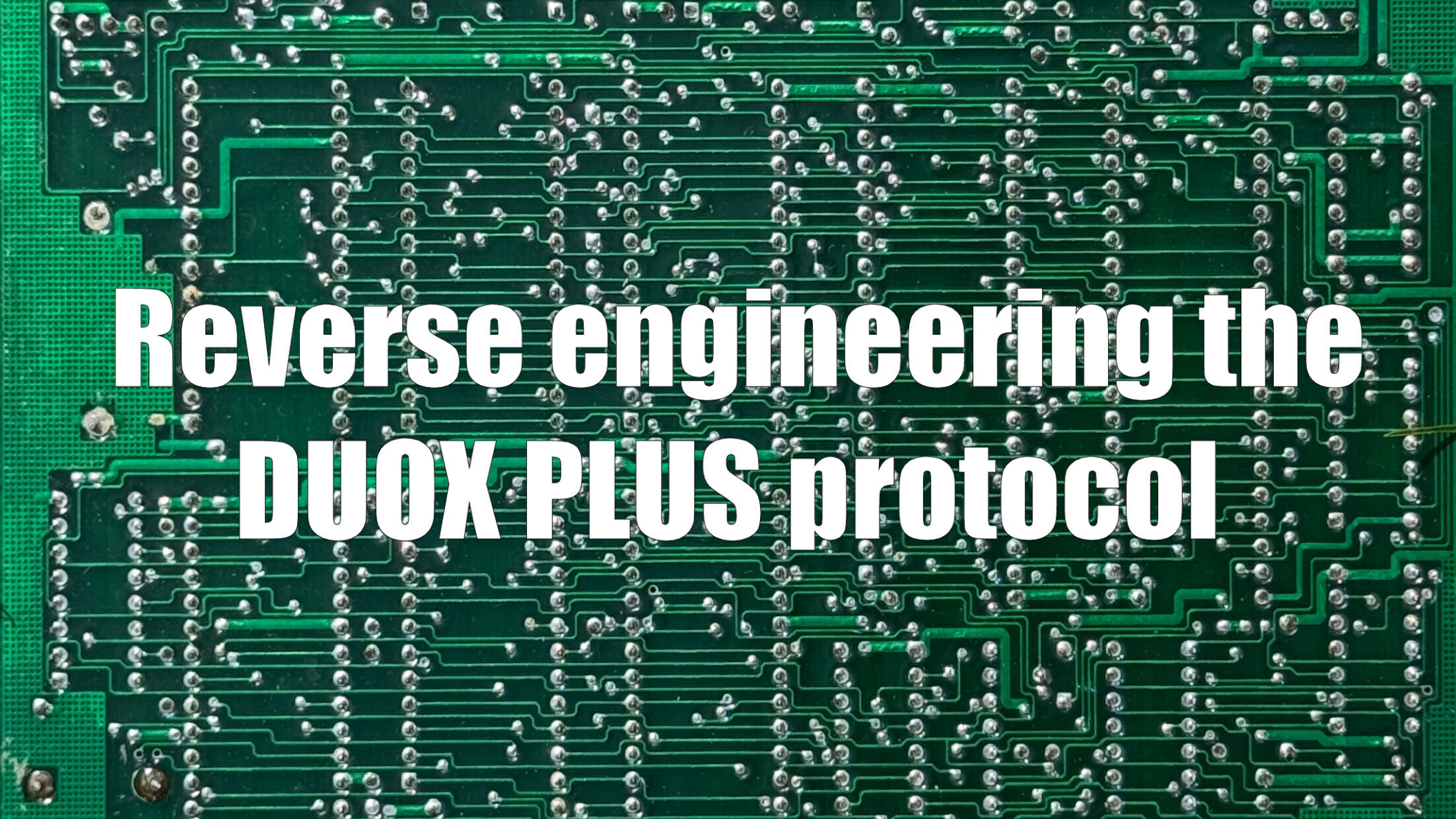


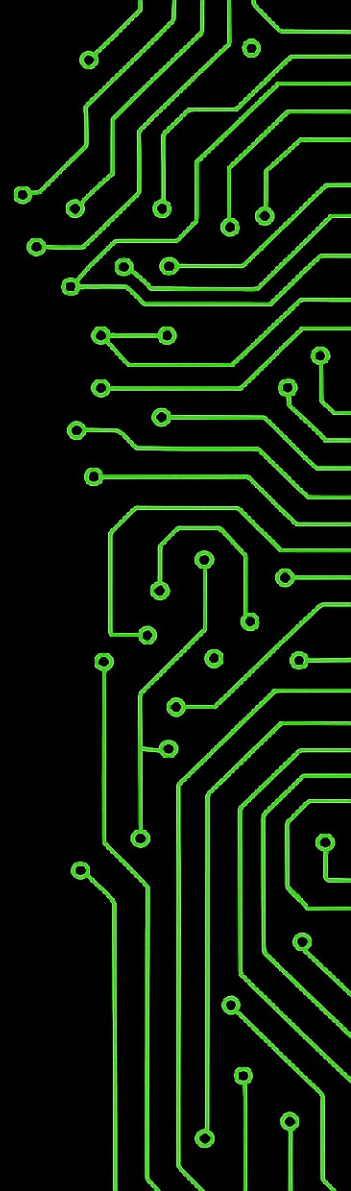


Reverse engineering the DUOX PLUS protocol

TWO YEARS IN A ROW?!



TROOPERS



THIS GUY AGAIN

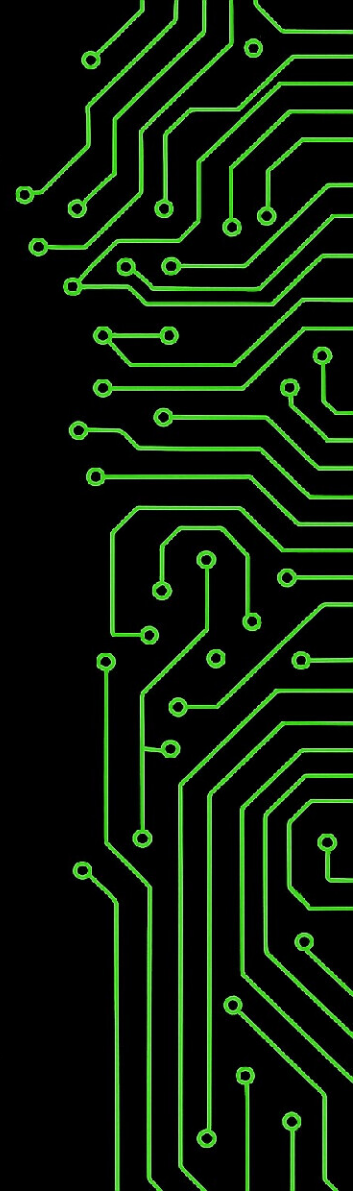


🐦 @k@chaos.social

Follow me today
for free

- Founder and lead researcher at Possible Security, Latvia
 - <https://possiblesecurity.com/>
- Mg. sc. comp., Mg. phys.
 - <https://kirils.org/>
- Hacker of:
 - Network flow analysis & RE
 - Social engineering
 - Legal dimension

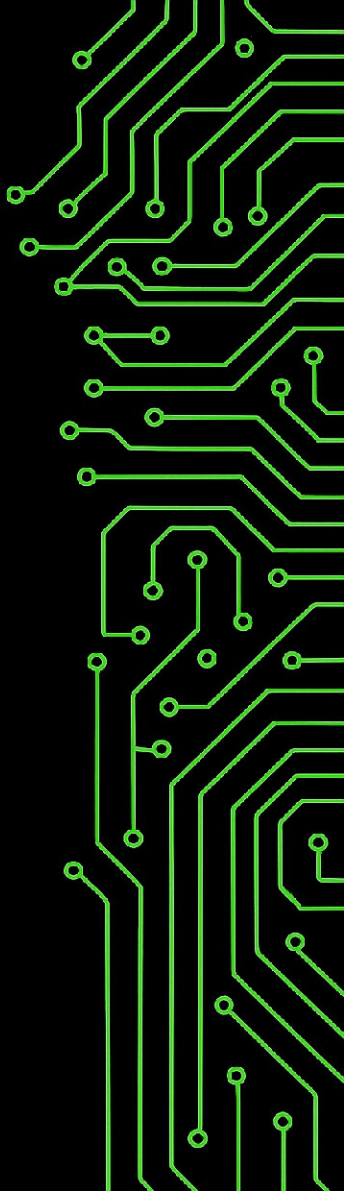


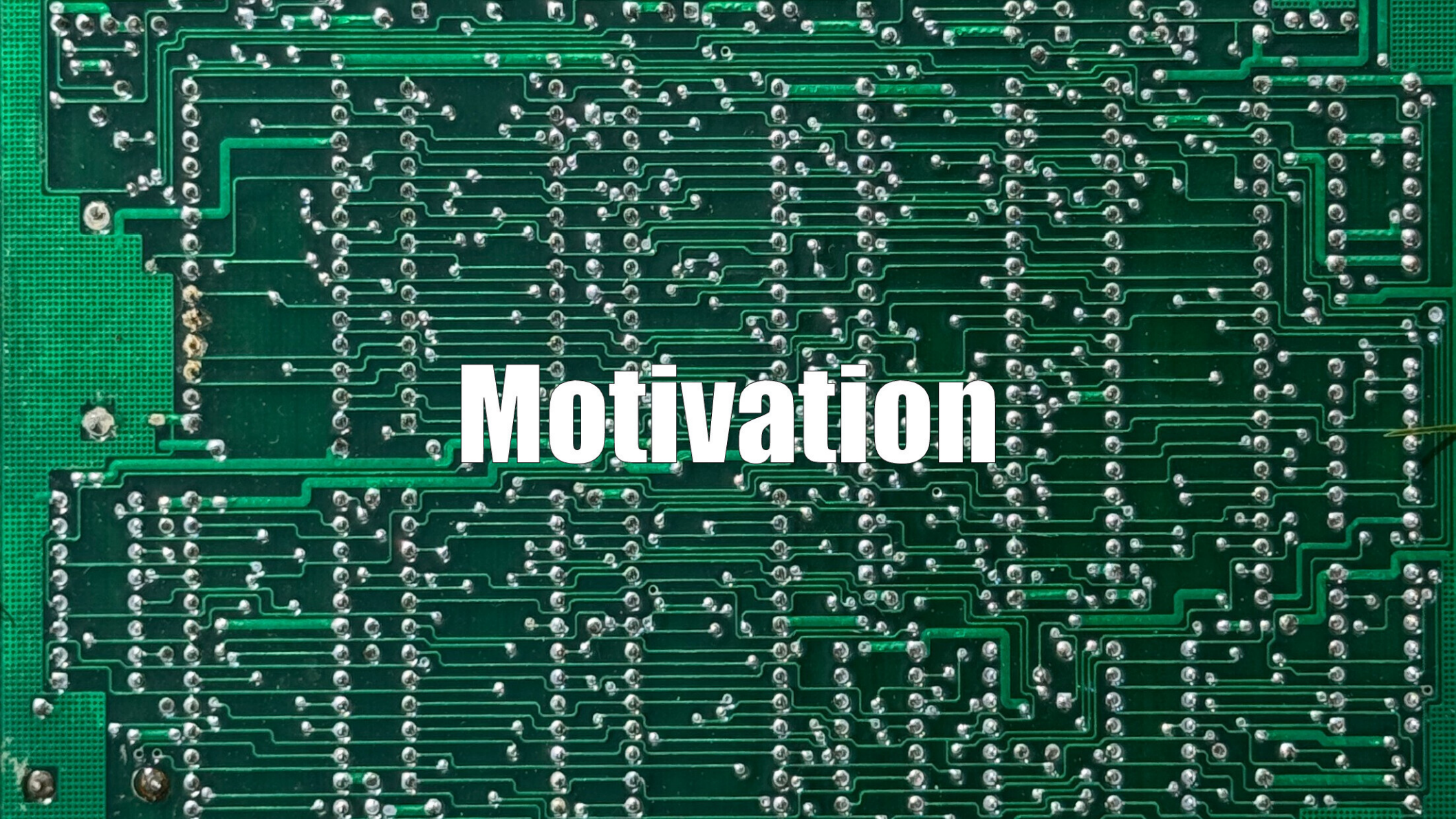


Source: Fermax(?)

CONTENTS

- motivation
- reversing process with limited tools
 - my journey
 - mistakes
- reversing process with limited knowledge
- circuit board reverse engineering
- firmware reverse engineering
- ongoing (future) work





Motivation

LOOK AT IT

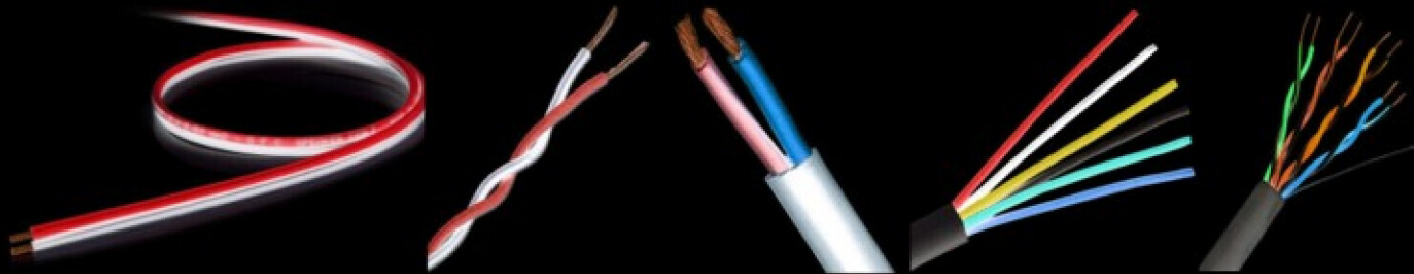
- get apartment
 - with an intercom
- work remotely
 - can't always walk to the door during meetings
- terminal apparently user-swappable



READ ABOUT IT

DUOX PLUS allows the use of many types of wiring as long as they comply with minimum quality characteristics. For the correct operation of the system, FERMAX recommends:

- Parallel or Twisted cable in different cross-sections (1mm^2 , 0.5mm^2)
- CAT5 (0.2 mm^2)



NOTE: For other replacement cables (loose wires or telephone cable) please consult our technical department. With these types of cable, additional installation elements such as additional sources or regenerators may be required to ensure correct operation of the system with the highest quality.



READ ABOUT IT

The video intercom system shall:

- a- Integrate visitor call units (VCU), user receiver units (URU) and security management units (SMU).
- b- Be based on 2 non-polarized wires bus infrastructure supporting any type of cabling (parallel or UTP, with different sizes: AWG18, AWG20, AWG22, AWG24...).
- c- Be fully digital. Audio, video and commands shall be transmitted digitally to avoid external interferences and maintain quality over the installation.
- d- Adopt audio and video standards to fulfil compatibility with other systems.
- e- Have audio and video information encrypted and provide conversation secrecy to avoid another tenant can monitor it.



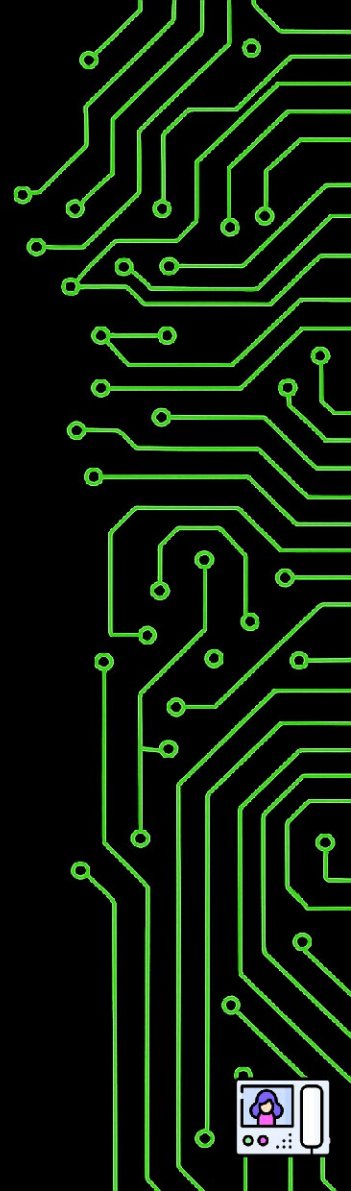
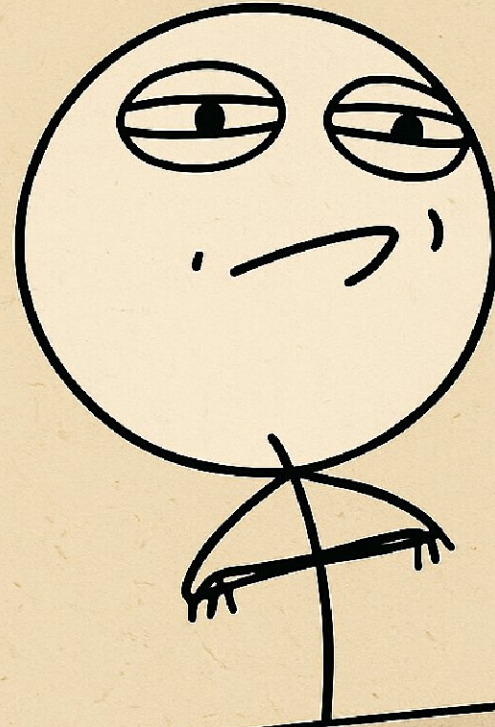
BUT IT'S USER-SWAPPABLE??? (ಠ_ಠ)

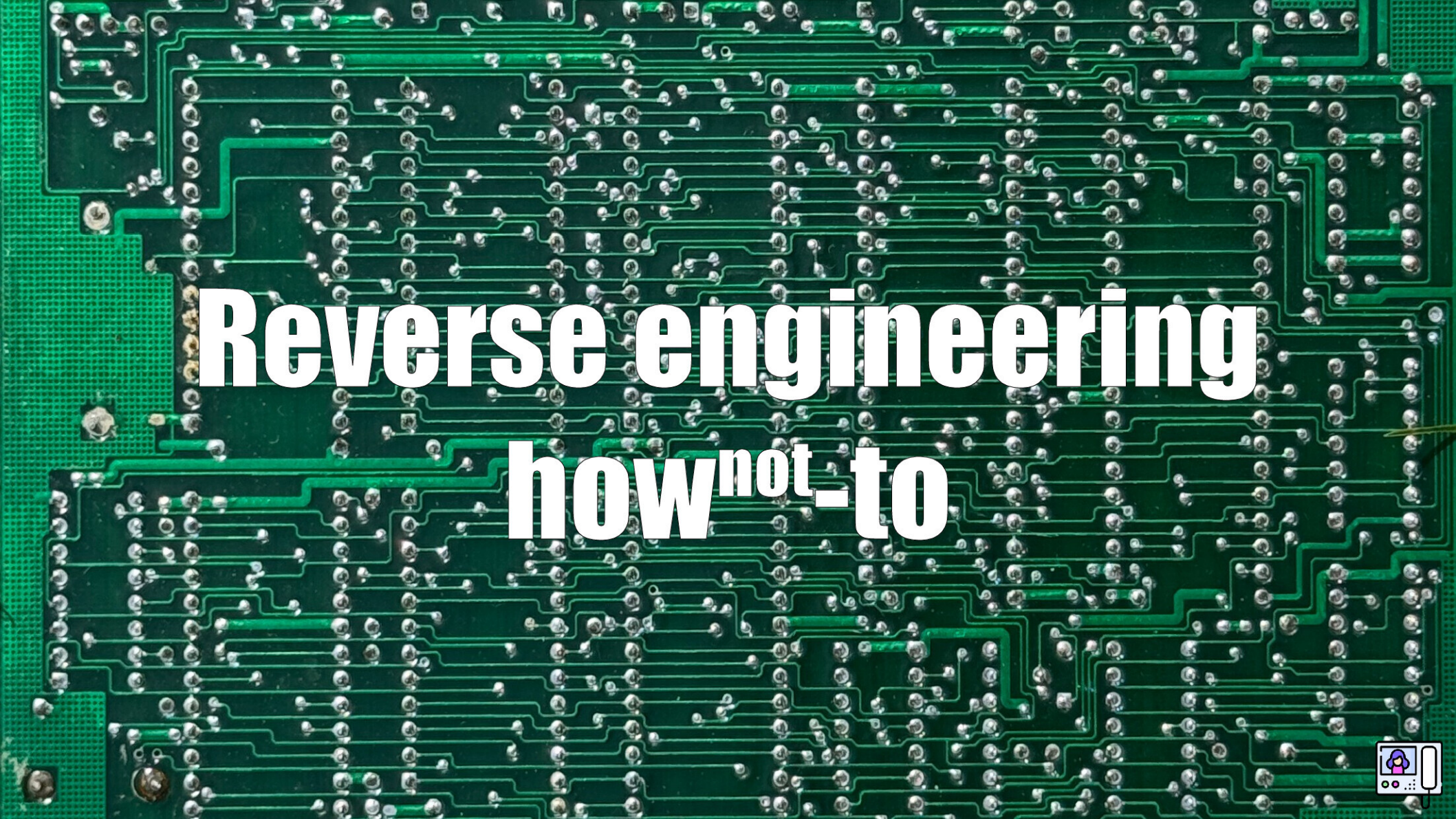


Source: Fermax Australia / Youtube



CHALLENGE ACCEPTED



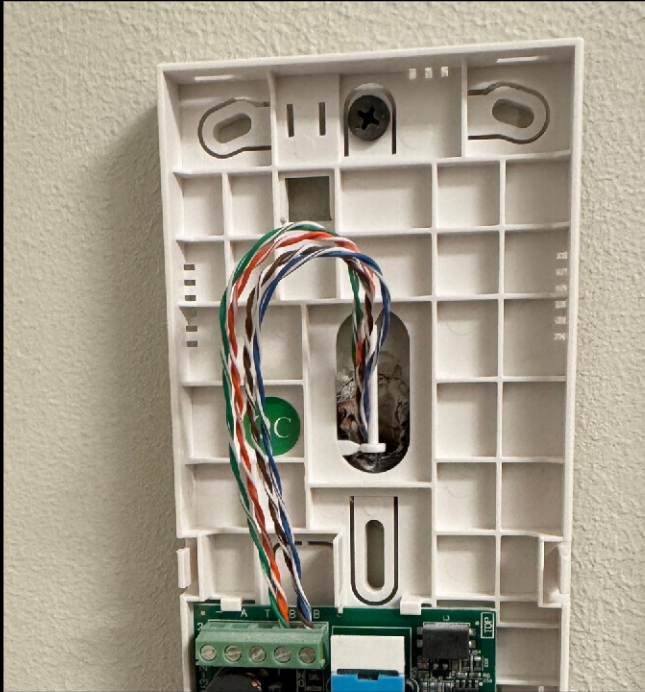


Reverse engineering how^{not}-to



OPEN IT UP

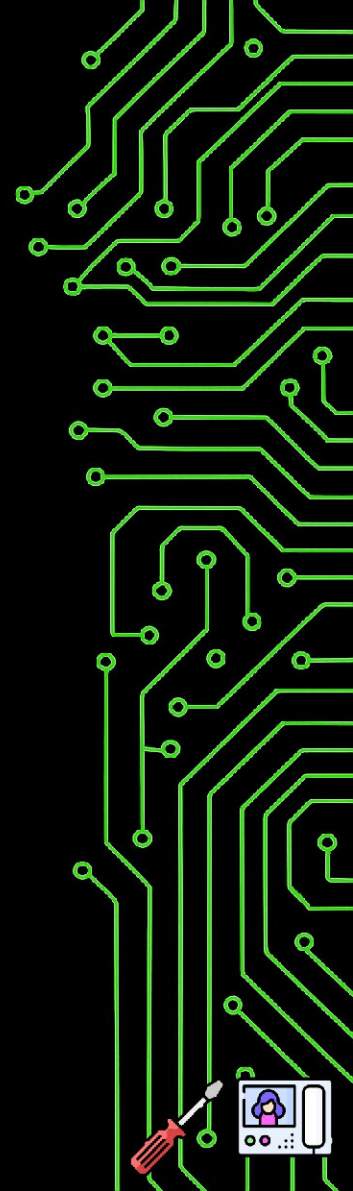
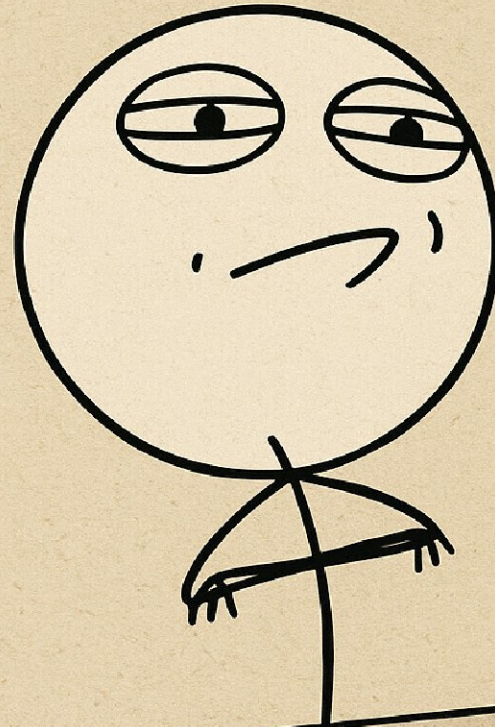
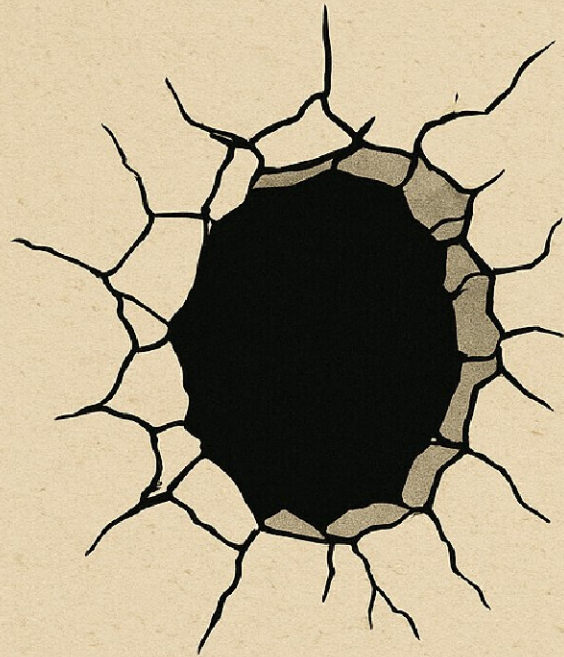
expectation



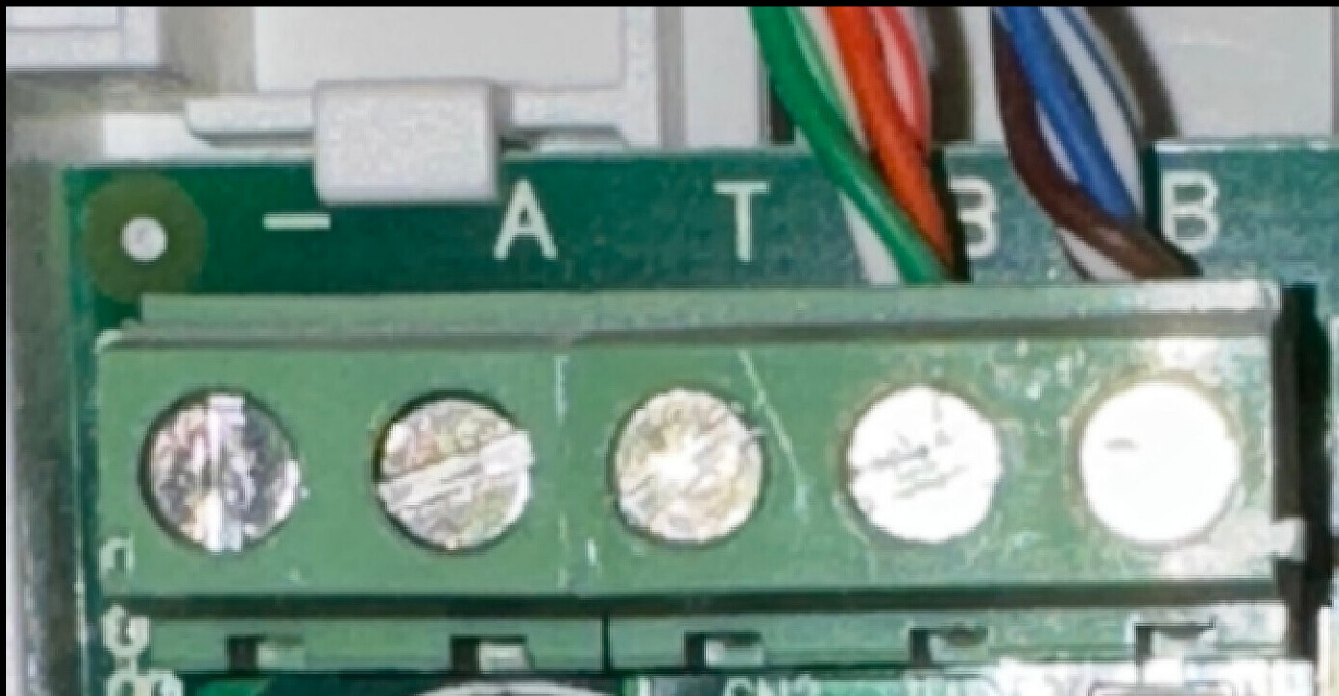
reality



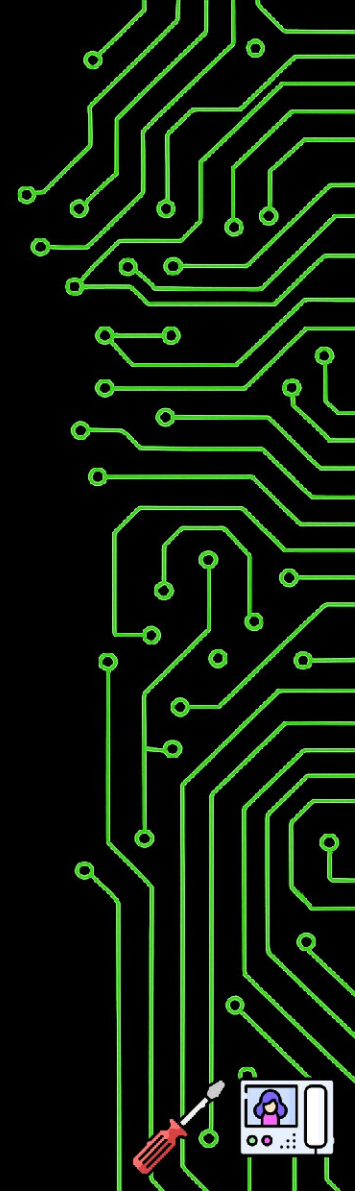
CHALLENGE ACCEPTED



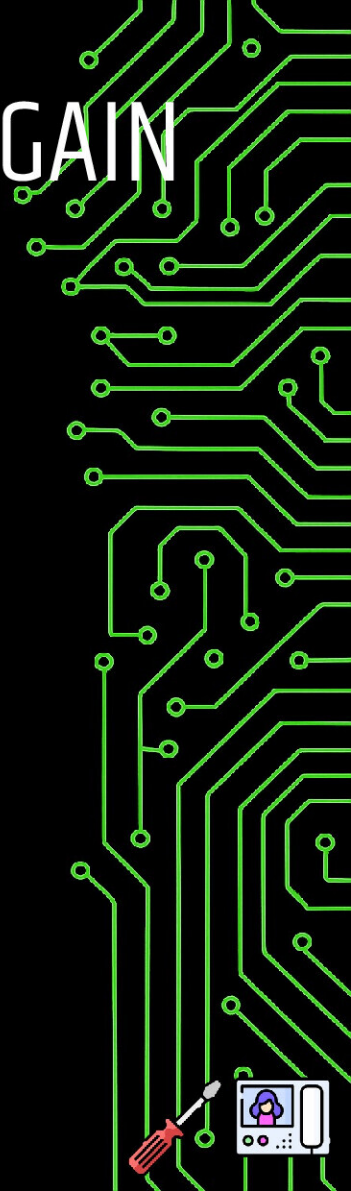
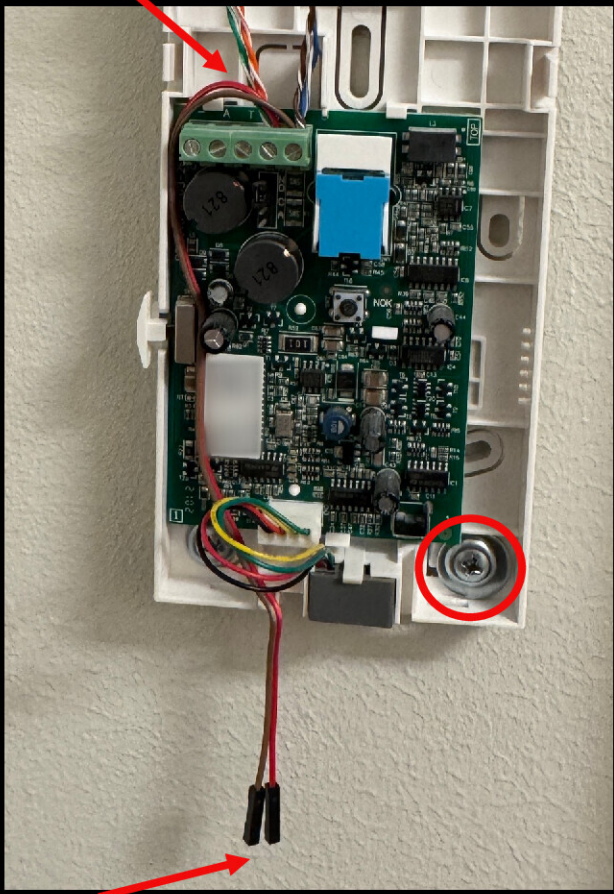
INTERFACE



- A T B B



MAKE SURE TO NOT HAVE TO OPEN IT UP AGAIN



MEASURE VOLTAGE

- 23.3÷24.0 DCV measured on B-B

DUOX PLUS is the first fully digital video door entry system in 2 non-polarized wires along the entire installation, versatile, flexible and scalable; capable of succes in any type of project.

- stuck for a month thinking about non-polarized DCV
- measured the other way around a month later



-24 DCV

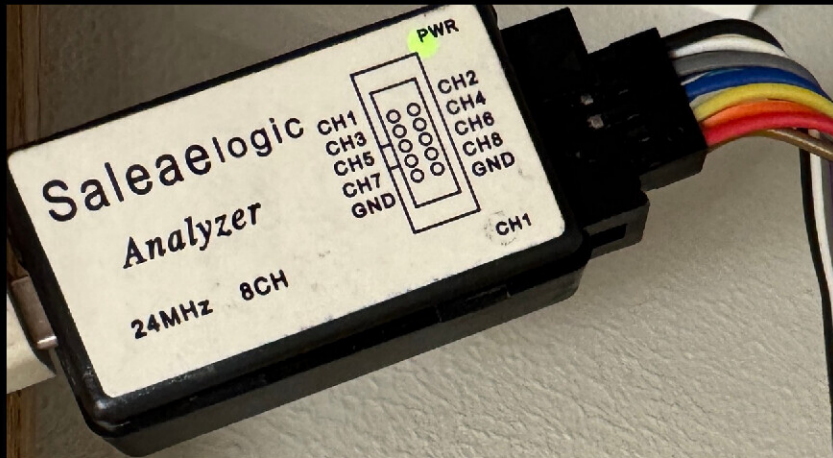


MEASURE VOLTAGE DURING INCOMING CALL

- because I remember POTS
 - 12 VDC idle
 - but 110 VAC when call incoming
- Here?
 - $23\text{VDC} \pm 1.8\text{VAC}$



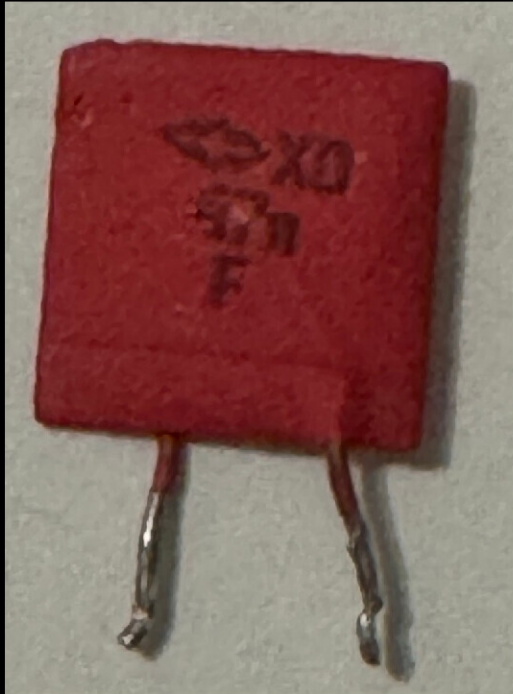
I'VE GOT SALEAE



- well, a clone
- 5V tolerant-ish
- 24MSps



BUT THE SIGNAL IS 24 VOLTS...



? tolerance

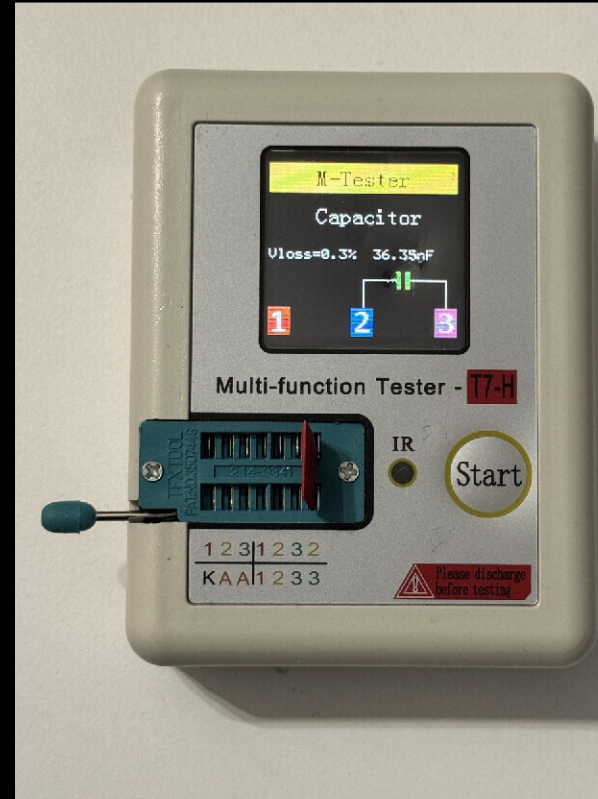
👎 $L = 0.8-; H = 1.2+$

- found some caps at home...

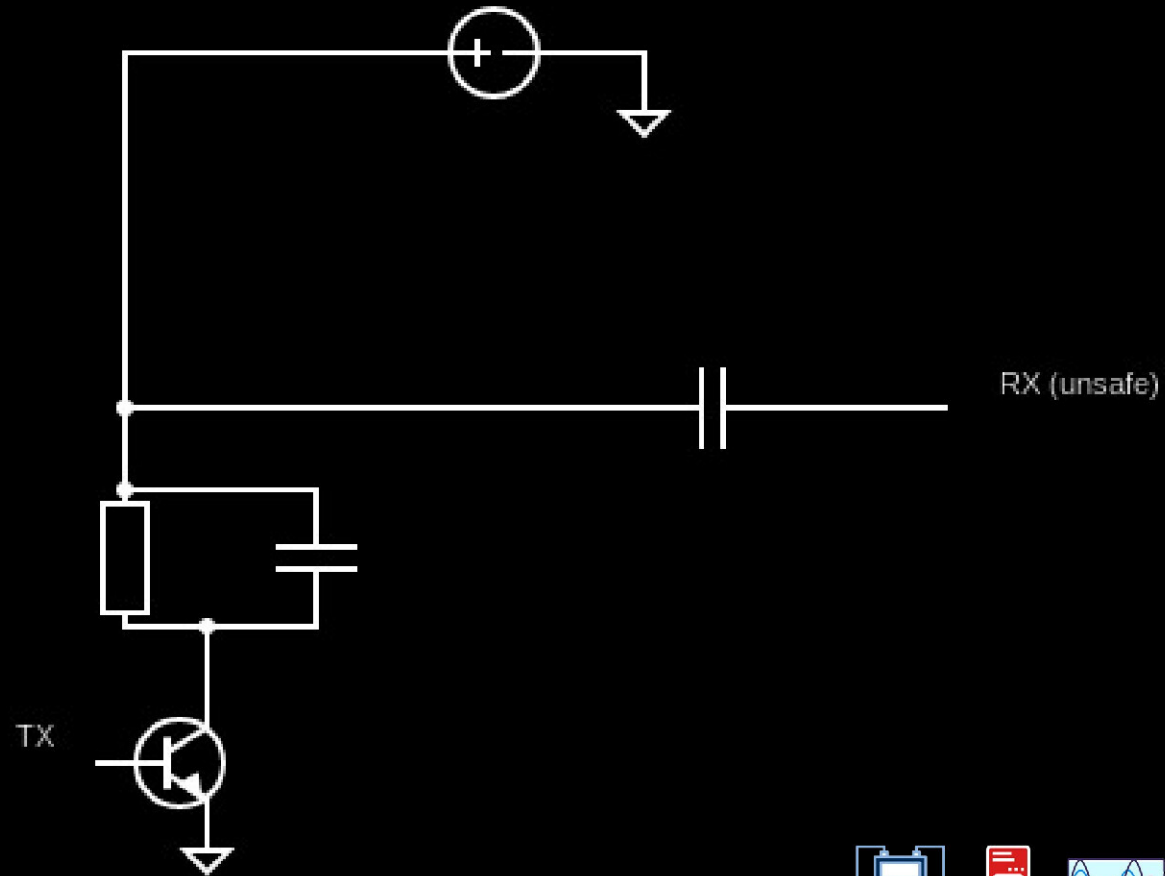


POST-SOVIET SCAVENGING

- how did I get them?
- well, I hope it still works
- $\sim 35\text{nF}$



DESIGNING AN INTERFACE

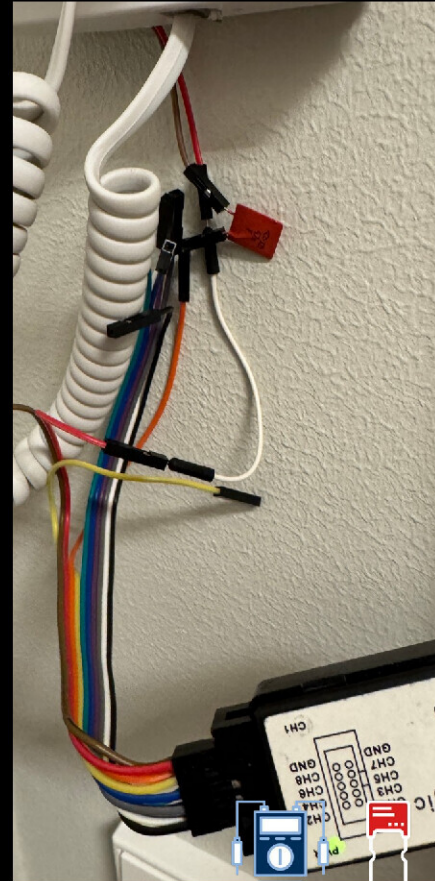


OH WAIT, I HAVE AN OSCILLOSCOPE!

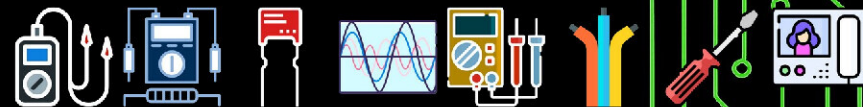
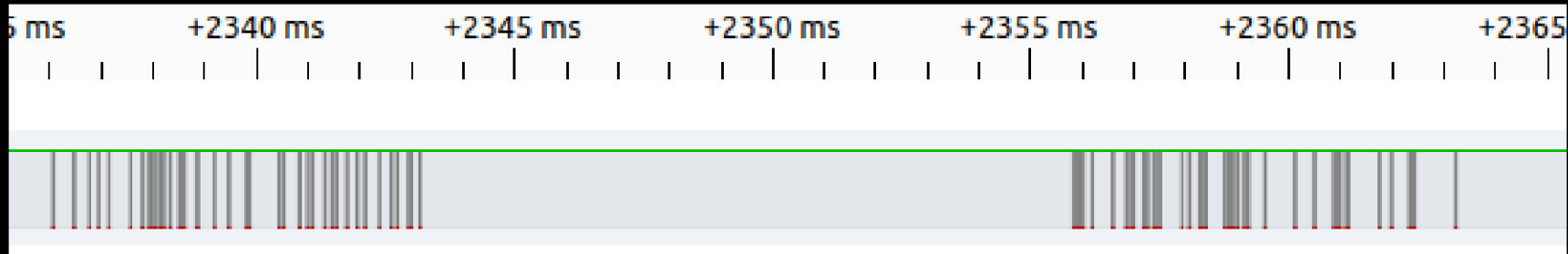
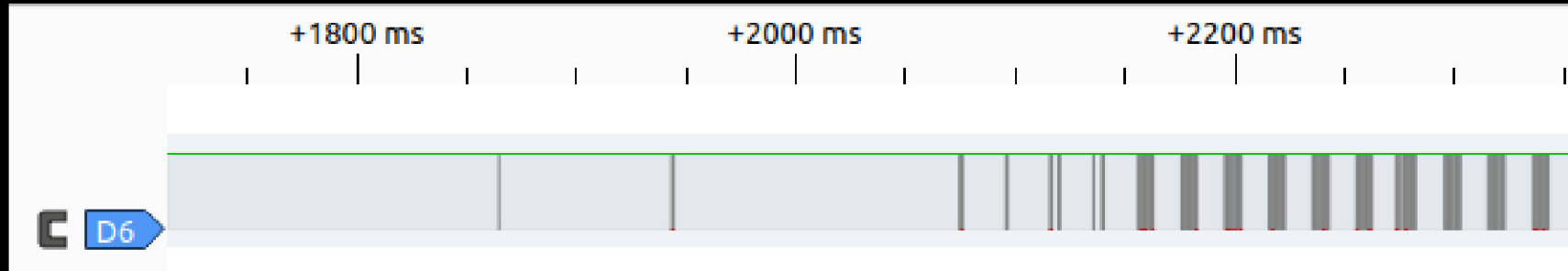


BACK TO SALEAE

- 24MSps... 😞
- 12MSps... 🙅
- 8MSps... 🙌



DATA!



WHAT DATA ARE WE LOOKING FOR?

1. DETECTOR
BLOCK: 00



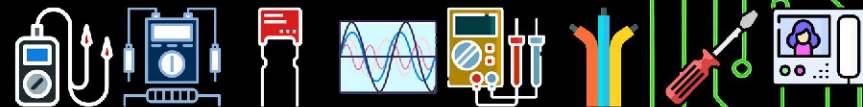
Enter the Block number that the telephones belong to verify.
Values admitted: 00 to 09



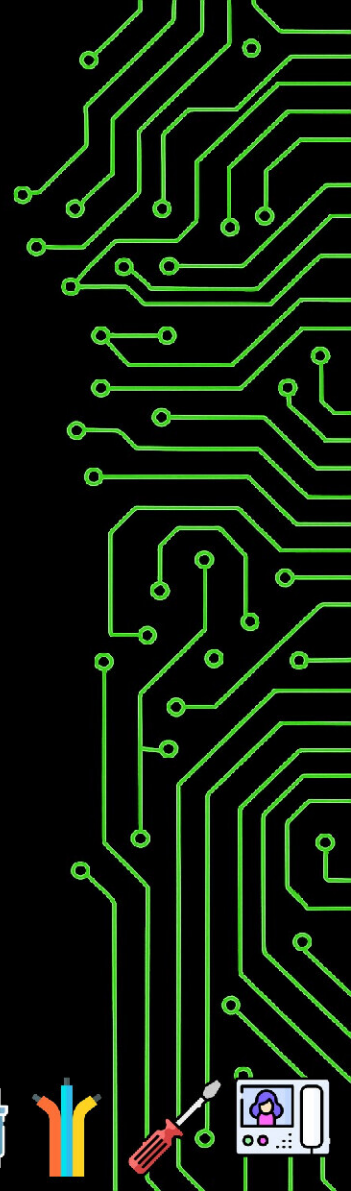
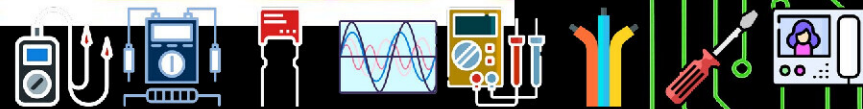
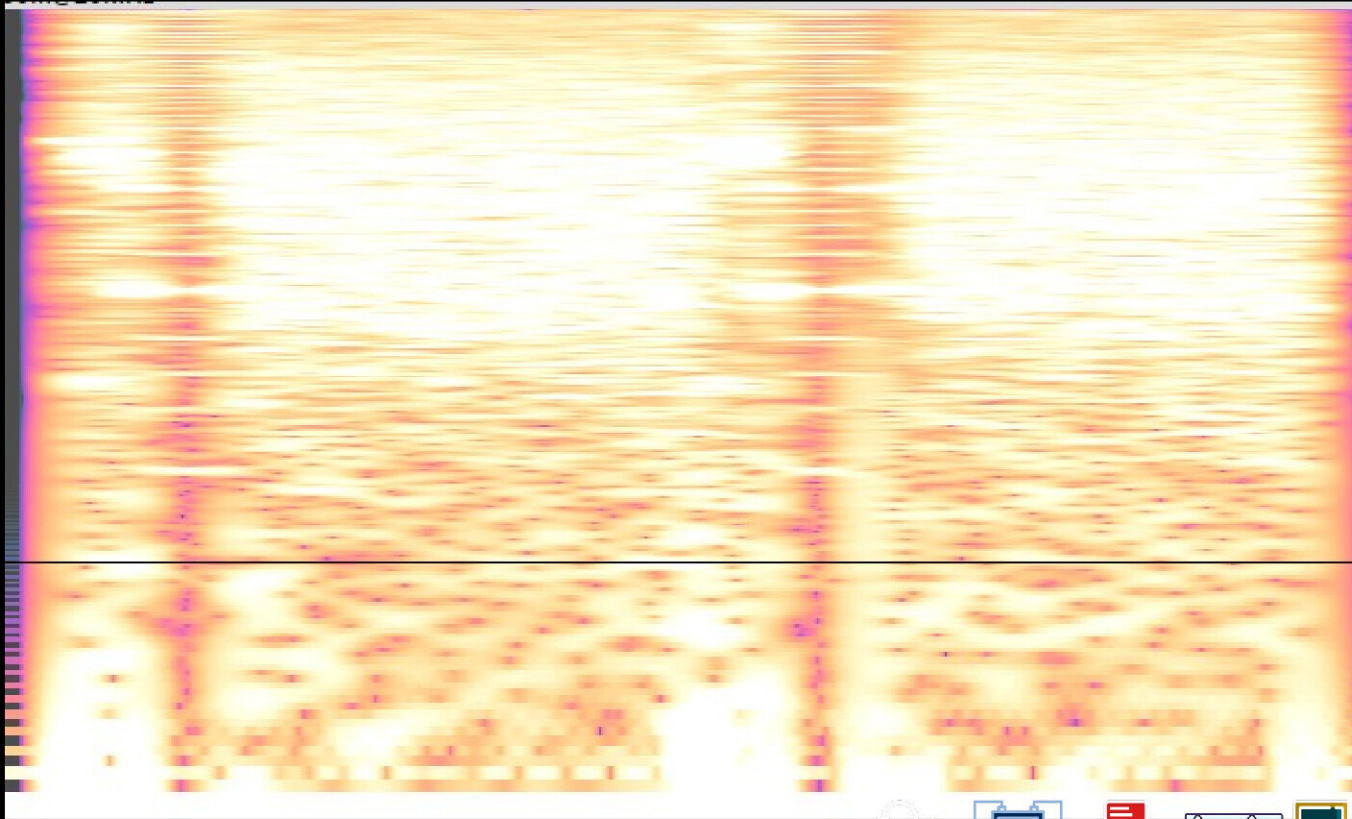
1. DETECTOR
START: 0000



Enter the number of the first telephone in the range.



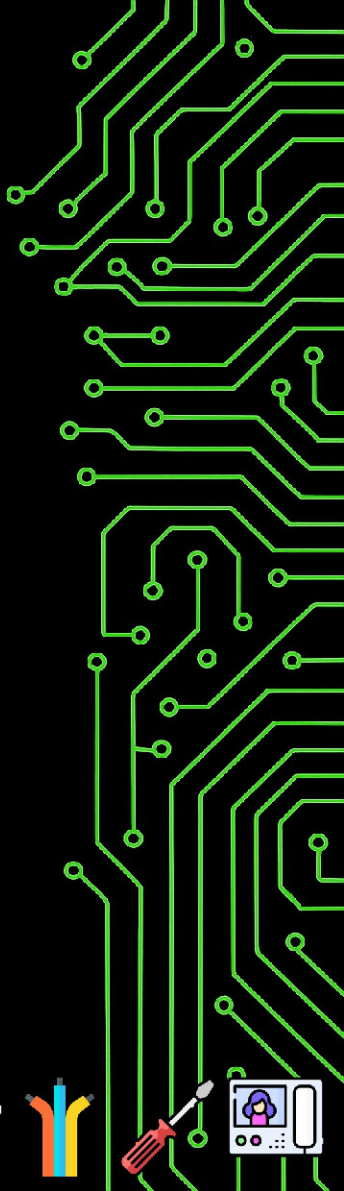
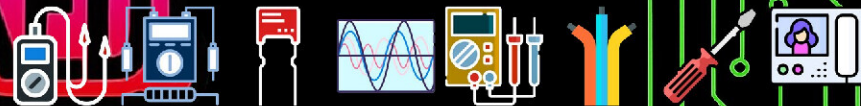
FAST FOURIER TRANSFORM



CAN WE FIND IT?



NO, IT'S FUCKED



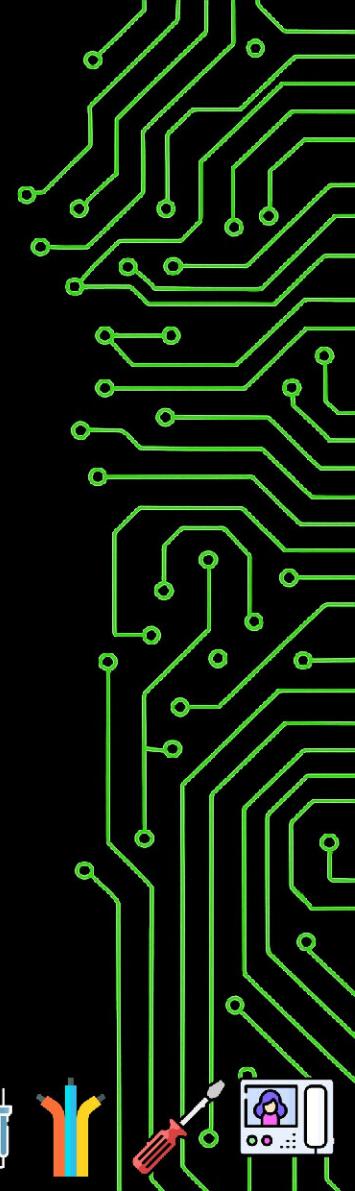
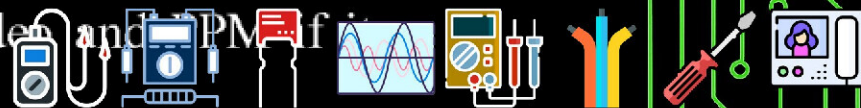
PATENTS TO THE RESCUE

The frames transmitted and received between the devices of the system, as indicated above, are **multiplexed over time**. We have six different frame types:

Data frame 1: command frame between devices. Always **BNFSK** modulation is used.

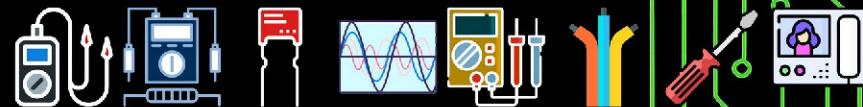
Channel 1 upstream audio (or audio+video) frame: frame with the coded and compressed audio (or audio+video) information from the caller to the recipient, corresponding to the first conversation channel. It employs **BNFSK** modulation if video is not included and **PPM** if it includes video.

Channel 1 downstream audio frame: frame with the coded and compressed audio information from the terminal receiving the call to the caller, corresponding to the first conversation channel. It employs BNFSK modulation if the rise frame does not include video and PPM if it includes video.



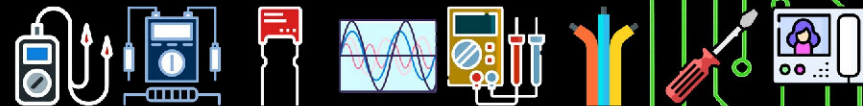
WANT MORE? TENDER SPECIFICATIONS!

- d. Provide at least 25 fps video frame.
- e. Provide at least 480p image resolution.
- f. Have option to set a fixed zoomed video image on VCU camera to a desired area.
- g. Have at least a video image angle of 120°H, 80°V.
- h. Have a video compression based on H264 high profile ISO/CEI14496-10.
- i. Have audio codecs based on G.726 and OPUS.
- j. Reach distances between VCU or SMU and URU of 500 meters without the need for repeaters.
- Audio codecs: G.726 (with telephone) and OPUS (with monitor).
- Bitrate $\geq 40\text{Kbps}$



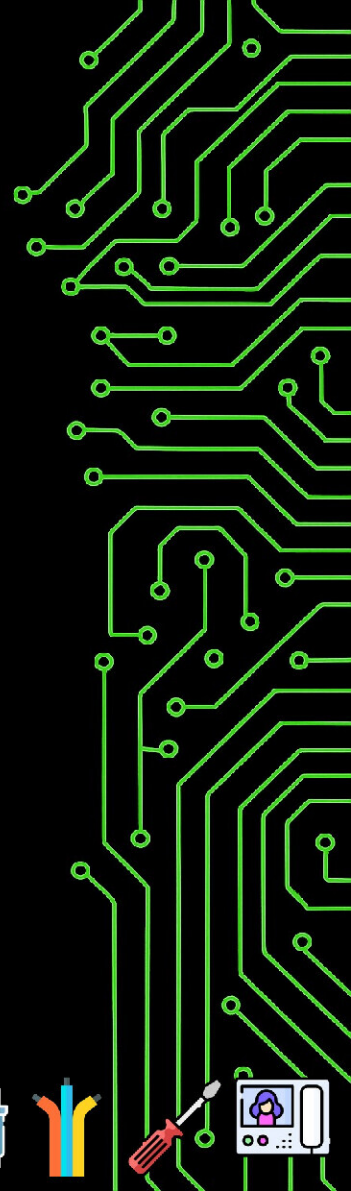
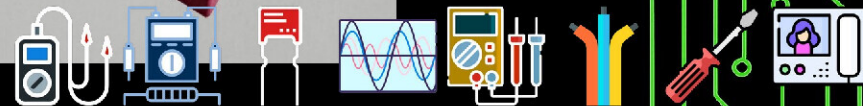
CONCLUSIONS FROM DOCUMENTATION

- modulation
 - TDM + NBFSK/PPM
- encoding
 - G.726, 40kbps
 - low enough bitrate
- encryption
 - (?)



GETTING A BIT OVERWHELMED

- at this point I'm thinking
 - can I really do it without ordering test devices?
 - yes, I think I can
 - but ...



BTW, CHEATCODES EXIST! AND ARE ON SALE

Home > Doorbells > Video Wired Doorbells > Fermax > Accessories

Fermax 3254 DUOX PLUS PROGRAMMER



Catalogue No: 3254
£339.46 ~~£499.20~~
Saving Of £159.74 (32%)
(Exc. 20% VAT)
(£407.35 Inc. VAT)
Free shipping on orders over £50

Special Order Item email for lead time

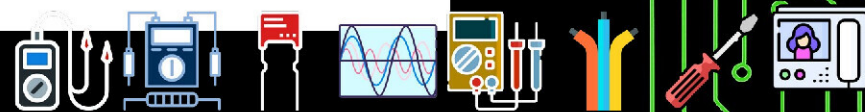
— 1 + Add to cart

 **PayPal** Pay in 3 interest-free payments of £113.15. [Learn more](#)

      Trustpilot



Source: onlinesecurityproducts.co.uk



COMMAND ANALYSER MODE (IM)

FERMAX

This function allows you to view any DUOX command transmitted by any device on the installation.

These commands are stored in the programmer's internal memory, up to a maximum of 100 entries.

To capture the commands, follow these steps indicated in the diagram:

2. ANALYZER

We start from option 2 ANALYSER from the MAIN MENU

OK

* * * * *
WAIT COMMAND

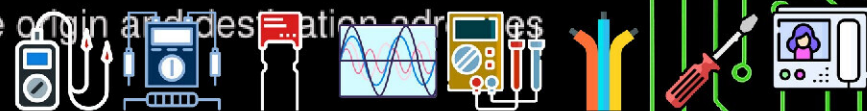
The device remains available to receive commands. WAIT COMMAND blinks.

A command is made...

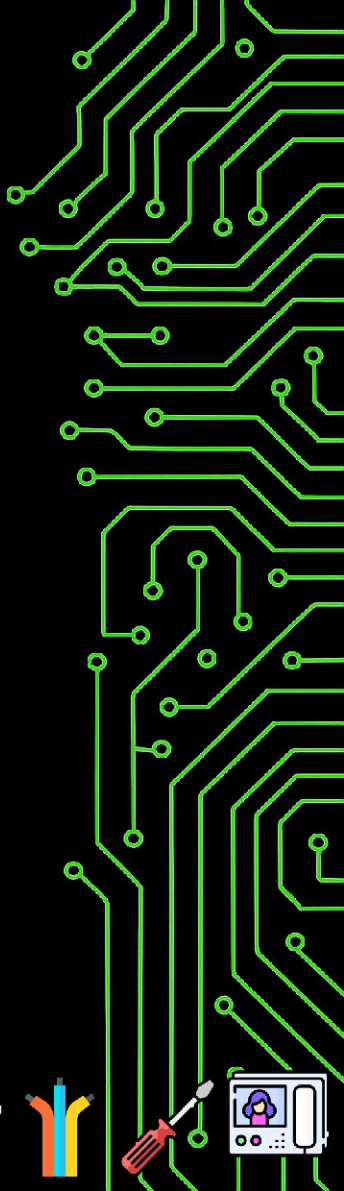
DISABL-CHANNEL 1
00F010 - FFF0FF

The command appears on the display and is saved in the memory. In the bottom line the origin and destination addresses appear, respectively.

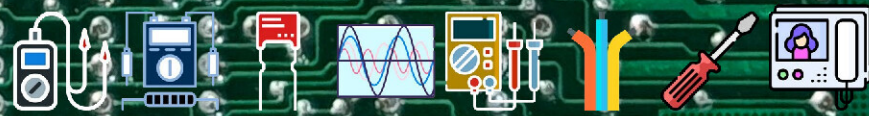
Source: Fermax



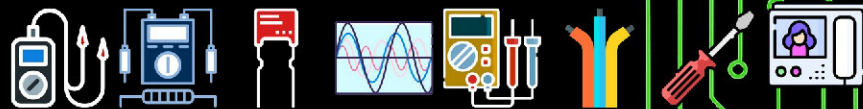
A decorative graphic on the right side of the page. It features a black background with a complex network of green lines representing a circuit board. At the bottom of this graphic, there are three icons: a red and blue cable, a red screwdriver, and a white smartphone displaying a person's face on its screen.



Reverse engineering (more) properly

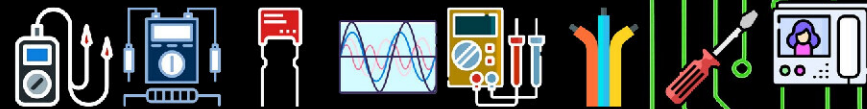


INTERMISSION ON POSSIBLE WAYS FORWARD

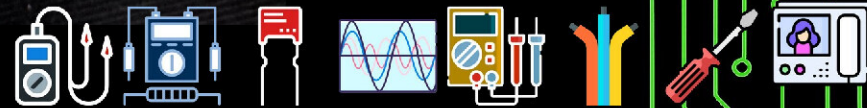


PEER REVIEW

- “your data ~~probably sucks~~ quality is likely insufficient”



A FRIEND WITH A REAL OSCILLOSCOPE



FIRST ORDER OF BUSINESS: CAP

MSO5104 Mon Aug 18 00:08:12 2025

RIGOL

STOP

H

10.0us

50MSa/s
50Mpts

Measure

STOP/RUN

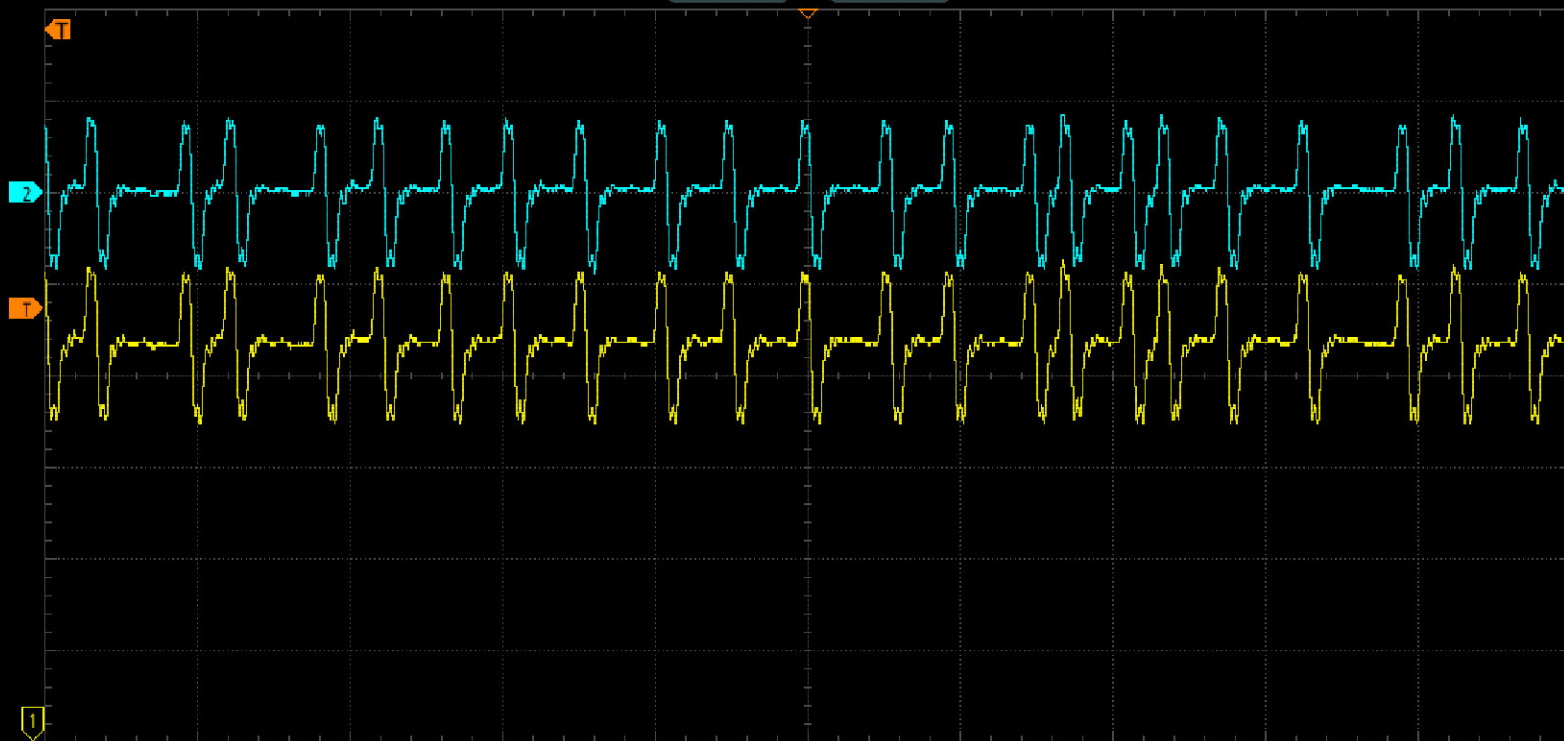
D

541.5213ms

T

24.2V

N



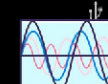
1 2.00V B
-22.7V

2 2.00V B
+4.00V

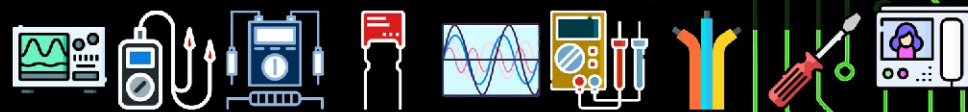
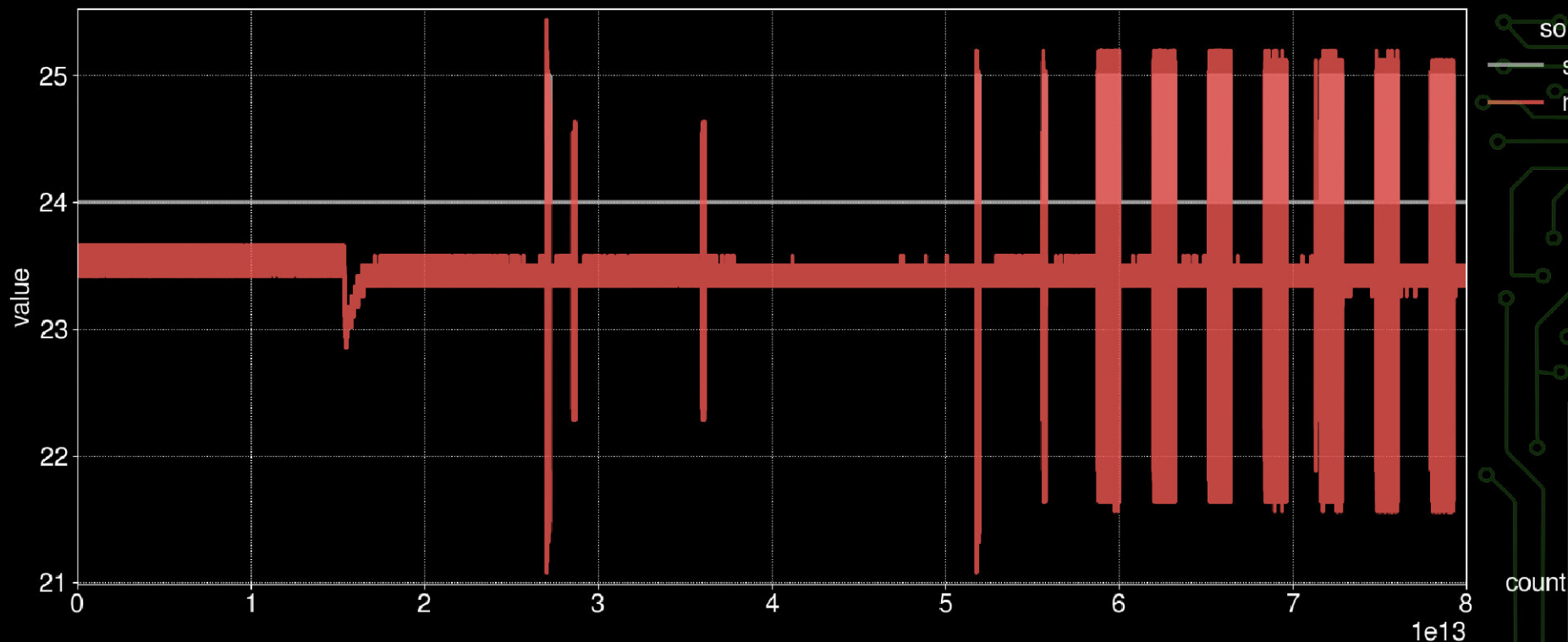
3 200mV B
-360mV

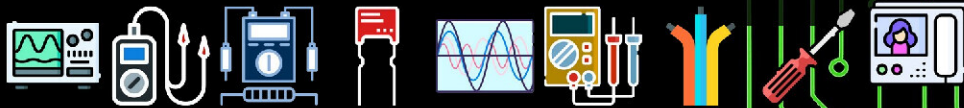
4 100V B
0.00V

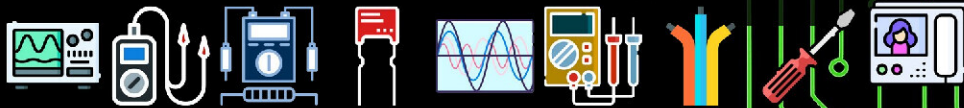
0 1 2 3 4 5 6 7
9 10 11 12 13 14 15

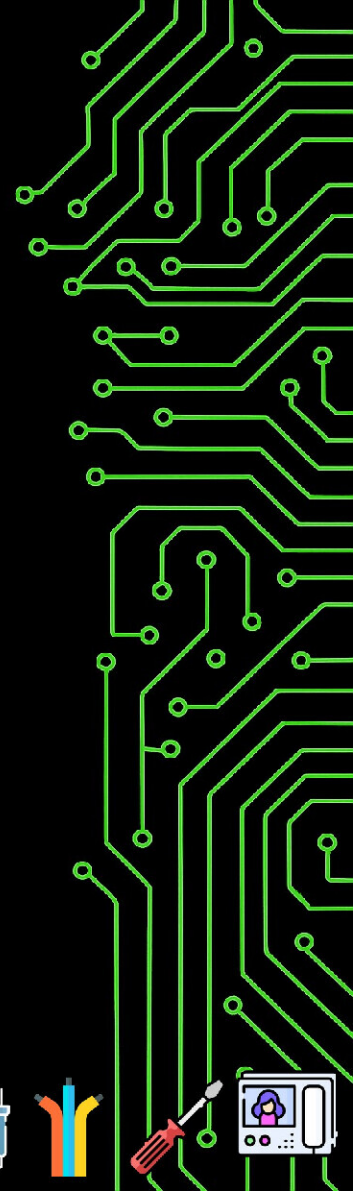
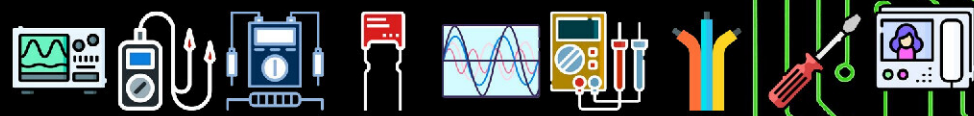


COMPARE RIGOL WITH SALEAE

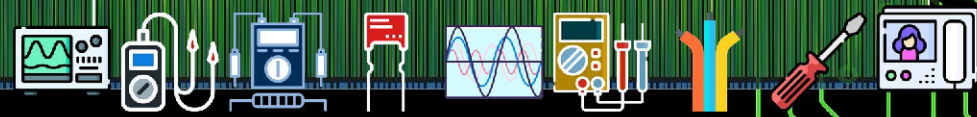




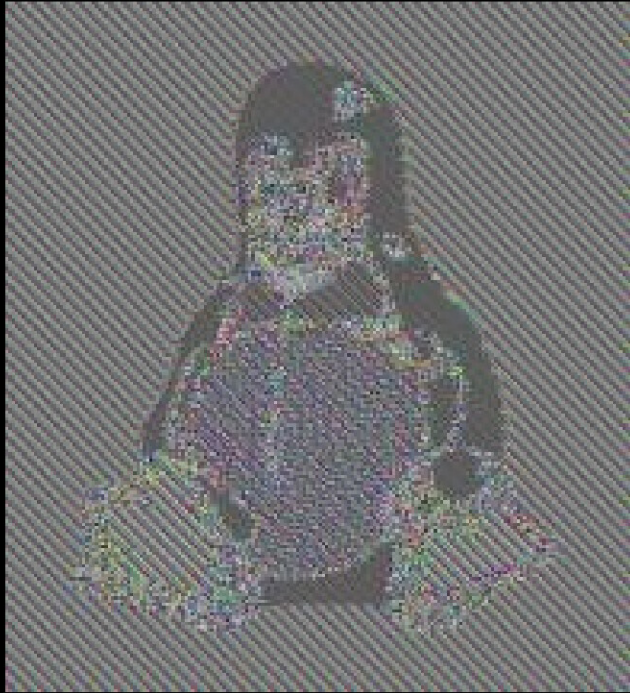




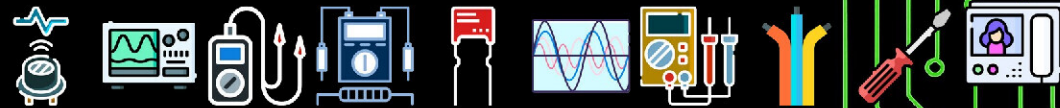
5.713 14284800



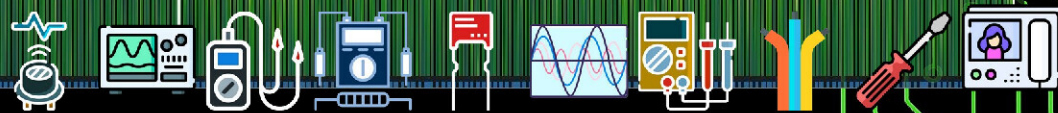
ECB PENGUIN



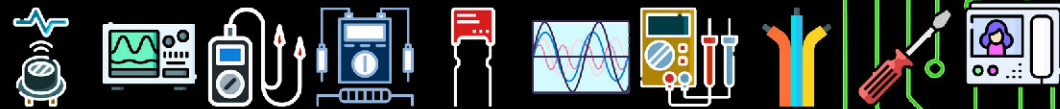
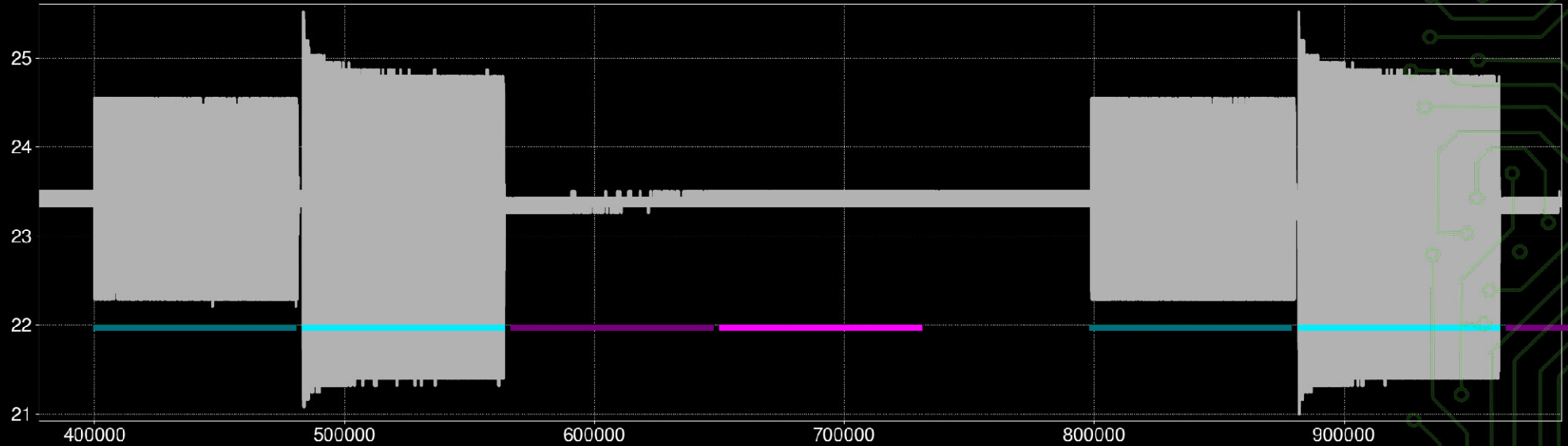
- remember Tux!



5.713 14284800

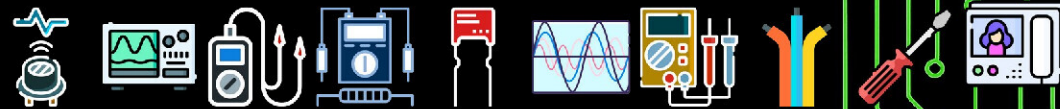


TIME-DIVISION MULTIPLEXING

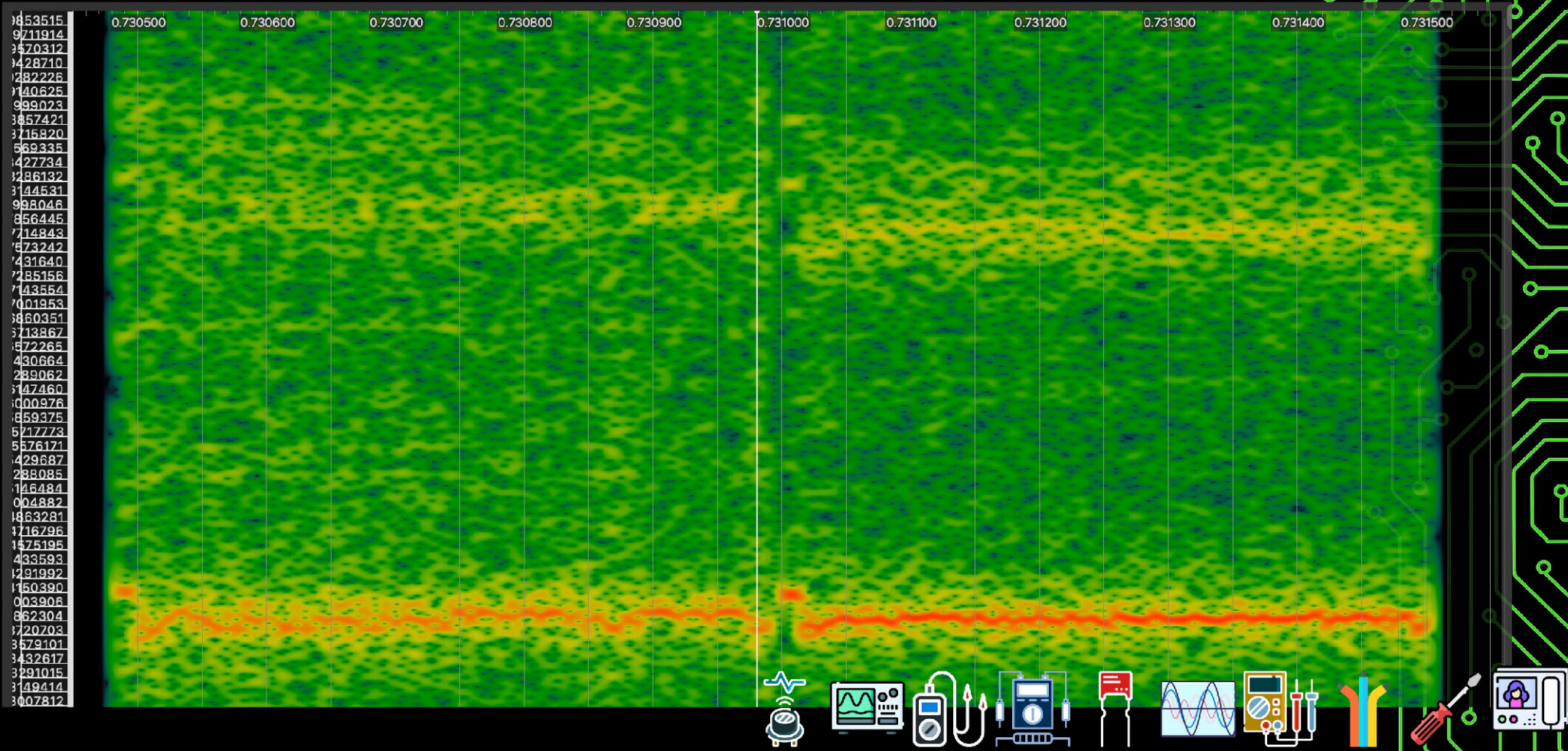


BINARY NARROW FREQUENCY SHIFT KEYING

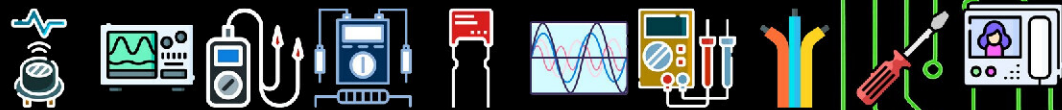
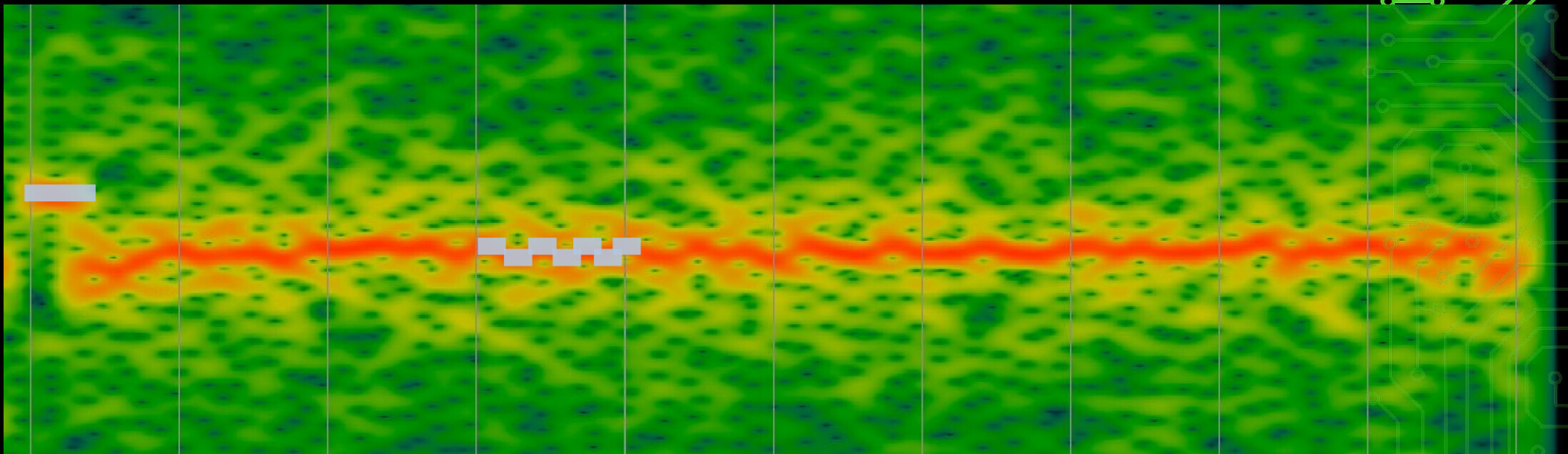
The BNFSK modulation uses a pair of frequencies to represent each one of the two values of one bit, plus the third frequency for the frame start alert.



BINARY NARROW FREQUENCY SHIFT KEYING



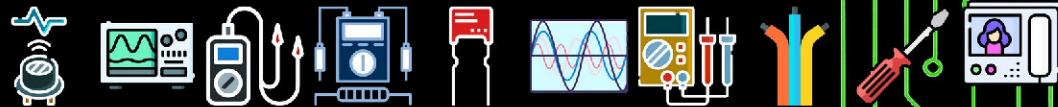
BINARY NARROW FREQUENCY SHIFT KEYING



BINARY NARROW FREQUENCY SHIFT KEYING

The BNFSK modulation uses a pair of frequencies to represent each one of the two values of one bit, plus the third frequency for the frame start alert.

- $f_0=2.3$ MHz
- $f_1=2.5$ MHz
- $f_s=2.6$ MHz

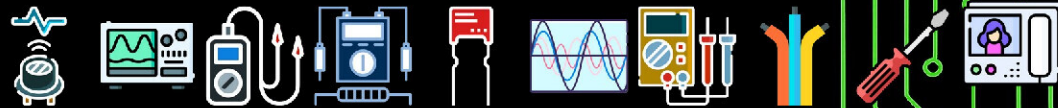


BNFSK PROCESSING

Butterworth filter =>

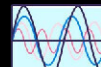
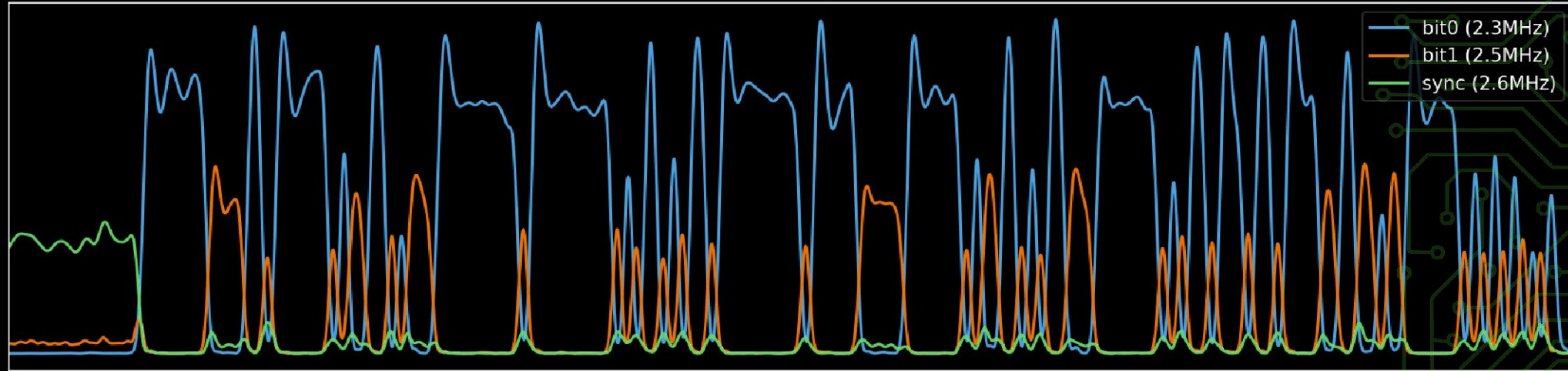
Hilbert transform =>

Instantaneous power



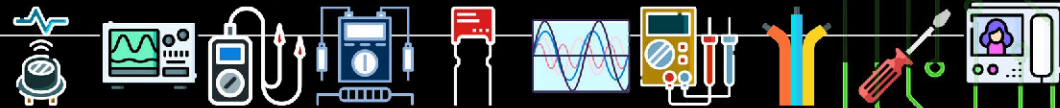
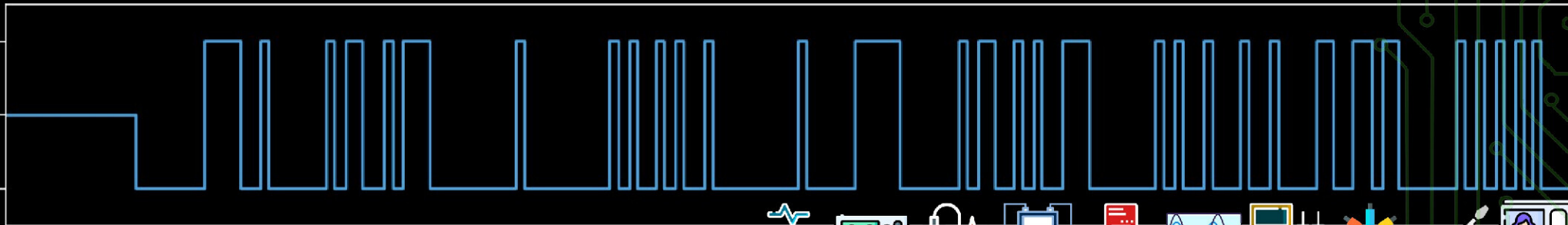
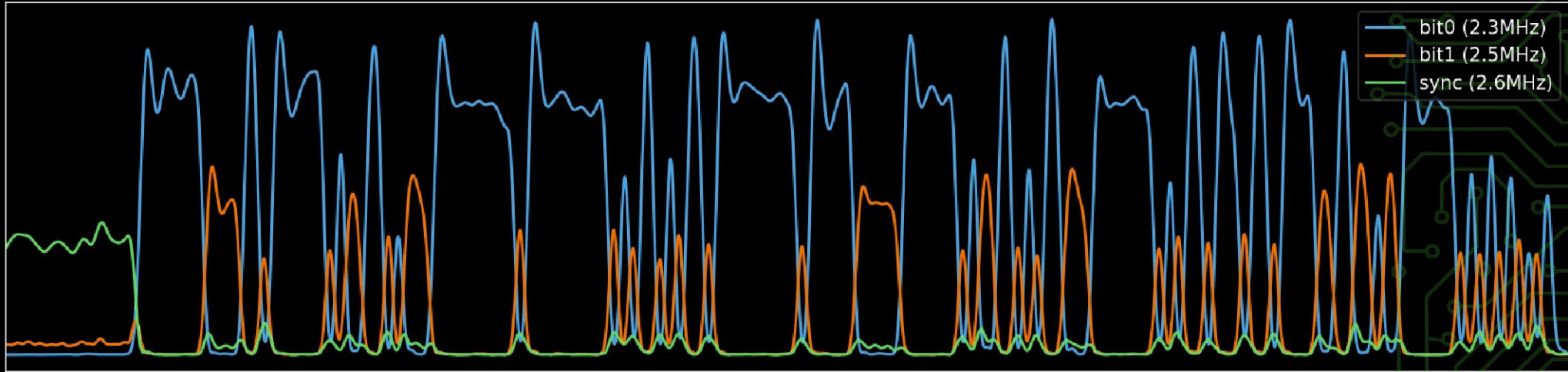
POWER @ F_0 , F_1 , F_S

Burst 0 @ 690.1ms

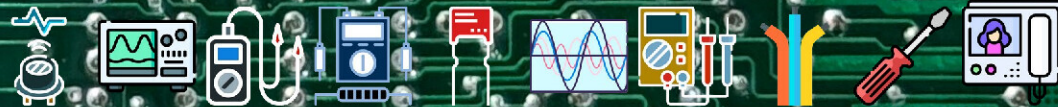


WE'VE GOT SYMBOLS AND BITS!

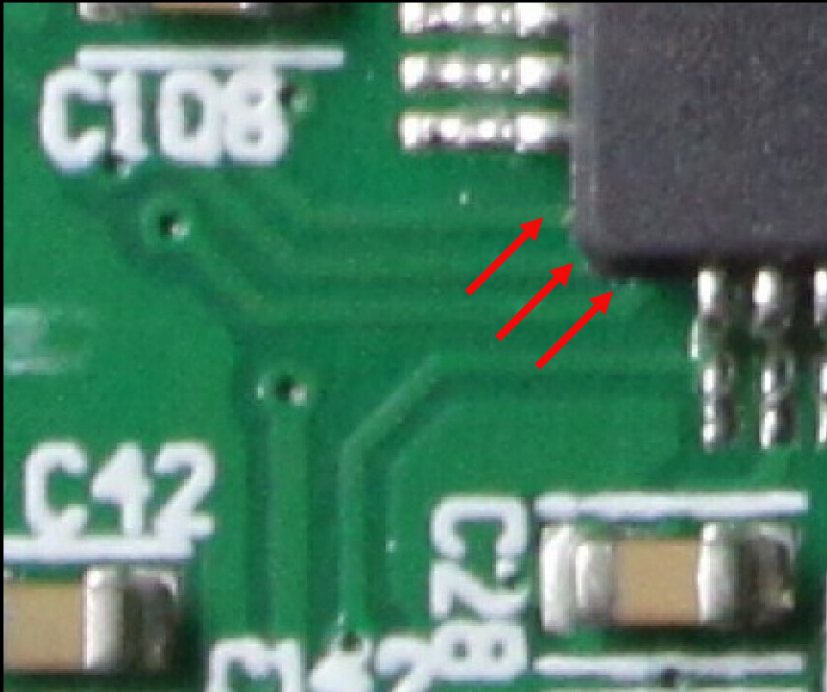
Burst 0 @ 690.1ms



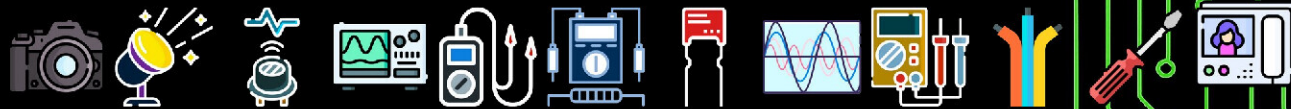
Reverse engineering the circuit board



LIMITATIONS



- x-ray or heatplate desolder
 - need two devices
- *alt.:* continuity probing



FERMAX
924728-RT01-28904

volume
switch

INPUTS AND
POWER
CIRCUIT

OP-
AMP

HS
conn

OP-
AMP

8MHz
clock

door
rele
ase

HS HOOK

SWD*

ARM
nuvoton
M453RE
6AE

prog

OP-
AMP

TOP

2

a ton of
test points
on the
bottom



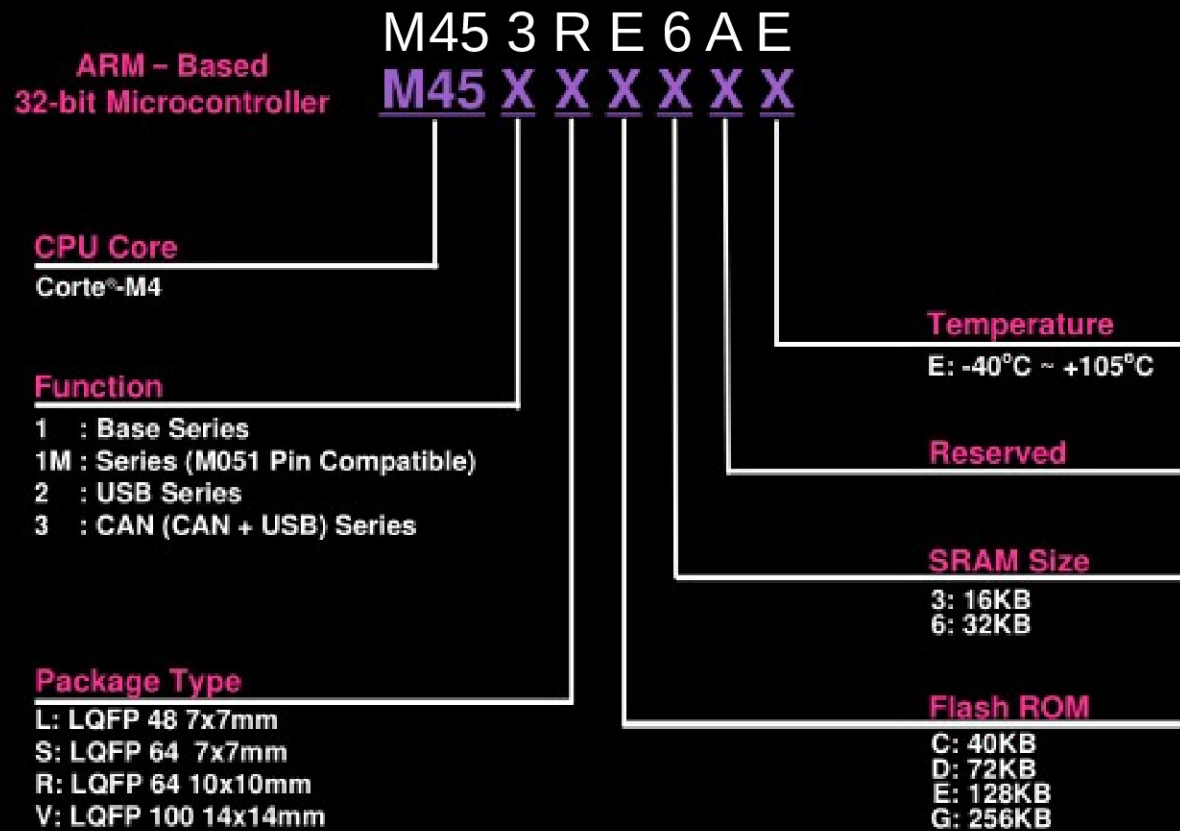
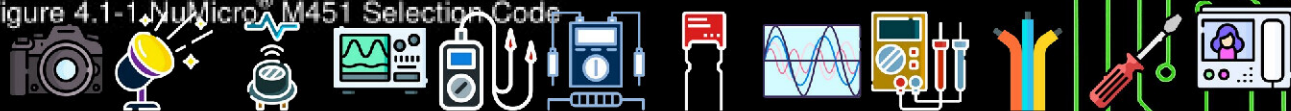


Figure 4.1-1. NuMicro® M451 Selection Code



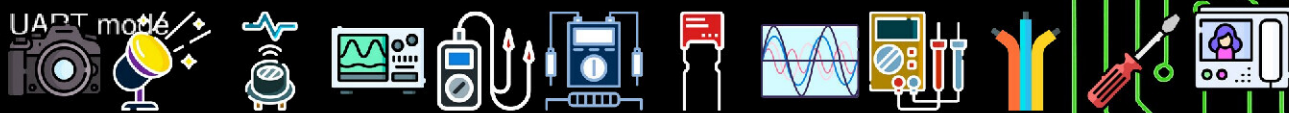
4.1.5 NuMicro® M453 CAN Series (CAN+USB) Selection Guide

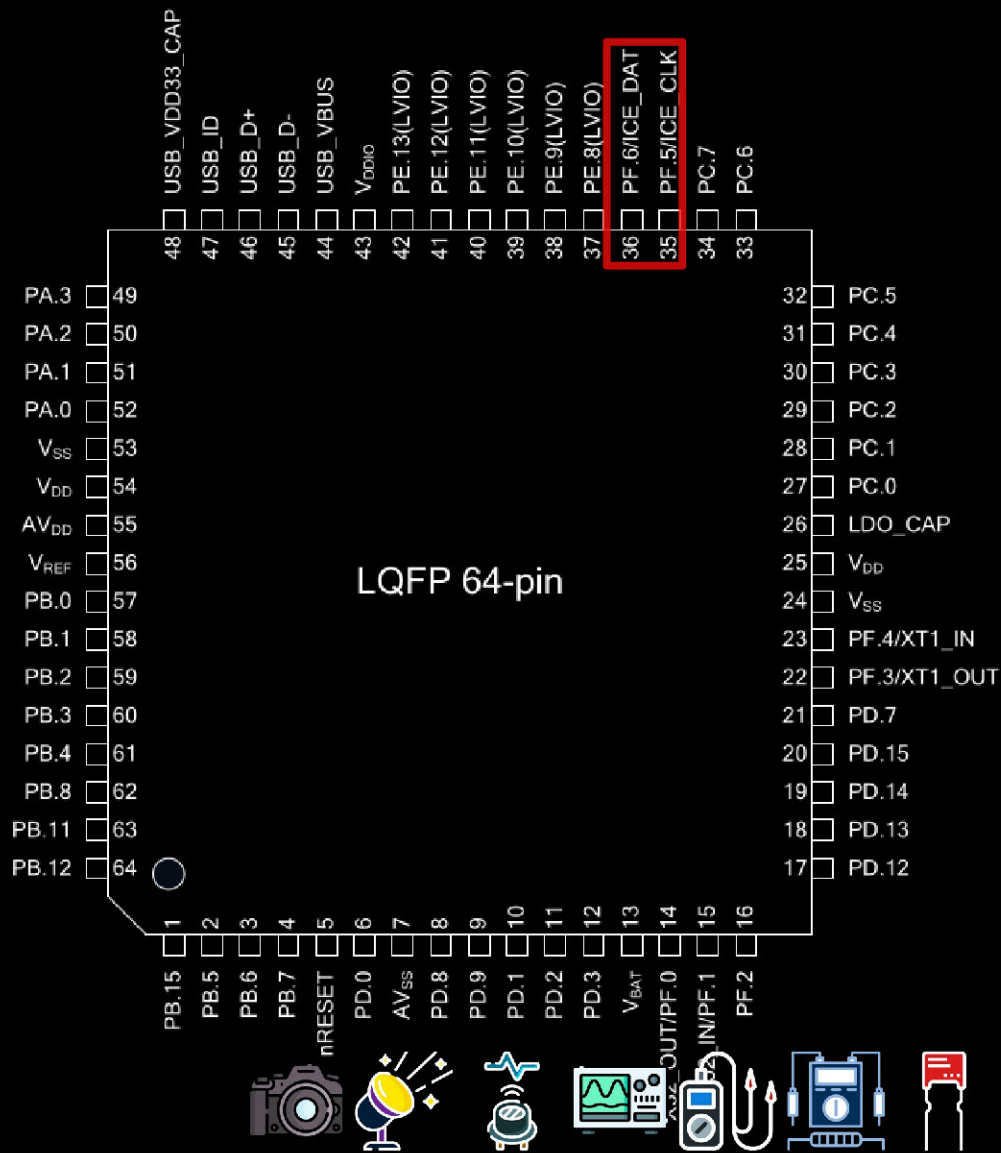
Part Number	Flash (KB)	SRAM (KB)	ISP Loader ROM (KB)	I/O	Timer	Connectivity						I ² S	USB	PWM	Analog Comp.	DAC (12-Bit)	ADC (12-Bit)	RTC	EBI	ICP/ISP/AP	Package
						UART*	SC* (ISO-7816)	SPI	I ² C	CAN	LIN										
M453LG6AE	256	32	4	34	4	3+1	1	3	2	√	2	2	OTG	10	2	1	8-ch	√	√	√	LQFP 48
M453LE6AE	128	32	4	34	4	3+1	1	3	2	√	2	2	OTG	10	2	1	8-ch	√	√	√	LQFP 48
M453RG6AE	256	32	4	48	4	4+1	1	3	2	√	2	2	OTG	12	2	1	12-ch	√	√	√	LQFP 64
M453RE6AE	128	32	4	48	4	4+1	1	3	2	√	2	2	OTG	12	2	1	12-ch	√	√	√	LQFP 64
M453VG6AE	256	32	4	80	4	4+1	1	3	2	√	2	2	OTG	12	2	1	16-ch	√	√	√	LQFP 100
M453VE6AE	128	32	4	80	4	4+1	1	3	2	√	2	2	OTG	12	2	1	16-ch	√	√	√	LQFP 100
M453LD3AE	72	16	4	35	4	4+1	1	2	2	√	2	1	Device	10	2	1	10-ch	√	√	√	LQFP 48
M453LC3AE	40	16	4	35	4	4+1	1	2	2	√	2	1	Device	10	2	1	10-ch	√	√	√	LQFP 48
M453RD3AE	72	16	4	49	4	4+1	1	2	2	√	2	1	Device	12	2	1	16-ch	√	√	√	LQFP 64
M453VD3AE	72	16	4	72	4	4+1	1	2	2	√	2	1	Device	12	2	1	16-ch	√	√	√	LQFP 100

*Marked in this table (4+1) means 4 UART + 1 SC UART

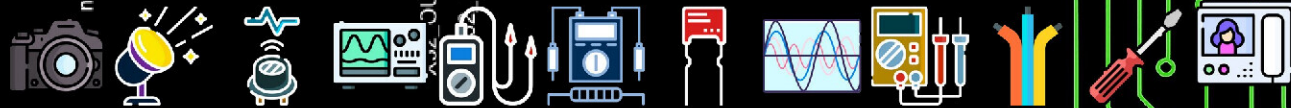
*SC (ISO-7816) supports full duplex UART mode

Source: nuvoton

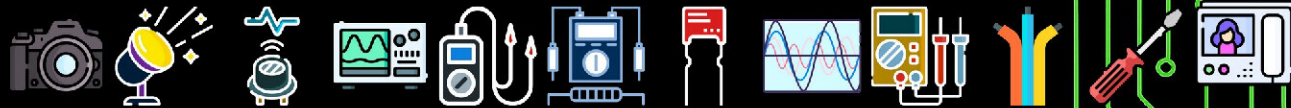
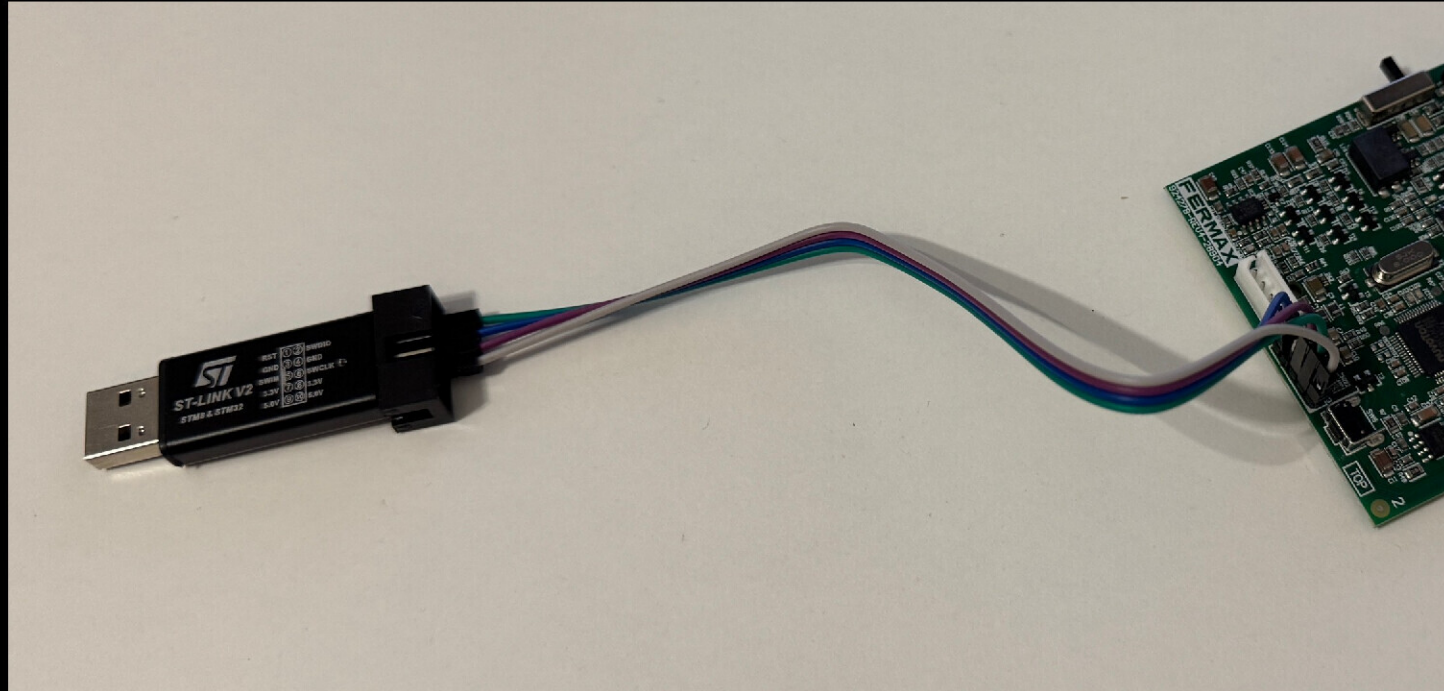




Source: nuvoton



QUICK SOLDER JOB LATER ...



Reverse engineering the firmware

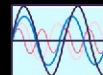
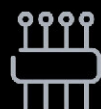


DUMPED

- Size: 131072 Blocks: 256
IO Block: 4096 regular file
- dump.bin: ARM Cortex-M firmware,
initial SP at 0x20008000, reset at
0x000115b8, NMI at 0x00011624,
HardFault at 0x00010fb8, SVCall at
0x00011624, PendSV at 0x00011624



BINWALK-E



WHAT'S INSIDE

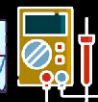
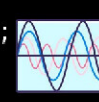
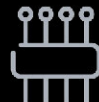
- More than 100 strings, mostly debug
- 2 unique paths
 - ../User/libspeexdsp/kiss_fftr.c
 - /mnt/workspace/workspace/GCC-10-pipeline/jenkins-GCC-10-pipeline-338_20211018_1634516203/src/newlib/newlib/libc/stdlib/rand.c



PROGRAMMING MODE

- command 0x33AA
- command 0x0088

```
51 switch(command) {  
52 case 0x33:  
53     if (command_2 == 0x40) {  
54         DAT_20000b54 = 1;  
55         return;  
56     }  
57     if (0x40 < command_2) {  
58         if (command_2 == 0xdd) {  
59             DAT_20000b52._1_1_ = 0;  
60             return;  
61         }  
62         if (0xdd < command_2) {  
63             if (command_2 != 0xee) {  
64                 DAT_20000abc = 0;  
65                 DAT_20000abd = 0;  
66                 return;  
67             }  
68             DAT_20000abc = 0;  
69             DAT_20000abd = 0;  
70             DAT_20000b52._1_1_ = 1;  
71             return;  
72         }  
73         if (command_2 == 0xaa) {  
74             uVar2 = get_duox_my_direction();  
75             FUN_00009a60(uVar2,0xffff0ff,0x88,0);  
76             DAT_2000130c = 3;  
77             SetEventState(9,8);  
78             SetEventState(10,8);  
79             if (command_2 == 0x88) {  
80                 DAT_20000b54 = 1;  
81                 return;  
82             }  
83         }  
84     }  
85 }
```



THERE ARE CHECKSUMS

xor_checksum = XOR of all 32-bit words

sum_checksum = byte-wise additive
checksum over all words



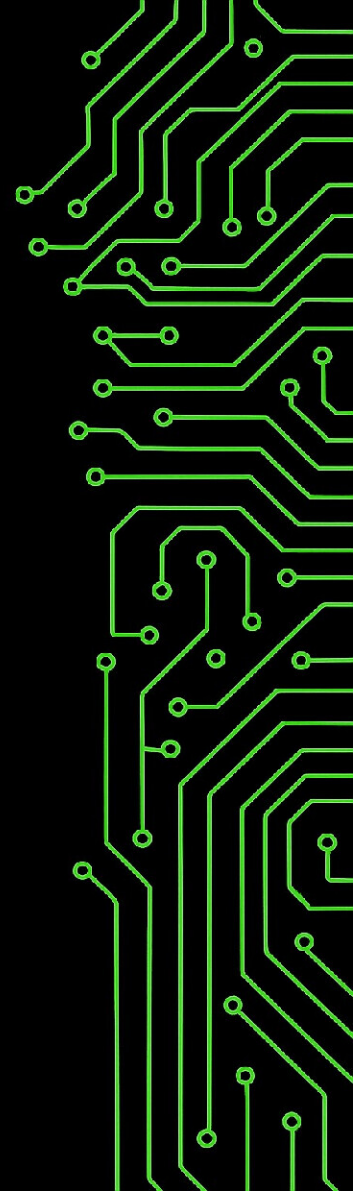


Microphone-to-wire

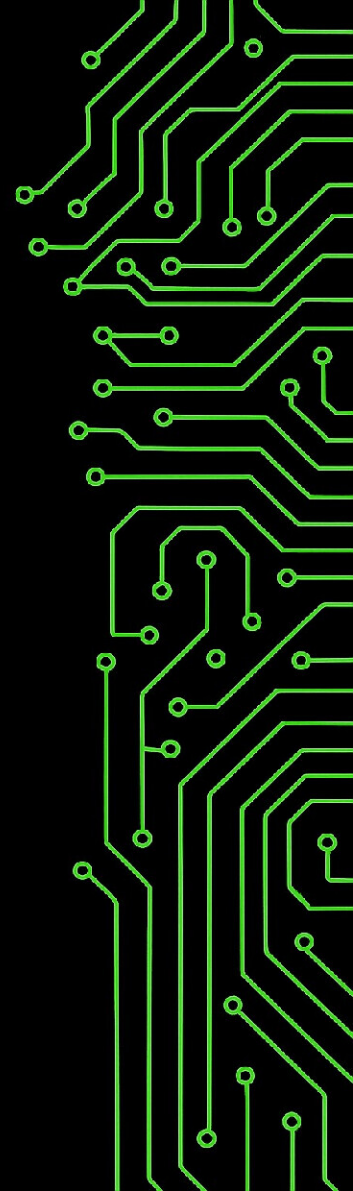

```

2 uint codec_gain_control_select(int param_1,int param_2,undefined4 param_3)
3
4 {
5     uint uVar1;
6     int iVar2;
7     int iVar3;
8     undefined4 uVar4;
9     uint uVar5;
10    uint uVar6;
11
12    if (param_2 == 2) {
13        param_1 = DecodePackedFixedPointValue();
14        param_1 = param_1 >> 2;
15    }
16    else if (param_2 == 3) {
17        param_1 = param_1 >> 2;
18    }
19    else {
20        if (param_2 != 1) {
21            return 0xffffffff;
22        }
23        param_1 = FUN_0000fcac();
24        param_1 = param_1 >> 2;
25    }
26    iVar2 = codec_subband_energy_sum(param_3);
27    iVar3 = codec_dual_subband_energy(param_3);
28    iVar3 = iVar3 + iVar2 >> 1;
29    uVar4 = FUN_0000efc8(param_3);
30    uVar5 = FUN_0000f000(param_1 - iVar3,uVar4,&DAT_200000e0,3);
31    uVar6 = FUN_0000f104(uVar5 & 4,(int)*(short *)(&DAT_00017944 + uVar5 * 2),uVar4);
32    uVar1 = uVar6;
33    if ((int)uVar6 < 0) {
34        uVar1 = -(uVar6 & 0x3fff);
35    }
36    codec_agc_state_update
37        (3,uVar4,(int)*(short *)(&DAT_00017964 + uVar5 * 2),
38         (int)*(short *)(&DAT_00017954 + uVar5 * 2),uVar6,iVar3 + uVar1,
39         (iVar3 + uVar1 + (iVar2 >> 1)) - iVar3,param_3);
40    return uVar5;
41 }
42

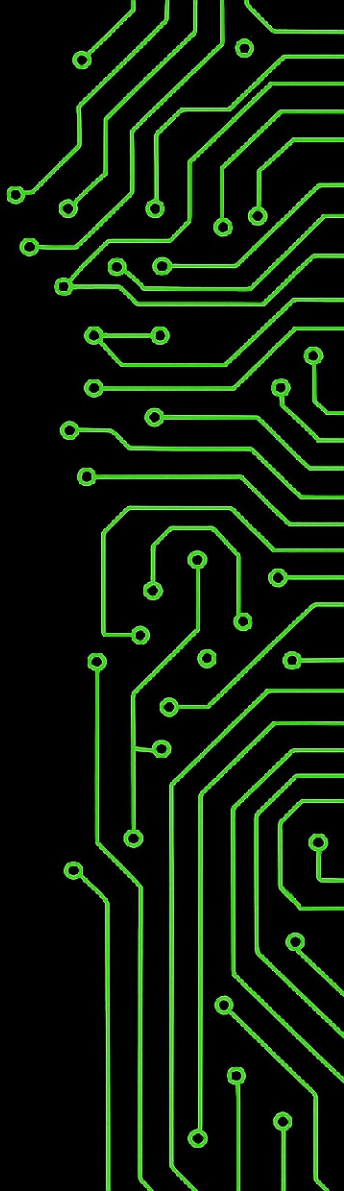
```

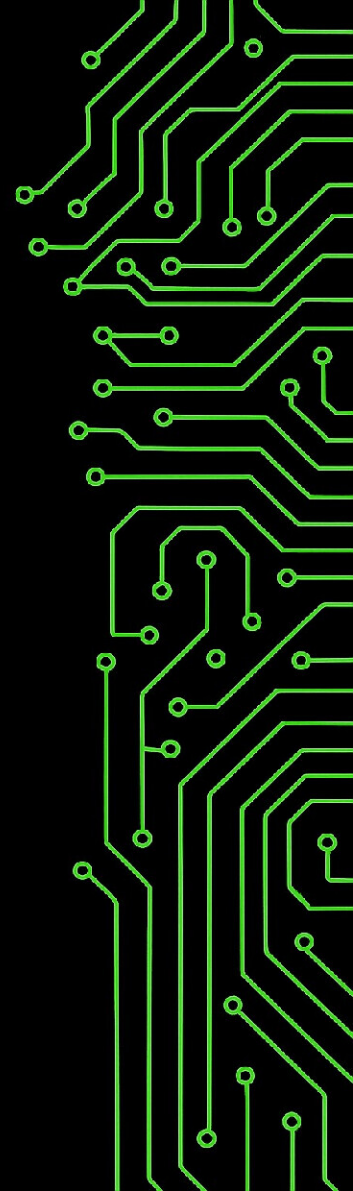
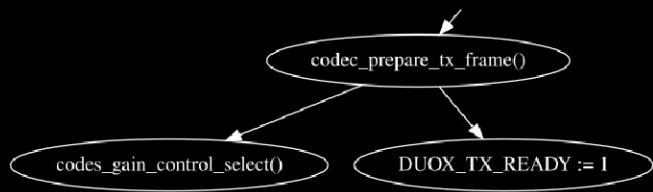


`codes_gain_control_select()`



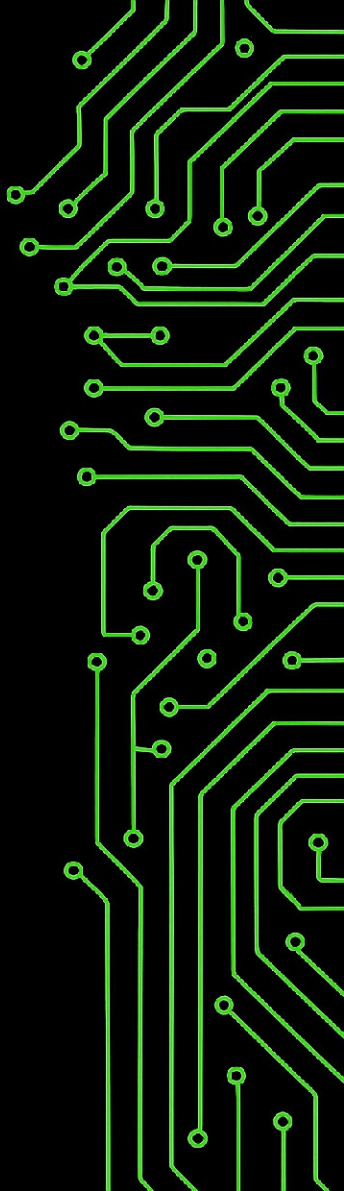

```
56 do {
57     puVar8 = &uStack_29;
58     psVar7 = psVar11;
59     do {
60         psVar7 = psVar7 + 1;
61         uVar4 = codec_gain_control_select((int)*psVar7,3,&DAT_20000a00);
62         puVar8 = puVar8 + 1;
63         *puVar8 = uVar4;
64     } while (puVar8 != local_28 + 7);
65     puVar10 = (undefined4 *)((int)puVar9 + 3);
66     FUN_0000ed40(local_28,puVar9);
67     psVar11 = psVar11 + 8;
68     puVar9 = puVar10;
69 } while (puVar10 != &DAT_200004ac);
70 MAIN_LOOP_CUSTOM_EXEC_ADDRESS = 0x518d;
71 DUOX_TX_READY = 1;
72 return;
73 }
74
```

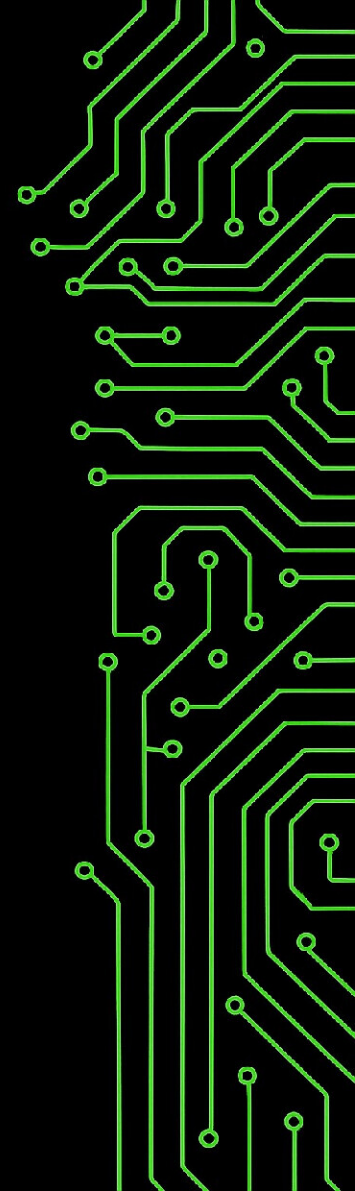
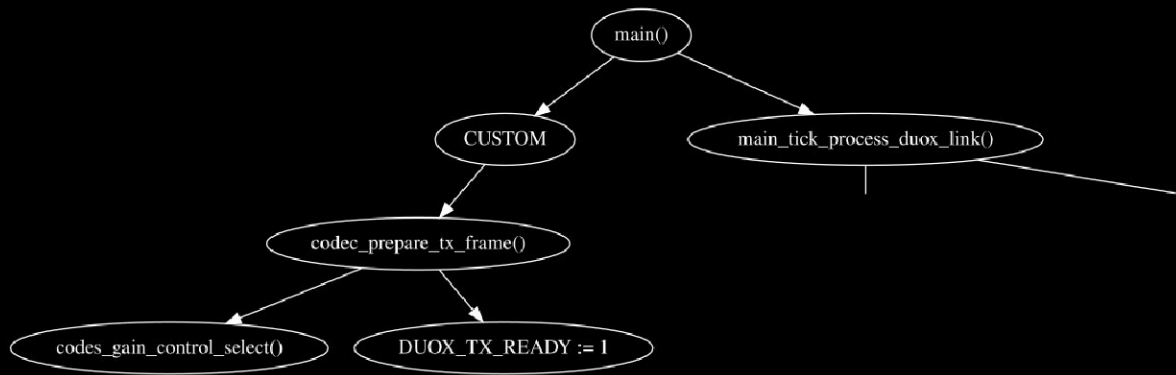




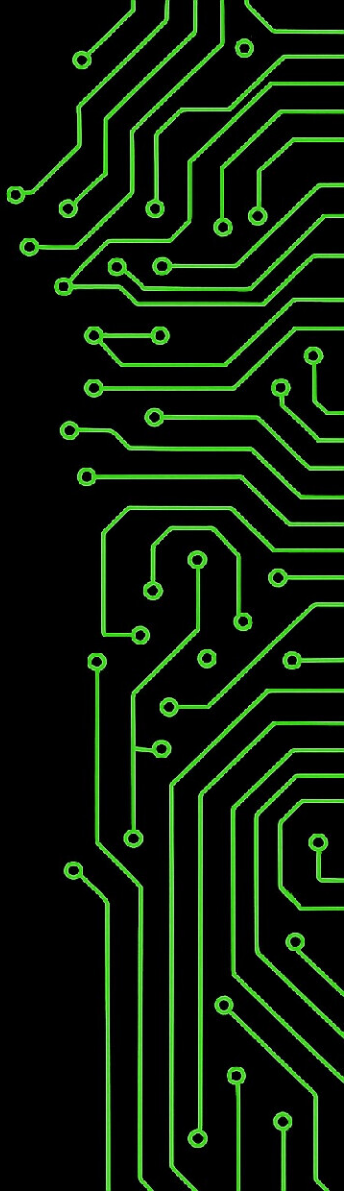
MAIN() LOOP

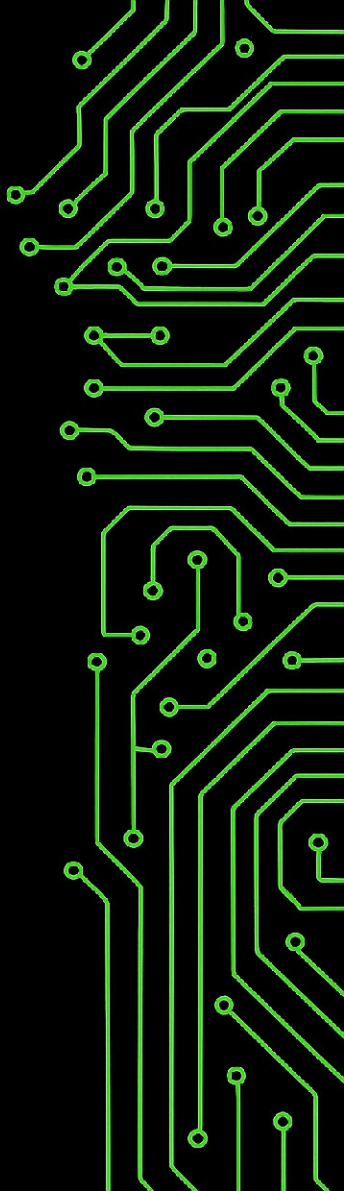
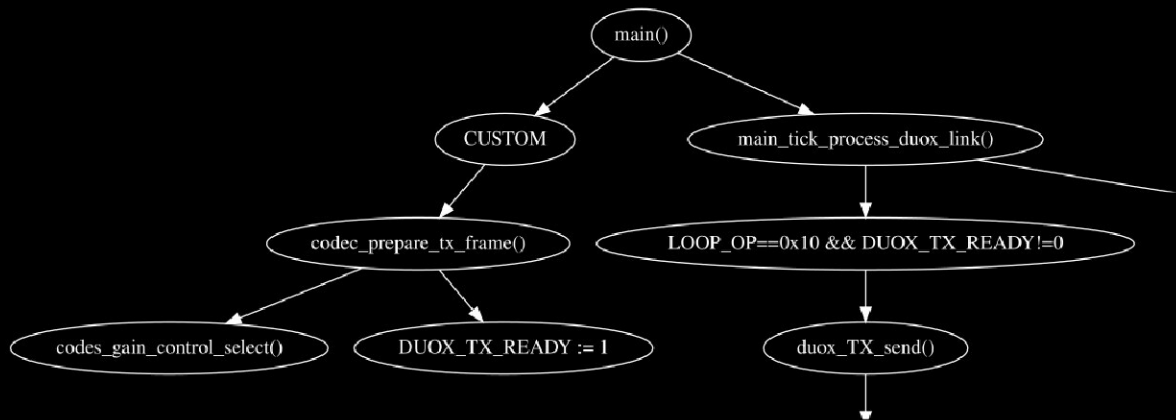
```
do {  
    delay_500ns();  
    main_tick_process_duox_link();  
    (*MAIN_LOOP_CUSTOM_EXEC_ADDRESS)();  
} while( true );
```



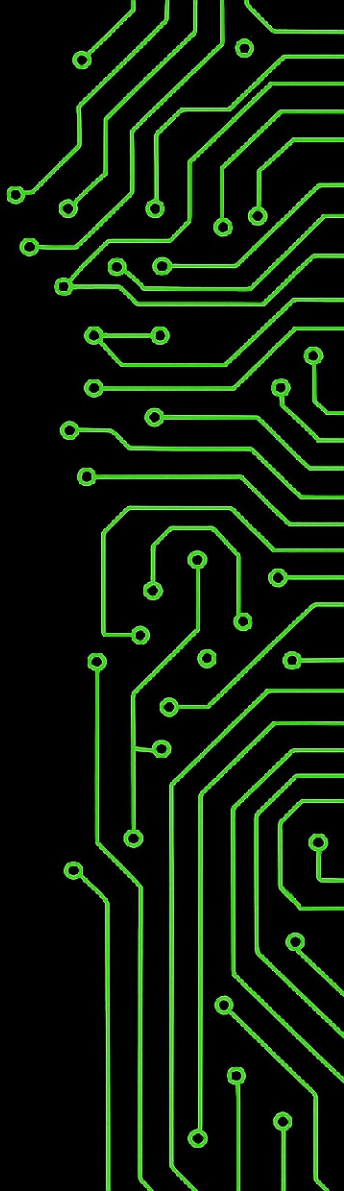


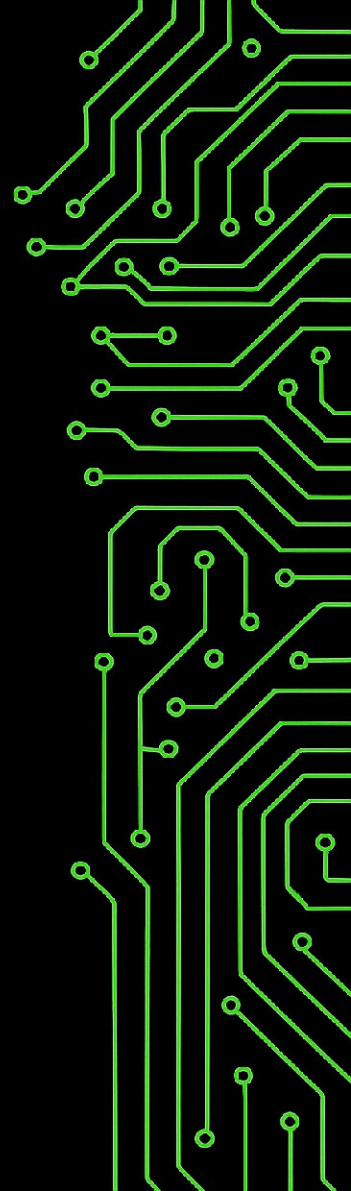
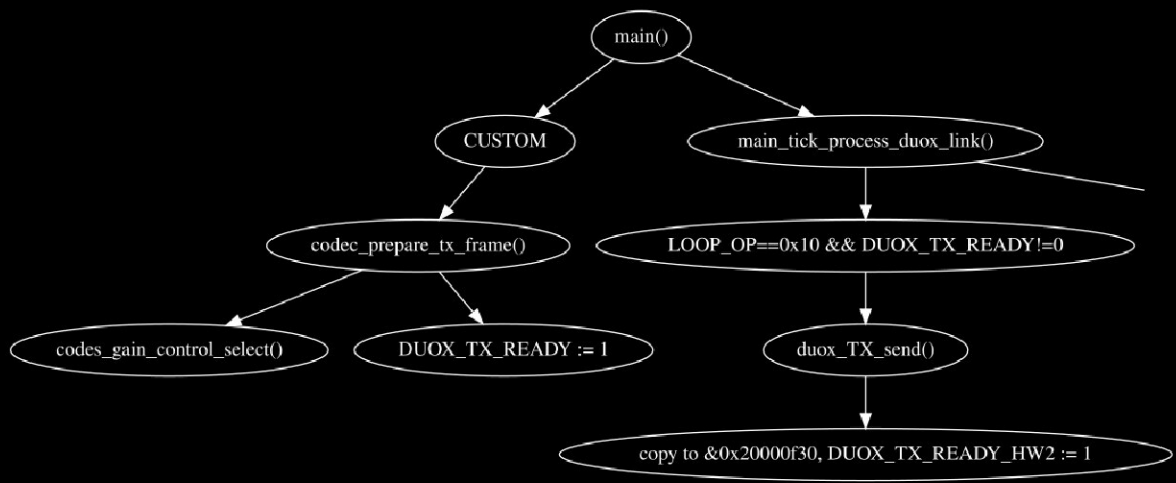
```
7 void main_tick_process_duox_link(void)
8
9 {
10     int iVar1;
11     uint uVar2;
12
13     switch(L00P_0P) {
14     case 0:
15         DUOX_parse_RX_packet();
16         iVar1 = duox_get_proto_version();
17         if (iVar1 == VERSION_ILOFT) {
18             process_event_iloft();
19         }
20         else {
21             process_event_veo();
22         }
23     case 6:
24     case 0x10:
25         if (DUOX_TX_READY_HW1 != '\0') {
26             DUOX_TX_READY_HW1 = '\0';
27             PWM_TX_hardware_1();
28         }
29         break;
30     case 1:
31         iVar1 = get_my_status();
32         if (((iVar1 != 1) && (DAT_20001310 == 0)) && (uVar2 = get_timer0(), 0x4c < uVar2)) &&
33             ((iVar1 = duox_get_active_channels(), iVar1 == 1 ||
34              (iVar1 = duox_get_active_channels(), iVar1 == 3))) {
35 LAB_0000a41c:
36             duox_TX_send();
37         }
38         break;
```

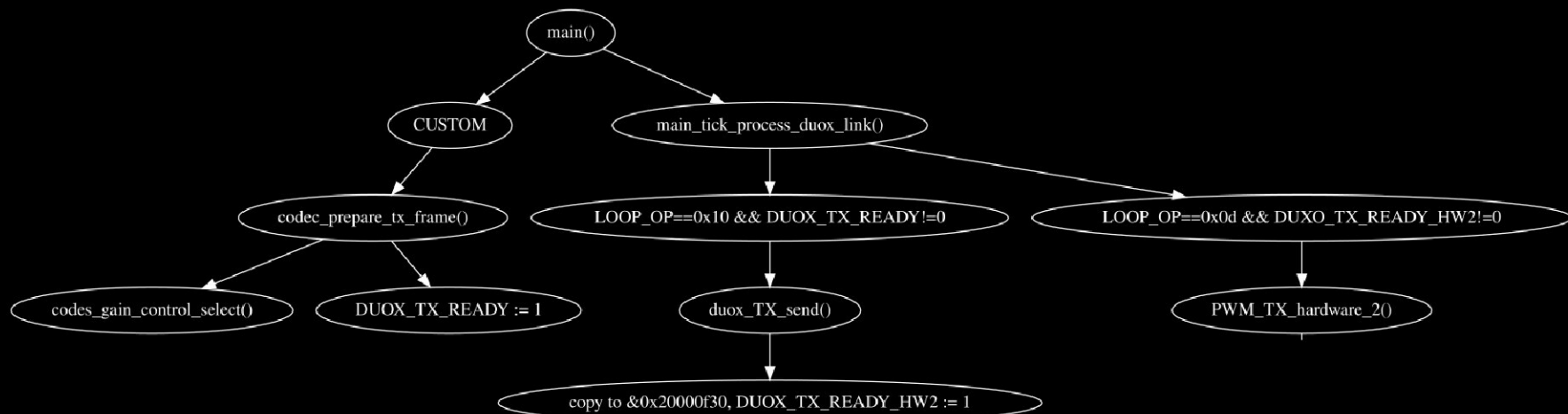




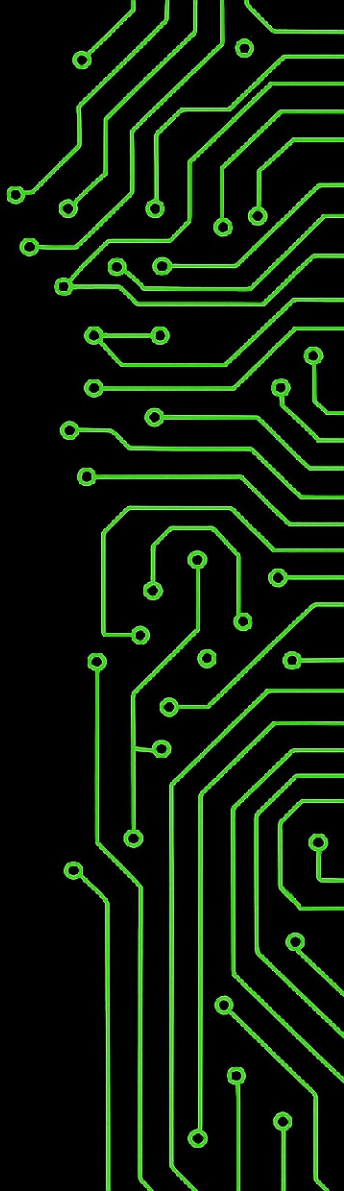

```
1
2 void duox_TX_send(void)
3
4 {
5     uint uVar1;
6     ushort *puVar2;
7     byte *pbVar3;
8
9     if (((DUOX_TX_READY != 0) && (DUOX_TX_READY_HW2 != 1)) &&
10         (uVar1 = duox_get_active_channels(), uVar1 != 0)) {
11         pbVar3 = (byte *)0x2000046f;
12         puVar2 = &DAT_20000f34;
13         DAT_20000f30 = 0x20403c0;
14         DAT_20000f34 = 0x362;
15         if ((uVar1 & 0xfd) != 1) {
16             DAT_20000f34 = 0x364;
17         }
18         do {
19             pbVar3 = pbVar3 + 1;
20             puVar2 = puVar2 + 1;
21             *puVar2 = (ushort)*pbVar3 << 1 | 0x200;
22         } while (pbVar3 != &DAT_200004ab);
23         DAT_20000fae = 0x2aa0206;
24         DUOX_TX_READY_HW2 = 1;
25         DUOX_TX_READY = 0;
26     }
27     return;
28 }
29
```



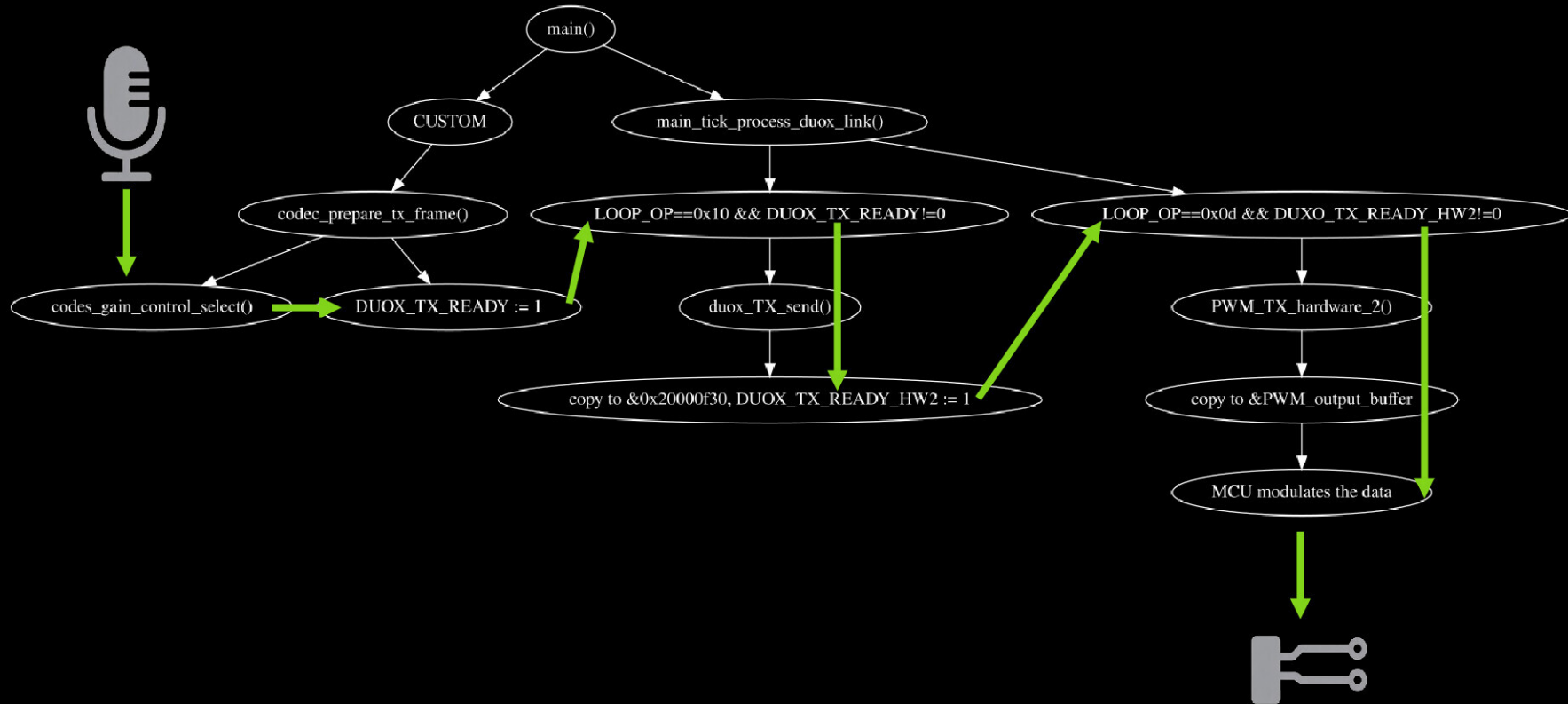


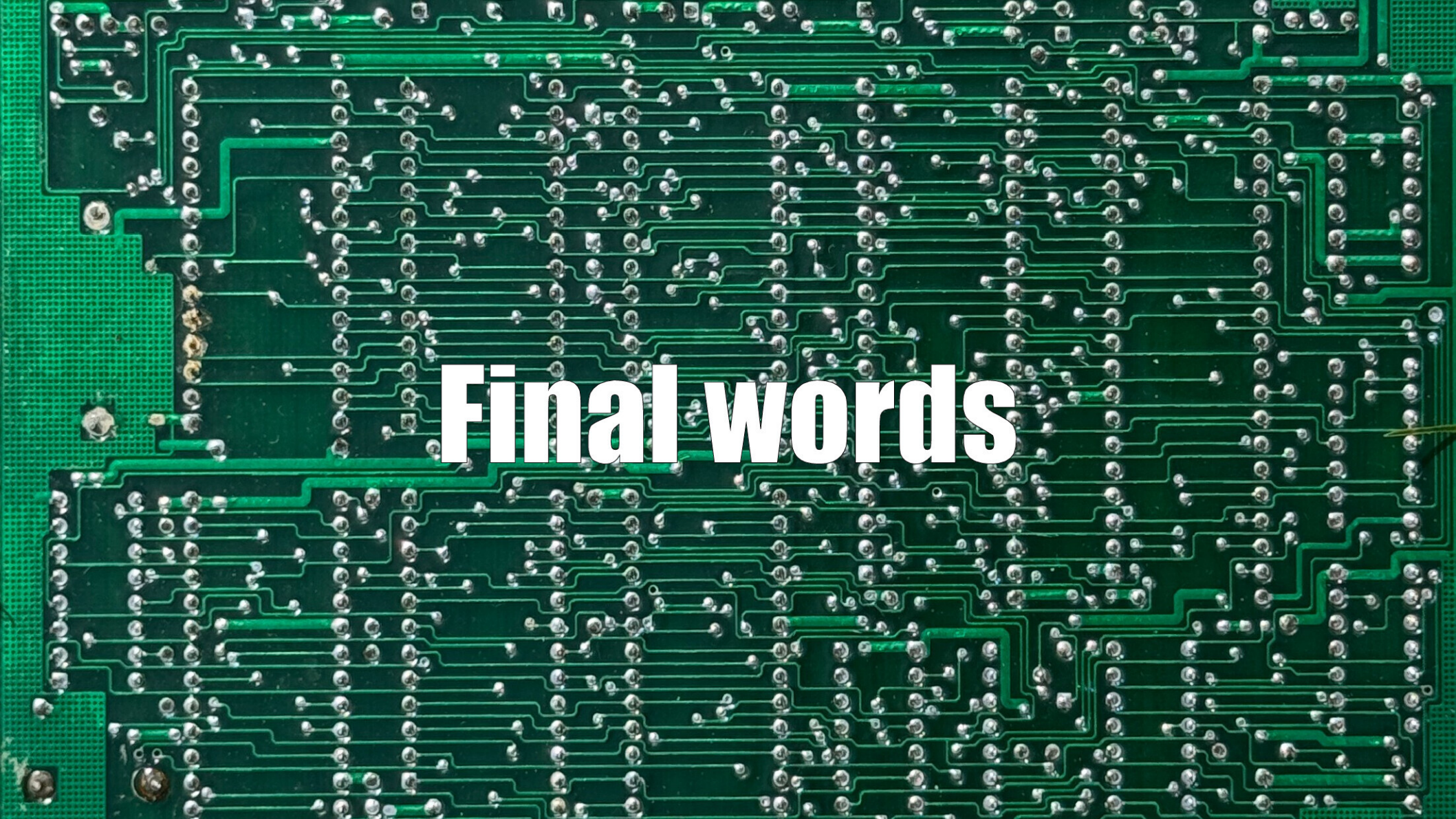



```
1
2 void PWM_TX_hardware_2(void)
3
4 {
5     undefined4 unaff_r4;
6     undefined4 unaff_lr;
7
8     DAT_20001060 = 0;
9     DAT_20000fd3 = 0;
10    DAT_20001054 = 0;
11    DAT_20001055 = 0x40;
12    memcpy(&PWM_output_buffer,&DAT_20000f30,0x80);
13    FUN_00011394();
14    thunk_FUN_00010cd4(&PWM1_BA,4,520000,0x32,unaff_r4,unaff_lr);
15    thunk_FUN_00010cd4(&PWM1_BA,5,520000,0x32);
16    set_PWM_output(&PWM1_BA,0x10);
17    set_PWM_output(&PWM1_BA,0x20);
18    PWM_INTEN01 = PWM_INTEN01 & 0xffffffff;
19    PWM_POLCTL1 = PWM_POLCTL1 | 0x30;
20    DAT_40059004 = DAT_40059004 | 0x4000000;
21    DAT_40058020 = DAT_40058020 & 0xfffffffffe;
22    DAT_40058250 = DAT_40058250 & 0xfffffffffe;
23    CPU_timer_init(0x2968,0);
24    PWM_CNTEN1 = PWM_CNTEN1 | 0x30;
25    DAT_40004854 = 1;
26    DAT_2000105c = 0xd5c9;
27    set_PWM_output(&PWM1_BA,0x20);
28    return;
29 }
30
```



WHERE'S MY CRYPTO?

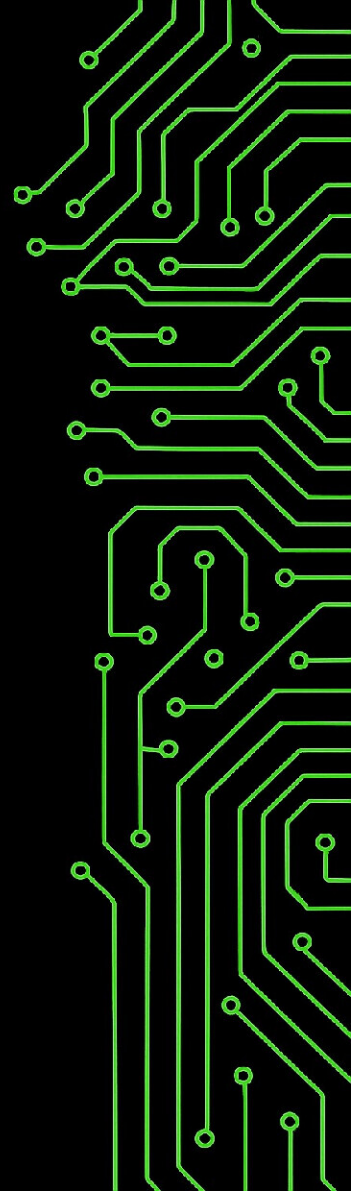




Final words

ONGOING WORK

- Understanding more of the protocol
 - Non-A/V data decoding
- Traffic injections



Q&A TIME

Research brought to you by
Kirils Solovjovs of Possible Security

@k@chaos.social

<https://possiblesecurity.com/>

<https://kirils.org/>

With thanks to Renze Nicolai, PE1BMC, Mitsync, Ivo Mezits, and nemanjan00
icons by freepik