shattersample

user manual

introduction

shattersample is a monophonic texture-based synthesizer / sampler / glitch machine. it produces sound using a collection of samples stored on a microSD card, chopping them into small pieces of audio (or *grains*) and rearranging them to create a new texture. the start point, length, density, and randomization of these grains can be flexibly configured to create a wide range of timbres from the same sample of audio. it also features 3 playback modes (granular, sampler, and arpeggiator).

navigating the user interface

playing notes

notes are played via the keyboard. letters Q – B correspond to a two-octave range on the chromatic scale, with Q and D being the note C. note values increase from left to right and from top to bottom, like reading a book. the up and down arrows are used to change octave in a 3-octave range.

controlling parameters

all parameters are interfaced using the buttons, switches, LEDs, and hex display on the FPGA. the hex display shows the name of the parameter or parameters to modify, and the LEDs above the switches show their current value. to change a parameter, set the switches to the desired value and press the leftmost button on the FPGA. the LEDs will update to reflect the new value of the parameter. to navigate between parameters, press the inner left and right buttons BTN2 and BTN1.

the hex display will show either one or two parameters at a time. parameters which come in pairs are 8-bit values (except ARP, but more on that later) represented in binary. the parameter on the left hex display corresponds to LEDs / switches 15 – 8, while the parameter on the right corresponds to LEDs / switches 7 – 0. solo parameters are reserved for those with a smaller range of values and range from 0 to 16. for these parameters, only one switch is needed to set its value. the highest-value switch selected indicates the parameter's value, and it, along with all LEDs to its right, light up to show this selection. in this way, the switches act as a quasi-slider.

using the controls

pos / spr

pos and spr correspond to the *position* and *spray* parameters. position assigns the starting position of grains and can range from the beginning of the clip to close to the end. spray introduces randomness into the starting position. a low spray value results in grains beginning from around the same spot every time, while a higher spray value results in greater deviation in the grains' starting positions.

dens / rand

dens and rand correspond to the *density* setting and its randomization parameter. grains are automatically triggered as long as a note is held down, and the density parameter controls how often grains are played. a low density value results in grains being triggered more infrequently, while a high value results in a higher frequency. rand controls the randomization of the density parameter in the same way that spray does. at a low value, grains are triggered at a constant rate, while at a high value grains may play back slowly and then very quickly.

len / rand

len and rand correspond to the *length* setting and its randomization parameter. a low length value results in short grains, while higher values result in longer grains. the rand parameter works the same way as in the previous settings.

ARP / flip

the ARP and flip parameters correspond to the *arpeggiator mode on/off* setting and the *reverse ratio* and are the only unrelated pair of parameters. because the arpeggiator mode controller can only be on or off, if any of the switches corresponding to the arpeggiator setting are active, arpeggiator mode will be turned on, indicated by all of its LEDs being lit. in arpeggiator mode, up to four keys can be held down at once, and the synth will cycle between these notes, arpeggiating them. honestly, this mode is kind of buggy. it doesn't work correctly *unless the density parameter has been partially randomized*, so if you want to use the arpeggiator make sure you do that.

the reverse ratio parameter controls the percent chance that grains will be played in reverse. at zero, grains will only be played forwards, at its maximum value grains will only be played in reverse, and at 50% grains will play forwards and backwards equally.

sound

this parameter selects the sample to play back. there are seventeen samples available, one for each switch and then one for when all are set to zero. to match the indexing convention on the FPGA's switches, sounds are indexed from -1 to 15.

loudness

this parameter controls the output gain of the instrument. on speakers with their own volume control, this setting is redundant, but for headphones or other speakers without a volume control this setting is crucial to not have your eardrums ruptured.

sampler mode

the only button which hasn't been mentioned yet is the rightmost button BTN0. this button toggles the *sampler mode on/off* setting. if you'd prefer a more standard sampler experience, turn on sampler mode to bypass the granular engine and simply play the sample back from start to finish. when sampler mode is active, the hex display will read "SAMPLE" and the only accessible parameter will be the sound select. returning to granular mode will restore all settings to their default value.

loading sounds

loading custom sounds onto the microSD card is a straightforward albeit somewhat tedious process and requires the free programs HxD and Audacity. audio is stored on the SD card in 10-second-long clips of 8-bit unsigned stereo PCM data with a 44.1kHz sample rate. to format your desired samples, load them into Audacity and export the project. make sure to export each track separately and specify the audio format to be raw headerless 8-bit stereo PCM at 44.1kHz.

the starting sectors in memory of sounds are selected as a function of the sound index (from 0 to 16) where sector = 2800 x sound# + 350. to store data directly into specific sectors of memory, the program HxD is required to bypass the SD card's file system. first, open your audio data exported from Audacity using Notepad for Windows or TextEdit for macOS. it will appear as a massive string of ASCII characters. copy this string and then reopen HxD. select a block at least 3800 sectors away from your target address and then go to edit -> select block and then choose the starting address to be your target address. you can now paste your audio data into the memory block and it will be able to be accessed on the FPGA. for convenience, the starting sector addresses for each sound index is given below.

sound #0: sector 350 / x2BC00	sound #1: sector 3150 / x189C00	sound #2: sector 5950 / x2E7C00
sound #3: sector 8750 / x445C00	sound #4: sector 11550 / x5A3C00	sound #5: sector 14350 / x701C00
sound #6: sector 17150 / x85FC00	sound #7: sector 19950 / x9BDC00	sound #8: sector 22750 / xB1BC00
sound #9: sector 25550 / xC79C00	sound #10: sector 28350 / xDD7C00	sound #11: sector 31150 / xF35C00
sound #12: sector 33950 / x1093C00	sound #13: sector 36750 / x11F1C00	sound #14: sector 39550 / x134FC00
sound #15: sector 42350 / x14ADC00	sound #16: sector 45150 / x160BC00	

presets

ecco – *call me Chuck cuz I'm that Person. try hovering around D, O, Y, W.*

pos	spr	dens	rand	len	rand	ARP	flip
01011010	00110000	10100000	00110000	11111111	00000000	OFF	00000000

sound: 12

earthbound – *a recreation of the synth from the "Cave of the Past" theme from the game Earthbound. to play the original melody, play "YTRE, JKLZ, ETQ".*

pos	spr	dens	rand	len	rand	ARP	flip
00001100	00001111	01111111	00010000	11111111	00000000	OFF	00000000

sound: 6

flashback – *a dark and gritty analog bass i pulled from a movie soundtrack. fun to play along with bass-heavy songs.*

pos	spr	dens	rand	len	rand	ARP	flip
10011100	00001000	01111000	00000000	11111111	00000000	OFF	00000000

sound: 1

mirror forest – *chimes extending to infinity. try playing "S I T Q", or drop the octave all the way down and listen to the audio dissolve.*

pos	spr	dens	rand	len	rand	ARP	flip
01000000	01111111	01111000	00000000	11111111	00000000	OFF	00000000

sound: 3

preyouandi – *abstract, fluttering percussion.*

pos	spr	dens	rand	len	rand	ARP	flip
01000000	11111111	10111000	00010111	11111111	00000000	OFF	01000000

sound: 0

receiver / transmitter – *your call has been forwarded to an automated voice message system. the number you have dialed is not available. at the tone, please record your message. if you are finished recording you may hang up, or press one for more options.*

pos	spr	dens	rand	len	rand	ARP	flip
01110010	11111111	11101000	00110000	01000111	11111000	ON	01000000

sound: 4

replica – *a patch inspired by the album of the same name by Oneohtrix Point Never. try looping slowly over "DIDIAIA"*

pos	spr	dens	rand	len	rand	ARP	flip
01001000	00000000	10110000	00000000	11111111	00000000	OFF	00000000

sound: 2

spooky jeans – *turns a groovy R&B rhythm into a spooky sound.*

pos	spr	dens	rand	len	rand	ARP	flip
10011000	00000000	11100000	00001011	11111111	00000000	OFF	00000000

sound: 7