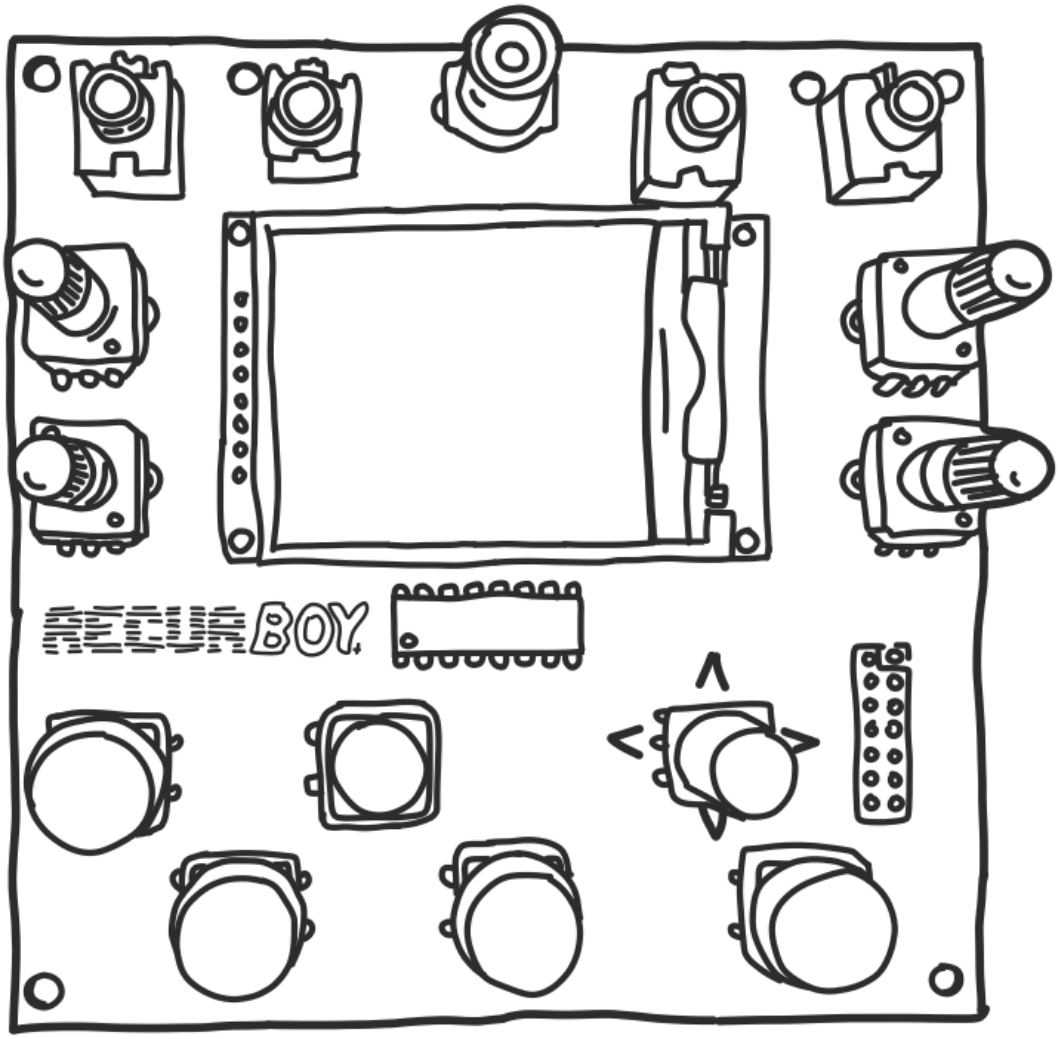
cyberboy666 & underscores.shop present
a circuit designed with guergana tzatchkova

recurBOY

rpi_zero video instrument



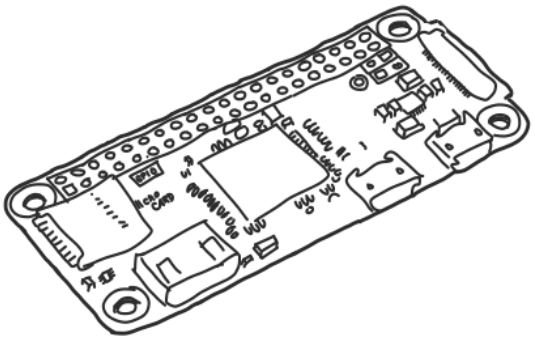
instruction manual and build guide
V0_6

View this project online at
underscores.shop/_recurBOY_

BACKGROUND

recurBOY is a raspberry pi based diy video-instrument for live performance. it was originally designed by tim & guergana to be built together with others in group workshop soldering sessions

The design is a spinoff from **r_e_c_u_r** an existing project tim created and maintains. **r_e_c_u_r** is simple to assemble but more complex to operate due to its scope and customizablity. *recurBOY* distills the best parts, aiming to be simpler and more beginner friendly. it uses cheaper parts and runs on a **raspberry pi zero** which can be a fraction the price of the pi3 used in r_e_c_u_r.



RASPBERRY PI ?

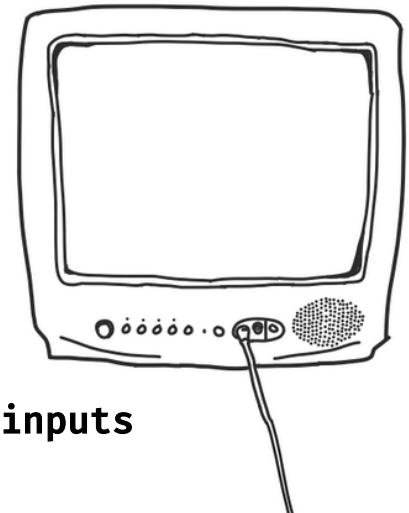
a raspberry pi is a small computer – often referred to as a Single Board Computer or SBC. it has 40 pins on it called GPIOs - General Purpose In Out - which can be used to connect it with other components. it also has a pin called tv_out for outputting composite video.

Your rpi0 must be sourced separately

HOW WAS IT MADE ?

the recurBOY application is built in openframeworks – a collection of open-source c++ libraries for creative coding. In particular it uses *ofxVideoArtTools* - an abstraction of openframeworks libraries and extensions into modules specifically for making video instruments on raspberry pi. integration with the TFT display is made in python.

FEATURES

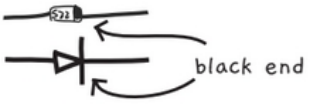


- outputs sd composite video
- 2 source modes : sampler & shaders
- process any source with additional FX
- control shader/fx parameters directly with **4x knobs** and externally with **4x cv inputs**

BUILD INSTRUCTIONS

It is highly recommended to use the interactive BOM to help with placement on this build – type *kutt.it/iccIqU* into a browser or find the links from the github page - components should be soldered in this order:

Step 1: R1 - R4, D1 - D8
start by placing the resistors and diodes. it is important that the diodes are placed in the correct direction ! for resistors it does not matter.



Step 2: J8, J6
there are two parts that need to be *placed from the bottom*. this is so the raspberry pi can be attached underneath. turn the board over and find j8 - the 2x20 pin socket will connect to the pi0 gpio pins. it is very important that these are soldered on straight. next to this is j6 – a double pin socket that will connect to the pi0 tv out.

Step 3: U1, SW1-5, J9
flip back to the front. now you can solder the ic and buttons. use the ic socket. also try to make sure the 5-way button is straight before soldering all the pins

Step 4: J5
for screen make sure the included 8x1 pin header is soldered to it first - with short pin end to the screen. then you can solder the screen - with long header pin ends through the recurBOY pcb.

Step 5: J1-4, J7
next solder the top row of jacks and the RCA connector

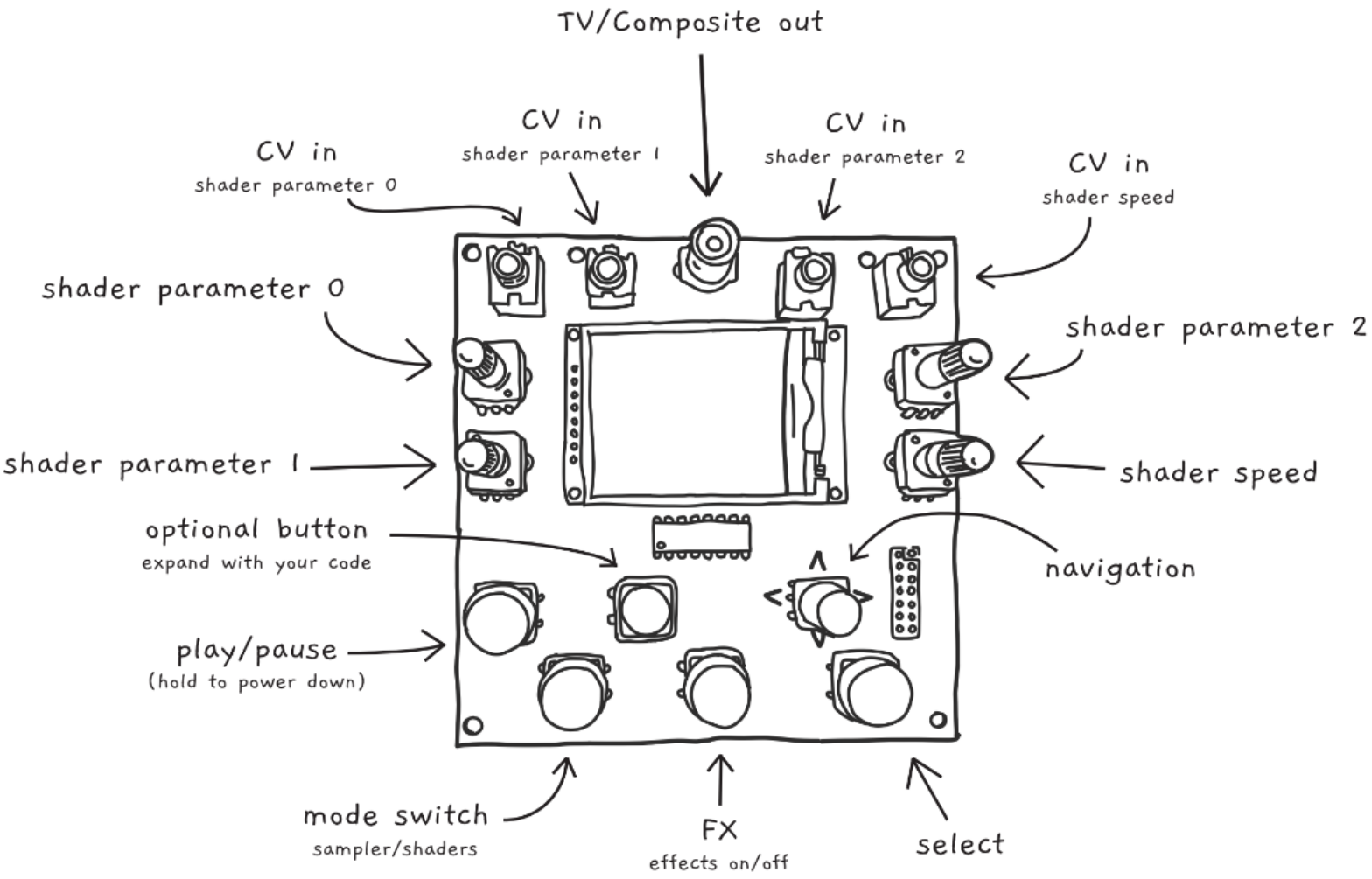
Step 6: RV1 - RV5
now you can solder the potentiometers.

you may need to also solder the 2x20 gpio-header to the raspberry pi and a 1x2 pin-header for the tv-out - then your raspberry pi should slot into the bottom of the pcb.

flashing image to sd card

you can flash the recurBOY image to a sd card if you didnt get one from me, or to update its firmware by running a program like *balena etcher* – links to downloading the image file can be found on project github page

OPERATING INSTRUCTIONS



NAVIGATION

recurBOY has 2 source modes : sampler & shaders. pressing the MODE button will cycle through these modes. you can tell which mode is selected by looking at the title and colour of the display.

CONTENT SELECTION

the nav_button can be pressed UP , DOWN , LEFT, RIGHT and IN. we will not use the IN button for now. pressing UP and DOWN lets you scroll through the list of content - either samples or shaders depending on the MODE. pressing SELECT on a row will start playing it. the playing content will be highlighted on
the display and the play symbol will display next to the MODE. pressing the play/pause button will toggle this state. when the content is stopped the will be displayed.

to safely turn off recurBOY hold down the play/pause button for 5 seconds. it is not recommended to remove your usb-drive while recurBOY is operating.

OPERATING INSTRUCTIONS CONT.

Videos

the content list while in sampler MODE comes from the **~/Videos** folder on the pi's SD card and the **/Videos** folder on top level of an attached usb-drive. any .mp4, .mkv, .avi or .mov file will be shown although it is not guaranteed it will work with the player - we find sd h264 mp4 to be most reliable.

Shaders

the content list while in the shaders MODE comes from the **~/Shaders** folder on the pi and in the **/Shaders** folder on top level of an attached device.

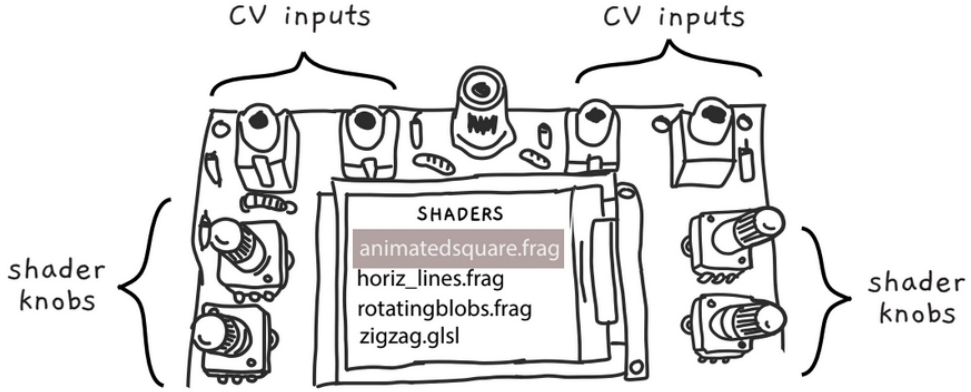
while the display is in SHADER mode you can use the 4 knobs or CV inputs to manipulate the shader parameters. each shader has 3 parameters mapped to input 0, 1 and 2. input 3 always controls the speed.

Fx

from any source mode (SAMPLER or SHADERS) you can press the *RIGHT* nav_button to enter **FX mode**. This mode applies effects or filters to the media that is currently playing. Once you enter **FX mode**, you can navigate the same way as in SAMPLER or SHADERS with *UP*, *DOWN* and start the effect with *SELECT*. pressing *LEFT* will return to the previous screen.

these shaders need to be stored at **~/Fx** folder on the pi and **/Fx** folder on top level of attached usb drive. pressing the **FX** button will toggle the selected effect on and off. this effect will
process whichever of the sources is selected.

the 4 knobs / cv inputs will control parameters of the effect when source SHADERS mode is not selected.

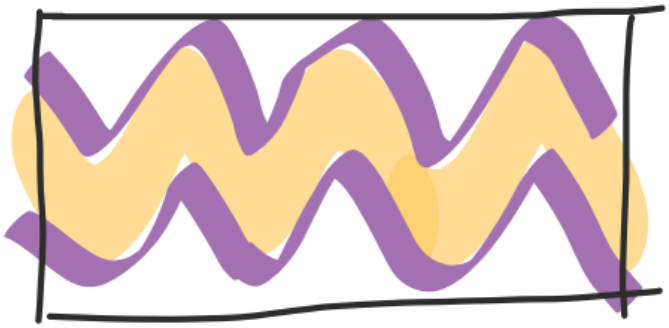


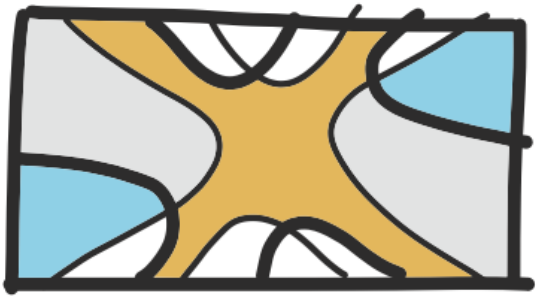
WHAT ARE SHADERS ANYWAY?

shaders are small text files of code that tell a graphics card what to draw. they use a language called glsl to communicate what colour a pixel should be and where. you don't have to understand every line to begin playing around with them.

in recurBOY we use shaders in two places - the **SHADERS** mode is used for launching shaders that generate video. these shaders take no video inputs. we also use shaders in the **FX** mode. here we are selecting shaders that process video. these take one video source as input and pass it through the fx shader.

the type of shader recurBOY can play is called *GLES*. this is the shader language used for **e**mbedded **s**ystems, including raspberry pi's and mobile phones.





thanks to *Erogenous Tones* - a modular synth company who have a mature and very powerful shader-playing video instrument called **STRUCTURE** - we now also have a web-based environment for browsing, modifying and creating shaders to perform with.

go to *glsl.erogenous-tones.com* - if you select any example you will see the code used to create the patch. try changing some of the numbers - modifying the input parameters is especially interesting.

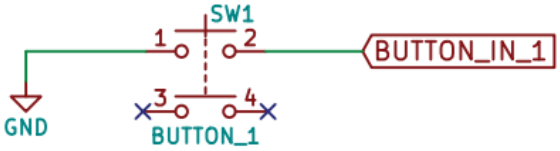
when you are happy with the results select the save as .glsl file. Now copy this file onto your USB in the correct folder - **/Shaders** and connect to recurBOY

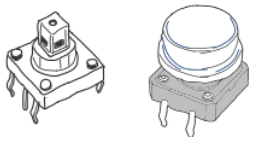
for more information on understanding and writing shaders yourself check out *The Book of Shaders* by Gonzalez Vivo.

HOW THE CIRCUIT WORKS

reading button presses

we can use some of the GPIO pins on the raspberry pi to know when a button is pressed. one side of the button is connected to the pin and the other is connected to ground. on the raspberry pi we tell these pins to pull up. this means they are HIGH by default.
when the button is pressed however the circuit connecting the pin to ground is completed and the pin becomes LOW. in the code we ask the state of a GPIO pin and if LOW we know the button is pressed.



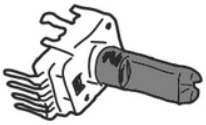


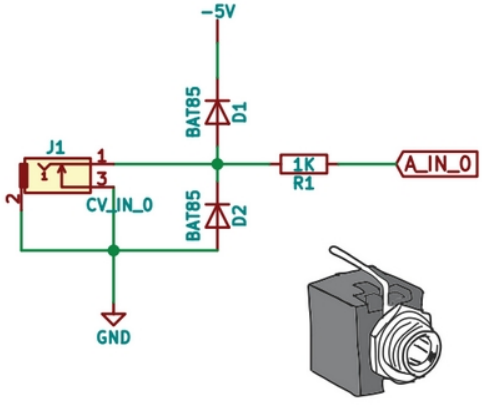
reading continuous inputs

digital pins - HIGH/LOW are good for discrete inputs like button presses which can only be OFF/ON. however we also want to have continuous inputs for example from knobs which can be set to any amount between LOW and HIGH. this kind of input is called analog - the reading is analogous to the voltage on the pin.
since raspberry pi has no analog GPIO pins we need to introduce a new part - the **MCP3008**. this type of ic is called an analog to digital converter or adc for short. it has 8 analog pins which read the voltage applied to them and converts it to digital information. this information can be understood by the raspberry pi's digital pins. in this case using a digital protocol called SPI.

potentiometers

four of the MCP3008 channels are connected to potentiometers which in this circuit act as voltage dividers. one side of the pot is connected to +5V and the other to GND. The output is always some voltage between these. after passing through the adc this voltage value between GND and +5V is converted to a number between 0 and 1024.





cv inputs

the other four MCP3008 channels are connected to 3.5mm jacks. this allows the voltage to be set by external devices - this kind of interaction between instruments is called Control Voltage or CV each of the CV inputs also use a resistor and two diodes - these are to protect the IC from incoming voltages above +5V or below GND

display

the raspberry pi also connects to the display screen with GPIO pins - this time the pins are used as outputs, telling the screen which pixels to colour - again the protocol used here is SPI - but we dont need to worry about how exactly this works - there is a python library that is used to describe what the screen should show.

CREDITS AND MORE INFO

This circuit is distributed through UNDERSCORES – open video hardware label – visit underscores.shop for more info

The pcb was designed using KICAD , this booklet was created in LibreOffice Draw

Everything from gerbers, cad files, panels and documentation is freely available online and distributed under CC-BY-SA / open-source licenses – help us contribute to the commons !

Ask any questions or start discussions related to this project on the *scanlines.xyz* forum – an online community space dedicated to diy av / electronic media art

You can contact me directly at *tim (at) cyberboy666 (dot) com*
Please get in touch if you are interested in hosting a workshop !

Thanks to Guergana Tzatchkova for all the love put into creating this project, and the design inspiration used across my work. to Bastien Lavaud for circuit advice, always. To Ben Caldwell for project advice. To everyone who has or will contribute ♥♥♥