

| Details | Hackaday.io

18-22 minutes



Enforcement, Brand (Nokia - Global)

发送至 我 ▼

📎 9月14日周四 21:18 (2小时前)



Confidential & Without prejudice

Nokia ref.: Nokia Corp/LIT01207 (Please always quote)

Dear Sir or Madam,

Further to our correspondence below, Nokia regrets to note that the website <https://hackaday.io/project/185645-notkia> still makes use of the term “**NOTKIA**” in relation to phones and related software.

As over 14 months have now passed, Nokia considers that more than a reasonable time has lapsed to allow you to change "NOKTIA" to something not imitating nor recalling our NOKIA brand in any manner whatsoever.

Nokia thus expects to see the change materialize without further delays.

We thank you in advance for your prompt and full cooperation towards an amicable resolution and, in the meantime, Nokia reserves all its rights.

Best regards

Nokia Trademark Team

Legal & Compliance

Nokia



At Nokia, we create technology that helps the world act together.

This e-mail, and any attachments thereto, is intended to be read and/ or used only by the intended recipient(s). It may contain legally privileged and/or confidential information.

If you are not the intended recipient of this e-mail, you are hereby notified that any dissemination, distribution or copying of this email, and any attachments thereto, is strictly prohibited. If you receive this email in error please immediately notify me at above number or per e-mail. Please also permanently delete the original copy and any copy of any e-mail, and any printout thereof.

...

▲ Latest censor attempt by **Nokia Corporation.**



Elliot Williams wrote 07/21/2022 at 07:08 • 2 points

I'm not a lawyer, but Nokia's scare letter referenced our terms of use here and suggested that we might think that you are in violation.

We don't.

We think Nokia is overstepping, and that they should back down. An apology wouldn't hurt, either.

This project is pretty clearly Not endorsed by Nokia. :)

reply



Reimu NotMoe wrote 07/21/2022 at 13:57

Thank you. It seems that you're a Hackaday staff. Do you think this project can still participate in your competition despite Nokia's efforts to take it down?

reply edit delete



Elliot Williams wrote 07/27/2022 at 12:48 • 1 point

Yeah, sure!

▲ Support by Hackaday staff.

Note: This is NOT a custom ROM for Nokia phones. It is a freshly designed PCB with exactly the same dimensions as the original PCB of the Nokia phone, so it can be put in the Nokia's shell.

Last edit: 2023-09-15

This project is officially obsoleted.

Also please read my [declarations on similar projects](#).

Hardware & Software design files are all open sourced on our [GitHub repo](#), including:

- Schematic, PCB and Gerber files
- Complete BOM lists and SMT position lists
- Linux kernel patches and Buildroot config
- Related software code
- Datasheets of rare components

The "Components" list here only shows the major components. View the BOM lists for a complete list of all components.

If you have any suggestions or questions, please join our [discord server](#) or [send us](#)

[an email.](#)

Brief introduction

- Nokia 168x shape: **Easy to operate with one hand & fit in pockets easily**
- **Runs mainline Linux**
- Processor: [Ingenic X1000E](#), 2200+ CoreMark, 64MB RAM
- Storage: 32MB NOR + 4GB **SLC** NAND
- Connectivity: **Lora**, WiFi 2.4GHz, Bluetooth
- Display: 2.0" 240x320 **IPS** LCD, 3/4 visible
- Audio: [Yamaha MA-3 \(YMU762\) music synthesizer](#) + Regular I2S PCM codec
- Power: Battery management solution from TI, and **standard BL-5C battery**
- Misc: Type-C OTG, RGB indicator LED, temperature & humidity sensor, **powers on without battery, zero-conflict keypad**

Why do we need it?

- People who don't like HUGE smartphones do exist on this planet
- Modern smartphones are becoming increasingly privacy unfriendly & hacker unfriendly, and the worst thing is: **privacy is already a business for these businessmen:**
 - There are many "privacy oriented" products on the market that aren't even designed by people with privacy awareness
 - And it's the same for hackability
- Find more in the "Afterword" section in the end of this article

See it in action:

- [Running the LVGL music player demo](#) (on the v1.0 prototype board)
- [Playing the "Bad Apple" animation](#) (on the v2.0 board w/ shell and so on)
- [Keypad demo](#)
- [LCD backlight dimming](#)
- [GBC emulator](#)
- [T9 Keypad typing Linux commands](#)

Form factor

The Notkia uses the Nokia 1680/1681/1682 form factor. **Yes, it has a proper shell.** The 1680 has a camera, and the other models don't. **It can be comfortably operated one handed, no matter in public transport or in bed. Having it accidentally falling into your face won't cause extreme pain. It can be put in almost all pockets and bags without a problem, and won't scratch your clothes or pull your beach pants down.**



▲ It can be comfortably operated with only one hand



▲ The weight is only around 66 grams

The main reason of we choosing the 168x series is because it has the largest inner space among all the feature phones we inspected so far. Its inner space is big enough to put a spring antenna inside! Also, its power connector has its own piece of plastic on the PCB, this means it can be changed to another type of connector without problems. However it doesn't look very pretty, and I hope we can get enough money to replace the plastic clip surrounding the charging port.



▲ I tried my best to make the Type-C surroundings look good

This design is only [patented](#) in US and AU, and will expire on 2023-06-23. After it's expired, we can produce brand new plastic cases based on this design, assuming the fundraising collected enough money.

Processor

The [X1000E](#) processor from [Ingenic Semiconductor](#) is used. It has a 1GHz MIPS32r2 core with 64bit FPU, and 64MB built-in LPDDR RAM.

Why the X1000E?

- Very low power consumption: 0.3W running the [CoreMark](#) benchmark and **0.003W in standby**
- Acceptable performance: **2200+ CoreMark**, a bit faster than the RPi Zero
- Built-in RAM: saves PCB space, while enough to launch 16,384 Apollo 7s
- [Correct toolchain](#) for it can be installed in minutes, not hours or days
- **Supported by mainline Linux kernel**
- **Datasheets are [publicly available](#): no more hide & seek with vendor**

With some compiler tricks and a bit overclocking (1.2GHz, stable), it's able to achieve 3000+ CoreMark.

CPU usage in common tasks in default clock frequency (1GHz):

- FLAC decoding: 8%
- Vorbis & OPUS decoding (FP): 15%
- GBC emulation: 23%
- ... more tests are needed

It's rated to operate under -40°C to 85°C.

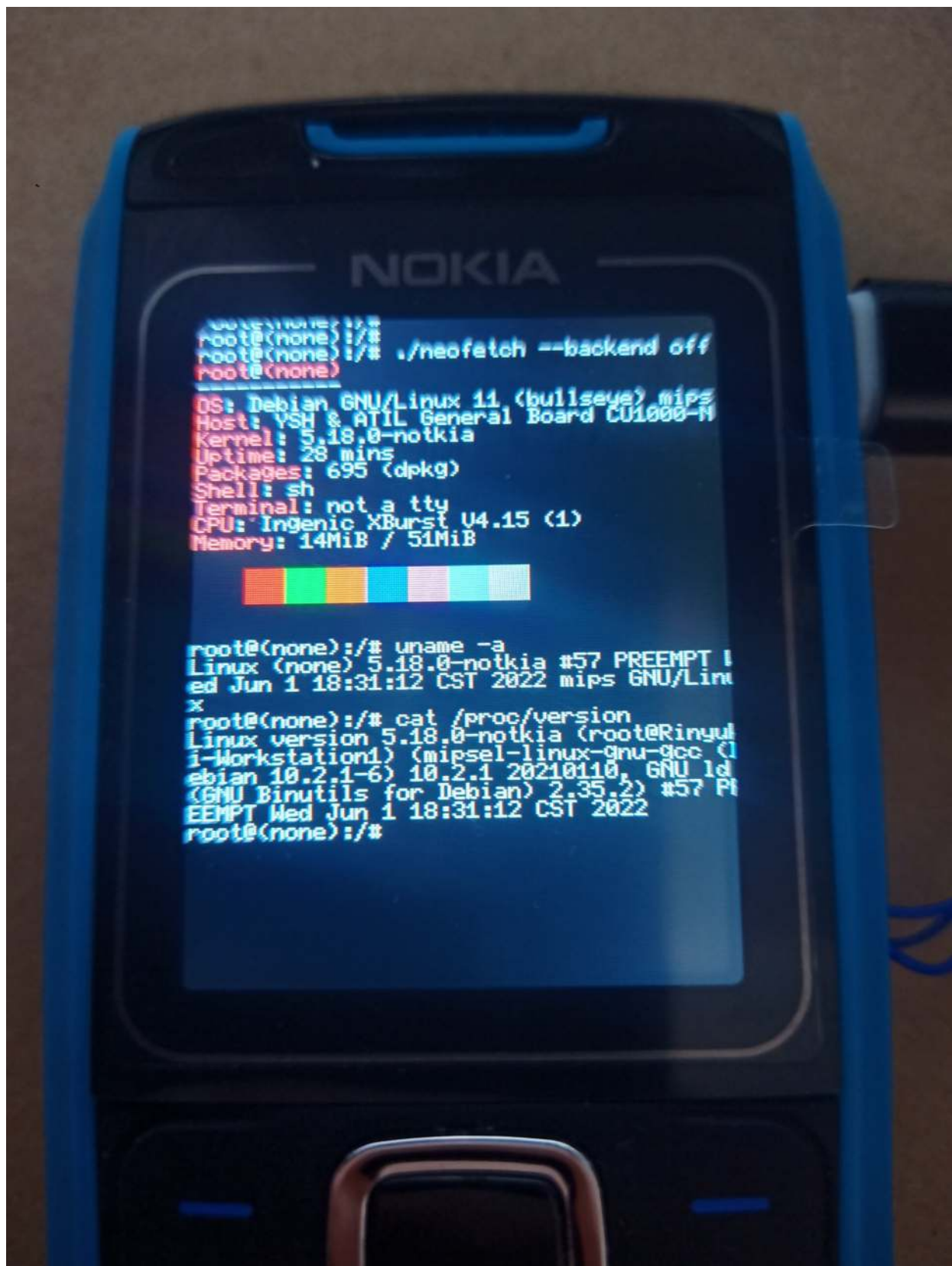
Software

First of all, it runs mainline Linux, yay!

Software that needs to be written (sorted by importance):

(Thank god, Hackaday has a better editor that handles bullet points correctly)

- A desktop environment:
 - Based on LVGL and uses memory efficiently
 - Or maybe revive MeeGo/OpenMoko stuff?
 - Modify Xfbdev to use ``/dev/mem`` directly ("the fbcv mechanism")
 - Suitable to use with no pointer devices
 - T9 input method for English, Chinese, Linux commands and PL keywords
 - Of course you can contribute new languages
 - Existing IME (frameworks) are too memory hungry, any ideas?
 - Terminal
 - S60-like multimedia player
 - Lora messenger
 - ...
- Yamaha MA-3 software driver:
 - ALSA MIDI interface
 - OPL3 compatible interface
 - Better to be a clean room of the original Yamaha code
- Ingenic X1000 series kernel drivers:
 - DVP interface driver
 - UART DMA support
 - ... there could be more
- Entertainment:
 - Port RetroArch
 - Port a J2ME emulator and make it work with the Yamaha MA-3
 - Port PICO8 or its equivalent (depending on the author's willingness)
 - Rewrite some classic games from scratch: Snake, Tetris, Q*Bert, etc
- Security:
 - T9 input for boot time cryptsetup
 - Ingenic X1000 EFUSE & secure boot support

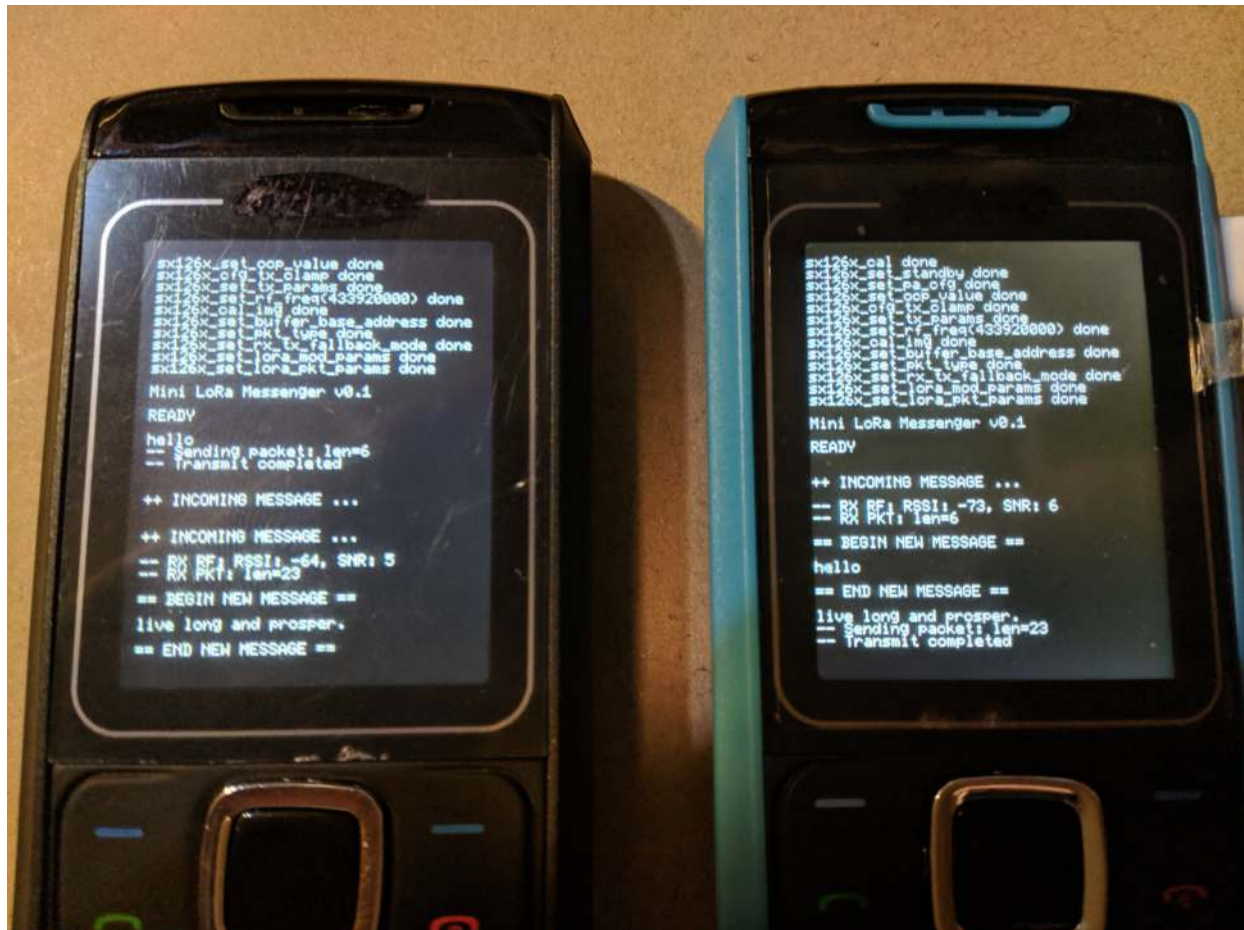


▲ It runs mainline Linux! The screen is too small for a Debian logo.

Connectivity

Lora

The Notkia is equipped with a Semtech SX126x [LoRa](#) transceiver. It enables you to exchange data with distant people in the unlicensed spectrum. The SX126x is connected to a TCXO for reliable operation in narrow bandwidth, and DIO1 & DIO2 are directly connected to the processor for fast interrupt processing.



▲ You can send messages over LoRa. That works even when everything went out.

WiFi+BT

An Ampak AP6214 is used to save space. It contains the same BCM43438 chip as AP6212, but a lot smaller.

UART DMA (mentioned above) may be required for reliable Bluetooth operation.



▲ Browsing the Internet with the Lynx TUI browser.

No 4G?

Actually we considered this option very seriously. Currently the only known 4G/LTE module that is small enough to be put inside is the [SIMCom A7680C](#). However it only supports Chinese bands (1/3/5/8/38/39/40/41). There are a few overlaps with the bands in some Asia & Europe countries, but there are none for the US. Also it uses a Marvell LTE baseband, and there's no known way to incorporate new VoLTE profiles. It would be impossible to use this

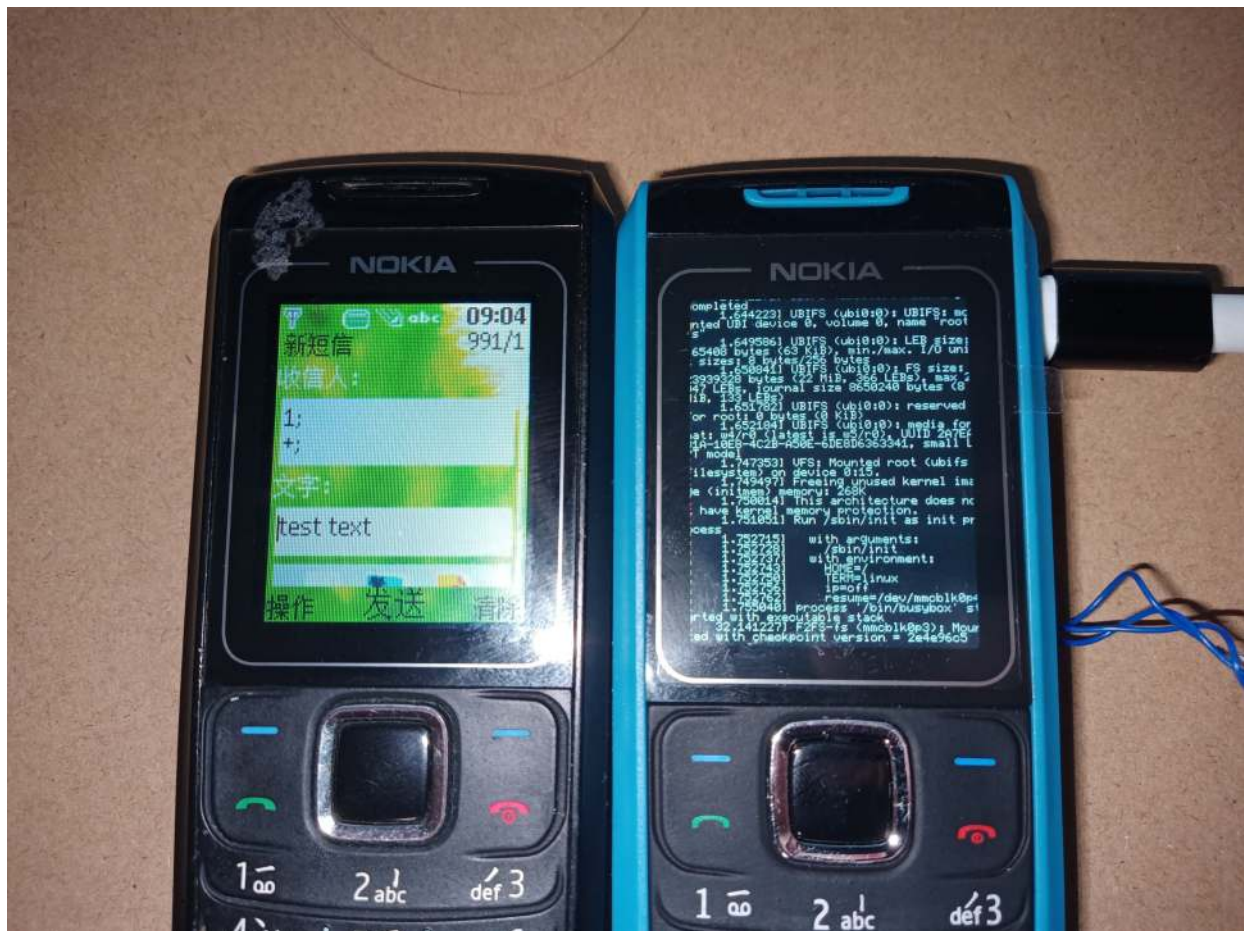
device globally.

-

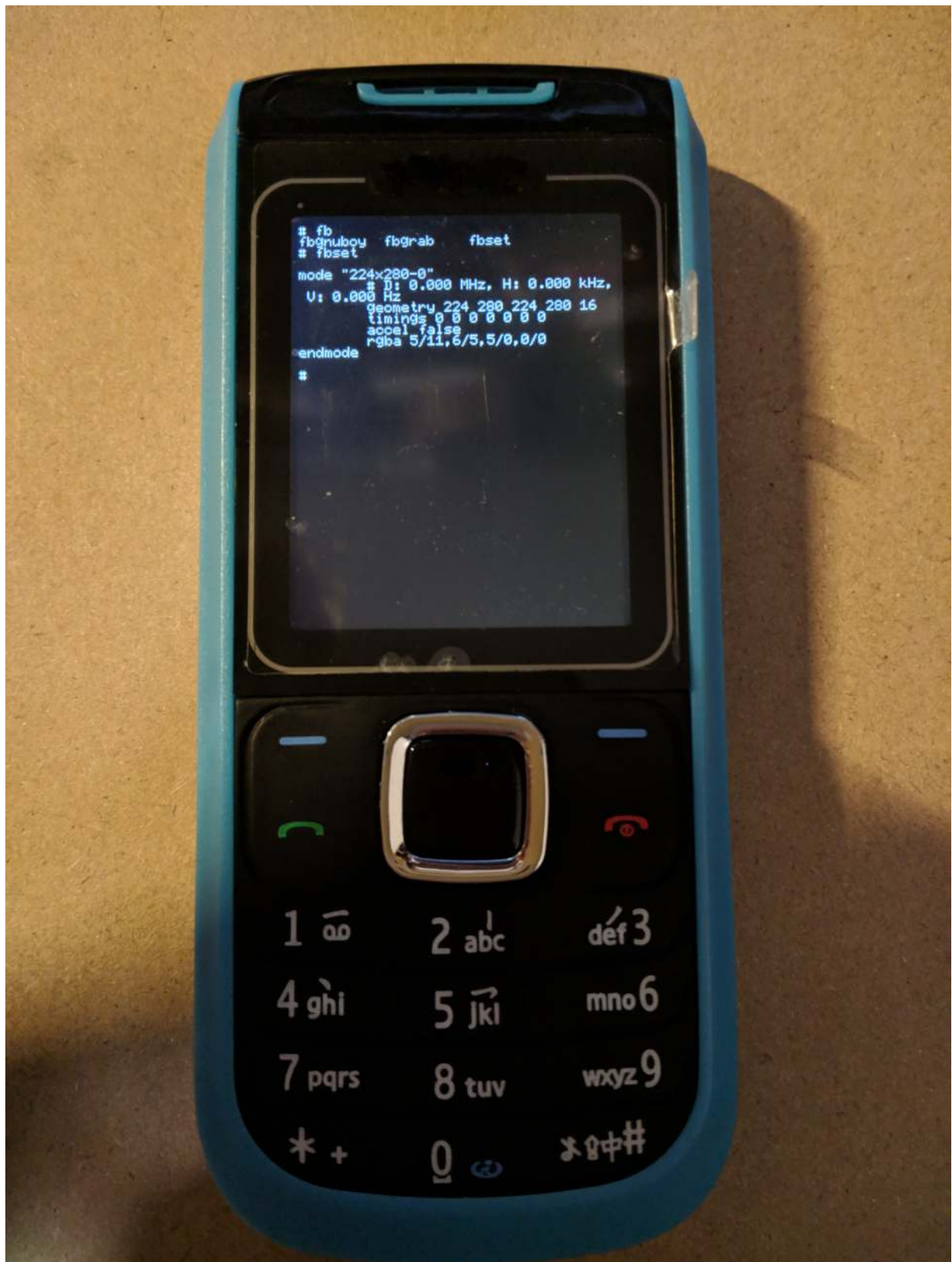
I tried my best to make the RF traces impedance matched. Before the final version, we will consult RF experts to ensure all of them are designed properly.

Display

The original Nokia 168x series used a laggy 128x160 TFT LCD, which is even smaller than GBA's LCD. So we changed it to a 240x320 IPS LCD. However, it's a bit larger than the original one. So, some pixels are hidden by the plastic borders, yielding a visible space of ~220x280 pixels. But it's still a lot more pixels than the original LCD, this allows more information to be displayed on the screen, and more importantly, better color and response time.



▲ Screen comparison. Left: Original Nokia 1680, Right: Notkia.



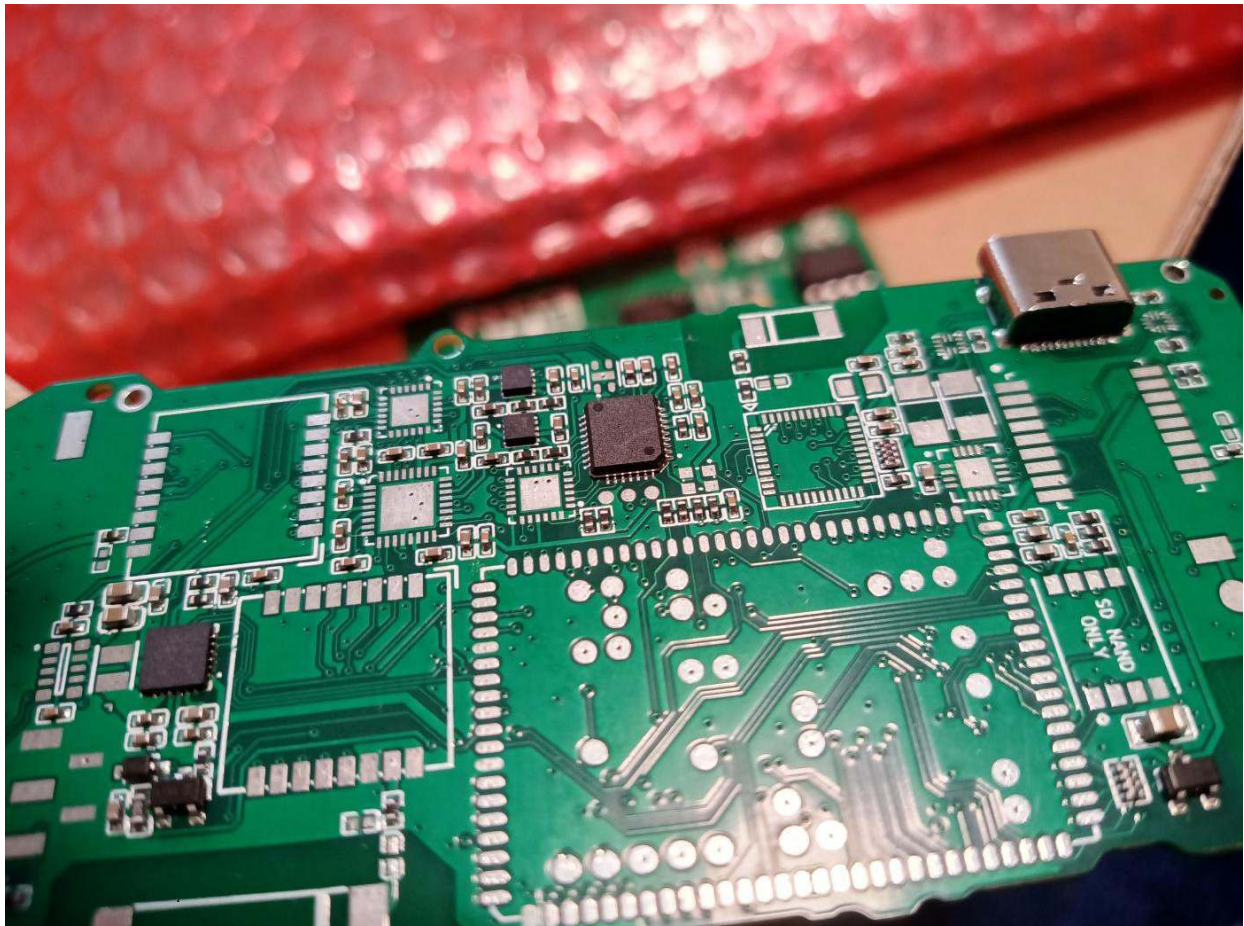
▲ The usable screen resolution is cropped.

Audio

Ever missing the 00s phone ringtones? We got you right - the Nokia features a [Yamaha MA-3 music synthesizer](#), which can be found in [many old feature phones](#). Some software efforts are needed to get it working.

The regular PCM codec is a Nuvoton NAU88C22. It's good enough to act as a mini mixer and equalizer for the internal speakers and convert the analog signals from the Yamaha MA-3 to digital (so you can listen to them from a Type-C USB headset). An analog MEMS microphone is also included for you to use the Notkia as a Lora walkie-talkie.

In earlier prototype revisions, analog audio over Type-C was supported. But that needs another 2 chips - one for switching between the D+/D- and L/R, and another for figuring out which SBU pin is ground/microphone connection. This is ridiculous and [highly illogical](#). So this section was removed. You can use a \$9 [Apple Type-C USB to 3.5mm audio jack connector](#) for your 3.5mm earphones, or even a Chord Mojo for HiFi playback :)



▲ An earlier prototype board. The big chip in center is the Yamaha MA-3. The "illogical" chips are around the Type-C port.

SAMSUNG

Voice Calls Ring

☐ Rooster

☐ Bird

☐ Horse

☐ Cuckoo

☐ Ocean

☒ Gan Cui



Save

CDMA2000



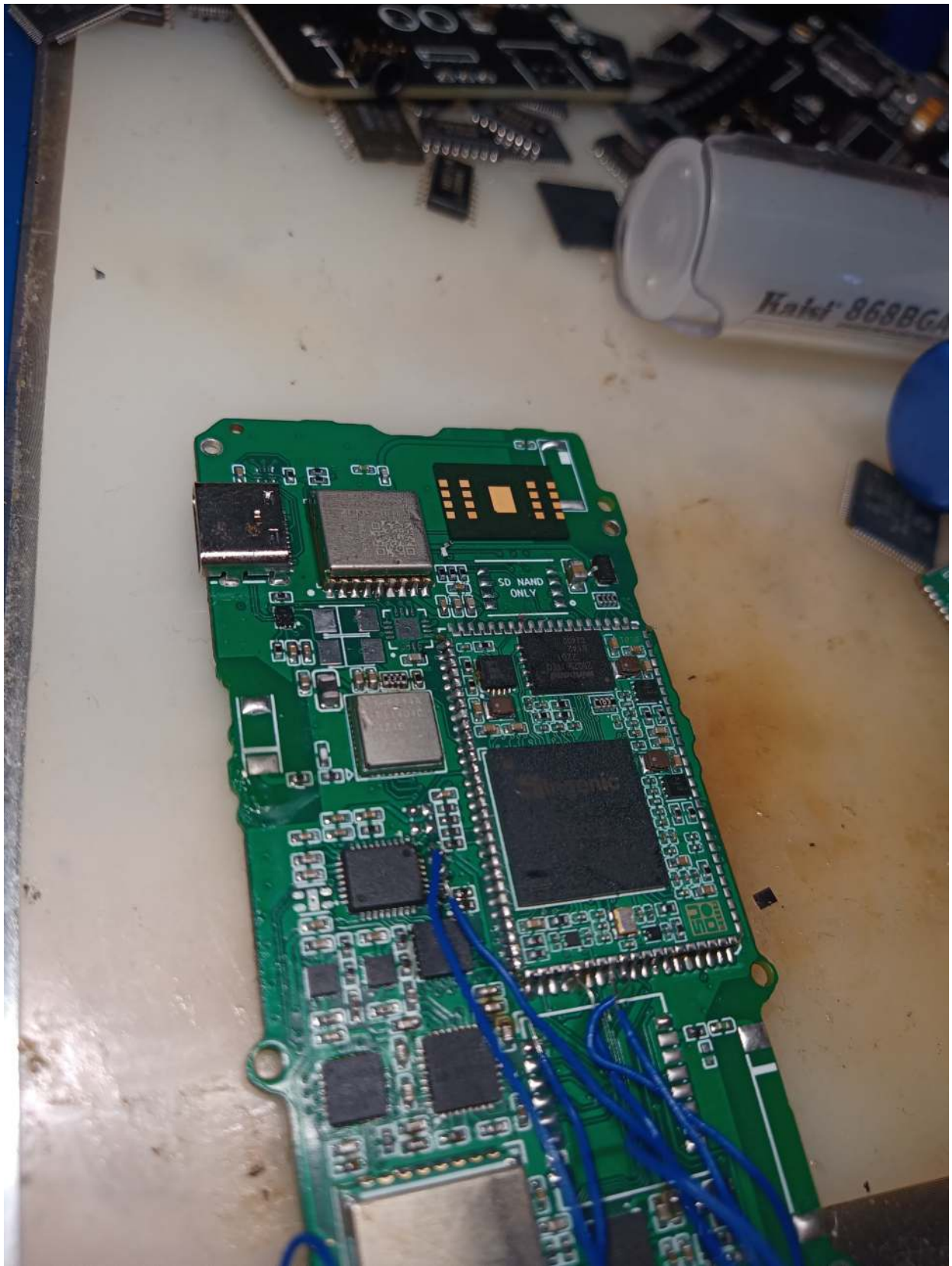
▲ The Samsung SCH-X199. It has the same Yamaha MA-3 synth chip, and its ringtones are awesome.

Storage

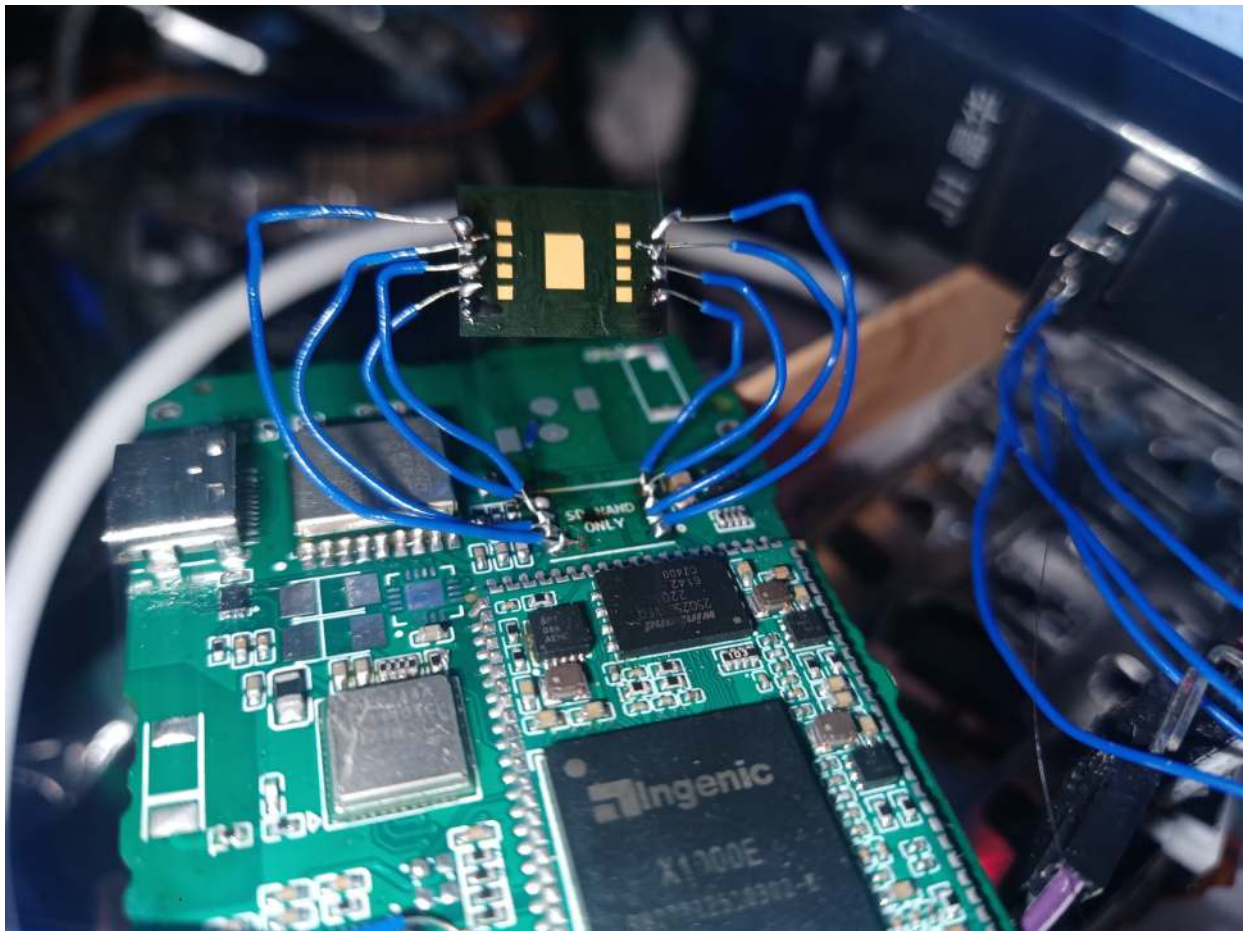
There are two storage devices, a 32MB SPI NOR flash, and a 4GB SLC SD NAND flash.

The NOR flash is rated for 100K P/E cycles and 20+ years of data retention, so it will be used to store important data like the bootloader and OS kernel.

The NAND flash is suitable for storing user files. It's not as rugged as the NOR flash, but it's still SLC. It should be a lot more durable than most microSD cards and TLC/QLC eMMCs.



▲ BIG SD NAND: You expected a WSON-8, but it was me, somewhat-bigger-than-WSON8!



▲ It looks like a crab LOL

Power

We selected a very overkill PMIC, the TI BQ25890, for this tiny little device. It has a very important feature: power on without battery. I always wondered why most phones can't power on with only a power cable. Anyway, this feature is supported in our case. This also enables you to switch batteries without powering off with power cable connected. Moreover, it provides the power switch functionality, and a steady 5V 1A power supply for USB OTG.

The battery is the standard BL-5C - you can get them at pretty much everywhere, and just be careful with the quality. It uses the MAX17048G zero-configuration battery gauge, although I'm not very confident about its accuracy, since it guesses everything from voltage. Here's the story: We have a product called the [Dilithium Mini](#), and it's a Raspberry Pi UPS board. Unlike other UPS boards, it features 5V 4A output and a very accurate TI [Impedance Track](#) gauge. However some sort of user configuration are needed to get it working. The product didn't sell well, maybe that's because the configuration process is too difficult for an average RPi user.



▲ BL-5C batteries are everywhere and easy to get

Camera

An OV5640 5MP AF camera will be used. However that's only with the Nokia 1680 case, and depends on the DVP driver (mentioned above).

Misc Hardware

Zero-conflict keypad: All keys can be pressed at the same time

The Type-C port supports CC handshaking for USB OTG. No fast charging (that's too dangerous for a ~1000mAh battery). No analog audio (reason mentioned above).

There's a RGB indicator LED next to the Type-C port.

There's a SHT20 temperature/humidity sensor.

Long-press the red power key for more than 10 secs to force a reboot.

Afterword: The Story

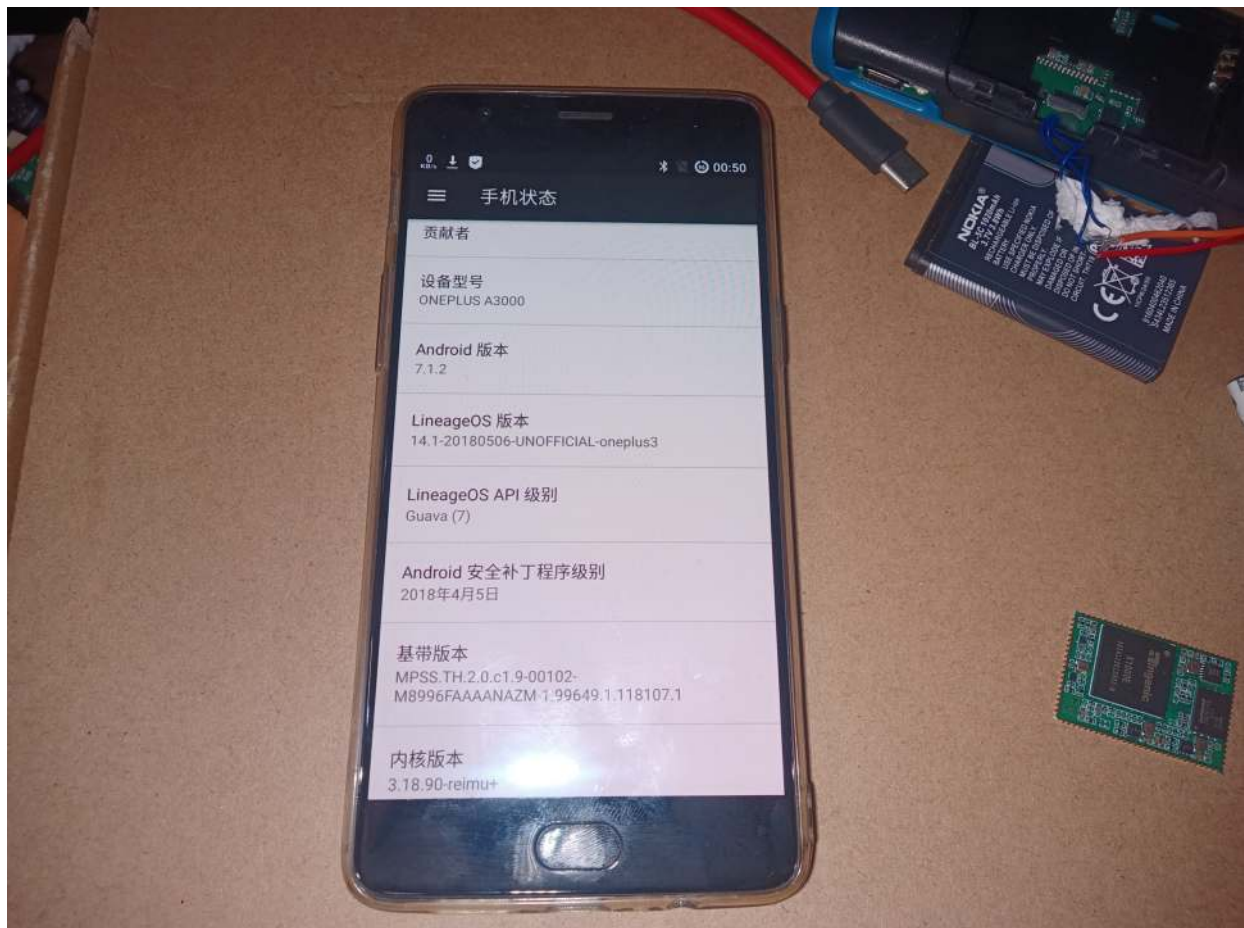
Motivation

This project was unofficially started almost 2 years ago. The original reason was because

the modern smartphones are becoming increasingly hacker-unfriendly and privacy-unfriendly. I'm a CyanogenMod/Lineage user since around 2011. If you experienced that era, you may have the same feelings.

▲ CyanogenMod was the game changer at that time (photo by me, sadly I lost this phone)

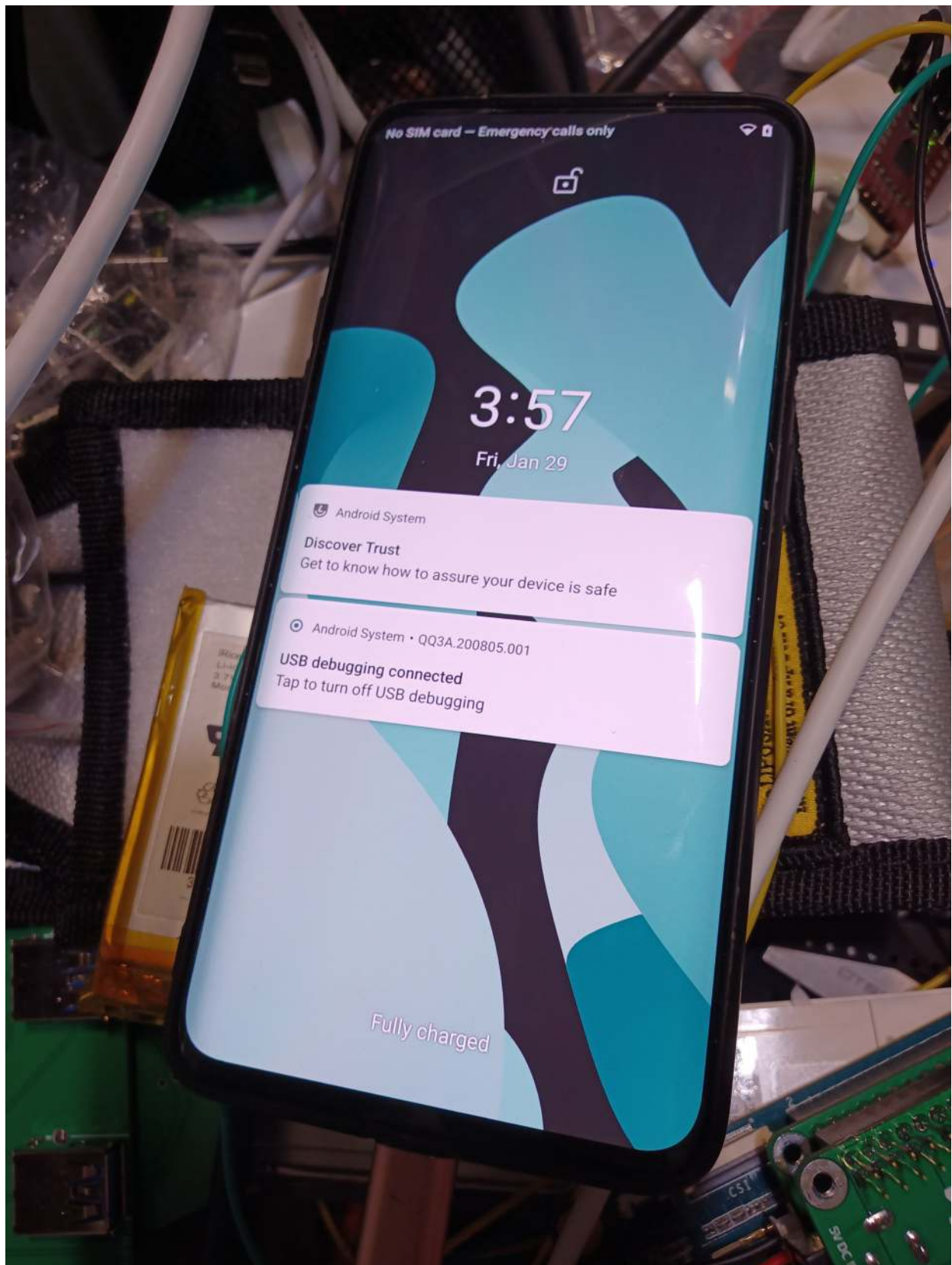
I was a loyal OnePlus user since OnePlus 3 because it was the best "hackable" smartphone at that time. I bought an OnePlus 7 Pro in late 2019, and then I went very angry with the Android resizable partitions ("reinvent LVM and readonly") nonsense. I already went angry with the merge of boot and recovery years ago. It became very hard to modify the system partition on the fly. More and more tools, the Magisk, the App Cloner, are invented and continuously improved by humans using their free time to solve more and more man-made (or Google-made) problems. Many good things disappeared these years, such as the KDE4, Win7, Google+, and people's privacy awareness. Personally I don't think this is right. I don't want to compromise at all. I'm stubborn enough for this to be bothering me everyday. Then I began to think: why not make a phone by myself?



▲ The Oneplus 3. I hacked a lot on it, and I really liked it. It's running with my custom baseband NV and has support for more LTE bands and VoLTE configurations.



▲ Random photo taken in 2016. Android was pretty hackable at that time.



▲ The Oneplus 7 Pro. It became an ARM64 CI builder machine.

It's HARD

I actually heard about the Librem 5 a long time ago, but I was too poor to buy it (it seems to be cheaper in the recent years). So I did I very nerdy test. A Raspberry Pi, a touchscreen HAT, a 3000 mAh battery, and a EC20 4G module. I can surf the Internet using Firefox and chat using Pidgin, but typing and calling became a headache. The xvkbd is the best virtual keyboard I can find. I wrote a small script to send AT commands and setup PCM over serial

port in order to call someone. It conflicted with the modem manager so I can only choose SMS or calling at one time. The battery lasted only for 3 hours or so. It failed completely.

▲ A demonstration. Not the exact setup at that time.

Moreover, the plastic case becomes a problem. I asked multiple plastic mold factories, and the average price for a suite of phone cases is 200K CNY (that's almost 30K USD). The only thing I can imagine at that time is a FDM printed case with lots of layer lines. I began to feel discouraged.

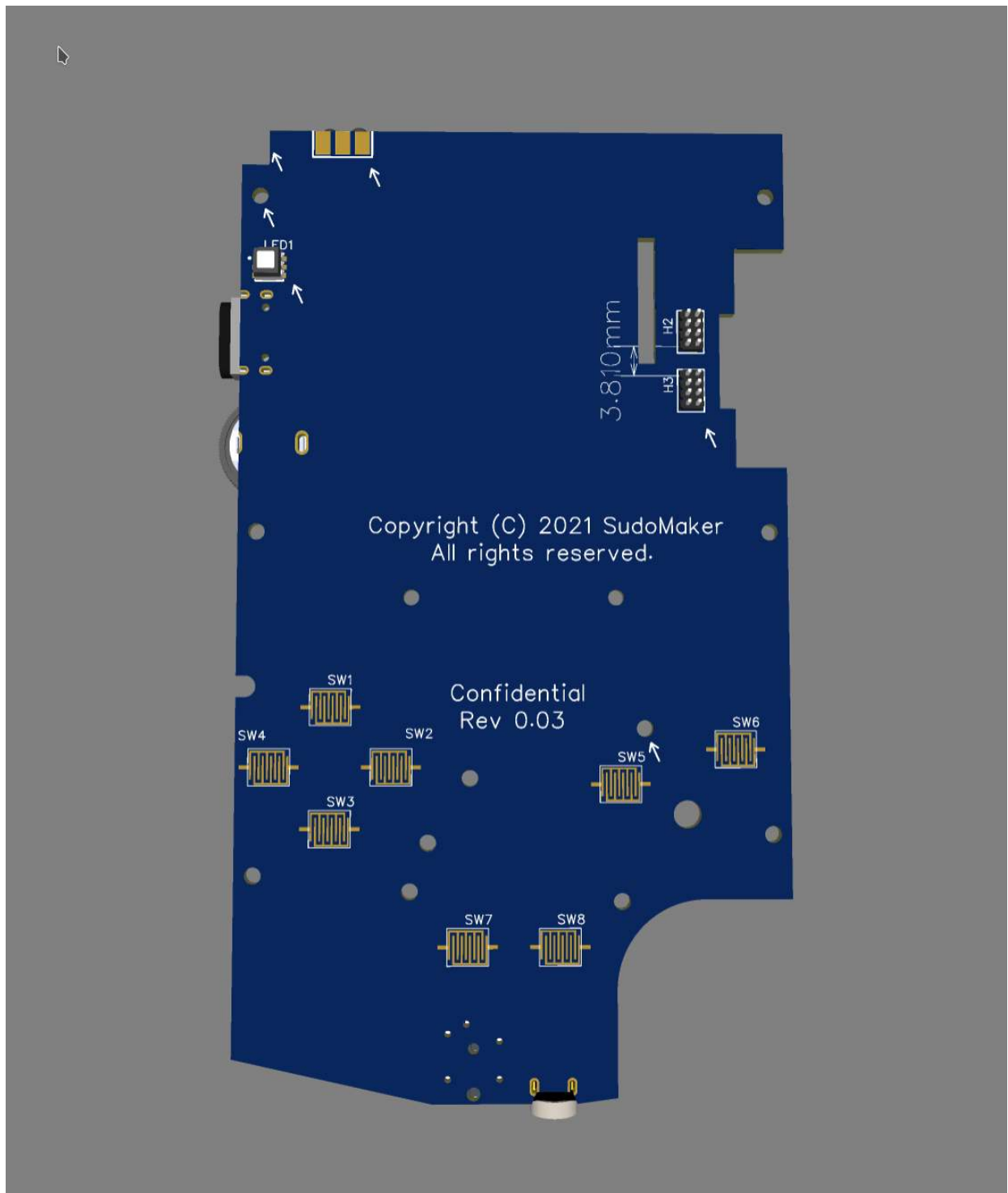
Seek for help instead of this?

I shared this idea in a local electronics group. It's full of EE gurus that were graduated from the best universities here. Some of them got their chip R&D jobs in Germany, Belgium and the USA, and a few of them even had their own chips built in fabs like the TSMC. **Almost everyone laughed at me, and some of them even teased me. Most of them didn't believe there's a privacy "crisis" ongoing.** I demonstrated a way to prove the phones are listening to us everyday: attach a gdb on the Android audioserver process, set a breakpoint in the sound_trigger.so provided by device vendor, and see when it breaks. They seemed to be believing my claim, but I guess nobody was ever bothered to try for themselves. **Maybe this is why these app vendors can spy on people without the slightest scruple.**

The same thing was happening everywhere else. Maybe it's because I had too few friends, most of my friends weren't actually interested in this idea. I began to feel hopeless.

Hope

In the end of 2020, I heard about the RetroPie project. It puts a RPi Zero inside a GameBoy-like case. Then I immediately began my research on reusing existing game console plastic cases. I was planning to use the GBC plastic case. I thought at least I could get a "Pokemon Communicator"-like stuff. I spent a lot of time using a scanner to replicate a PCB with the same shape. Everything was good, the Type-C port was placed at the place of the GB communications port, and there are 14550 Li-ion batteries. Then I was stuck in finding a correct power switch and a suitable LCD screen. I even wanted to use these white goods touch button IC to emulate a touchscreen. Time passes, tons of money wasted, and this idea was stuck forever.

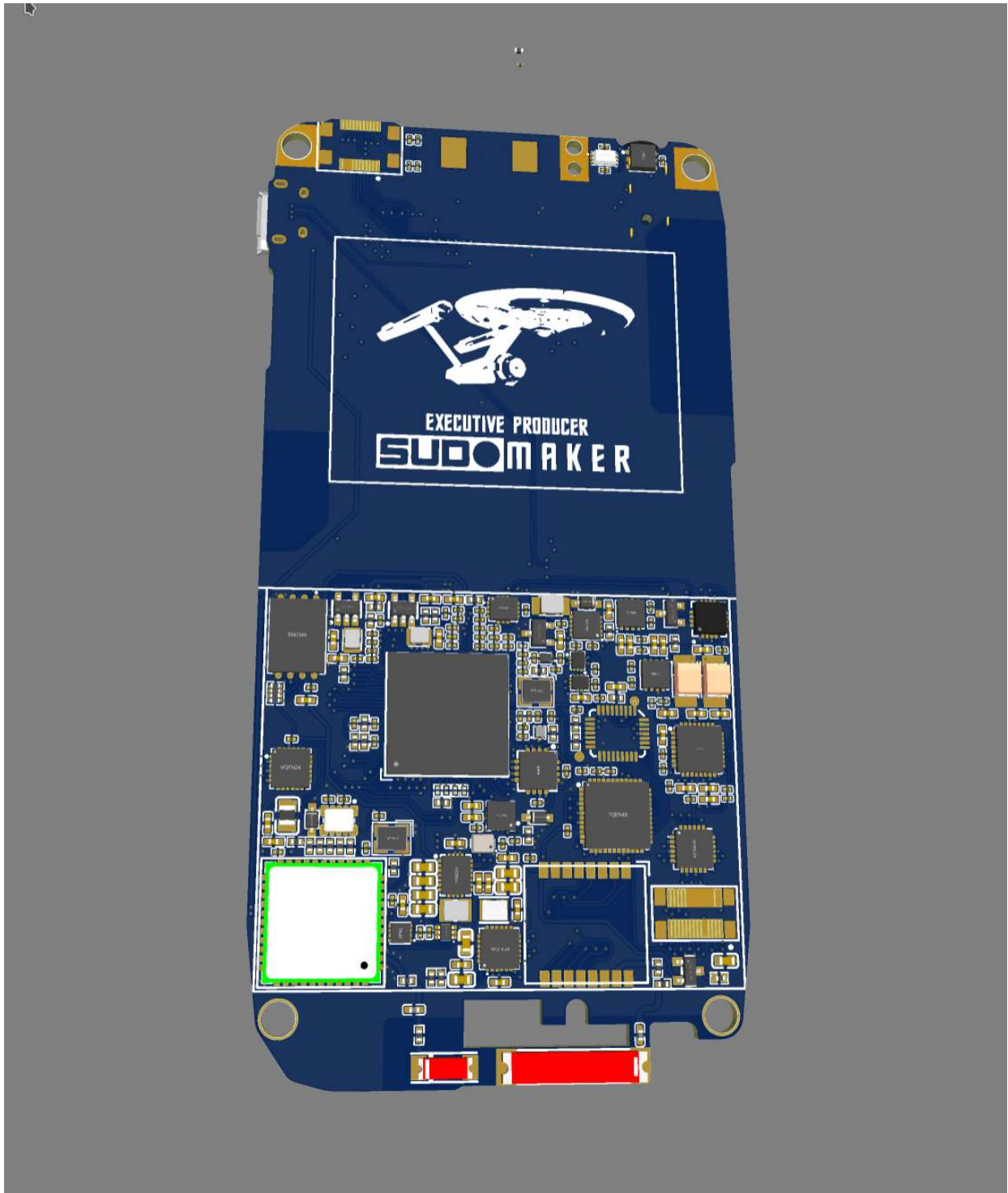


▲ The GBC replacement PCB. It was stuck forever.

Hope x2

In the end of 2021, I heard about the "[OTG Messenger](#)" project. It repurposed the Nokia E63 plastic case to make a Lora messenger. It inspired me that plastic cases of old phones can be repurposed, and the value of Lora is huge. Then I started to find suitable phone cases. At first, I used the E63 as well. But it has multiple problems, the only available area to place components is under the keyboard, that's only 50x40mm, and height is limited to 2mm. It ruled out the possibility to place any SoM there. The only option is to design the processor surroundings from scratch. Initially I designed using the Ingenic X2000 (8000+ CoreMark), but it requires 4 power rails with different voltages and tons of decoupling caps. I eventually gave up. Then I changed to the X1000, it only needs 3 power rails and all caps can be put in

one layer. I actually did it, but never felt satisfied with it. The SDIO wires were too long and crossed 2 layers, and I don't have any confidence that it will run over 20MHz. I had the same concern for the LCD (8080 bus) and camera (DVP). It never went into production.



▲ The E63 replacement PCB. It never went production.



▲ Earlier test PCB to ensure the shape is correct. I wanted to put some of the components on the back side, but this will rule out the possibility to use stock battery, and I never achieved inner peace of mind with this.

In the meantime, I started to get bored with these HUGE modern smartphones. They're impossible to operate with one hand comfortably, and it's extremely painful to have them fall into your face. The influence of the old iPhone SE is fading, and it appears that nobody cares to produce smaller smartphones any more. This is not correct either. Fun fact: When I was in

elementary school, my classmates and I can text each other with our eyes still staring at the teacher and blackboard. This can decrease the probability of being discovered by the teacher. We were indeed bad kids, but this effectively demonstrates the advantages of a hardware T9 keyboard.



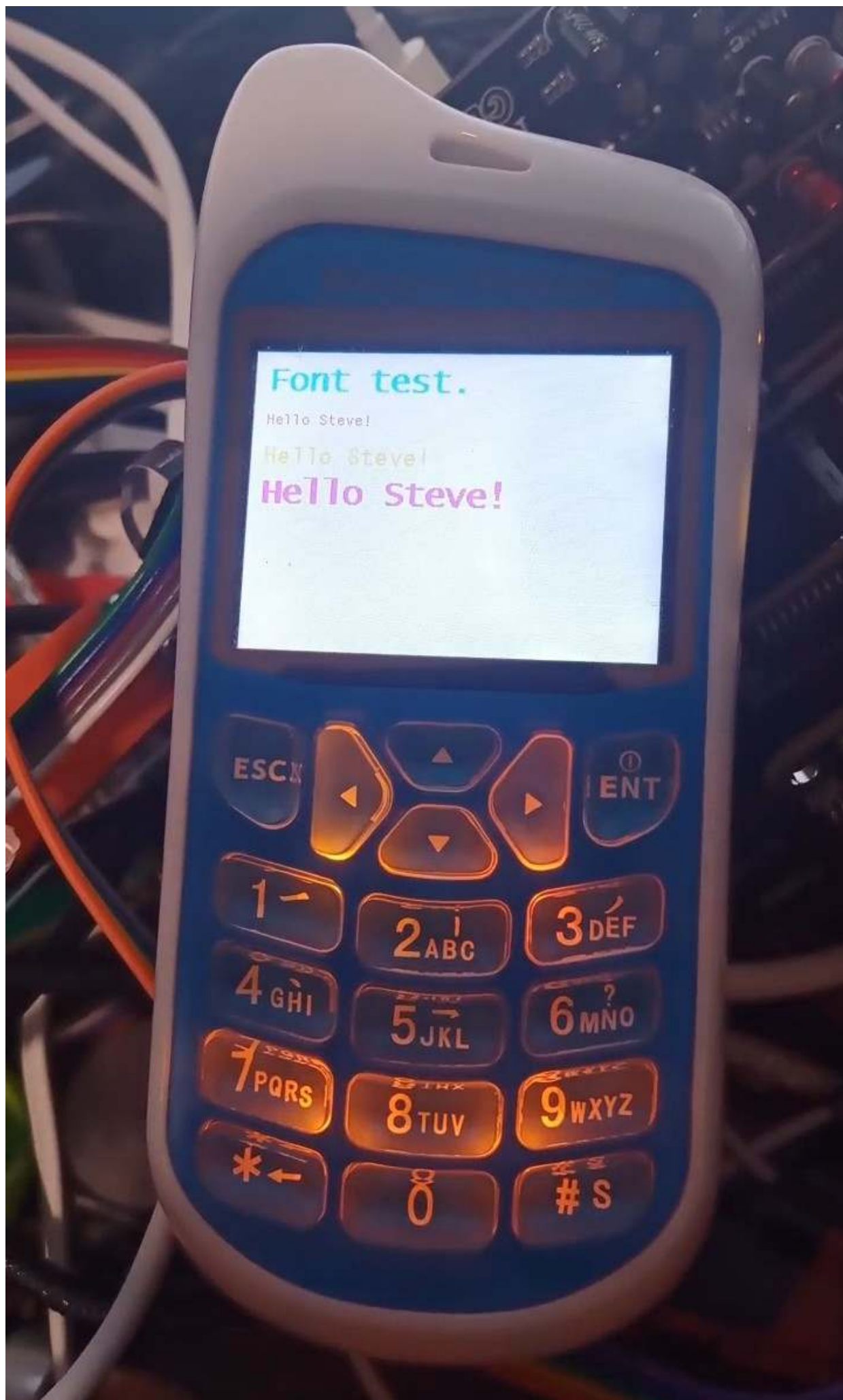


▲ The efficiency of a properly implemented T9 IME unbeatable. At least this is true for Chinese and English.

I continued to search for usable plastic cases. I searched them on Taobao, Alibaba, everywhere. Nokias, Samsungs, and their clones in Shenzhen. I even tried these POS machine ones. Each one has its unavoidable shortcomings. If you were in our Discord server, you can see this part of history, including the funny looking "Wireless Terminal" food ordering machine. I had no luck, until recently.



▲ Different plastic cases tried.



Font test.

Hello Steve!

Hello Steve!

Hello Steve!

ESCK

ENT

1 -

2 ABC

3 DEF

4 GHI

5 JKL

6 MNO

7 PQRS

8 TUV

9 WXYZ

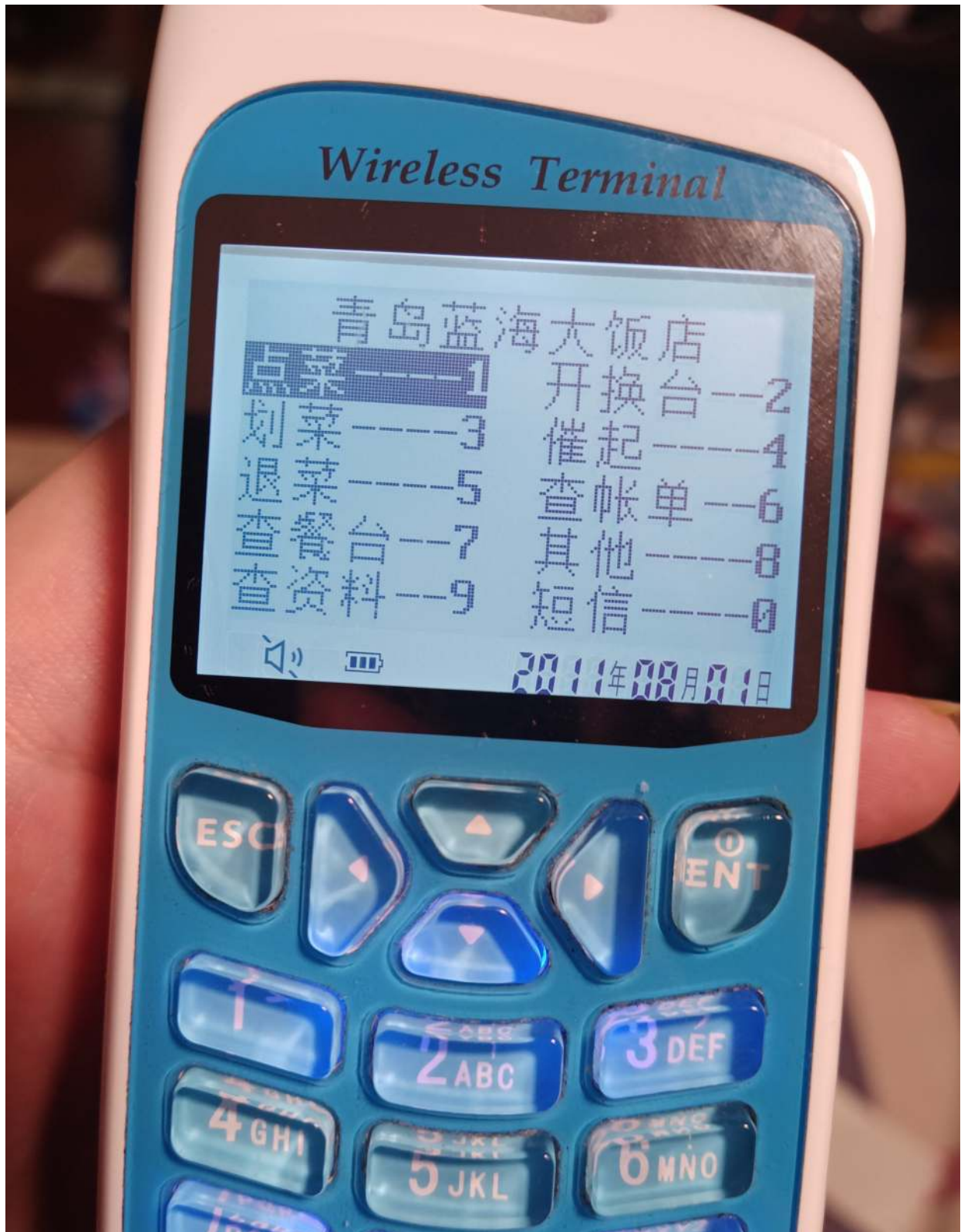
* ←

0

S



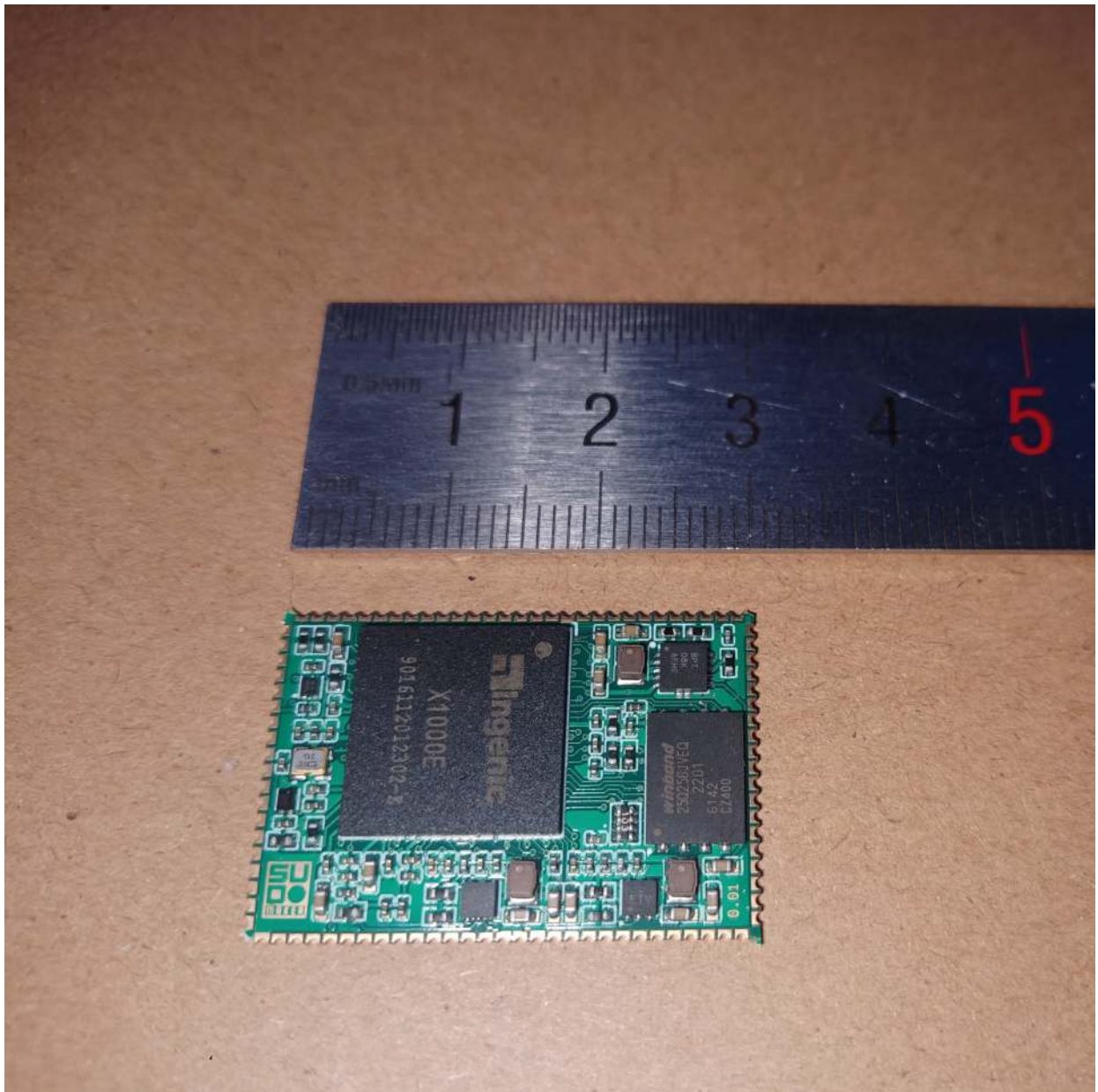
▲ The funny-looking "Wireless Terminal"



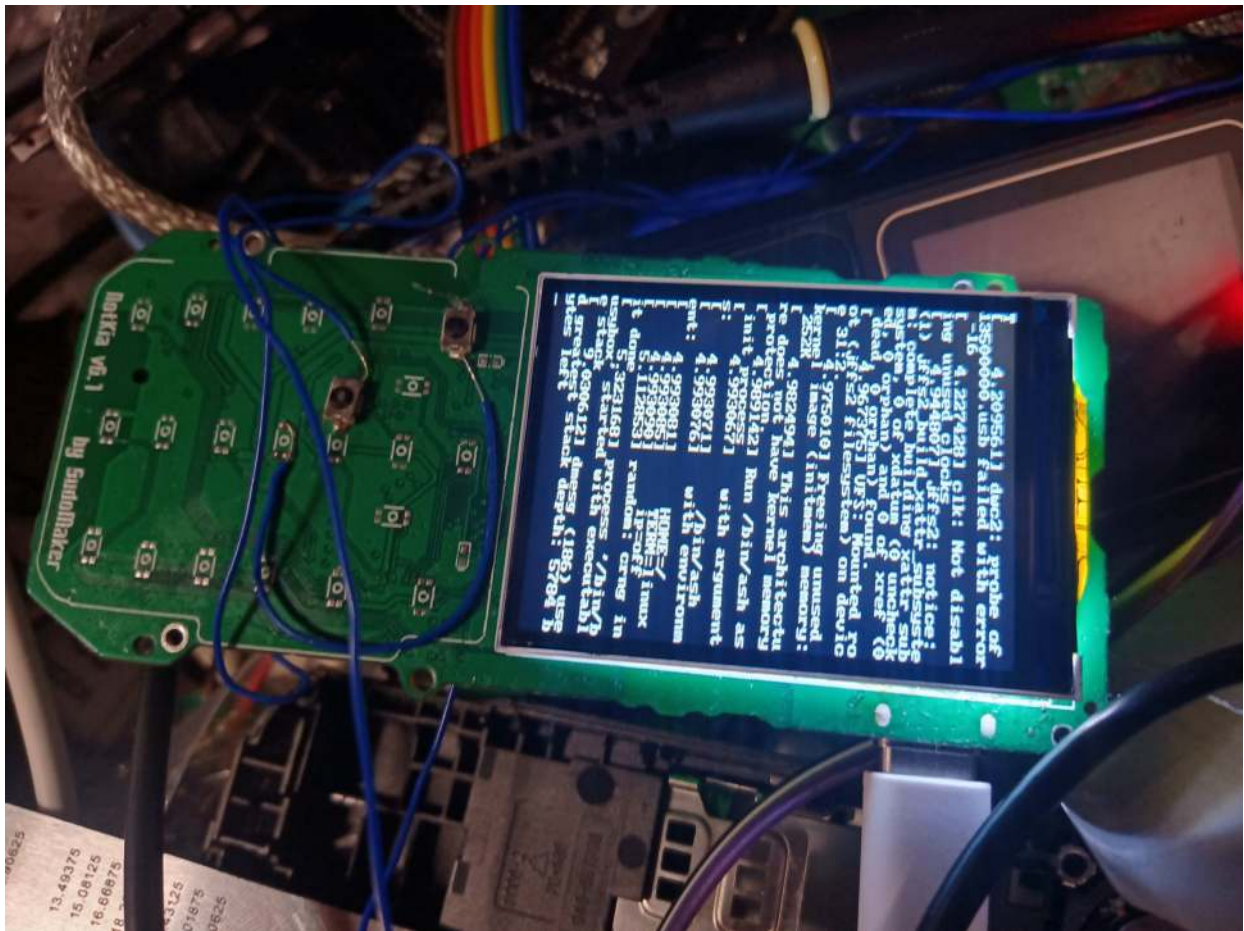
▲ It was used by the waiters in restaurants before iPad ordering became a thing.

Early this year (2022), I found the Nokia 168x series in random recommendations (no they aren't random at all, these apps spy on users all the day) of shopping apps. It looked cool for

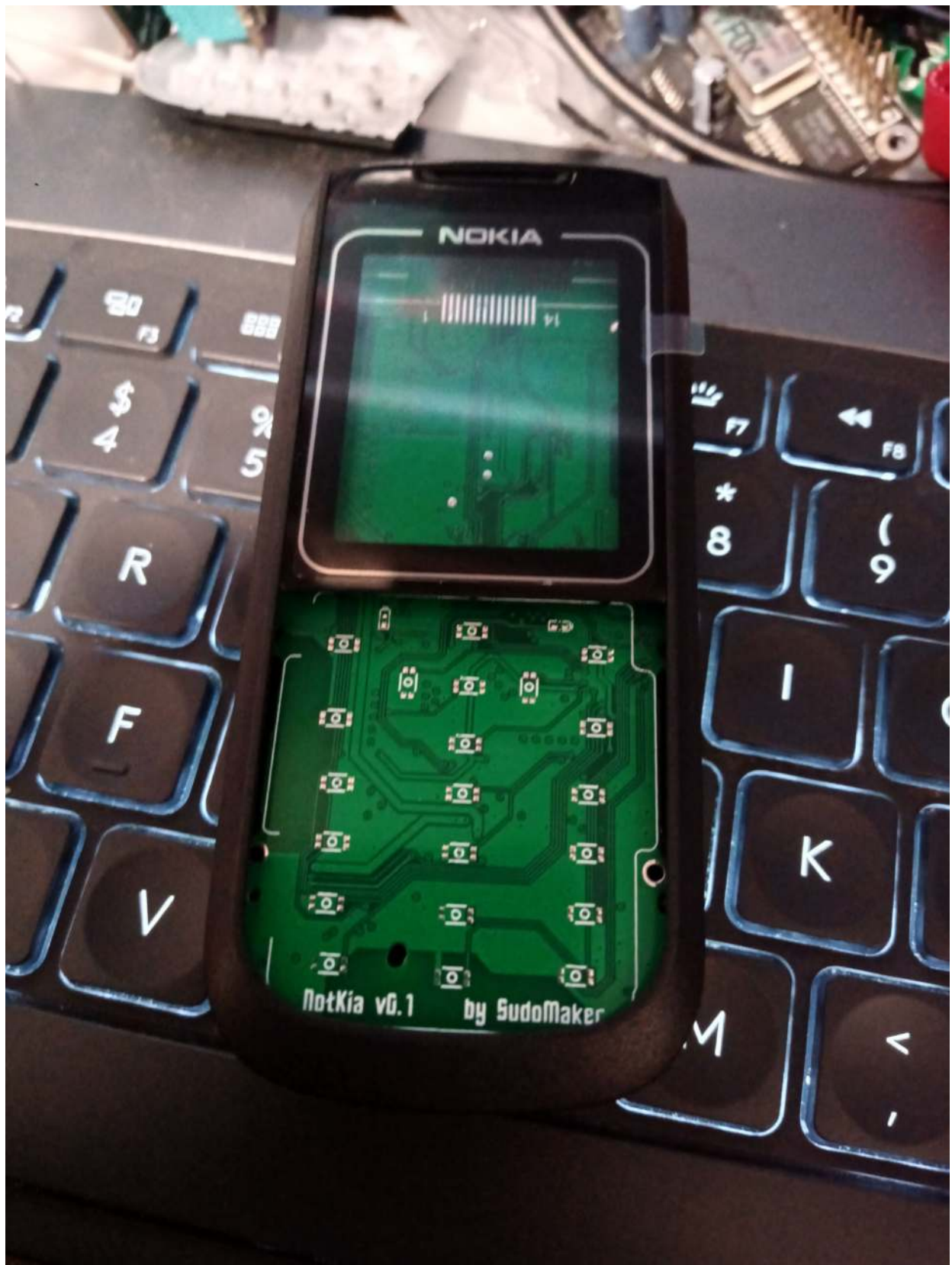
me. Then I searched photos of its PCB on the Internet. Wow, there's so much space in it! I became excited. Then the usual procedure happened: buy, disassemble, picturing, and replicate the PCB shape. In this time I became a little smarter, I made a tiny SoM of the X1000 processor first. And it went production & was tested before anything else. The SoM was a success. Then I started designing the rest. It became a great success.



▲ The tiny Ingenic X1000 SoM designed by me



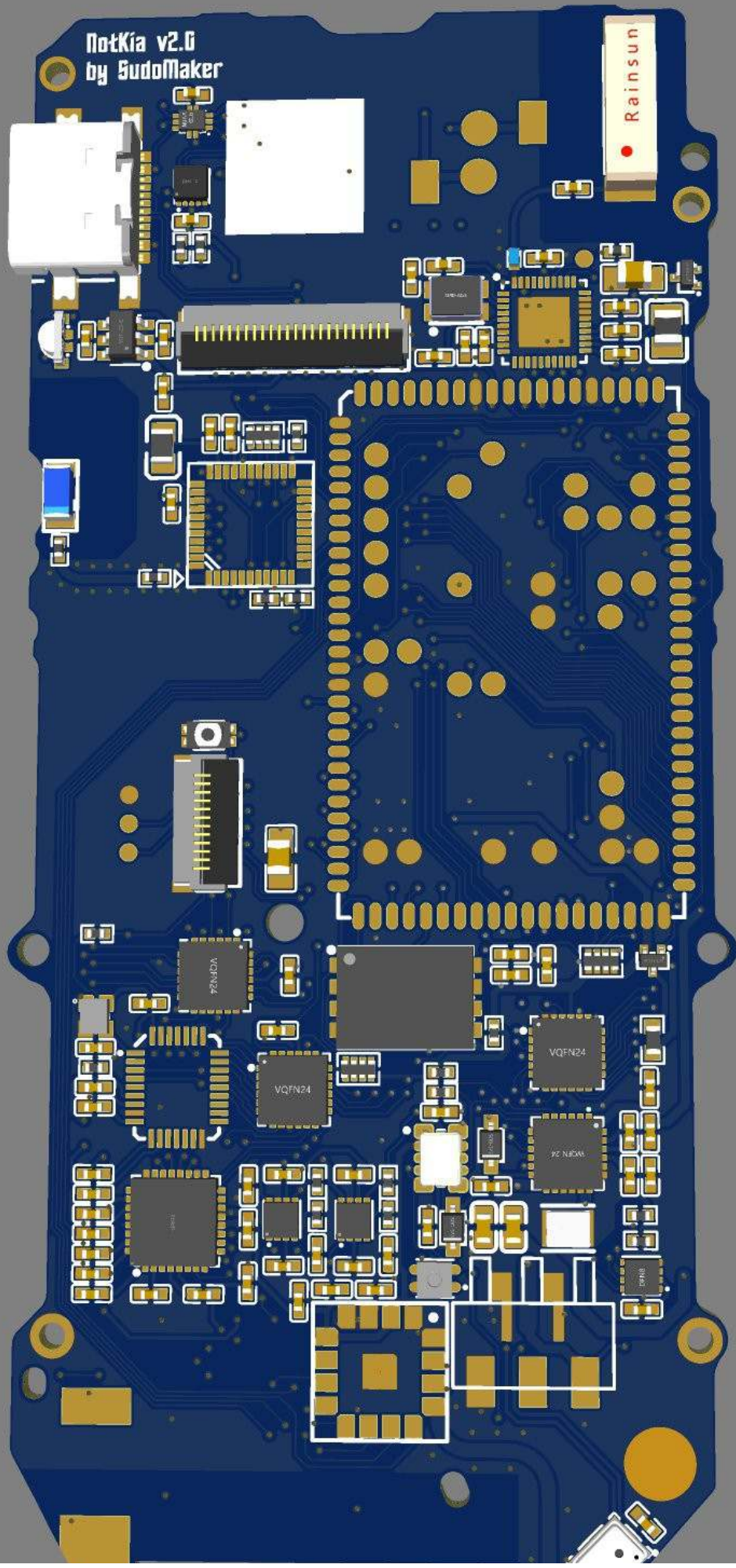
▲ Earlier prototype



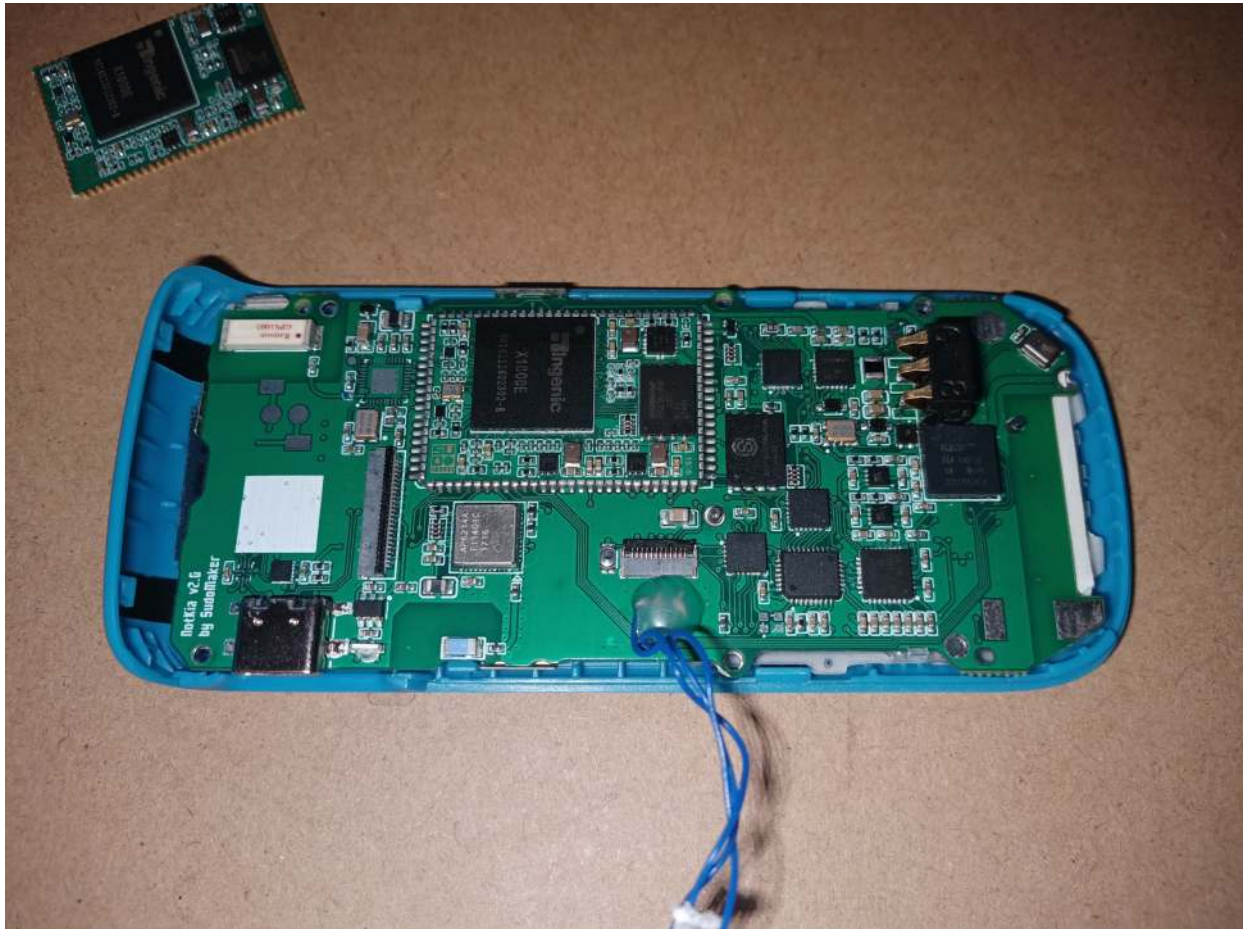
▲ In the 1st version, I forgot a hole near one of the keyboard LEDs :(

NotKia v2.0
by SudoMaker

Rainsun



▲ Version 2.0



▲ Version 2.0 PCB in reality