

電子辞書は組み込み Linux の夢を見るか？

末田 卓巳 @puhitaku

Ver. 2.0.0

分解のススメ 第6回 2021/1/30

自己紹介

末田 卓巳 Takumi Sueda @puhitaku



- ・全レイヤーやるタイプ
中でも低レイヤーが好き
- ・現在はフリーランス
NICT (情報通信研究機構) など
- ・人生で初めてバラしたのは PS2



過去の主な仕事

- ・ GROOVE X 社 "LOVOT"

主な趣味

- ・ 回路解析を含むハック
(電子辞書・ルーター etc.)
- ・ 3D プリント
- ・ Python, Go で便利ツール作り



**去る2010年… 津山高専の教科書販売にて
16歳の puhitaku 少年はある電子辞書を手**





SHARP Brain PW-GC610



puhitaku少年



本体裏をよく見ると...

Windows® Embedded CE 6.0
Core

00039-417-367-097

X13-12056

リセット

Windows® Embedded CE 6.0
Core

00039-417-367-097

X13-12056

電子辞書で

Windows CE が動いている！？

SHARP Brain シリーズ

特徴

- Windows CE を搭載
- CE 向けアプリの一部や自分で開発したアプリが動く
- 2010年には既に Brain ハック界隈が 2ch で成立

当時あったアプリ

- アプリランチャー
- オフラインで Wikipedia を読むソフト
- matplotlib
- 移植されたギャルゲー
などなど…



SHARP Brain PW-GC610



puhitaku少年

マジかよ！最高！

少年の心にバッチリ刺さった

しかし

ソフト・ハード両面の貧弱さにより
界隈の活気はみるみる衰退



puhitaku少年

僕にできることは…

ろくに技術もなかった私は
ただ眺めるしかなかった

時は流れ 2019年…

時は流れ 2019年…

引き出しに眠る Brain を見ながら

ふと疑問が浮かんだ



puhitaku

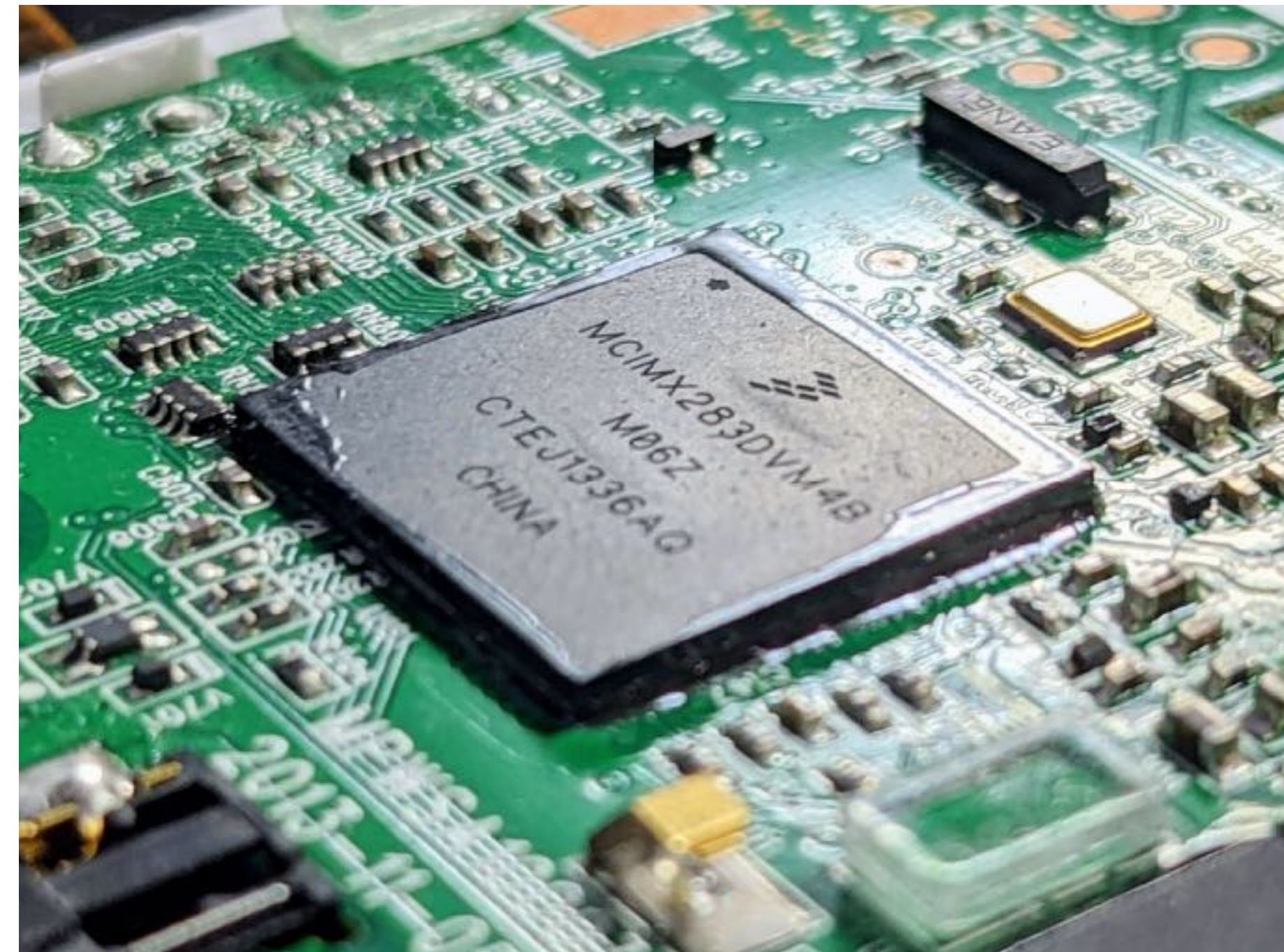
そういえば
Brain の CPU ってなんだろう？

ググってみた…
するとビックリ！

**組み込み Linux 向け SoC の代名詞
i.MX シリーズを搭載していた！**

i.MX シリーズとは？

- Freescale（現: NXP）の SoC
- 組み込み Linux の生き証人
- Linux と U-Boot（ブートローダ）のソースコードが GitHub にある
- データシートとリファレンスマニュアルが無料で手に入る
- Brain と同じ SoC の評価ボードが買える
- 過去に仕事でガッツリ使ってた

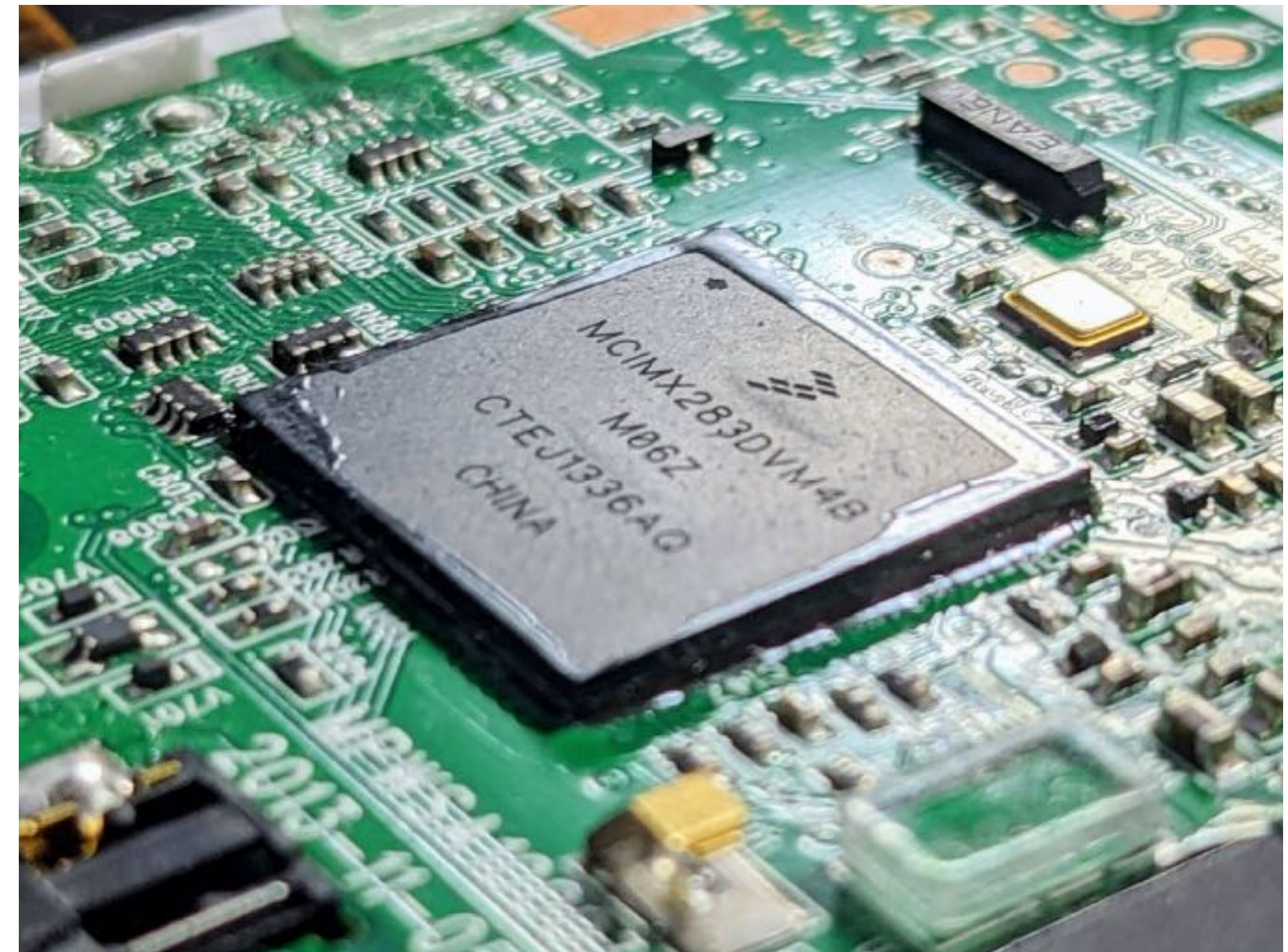


SHARP Brain PW-SH1 に
搭載されている i.MX 283

i.MX シリーズとは？

- Freescale（現: NXP）の SoC
- 組み込み Linux の生き証人
- Linux と U-Boot（ブートローダ）のソースコードが GitHub にある
- データシートとリファレンスマニュアルが無料で手に入る
- Brain と同じ SoC の評価ボードが買える
- 過去に仕事でガッツリ使ってた

**Linux を動かしてくれと
言っているようなもの**



SHARP Brain PW-SH1 に
搭載されている i.MX 283



puhitaku

もしかして
Brain で Linux が動くんじゃないや…？

いてもたってもいられず
メルカリで秒で購入

**SHARP Brain への Linux 移植という
壮大な旅が始まった**

バラして中身を見してみる

外観



SHARP Brain PW-SH1

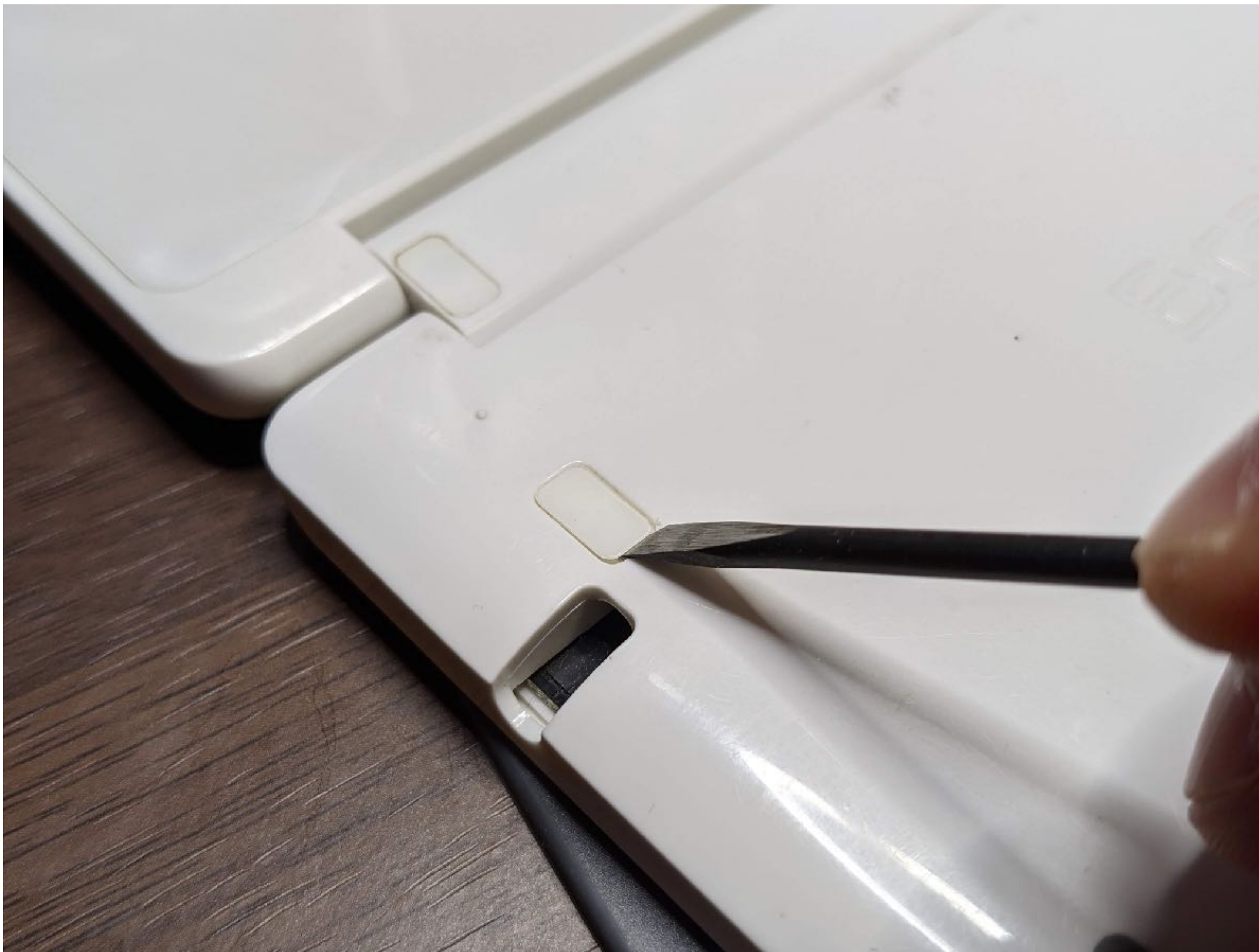
ひっくり返す



ネジを隠しているステッカー



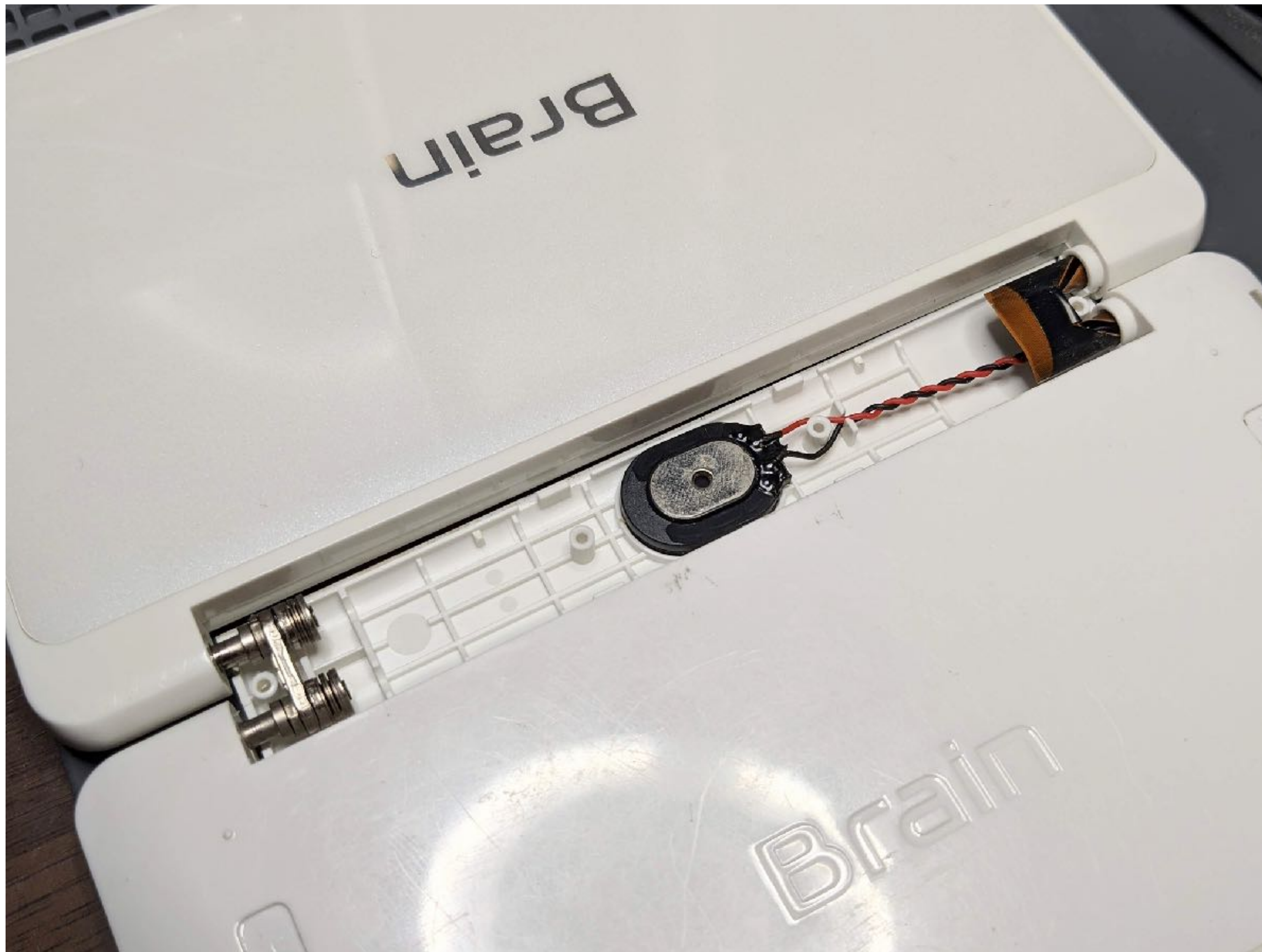
ネジを隠しているステッカー



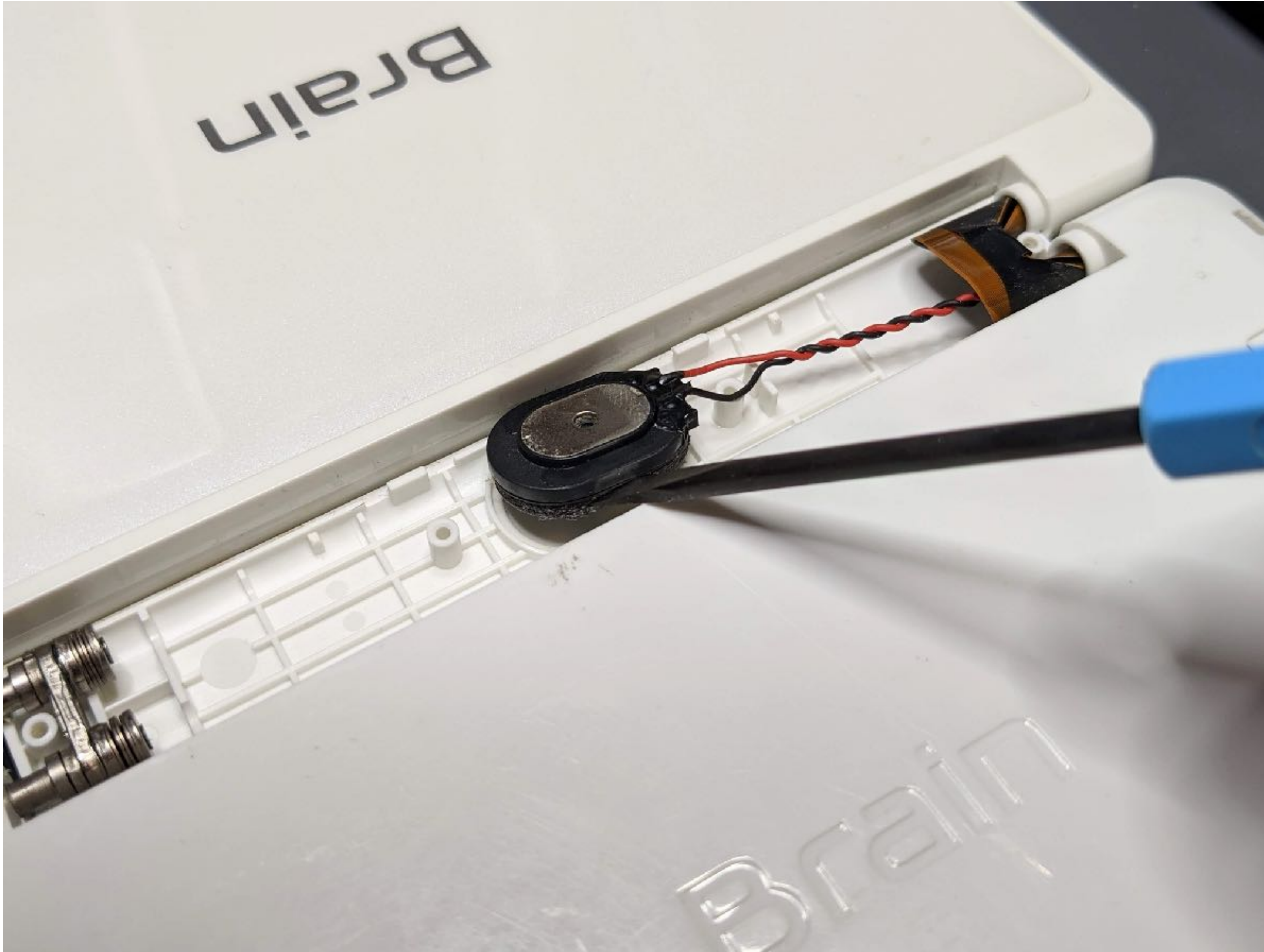
すべてはがしてネジを外す



ヒンジ部の背中側を取り外す



ヒンジ部の背中側を取り外す



途中経過



あとはノリでツメを浮かせて…



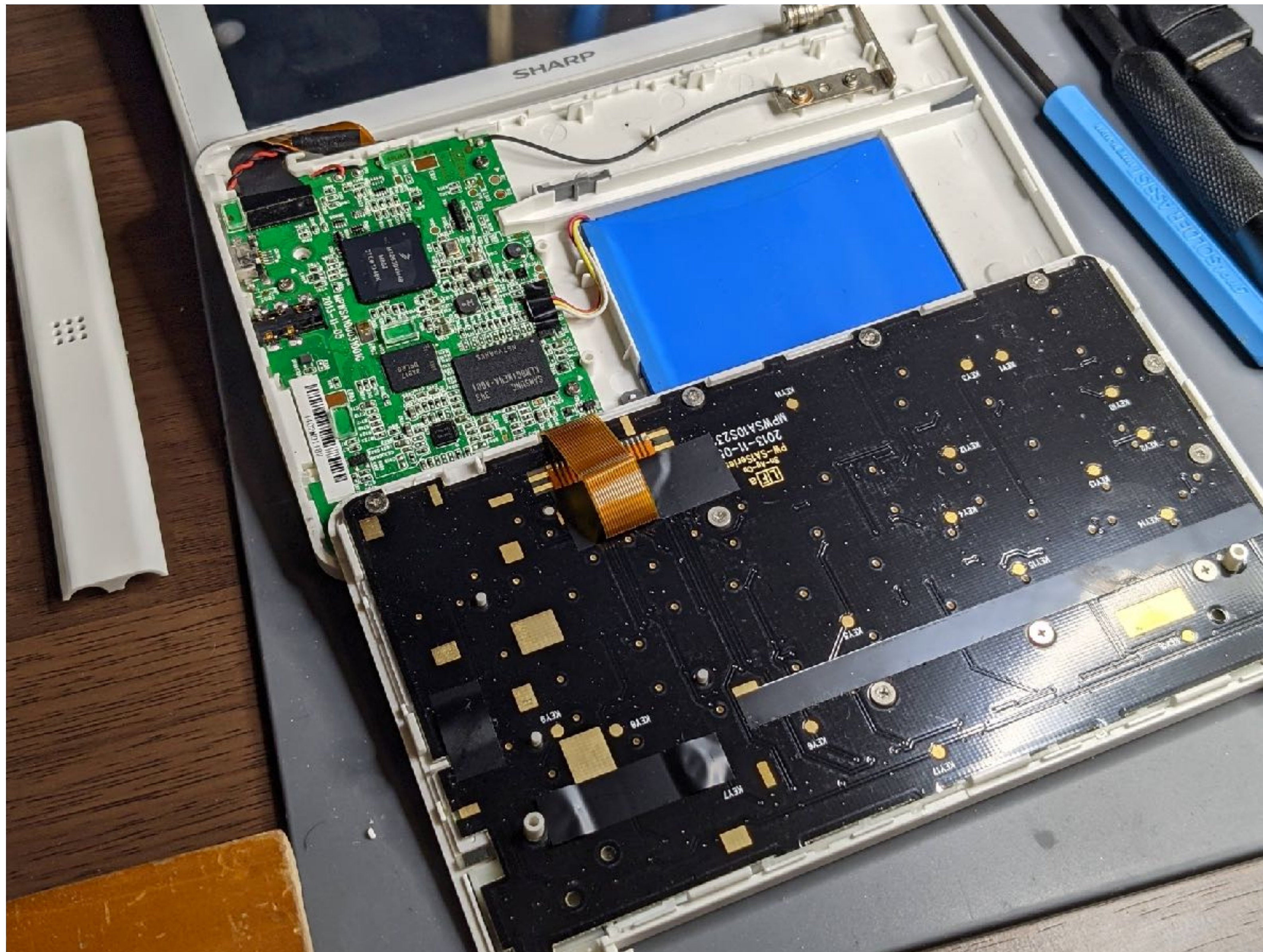
キーボードのフレキに注意しながら浮かせる



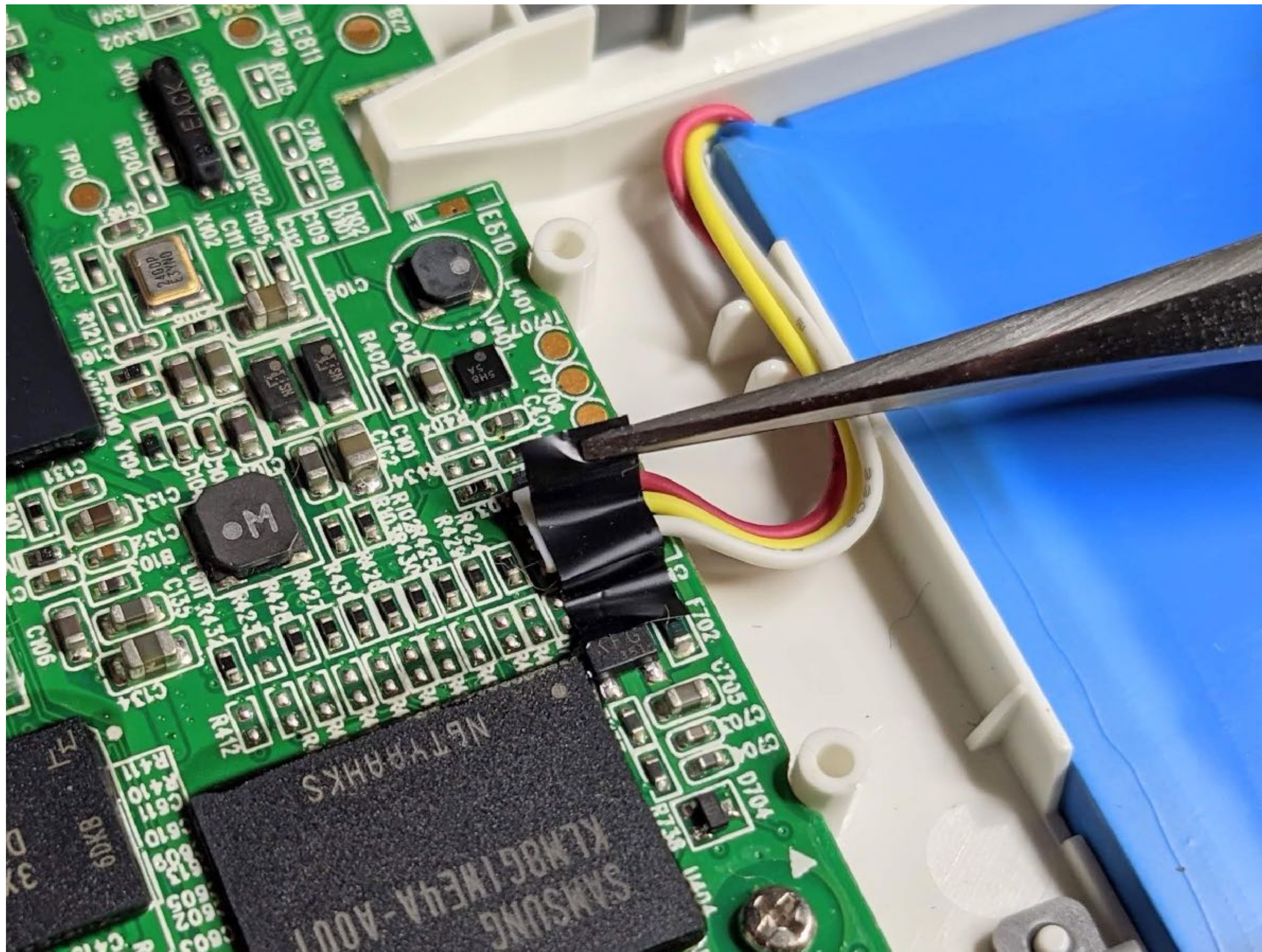
キーボードのフレキに注意しながら浮かせる



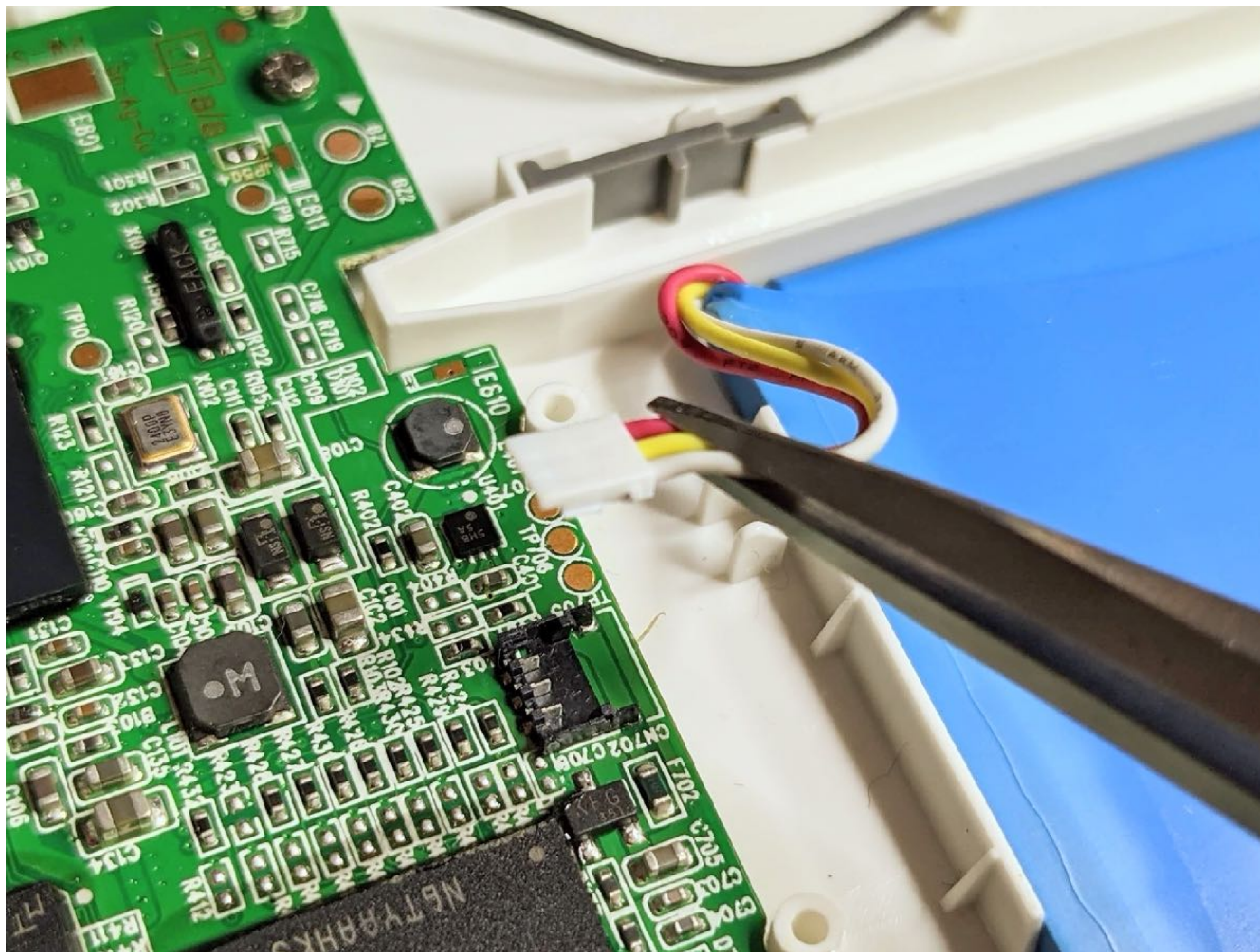
キーボードのフレキに注意しながら浮かせる



バッテリーを取り外す



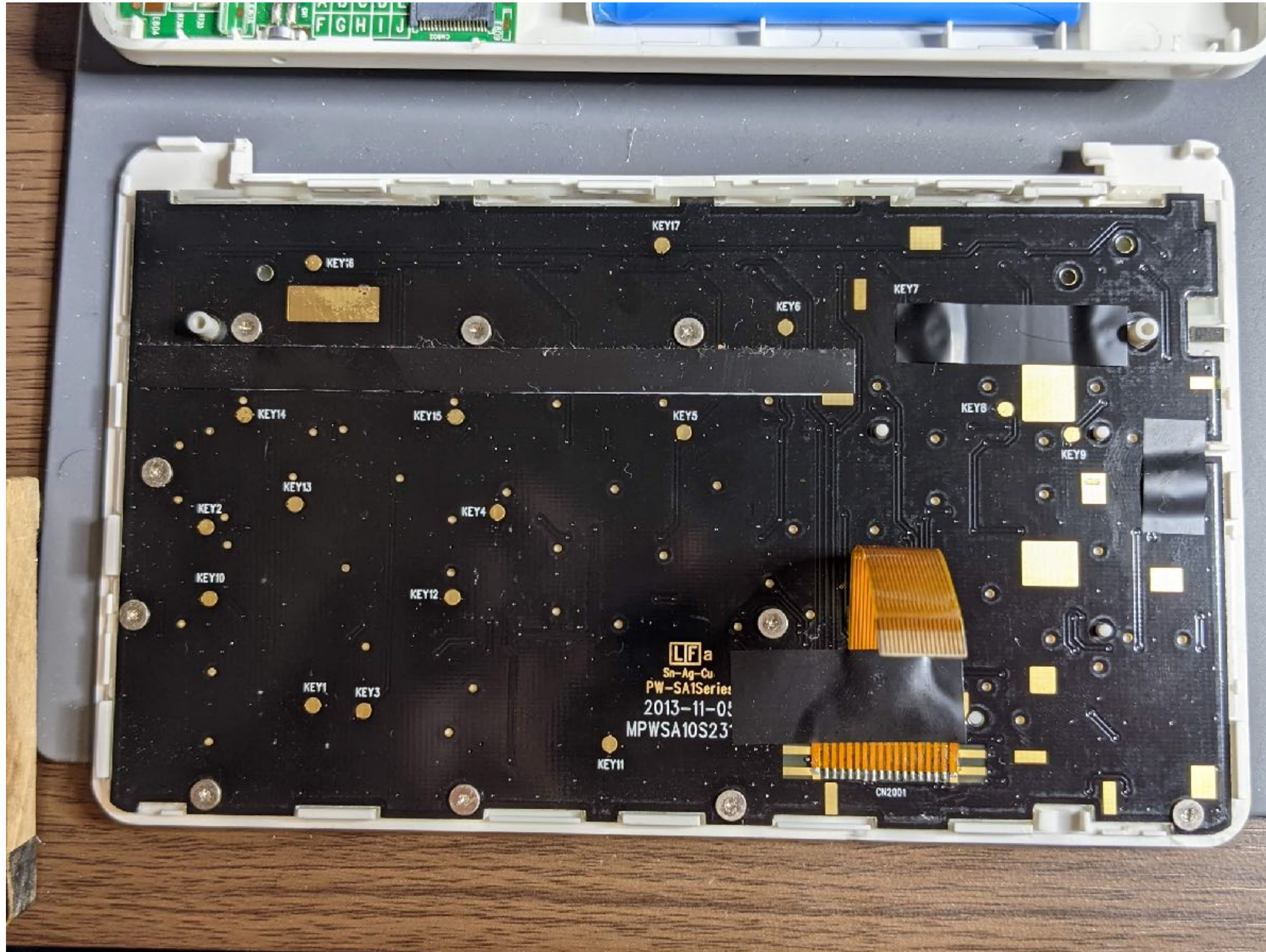
バッテリーを取り外す



キーボード側 分解完了



キーボード 裏側

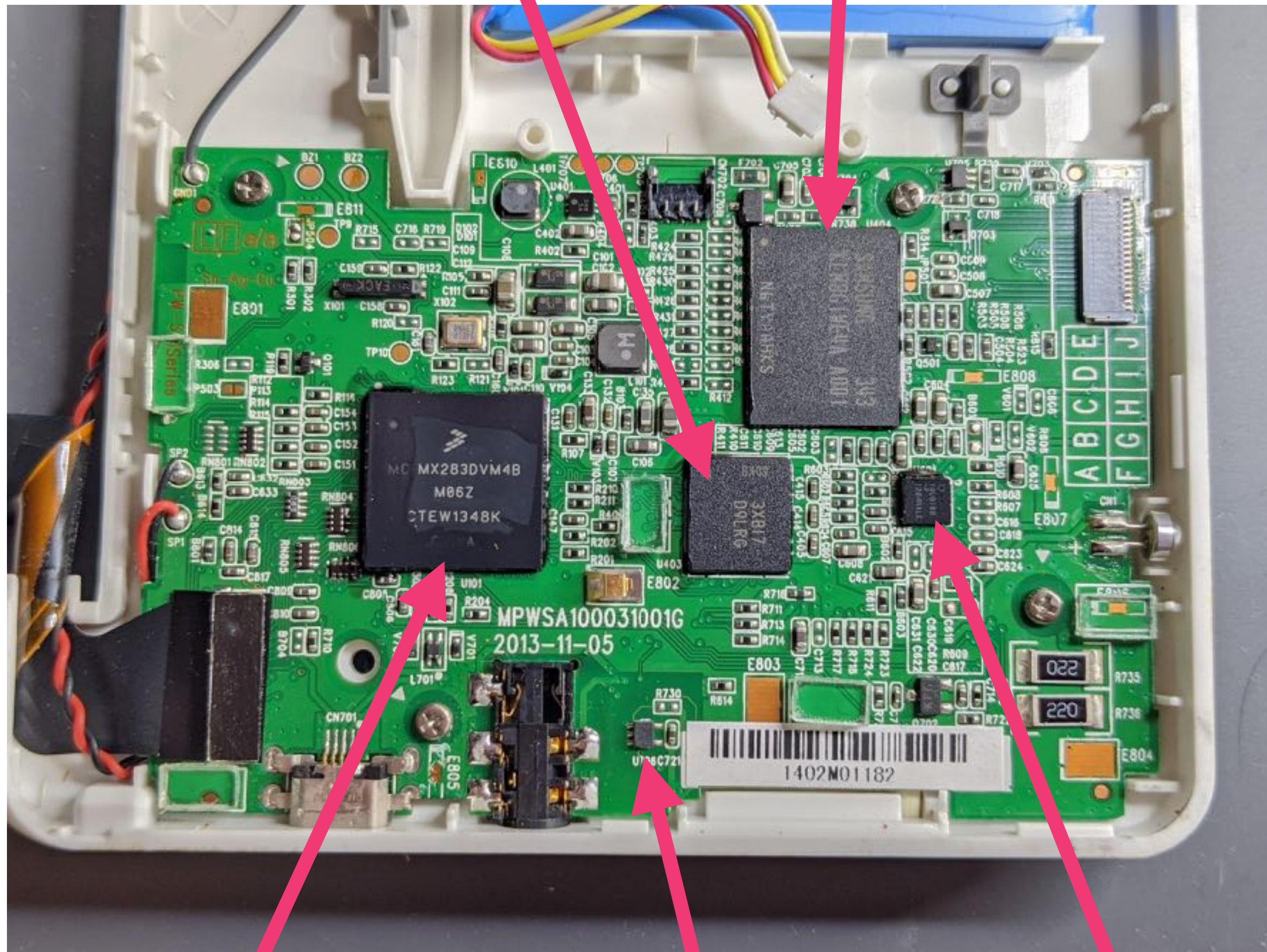


メインボード

Micron MT46H64M16LFBF-5

Samsung KLM8G1WE4A

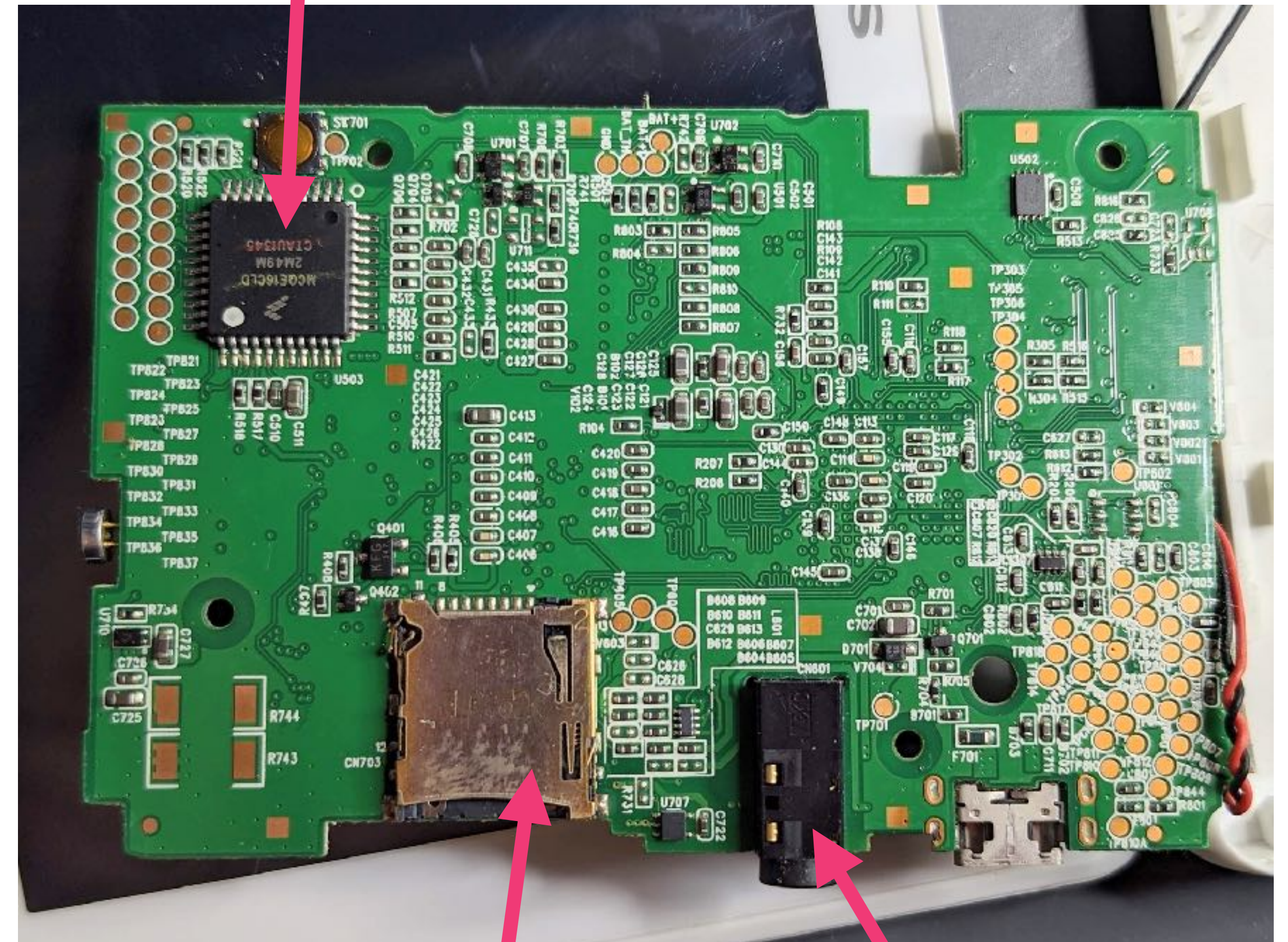
Freescale MCQE16CLD



NXP i.MX283

磁気センサー

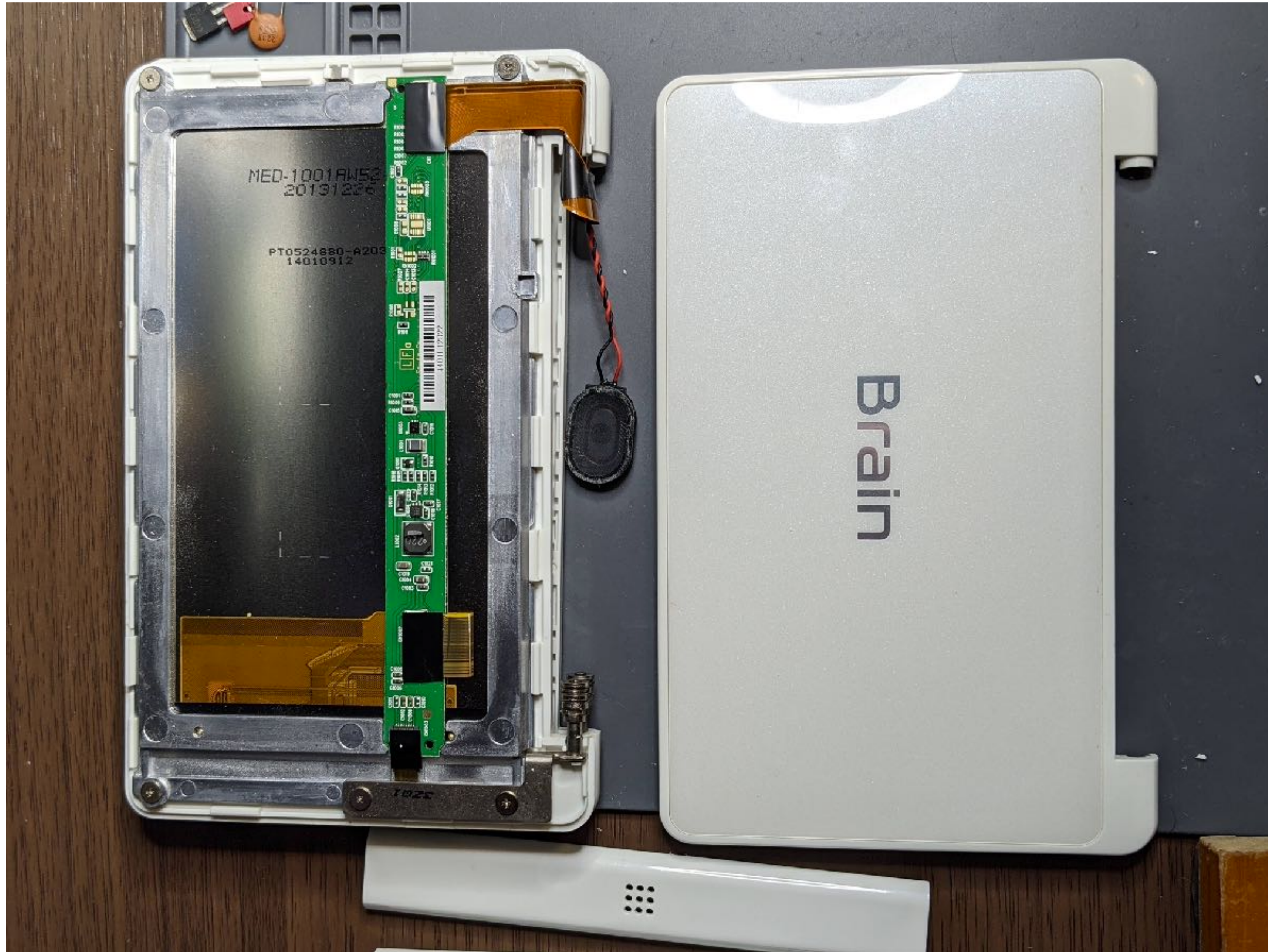
Yamaha YMU818B



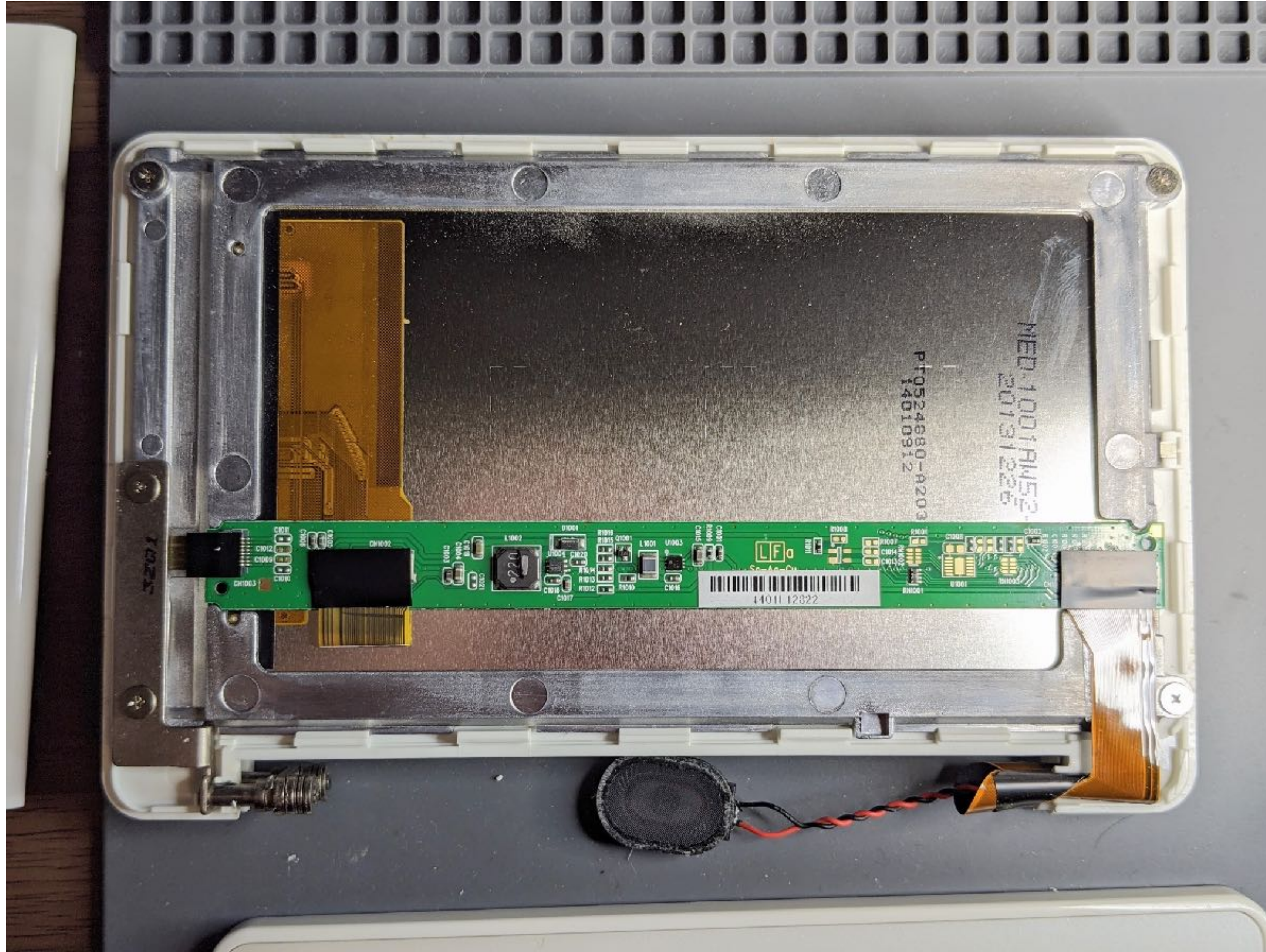
Micro SD スロット

イヤホンジャック

液晶側



液晶側



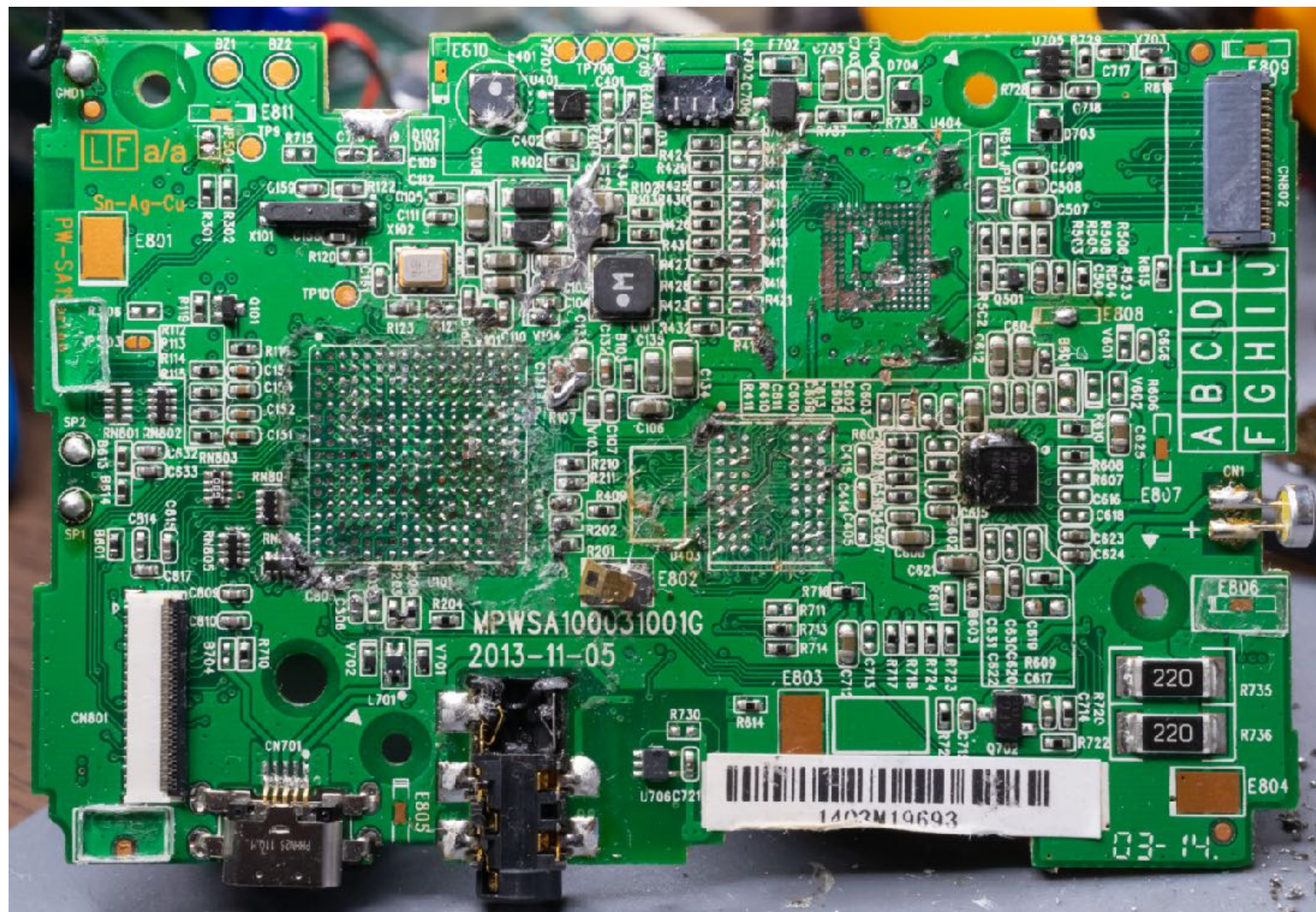
**分解してパーツも詳しく判明
Linux の移植へ**

第一の壁 「回路」

**周辺回路をすべて把握しないことには
動くものも動かない**



ヒートガンで...

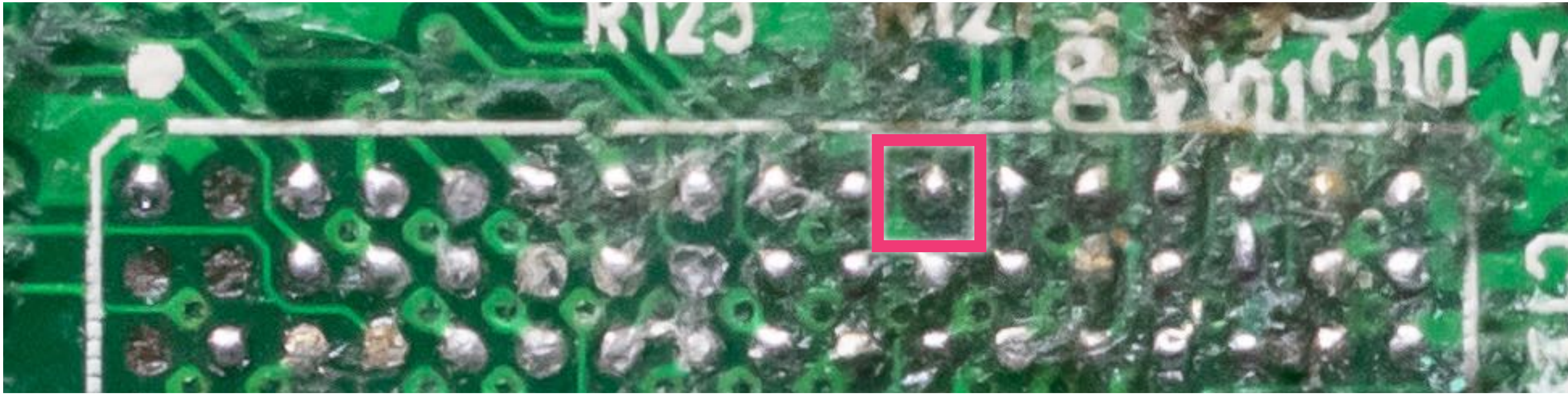


チップをはがします

チップをはがしてわかること

それは

ブートローダを流し込む鍵の在り処



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	VSS	NC	SSP2_SCK	SSP0_CMD	SSP0_DATA3	SSP0_SCK	VDDIO33	USB1DP	VSS	USB0DM	PSWITCH	XTALI	VDD4P2	RESETN	BATTERY	DCDC_LP	DCDC_GND
2	NC	NC	SSP2_MISO	SSP0_DATA7	SSP0_DATA4	SSP0_DATA0	VSS	USB1DM	DEBUG	USB0DP	VSSA2	XTALO	VSSA1	HSADC0	DCDC_BATT	DCDC_VDDA	DCDC_LN1
3	NC	NC	SSP2_MOSI	SSP2_SS0	SSP0_DATA5	SSP0_DATA1	I2C0_SCL	LRADC2	LRADC1	TEST MODE	RTC_XTALO	VDD_XTAL	VDDA1	LRADC6	LRADC0	VSS	DCDC_VDDIO

1行11列にある **PSWITCH** ピンに電圧をかけると
パソコンから**USBでブートローダが流し込める**

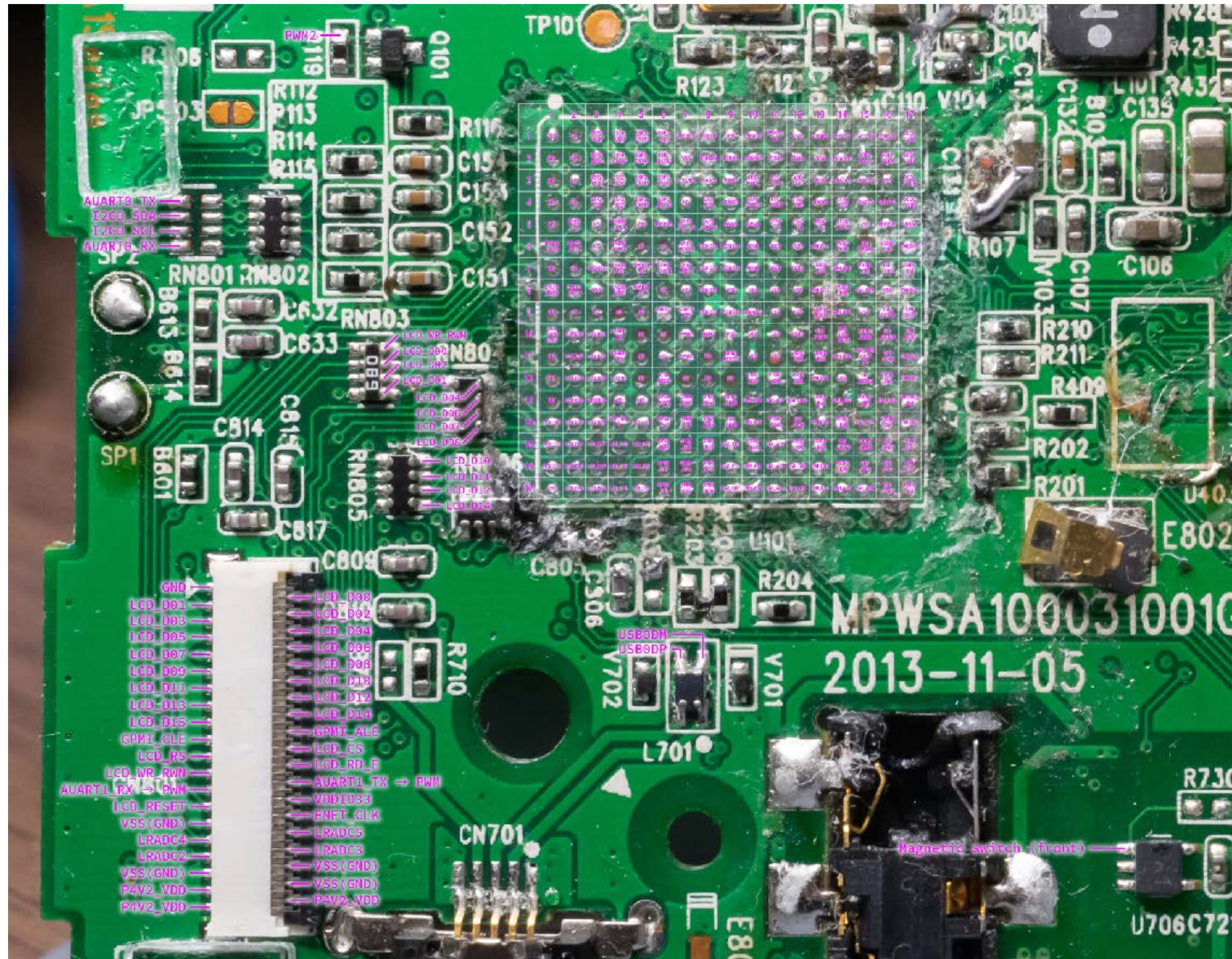
実際に見てみましょう

さらに

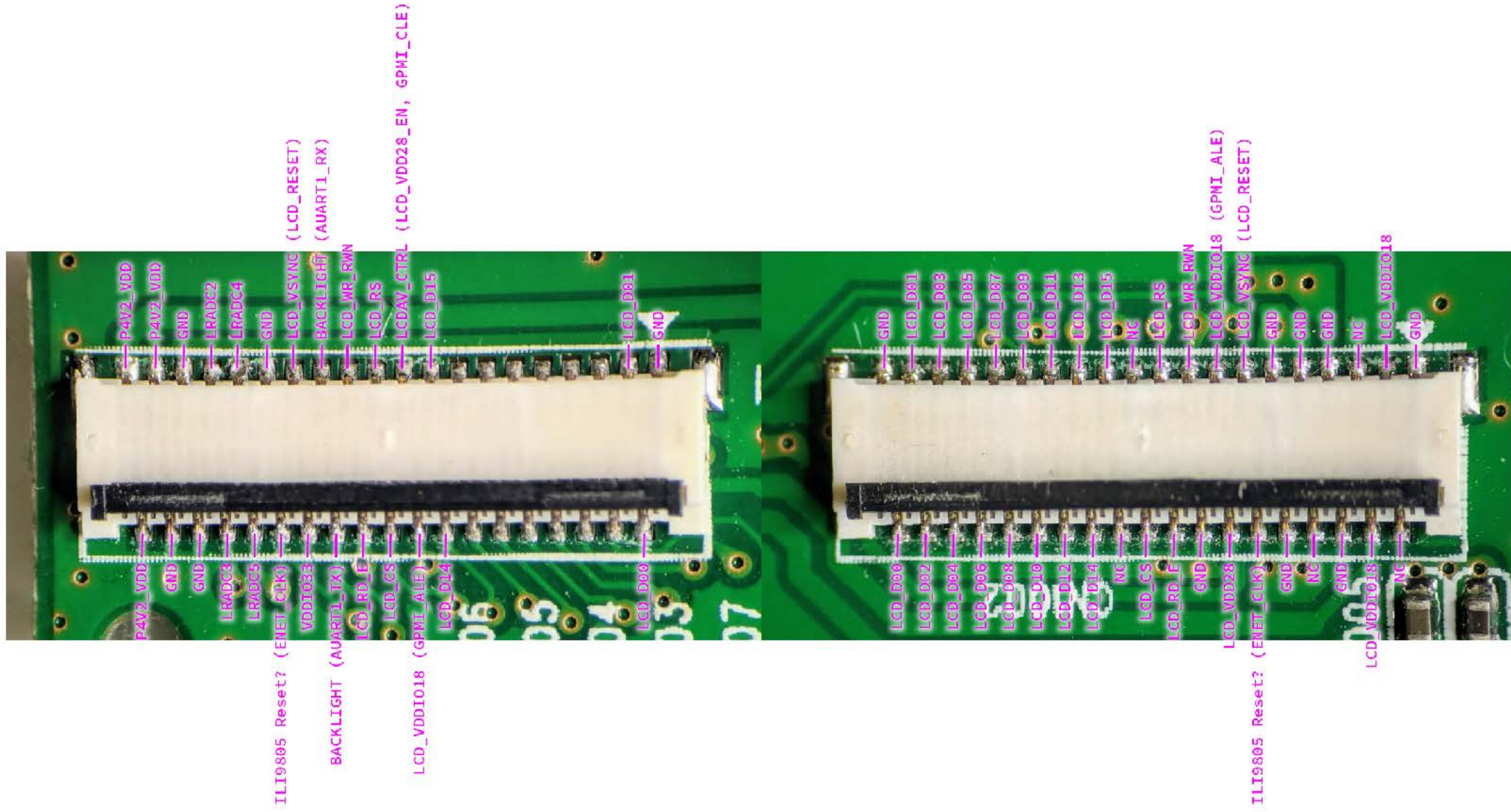
SoC のピンと周辺回路に

テスターを当てまくる！！！！

結線をひたすら調べていく



結線をひたすら調べていく



こうして

**基板の電氣的接続が解明され
第一歩を踏み出した**

第二の壁 「ブートローダ」

第二の壁「ブートローダ」

- U-Boot（ブートローダ）は GitHub にある
- Brain と同じ SoC の評価ボードに向けた config がある

**つまり評価ボード向けに
U-Boot をコンパイルして
Brain に流し込めば動くんじゃない？**

コンパイルは難なく成功！

シリアルポートも接続完了！

動かしてみると…

```
HTLLCLLC  
Undefined Ins
```

は？

何が起きているのか？

HTLLCLLC

← Boot ROMの出力（正常）

Undefined Ins

← "Undefined Instruction"

**Forum 曰く、DRAMの設定がおかしくて
DRAMの読み書きができてない**

第三の壁

「DRAM」

第三の壁「DRAM」

- 評価ボードは **DDR2** の DRAM を搭載
- Brain は **LPDDR** の DRAM を搭載

**評価ボード向けの U-Boot の実装は
LPDDR では動かない**

**「じゃあ LPDDR を扱う実装を
書けばええやんw」**

**「じゃあ LPDDR を扱う実装を
書けばええがw」**

…と言うのは簡単ですが

14.8.150	DRAM Control Register 155 (HW_DRAM_CTL155).....	1300
14.8.151	DRAM Control Register 156 (HW_DRAM_CTL156).....	1300
14.8.152	DRAM Control Register 157 (HW_DRAM_CTL157).....	1301
14.8.153	DRAM Control Register 158 (HW_DRAM_CTL158).....	1301
14.8.154	DRAM Control Register 159 (HW_DRAM_CTL159).....	1302
14.8.155	DRAM Control Register 160 (HW_DRAM_CTL160).....	1302
14.8.156	DRAM Control Register 161 (HW_DRAM_CTL161).....	1303
14.8.157	DRAM Control Register 162 (HW_DRAM_CTL162).....	1303
14.8.158	DRAM Control Register 163 (HW_DRAM_CTL163).....	1304
14.8.159	DRAM Control Register 164 (HW_DRAM_CTL164).....	1305
14.8.160	DRAM Control Register 171 (HW_DRAM_CTL171).....	1306
14.8.161	DRAM Control Register 172 (HW_DRAM_CTL172).....	1308
14.8.162	DRAM Control Register 173 (HW_DRAM_CTL173).....	1309

14.8.150 DRAM Control Register 155 (HW_DRAM_CTL155).....1300

14.8.151 DRAM Control Register 156 (HW_DRAM_CTL156).....1300

14.8.152 DRAM Control Register 157 (HW_DRAM_CTL157).....1301

14.8.153 DRAM Control Register 158 (HW_DRAM_CTL158).....1301

14.8.154 DRAM Control Register 159 (HW_DRAM_CTL159).....1302

14.8.155 DRAM Control Register 160 (HW_DRAM_CTL160).....1302

14.8.156 DRAM Control Register 161 (HW_DRAM_CTL161).....1303

14.8.157 DRAM Control Register 162 (HW_DRAM_CTL162).....1303

14.8.158 DRAM Control Register 163 (HW_DRAM_CTL163).....1304

14.8.159 DRAM Control Register 164 (HW_DRAM_CTL164).....1305

14.8.160 DRAM Control Register 171 (HW_DRAM_CTL171).....1306

14.8.161 DRAM Control Register 172 (HW_DRAM_CTL172).....1308

14.8.162 DRAM Control Register 173 (HW_DRAM_CTL173).....1309

DRAM に関するレジスタ
その数 約200個

- 自作の Python スクリプトによる解析
- Brain 実機のレジスタのダンプ
- GitHubに転がる謎のコード

**あらゆる手を尽くし
試行錯誤を繰り返した結果…**

```
HTLLCSPL: Initialising FRAC0
SPL: FRAC0 Initialised
SPL: Configuring VDDA
SPL: Initialising mx28 SDRAM Controller
hw_pinctrl_emi_ds_ctrl_set: 80019b84
SPL: Setting mx28 board specific SDRAM parameters
SPL: Applying SDRAM parameters
SPL: Setting CPU and HBUS clock frequencies
dram_size=08000000 mode=00000001
LLC

U-Boot 2018.09-00032-g7cc38fda11-dirty (Aug 12 2019 - 15:43:00 +0900)

CPU:   Freescale i.MX28 rev1.2 at 454 MHz
BOOT:  I2C #0, master, 3V3
SPI:   ready
DRAM:  128 MiB
No arch specific invalidate_icache_all available!
NAND:  0 MiB
MMC:   MXS MMC: 0
Loading Environment from MMC... MMC0: No card detected!
*** Warning - MMC init failed, using default environment

Video: MXSFB: 'videomode' variable not set!
In:    serial
Out:   serial
Err:   serial
Net:   FEC0 [PRIME], FEC1
Hit any key to stop autoboot:  0
=>
```

動いた！！！！！！

これで
microSD から
念願の Linux をブートできる！

「それじゃ microSD から
Linux kernel をロードし…あれ？」

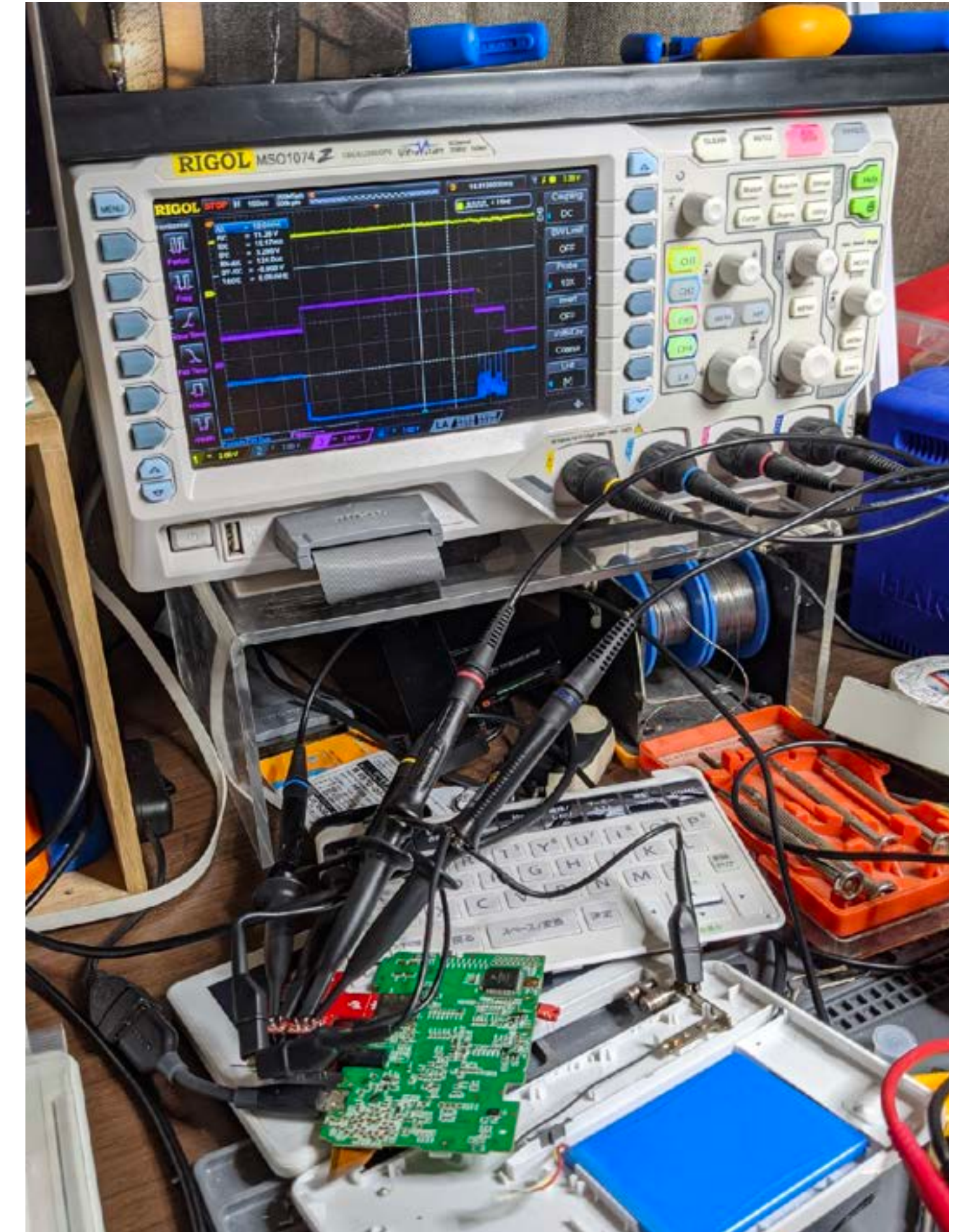
第四の壁 「microSD」

**U-Boot が microSD を認識しないどころか
オシロスコープで見ると信号すら出てない**

**U-Boot が microSD を認識しないどころか
オシロスコープで見ると信号すら出てない**

**— 見地味に思えるが
解決に大変苦労した**

- SD の信号を横取りしてオシロ接続
- 電源を供給する MOSFET の制御
- SoC の I/O のクロック
- I/O マルチプレクサの見直し
- おびただしい量のprintf



幾多の試行錯誤を経て…

```
⇒ mmc dev 1
switch to partitions #0, OK
mmc1 is current device
⇒ mmc info
Device: MXS MMC
Manufacturer ID: 45
OEM: 2d42
Name:
Bus Speed: 480000000
Mode : SD High Speed (50MHz)
Rd Block Len: 512
SD version 3.0
High Capacity: Yes
Capacity: 7.5 GiB
Bus Width: 4-bit
Erase Group Size: 512 Bytes
⇒
```

認識した！！！！

第五の壁

「Linux kernel と Debian」

結論から言うと

Linux は微調整だけで動いたし

Debian 10 もあっさり動作

```
root@brain:~# cat /etc/debian_version
```

```
10.3
```

```
root@brain:~# cat /proc/cpuinfo
```

```
processor       : 0
model name     : ARM926EJ-S rev 5 (v5l)
BogoMIPS      : 226.09
Features      : swp half thumb fastmult edsp java
CPU implementer : 0x41
CPU architecture: 5TEJ
CPU variant    : 0x0
CPU part      : 0x926
CPU revision   : 5
```

```
Hardware       : Freescale MXS (Device Tree)
Revision      : 0000
Serial        : 00000000000000000000
```

```
root@brain:~# uname -a
```

```
Linux brain 5.1.15-00027-gcd1d083333e7 #2 Fri Jan 24 00:39:14 JST 2020 armv5tejl GNU/Linux
```

```
root@brain:~#
```

```
root@brain:~#
```

```
root@brain:~# █
```

Linux は楽にクリア！
ようやくこれで移植は完ry...



puhitaku

いやでもな~~~~~
本体のLCDが映らないことにはな~



Linux がブートしても 画面が灯りすらしない

第六の壁

「LCD」

- 画面に変化があった時だけ信号を送信する
特殊な規格が使われている
- Linux にある公式ドライバが使えない
- LCD の型番がわからないので
どんなデータを送ればよいかわからない

ぶっちゃけ かなり厳しかった

Windowsの起動ログに奇跡的な一行を発見



puhitaku
@puhitaku

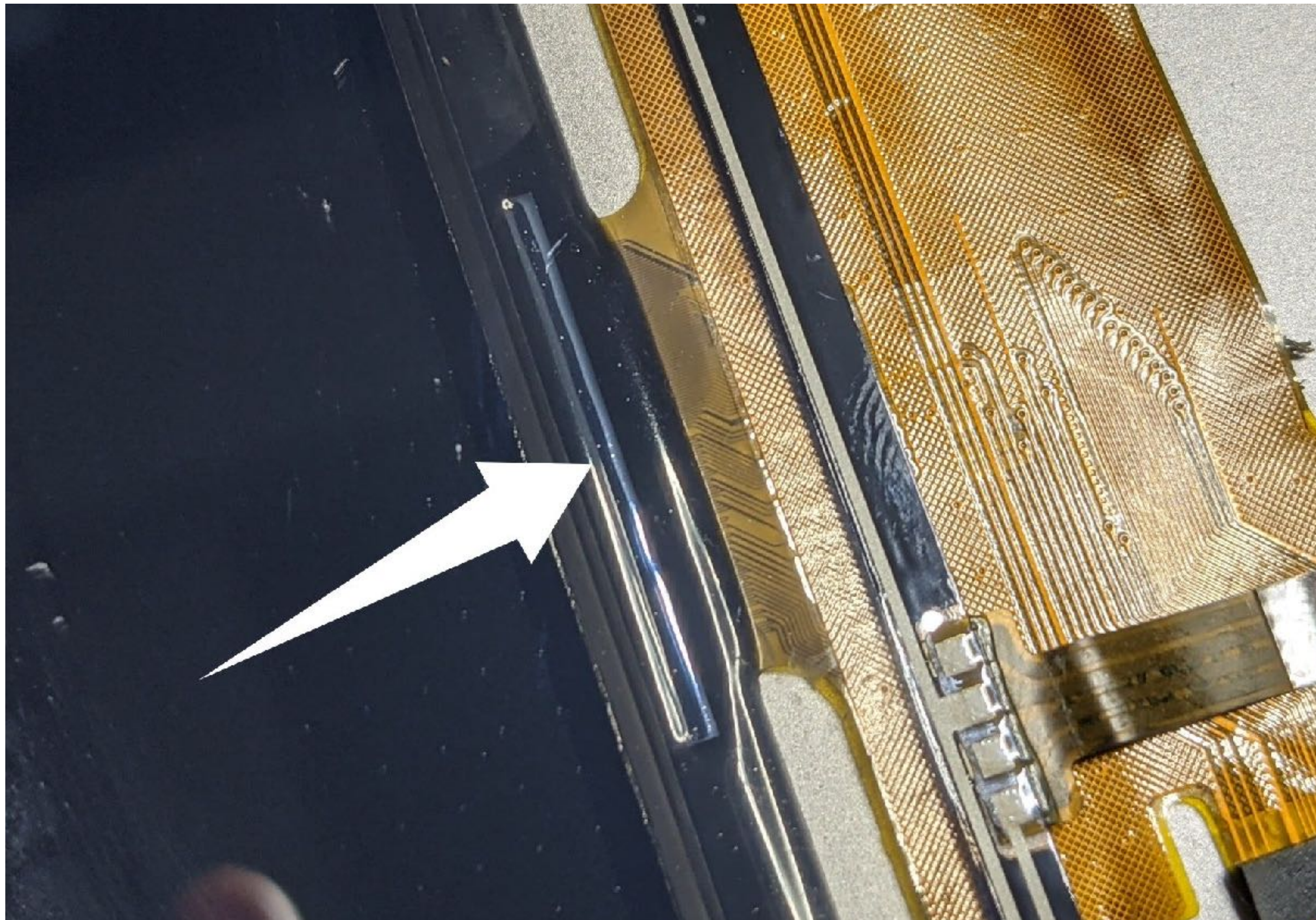
オツツツツ!!!!マジ????

```
GetDisplayGuid:  
Initializing ILI9805 controller 16bit-2
```

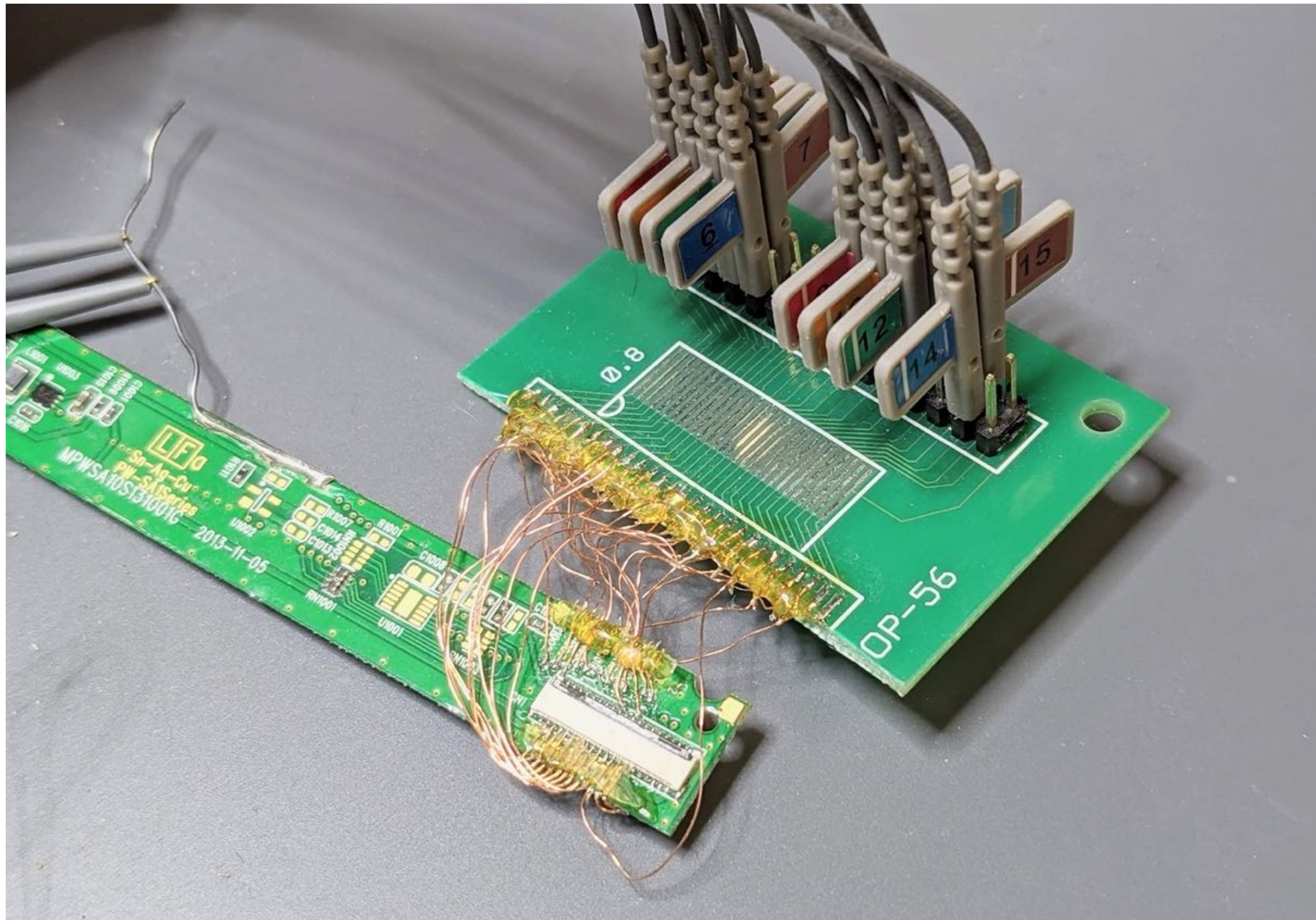
午後10:49 · 2020年5月15日 · Twitter Web App

"ILIxxxx" という型番は SPI 液晶などで使われる
ILITEK の LCD ドライバを指している

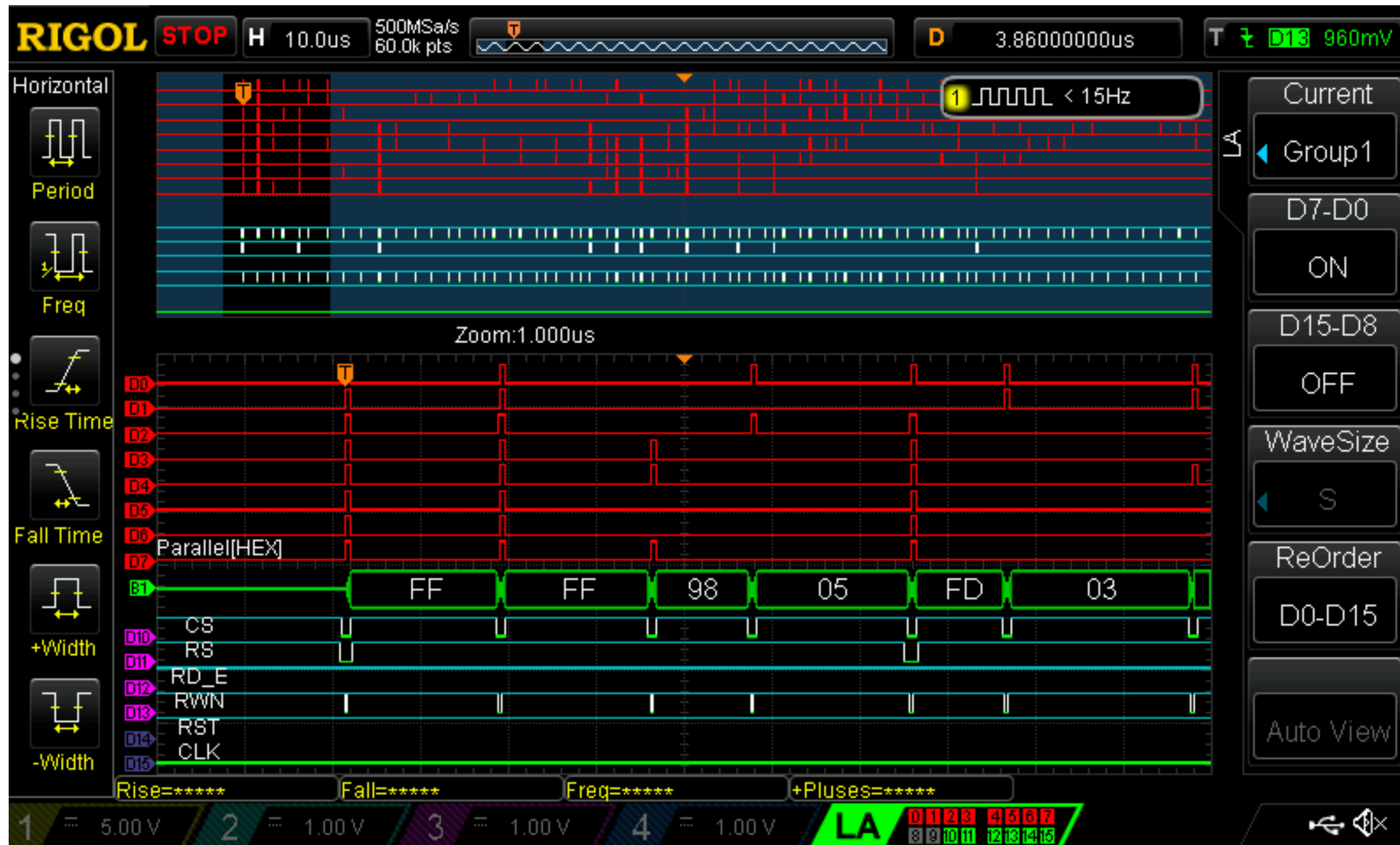
検索すると
ILI9805 のデータシートは無かったが
ILI9806 のデータシートを発見



ILI9805 の外観



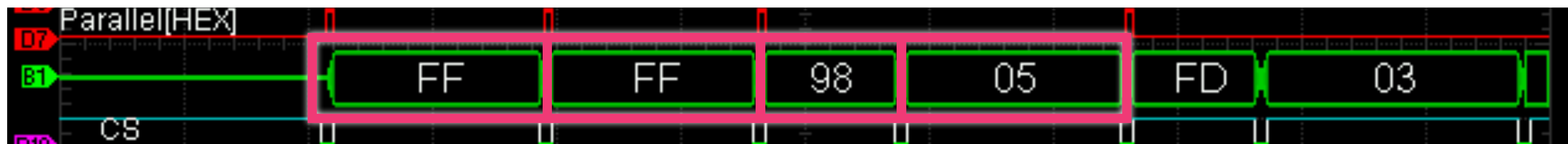
LCD の信号を全部引き出し ロジアナで解析



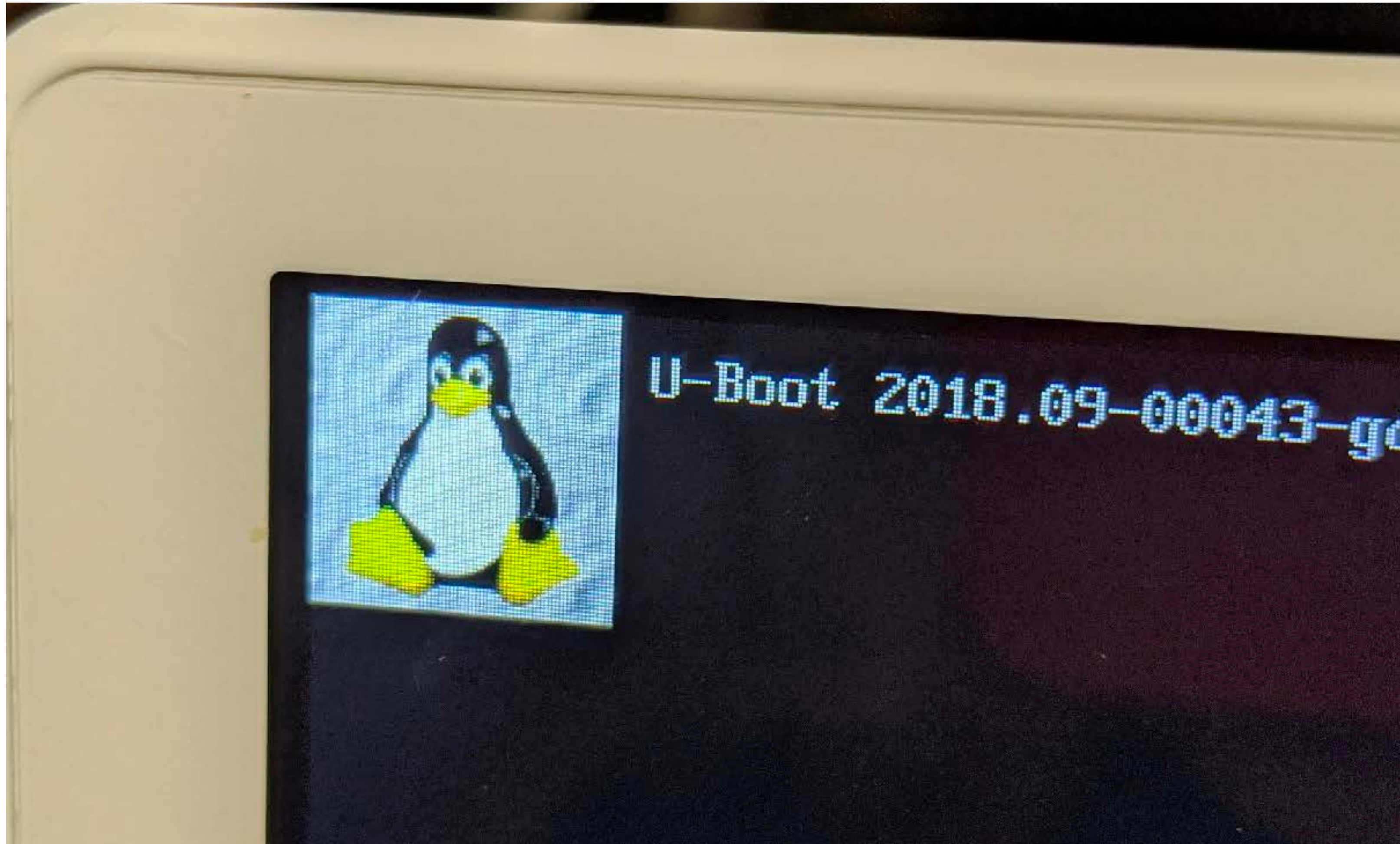
LCD の信号を全部引き出し ロジアナで解析

4.3.42. EXTC Command Set enable register (FFh)

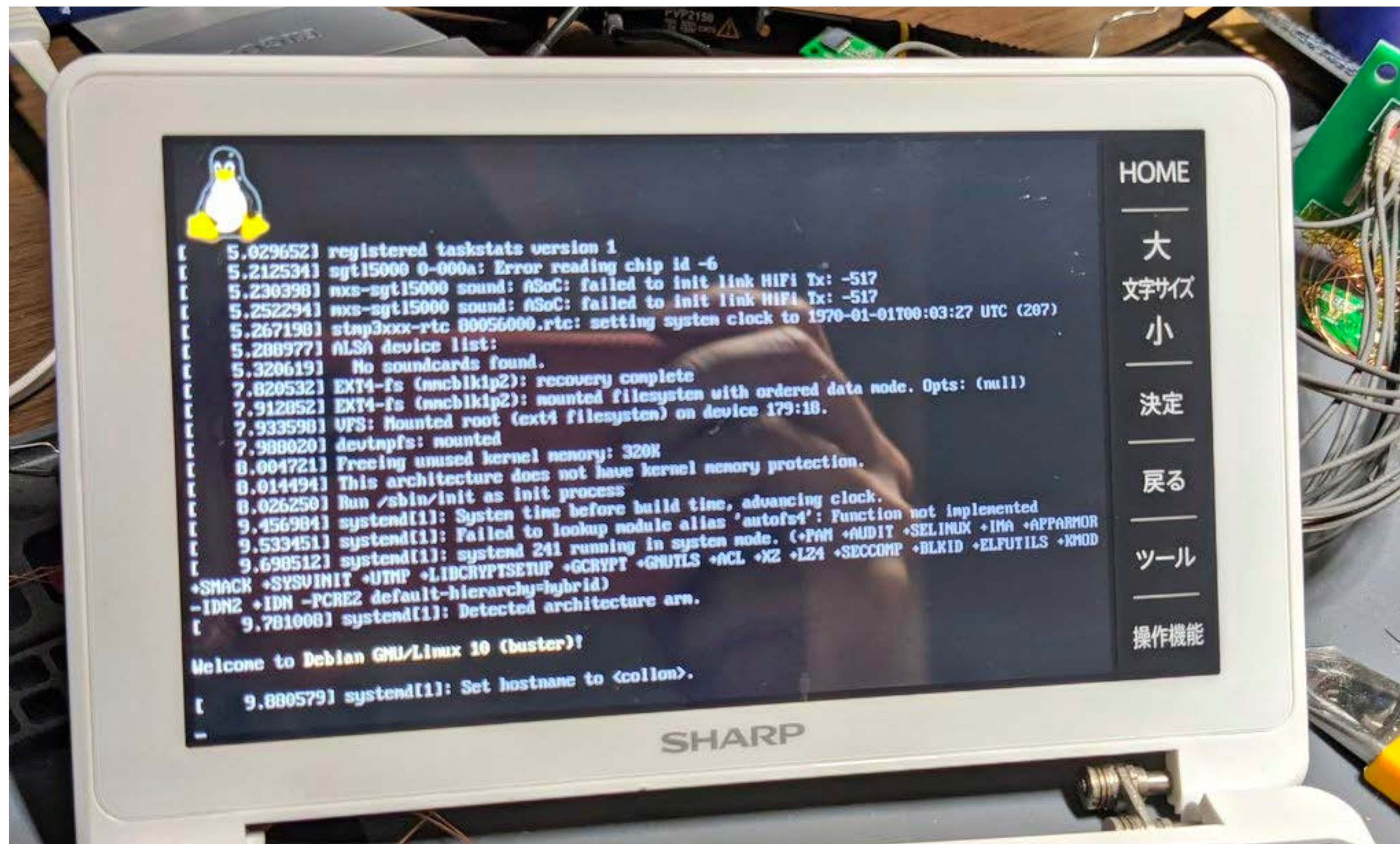
FFh	ENEXTC (EXTC command set enable register)												
	DCX	RDX	WRX	D [23:8]	D7	D6	D5	D4	D3	D2	D1	D0	HEX
Command	0	1	↑	XX	1	1	1	1	1	1	1	1	FFh
1 st Parameter	1	1	↑	XX	1	1	1	1	1	1	1	1	FFh
2 nd Parameter	1	1	↑	XX	1	0	0	1	1	0	0	0	98h
3 rd Parameter	1	1	↑	XX	0	0	0	0	0	1	1	0	06h



ロゼッタストーン的に解説を敢行



U-Boot での LCD 初期化 & ピクセル転送 に成功！



そのままの勢いで Linux での実装も成功！



そのままの勢いで Linux での実装も成功！

次なる壁は？

- キーボード（現在実装中！）
- 音
- 開閉検知
- バッテリ管理
- パフォーマンスチューニング
- 安定性向上

まだまだ楽しめそう

最近の進捗



Search or jump to...



Pulls

Issues

Marketplace

Explore



Brain Hackers

It's where the hackers who love SHARP Brain (electronic dictionary) gather at.

**Brain ハックのコミュニティ
"Brain Hackers" を作りました**

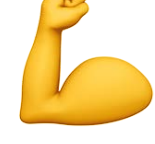
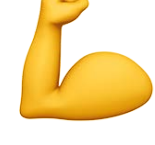
Discord のメンバーは現在70人弱くらい

学生さんを中心にハックに興じています

招待リンクなど 詳細は GitHub へ！

<https://github.com/brain-hackers>

実装側の進捗は…

-  Linux を分解なしで起動
-  Linux (U-Boot) を Windows から起動
-  GitHub Actions に乗せてカーネルを簡単に入手可能に
-  キーボードドライバの実装（進行中）
-  Wiki への情報集約（進行中）

etc...

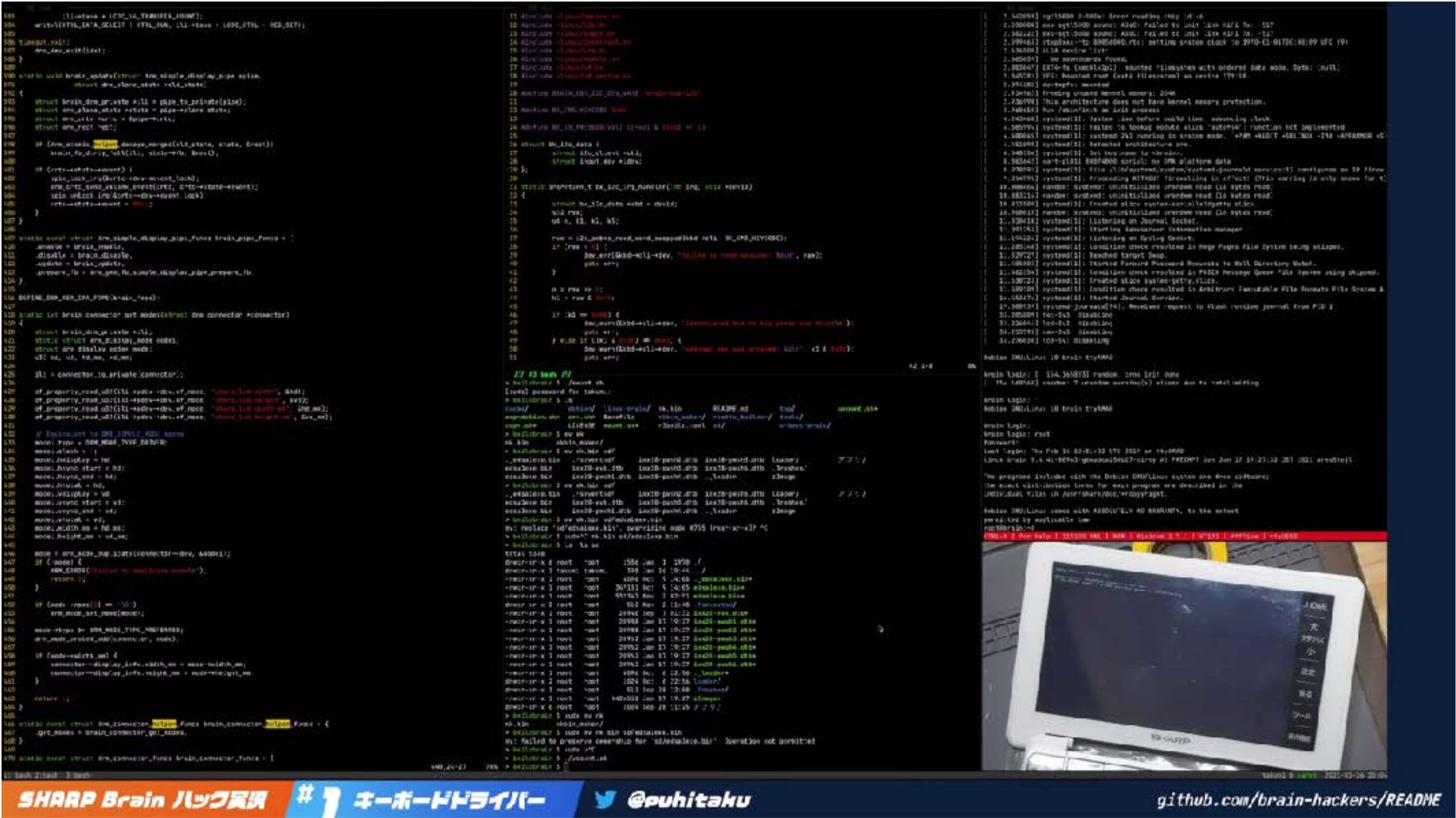
**今後は Raspberry Pi のように
SD カードに焼くだけで Linux が動かせる
ディストリビューションを提供予定**

ハック実況はじめました



第2回が今夜18時より開始！詳細は私のTwitter @puhitaku へ！

ハック実況はじめました



第2回が今夜 18 時より開始！詳細は私の Twitter @puhitaku へ！

これからも
私 ひいては Brain Hackers の
ハックに乞うご期待！



Brain Hackers