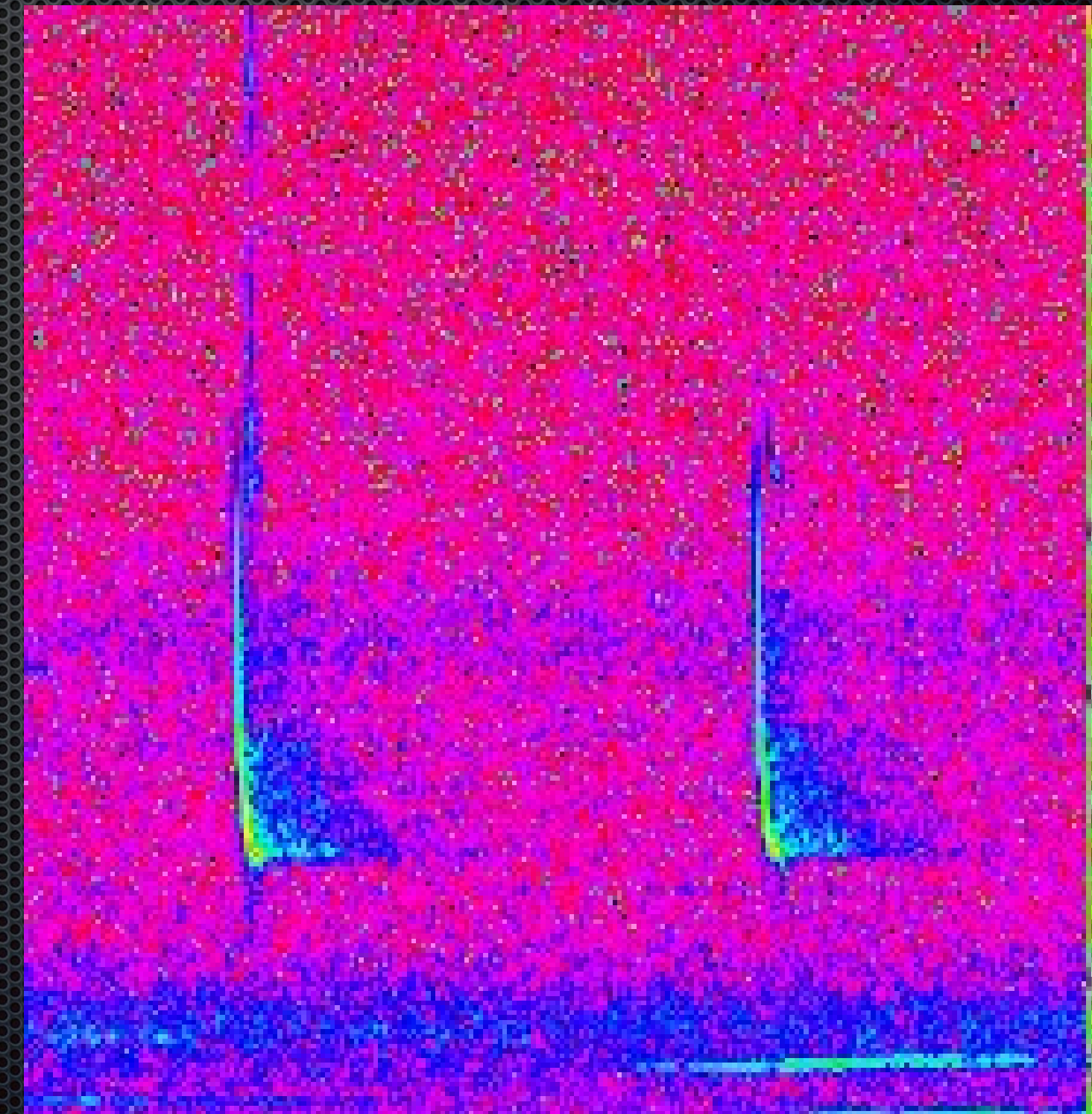


Dr. Olaf Flebbe

Fledermäuse oder C neu entdeckt





# About me

PhD in computational physics  
(Theor. Astrophysik Tübingen)

Former projects: Minix68k (68k FP  
Emulation), Linux libm.so.5 (High  
Precision FP), perl and python for  
epoc, flightgear, msktutil

Member Apache Software  
Foundation, PMC Apache Bigtop

Lead Developer/System Specialist  
Bosch eBike





# Motivation



- Habe professionelles Fledermausdetektoren gesehen/gehört
- „Das kann doch nicht so schwer sein“



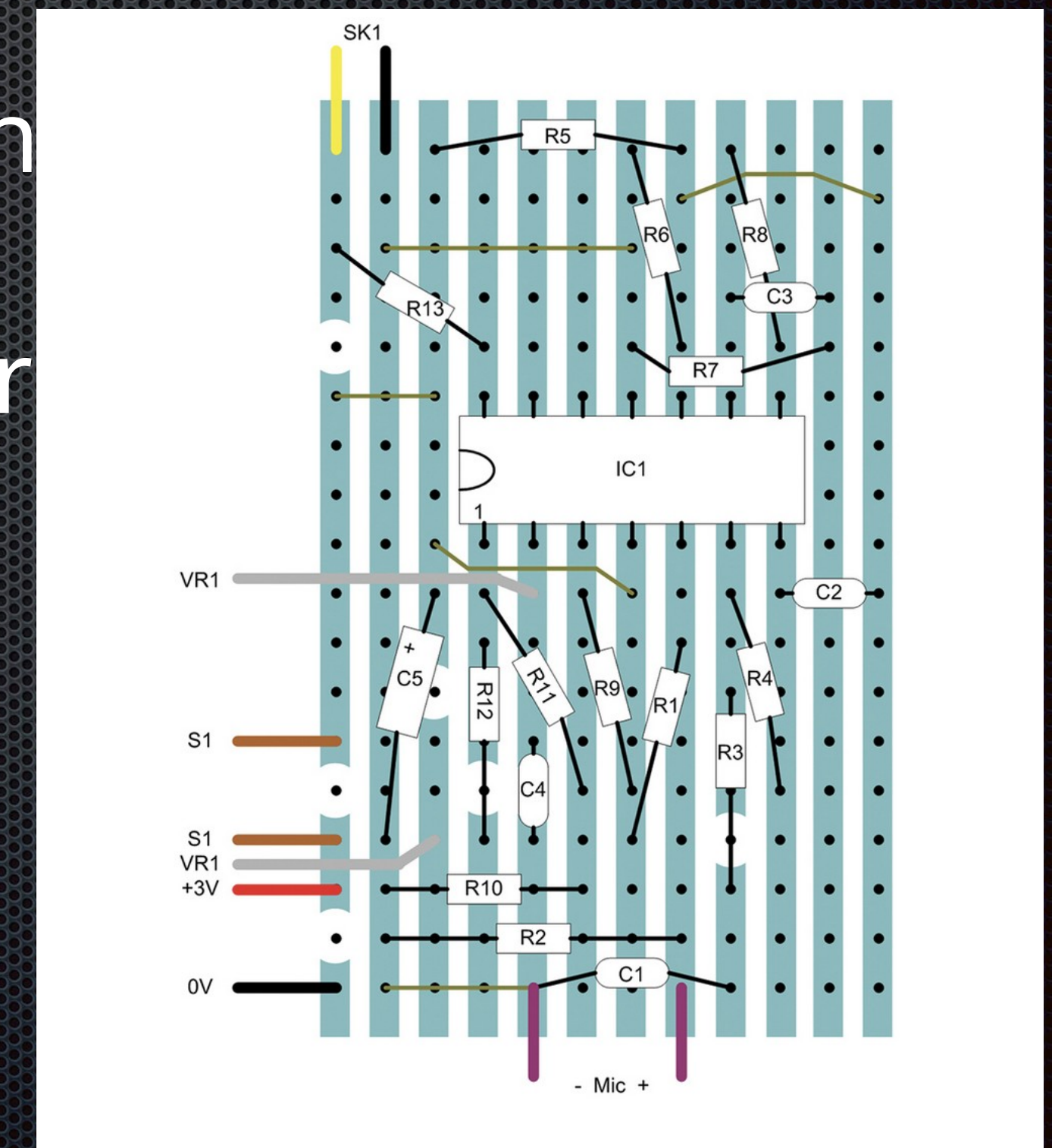
# Recap

- Fledermäuse können im Ultraschallbereich (20kHz – 200kHz) rufe aussenden, die spezifisch für die jeweilige Art sind
- Detektoren sind entweder Heterodynverstärker (Hörbar machen mit Differenz eines Referenzsignal) oder Spektrumanalysatoren (Frequenzverlauf/Umfang sichtbar machen)



# Erster Versuch

- <https://www.nutsvolts.com/magazine/article/jun>
- Zum ersten mal von einem Heterodynverstärker gehört
- Hat nie funktioniert  
Probleme: Ohne Oszilloskop nicht zu debuggen  
Breadboard plan ist fehlerhaft  
stimmt nicht mit Schaltung überein





# Learning Versuch 1


- Learnings: Lochstreifen Platinen statt Breadboard
- Man braucht ein Oszilloskop
- Mikro ist nicht so einfach wie dargestellt:  
einfach Piezo Lautsprecher ausprobieren





# Zweiter Versuch

- Von ELV Final zusammenlöten
- Hat auf Anhieb funktioniert
- MEMS Mikro auch einzeln bestellbar

**ELV Bausatz Fledermaus-Detektor FMD1**



**BAUSATZ**



**39,95 €**  
inkl. MwSt.  
ggf. zzgl. Versandkosten

● sofort versandfertig Lieferzeit: 1-2 Werktage<sup>2</sup>

1

Bezahlen Sie nach 30 Tagen. [Mehr erfahren](#)

★★★★☆ (8)

Artikel-Nr. 151462  
EAN: 4047976514625

Steigen Sie ein in die Welt der sonst nicht hörbaren Ultraschallsignale!

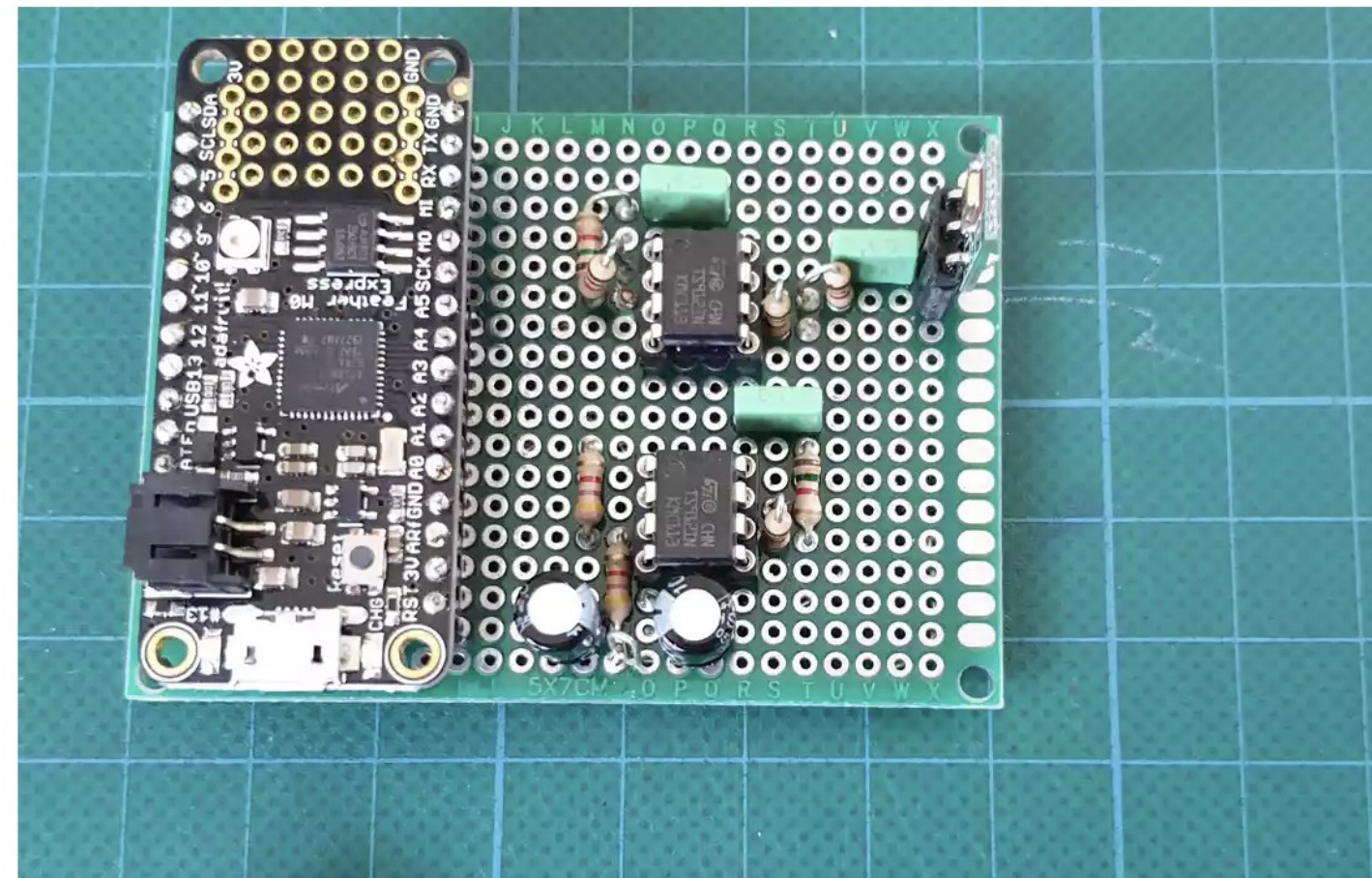


# Dritter Anlauf

- Make Magazin 6/2020 , 4/2021



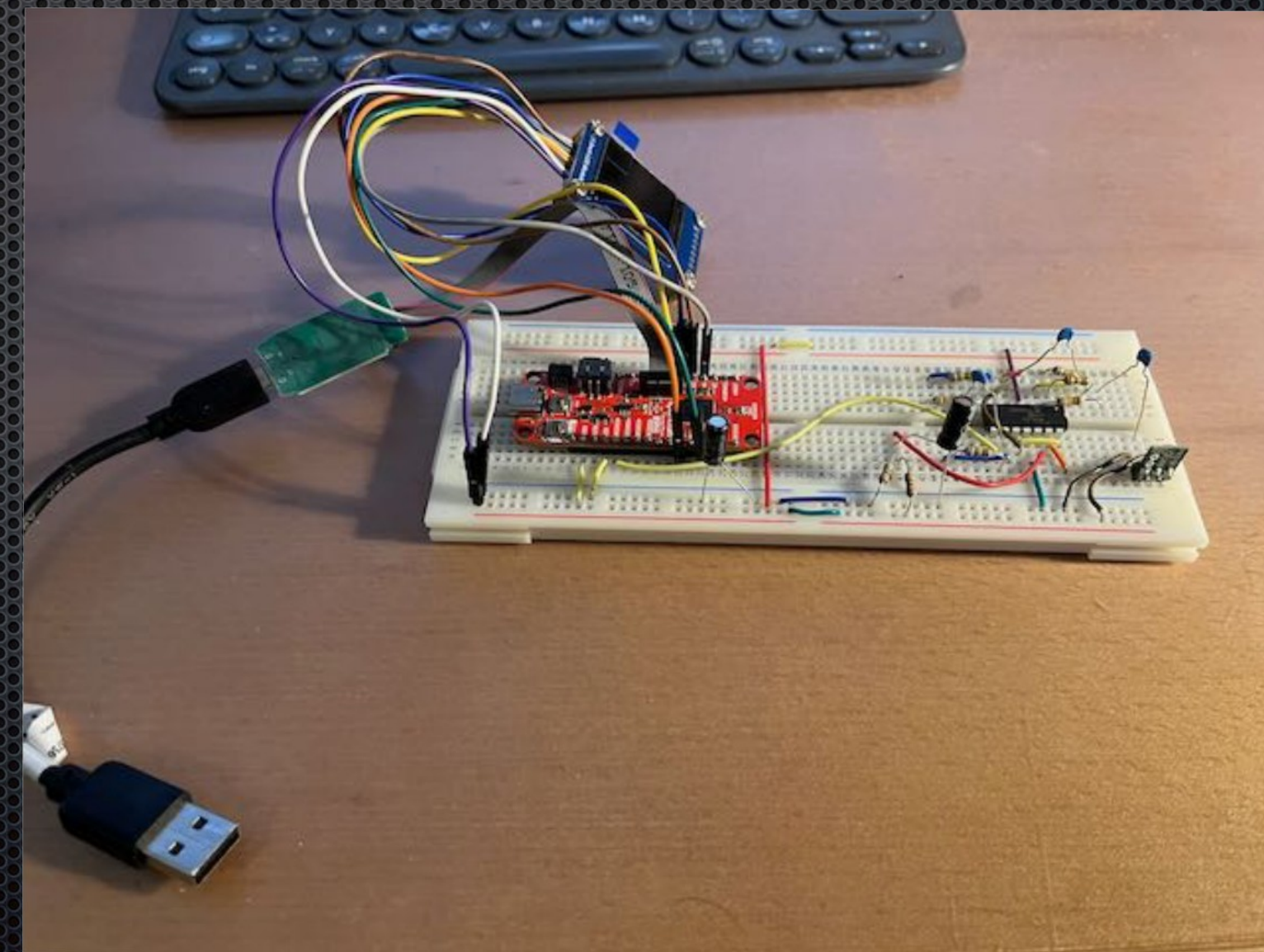
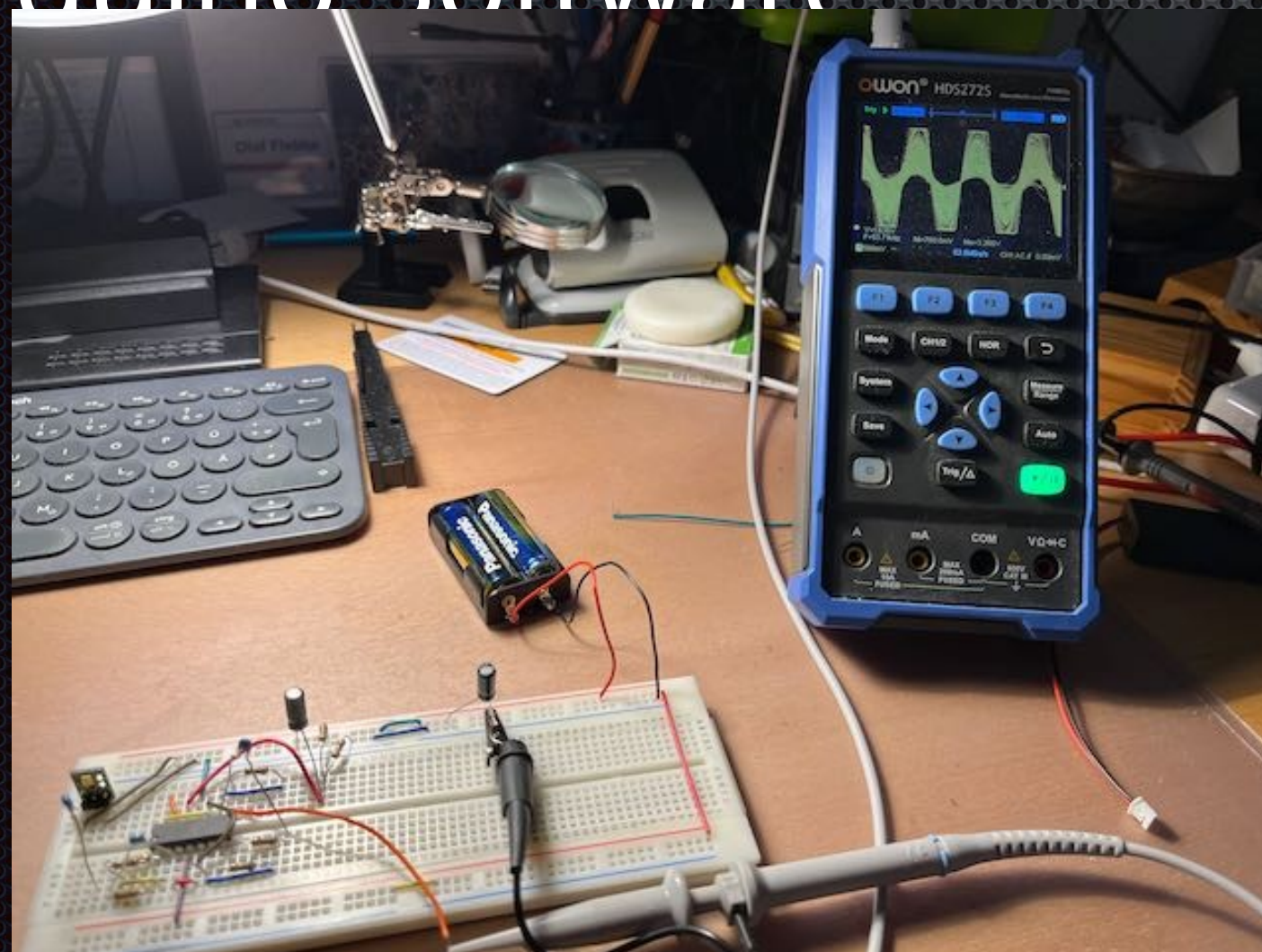
terfahrung.





# Dritter Anlauf

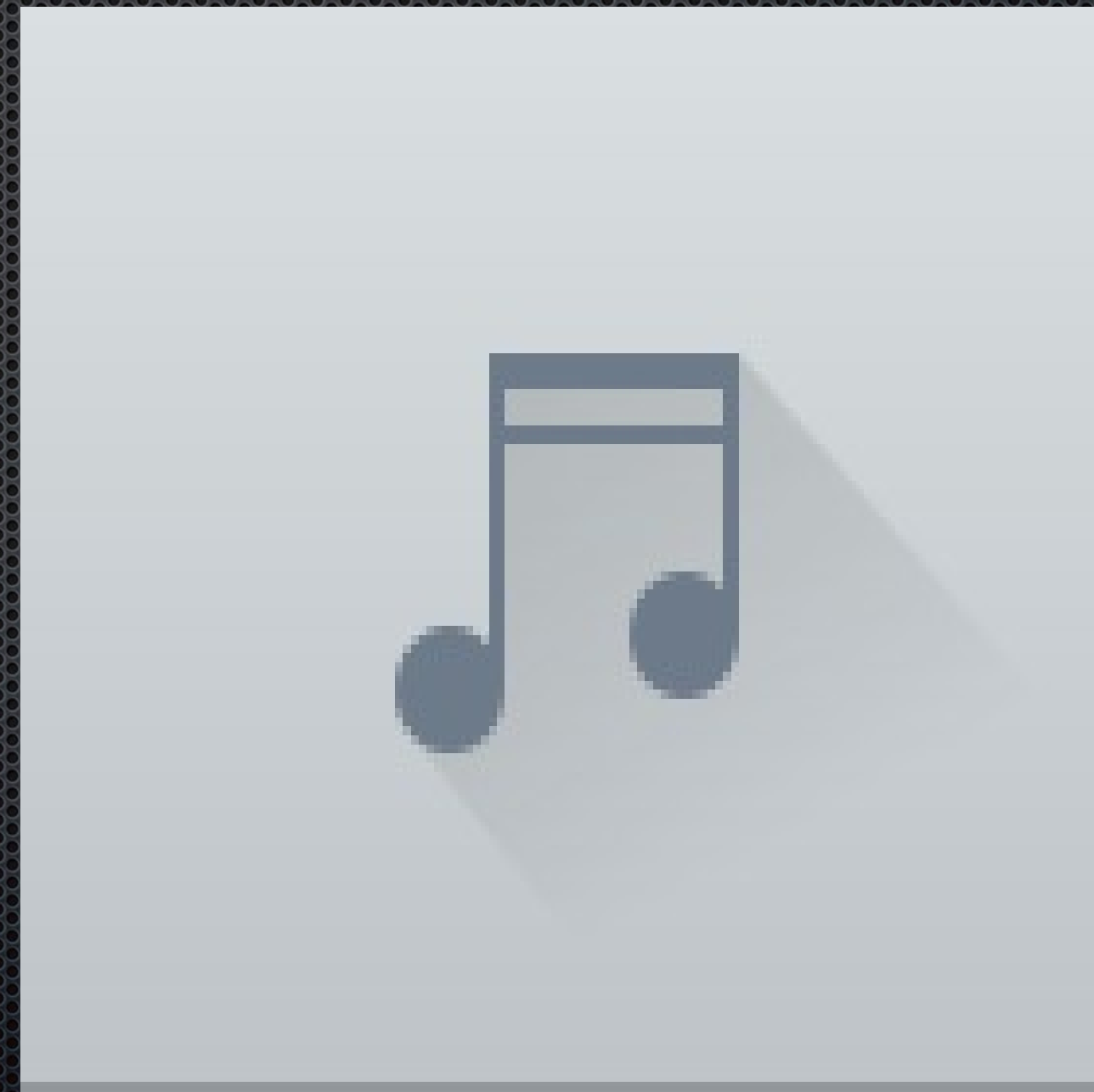
- Ein verständliche Schaltung aus 2 Opverstärkern (Quark: geht auch mit 1 doppelten)
- Verwendet das ELV MEMS Micro
- Arduino Software





## Dritter Anlauf

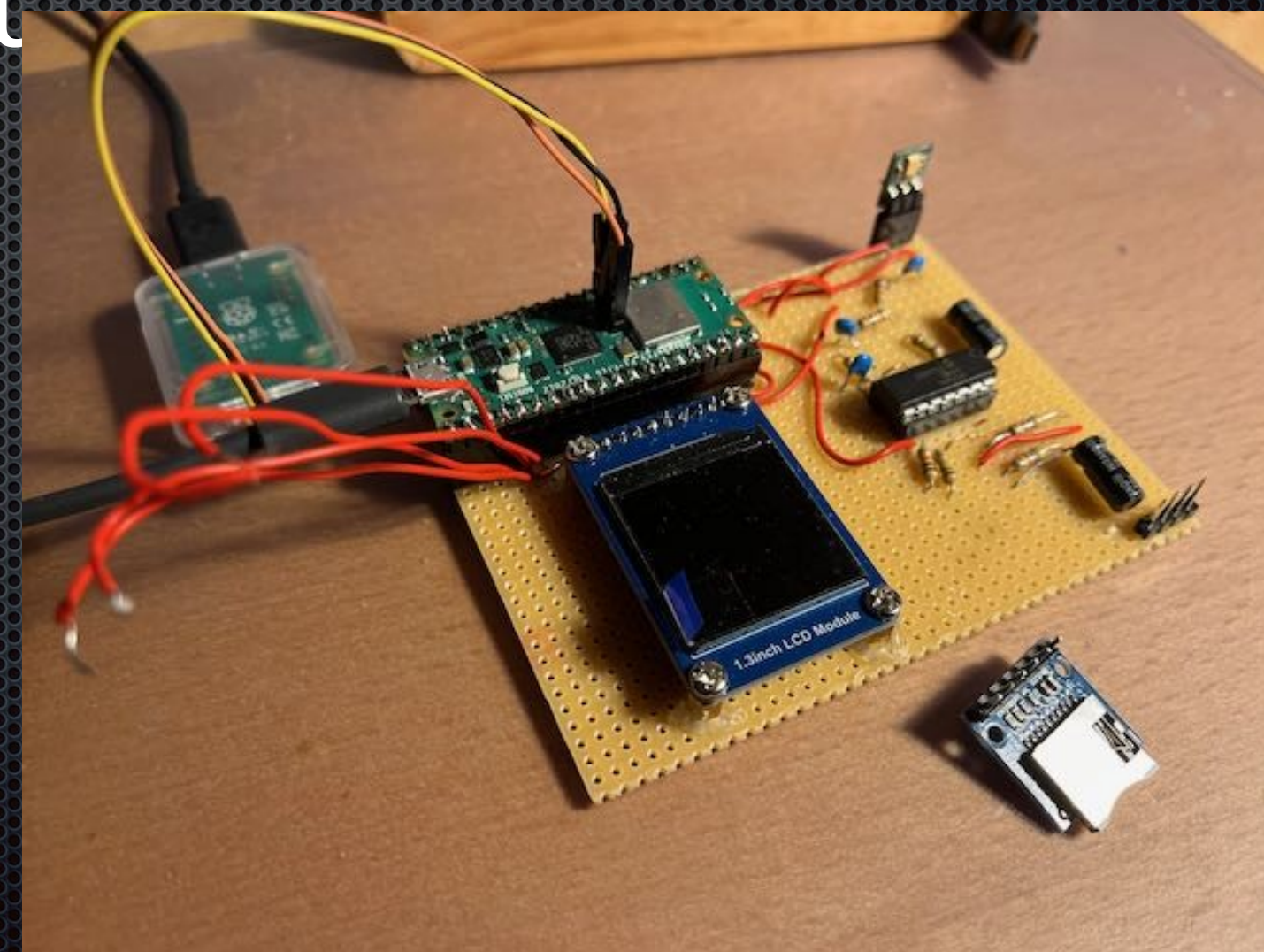
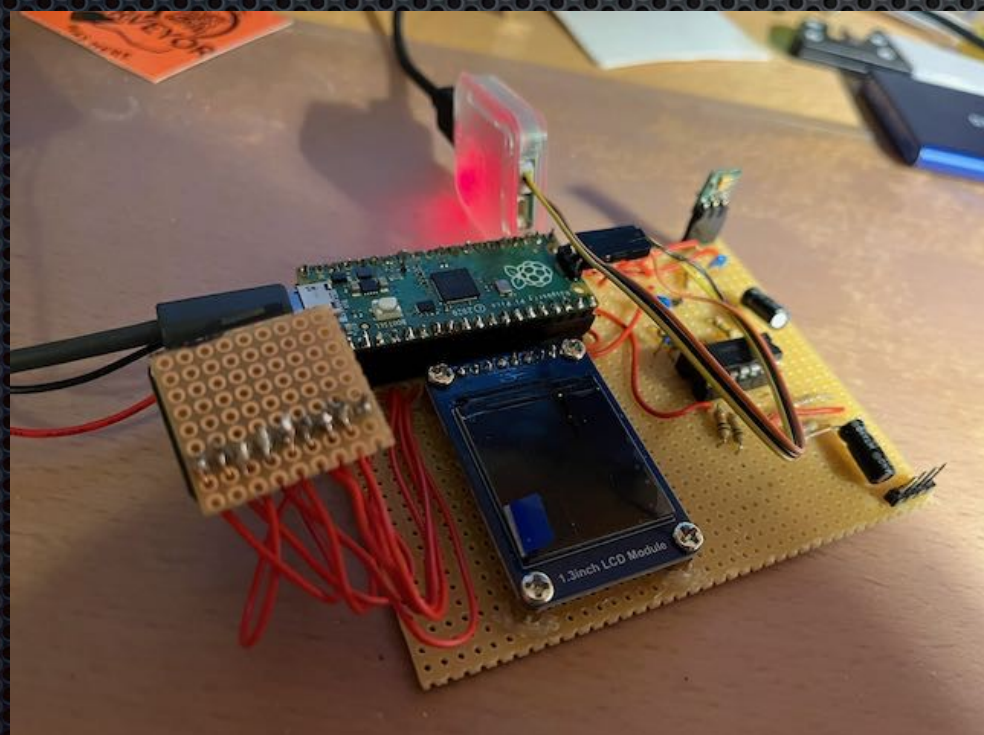
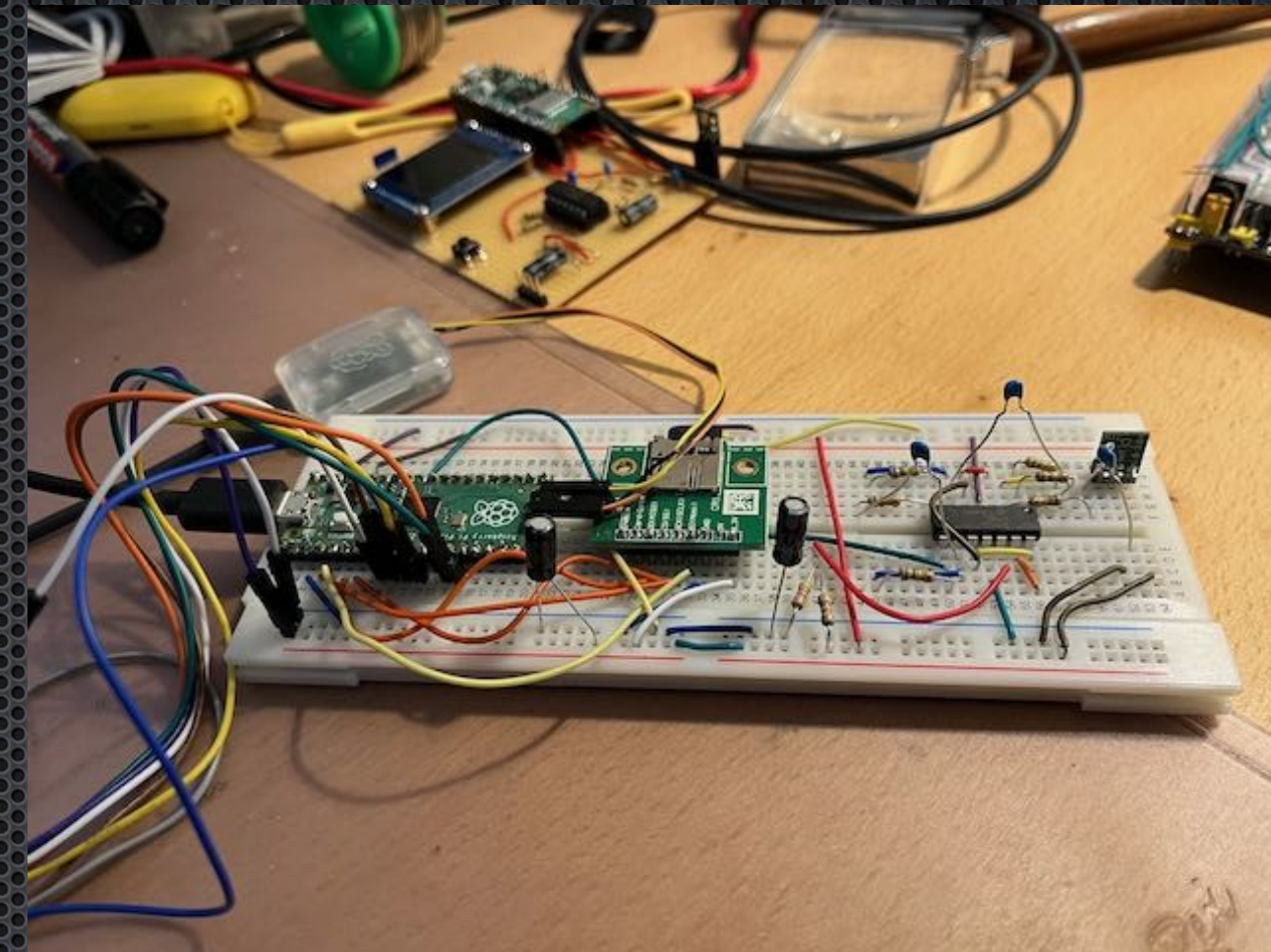
- Auf Breadboard aufgebaut: Da kommt was
- Lochstreifen Board geplant und bestückt
- Mit rp2040 Microcontroller
- Da geht was
- Dann aus vorherigem Projekt st7789 Display dazu
- Spektrum (Februar)





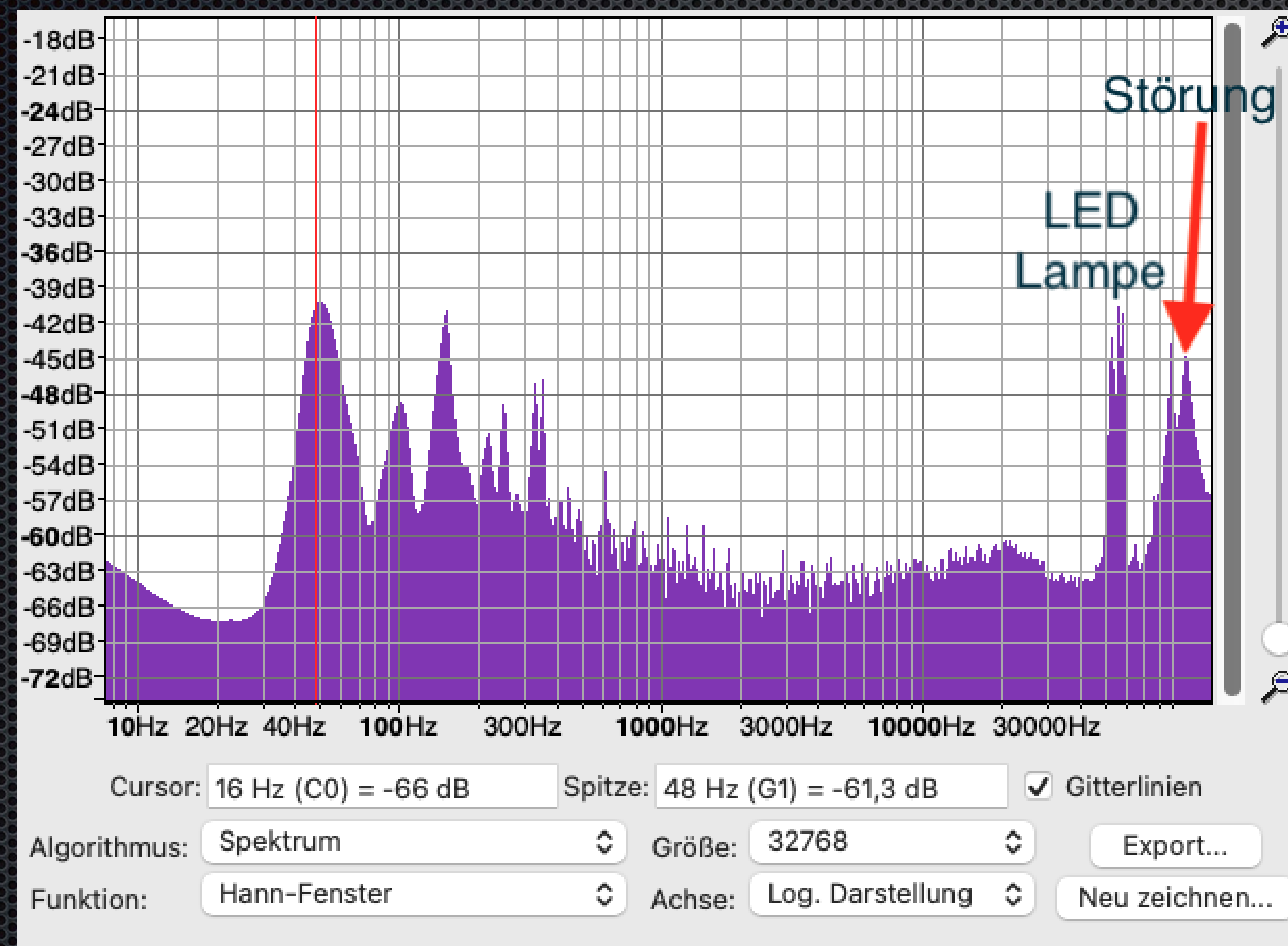
# Verfeinerung (?)

- Sdcard zum Datenloggen mit SPI Bus
- Billig SDCard
- Nochmal am Breadboard aufgebaut:
  - Katastrophale Störungen





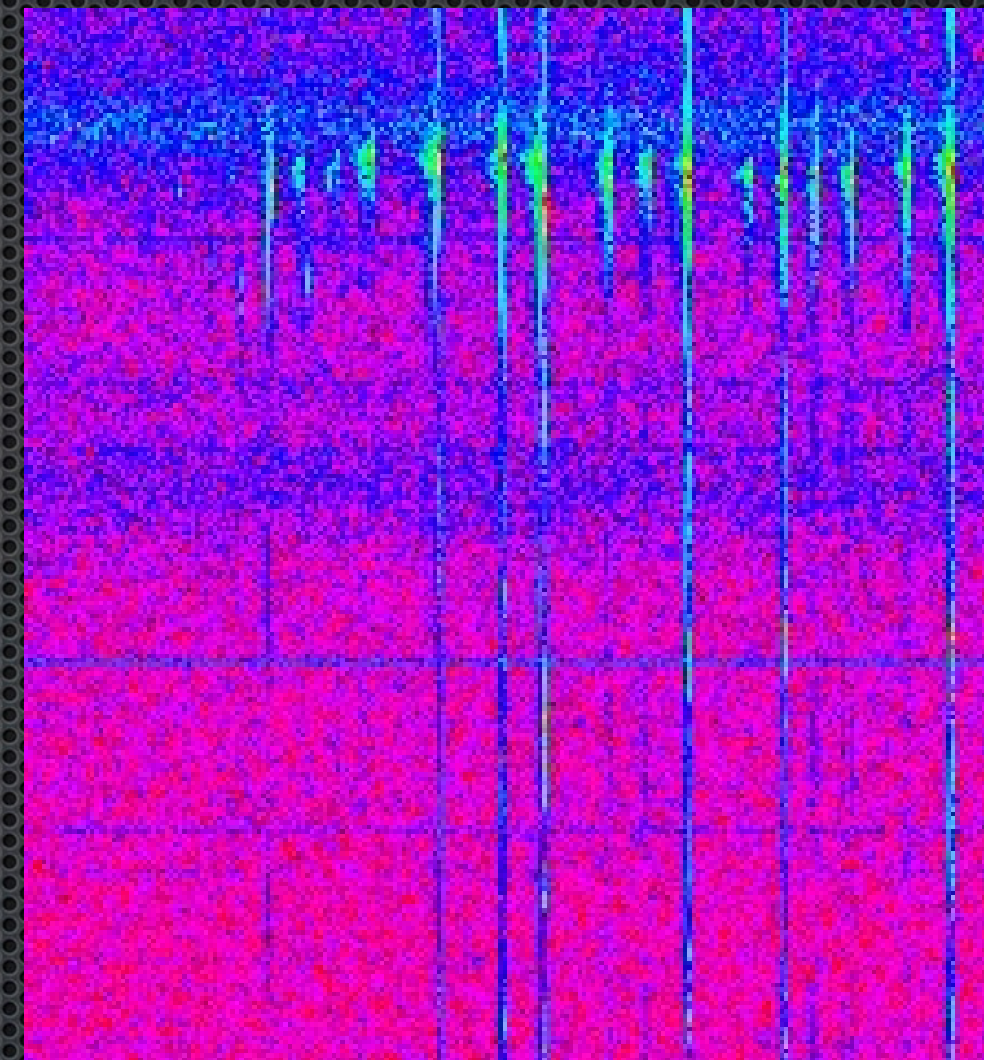
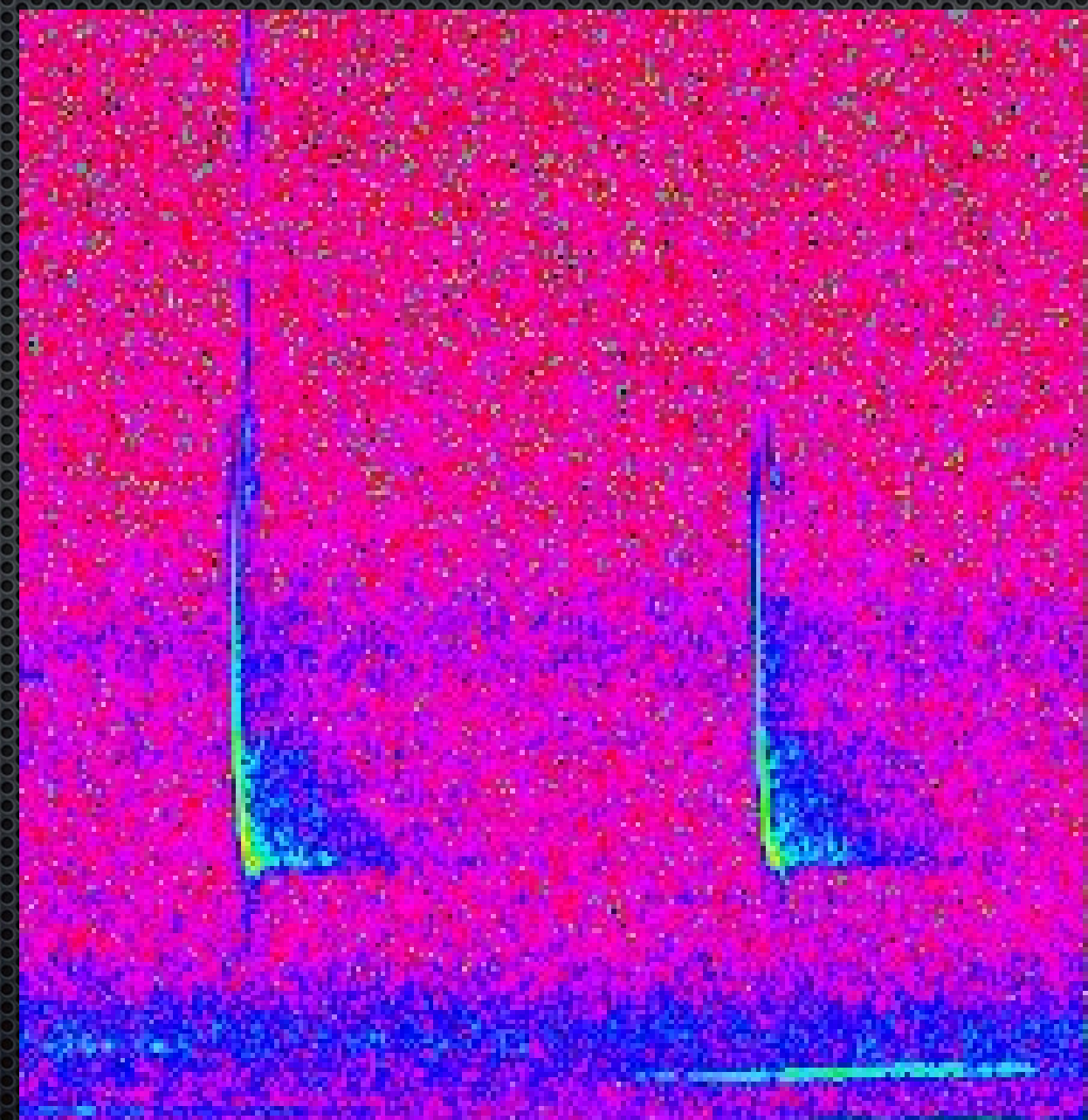
# Analyse mit Audacity





# Knoten Löst sich

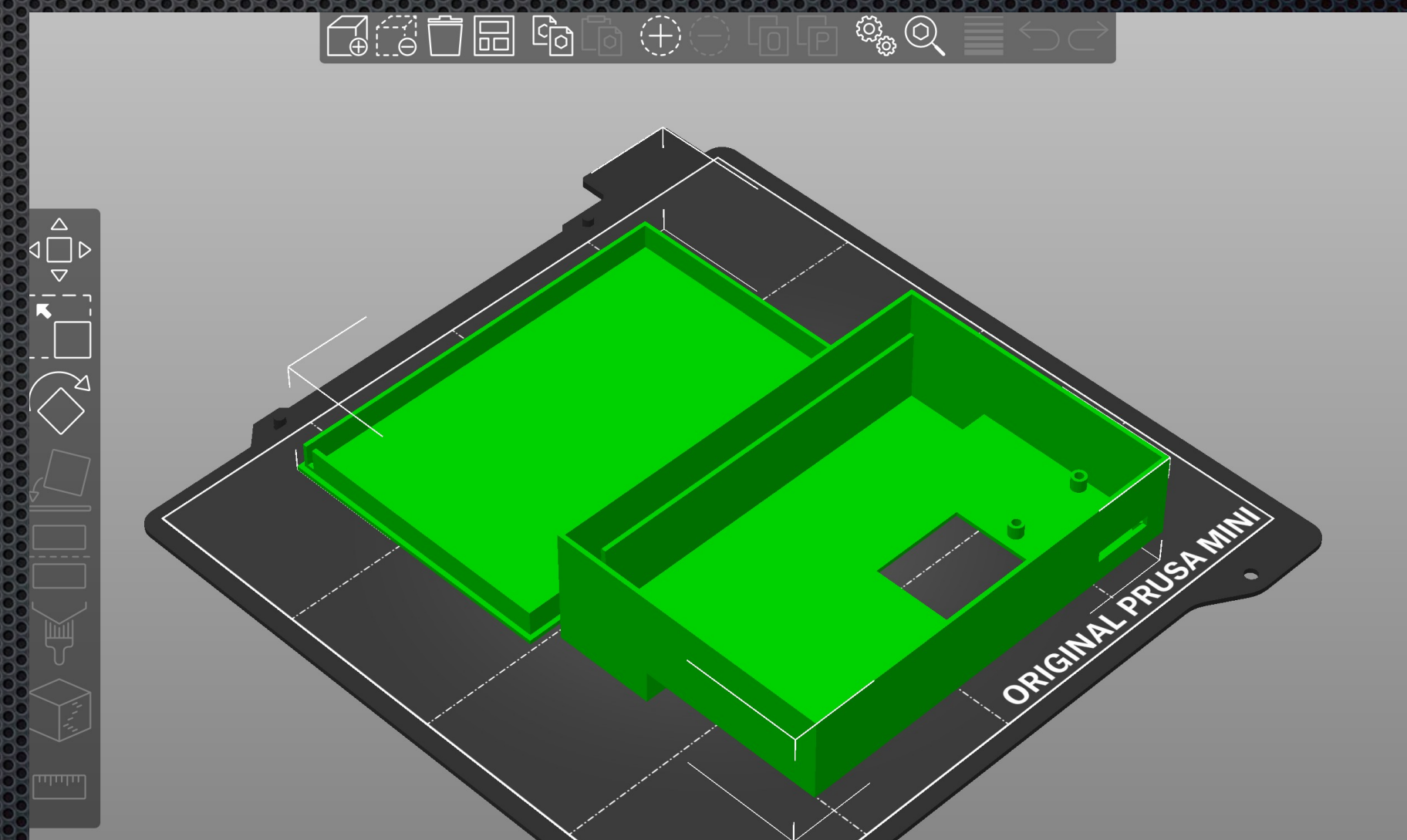
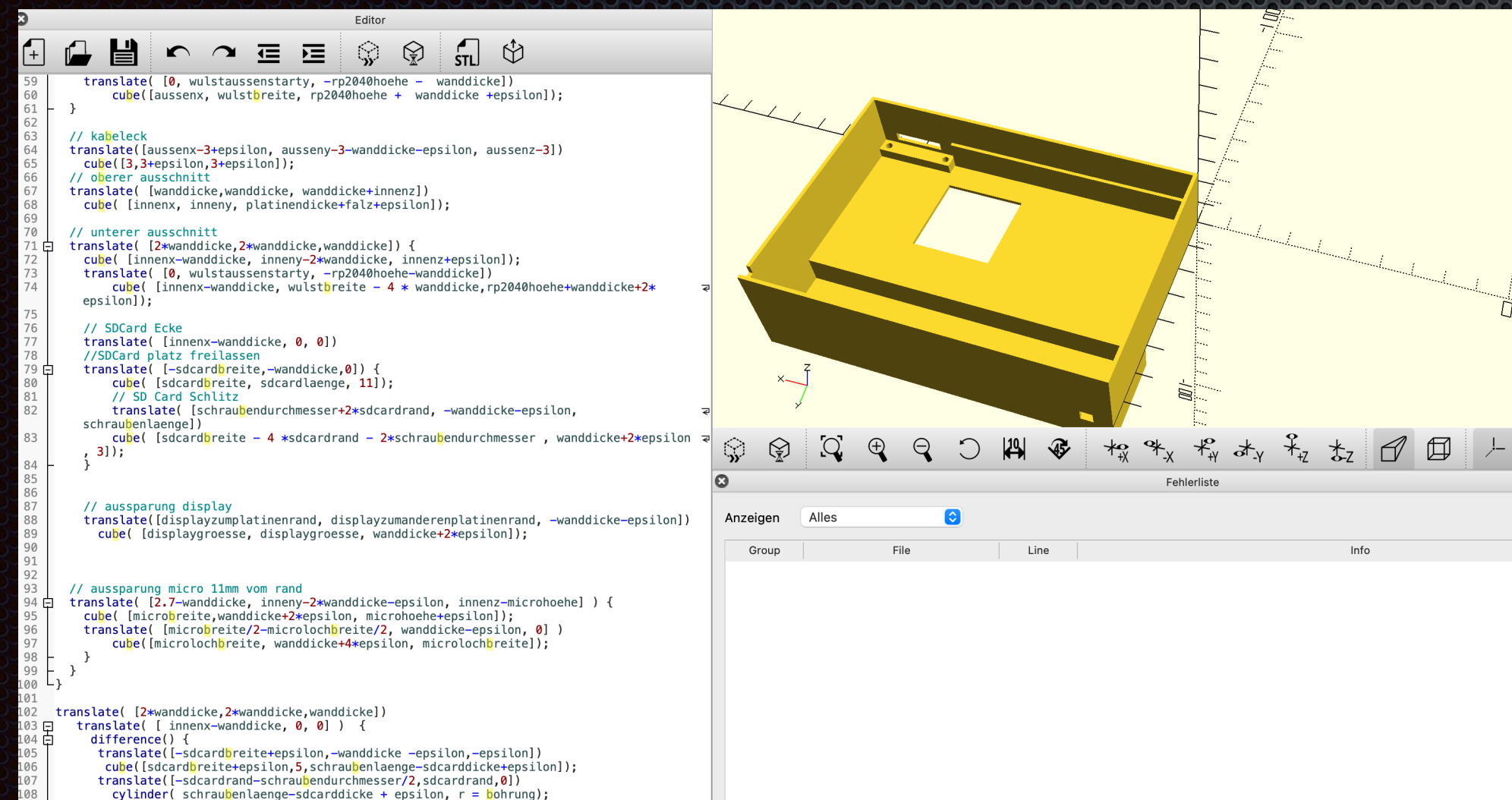
- Erstes Spektrogramm (Ende März)
- SDIO Karteninterface
- Verbessert (April)
- Zwergfledermäuse!



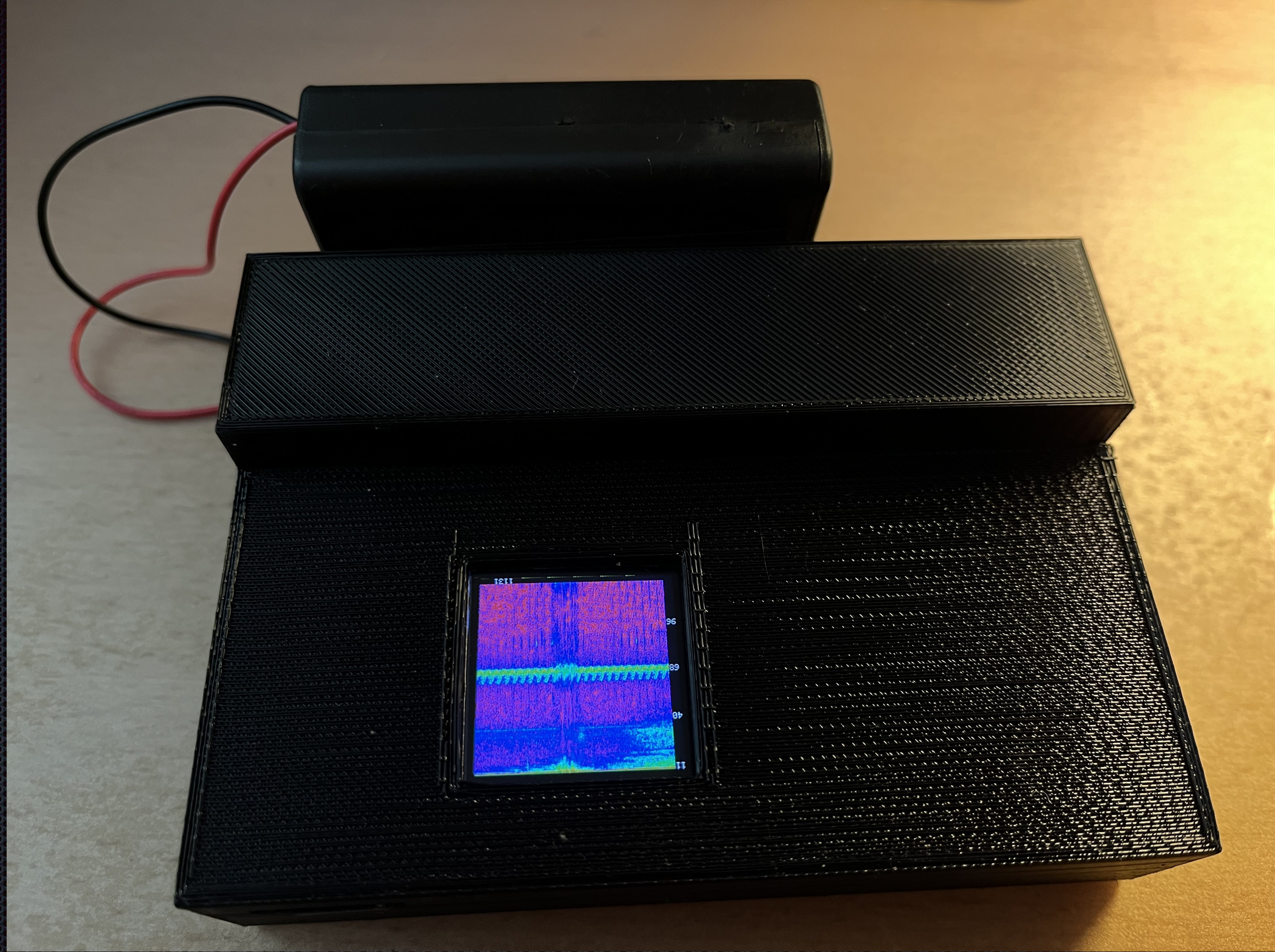


# Sidequest

- OpenSCAD / Slic3r









# Microcontroller

- Was unterscheidet Microcontroller von einer CPU / Processor (zB Intel/AMD)?
  - Er ist viel billiger (1EUR statt 100 EUR)
  - Wenig RAM Arbeitsspeicher (256 kB) auf dem Chip
  - Keine virtuelle Speicherverwaltung, keine spezial Befehle für Grafik, Numerik
  - GPIO, PWM, SPI, I2C, DMA
  - Brauchen kein Betriebssystem, deswegen Echtzeit / exaktes Timing einfacher



# Display ST7789 / LCD Module : Waveshare

The screenshot shows the Waveshare website's product page for a 240x240 LCD module. The page features a navigation menu at the top with links for HOME, PRODUCTS, PAYMENT, SHIPPING, WIKI, SUPPORT, and IC NEW. A search bar and a shopping cart icon are also present. The breadcrumb trail indicates the product's location: Home > Displays > LCD > 0.9" - 2.4" > 18079. On the left, a 'PRODUCTS' sidebar lists various categories like Raspberry Pi, AI, RISC-V, Linux Computer, Displays, IoT / Communication, Misc Modules, Robotics, Arduino-Related / Nucleo, micro:bit, MCU / ARM, FPGA, and Sockets / Adapters. The main content area displays the product title '240x240, General 1.54inch LCD Display Module, IPS, 65K RGB', its SKU (18079), part number, and brand (Waveshare). The price is listed as \$9.49, with a quantity selector set to 1 and an 'ADD TO CART' button. A table below shows bulk pricing: \$9.09 for 10+ units, \$8.89 for 50+ units, and \$8.81 for 100+ units. A 'Related Products' section is partially visible at the bottom.

WAVESHARE share awesome hardware

HOME PRODUCTS PAYMENT SHIPPING WIKI SUPPORT IC NEW

Search here 0 ITEM(S) - \$0.00

Home > Displays > LCD > 0.9" - 2.4" > 18079

PRODUCTS

- Raspberry Pi
- AI
- RISC-V
- Linux Computer
- Displays
- IoT / Communication
- Misc Modules
- Robotics
- Arduino-Related / Nucleo
- micro:bit
- MCU / ARM
- FPGA
- Sockets / Adapters

240x240, General 1.54inch LCD Display Module, IPS, 65K RGB

SKU: 18079  
Part Number: 1.54inch LCD Module  
Brand: Waveshare

\$9.49 1 ADD TO CART

\$9.09	10+
\$8.89	50+
\$8.81	100+

Related Products:

A summary of the product specifications is presented in a grid format. Each specification is accompanied by a representative icon: a diagonal line for size, a grid for resolution, a color wheel for display color, the text 'IPS' for the display panel, a connector pin header for the interface, and a microchip for the driver.

<b>Size</b> 1.54"	<b>Resolution</b> 240x240	<b>Display Color</b> 65K RGB
<b>Display Panel</b> IPS	<b>Interface</b> SPI	<b>Driver</b> ST7789



# SDK Tooling

- rPi Toolchain (PICO\_SDK)
  - Checkout eines Git repositories
  - Arm gcc (zB von arm, oder linux distro, Mac Homebrew)
  - Multiarch gdb
  - Arm binutils
- Cmake
- Controller als USB Storage uf2 Datei kopieren fertig



# Speichermanagement

- 256kB RAM
- ADC 12Bit (16Bit/ 2Byte) : Mindestens 250kHz Abtastrate
- => 500 kByte/sec 0.5 sec → Speicher voll  
Brauche den Speicher 2x → 0.2 sec



# st7789 Treiber

- Keinen fertigen Treiber gefunden
- Gestartet mit einem micropython SPI C Treiber  
[https://github.com/devbis/st7789\\_mpy/tree/master](https://github.com/devbis/st7789_mpy/tree/master)
- Dann micropython layer entfernt
- DMA Support hinzugefügt, vereinfacht



# SPI

- Serial Peripheral Interface
- Interface MCU <-> st7789
- Tut aus der Schachtel ([link](#))



# PIO Programmable I/O

- MCU Bit Bang Problemen: Ein Protokoll im korrekten Timing sprechen.
- Unter Linux ist das besonders schwer
- Wird aber zu einem Problem wenn man nebenbei noch was anderes tun muss
- Kleine Programme die Input und Output Übernehmen



# Microcontroller Tricks

- DMA für ADC (effektiv kann die MCU drauf warten)
- DMA, PIO für SPI0 ( [no-OS-FatFS-SD-SDIO-SPI-RPi-Pico](#) )
- DMA für SPI (st7789) (Bildschirm übertragen)
- Multicore: FFT

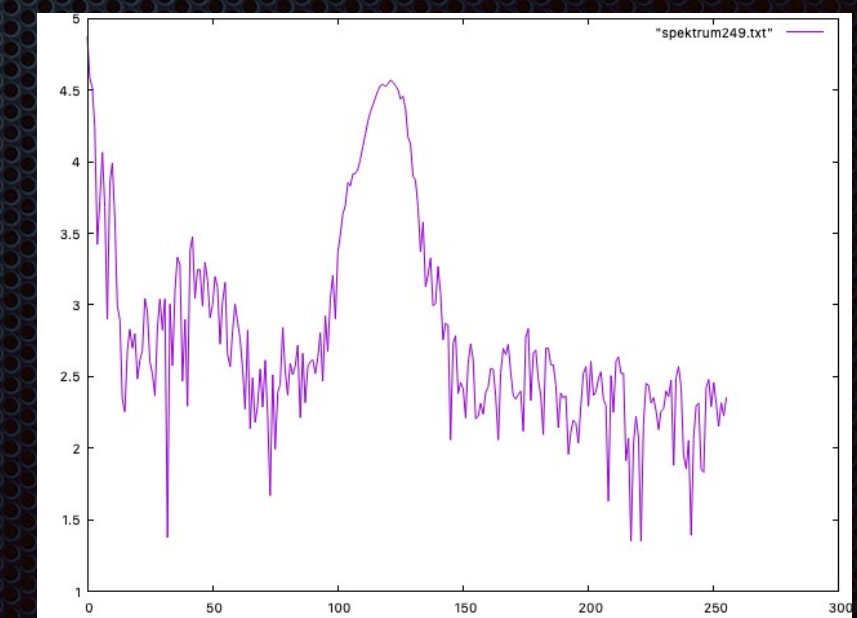
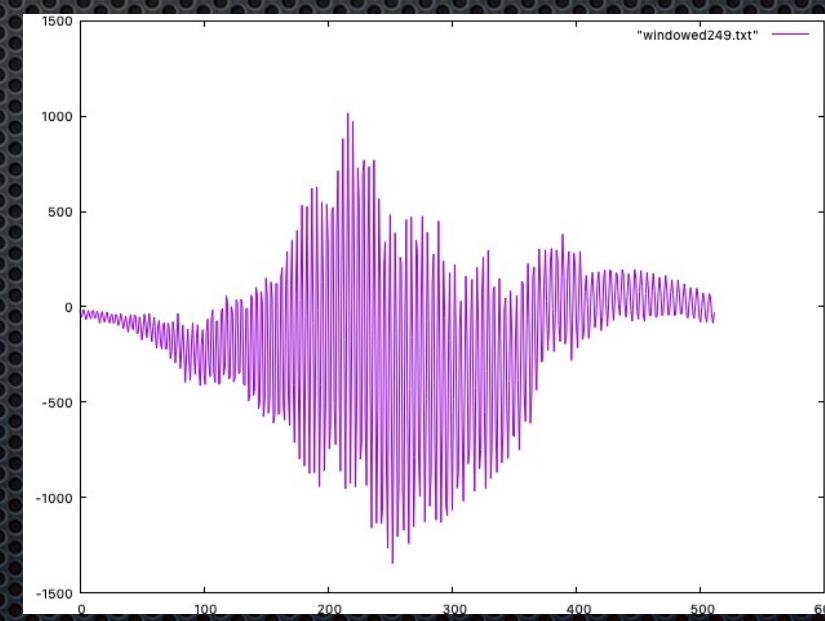
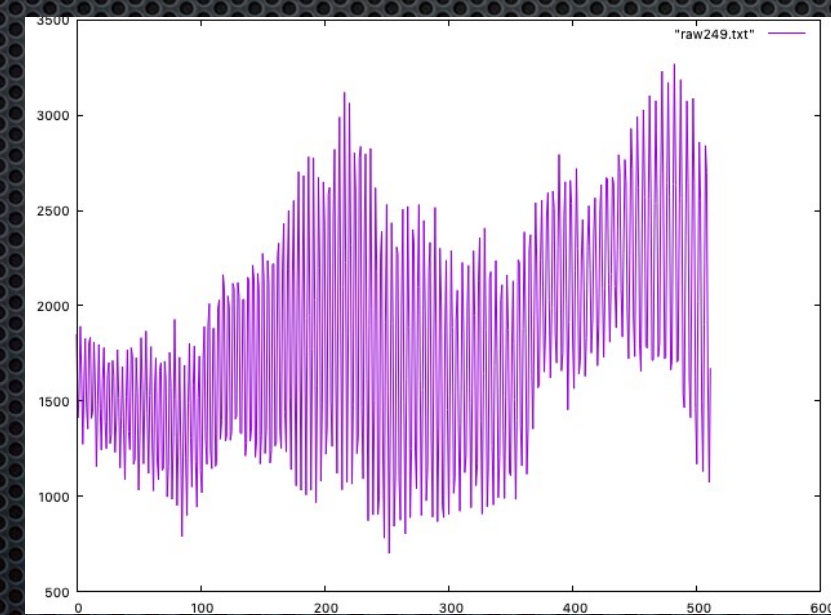
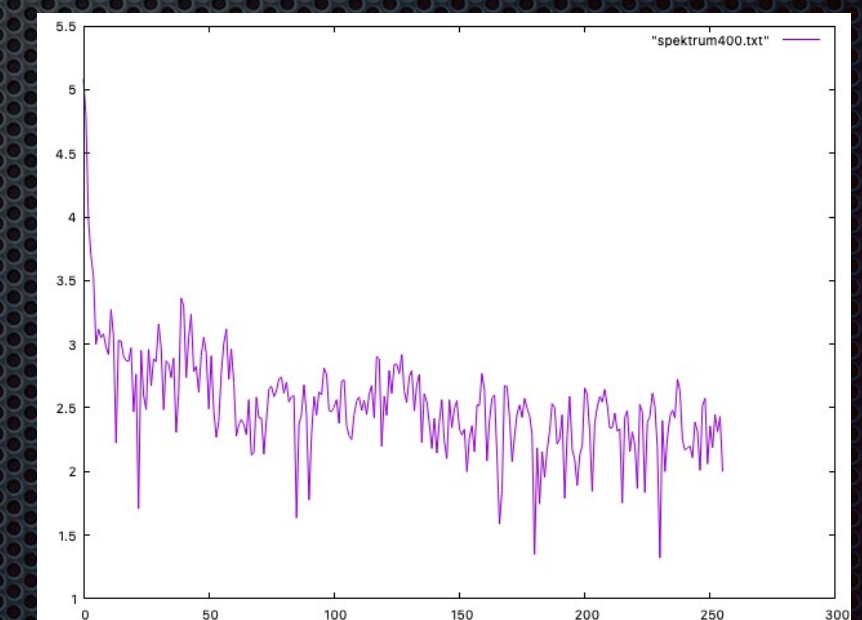
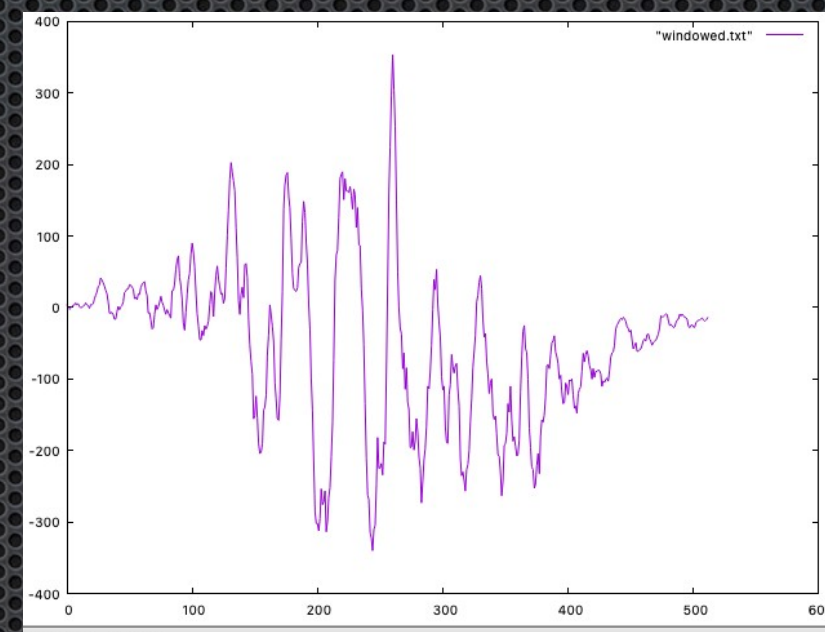
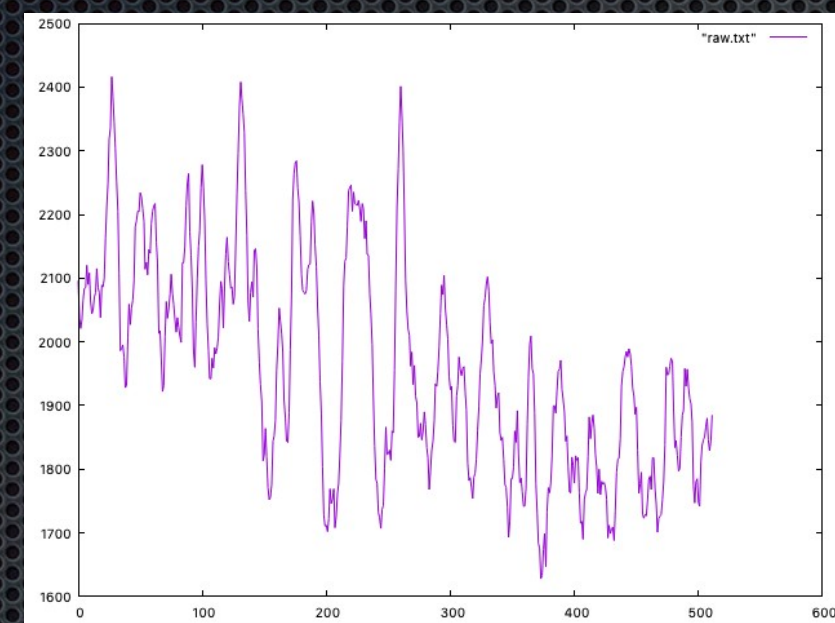
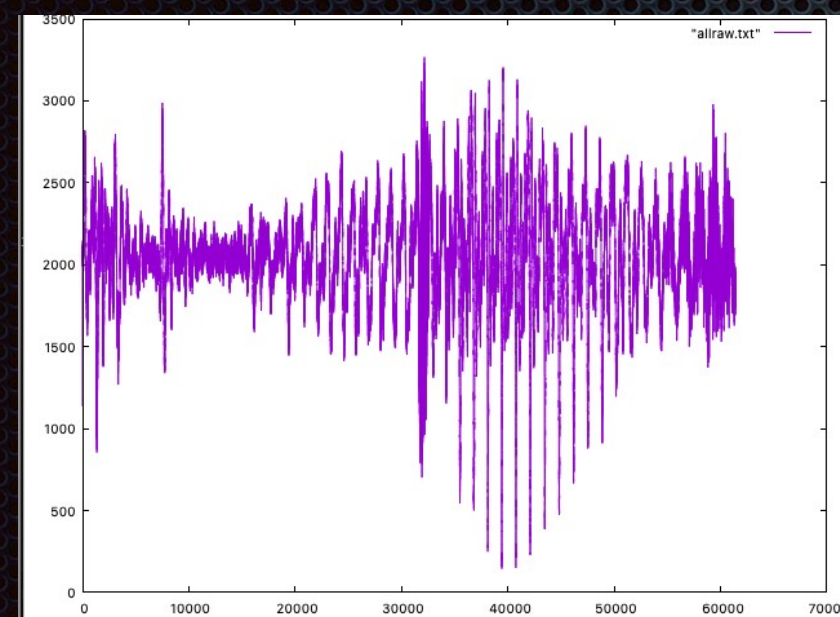


# Signalverarbeitung

- 12 Bit Daten (0 – 4096)
- 512 Werte auswählen
- 512 real FFT
- Hamming Window
- Spektrogramm
- Logarithmisch darstellen

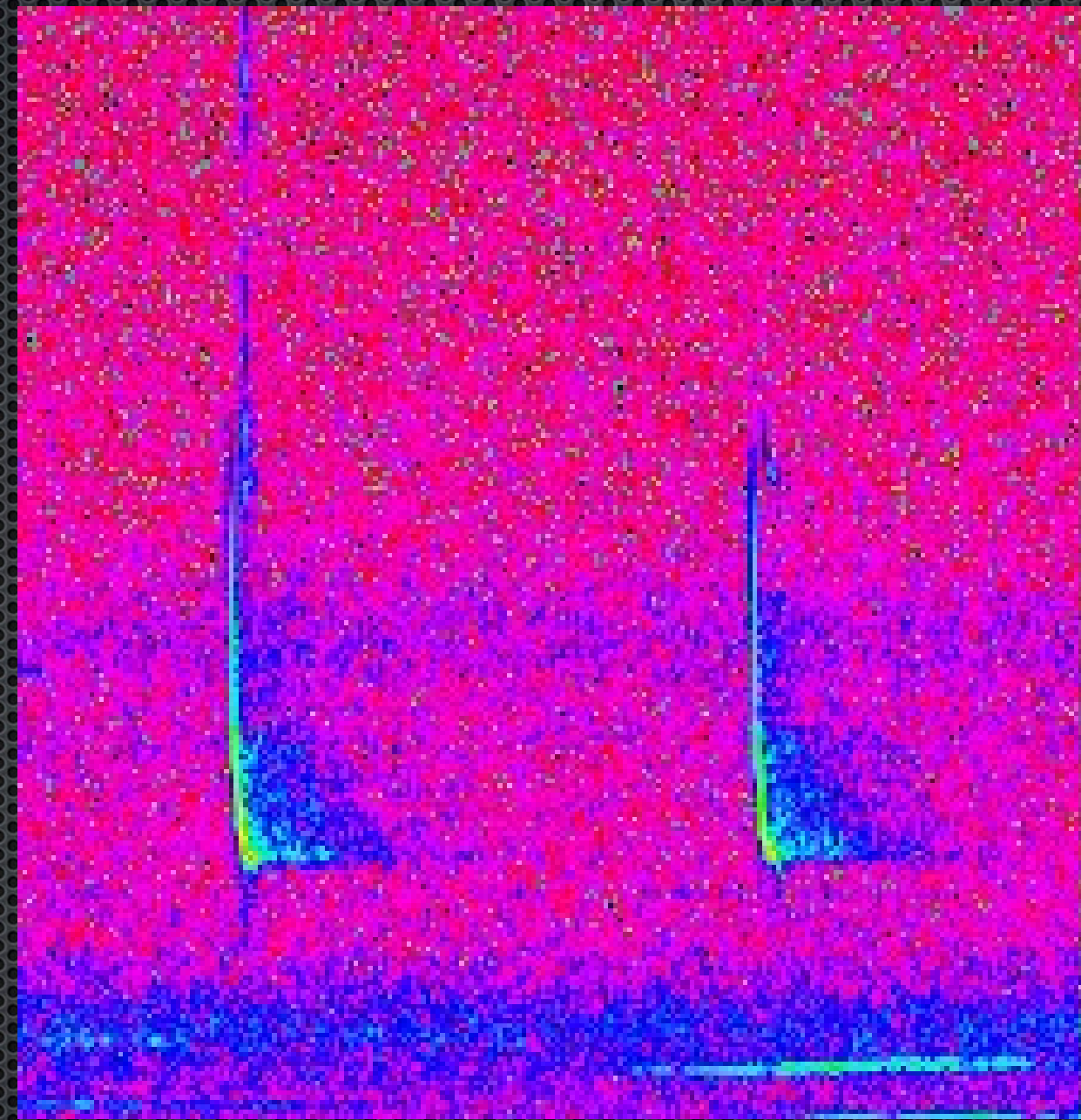


# Signalverarbeitung





HSL Kodiert  
(und C Programmierfehler am rechten Bildrand)





# K(r)ampf der Programmiersprachen

- Früher viel C,C++ privat und g'schäftlich
- Seit (ungefähr) 2000 python, C#, java, go, (rust)
- Memory safe Programmiersprachen Hype
- Was nehme ich hier nur ?
- Python: Kein st7789 treiber, bzw keiner der performant ist (auch wenn mein st7789 auf einem micropython Treiber aufbaut)



# C++ sucks

- Jeden Hype in Sprachstandard aufgenommen
  - Coroutinen, Concepts, Lambda, Template Meta programming, Attributes
- Aber Memory Safety trotzdem nicht implementiert  
(Herb Sutter / Stroustrup: Subset of Superset needed)
- Really?



## (embedded) Rust

- Eigentlich eine ganz nette Sprache, aber:
  - (im Januar 2024) DMA ist nicht standardisiert worden, habe SPI treiber nicht verstanden
  - Fehlende Treiber: Müsste die HW viel tiefer verstehen um hier weiterzukommen  
(next Rabbit Hole)



# Go/Tinygo

- Eigentlich eine ganz nette Sprache, aber:
  - Kein const ausser für Strings (Problem für Embedded, ROM Daten)
  - DMA support kaum vorhanden
  - Kein Multicore support
  - Escape Analysis muss sehr oft geholfen werden (ansonsten GC),
  - Attributierung der Funktionen nicht dokumentiert
  - Fehlende Treiber: Müsste die HW viel tiefer verstehen um hier weiterzukommen  
(next Rabbit Hole)



# C wiederentdeckt

- DMA ist halt unsafe
- Back to the roots feeling

Nice C11/C17

- Structs (zusammen mit compound literals)  
verhalten sich wie Typen als Automatic Variables (on Stack), call by value
- Restricted Pointer
- Arrays statt Pointer
- const
- Tagged Union / Sum Types (Anonymous structs)
- Complexe Arithmetik (Fortran FTW)
- Variable Length Arrays für Stack Allokationen (optionales Feature), als Funktionsparameter
- Flexible Array Members (letztes Feld in einem Struct )
- <bool.h>, <stdint.h>



# Jede Menge gute Literatur but YMMV

- Jens Gustedt Modern C (Das ist C???)
- <https://github.com/mcinglis/c-style/blob/master/readme.md>



# Tips:

- Cppcheck als Statische Typechecker
- Valgrind auf Linux für tests
- VS Code mit cmake
- VS Code mit Debugger
- Immer -Wall (warum nicht default?), pedantic mal anschauen
- Immer Array Notation statt Pointer verwenden
- VLA als Funktionsparameter, manueller Rangecheck nötig
- Typedef structs als Typen auf stack und call by value
- Const verwenden
- int16\_t usw verwenden, bool



# Erfahrungen

- Es ist wirklich grenzwertig:
  - Rangecheck mal übersehen (hier kann ja nichts passieren): Bäng (Bilder enthielten erratische Pixel)
  - Tippfehler: Falschen Typ beim sizeof Allozieren: Bäng
  - Es gibt debugger, aber warum braucht man sie?
  - Array Decay ist echt ein Problem
  - CMake kann bei Tippfehlern sehr ungnädig sein



# Timing des Codes

- Starte DMA Transfer vom ADC → Rohdaten
- Warte auf Ende
  - Starte FFT Thread mit Rohdaten
  - Rohdaten auf SD Karte schreiben
- Warte auf Ende FFT → Bilddaten
  - Starte DMA Transfer Bilddaten zum Display



# Credits

- <https://github.com/carlk3/no-OS-FatFS-SD-SDIO-SPI-RPi-Pico>  
Carl J Kugler III : SDIO SD Card Treiber
- <https://github.com/mborgerding/kissfft>  
Mark Borgerding Kiss: FFT
- [https://github.com/devbis/st7789\\_mpy](https://github.com/devbis/st7789_mpy)  
Ivan Belokobylskiy: st7789 micropython treiber
- <https://github.com/raspberrypi/pico-sdk>  
Raspberrypi (kilogramham)



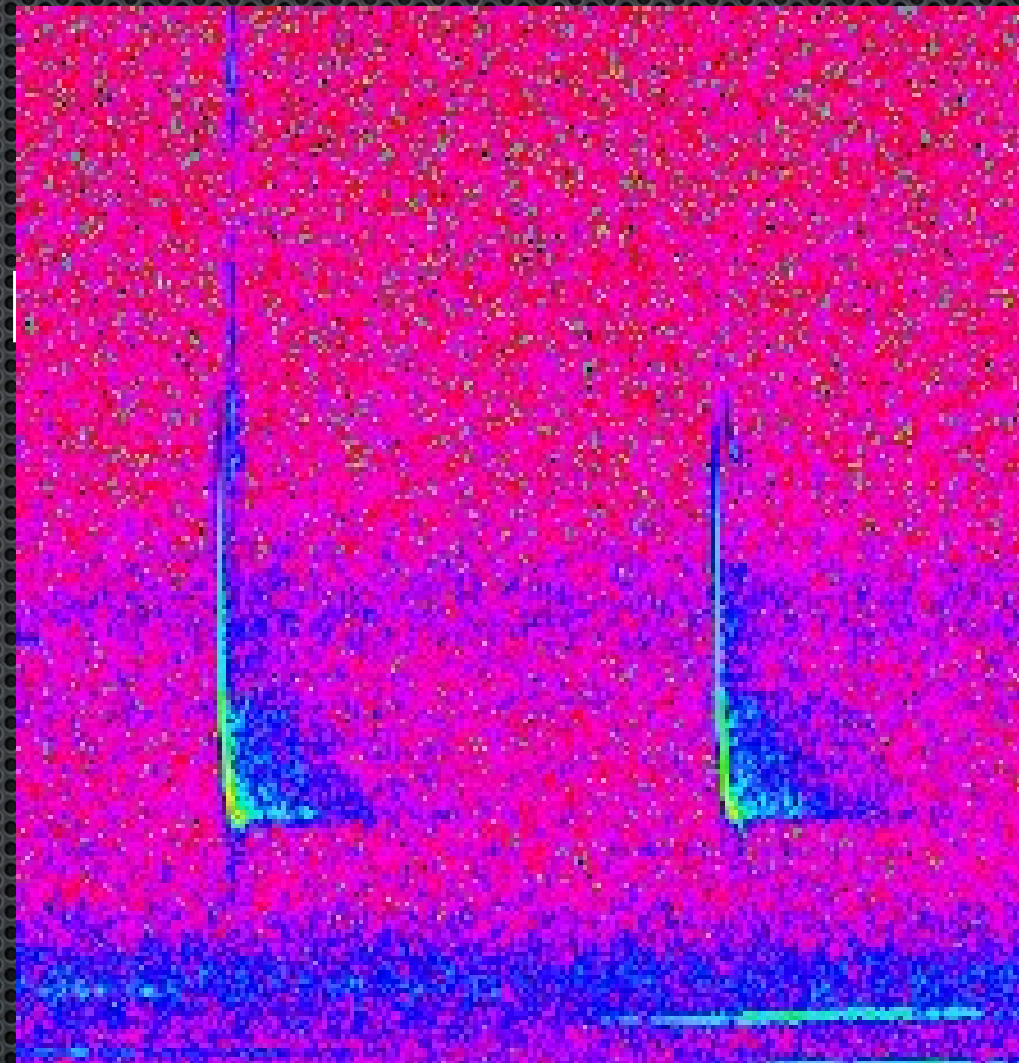
# Fledermäuse

- Soziallaute (Maus zu Maus)
- Orientierung (Maus zu Umwelt) sind FM signale (Sweeps)
- Bei mehreren Exemplaren weichen sie im Frequenzraum aus
- Jede Mausart hat sehr spezifische Ruf Eigenschaften
  - Dauer
  - Frequenzumfang
  - Art der Modulation
  - Sequenzen
  - Obertöne



# Zwergfledermaus

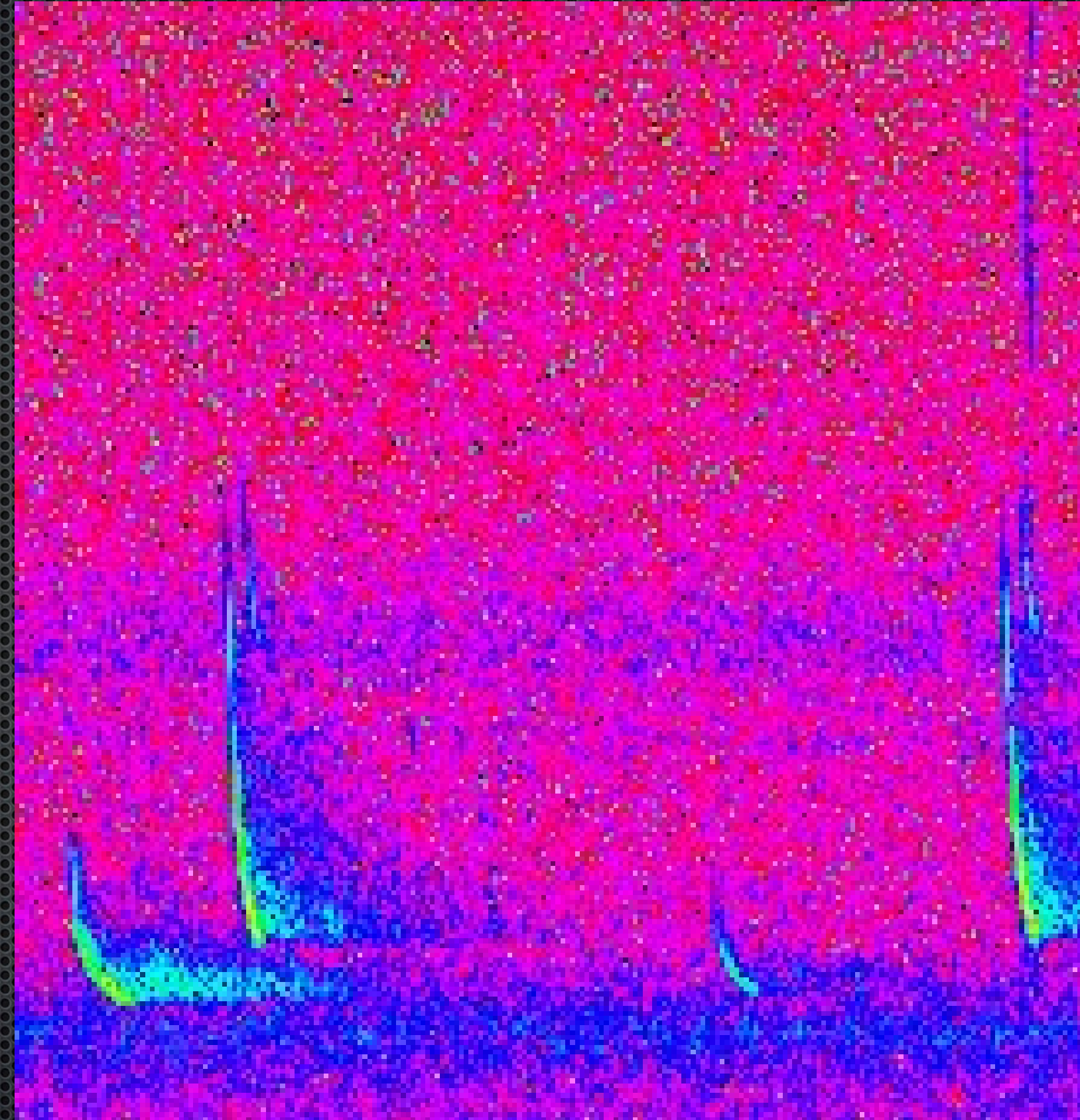
- (Oton man muss sich eher Sorgen machen wenn man diese NICHT findet)
- Schnelle Sequenzen von
- L Förmigen Spektren mit Haken
- Oberton sichtbar





# Ausweichen

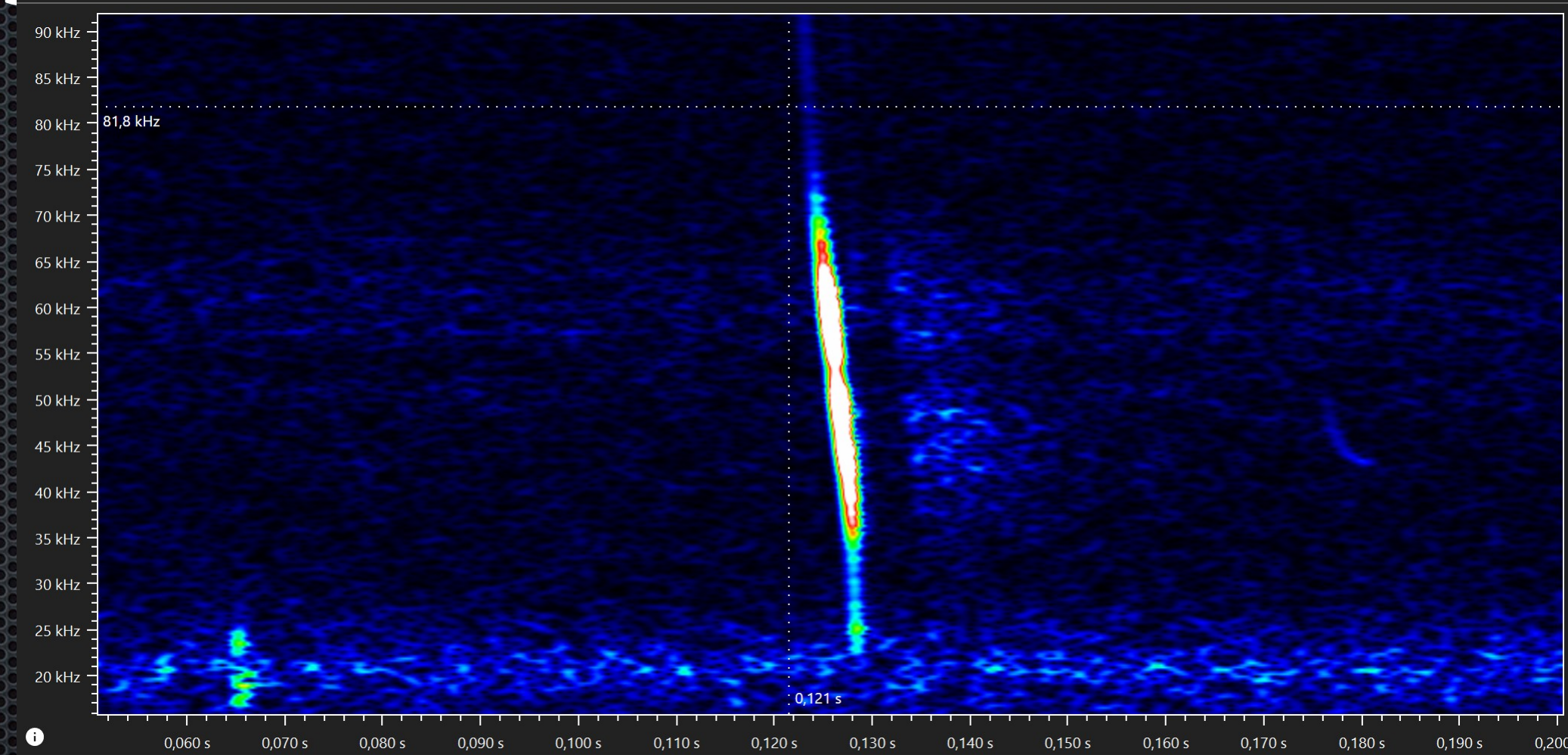
- Zwei Zwergfledermäuse (?)





# Myotis Myotis

- Grosses Mausohr
- Sehr kurzes Signal mit extremem Umfang 90kHz zu 25kHz in  $< 0.01$ s
- Und Optik (grosse Fledermaus!), Umgebung (kirchendachstuhl) passt





## Next Rabbit holes ahead

- Desktop Code zur Analyse : Programmiersprachenkrampf (again)  
–Jetzt mit GUI
- Neues Platinendesign: Eventuell SMD Platine?
- Gehäuse besser machen (Display mit gehäuse verschrauben)
- Automatische Rufanalyse statt manuell Bilder durchkämmen
- Eventuell bei Ruf Erkennt längere Aufnahme ohne Grafik ?
- Oder doch mit einem SBC (Mehr Speicher, aber ADC Timing komplex)...



# Fragen?

- <https://github.com/oflebbe/batdetect>
- <https://www.oflebbe.de>
- Fosstodon <https://fosstodon.org/@0x01af>

