ZOIA Patch File binary Format Technical Document

The ZOIA is a musical instrument effect pedal created by Empress Effects Inc. The ZOIA is a very complex device in which audio effects are created by adding virtual modules and connecting them together to manipulate the sounds. Such configurations are called patches. Furthermore, the ZOIA has a slot for a micro-SD flash card where patch files can be stored and transferred to and from the ZOIA's active list of patches.

This document is a walkthrough of what is known and speculated about the ZOIA patch files. The goal is to help people who would like to create software that could read and display the content of patch files.

This document is part of the [meanmedianmoge/zoia\_lib] GitHub repository and has been created and maintained by the contributors with the help of Steeve Bragg of Empress Effects Inc.

ZOIA and its patch binary format are a trademark of Empress Effects, Inc. and are used with permission.

Neither the developers nor Empress Effects, Inc. are liable for any issues caused or raised by the use or modification of this document.

# Applicability

Note that there is no guarantee that the information from this document is exact. We tried to verify its exactitude as much as possible but there is no guarantee.

The information herein has been verified against version 2.50 of ZOIA's firmware.

See the appendix for the revision history of this document.

Also pay attention to the limitation mentioned in the Zoia Specifics section below concerning module types configuration.

# General Definitions and Conventions

* Numeric values without any special notation are in base 10 (as opposed to hexadecimal).
* Hexadecimal values are prefixed with 0x, ex.: 0x3e or 0x2.
* When the text refers to a particular section or field in the patch file, the corresponding name is enclosed in square brackets ([]) for clarity.
* The «**n/a**» mention means «not available» or «unknown» or «irrelevant» depending on the situation.
* The term «**index**» is a numerical value that identifies an item in a list of objects. The first item is zero, the second is one, etc.

# Zoia Specifics

## About Modules

To be able to decode a patch file, it's important to understand some things about modules and set some terminology about them.

Module **Types** vs Module **Instances** – with the Zoia, we create patches by creating instances of module types on pages and making connections between them. For example, when inserting a «Reverb Lite» effect module on a page, we created an instance of the module type «Reverb Lite». The instance lives in the page and is what we find inside the patch files that this document talks about. Module types, on the other hand, are like object classes from which instances are created.

Modules types and instances alike, are composed of what we call «**blocks**» with which we interact when creating patches. Blocks correspond to buttons on the Zoia pad. Blocks can be seen as inputs and outputs for the module. Blocks can be used to receive or send audio data while others can be used to input or output CV values.

Most modules types have different **options** that can be used to define some specific behavior of a module instance. In many cases, selected option's values will dictate which blocks will be available (**visible**) for a specific module instance. For that reason, we have to be careful with the way we interpret a **block index** reference from the patch data because we might be tempted to think it corresponds to the visible blocks of the module instance while it might be something else. If we look at the data described in the [Connections section] for instance, the [Source Block Number] is an index in the list of blocks defined in the Module Type, not those necessarily visible on the module instance.

The patch data also contains a module **version** field. Although not verified at this time, we think that it refers to a specific version of the module type associated with the module instance. As Zoia firmware version changes, some module types behavior also changes and we suspect that this is what this fields tells us.

**LIMITATION**: Of course, the goal of this document is to be able to write software that can read a Zoia patch file and display (or process) it's data in a manner useful for Zoia users. Such a task would require to have a good knowledge of each module type, what the available options are, the list of block names and for each of them, knowing which are inputs and outputs, audio or CV etc. The authors decided that the module types configuration is outside the scope of this document. Only a minimal set of data concerning module types are provided in appendix A.

# Patch File Format

Patches are all 32 KB in size. Unused portions are zeroed out.

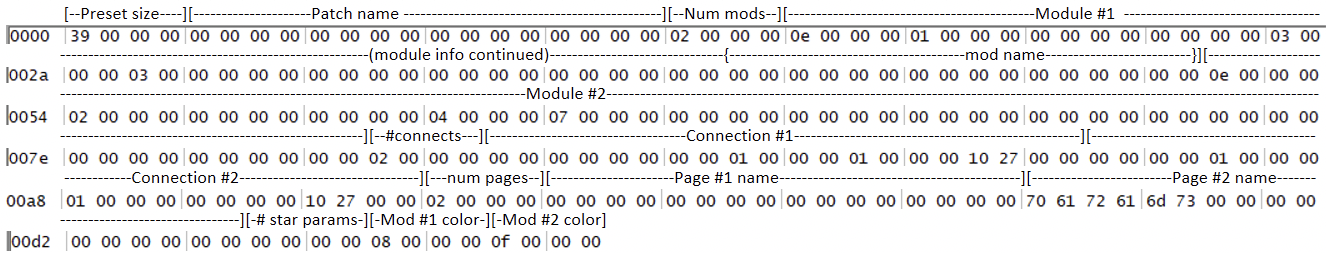
## Some data format conventions:

* Unsigned multiple-bytes numeric values are stored least-significant-byte-first. For example, value 123456 (0x01E240) would be stored as the following sequence of values: 0x40, 0xE2, 0x01, 0x00.
* Texts like patch names, modules names, etc., are usually of fixed size and are stored one byte per character and are coded as ASCII characters. Since the ZOIA interface only allow for the following characters, we don’t expect to find anything else except for values 0x00 when the text is shorter than the maximum allowed size:
* lowercase letters a-z (0x61 – 0x7A)
* Uppercase letters A-Z (0x41 – 0x5A)
* Numbers 0-9 (0x30 – 0x39)
* Special characters: space (0x20), dash (0x2D) and dot (0x2E).
* The «field offset» and «field length» values in the following sections are stated as a number of bytes unless otherwise noted.
* All offset values are zero-relative meaning that the offset of the first byte in a group is zero (0).
* In the tables below, the «field offset» values are relative to the start of the section or the sub section that they describe. Meaning that the first byte in the section is zero even if the section might start at position 1000 in the patch file.

Patch file are composed of sections as described in the following table. Each section is defined in more details in the following sub sections of the document.:

|  |  |  |
| --- | --- | --- |
| **Position** | **Name** | **Description** |
| 1 | Patch Header | This section defines the size of the patch definition and the patch name. |
| 2 | Modules | This section contains data for each module used in the patch. |
| 3 | Connections | This section describes the connections between modules inputs and outputs. |
| 4 | Pages Names | This section contains the names assigned to page in the patch, if any.  NOTE: This section can be empty (contains no page name) if none of pages were given names when the patch was created. This means that we have to look at the [Page Number] field in the modules list to figure out the actual number of pages in the patch. |
| 5 | Starred Parameters | This section defines information on starred parameters inside the patch. |
| 6 | Modules Colors | Since firmware version 1.10 and above, this section defines the (extended) color assigned to each module. The number of colors defined and the order is the same as in the [Modules] section. |

Here is a hexadecimal representation of binary patch file which includes the modules Audio in, Audio out with two connections established between those two modules:



## Patch header section

|  |  |  |  |
| --- | --- | --- | --- |
| Field Offset | Field Length | Field Name | Description |
| 0 | 4 | Patch Size (or preset size) | Unsigned long integer – the size of the patch file as a number of 4 bytes «chunks», including of itself. Multiply this value by 4 to get the total number of bytes. |
| 4 | 16 | Patch Name | Text – the patch name. |

## Modules section

The modules section contains data for each module used in the patch. Because each module takes up a variable number of bytes in the patch file, it's easier to describe this section using two tables. The following table defines the general format for the whole [Modules] section while the [Module definition] table defines the fields for one module in the list.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Offset | Field Length | Field Name | Description |
| 0 | 4 | Number of modules | Unsigned long integer – the number of [Module definition] that follows. |
| 4 | variable | List of [Module definition] | A list of [Module definition]. The number of modules is given by the previous field. The format of an individual module in this list is described in the table below. |

### Module definition

The following table describes the format of each module in the list of modules. The field offset is relative to the start of the module in the list.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Offset | Field Length | Field Name | Description |
| 0 | 4 | Module Size | Unsigned long integer – the size of the module definition as a number of 4 bytes «chunks», including of itself. Multiply this value by 4 to get the total number of bytes. |
| 4 | 4 | Module Type ID | Unsigned long integer – the module type ID. See appendix A for more details. |
| 8 | 4 | unknown | Unsigned long integer – seems to be always zero. ?? Figure this out |
| 12 | 4 | Page number | Unsigned long integer – the page number where the module is located. Starts at zero for the first page. |
| 16 | 4 | Old Color | Unsigned long integer – the old color number that would be used to display this module on a Zoia with pre 1.10 firmware. Colors are defined in appendix B.  For patches created with firmware 1.10 and beyond, the color used to display the module is defined in the [Modules Colors] section. In this case, the value in this field is the closest match old color. Details appear in appendix B. |
| 20 | 4 | Grid Position | Unsigned long integer – The position of the first (leftmost) module cell (led) on the page. It's a number from 0 to 39. From left to right, the first row are numbers 0-7, second row are 8-15, etc. |
| 24 | 4 | Number of User Parameters | Unsigned long integer – ?? What exactly we mean by «user» ?? Most of the times (but not always) this is the number of values in the [Additional Options] field. |
| 28 | 4 | Module Version | Unsigned long integer – Probably more appropriately called a «module type version». We suspect that with different firmware, some module behavior changes and some blocks and or options can be added or changed. This probably helps identify which module type behavior this was created under. |
| 32-39 | 1 | Module Option 0 to Option 7 | Unsigned byte integer – Values for options 0 to 7 selected when creating or editing the module. These are all single-byte values so they can range from 0 to 255. Here, there is room for 8 option's value but the number of values that are really used depends on the module type and version. Used options are always at the beginning of this list. |
| 40 | ([Module size]\*4) - 56 | Additional 0ptions | List of unsigned long integers – After the [Module Option] field there could be any number of additional values stored here. Note that this list can also be empty.  This seems to depend on the module and what extra values the module needs to store. Additional R&D is required to interpret this completely. So far, they seem to be the value assigned to the module inputs, like the gain for a «Audio Output» module if the gain option is selected. |
| End - 16 | 16 | Module Name | Text – the module's name.  This will be empty if the user did not change the default module name. |

## Connections section

This section provides information on each connection between one module input and output.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Offset | Field Length | Field Name | Description |
| 0 | 4 | Number of connections | Unsigned long integer – the number of [Connection definition] that follows. |
| 4 | variable | List of [Connection definition] | A list of [Module definition]. The number of modules is given by the previous field. The format of an individual module in this list is described in the table below. |

### Connection definition

The following table describes the format of each connection. The field offset is relative to the start of the connection definition in the list.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Offset | Field Length | Field Name | Description |
| 0 | 4 | Source Module Number | Unsigned long integer – the module number at the origin (source) of the connection. It's an index in the list of modules defined in the [Modules] section. |
| 4 | 4 | Source Block Number | Unsigned long integer – the block index from which the connection starts in the source module. It's an output block.  See note 1 below. |
| 8 | 4 | Destination Module Number | Unsigned long integer – the module number where the connection ends. It's an index in the list of modules defined in the [Modules] section. |
| 12 | 4 | Destination Block Number | Unsigned long integer – the block index to which the connection ends in the destination module. It's an input block.  See note 1 below. |
| 16 | 4 | Connection Strength | Unsigned long integer. The values are in the range 0 to 10000 (base 10) or 0x0 to 0x0270.  On the Zoia, connection strength can be set either as a dB value or a % value. In dB, the range is –100dB to +12.00dB. In %, the range is 0.001% to 398.1%.  The formula to convert the connection strength value to a dB value is: 0 - ((10000 - «connection strength») / 100).  The formula to convert the dB value to a % is: 100 \* (10 ^ («dB value» / 20)). |

Note 1 – about [Source Block Number] and [Destination Block Number]: These values are indexes (0 relative) of blocks in the origin and destination modules referenced by the connections. It is **important** to note that it does NOT refer to the module instance's visible blocks but instead, it refers to the block index in the list of all the blocks provided by the module type, like if all the possible module's blocks were visible. Even though some blocks might be hidden because of the options selected for a specific module.

## Pages Names section

This section contains the names assigned to page in the patch, if any.

NOTE: This section may not contain any page name (the [Number of pages] may be zero) if none of the pages were given a name. This means that we have to look at the [Page Number] field in the modules list to figure out the actual number of pages used in the patch.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Offset | Field Length | Field Name | Description |
| 0 | 4 | Number of page names | Unsigned long integer – the number of page names that follows.  Note that the value might be zero. |
| 4+(n\*16) | 16 | Page Name | Text – for each page, the page name. |

## Starred parameters section

|  |  |  |  |
| --- | --- | --- | --- |
| Field Offset | Field Length | Field Name | Description |
| 0 | 4 | Number of starred parameters | Unsigned long integer – the number of [Starred parameter definition] that follows.  Note that the value might be zero. |
| 4 | variable | List of [Starred Parameter Definition] | A list of [Starred parameter definition]. The format is described in the table below. |

### Starred parameter definition

The following table describes the format for the definition of one starred parameter in the [Starred parameters] section, in the [List of Starred Parameter Definition] field.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Offset | Field Length | Field Name | Description |
| 0 | 4 | Value | Unsigned long integer. This number combines 3 information in a 32 bits value as follows. Note bit 0 is the least significant bit.  Bits 0-15 => The module index number.  Bits 16-22 => The input block number in the module (see note 1).  Bits 23-30 => The MIDI CC value (see note 2). |

Note 1: the block number follow the same numbering scheme as in the [Connection] section. Refer to that section for more details.

Note 2: the MIDI CC value in bits 23-30 is zero if the starred parameter has no MIDI CC value defined. If the value in bits 23-30 is not zero, the MIDI CC number is obtained by subtracting 1 from that value. In other words, MIDI CC #0 appears as value 1, MIDI CC #1 appears as value 2, etc.

## Modules colors section

The format for this section is different from the previous ones because there is no field telling us the number of colors present in the section. Instead, it is assumed that the number of colors is the same as the number of modules in the [Modules] section.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Offset | Field Length | Field Name | Description |
| 0 | 4 | Color 0 | Unsigned long integer – (extended) color for module 0. See appendix B for more details. |
| 4 | 4 | Color 1 | Unsigned long integer – color for module 1. |
| 8 | 4 | Color 2 | Unsigned long integer – color for module 2. |
| … |  |  |  |
| N\*4 | 4 | Color N | Unsigned long integer – color for module N. |

Appendix A – Module types

The following table shows the list of [Module type] with their numeric value and their corresponding name and category.

Analysis of a patch definition requires the knowledge of the number and the nature of each block for a particular module instance. This can vary depending on which options are set for a module instance. This appendix will have to be expanded (greatly) in order to provide such information.

|  |  |  |
| --- | --- | --- |
| Type ID | Module Type | Module Category |
| 0 | SV Filter | Audio |
| 1 | Audio Input | Interface |
| 2 | Audio Output | Interface |
| 3 | Aliaser | Audio |
| 4 | Sequencer | Control |
| 5 | LFO | Control |
| 6 | ADSR | Control |
| 7 | VCA | Audio |
| 8 | Audio Multiply | Audio |
| 9 | Bit Crusher | Audio |
| 10 | Sample & Hold | Control |
| 11 | OD & Distortion | Effect |
| 12 | Env Follower | Analysis |
| 13 | Delay Line | Audio |
| 14 | Oscillator | Audio |
| 15 | Pushbutton | Interface |
| 16 | Keyboard | Interface |
| 17 | CV Invert | Control |
| 18 | Steps | Control |
| 19 | Slew Limiter | Control |
| 20 | MIDI Notes in | Interface |
| 21 | MIDI CC in | Interface |
| 22 | Multiplier | Control |
| 23 | Compressor | Effect |
| 24 | Multi-Filter | Audio |
| 25 | Plate Reverb | Effect |
| 26 | Buffer Delay | Audio |
| 27 | All-Pass Filter | Audio |
| 28 | Quantizer | Control |
| 29 | Phaser | Effect |
| 30 | Looper | Audio |
| 31 | In Switch | Control |
| 32 | Out Switch | Control |
| 33 | Audio In Switch | Audio |
| 34 | Audio Out Switch | Audio |
| 35 | Midi Pressure | Interface |
| 36 | Onset Detector | Analysis |
| 37 | Rhythm | Control |
| 38 | Noise | Audio |
| 39 | Random | Control |
| 40 | Gate | Effect |
| 41 | Tremolo | Effect |
| 42 | Tone Control | Effect |
| 43 | Delay w/Mod | Effect |
| 44 | Stompswitch | Interface |
| 45 | Value | Control |
| 46 | CV Delay | Control |
| 47 | CV Loop | Control |
| 48 | CV Filter | Control |
| 49 | Clock Divider | Control |
| 50 | Comparator | Control |
| 51 | CV Rectify | Control |
| 52 | Trigger | Control |
| 53 | Stereo Spread | Audio |
| 54 | Cport Exp/CV in | Interface |
| 55 | Cport CV out | Interface |
| 56 | UI Button | Interface |
| 57 | Audio Panner | Audio |
| 58 | Pitch Detector | Analysis |
| 59 | Pitch Shifter | Audio |
| 60 | Midi Note Out | Interface |
| 61 | Midi CC Out | Interface |
| 62 | Midi PC Out | Interface |
| 63 | Bit Modulator | Audio |
| 64 | Audio Balance | Audio |
| 65 | Inverter | Audio |
| 66 | Fuzz | Effect |
| 67 | Ghostverb | Effect |
| 68 | Cabinet Sim | Effect |
| 69 | Flanger | Effect |
| 70 | Chorus | Effect |
| 71 | Vibrato | Effect |
| 72 | Env Filter | Effect |
| 73 | Ring Modulator | Effect |
| 74 | Hall Reverb | Effect |
| 75 | Ping Pong Delay | Effect |
| 76 | Audio Mixer | Audio |
| 77 | CV Flip Flop | Control |
| 78 | Diffuser | Audio |
| 79 | Reverb Lite | Effect |
| 80 | Room Reverb | Effect |
| 81 | Pixel | Interface |
| 82 | Midi Clock In | Interface |
| 83 | Granular | Audio |
| 84 | Midi Clock Out | Interface |
| 85 | Tap to CV | Control |
| 86 | MIDI Pitch Bend In | Interface |
| 103 | Device Control | Interface |
| 104 | CV Mixer | Control |

Appendix B – Colors

The following table shows the numeric values for the color parameters used in ZOIA patches.

Since firmware 1.10, a new (extended) set of colors were made available (see the [Modules colors] section).

The table shows both sets of colors since they share the same numeric values. Old colors have numeric values from 0 to 7.

Note that with firmware 1.10 and beyond, when a extended color is assigned to a patch module, the ZOIA also sets the [Old Color] field for the module to the corresponding old color as described in the following table.

|  |  |  |
| --- | --- | --- |
| Color Number | Color | Corresponding Old Color |
| 0 | ? Unknown Meaning ? |  |
| 1 | Blue (\*1) | [1] Blue |
| 2 | Green (\*1) | [2] Green |
| 3 | Red (\*1) | [3] Red |
| 4 | Yellow (\*1) | [4] Yellow |
| 5 | Aqua (\*1) | [5] Aqua |
| 6 | Magenta (\*1) | [6] Magenta |
| 7 | White (\*1) | [7] White |
| 8 | Orange (\*2) | [3] Red |
| 9 | Lime (\*2) | [2] Green |
| 10 | Surf (\*2) | [5] Aqua |
| 11 | Sky (\*2) | [1] Blue |
| 12 | Purple (\*2) | [6] Magenta |
| 13 | Pink (\*2) | [3] Red |
| 14 | Peach (\*2) | [3] Red |
| 15 | Mango (\*2) | [4] Yellow |

\*1: this color belongs to the old color set and to the extended color set.

\*2: this color belongs to the extended color set only.

# Appendix H – Revision History

2021-12 – (marcuslupinus on github) Reformatted the document, corrected some typos, added some context at the beginning and pursued reverse engineering of the patch file format using v 2.50 of Zoia Firmware. This version includes reverse engineering of these topics: relation between extended and old color, module block numbers, starred parameters interpretation, connections strength interpretation, module options interpretation, module version interpretation.