Restoring the Vacuum

by pancake & brainstorm

Agenda

0. Why?
1. Today's target
2. U-boot on Allwinner A20
3. Complex target && ENOTIME?: use r2ai & co.
4. UART-SSH persistence via SSH-Stamp.
5. Future

Why?

- Past r2con (2016): Parrot ARDrone's motor (AVR).
- Last r2con (2024): A no brand treadmill (STM8).
- This r2con (2025): Samsung's vacuum cleaner (ARM).
- Next r2con (2026): 🤷

Unassorted r2 wins

https://github.com/radareorg/radare2-pm/commit/10cbd71221d9f6378454b897d44a2376c883c6ca

https://github.com/radareorg/r2ai/commit/3537d006debc1ecc3c7436d9028dc99be3d77f16

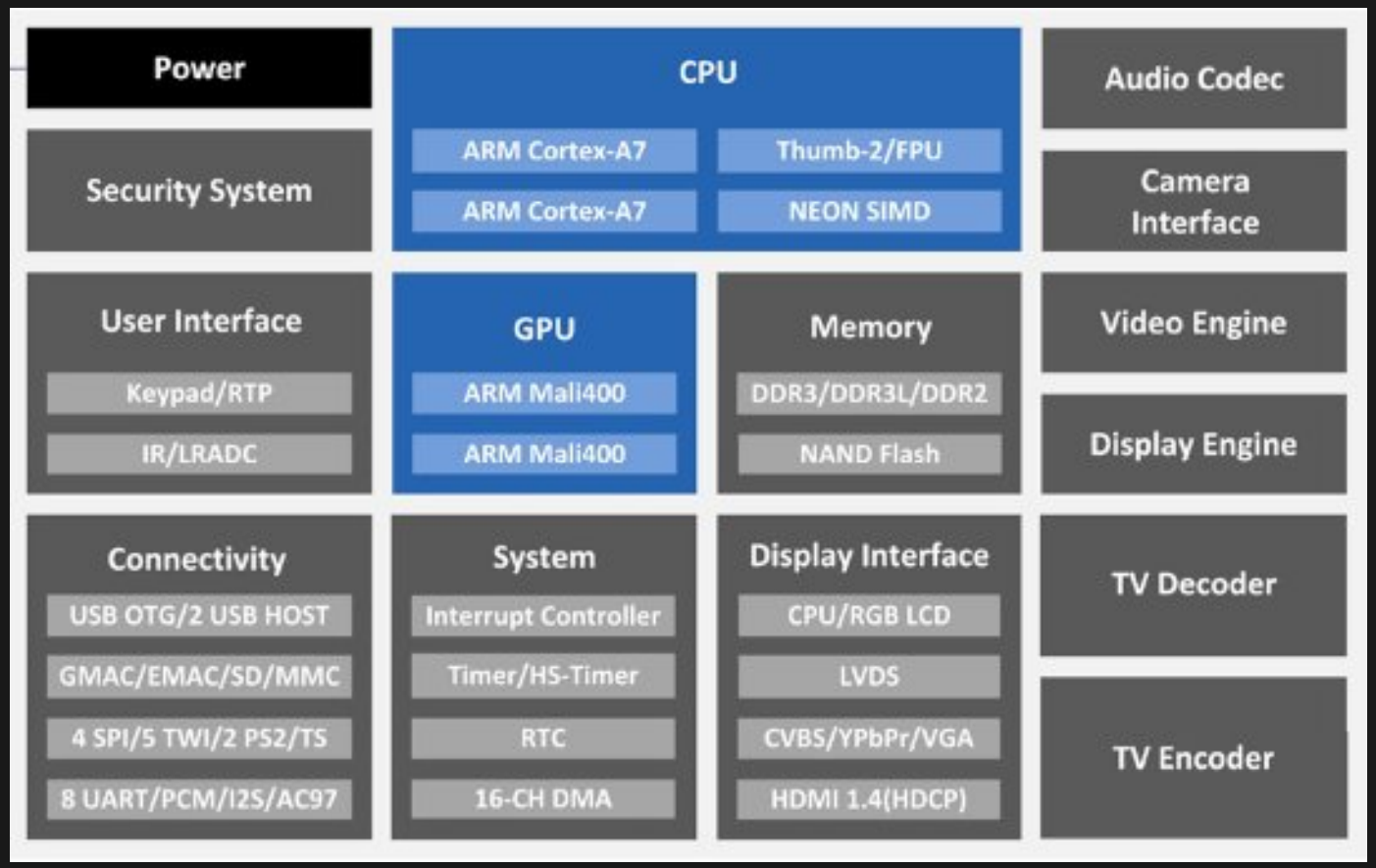
https://github.com/radareorg/radare2/commit/045df9fe9b2ef77f9a5a1d24bb97a866a13eddb9

https://github.com/radareorg/radare2/commit/9d30039c0e86f40025380230309645d0c4dea83c

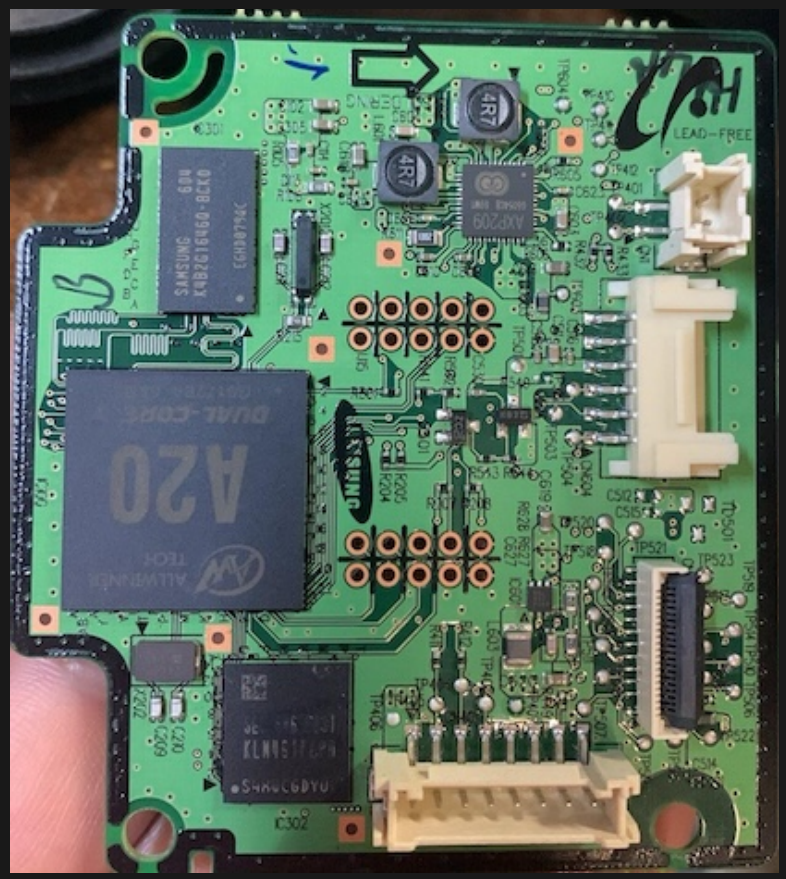
https://github.com/radareorg/radare2/commit/9964c184532a54bfcac8b1ac5f4b8c92ac241d70

Today's e-waste menu

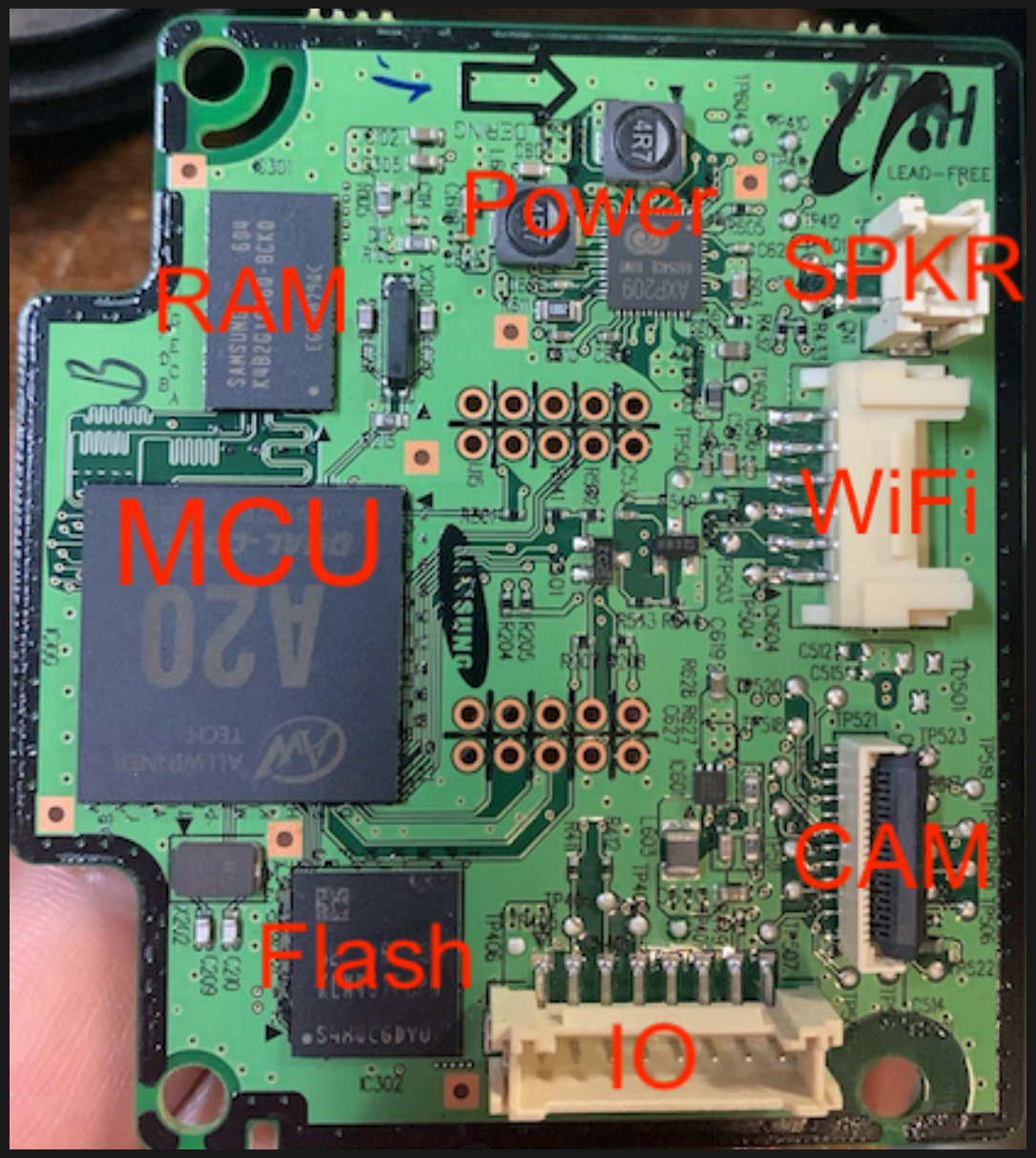
Allwinner A20 specs



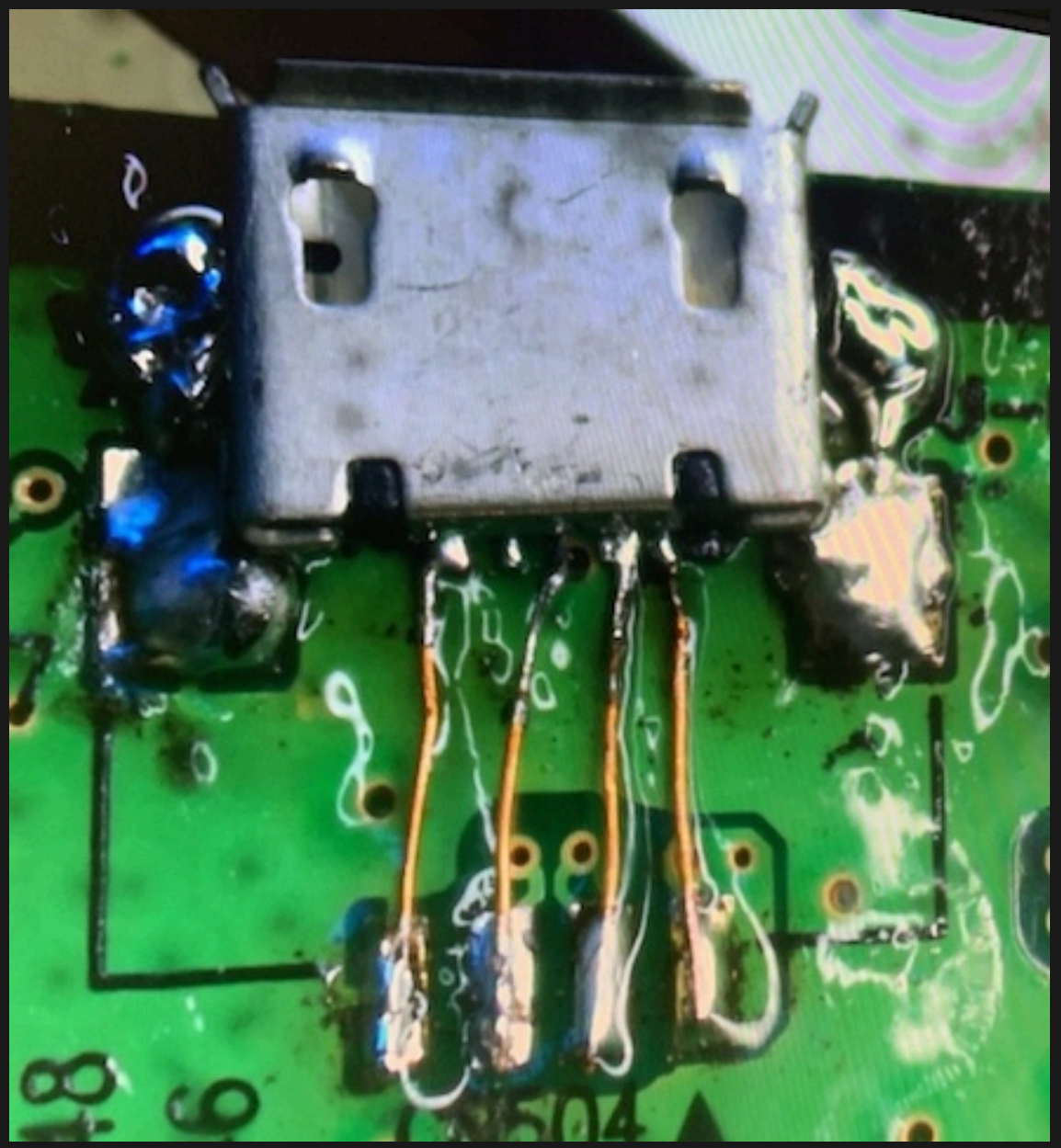
The A20-brains

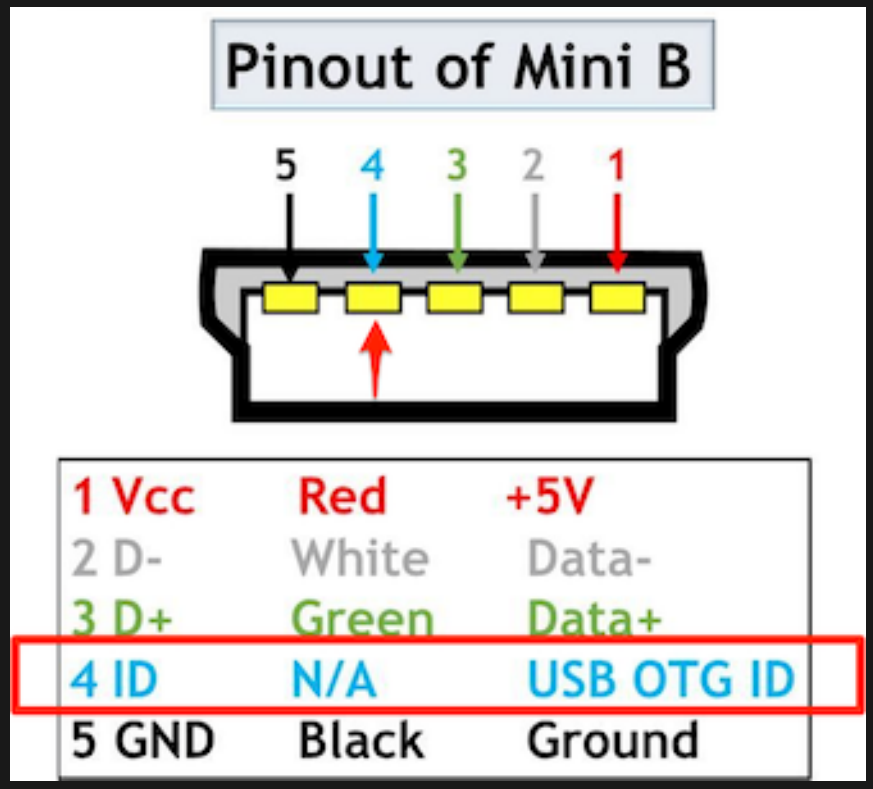


The A20-brains, labeled



When all else FELs...





```
$ ./sunxi-fel version
AWUSBFEX soc=00001651(A20) (...)

$ ./sunxi-fel -l
USB device 003:043   Allwinner A20 (...)
sunxi SoC OTG connector in FEL/flashing mode <--- 🎉
```

TL;DR: OTG to GND and press 2 for FEL via UART

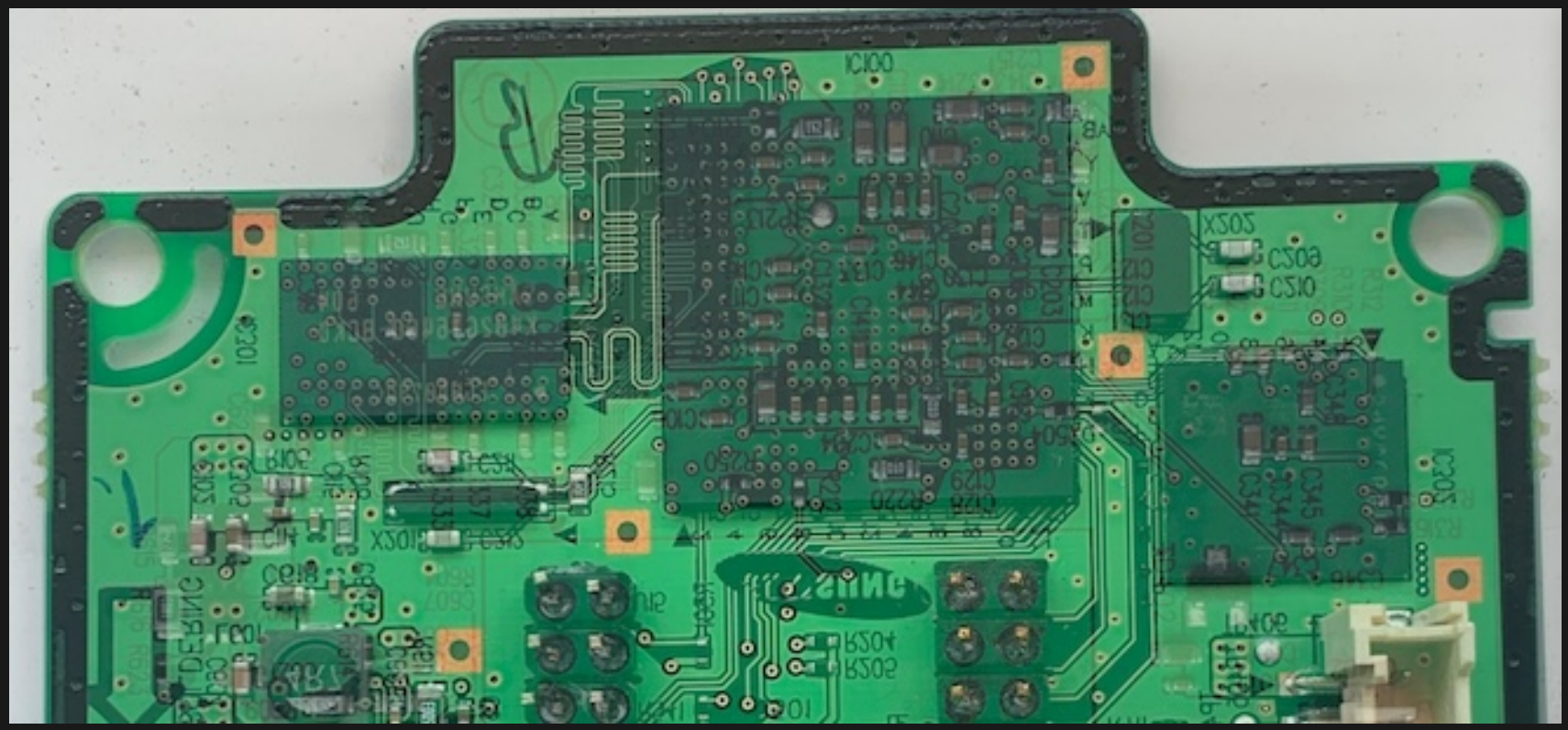
UART?

No clearly labelled pin headers?

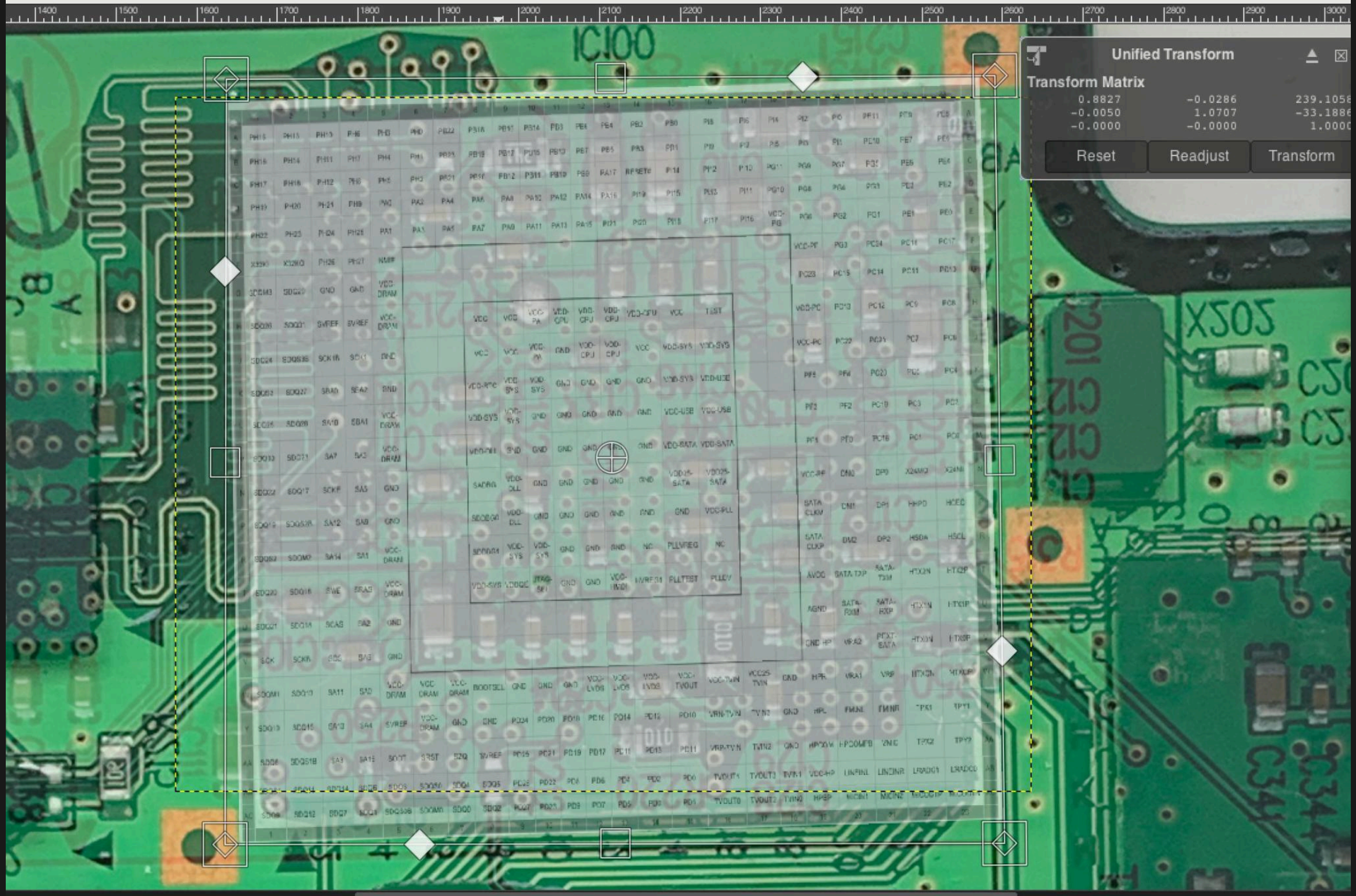
Keep calm and use GIMP

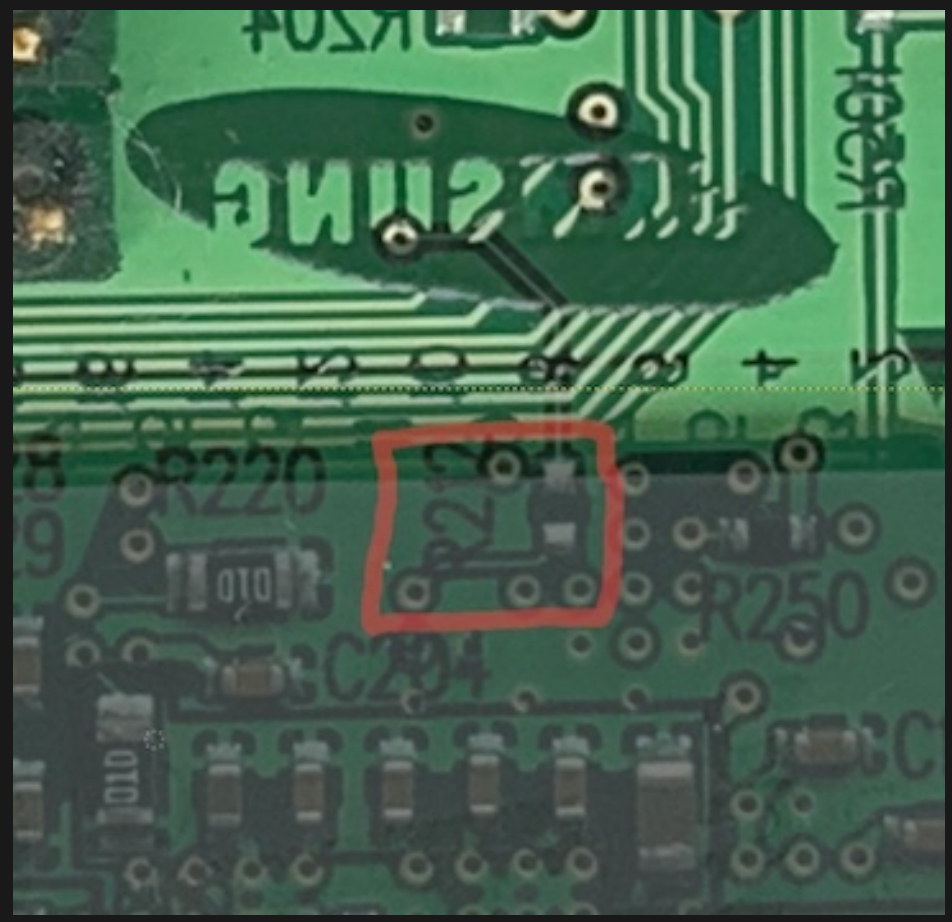
Port	Default Function	IO Type	Default IO State	Default Pull-up/ down	Multi2	Multi3	Multi4	Multi5	Multi6	Multi7
PB22	GPIO	I/O	I	Z	UART0-TX	IR1-TX	-	-	-	-
PB23	GPIO	I/O	I	Z	UART0-RX	IR1-RX	-	-	-	-
PC0	GPIO	I/O	I	Z	NWE#	SPI0-MOSI	-	-	-	-
PC1	GPIO	I/O	I	Z	NALE	SPI0-MISO	-	-	-	-
PC2	GPIO	I/O	I	Z	NCLE	SPI0-CLK	-	-	-	-
PC3	GPIO	I/O	I	Pull-Up	NCE1	-	-	-	-	-
PC4	GPIO	I/O	I	Pull-Up	NCE0	-	-	-	-	-
PC5	GPIO	I/O	I	Z	NRE#	-	-	-	-	-
PC6	GPIO	I/O	I	Pull-Up	NRB0	SDC2-CMD	-	-	-	-
PC7	GPIO	I/O	I	Pull-Up	NRB1	SDC2-CLK	-	-	-	-
PC8	GPIO	I/O	I	Z	NDQ0	SDC2-D0	-	-	-	-

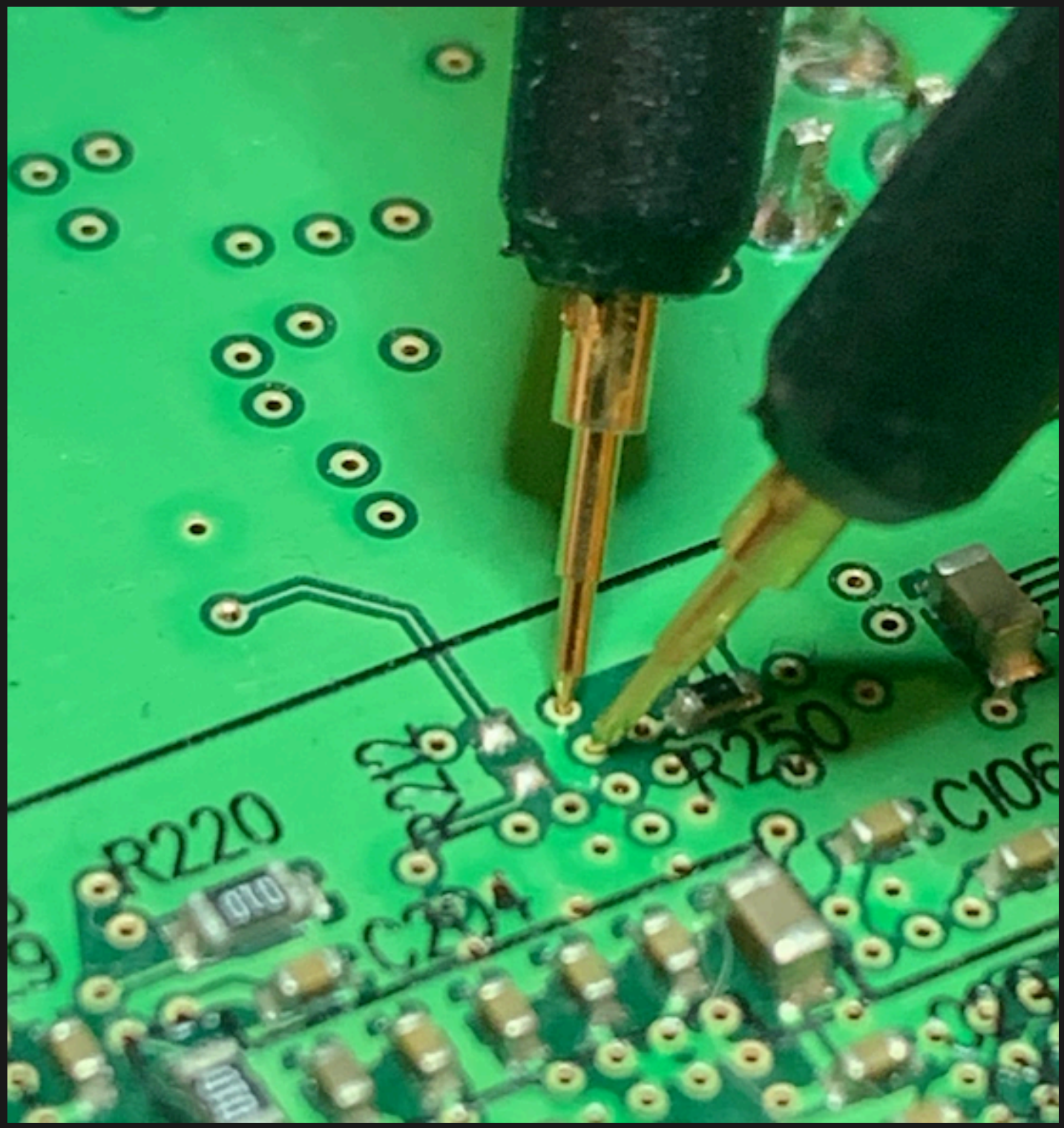
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
A	PH15	PH13	PH10	PH6	PH3	PH	PB22	PB	PB16	PB14	PB8	PB6	PB4	PB2	PB0	PI8	PI6	PI4	PI2	PI0	PE11	PE9	PE8	A
B	PH16	PH14	PH11	PH7	PH4	PH	PB23	PB	PB17	PB15	PB13	PB7	PB5	PB3	PB1	PI9	PI7	PI5	PI3	PI1	PE10	PE7	PE6	B
C	PH17	PH18	PH12	PH8	PH5	PH			PB12	PB11	PB10	PB9	PA17	RESET#	PI14	PI12	PI10	PG11	PG9	PG7	PG5	PE5	PE4	C
D	PH19	PH20	PH21	PH9	PA0	PA2	PA4	PA6	PA8	PA10	PA12	PA14	PA16	PI19	PI15	PI13	PI11	PG10	PG8	PG4	PG3	PE3	PE2	D
E	PH22	PH23	PH24	PH25	PA1	PA3	PA5	PA7	PA9	PA11	PA13	PA15	PI21	PI20	PI18	PI17	PI16	VCC-PG	PG6	PG2	PG1	PE1	PE0	E
F	X32KI	X32KO	PH26	PH27	NMI#														VCC-PE	PG0	PC24	PC18	PC17	F
G	SDQM3	SDQ29	GND	GND	VCC-DRAM														PC23	PC15	PC14	PC11	PC10	G
H	SDQ26	SDQ31	SVREF	SVREF	VCC-DRAM			VCC	VCC	VCC-PA	VDD-CPU	VDD-CPU	VDD-CPU	VDD-CPU	VCC	TEST			VCC-PC	PC13	PC12	PC9	PC8	H
J	SDQ24	SDQS3B	SCK1B	SCK1	GND			VCC	VCC	VCC-PA	GND	VDD-CPU	VDD-CPU	VCC	VDD-SYS	VDD-SYS			VCC-PC	PC22	PC21	PC7	PC6	J
K	SDQS3	SDQ27	SBA0	SBA2	GND			VDD-RTC	VDD-SYS	VDD-SYS	GND	GND	GND	GND	VDD-SYS	VDD-USB			PF5	PF4	PC20	PC5	PC4	K
L	SDQ25	SDQ28	SA10	SBA1	VCC-DRAM			VDD-SYS	VDD-SYS	GND	GND	GND	GND	GND	VCC-USB	VCC-USB			PF3	PF2	PC19	PC3	PC2	L
M	SDQ30	SDQ23	SA7	SA3	VCC-DRAM			VDD-DLL	GND	GND	GND	GND	GND	GND	VDD-SATA	VDD-SATA			PF1	PF0	PC16	PC1	PC0	M
N	SDQ22	SDQ17	SCKE	SA5	GND			SADBG	VDD-DLL	GND	GND	GND	GND	GND	VDD25-SATA	VDD25-SATA			VCC-PF	DM0	DP0	X24MO	X24MI	N
P	SDQ19	SDQS2B	SA12	SA9	GND			SDDBG0	VDD-DLL	GND	GND	GND	GND	GND	GND	VCC-PLL			SATA-CLKM	DM1	DP1	HHPD	HCEC	P
R	SDQS2	SDQM2	SA14	SA1	VCC-DRAM			SDDBG1	VDD-SYS	VDD-SYS	GND	GND	GND	NC	PLLVREG	NC			SATA-CLKP	DM2	DP2	HSDA	HSCL	R
T	SDQ20	SDQ16	SWE	SRAS	VCC-DRAM			VDD-SYS	VDDQE	JTAG-SEL	GND	GND	VCC-HMDI	HVREG1	PLLTEST	PLLDV			AVCC	SATA-TXP	SATA-TXM	HTX2N	HTX2P	T
U	SDQ21	SDQ18	SCAS	SA2	GND														AGND	SATA-RXM	SATA-RXP	HTX1N	HTX1P	U
V	SCK	SCKB	SCS	SA6	GND														GND-HP	VRA2	REXT-SATA	HTX0N	HTX0P	V
W	SDQM1	SDQ13	SA11	SA0	VCC-DRAM	VCC-DRAM	VCC-DRAM	BOOTSEL	GND	GND	GND	VCC-LVDS	VCC-LVDS	VCC-LVDS	VCC-TVOUT	VCC-TVIN	VCC25-TVIN	GND	HPR	VRA1	VRP	HTXCN	HTXCP	W
Y	SDQ10	SDQ15	SA13	SA4	SVREF	VCC-DRAM	GND	GND	PD24	PD20	PD18	PD16	PD14	PD12	PD10	VRN-TVIN	TVIN3	GND	HPL	FMINL	FMINR	TPX1	TPY1	Y
AA	SDQ8	SDQS1B	SA8	SA15	SODT	SRST	SZQ	SVREF	PD25	PD21	PD19	PD17	PD15	PD13	PD11	VRP-TVIN	TVIN2	GND	HPCOM	HPCOMFB	VMIC	TPX2	TPY2	AA
AB	SDQS1	SDQ11	SDQ14	SDQ6	SDQ3	SDQS0	SDQ4	SDQ5	PD26	PD22	PD8	PD6	PD4	PD2	PD0	TVOUT1	TVOUT3	TVIN1	VCC-HP	LINEINL	LINEINR	LRADC1	LRADC0	AB
AC	SDQ9	SDQ12	SDQ7	SDQ1	SDQS0B	SDQM0	SDQ0	SDQ2	PD27	PD23	PD9	PD7	PD5	PD3	PD1	TVOUT0	TVOUT2	TVIN0	HPBP	MICIN1	MICIN2	MICOUTP	MICOUTN	AC
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	


IC100
C509
C510
SHARP

*A20_board_stackup.xcf-7.0 (RGB color 8-bit gamma integer, GIMP built-in sRGB, 3 layers) 4032x3024 – GIMP




R220
010
R250
C204
010


R220
C294
R250
C106

Abridged bootlog

```
[    0.000000] Linux version 3.4.39 (...)
[    0.000000] CPU: ARMv7 Processor [410fc074] (...)
[    0.000000] Machine: sun7i
(...)
Welcome to Tizen Light (2.3.0)!
(...)
smack.mount: Directory /smack to mount over is not empty,
mounting anyway.
(...)
**************************************
     SAMSUNG LINUX PLATFORM
**************************************

localhost login:
```

A good u-boot book

OpenWrt's U-Boot for Olimex

```
$ ./sunxi-tools/sunxi-fel uboot \
olimex_a20-olinuxino-lime2-emmc-u-boot-with-spl.bin

U-Boot SPL 2024.01-OpenWrt-r27076-d604b2699b (...)
CPU: 912000000Hz, AXI/AHB/APB: 3/2/2
Trying to boot from FEL
U-Boot 2024.01-OpenWrt-r27076-d604b2699b (...)
CPU:   Allwinner A20 (SUN7I)
Model: Olimex A20-OLinuXino-LIME2-eMMC
DRAM:  256 MiB
Core:  65 devices, 25 uclasses, devicetree: separate

No ethernet found.
Could not get PHY for ethernet@1c50000: addr 1
=>
```

No ethernet found?

1. Olimex board definition does not match Samsung's CLEANER.
2. Lucky that I didn't fry anything: voltage regulators are typically configured via Device Tree (.dts/.dtc files).
3. A custom format ("FEX") is used for early DRAM, voltage regulator settings and many other early boot peripherals.
4. Found under `/boot/sys_config.bin`.

BIN2FEX, FEX2DTS

ChatGPT

Share

sys_config.fex
File

Given this DTS definition: https://gitlab.com/linux-kernel/linux/-/blob/master/arch/arm/boot/dts/allwinner/sun7i-a20-olinuxino-lime2.dts and the Allwinner FEX file (https://linux-sunxi.org/Fex_Guide) I attached, could you generate an output .dts file specific for my "SAMSUNG_CLEANER.dts" board?

Thought for 54s >

Samsung Cleaner · other

Copy Edit Download

```
/*
 * Generated DTS for board: SAMSUNG_CLEANER
 * Source references:
 *  - sun7i-a20-olinuxino-lime2.dts (used as a template)
 *  - Converted from provided FEX: /mnt/data/sys_config.fex
 *
 * NOTE: This is an automated, best-effort translation of the FEX to
 * Review pinctrl names, regulator names and power domains against o
 * Allwinner A20 DTSI files before using on real hardware.
 */

/dts-v1/;

#include "sun7i-a20.dtsi"
#include "sunxi-common-regulators.dtsi"

/ {
```

Handy U-boot commands

OpenWrt's U-Boot config should ideally have:

```
CMD_CONFIG_LS
CMD_CONFIG_CAT
```

```
=> ls mmc 1:4 /usr/apps/robot-network/data/scs/conf/
<DIR>       4096 .
<DIR>       4096 ..
1396 danet.json <-- DA (robot) NETwork
```

It's nice to poke files directly from u-boot (DFIR)...

root:1zSrf7/kI$taalXPoxdVGKvtnjD7KeB.

Dumping the eMMC

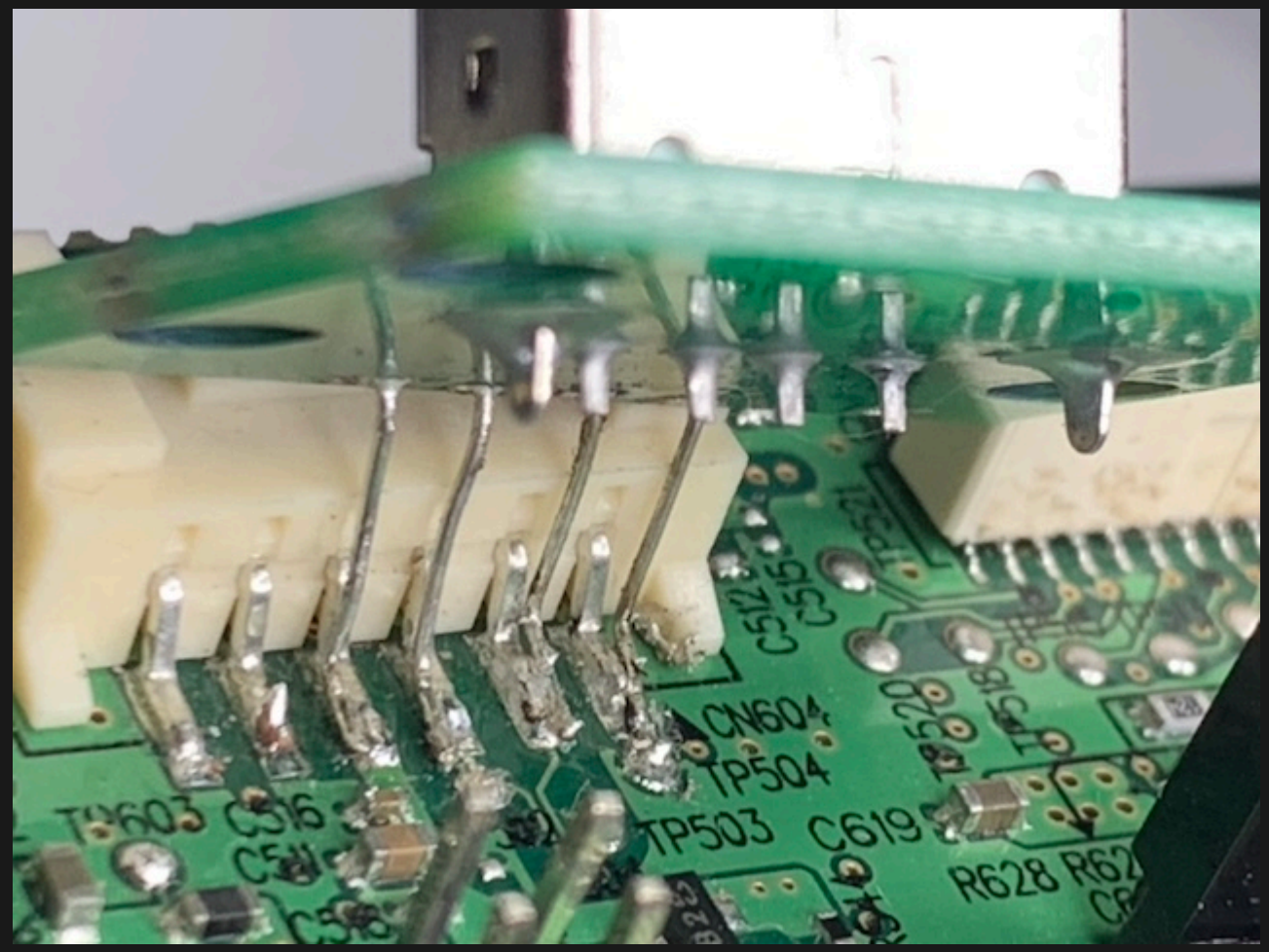
```
use serialport::SerialPort;

let chunk_size = 0x10000;      // 64KB chunks
let total_sectors = 0x72A000; // Approx 3.6 GiB flash
let addr = 0x42000000;         // RAM copy buffer

 while sector < total_sectors {
        let read_cmd = format!(
"mmc read 0x{:X} 0x{:X} 0x{:X}\r\n",
          addr,  sector, chunk_size);

        let write_cmd = format!(
"usb write 0x{:X} 0x{:X} 0x{:X}\r\n",
           addr,  sector, chunk_size);
```

Board grows another USB port (USB Storage)



There was a WiFi dongle there, but we need to borrow that USB 2.0 port for eMMC to USB copy.

USB errors

```
(...)
Sending: mmc read 0x42000000 0x18F00 0x100
Sending: usb write 0x42000000 0x18F00 0x100
Sending: mmc read 0x42000000 0x19000 0x100
Sending: usb write 0x42000000 0x19000 0x100
(stops)
^C
(...)
usb write: device 0 block # 102400, count 256 ...
EHCI timed out on TD - token=0xd0008c80
```

Quick online search: many boards suffer this, even with newer u-boot 2025 releases. No "robust" fix.

u-boot dev speaks

I spent too much time and money already on trying to track down all these weird issues the USB sticks have.

If you have some magic solution, patches are welcome ;-/

Just swap the pendrives

eMMC flash dump successful!

```
(...)
Sending: mmc read 0x42000000 0x710000 0x10000
Sending: usb write 0x42000000 0x710000 0x10000
R/W iteration count is: 114
Sending: mmc read 0x42000000 0x720000 0x10000
Sending: usb write 0x42000000 0x720000 0x10000
R/W iteration count is: 115
Done.
```

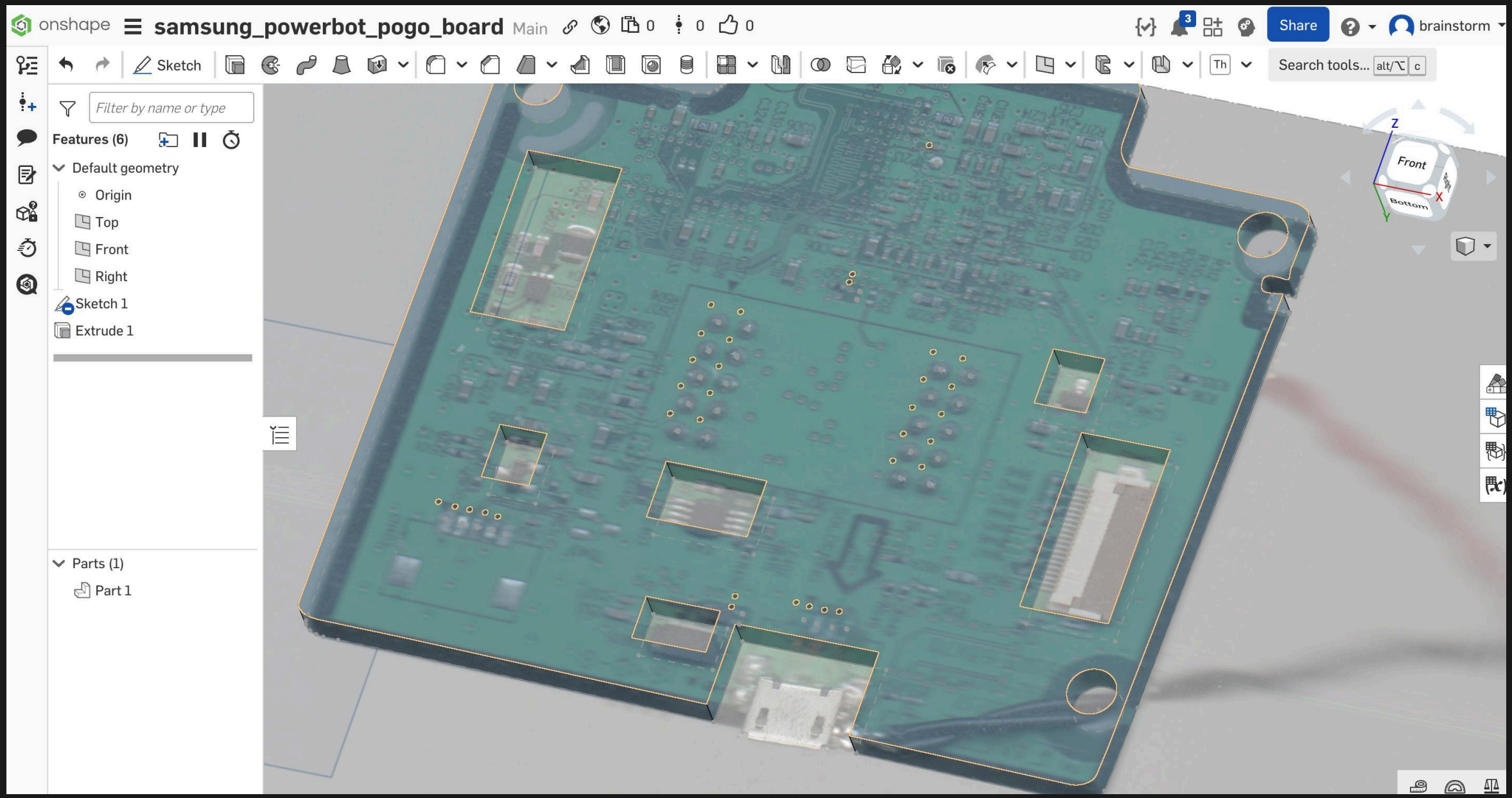
🎉

ENOTIME, use LLMs?

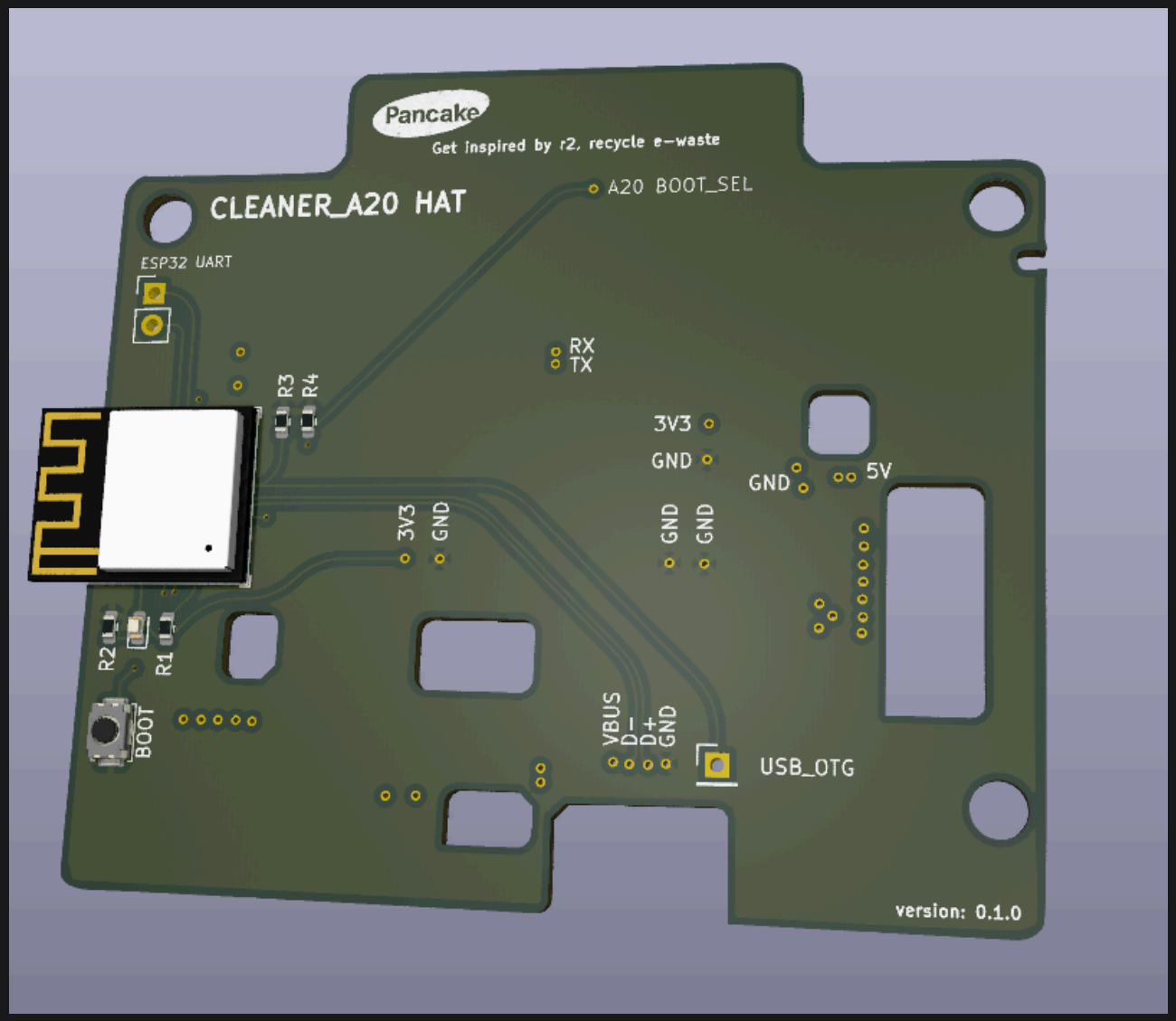
1. We have almost 4GB of flash to comb through.
2. Goal: find out how motors and sensors work overall.

<Interactive time>

Attaining UART persistence



KiCad design time



Reality check time

Not selling you PCBs!

SSH Stamp

1. Funded by NLNet (thank you!).
2. `no_std`, thanks to `esp_hal` and `sunset`.
3. Careful `unsafe` use (if at all needed).
4. WIP, all help is welcome.

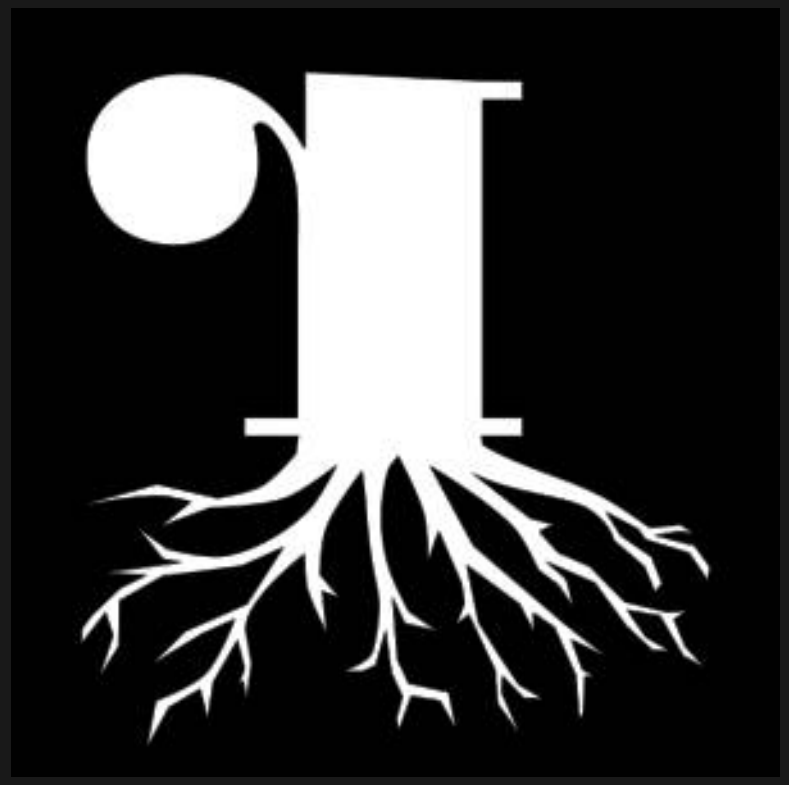
TL;DR: Aiming at a robust way to access IoT (UART) backdoors via a Rust embedded SSH server.

https://github.com/brainstorm/ssh-stamp

Future

1. Adapt the HAT accordingly for the peripherals we can reach.
2. Port to Valetudo?
3. Fix the device tree and upstream a new OpenWrt board called "CLEANER_A20"?
4. For radare2: Keep improving C++/LLM reversing (RTTI, class resolution, *this, vtables, etc...) for better automated insights.

FIN



USE EMAIL;

brainstorm@nopcode.org