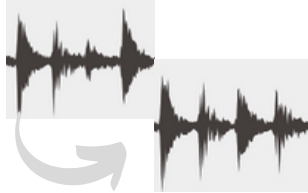


the low down

the knobs can adjust probabilities for **jumping**, **retriggers**, **stuttering**, and **stretching**.

jumping is where sound skips between slices to create new segments from the original audio.



stuttering/retriggering
creates repeats of the
same sample. for
stuttering, there is an
added volume ramp.



stretching will randomly slow down the playback.

bytebeat (alt. firmware)

an alternative firmware is available at

nyblcore.com/bytebeat

that turns nyblcore into a synthesizer with different instructions:

selector fully ccw: **knob a** will change volume + distortion. **knob b** will select a bytebeat formula from ~40 different formulas.

selector knob fully clockwise: **knob a** will select a parameter in the current bytebeat formula. **knob b** changes that parameter in the current bytebeat formula. changes are reset upon power-cycling.

nyblcore.com

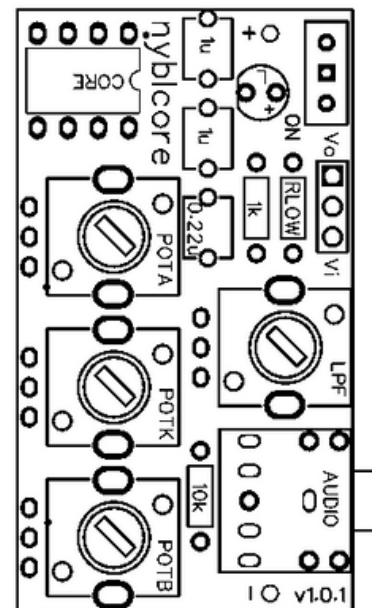
open-source

all code, schematics, and
gerber files are open-source
and available at my github at
github.com/schollz/nyblcore.



help

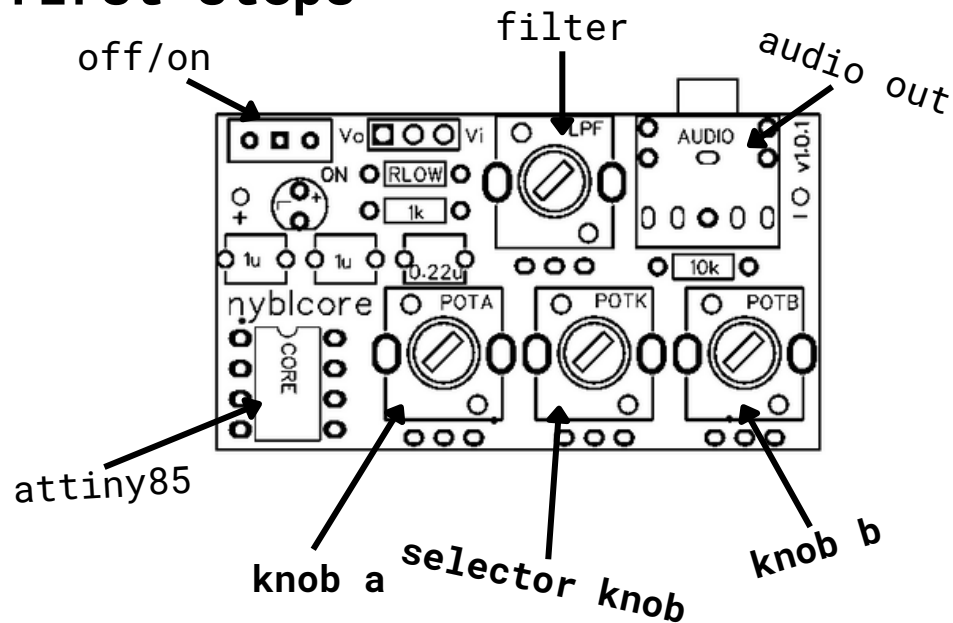
send any questions /
feedback to
zack@infinitedigits.co
or find me on insta
@infinitedigits.



nyblcore.com/demo

v1.0.1

first steps



step 1: turn the **selector knob** completely *clockwise*. in this position the a/b knobs are stretch & stutter probabilities. turn **knob a** completely counter-clockwise. turn **knob b** completely counter-clockwise. this resets the stretch & stutter probabilities to zero.

step 2: turn the **selector knob** to 12 o'clock. in this position the a/b knobs control jumping & retriggering. turn **knob a** completely counter-clockwise. turn **knob b** completely counter-clockwise. this resets the jumping & retriggering probabilities to 0.

step 3: turn the **selector knob** completely *counter-clockwise*. in this position the a/b knobs are volume & tempo. turn **knob a** until the audio comes through (turning further clockwise will add wavefolding). then turn **knob b** until you find a tempo you like.

at a glance

selector knob *fully counter-clockwise* sets **knob a** to control volume/distortion and sets **knob b** to control tempo.

selector knob *fully clockwise* sets **knob a** to control stretch probability and sets **knob b** to jump probability.

selector knob at *any other position* sets **knob a** to control jump probability and sets **knob b** to retrigger probability.

selector knob



uploading audio

uploading audio requires changing the firmware on the **attiny85** and requires an Arduino or an usb avr programmer (for more on this see nyblcore.com).

Choose audio files to upload (up to 1.2 seconds)

audio1.wav, file size 48.5MB.

audio2.wav, file size 44.5MB.

Seconds (max 1.2 seconds):

1.2

Slices (single files only):

5

Crossfade (ms):

0

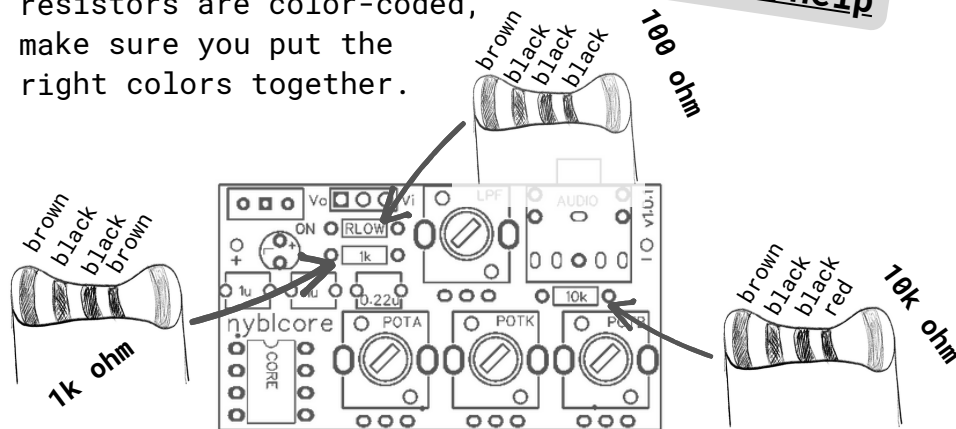
Submit

if you have a programmer, go to **nyblcore.com** and scroll to the bottom to find a webform that automatically generates new firmware from uploaded audio.

+resistors

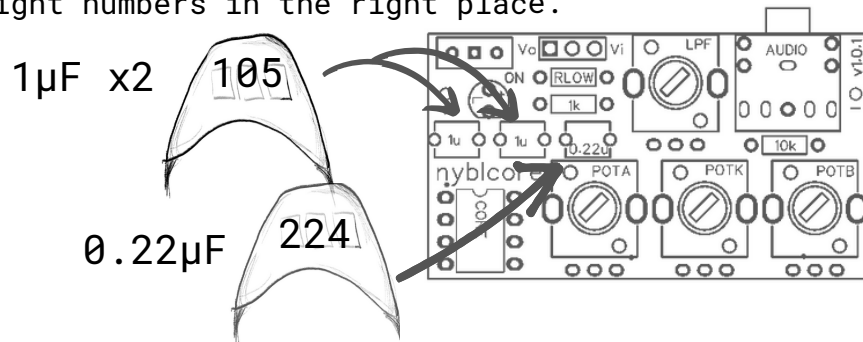
resistors are color-coded, make sure you put the right colors together.

help? visit
nyblcore.com/help



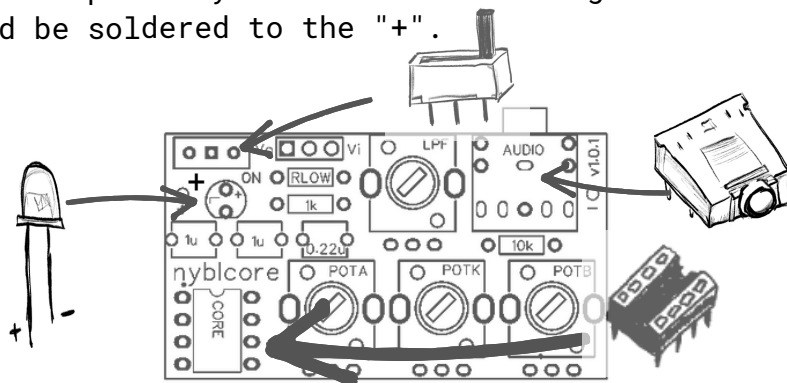
+capacitors

capacitors are numbered, make sure you use the right numbers in the right place.



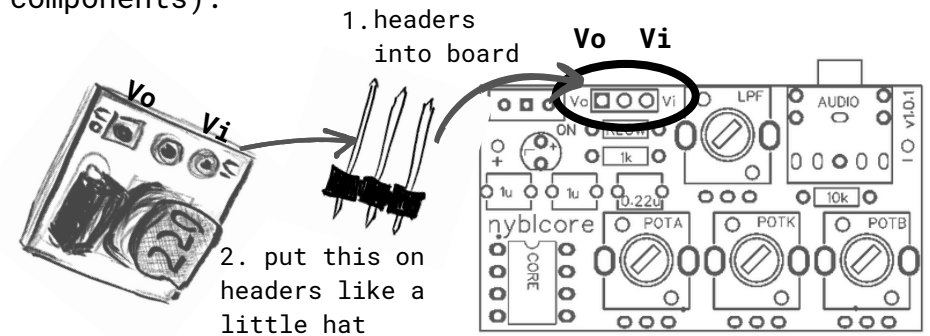
+ic socket, switch, led, audio

watch the polarity of the led! the long end should be soldered to the "+".

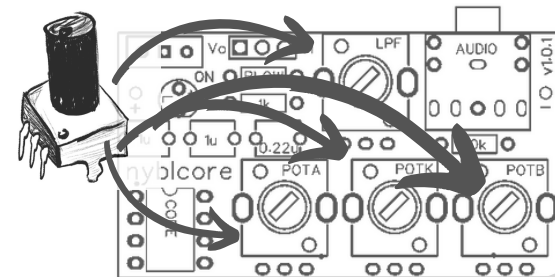


+dc-dc converter

solder the 1x3 header to the board, long legs up. then solder dc-dc converter to the header - **please make sure "Vi" and "Vo" are aligned!** (it's okay if the dc-dc converter sits on components).



+pots



help? visit
nyblcore.com/help

+battery holder

attach to the back and then solder the front side. make sure the "+" is on the left (closest to switch). note: the pins need to be bent out to fit.

