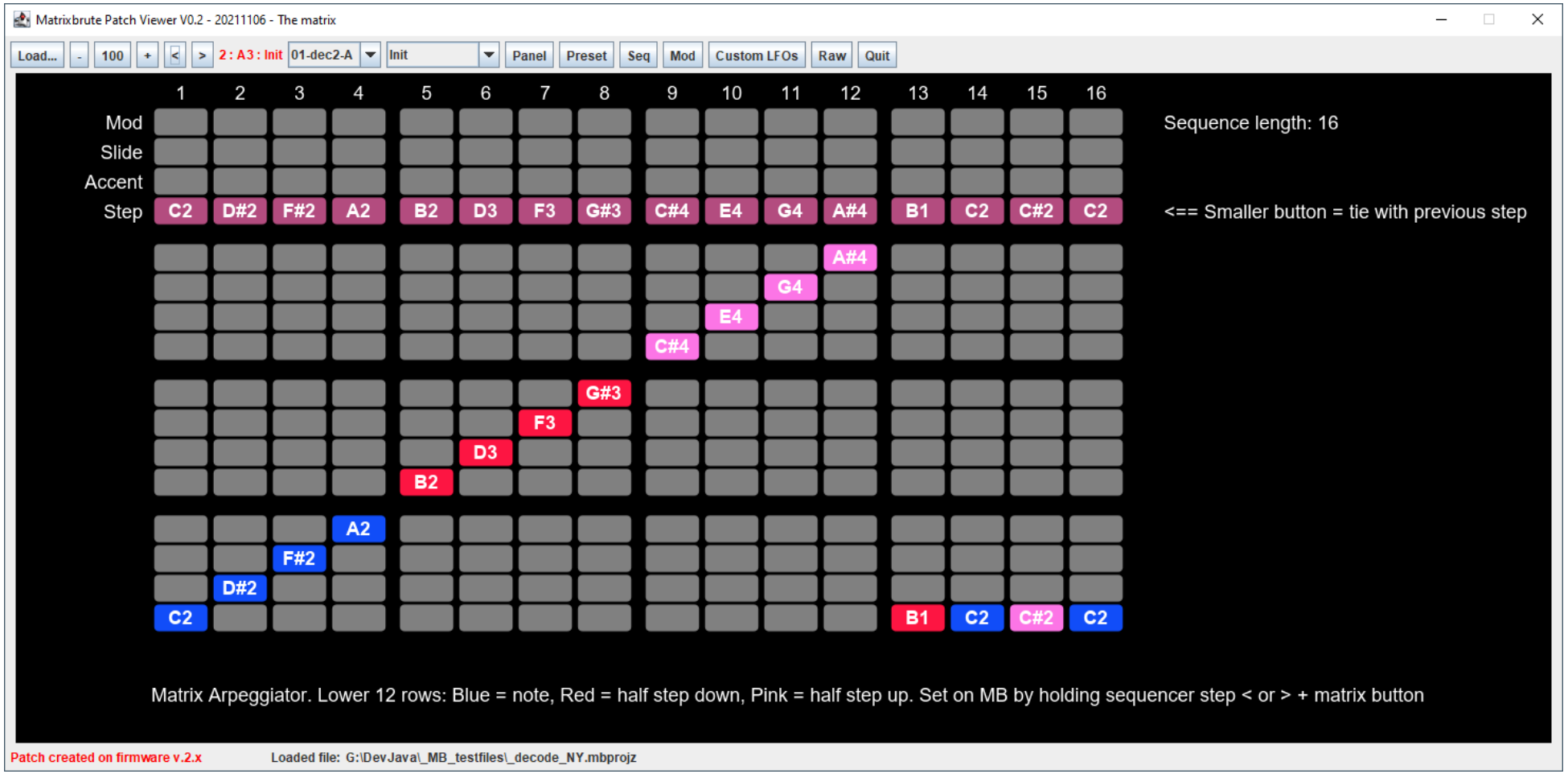
This view shows the sequencer including note values and modulation amounts.



- Mod row: Blue button pointing right = Mod amounted turned right (positive) with the amount given inside (above: A4 is +29)
- Mod row: Red button pointing left = Mod amounted turned left (negative) with the amount given inside (above: A15 is -23)
- Mod row: Grey empty button = not used. If it contains a value, but is grey it is disabled but has a value (that is not used)
- Slide row and Accent row: Red = on, grey = off
- Step row: Red = active, grey = inactive. A lower button is a tie with the previous (like D2 and D3 above). Note value inside.
    - A green line indicate the active sequence length (also written in top right)

The Matrix window – Sequence view ("Seq" button) – Matrix Arpeggiator mode

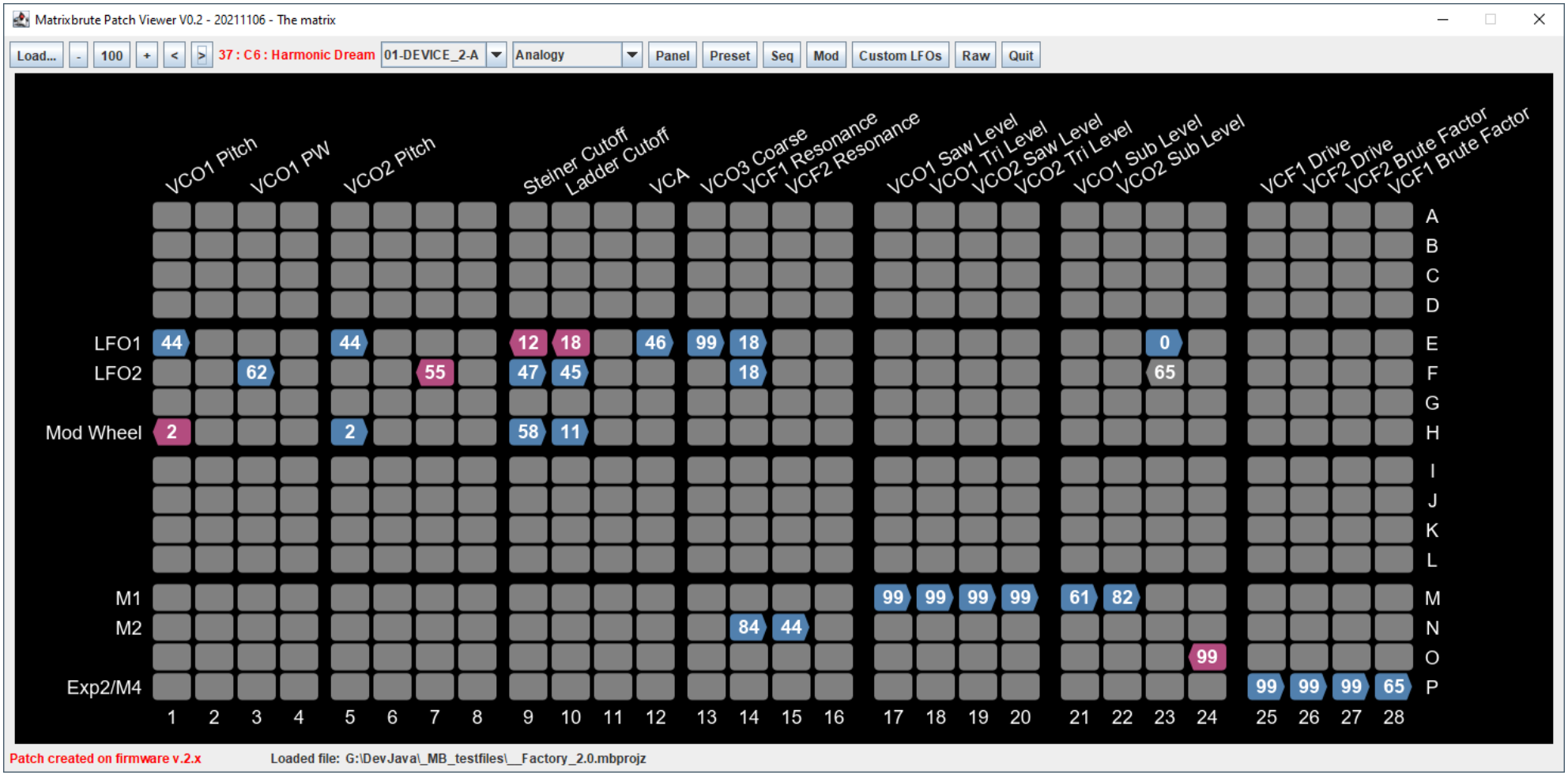
This view will show the matrix arpeggiator when relevant (= for presets with "M" on the right side of the preset name in the Preset view)



See the Matrixbrute manual for more details on the matrix arpeggiator! This view should reflect what's shown on the matrix.

The Matrix window – Modulation view ("Mod" button)

This view shows the mod matrix

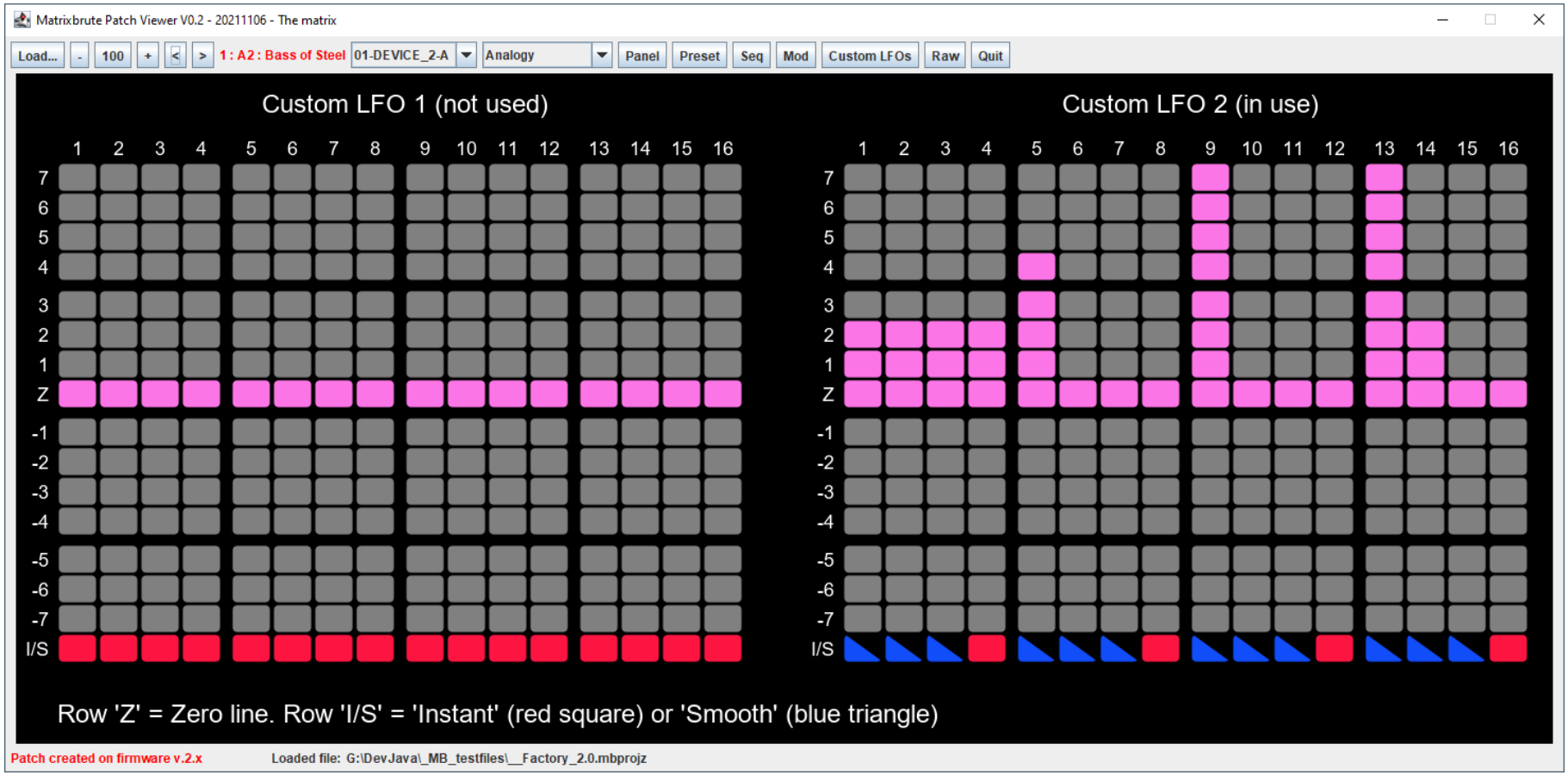


- Blue with right arrow: Value inside is positive modulation
- Red with left arrow: Value inside is negative modulation
- Grey with no value: not used. Grey with value: not active, but a value is set.
- Only labels for the sources and destinations that actually is used is shown
    - The rightmost blue "0" on row E might be a bug, or a destination "not set". Will be investigated…

The Matrix window – Custom LFOs

Shows the custom LFOs as shown on the matrix, also indicate if each of them is currently used.

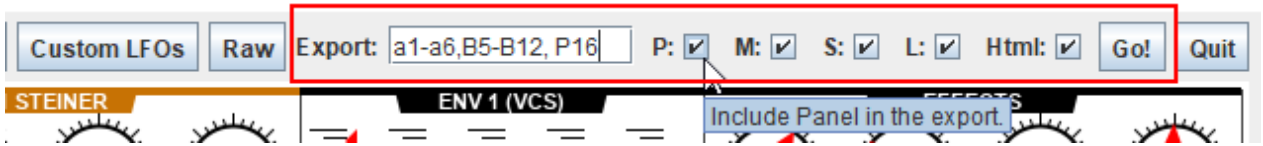
This is only available in firmware 2+ presets. See description in the bottom of the window.



Export to images and HTML

From version 0.4 it is possible to export all screens to images (.png). Html-files can also be created for easy navigation, or printing. See also the chapter **Configuring the application** for how to set the export folder using **–DexportPath** to avoid manually lookoing for it each time you start MBPV.

To export to images and htmls, use the controls in the menu in the Panel or Matrix window. Please hold the pointer above each button/field for tooltips.



In the export field you can enter the patches you want to export as a list or ranges (or a combination). See the pic above for an example. Ranges and single patches can be entered. If left empty, only the currently selected patch is exported. Please ensure patches exist for the range you enter, else the result might be undefined (=a lot of "empty" patched exported)

Then check some/all of the 5 checkboxes if you want the corresponding images exported: **P**=Panel, **M**=Mod matrix, **S**=Sequence, **L**=custom LFOs, and **Html** = generates HTML files for easy navigation/printing etc.

To export click "Go!" and you get a dialog for selecting a folder. All files exported will be created in an automatically created **sub-folder** inside the folder you select. The name will be the same as your opened project, and the current date/time is appended. So if you have opened **c:\MB\pathes.mbprojz** and you select to export to **c:\MB** at the time **13:52:05 on the 11 of March 2022** the exported files will go to **c:\MB\patches.mbprojz_20220311-135205**. This way you can easily sort your export folder and keep the history if you like, and the last one for each mbprojz will be what you have exported last (=the current one).

Inside this folder png's of the different screens will be saved with names as **<slot>-<patch name>.png**, f.ex **A01-Fat_String.png**. The patch name is cleaned for characters not suitable for filenames (in fact , only a-z and A-Z, and 0-9 is kept at the moment). All illegal characters are replaced by _ (underscore).

The size of each image will be decided by the zoom-factor the panel window and the matrix window when you engage export, so you can decide the size you want (to use different size on f.ex Custom LFO's than the other matrix windows, multiple exports must be done and size of matrix window changed in between. The image files will have the same names for a given export (if repeated) so you can export all, then change size of LFO's and export only those, then copy all result files into the first export to replace the original once – relevant if creating HTMLs (if un-checking LFOs in first export, the necessary html is not generated)

If you check "Html", some HTML files will also be created in the folder.

Special files:
00_Presets.png: The preset screen, listing all patches.
00_Index.html: Open this one to start browsing your export. Lists all patches with link to each
00_All.html: All patches listed with all images and navigation links. Can be used for printing to PDF (but page breaks are not necessarily so nice…)

The Raw window – dump & diff

Shows a raw dump of the patch data. Can print or diff 1-255 presets (patch data only at the moment). This is mostly useful for decoding the internal format, or diffing one or more patches to see if they are identical. Note that it is currently 2 different file formats (for patches created with firmware v. 1.x and firmware 2.x+). Some parameters are not available in FW1, and some memory locations are used for one thing in FW1 and something other in FW2 (ex: custom LFO in FW2 is used for matrix info in FW1). Note: Patches created with FW1 and FW2 can both exist in the same project, and might if diffed show up differently even if they are quite similar.

The menu bar items:

Reload: reloads the current file (mostly useful when decoding the internal data format and repeatedly change a value on the Matrixbrute, download to MCC, export to file, load in MBPV)

Clear: Clear the text area

Copy: Copy the text area to the clipboard

Dropdown bank: Select a whole bank to print/diff

Presets: select one or many presets to print/diff. Can be ranges and ","-separated list, f.ex **A1-A3,B5** will process A1,A2,A3,B5 as shown in the picture

Filter: If something is entered here, only the lines containing that text (not case sensitive) will be listed. Here all rows containing **LFO** is shown.

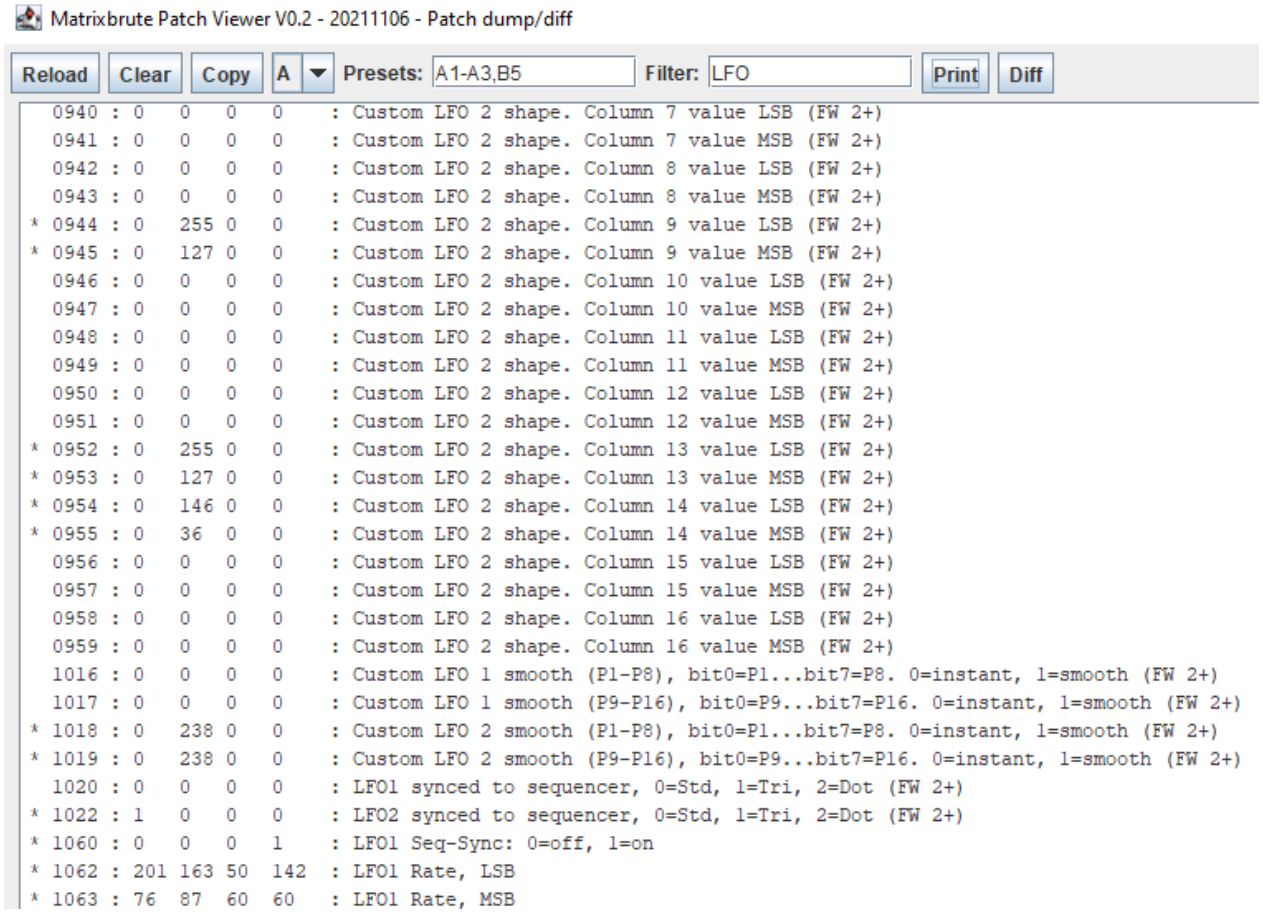
Print: Print all parameters (as shown on the right). An * in front means this value is different in at least one of the presets.

Diff: Show only the lines that are different in all the presets chosen. The fewer lines, the more equal the patches are.

The picture shows a "Print" of patch **A1,A2,A3** and **B5**, filtered on **LFO**.

Patch names and number is shown in the first rows, like:

```
A01 : Analogy
A02 : Bass of Steel
A03 : Vintage Era
A21 : SlowBeat
  Offs : A01 A02 A03 A21  : Parameter
  0896 : 0   0   0   0    : Custom LFO 1 shape. Column 1 value LSB (FW 2+)
  0897 : 0   0   0   0    : Custom LFO 1 shape. Column 1 value MSB (FW 2+)
```



Possible future functionality (suggestions are welcome!)

This contains short description of ideas I have for functionality that could be useful, and ***might*** be implemented in some form sometime in the future.

- Panel window:
  - Fixing missing parameters
  - Indicator on each parameter that is modulated by the matrix (i.e is a destination)
  - an option for more realistic graphics (low prio)
  - compare 2 patches (some indication of parameters being equal, and different)
- Preset window:
  - Details about the selected preset (as shown in MCC – like type and characteristics). Maybe as toggle buttons for filtering patches matching
  - Some indication of which one is selected
  - Some indication of presets that are equal
- Generally:
  - Hotkeys
  - Possibility to open multiple windows of the same type
  - Config window/settings replacing the –D switches
  - Rewrite the internal data structure, for better handling of FW1 and FW2 patches. Will also fix some other "less elegant" current solutions
  - *Export to patch sheet (textual sum-up of a preset)* – ***done in v.05 as a first simple solution***
  - Decode sysex, and get data directly from the Matrixbrute (or be able to paste in sysex dump from MCC)
  - A full description of the data format decoded so far
  - Some installer, to avoid manually installing Java