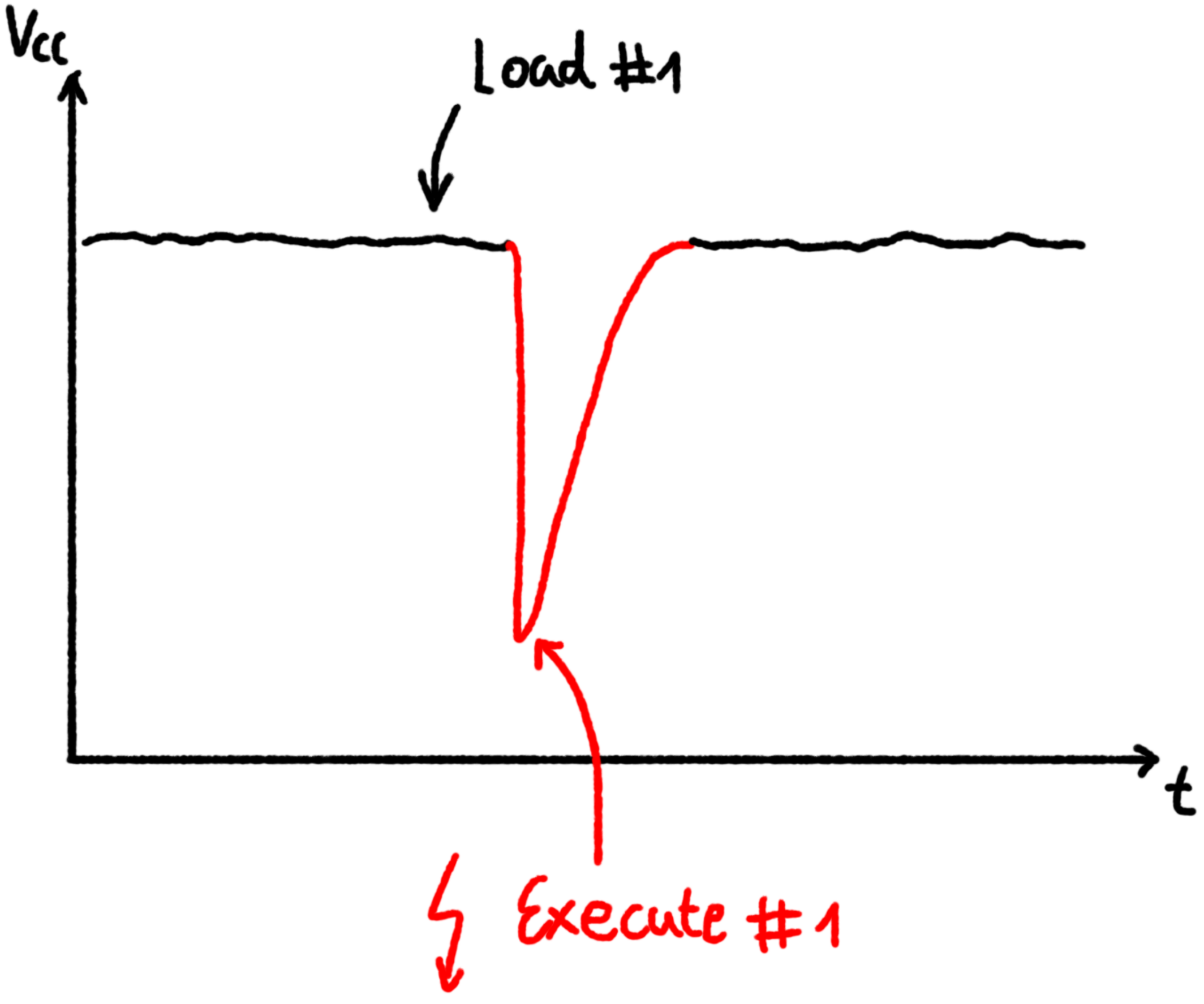
THE PENTEST EXPERTS.

YOUR SOC IMPLEMENTS A GLITCH DETECTOR? HOLD MY BEER!

Attacking the STM32L051 und nRF54L15




Vcc
Load #1
t
Execute #1

Home

Where to get the Pico Glitcher from?

Links and further resources

Overview

Getting started

Examples

Command line tools

Customize your Pico Glitcher

Comparing different glitching modes

Multiplexing with the Pico Glitcher v2

Pulse Shaping with the Pico Glitcher v2 and extension board

Schmitt Trigger inputs of the Pico Glitcher v2

Genetic algorithm

Using the analog digital converter

Changelog and hardware versions

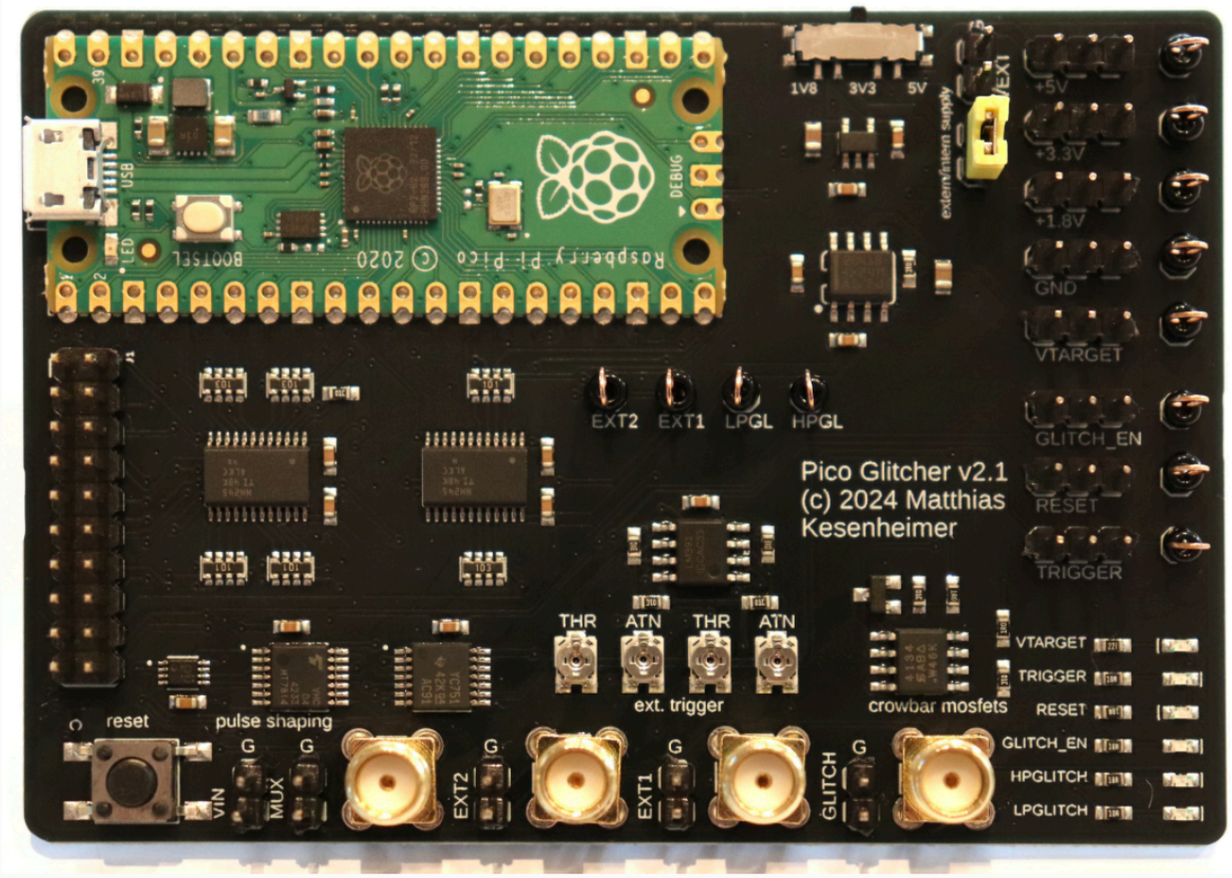
Findus

PicoGlitcher

Welcome to findus (aka fault-injection-library)

This library is intended to make fault injection attacks against microcontrollers accessible for hobbyists and to introduce the topic of voltage glitching. Findus offers an easy entry point to carry out your own attacks against microcontrollers, SoCs and CPUs. With the provided and easy to use functions and classes, fault injection projects can be realized quickly.

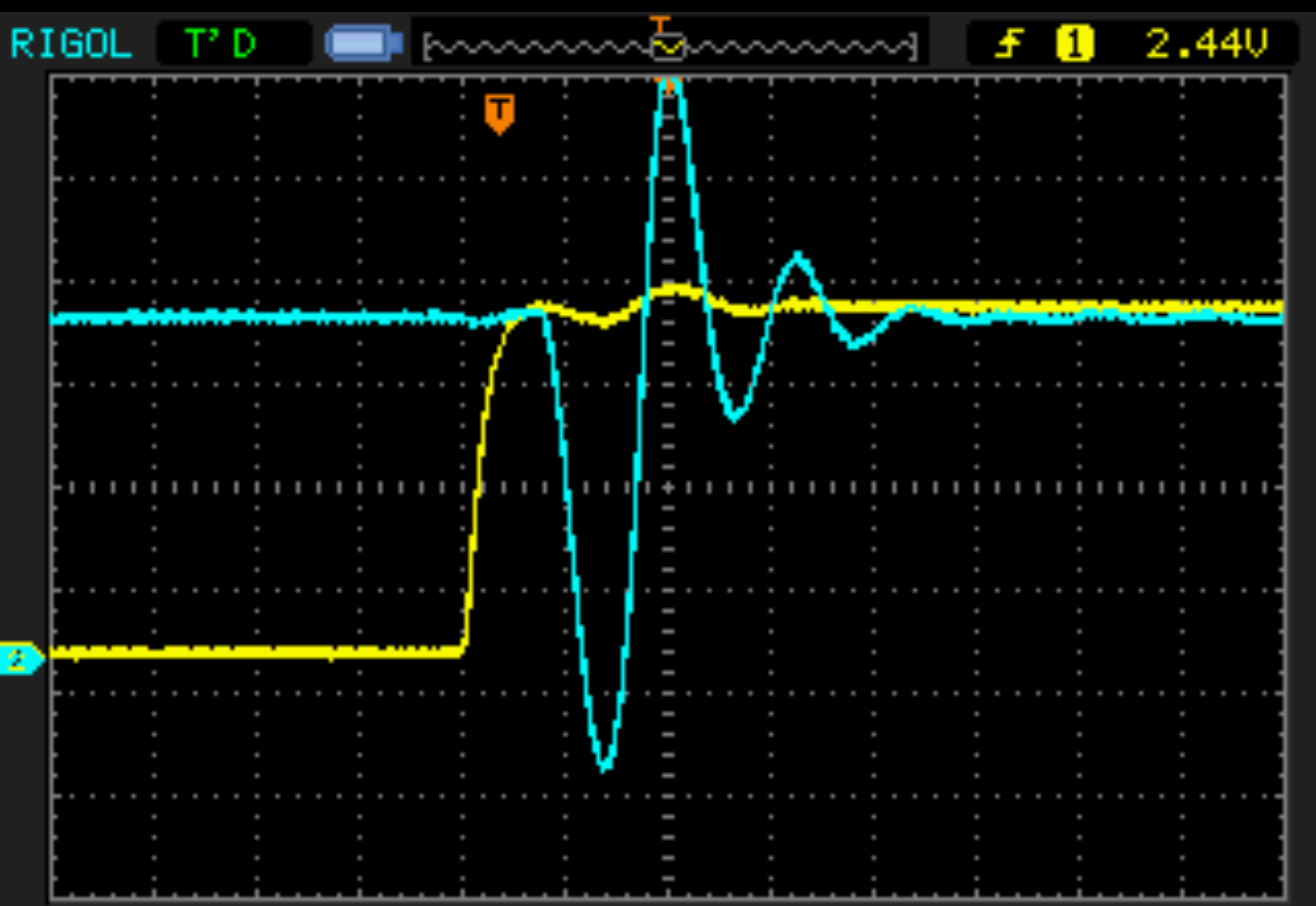
The fault-injection-library has been built to work with hardware that consists of a common MOSFET, the Raspberry Pi Pico as the controller and a few other cheap components. The Raspberry Pi Pico is not only cheap and available for the hobbyist, but also a very capable microcontroller. Furthermore, this library supports the ChipWhisperer Pro and the ChipWhisperer Husky as well.



The Pico Glitcher was developed to minimize the effort and requirements for the hardware setup. The Pico Glitcher is optimized for use with findus and makes it particularly easy to carry out fault injection attacks.

https://www.tindie.com/products/faulty-hardware/picoglitcher-v2/

https://fault-injection-library.readthedocs.io/en/latest/

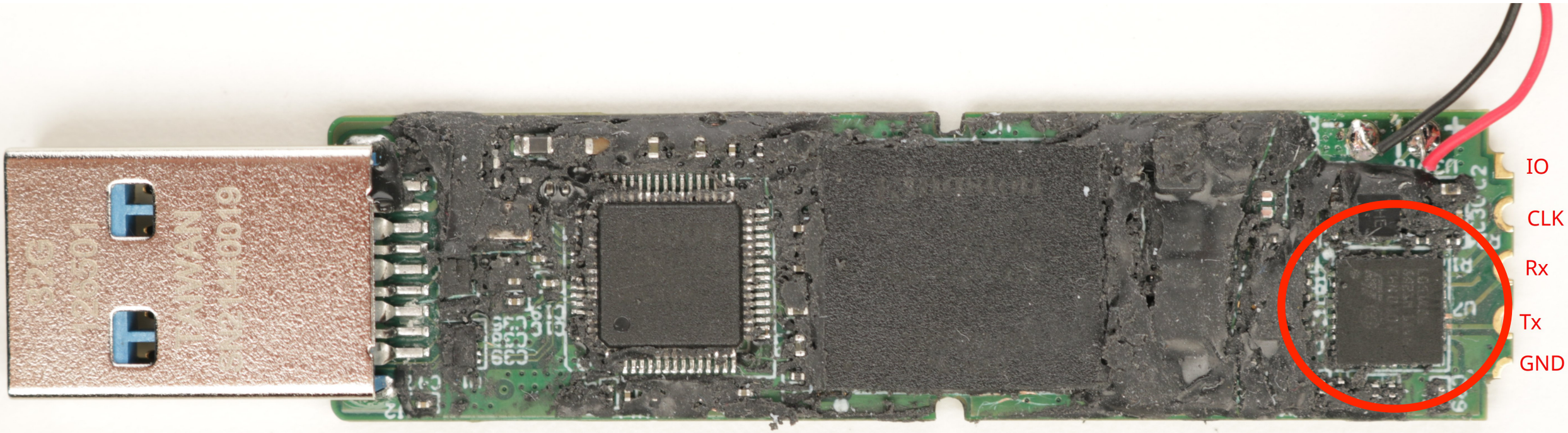

RIGOL
T' D
2.44V
CH1 1.00V
CH2 1.00V
Time 100.0ns
164.0ns

FLASH ERASE SUPPRESSION ATTACK

Attacking the STM32L051's read-out protection

https://blog.syss.com/posts/voltage-glitching-the-stm32l05-microcontroller/

https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2025-033.txt


IO
CLK
Rx
Tx
GND

→ RDP 0:

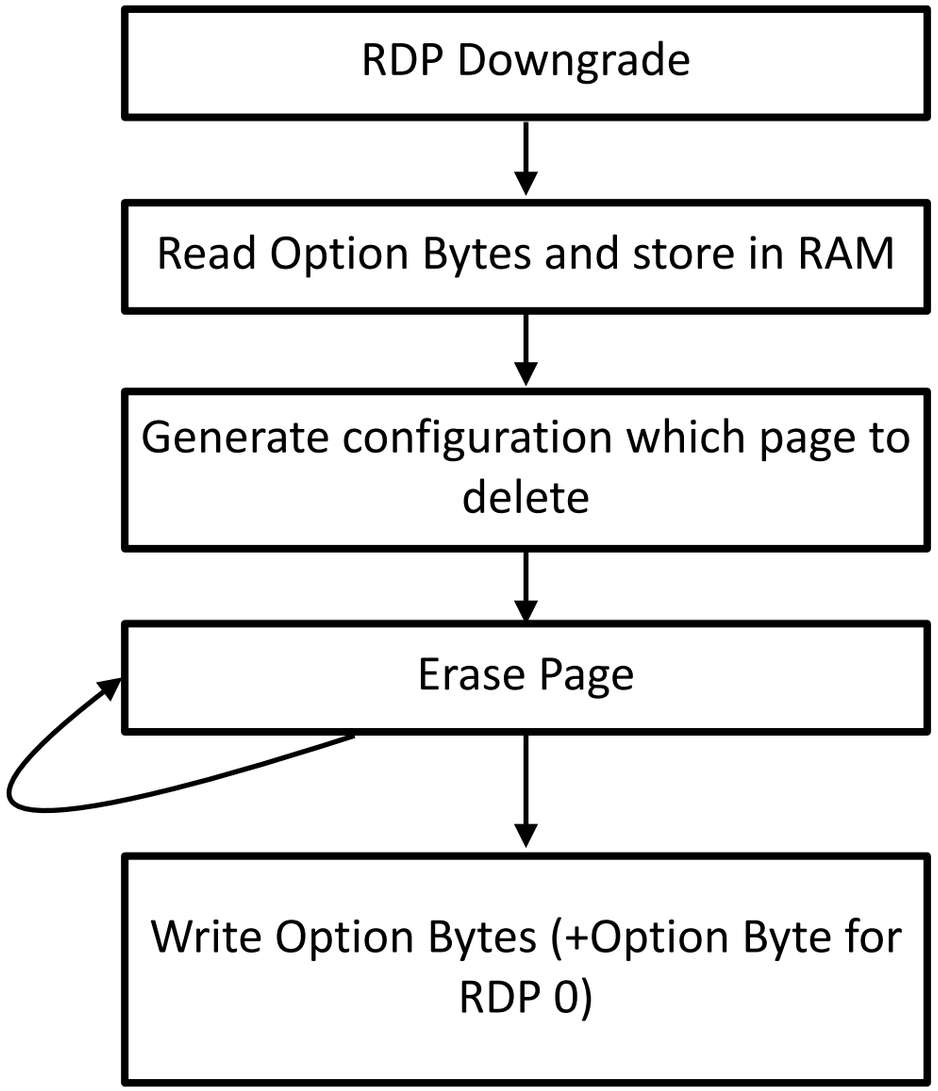
- Full debugging is allowed
- Full access while in system bootloader

→ RDP 1:

- Access (read, erase, program) to Flash memory is forbidden while debugging
- Other memories such as the bootloader memory can be accessed via JTAG
- Access (read, erase, program) to any memory is forbidden while in system bootloader
- Everything else can be read, e.g. RAM and registers
- Reset to RDP level 0 erases whole Flash memory

→ RDP 2:

- Debugging is forbidden by disabling the JTAG and SWD interfaces
- Starting system bootloader is forbidden
- Reset to RDP Level 0/1 is not possible


RDP Downgrade
Read Option Bytes and store in RAM
Generate configuration which page to delete
Erase Page
Write Option Bytes (+Option Byte for RDP 0)

RIGOL STOP H 50.00us 500.0MSa/s 7.00M pts D 318.000000us T 2 960mV

HORIZONTAL

TRIGGER

A1

A2

A3

A4

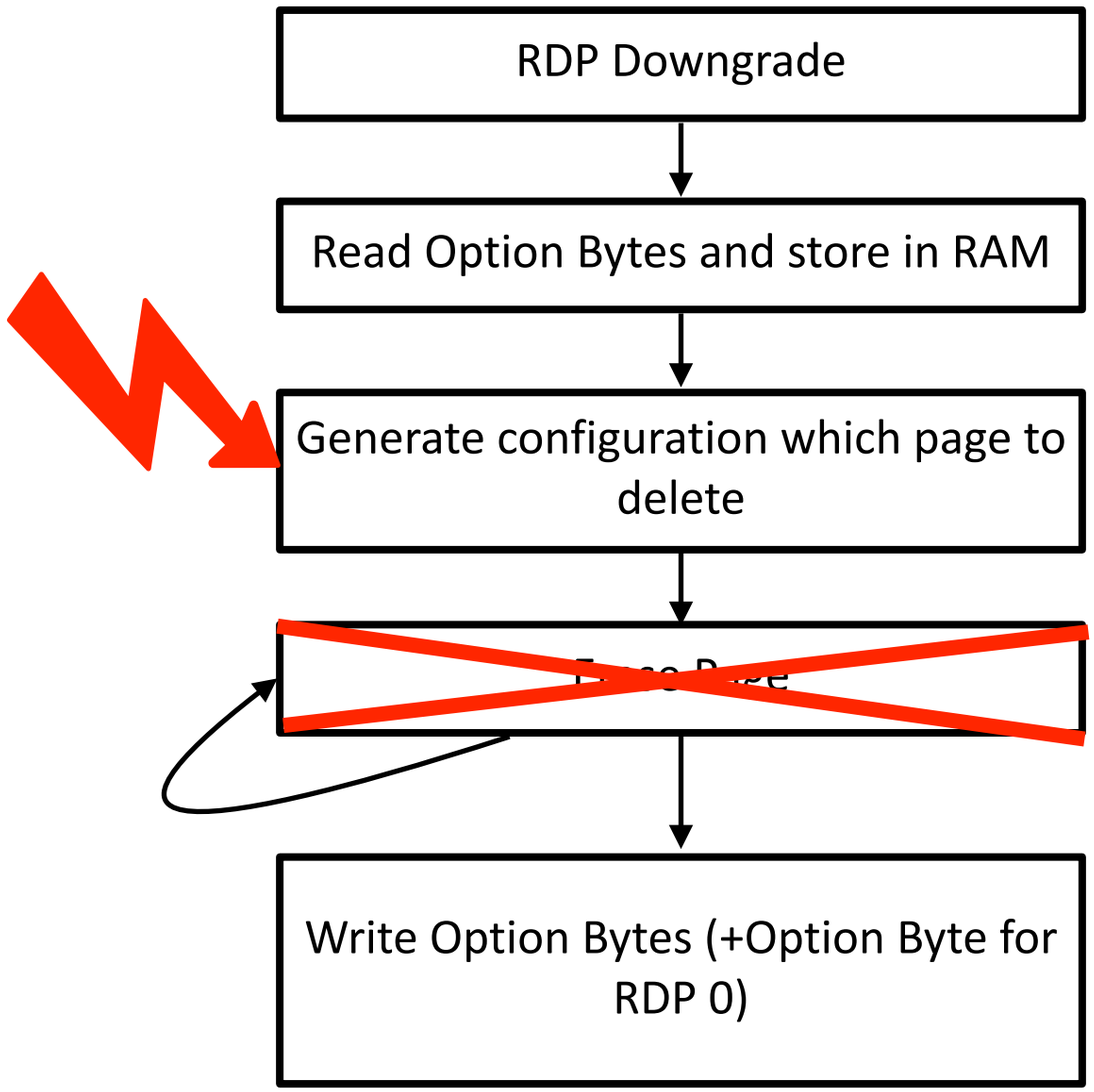
A5

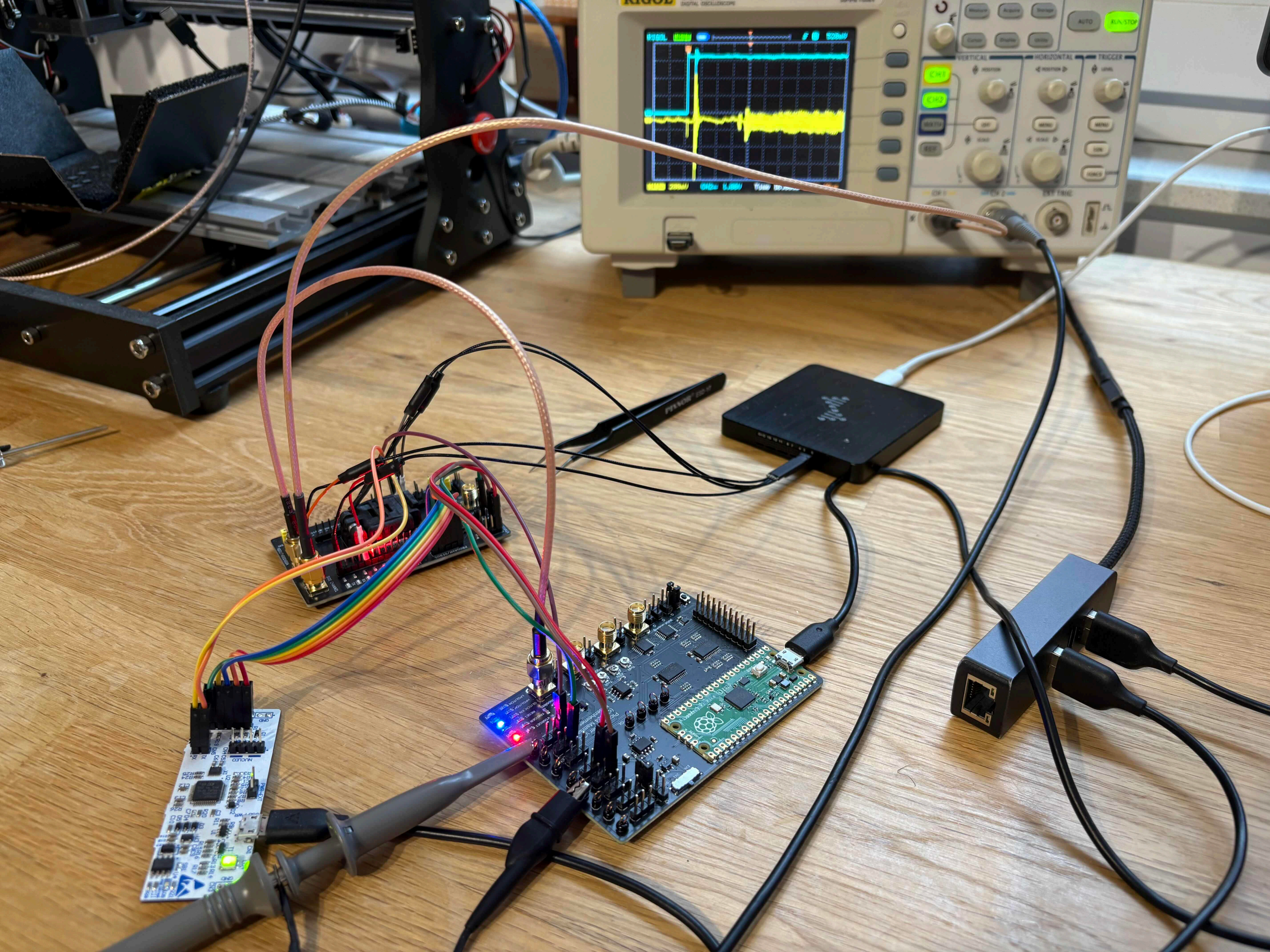
x 32

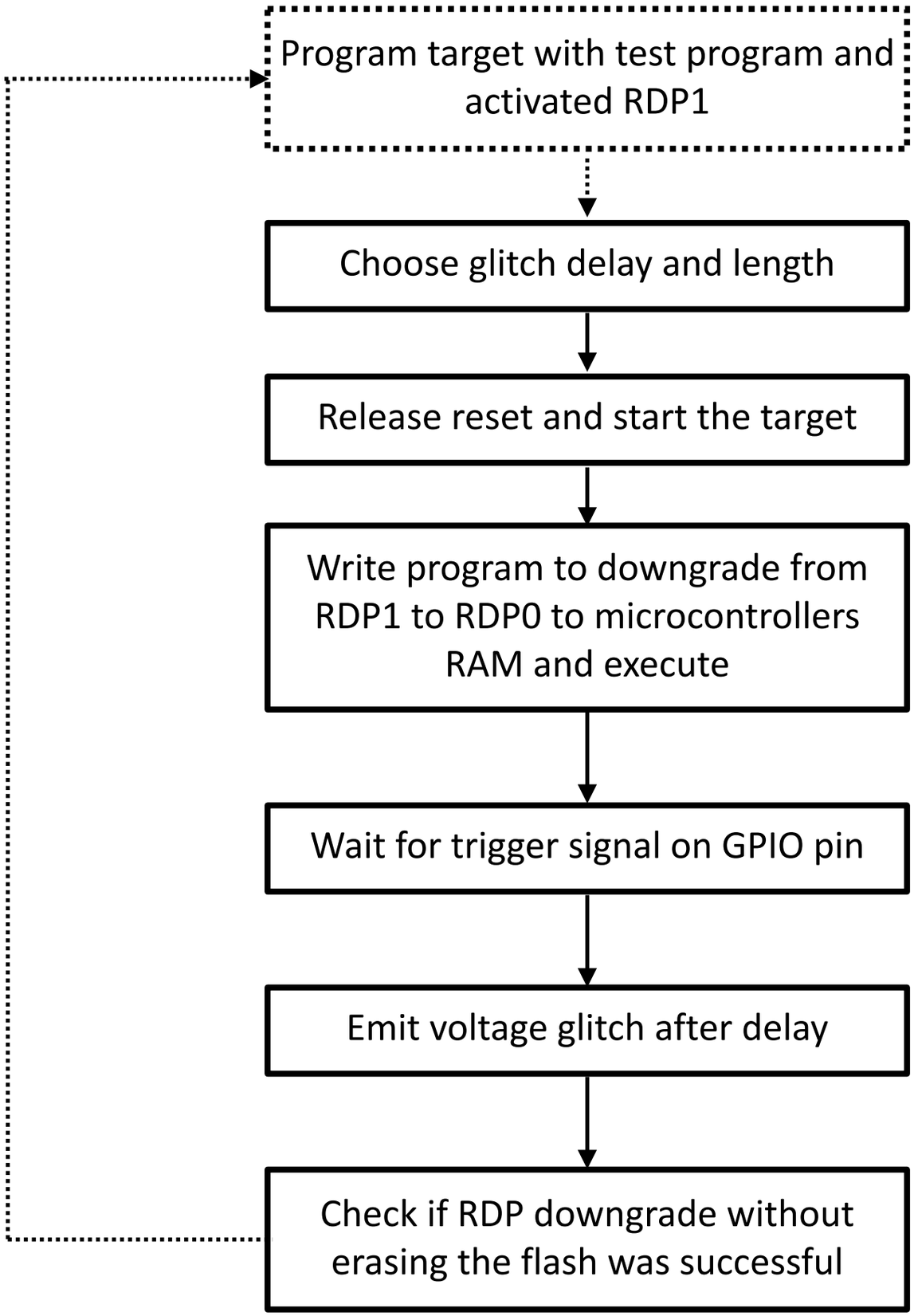
Rise=112.5us

1 ~ 100mV 2 = 1.00 V

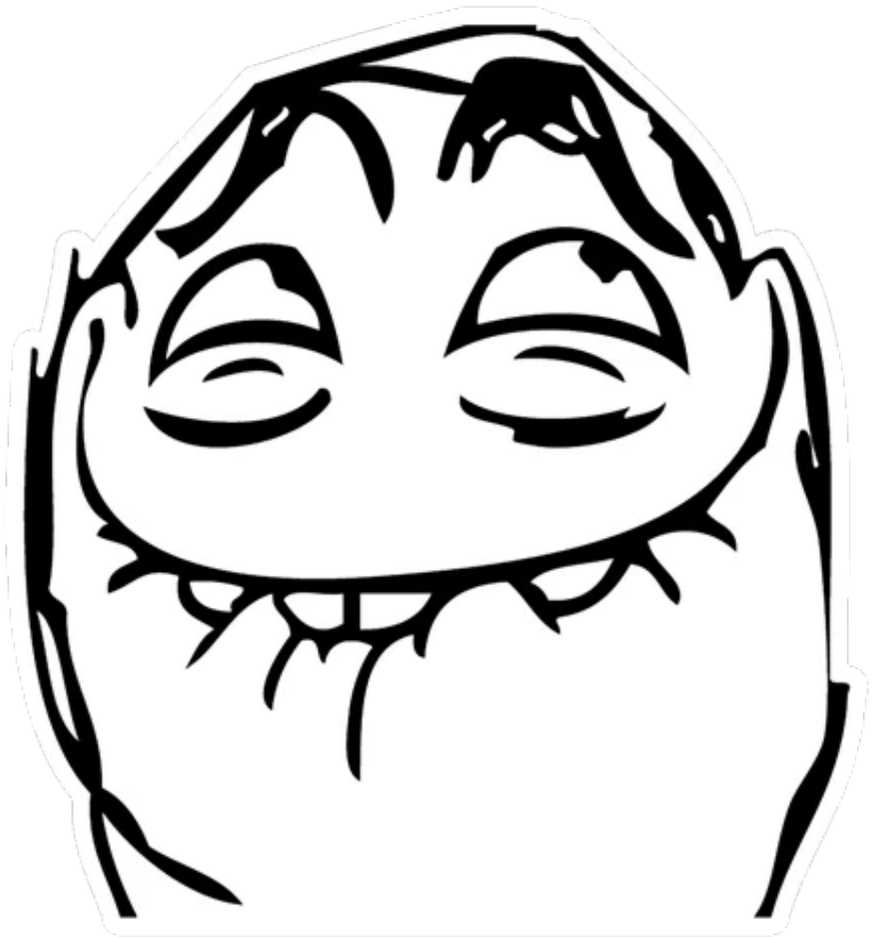
15:14


RDP Downgrade
Read Option Bytes and store in RAM
Generate configuration which page to delete
Write Option Bytes (+Option Byte for RDP 0)


RIGOL
DIGITAL OSCILLOSCOPE
AUTO
RUN/STOP
VERTICAL
HORIZONTAL
TRIGGER
CH1
CH2
MATH
REF
MENU
raspberry pi


Program target with test program and activated RDP1
Choose glitch delay and length
Release reset and start the target
Write program to downgrade from RDP1 to RDP0 to microcontrollers RAM and execute
Wait for trigger signal on GPIO pin
Emit voltage glitch after delay
Check if RDP downgrade without erasing the flash was successful

LIVE HACK



python stm32l0-rd

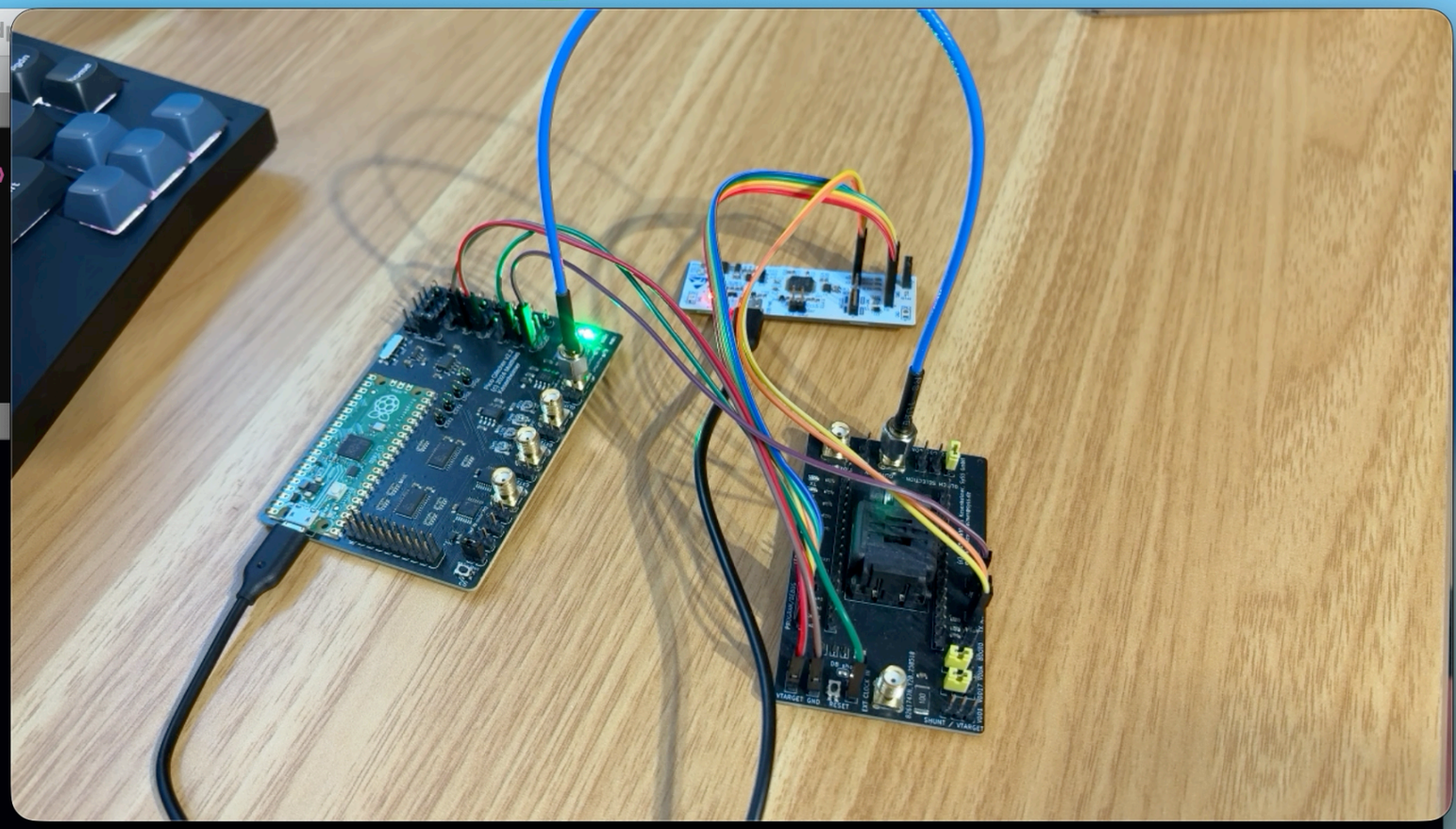
python (Python) ⌘1 ..m8-bootloader (-zsh) ⌘2

..ndus/firmware (-zsh)

```
Last login: Tue Sep 30 02:03:58 on console
~/Desktop/fault-injection/fault-injection-library/findus/firmware master* )
```

python (Python)

```
00000230  ae 40 26 43 02 9c 26 62 01 e0 03 2a df d1 ba 40
00000240  04 68 01 9d 25 40 2a 43 c0 24 02 60 62 46 a4 02
00000250  22 42 4f d0 01 25 28 4c a0 27 62 6b ff 05 2a 43
00000260  62 63 26 4a 9c 08 a4 00 a4 18 0f 22 02 35 1d 40
00000270  ad 00 aa 40 a6 68 96 43 00 22 b8 42 10 d0 20 4f
00000280  01 32 b8 42 0c d0 1f 4f 01 32 b8 42 08 d0 1e 4f
00000290  01 32 b8 42 04 d0 1d 4a 82 18 57 1e ba 41 05 32
000002a0  aa 40 67 46 32 43 a2 60 19 4a 00 9c 96 68 00 9d
000002b0  e4 43 35 43 ff 02 01 d4 35 00 25 40 67 46 95 60
000002c0  d6 68 00 9d 35 43 bf 02 01 d4 35 00 25 40 67 46
000002d0  d5 60 56 68 00 9d 35 43 bf 03 01 d4 35 00 25 40
000002e0  67 46 55 60 15 68 00 9e 2e 43 ff 03 01 d4 25 40
000002f0  2e 00 16 60 01 33 56 e7 00 10 02 40 00 00 01 40
00000300  00 04 00 50 00 08 00 50 00 0c 00 50 00 e4 ff af
00000310  00 04 01 40 00 2a 01 d0 81 61 70 47 81 62 fc e7
00000320  0d 48 85 46 ff f7 fb fe 0c 48 0d 49 0d 4a 00 23
00000330  02 e0 d4 58 c4 50 04 33 c4 18 8c 42 f9 d3 0a 4a
00000340  0a 4c 00 23 01 e0 13 60 04 32 a2 42 fb d3 00 f0
00000350  11 f8 ff f7 d9 fe fe e7 00 20 00 20 00 00 00 20
00000360  00 00 00 20 dc 03 00 08 00 00 00 20 1c 00 00 20
00000370  fe e7 00 00 70 b5 00 26 0c 4c 0d 4d 64 1b a4 10
00000380  a6 42 09 d1 00 26 00 f0 19 f8 0a 4c 0a 4d 64 1b
00000390  a4 10 a6 42 05 d1 70 bd b3 00 eb 58 98 47 01 36
000003a0  ee e7 b3 00 eb 58 98 47 01 36 f2 e7 d4 03 00 08
000003b0  d4 03 00 08 d8 03 00 08 d4 03 00 08 f8 b5 c0 46
000003c0  f8 bc 08 bc 9e 46 70 47 f8 b5 c0 46 f8 bc 08 bc
000003d0  9e 46 70 47 e9 00 00 08 c1 00 00 08 00 00 00 00
000003e0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003f0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[+] Terminating gracefully.
[+] Detaching debugger.
~/Desktop/fault-injection/fault-injection-library/projects/stm32l05x master* ) python stm32l0-rdp-downgrade.py --rpico /dev/tty.usbmodem21401 --delay 0 12_000 --length 142 154 --delay2 7
_500_000 8_000_000 --length2 1_000 1_050 --program-target 1 --test --demo
[+] Version of Pico Glitcher: [1, 11, 0]
[+] Version of findus: [1, 11, 2]
[+] Programming target.
```

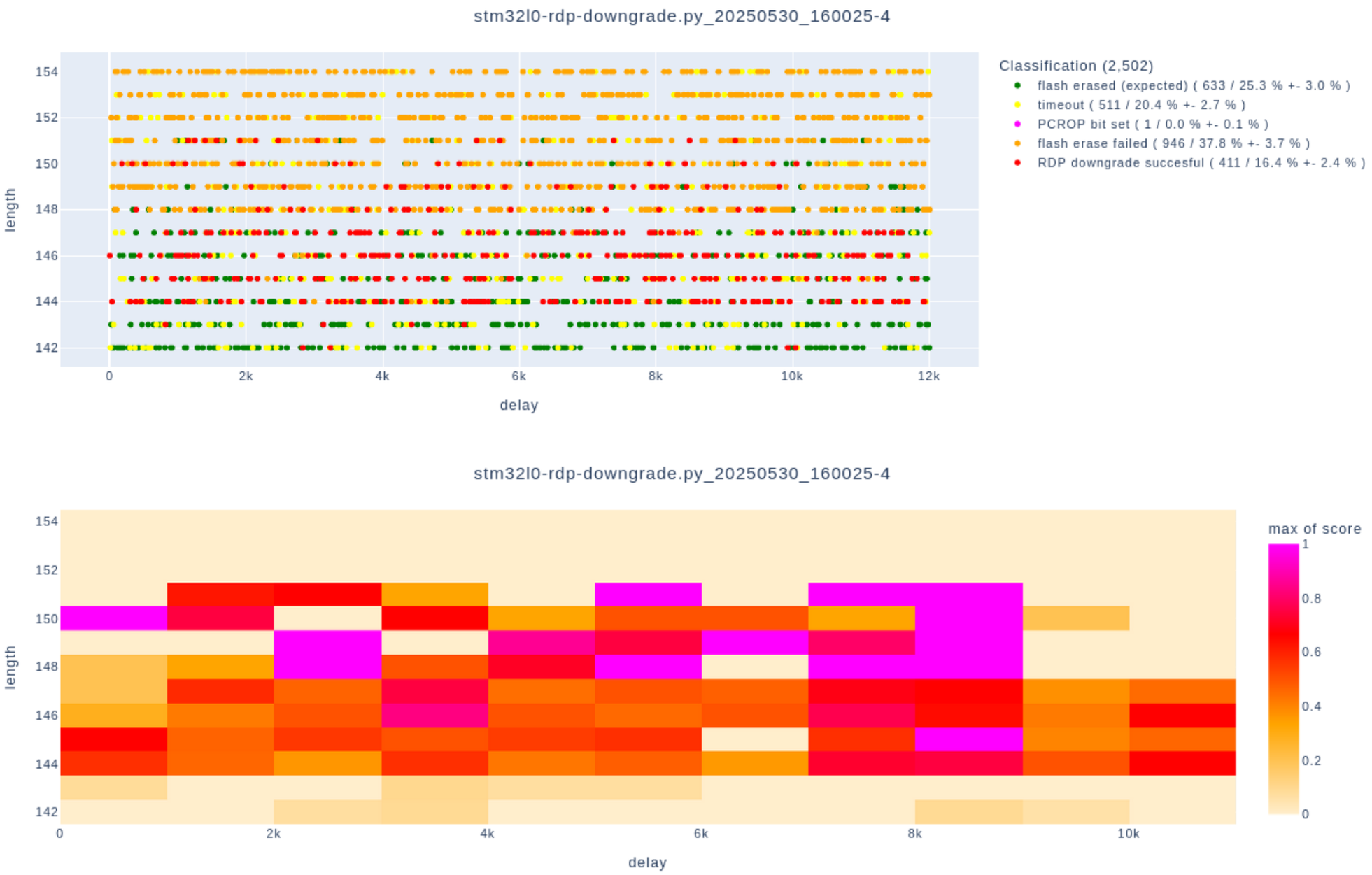


```
[+] Result whether programming was successful: ok
[+] Programming finished.
[+] Programming target with program to downgrade to RDP 0 (to RAM).
[+] Content at 0x08000000: 0x0
[+] Option bytes: rdp = 0xaa, pcrop = 0x0
[+] Experiment 50       0       (0)     3365    142     7863299 1027    G       b'expected: read empty flash'
[+] Preparing target.
[+] Option bytes: rdp = 0xaa, pcrop = 0x0
[+] Warning: rdp or pcrop not as expected. Programming target with test program (to flash) and enabling rdp level 1.
[+] Result whether programming was successful: ok
[+] Programming finished.
[+] Programming target with program to downgrade to RDP 0 (to RAM).
[+] Content at 0x08000000: 0x20002000
[+] Option bytes: rdp = 0xaa, pcrop = 0x0
[+] Experiment 51       0       (0)     8055    146     7758546 1005    R       b'success: RDP inactive'
[+] RDP downgraded successfully. Stop.
00000000  00 20 00 20 21 03 00 08 71 03 00 08 71 03 00 08
00000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000020  00 00 00 00 00 00 00 00 00 00 00 00 71 03 00 08
00000030  00 00 00 00 00 00 00 00 71 03 00 08 71 03 00 08
00000040  71 03 00 08 71 03 00 08 71 03 00 08 71 03 00 08
00000050  71 03 00 08 71 03 00 08 71 03 00 08 71 03 00 08
00000060  00 00 00 00 71 03 00 08 71 03 00 08 71 03 00 08
00000070  71 03 00 08 71 03 00 08 00 00 00 00 71 03 00 08
00000080  00 00 00 00 71 03 00 08 00 00 00 00 00 00 00 00
00000090  71 03 00 08 00 00 00 00 71 03 00 08 71 03 00 08
000000a0  71 03 00 08 71 03 00 08 71 03 00 08 71 03 00 08
000000b0  71 03 00 08 71 03 00 08 00 00 00 00 00 00 00 00
000000c0  10 b5 06 4c 23 78 00 2b 07 d1 05 4b 00 2b 02 d0
000000d0  04 48 00 e0 00 bf 01 23 23 70 10 bd 00 00 00 20
```

FAULT INJECTION ANALYSIS

Update | SELECT * FROM experiments WHERE

stm32l0-rdp-downgrade.py_20250530_160025-4.sqlite (2502)



Files

master

Go to file

stm32f40x
stm32f412
stm32f42x
stm32l05x
rdp-downgrade-stm32l051.elf
stm32l0-rdp-downgrade.py
toggle-led-stm32l051.elf

fault-injection-library / projects / stm32l05x /

MKesenheimer removed test elf ac8a677 · 3 months ago History

Name	Last commit message	Last commit date
..		
rdp-downgrade-stm32l051.elf	removed test elf	3 months ago
stm32l0-rdp-downgrade.py	removed test elf	3 months ago
toggle-led-stm32l051.elf	removed test elf	3 months ago

https://github.com/MKesenheimer/fault-injection-library/tree/master/projects/stm32l05x

Attacking the STM32L051’s read-out protection

https://blog.syss.com/posts/voltage-glitching-the-stm32l05-microcontroller/

https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2025-033.txt

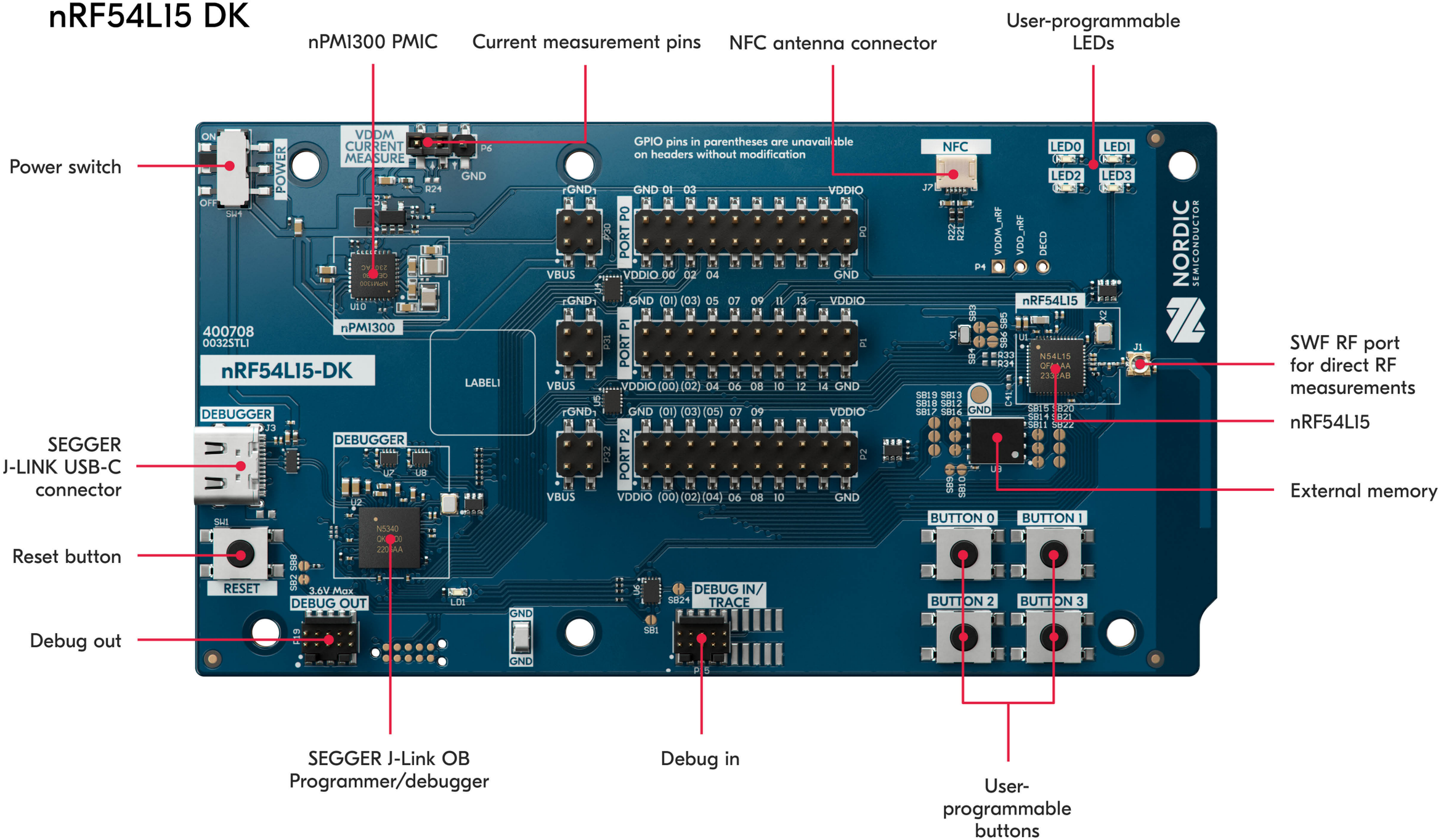
ELECTROMAGNETIC FAULT INJECTION (EMFI)

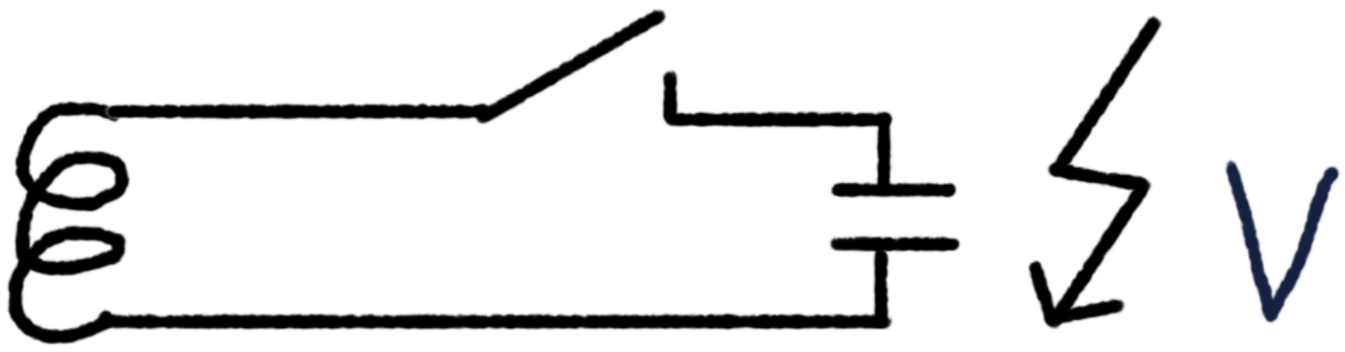
Attacking the nRF54L15 with photon rays

https://blog.syss.com/posts/nrf54-emfi/

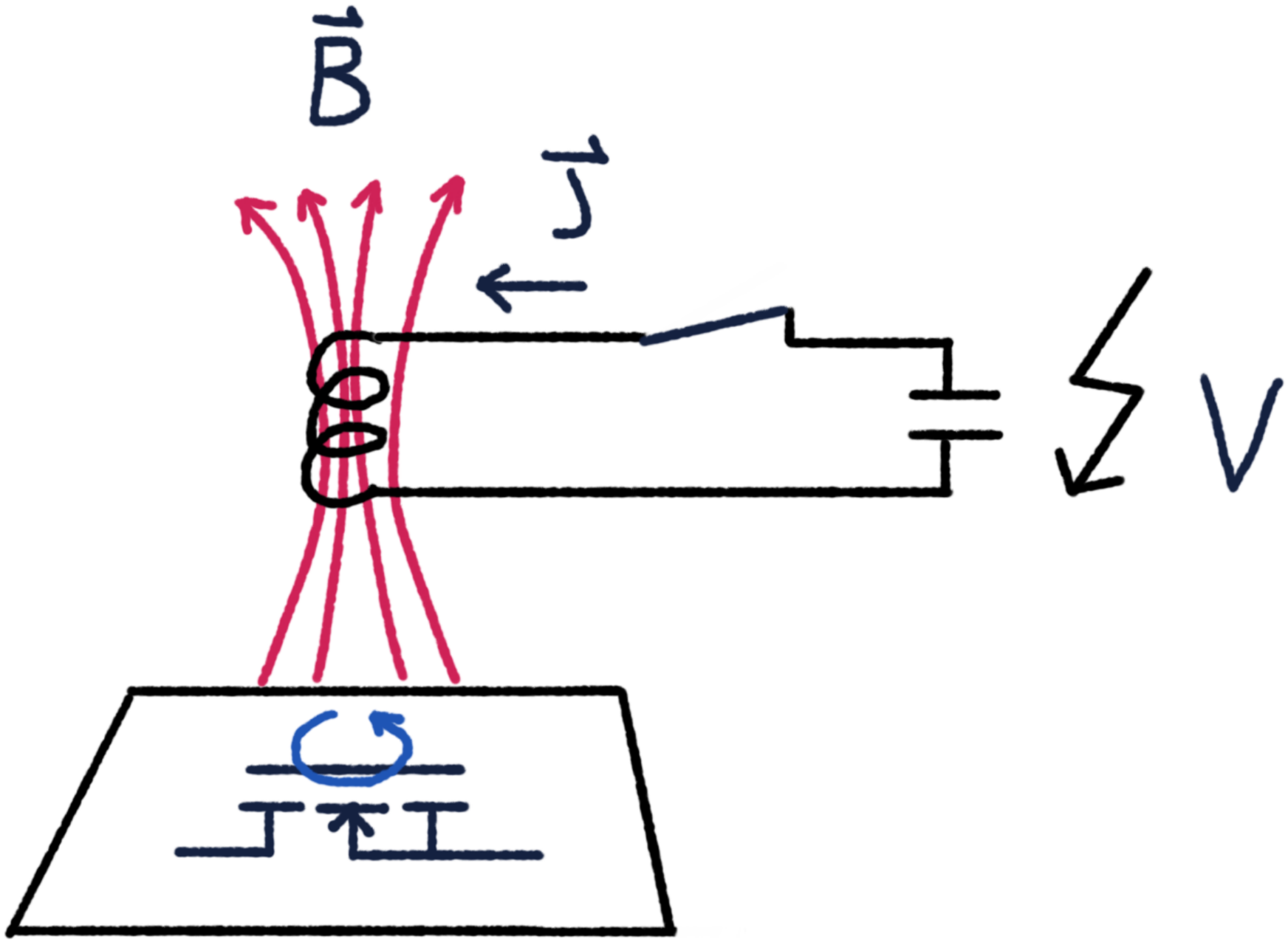
https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2025-022.txt

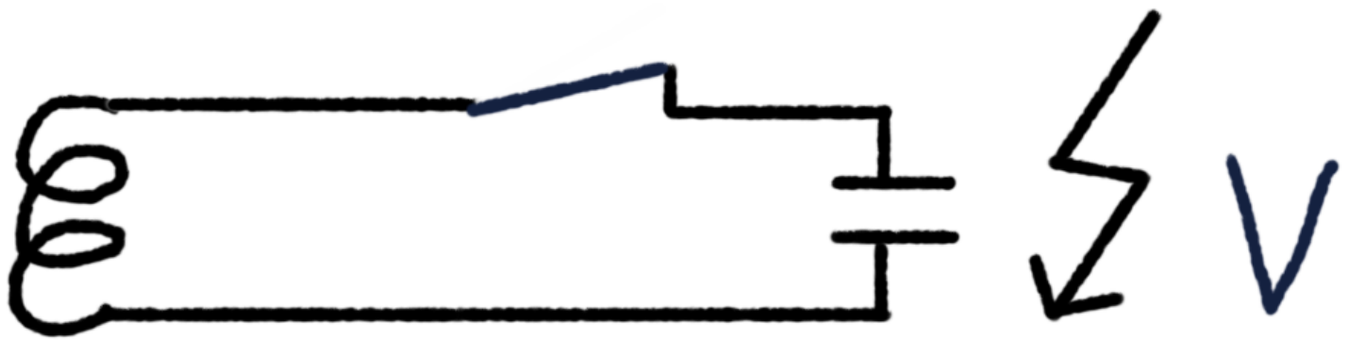
nRF54L15 DK


nPMI300 PMIC
Current measurement pins
NFC antenna connector
User-programmable LEDs
Power switch
SWF RF port for direct RF measurements
nRF54L15
External memory
SEGGER J-LINK USB-C connector
Reset button
Debug out
SEGGER J-Link OB Programmer/debugger
Debug in
User-programmable buttons
GPIO pins in parentheses are unavailable on headers without modification
nRF54L15-DK
400708
0032STL1
POWER
ON
OFF
SW4
VDDM CURRENT MEASURE
P6
GND
R24
nPMI300
U10
PORT P0
PORT P1
PORT P2
VBUS
VDDIO
NFC
J7
LED0
LED1
LED2
LED3
VDDM_nRF
VDD_nRF
DECD
P1
NORDIC SEMICONDUCTOR
DEBUGGER
J3
U7
U8
U2
N5340
SW1
RESET
SB2
SB88
3.6V Max
DEBUG OUT
P19
LD1
LABEL1
DEBUG IN/ TRACE
SB24
SB1
P15
BUTTON 0
BUTTON 1
BUTTON 2
BUTTON 3
U9
J1
X1
X2
U1
SB3
SB4
SB5
SB6
R33
R34


V



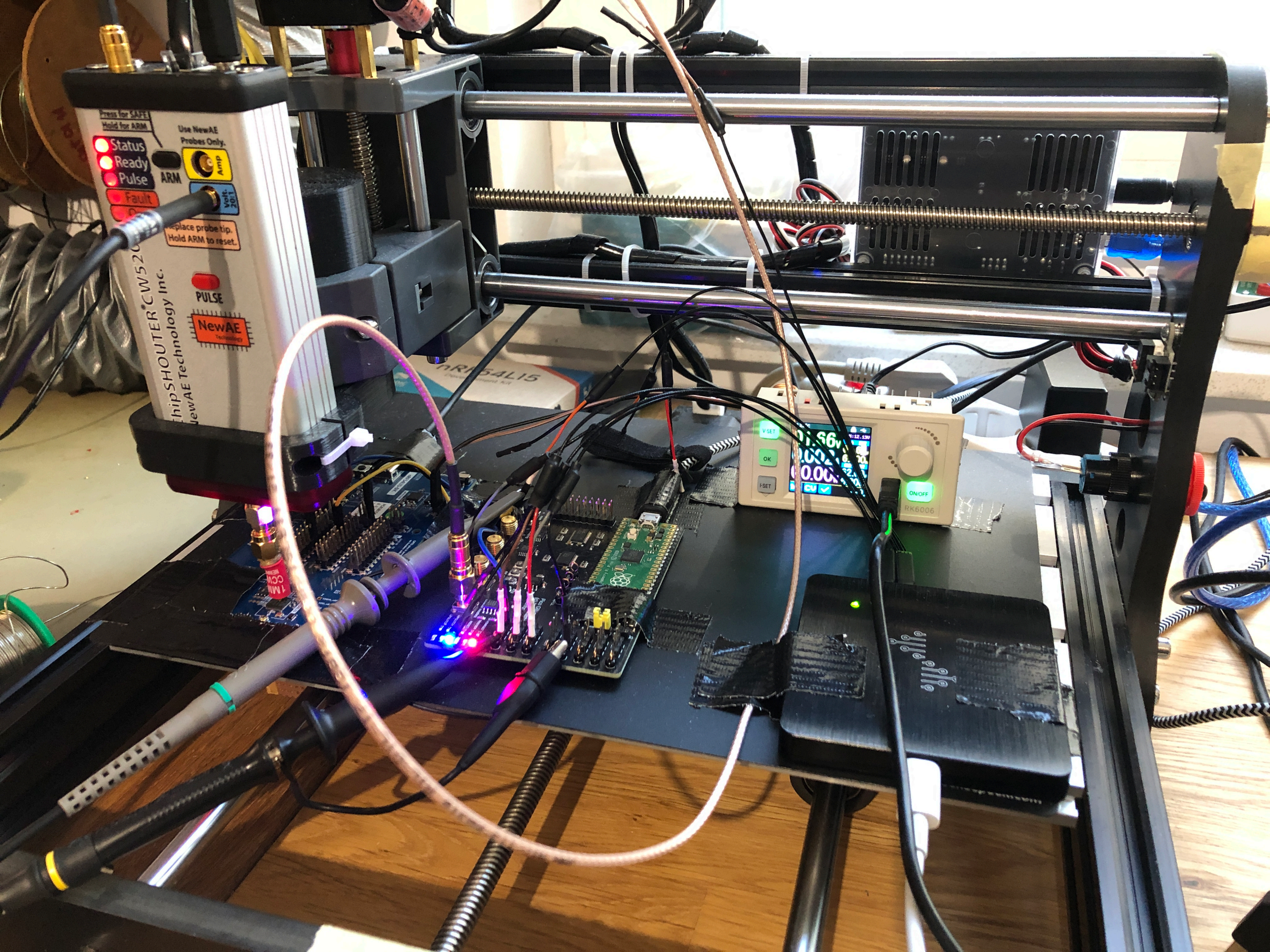

B
J
V

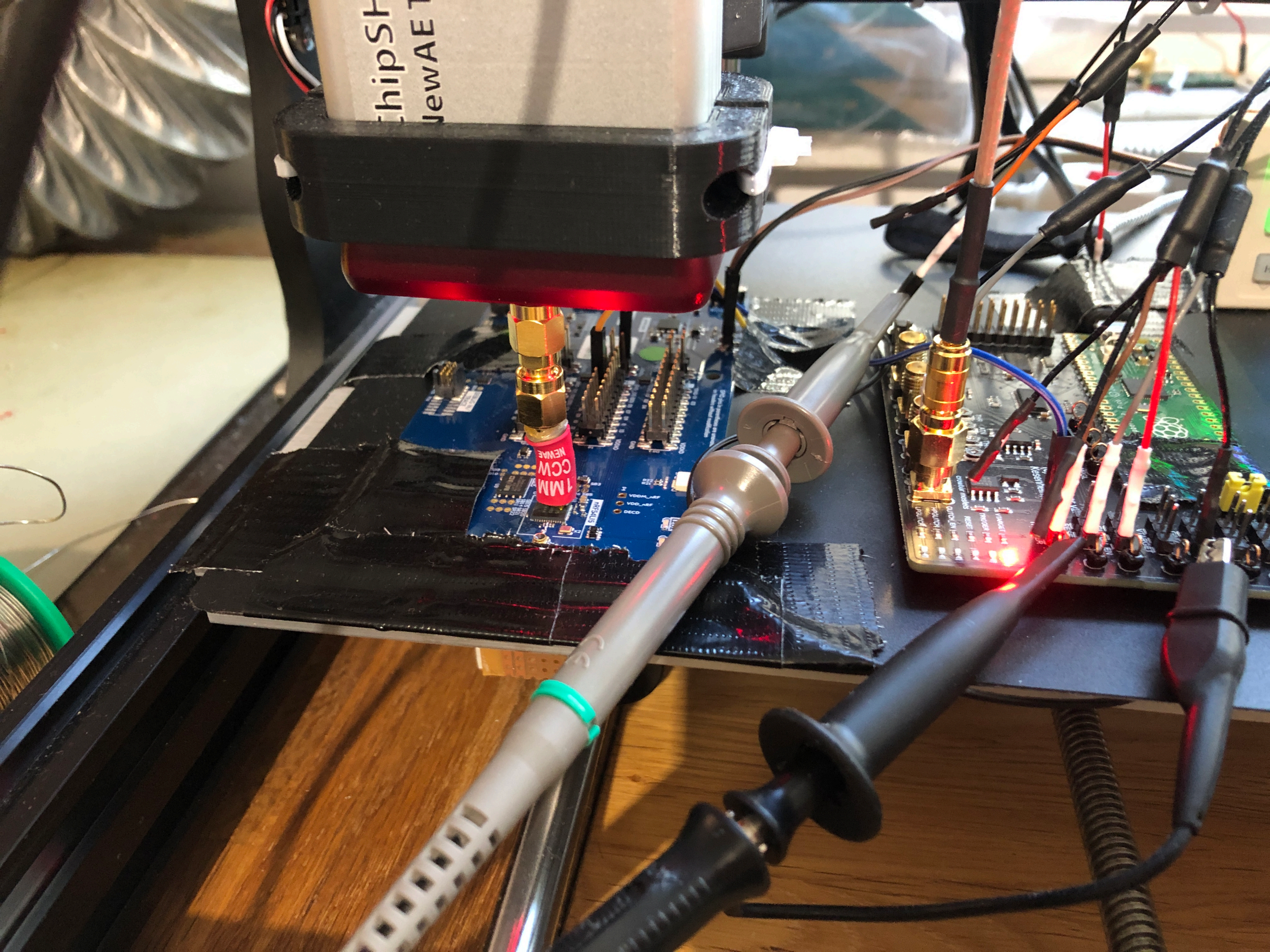


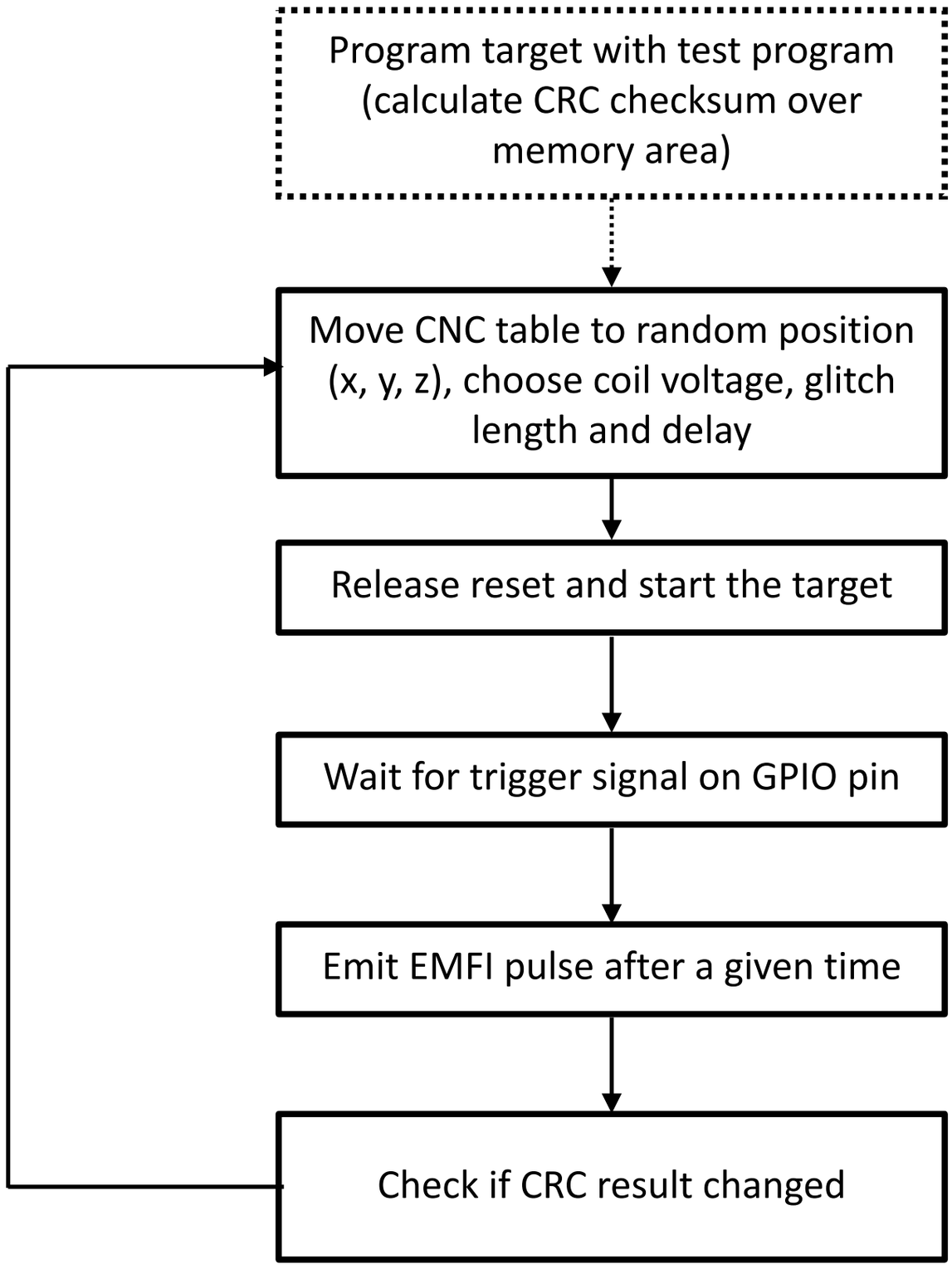


"0 → 1"


ChipSHOUTER® CW520
NewAE Technology Inc.
Status
Ready
Pulse
Fault
Open
ARM
PULSE
Replace probe tip.
Hold ARM to reset.
NewAE
DANGER HIGH VOLTAGE
Equipment to be used by trained
personnel only.


Press for SAFE
Hold for ARM
Use NewAE
Probes Only.
Status
Ready
Pulse
Fault
ARM
Amp
Volt.
Replace probe tip.
Hold ARM to reset.
PULSE
NewAE
ChipSHOUTER®CW520
NewAE Technology Inc.
V-SET
OK
I-SET
ON/OFF
RK6006


ChipSh
NewAE T
CCW
1MM
NEMA5
VDDA_REF
VDD_REF
DECD
RESULTS
KeenHeat
TRIGGER
RESET


Program target with test program (calculate CRC checksum over memory area)
Move CNC table to random position (x, y, z), choose coil voltage, glitch length and delay
Release reset and start the target
Wait for trigger signal on GPIO pin
Emit EMFI pulse after a given time
Check if CRC result changed

File Edit View Bookmarks Plugins Settings Help

10.03.25 16:28

nrf54-emfi-glitching : zsh — Konsole

New Tab Split View Copy Paste Find...

nrf54-emfi-glitching : zsh

```
Using cached zipp-3.21.0-py3-none-any.whl (9.6 kB)
Installing collected packages: pytz, pyserial, dash-table, dash-html-components, dash-core-components,
 zipp, urllib3, tzdata, typing-extensions, tqdm, six, setuptools, python-dotenv, pyparsing, pillow, pa
ckaging, numpy, nest-asyncio, narwhals, more-itertools, MarkupSafe, kiwisolver, itsdangerous, idna, fo
nttools, evdev, cycler, click, charset-normalizer, certifi, blinker, Werkzeug, scipy, retrying, reques
ts, python-xlib, python-dateutil, plotly, Jinja2, importlib-metadata, contourpy, chipshouter, adafruit
-ampy, pynput, pandas, matplotlib, Flask, dash, dash_bootstrap_components, dash_ag_grid, findus, emfin
dus
Successfully installed Flask-3.0.3 Jinja2-3.1.6 MarkupSafe-3.0.2 Werkzeug-3.0.6 adafruit-ampy-1.1.0 bl
inker-1.9.0 certifi-2025.1.31 charset-normalizer-3.4.1 chipshouter-1.1.0 click-8.1.8 contourpy-1.3.1 c
ycler-0.12.1 dash-2.18.2 dash-core-components-2.0.0 dash-html-components-2.0.0 dash-table-5.0.0 dash_a
g_grid-31.3.0 dash_bootstrap_components-1.7.1 emfindus-0.9.0 evdev-1.9.1 findus-1.3.0 fonttools-4.56.0
 idna-3.10 importlib-metadata-8.6.1 itsdangerous-2.2.0 kiwisolver-1.4.8 matplotlib-3.10.1 more-itertoo
ls-10.6.0 narwhals-1.30.0 nest-asyncio-1.6.0 numpy-2.2.3 packaging-24.2 pandas-2.2.3 pillow-11.1.0 plo
tly-6.0.0 pynput-1.8.0 pyparsing-3.2.1 pyserial-3.5 python-dateutil-2.9.0.post0 python-dotenv-1.0.1 py
thon-xlib-0.33 pytz-2025.1 requests-2.32.3 retrying-1.3.4 scipy-1.15.2 setuptools-76.0.0 six-1.17.0 tq
dm-4.67.1 typing-extensions-4.12.2 tzdata-2025.1 urllib3-2.3.0 zipp-3.21.0

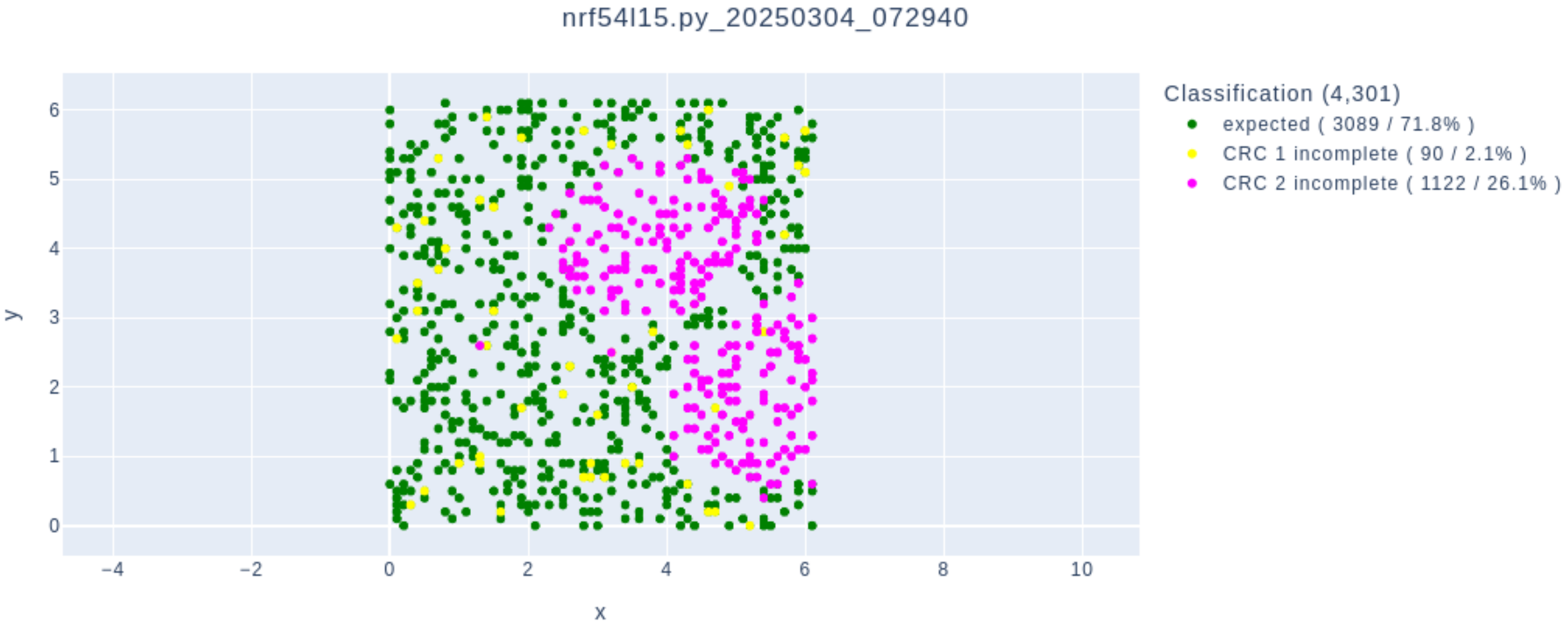
[notice] A new release of pip is available: 24.3.1 -> 25.0.1
[notice] To update, run: pip install --upgrade pip
(.venv) mkesenheimer@kesenheimer-latitude5450 research/nrf54-emfi-glitching »
```

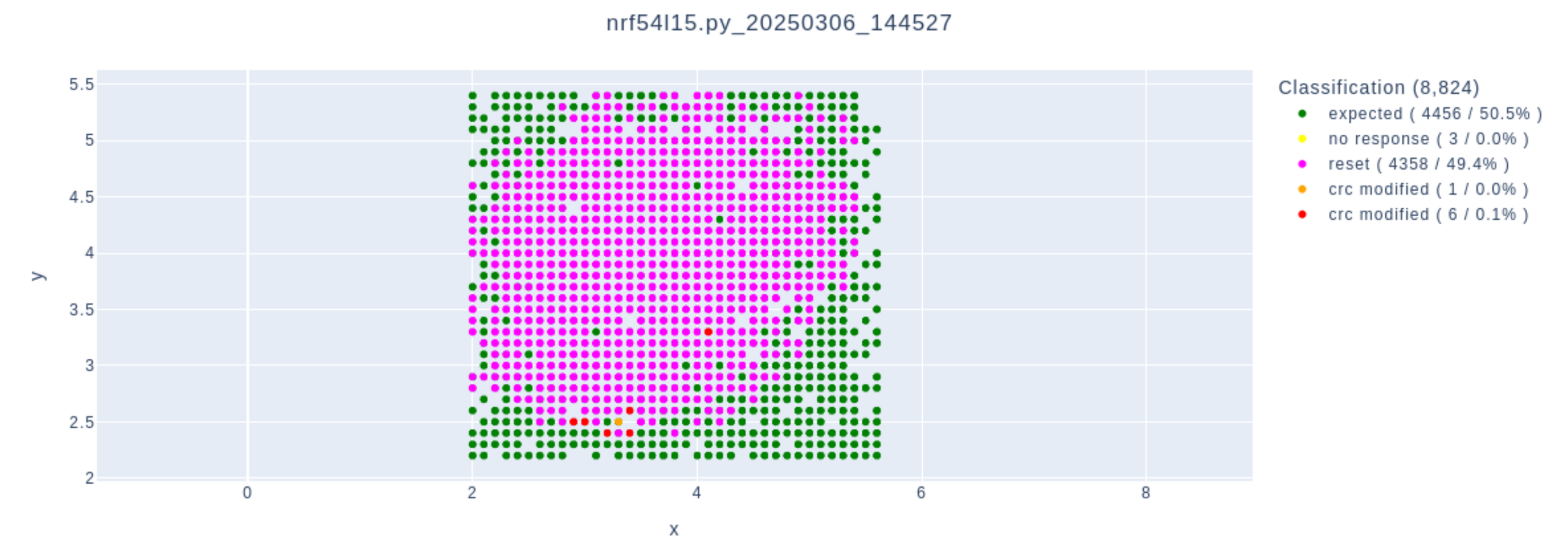
nrf54-emfi-glitching : analyzer

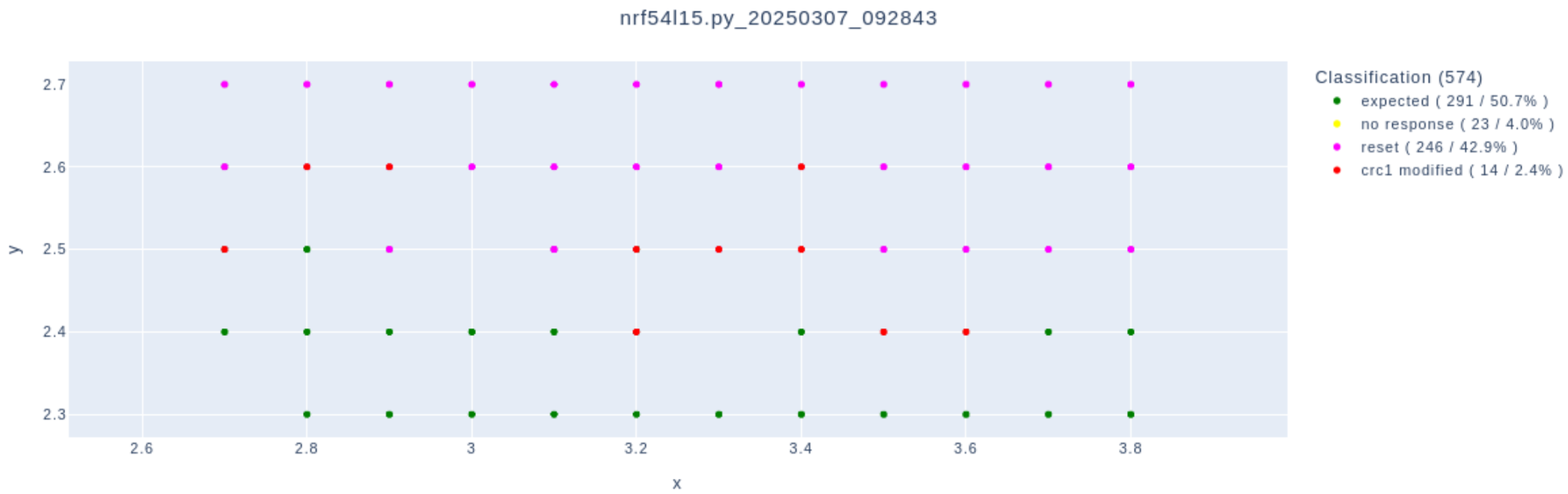
```
It took 35 miliseconds to generate this data.
It took 63 miliseconds to generate this data.
Copying nrf54l15.py_20250310_162144.sqlite to /tmp and opening
 from there
It took 28 miliseconds to generate this data.
Copying nrf54l15.py_20250310_162144.sqlite to /tmp and opening
 from there
It took 35 miliseconds to generate this data.
Copying nrf54l15.py_20250310_162144.sqlite to /tmp and opening
 from there
It took 41 miliseconds to generate this data.
Copying nrf54l15.py_20250310_162144.sqlite to /tmp and opening
 from there
It took 33 miliseconds to generate this data.
Copying nrf54l15.py_20250310_162144.sqlite to /tmp and opening
 from there
It took 29 miliseconds to generate this data.
Copying nrf54l15.py_20250310_162144.sqlite to /tmp and opening
 from there
It took 26 miliseconds to generate this data.
```

nrf54-emfi-glitching : zsh

```
[+] Absolute movement: x=3.1 y=2.5 z=0.1 with feed rate f=300
[+] Experiment 188      0       (0)     190     110     234861  3.1     2.5     0.1     G       b'StartConfig:\x01CRC:4M2\x00CRC(Config+CRC):\x96\xd5\t\x00'
[+] Experiment 189      0       (0)     190     90      162425  3.1     2.5     0.1     G       b'StartConfig:\x01CRC:4M2\x00CRC(Config+CRC):\x96\xd5\t\x00'
[+] Absolute movement: x=3.7 y=2.6 z=0.2 with feed rate f=300
[+] Experiment 190      0       (0)     190     97      304207  3.7     2.6     0.2     R       b'StartConfig:\x01CRC:||\xff\x00CRC(Config+CRC):Q#>\x00'
[+] Experiment 191      0       (0)     190     101     214691  3.7     2.6     0.2     M       b'Start'
[+] Absolute movement: x=3.4 y=2.5 z=0.4 with feed rate f=300
[+] Experiment 192      0       (0)     190     97      270050  3.4     2.5     0.4     G       b'StartConfig:\x01CRC:4M2\x00CRC(Config+CRC):\x96\xd5\t\x00'
[+] Experiment 193      0       (0)     190     94      176265  3.4     2.5     0.4     G       b'StartConfig:\x01CRC:4M2\x00CRC(Config+CRC):\x96\xd5\t\x00'
[+] Absolute movement: x=3.0 y=2.5 z=0.4 with feed rate f=300
[+] Experiment 194      0       (0)     190     99      232531  3.0     2.5     0.4     G       b'StartConfig:\x01CRC:4M2\x00CRC(Config+CRC):\x96\xd5\t\x00'
[+] Experiment 195      0       (0)     190     104     86722   3.0     2.5     0.4     G       b'StartConfig:\x01CRC:4M2\x00CRC(Config+CRC):\x96\xd5\t\x00'
[+] Absolute movement: x=3.5 y=2.5 z=0.2 with feed rate f=300
^C
Exitting...
[+] Moving CNC table back to starting position.
current z position: 0.4
[+] Terminating gracefully.
[+] Disarming ChipSHOUTER
(.venv) mkesenheimer@kesenheimer-latitude5450 research/nrf54-emfi-glitching » python nrf54l15.py --rpico /dev/ttyACM0 --target /dev/ttyACM2 --cnc /dev/ttyUSB1 --shoute
r /dev/ttyUSB0 --length 90 110 1 --delay 0 310_000 1 --voltage 190 190 0 -x 2.9 3.8 0.1 -y 2.5 2.8 0.1 -z 0.1 0.4 0.1 --number-of-tries 2 --external-trigger --resume
```


nrf54l15.py_20250304_072940
Classification (4,301)
expected (3089 / 71.8%)
CRC 1 incomplete (90 / 2.1%)
CRC 2 incomplete (1122 / 26.1%)
y
x
6
5
4
3
2
1
0
−4
−2
0
2
4
6
8
10


nrf54l15.py_20250306_144527
Classification (8,824)
expected (4456 / 50.5%)
no response (3 / 0.0%)
reset (4358 / 49.4%)
crc modified (1 / 0.0%)
crc modified (6 / 0.1%)
5.5
5
4.5
4
3.5
3
2.5
2
0
2
4
6
8
x



re.search(*, response)

green	expected	yellow	no response	magenta	reset	orange	crc2 modified
cyan	cyan-label	blue	blue-label	black	black-label	red	crc1 modified

Combine data?

Yes No

AMOUNT	COLOR	YMIN	YMAX	XMIN	XMAX	RESPONSE	RESPONSE_HEX
filter data							
289	G	2.3	2.7	2.7	3.8	b'StartConfig:\x01CRC:\xb4QG\x01CRC(Config+CRC):\x06\xc6\xd6\x00'	62 27 53 74 61 72 74 43
246	M	2.4	2.7	2.7	3.8	b'Start'	62 27 53 74 61 72 74 27
23	Y	2.4	2.7	2.7	3.6	b''	62 27 27
2	G	2.3	2.7	3	3.7	b'StartConfig:\x01CRC:\xb4QG\x01CRC(Config+CRC):\x06\xc6\xd6\x00*** Booting '	62 27 53 74 61 72 74 43
1	R	2.5	2.5	3.3	3.3	b'StartConfig:\x01CRC:]\xa7\xff\x00CRC(Config+CRC):\xbdq\x1c\x00'	62 27 53 74 61 72 74 43
1	R	2.4	2.4	3.5	3.5	b'StartConfig:\x01CRC:D\xb2a\x01CRC(Config+CRC):a0x\x00'	62 27 53 74 61 72 74 43
1	R	2.4	2.4	3.6	3.6	b'StartConfig:\x01CRC:\xb3E\xad\x00CRC(Config+CRC):x\xda\x10\x00'	62 27 53 74 61 72 74 43

github.com/SySS-Research/emfindus/tree/main/projects/nrf54

Platform Solutions Resources Open Source Enterprise Pricing Sign in Sign up

SySS-Research / **emfindus** Public

Notifications Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

Files

main

Go to file

- emfindus.egg-info
- emfindus
- examples
- projects
  - nrf54
    - test_program
    - configuration.json
    - nrf54l15.py

emfindus / projects / **nrf54** /

MKesenheimer Initial commit — b31df39 · 11 minutes ago — History

Name	Last commit message	Last commit date
..		
test_program	Initial commit	11 minutes ago
configuration.json	Initial commit	11 minutes ago
nrf54l15.py	Initial commit	11 minutes ago

https://github.com/SySS-Research/emfindus



https://blog.syss.com/posts/nrf54-emfi/

https://www.syss.de/fileadmin/dokumente/Publikationen/Advisories/SYSS-2025-022.txt