

# ZOIA Patch Binary Format Technical Document

This is a walkthrough of what is known and speculated about the ZOIA patch binary files that ZOIA saves and transfers when storing patches to SD card.

## Overview

Patches are all 32 KB in size. The memory which is not needed to store data inside the patch file has zeroes written to it.

The following are known sections within the binary format:

1. Patch information
  - a. Preset size
  - b. Patch name
2. Module information
  - a. Number of modules
  - b. List of modules
3. Connection information
  - a. Number of connections
  - b. List of connections
4. Page information
  - a. Number of pages
  - b. List of page names
5. Starred parameters
  - a. Number of starred parameters
  - b. List of starred parameters
6. Module colors
  - a. List of module colors

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:

```

[--Preset size--][-----Patch name-----][--Num mods--][-----Module #1 -----]
|0000 |39 00 00 00|00 00 00 00|00 00 00 00|00 00 00 00|02 00 00 00|0e 00 00 00|01 00 00 00|00 00 00 00|03 00
-----
|002a |00 00 03 00|00 00 00 00|00 00 00 00|00 00 00 00|00 00 00 00|00 00 00 00|00 00 00 00|00 00 00 00|0e 00 00 00
-----
|0054 |02 00 00 00|00 00 00 00|00 00 00 00|04 00 00 00|07 00 00 00|00 00 00 00|00 00 00 00|00 00 00 00|00 00 00 00
-----
|007e |00 00 00 00|00 00 00 00|00 00 02 00|00 00 00 00|00 00 00 00|00 00 01 00|00 00 01 00|00 00 10 27|00 00 00 00|00 00 01 00|00 00
-----
|00a8 |01 00 00 00|00 00 00 00|10 27 00 00|02 00 00 00|00 00 00 00|00 00 00 00|00 00 00 00|00 00 00 00|70 61 72 61|6d 73 00 00|00 00
-----
|00d2 |00 00 00 00|00 00 00 00|00 00 08 00|00 00 0f 00|00 00 00 00|00 00 00 00|00 00 00 00|00 00 00 00|00 00 00 00|00 00 00 00|00 00
-----

```

## Patch information

### Preset size

The first 4 bytes of the file (long integer) indicates the size of the preset. This size is in 4-byte (long) chunks. It is inclusive of this section. To determine the overall number of bytes used by the preset, multiply this value by 4.

In the above example

Preset size = 0x39 = 57

$57 * 4 = 228$

And 228 is the number of bytes being used by the patch information in the above example.

### Patch name

The next 16 bytes store the patch name as a string as 8-bit characters. It may be UTF8 encoding, or at least the strings seem to translate successfully using that encoding method.

## Module information

### Number of modules

The next 4 bytes is a long integer which indicates how many modules are to follow.

### List of modules

A module consists of the following sections:

Position	Start	Length	Name
0	0	4	Module size
1	4	4	Module type
2	8	4	<unknown>
3	12	4	Page number
4	16	4	Old color (7 values)
5	20	4	Grid position
5	24	4	Number of user parameters
6	28	4	<unknown>
7	32	1	Option 1 value
8	33	1	Option 2 value
9	34	1	Option 3 value
10	35	1	Option 4 value
11	36	1	Option 5 value
12	37	1	Option 6 value
13	38	1	Option 7 value

<b>14</b>	39	1	Option 8 value
<b>16+n</b>	$40+n*4$	4	Any number of additional values
<b>last</b>		16	Module name

**Module size** indicates the total number of 4-byte chunks that comprise this module. Modules vary in length, so this value can be used to know how many bytes need to be read. The value of Module size is inclusive of itself.

**Module type** is a long integer which represents the module. The list of modules and their types can be found in Appendix A.

**Page number** indicates the page on which the module is placed.

**Old color** is the 7-value color for pre-1.10 firmware.

**Grid position** is where on the grid the module starts within the page.

**Number of user parameters** seems to be the number of parameter inputs a user can access on the grid. This value is exclusive of inputs and outputs - only values a user can adjust.

**Option 1-8 value** are the options that are selectable within a Module – when creating or editing a module. These are all single-byte values so could store values 0 – 7.

**Additional values** - After Option 8 there could be any number of additional values stored. 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.

**Module name** - The last 16 bytes of a module are the Module name if the user has given the module a non-default name. It is seemingly in UTF8 encoding.

## Connection information

### Number of connections

These 4 bytes are a long integer which represents the number of Connection sections which appear after this value.

### List of connections

A Connection section is comprised of 5 long integer blocks.

Position	Start	Length	Name
0	0	4	Source module number
1	4	4	Output number of source
2	8	4	Destination module
3	12	4	Input number of destination
4	16	4	Connection strength

## Page information

### Number of pages

The next long int (4 bytes) represents the number of page names that follow. This value may be zero if the user had not named any pages to non-default names.

### List of page names

Each page name is 16 bytes in UTF8 encoding.

## Starred parameters

### Number of starred parameters

The next long int (4 bytes) represents the number of starred parameters.

### List of starred parameters

Each starred parameter is a 4-byte value, but currently it is unclear how to interpret them.

## Module colors

It is expected that there will be a number of Module color sections equal to the number of modules from the **Module information** section.

### List of module colors

A module color is 4 bytes long and maps to a 16-value color. Color values can be found in Appendix B.

## Appendix A – Module types

Type ID	Description
0	SV Filter
1	Audio Input
2	Audio Out
3	Aliaser
4	Sequencer
5	LFO
6	ADSR
7	VCA
8	Audio Multiply
9	Bit Crusher
10	Sample and Hold
11	OD & Distortion
12	Env Follower
13	Delay line
14	Oscillator
15	Pushbutton
16	Keyboard
17	CV Invert
18	Steps
19	Slew Limiter
20	MIDI Notes in
21	MIDI CC in
22	Multiplier
23	Compressor
24	Multi-filter
25	Plate Reverb
26	Buffer delay
27	All-pass filter

<b>28</b>	Quantizer
<b>29</b>	Phaser
<b>30</b>	Looper
<b>31</b>	In Switch
<b>32</b>	Out Switch
<b>33</b>	Audio In Switch
<b>34</b>	Audio Out Switch
<b>35</b>	Midi pressure
<b>36</b>	Onset Detector
<b>37</b>	Rhythm
<b>38</b>	Noise
<b>39</b>	Random
<b>40</b>	Gate
<b>41</b>	Tremolo
<b>42</b>	Tone Control
<b>43</b>	Delay w/Mod
<b>44</b>	Stompswitch
<b>45</b>	Value
<b>46</b>	CV Delay
<b>47</b>	CV Loop
<b>48</b>	CV Filter
<b>49</b>	Clock Divider
<b>50</b>	Comparator
<b>51</b>	CV Rectify
<b>52</b>	Trigger
<b>53</b>	Stereo Spread
<b>54</b>	Cport Exp/CV in
<b>55</b>	Cport CV out
<b>56</b>	UI Button
<b>57</b>	Audio Panner

<b>58</b>	Pitch Detector
<b>59</b>	Pitch Shifter
<b>60</b>	Midi Note out
<b>61</b>	Midi CC out
<b>62</b>	Midi PC out
<b>63</b>	Bit Modulator
<b>64</b>	Audio Balance
<b>65</b>	Inverter
<b>66</b>	Fuzz
<b>67</b>	Ghostverb
<b>68</b>	Cabinet Sim
<b>69</b>	Flanger
<b>70</b>	Chorus
<b>71</b>	Vibrato
<b>72</b>	Env Filter
<b>73</b>	Ring Modulator
<b>74</b>	Hall Reverb
<b>75</b>	Ping Pong Delay
<b>76</b>	Audio Mixer
<b>77</b>	CV Flip Flop
<b>78</b>	Diffuser
<b>79</b>	Reverb Lite
<b>80</b>	Room Reverb
<b>81</b>	Pixel
<b>82</b>	Midi Clock In
<b>83</b>	Granular

## Appendix B – Colors

The “old” 8-value colors are values 0-7.

0: ?

1: Blue

2: Green

3: Red

4: Yellow

5: Aqua

6: Magenta

7: White

8: Orange

9: Lime

10: Surf

11: Sky

12: Purple

13: Pink

14: Peach

15: Mango