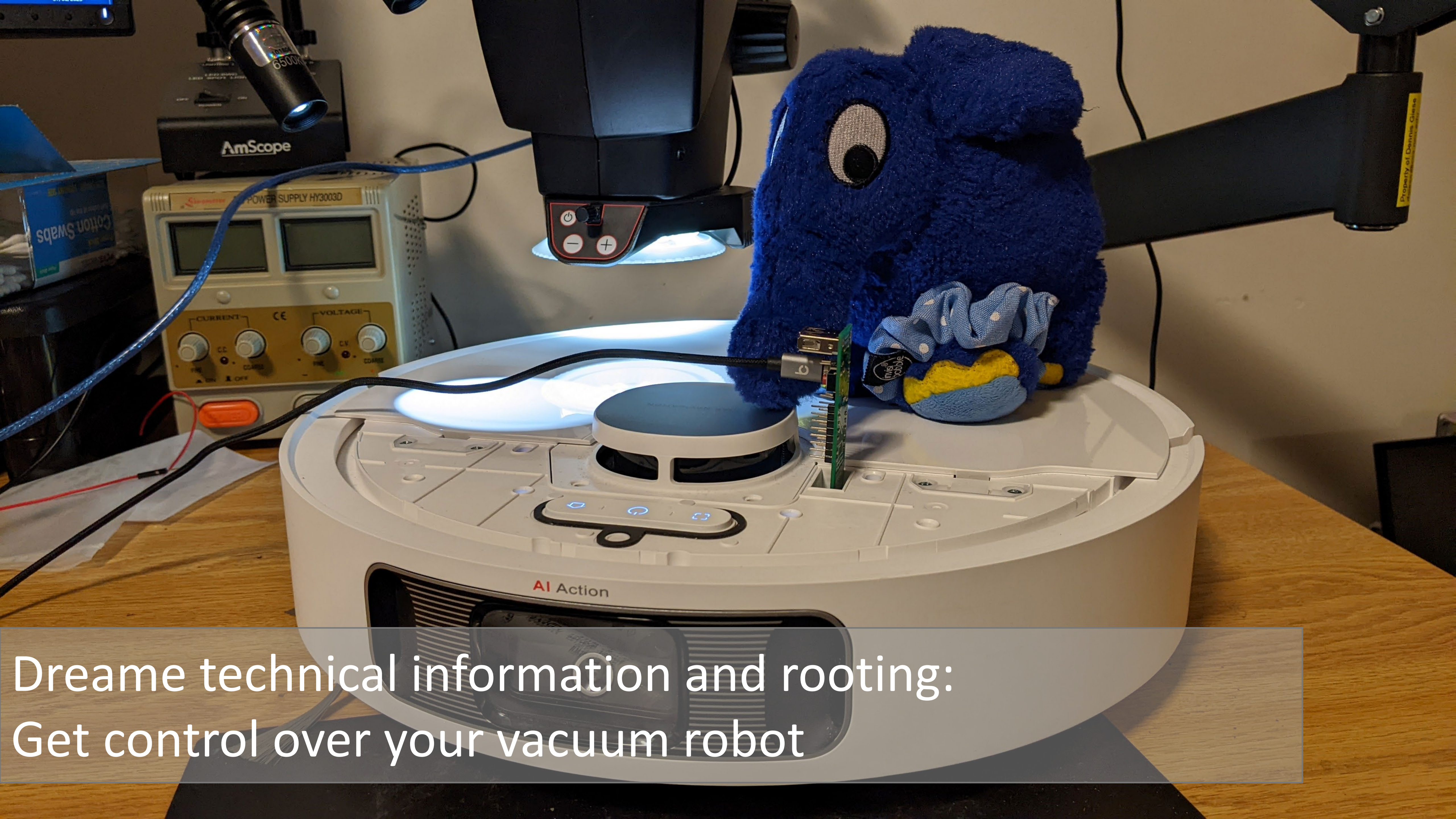


Dreame technical information and rooting: Get control over your vacuum robot

Before you continue: Please look thru all slides and before you start your adventure

You might want to join the Telegram channel:

https://t.me/dust_announce

Good news: No soldering or teardown required

Why get root access?

- Use Valetudo (https://valetudo.cloud/)
  - Replace the cloud functionality with an open-source software
  - Integrate the device into your home automation
- Install your own soundfiles/voices

Currently tested devices (06.2024)

- Dreame L10s Ultra (r2228)
- Dreame D10s Pro (r2250)
- Dreame D10s Plus (r2240)
- Dreame L10s Pro Heat (2338)
- Dreame X40 (r2416)

Risks

- A failed flash can leave the device in an undefined state
- Requires reflashing
- Problem:
  - All partitions need to be recreated
  - Device identity (Device ID, Cloud keys, MAC) gets lost
  - Calibration data lost
- Observations:
  - Device identity does not matter if you use Valetudo
  - Unknown consequences for lost calibration data

More risks

- Problem: Hardware differs even for the same „model“
- The root method cannot verify your exact model
- Flashing an incorrect firmware will perma-damage your device
  – Other, not obvious side effects might occur
  – Recovery might be tricky and can cause problems
- Important: Make sure that you have the correct firmware
  – Do not try to use the same generated firmware on multiple devices
  – If you are unsure, ask us first!

Sidenotes

- Rooting will permanently change files on your device*
  - Cloud connection / App usage should be still possible
- By installing a custom firmware, you cannot update your device with official firmwares anymore **
  - That is the price of rooting
  - You can update your device with custom firmwares
- Connect to your robot and write down the MAC address

* we needed to reconstruct parts from the firmware by extracting contents from flash and guessing unknown parts.

** if you backup the encryption keys, you can restore them at a future time. In special circumstances (dead hardware) we might provide you with stock firmware.

Important information

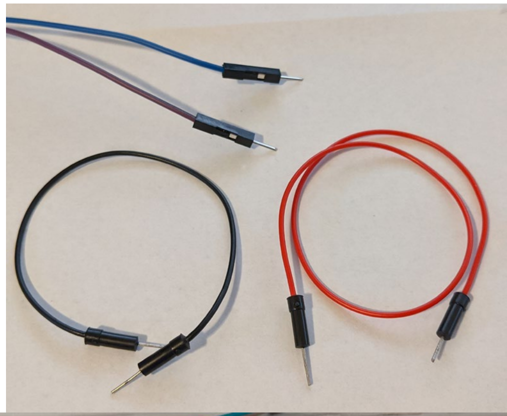
- DO NOT UPDATE THE FIRMWARE VIA THE APP if you ever plan to root your device
  - New firmwares have rooting methods patched
  - Root might be still possible, but is more complicated and risky
- Before rooting: test the device without using the app
  - Fully charge your device
  - Use the buttons to start the cleaning run
  - Let the robot clean multiple times
  - Background: Some devices come with defects from the factory. It is tricky to figure out if your robot behaves weirdly due to it being already broken or due to rooting

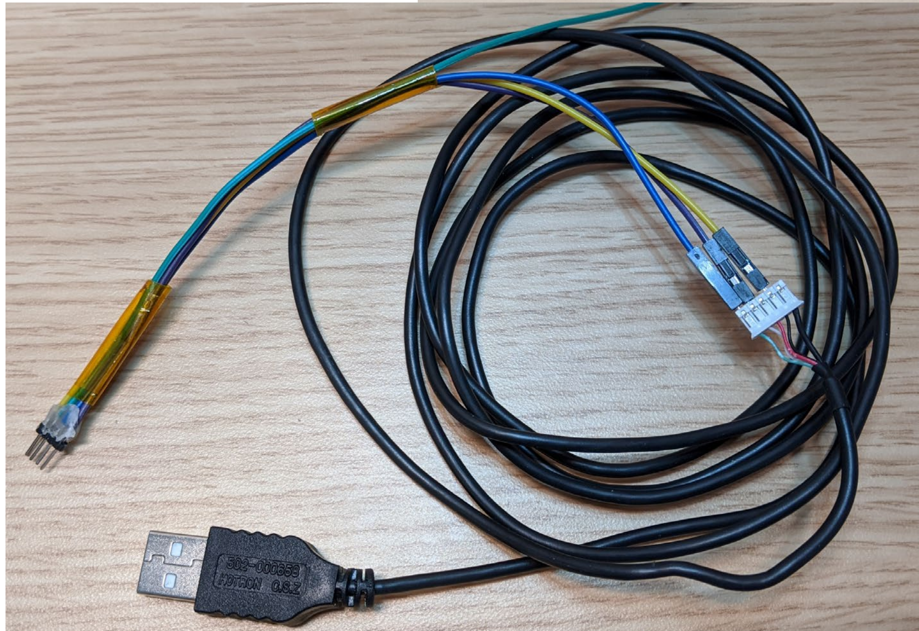
Current available rooting methods

- Livesuit
  - Requires: USB connection, UART (recommended but not required)
  - Idea: creation of special factory image, triggering bootrom mode (FEL), flashing of all partitions via Livesuit software (requires Windows)

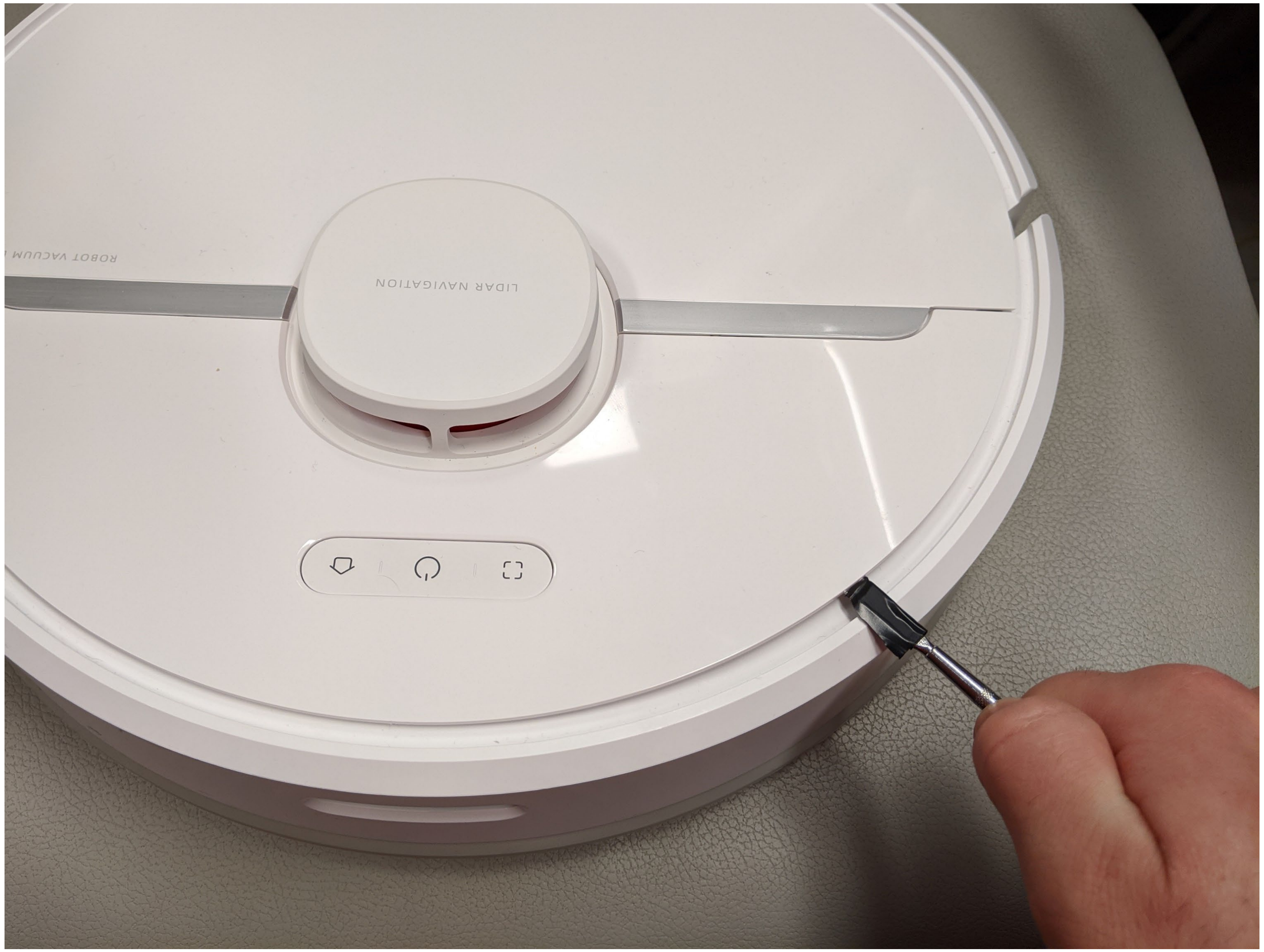
Tools required for root

- Breadboard Jumper Wires
- 2mm pitch headers
- USB cable (for Livesuit/FEL)
  - (e.g. from a broken USB mouse)
- Alternative: custom PCBs

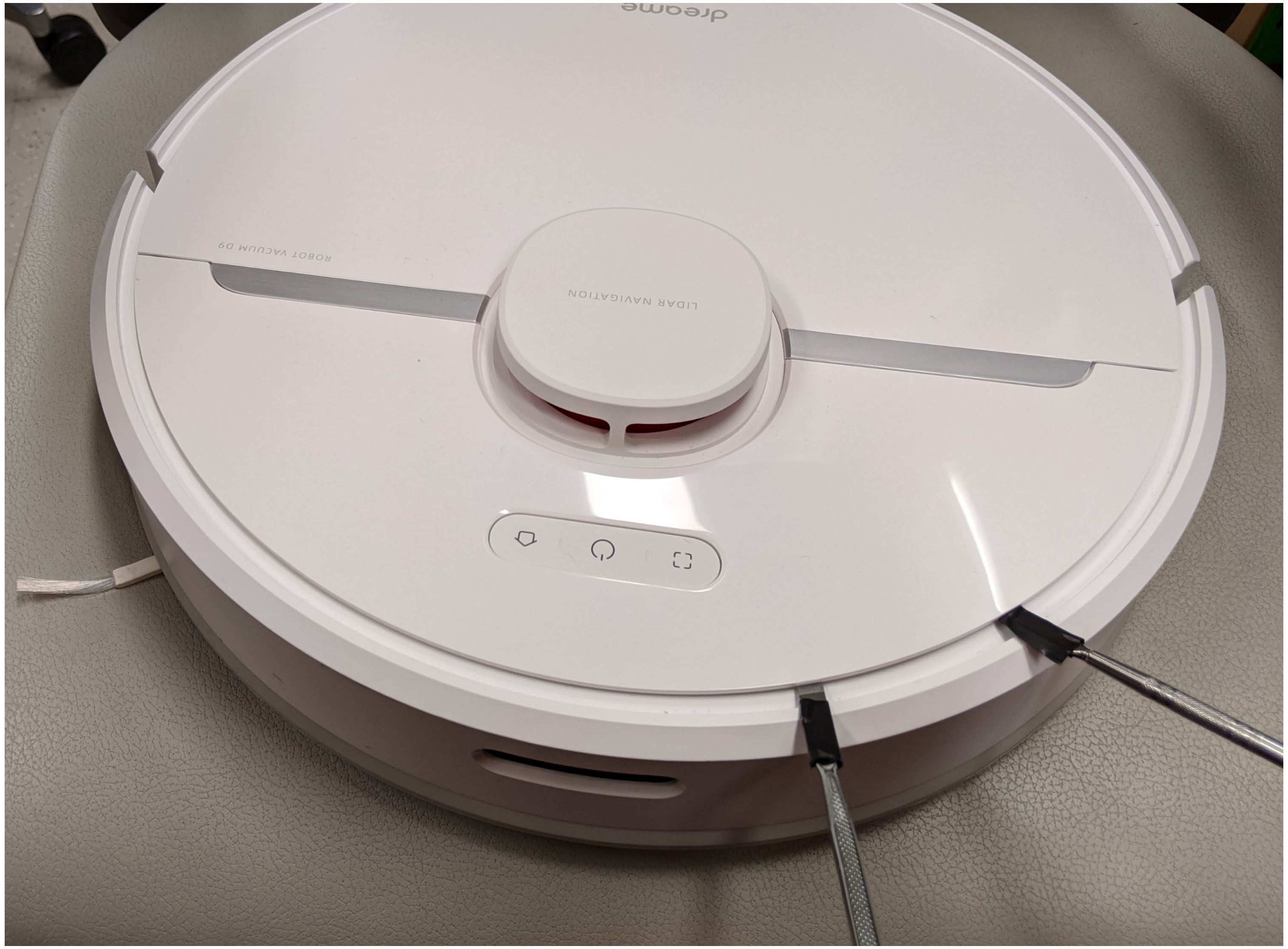




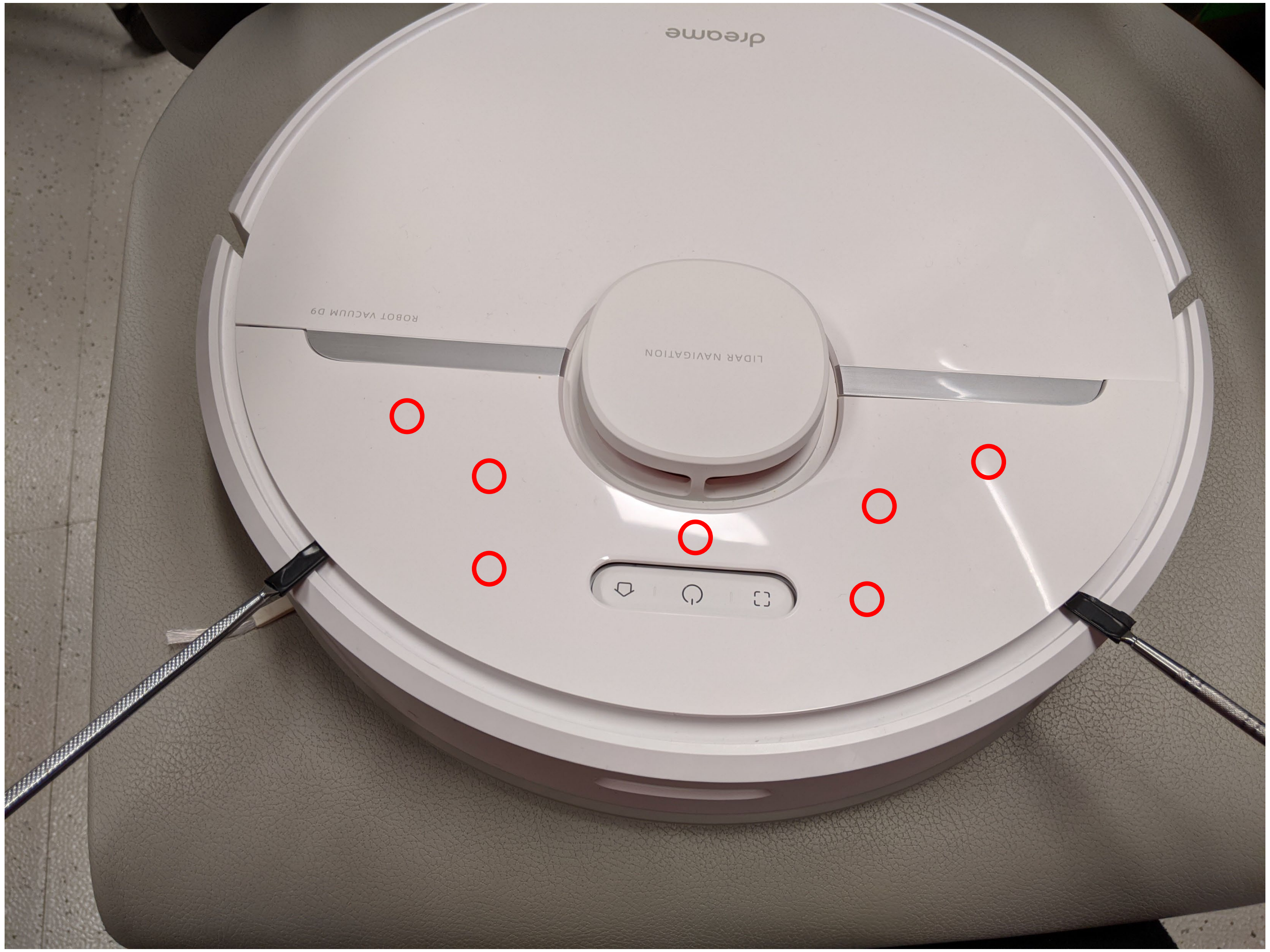
Opening the device



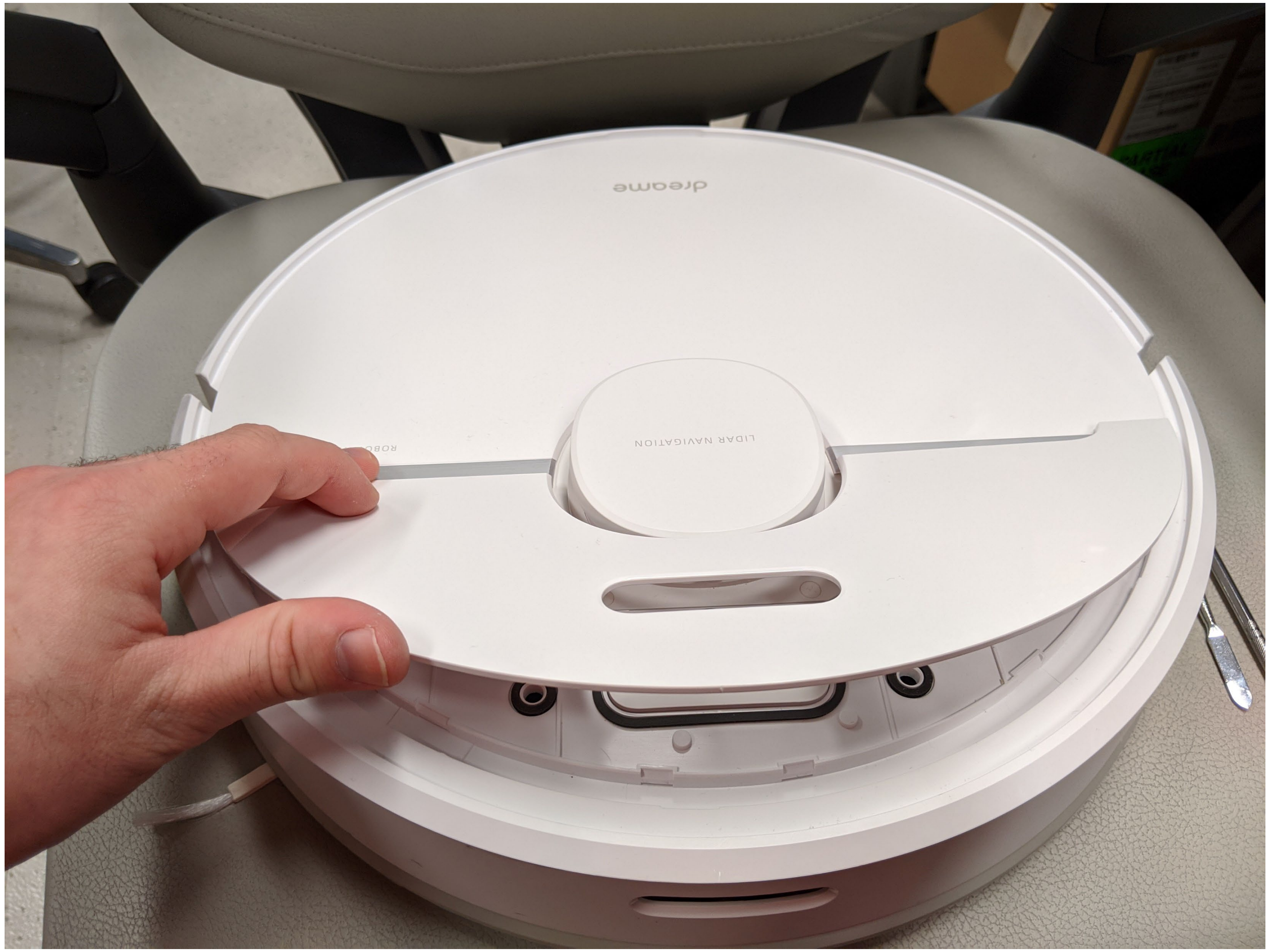
Opening the device



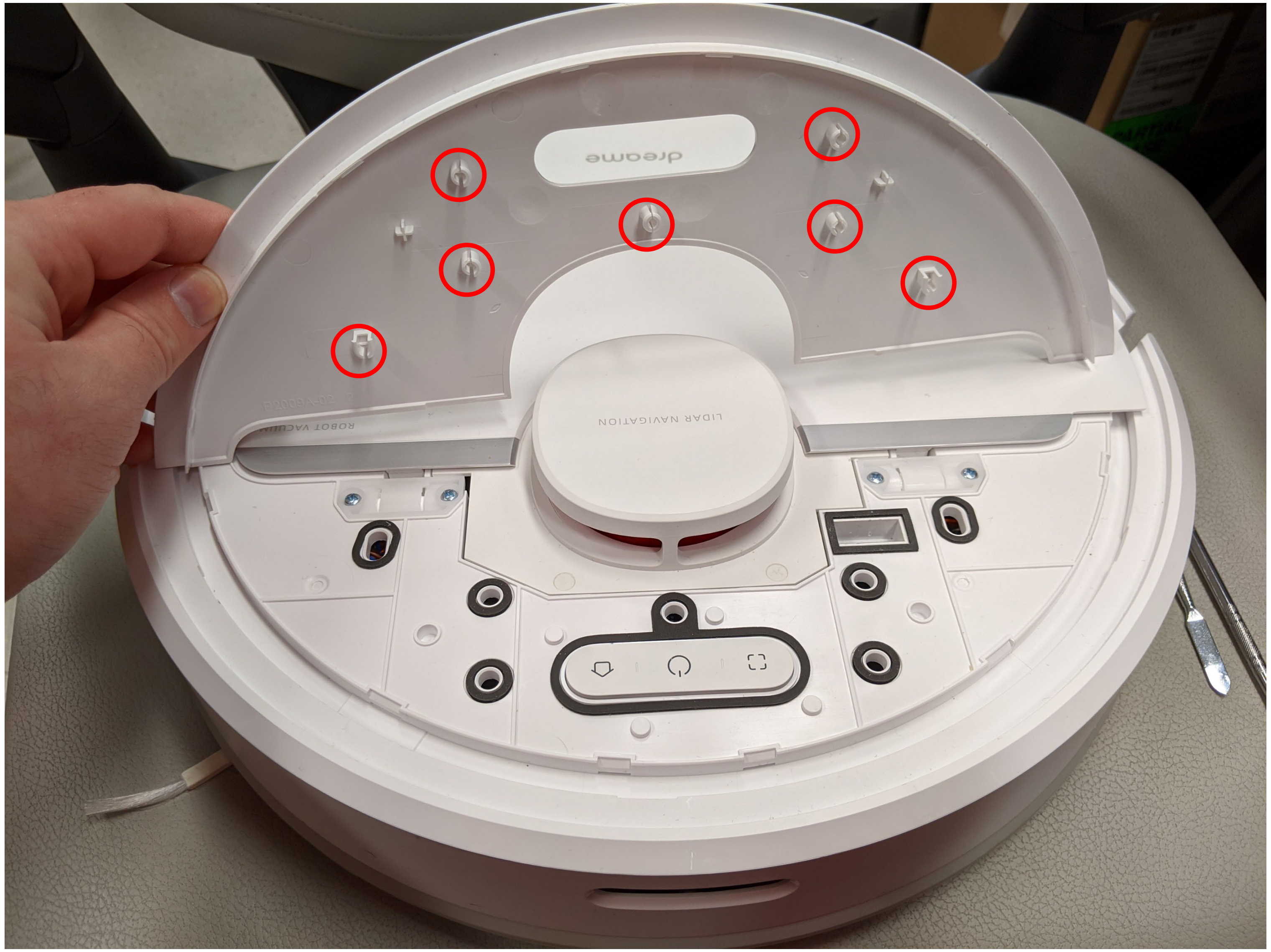
Opening the device



Opening the device



Opening the device



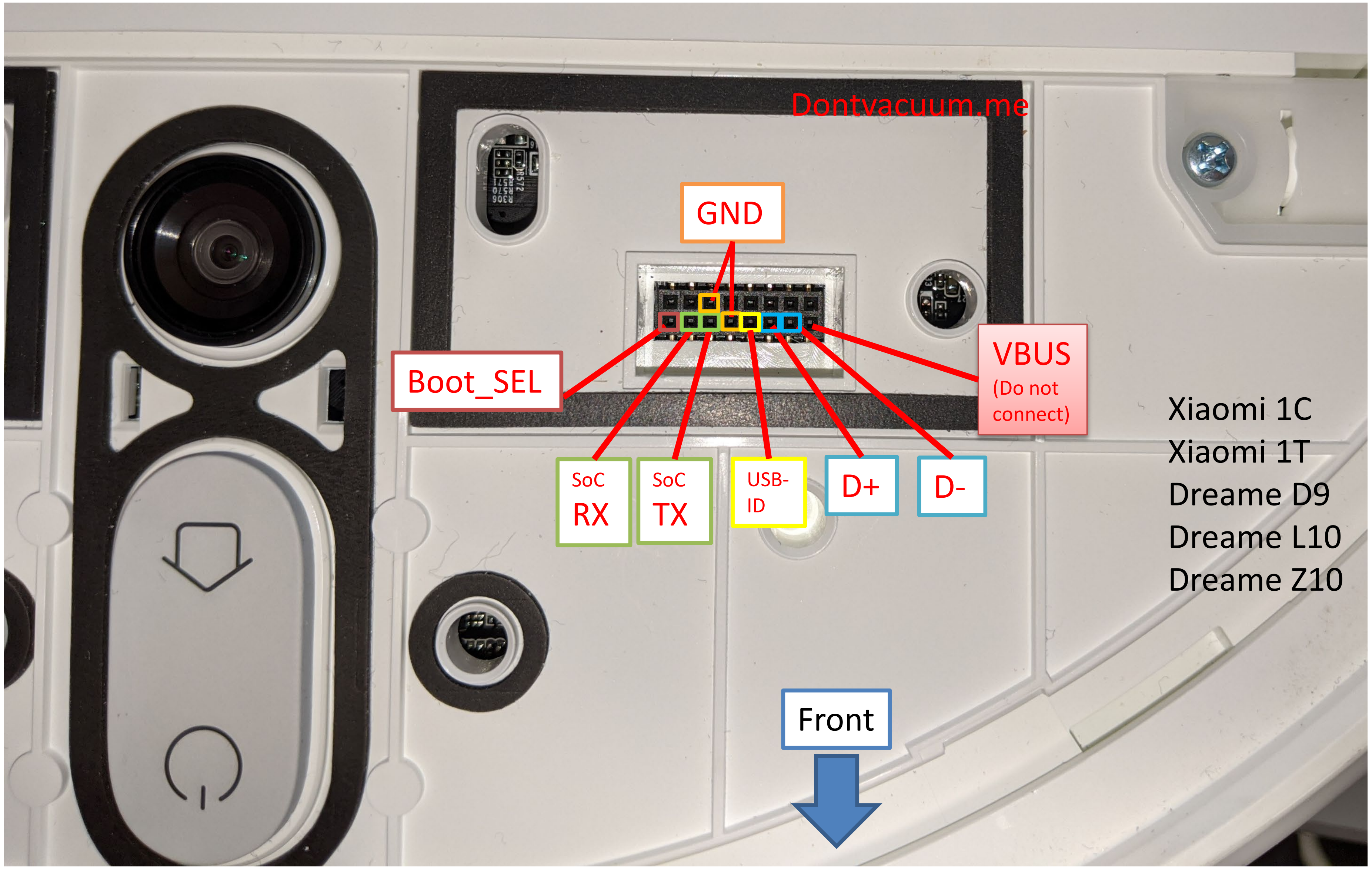
Debug pinout

- Debug interface
  - 2x8 pins
  - 2mm pitch size

Warning:
2mm pitch size is way smaller than the usual 2.54 mm

Warning:
Make sure you connect to the correct pins!

Warning:
Double-check the orientation
Of the pinout!



Xiaomi 1C
Xiaomi 1T
Dreame D9
Dreame L10
Dreame Z10

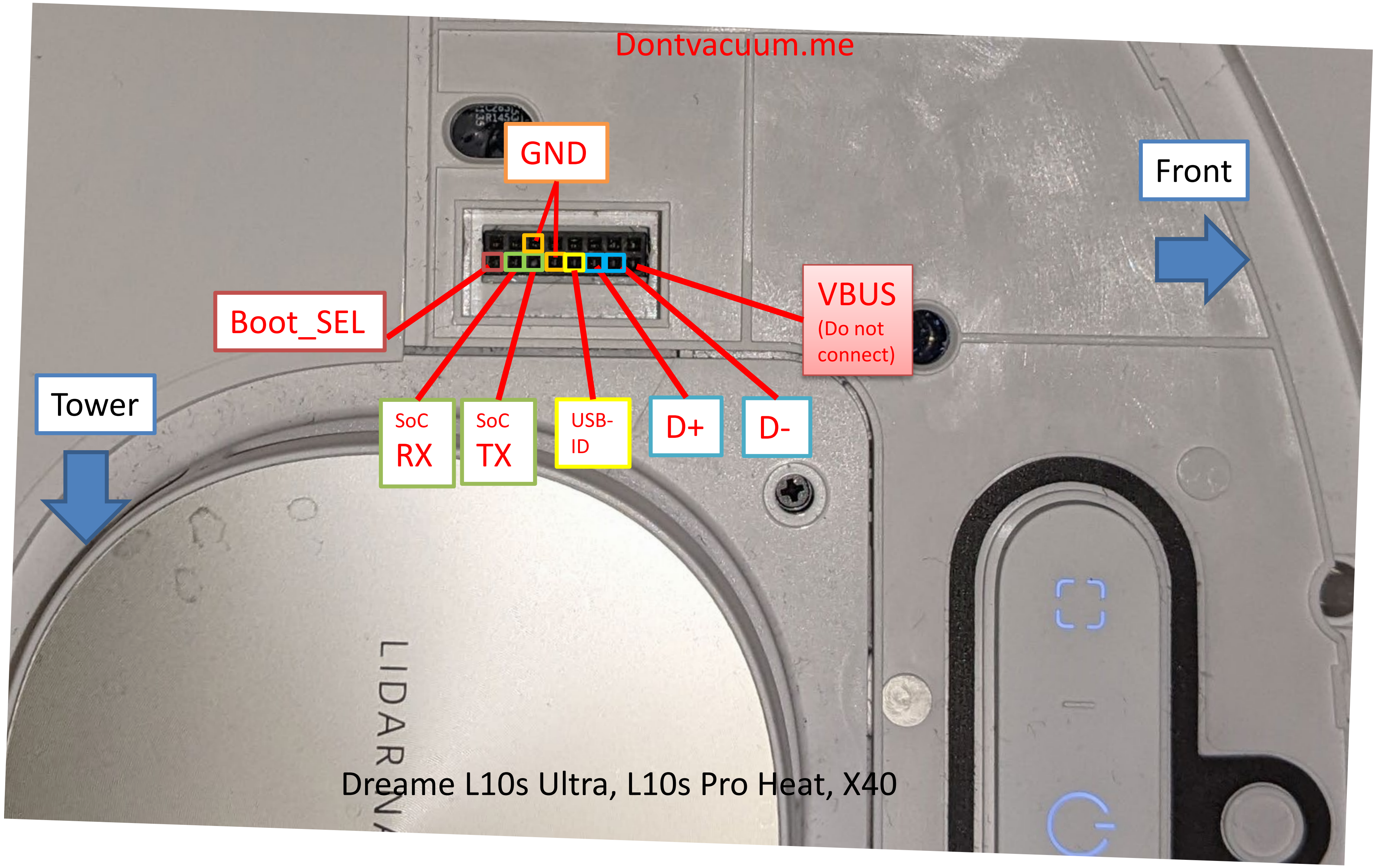
Debug pinout

- Debug interface
  - 2x8 pins
  - 2mm pitch size

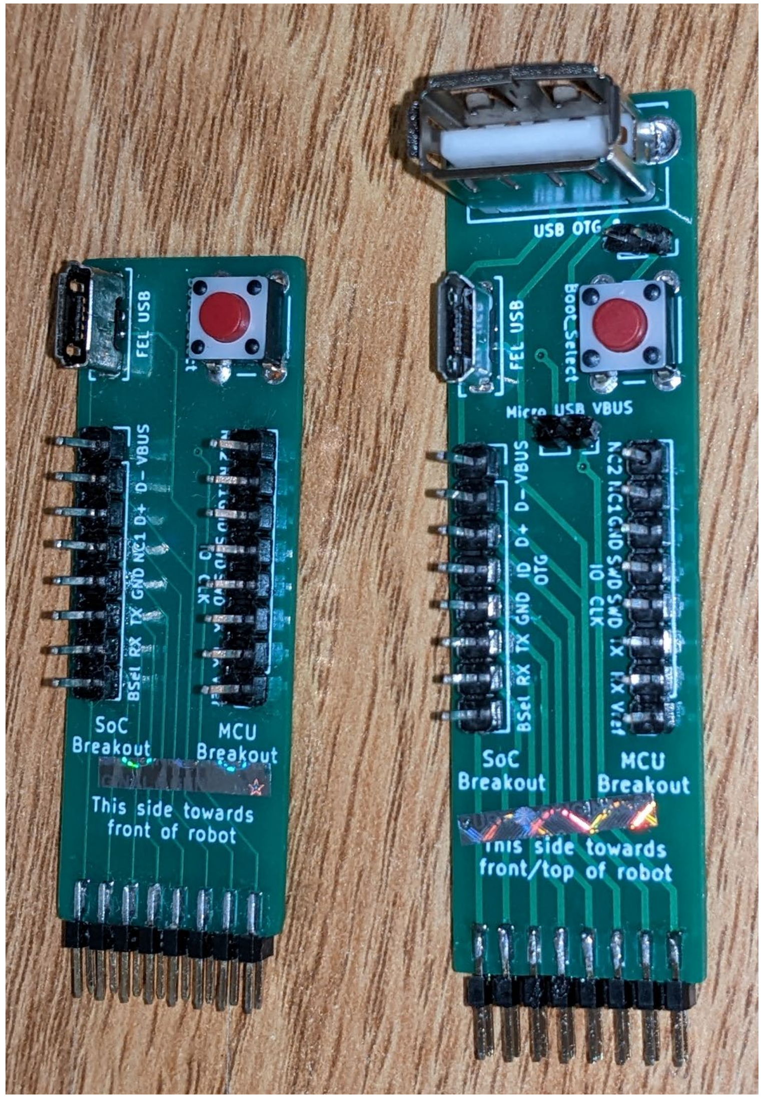
Warning:
2mm pitch size is way smaller than the usual 2.54 mm

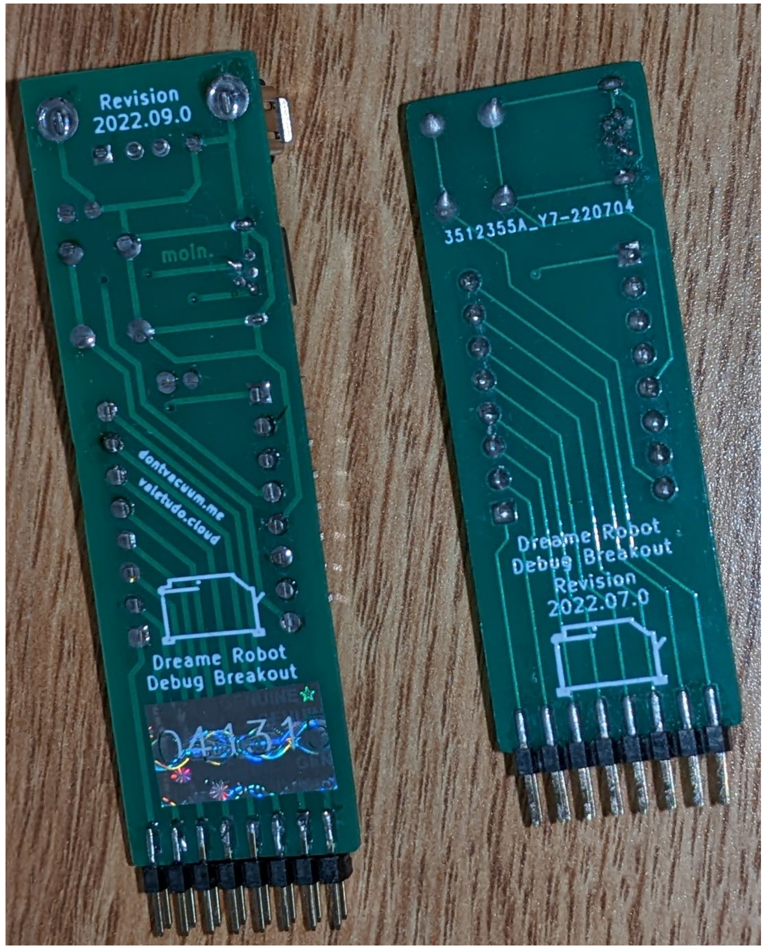
Warning:
Make sure you connect to the correct pins!

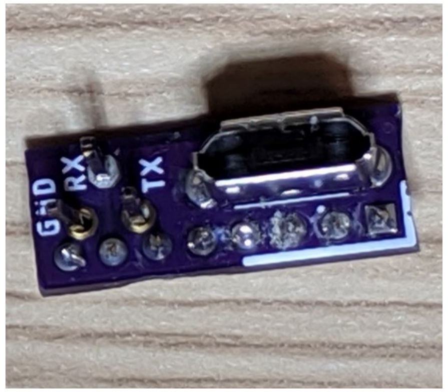
Warning:
Double-check the orientation
Of the pinout!

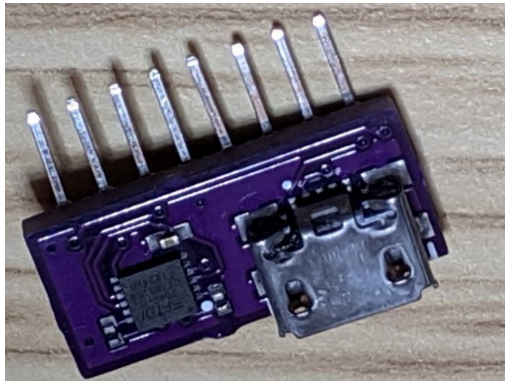


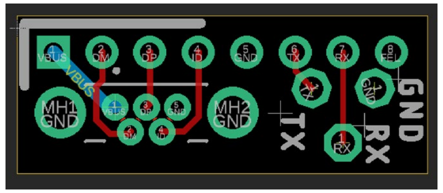
Rooting with custom PCBs

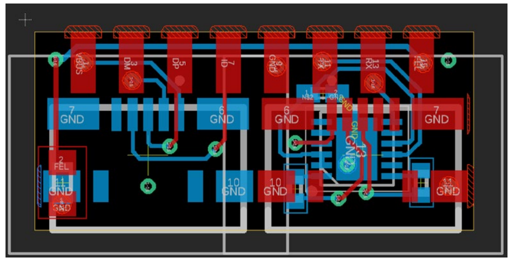

USB OTG
FEL USB
Boot_Select
Micro USB VBUS
SoC Breakout
MCU Breakout
This side towards front of robot
This side towards front/top of robot


Revision 2022.09.0
dontvacuum.me
valetudo.cloud
Dreame Robot Debug Breakout
3512355A_Y7-220704
Dreame Robot Debug Breakout Revision 2022.07.0


USB + UART headers (outdated)

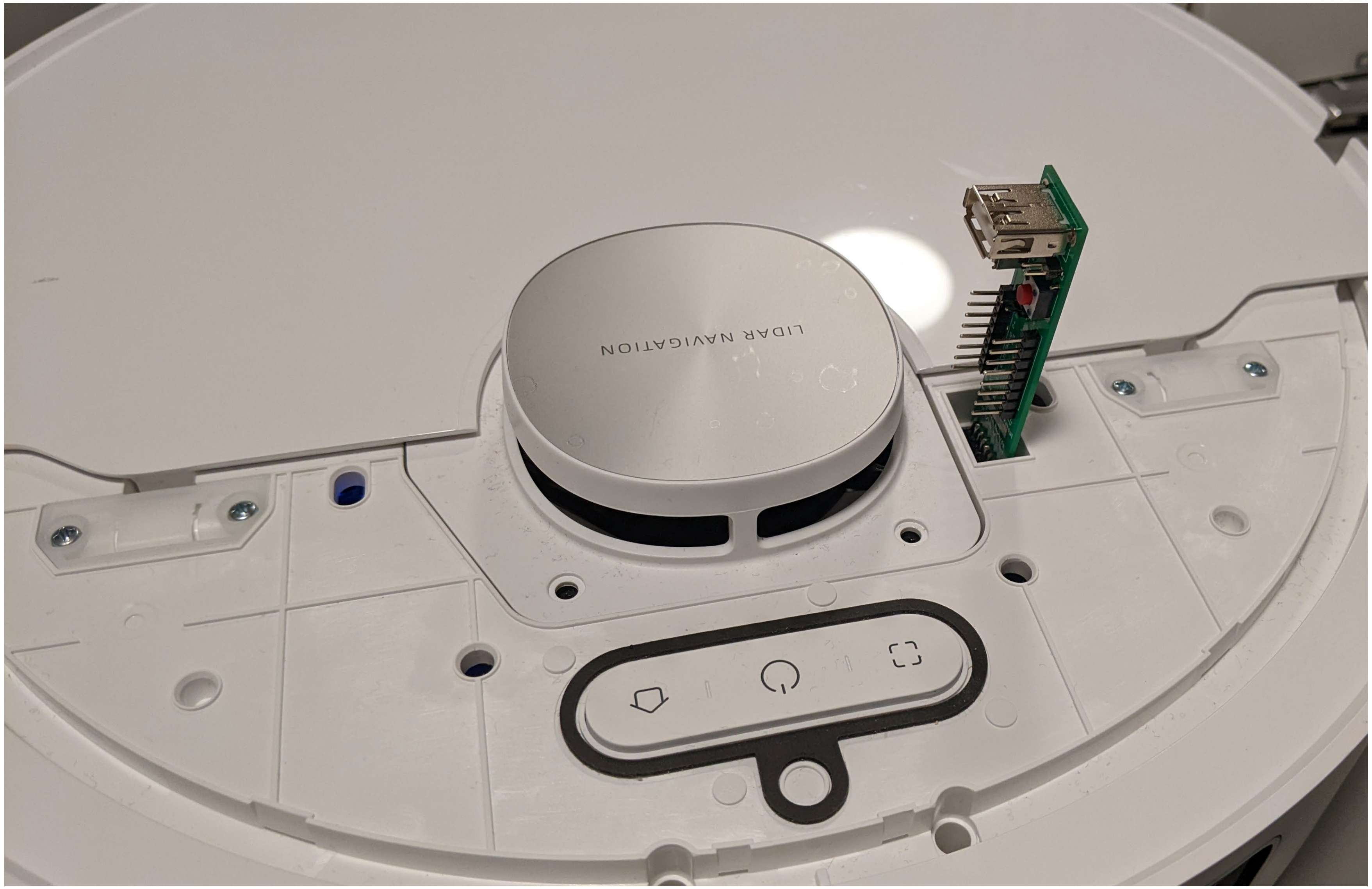

USB + integrated UART Adapter (outdated)

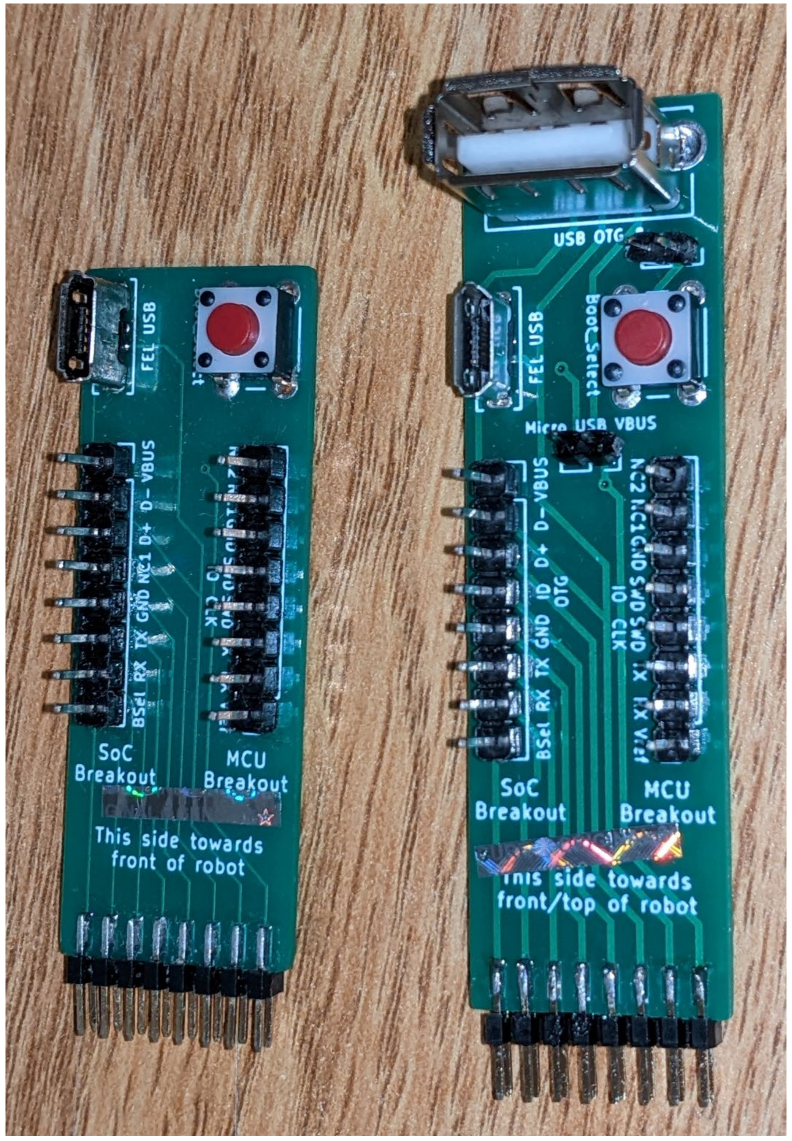




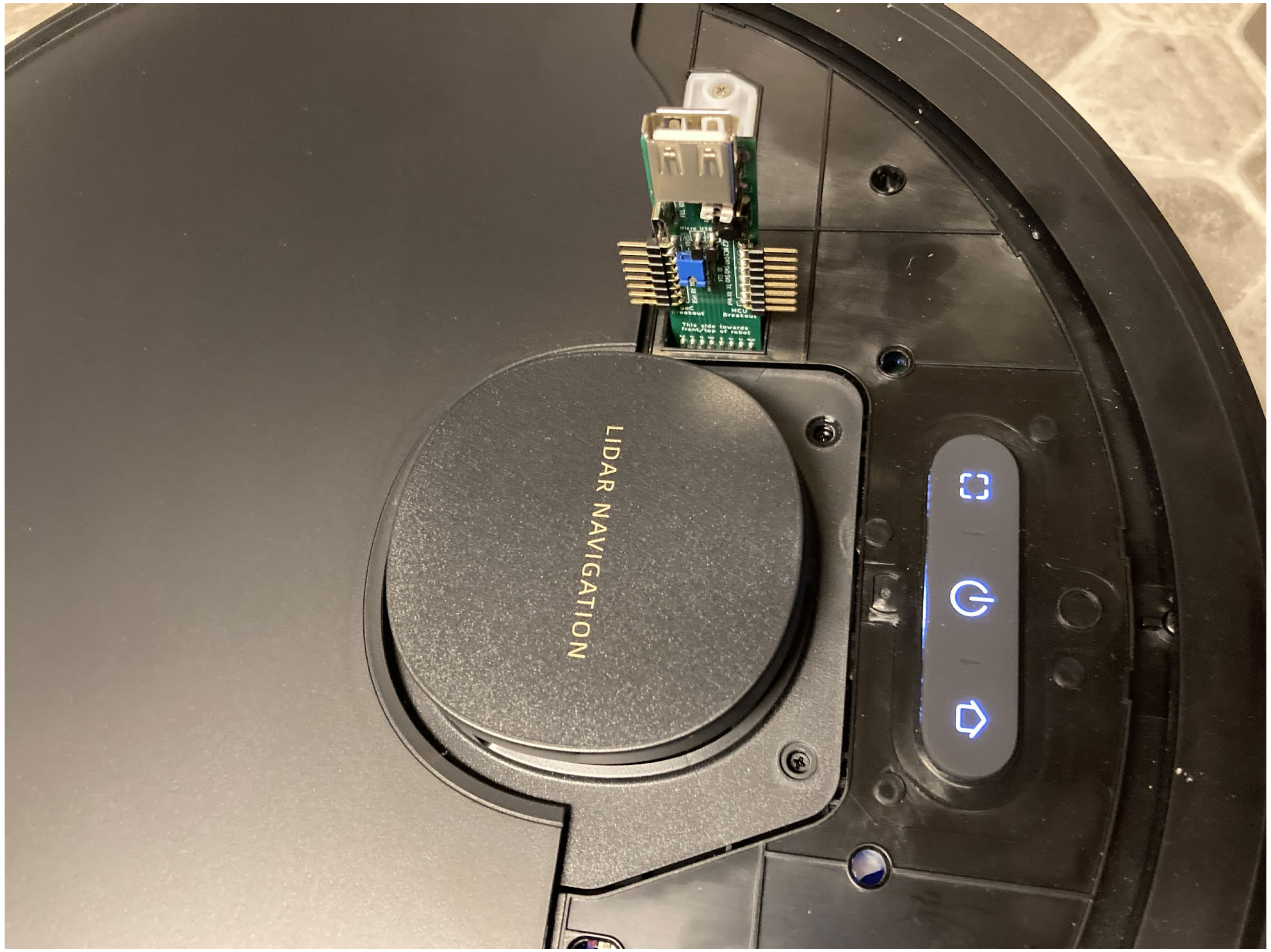
Check https://builder.dontvacuum.me/dreameadapter or https://Valetudo.cloud for the Gerber files

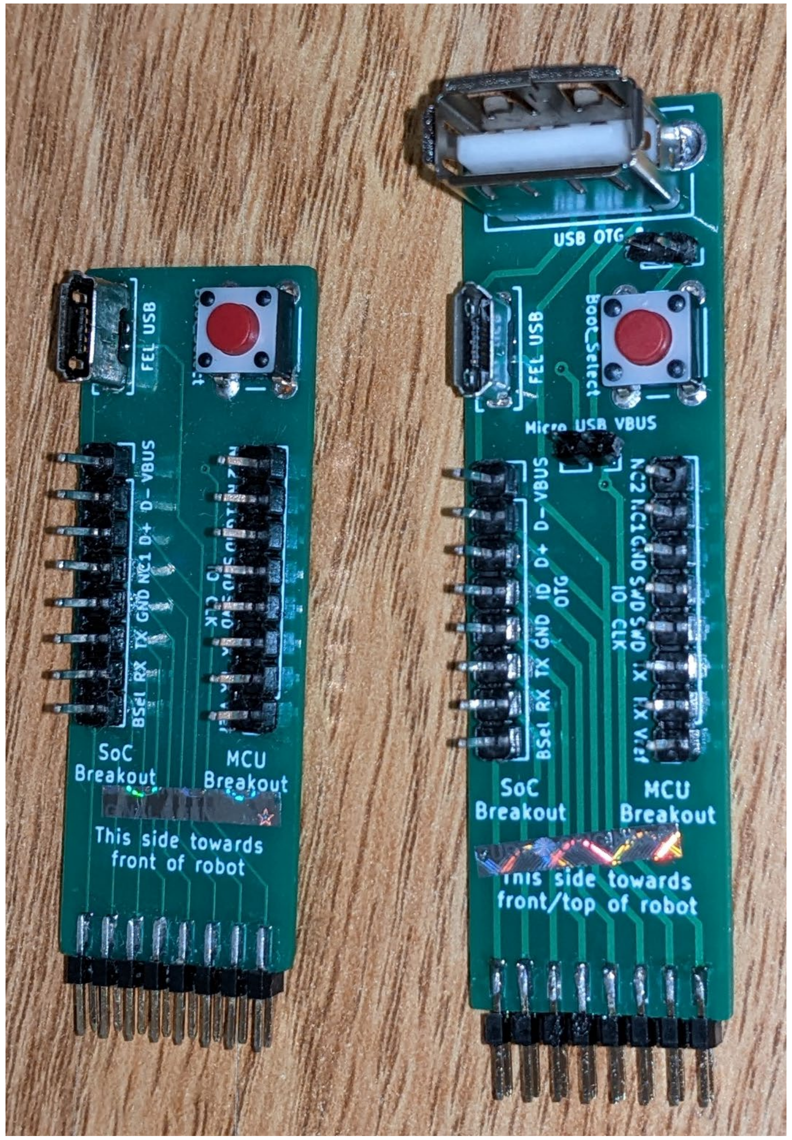
Usage of custom PCB



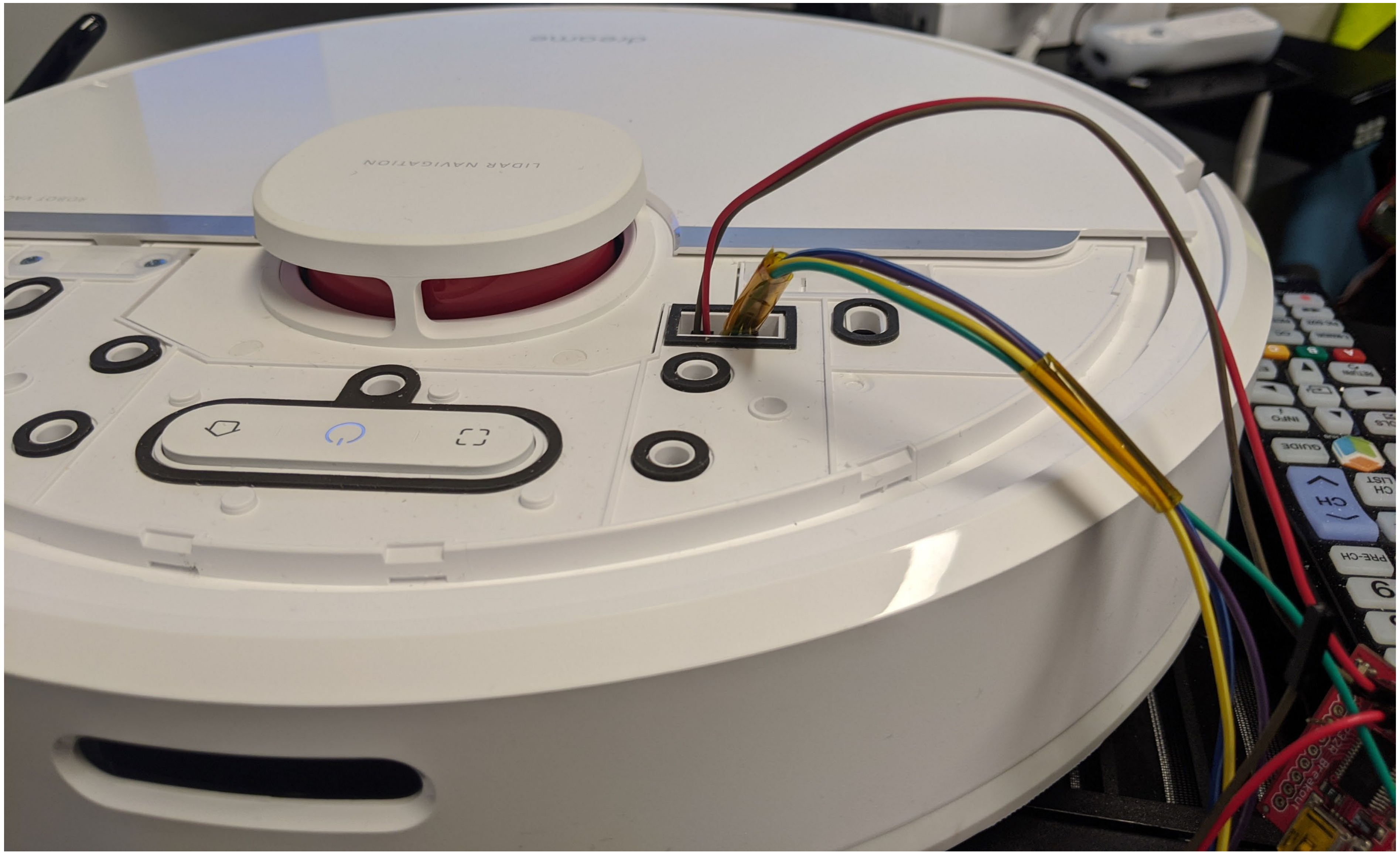

FEL USB
BSel RX TX GND NC1 D+ D- VBUS
SoC Breakout
MCU Breakout
This side towards front of robot
USB OTG
Boot_Select
Micro USB VBUS
BSel RX TX GND ID D+ D- VBUS
OTG
NC2 NC1 GND SWD SWD TX RX Vref
IO CLK
This side towards front/top of robot

Usage of custom PCB




FEL USB
SoC Breakout
MCU Breakout
This side towards front of robot
USB OTG
Boot_Select
Micro USB VBUS
BSel RX TX GND ID D+ D- VBUS
This side towards front/top of robot

Connecting jumper wires (2mm pitch)



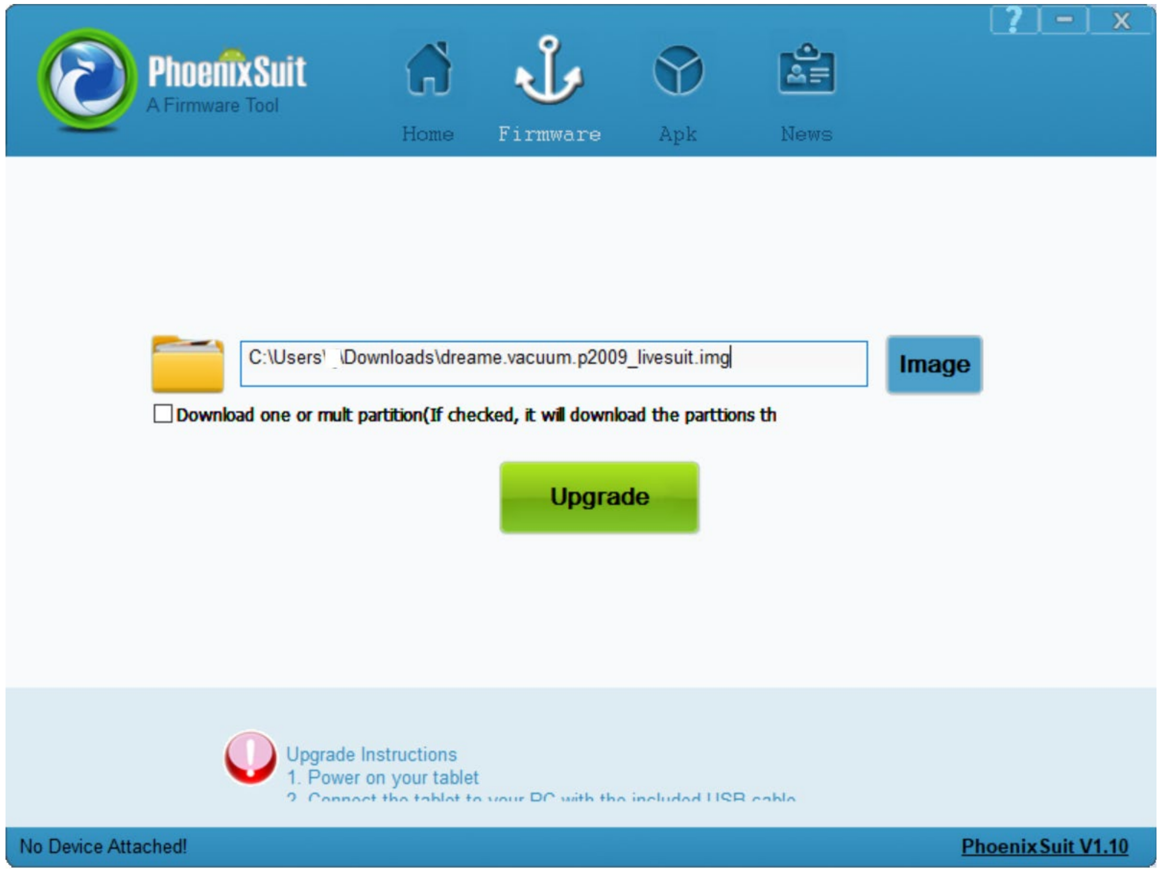
ROOTING

Root Preparations

- Requirements:
  - Windows (can be a VM)
  - (you can also use Linux, but it requires Kernel modules installation)
  - USB port
- Preparation:
  - Download and unpack Phoenixsuit from here: https://androidmtk.com/download-phoenixsuit
    - Alternative for Linux: https://github.com/linux-sunxi/sunxi-livesuite
    - Livesuit for Linux behaves the same as Phoenixsuit
  - Download and unpack ADB and Fastboot from here: https://developer.android.com/tools/releases/platform-tools
  - Download and unpack drivers from here: https://builder.dontvacuum.me/nextgen/usbdriver.zip
  - Download image from: https://builder.dontvacuum.me/nextgen/

Root Step 1

- Make sure that the robot is not connected over USB
- Open PhoenixSuit.exe and select image “dust-livesuit-mr813-ddrX.img”



C:\Users\ \Downloads\dreame.vacuum.p2009_livesuit.img

☐ Download one or mult partition(If checked, it will download the parttions th

Upgrade Instructions
1. Power on your tablet

No Device Attached!

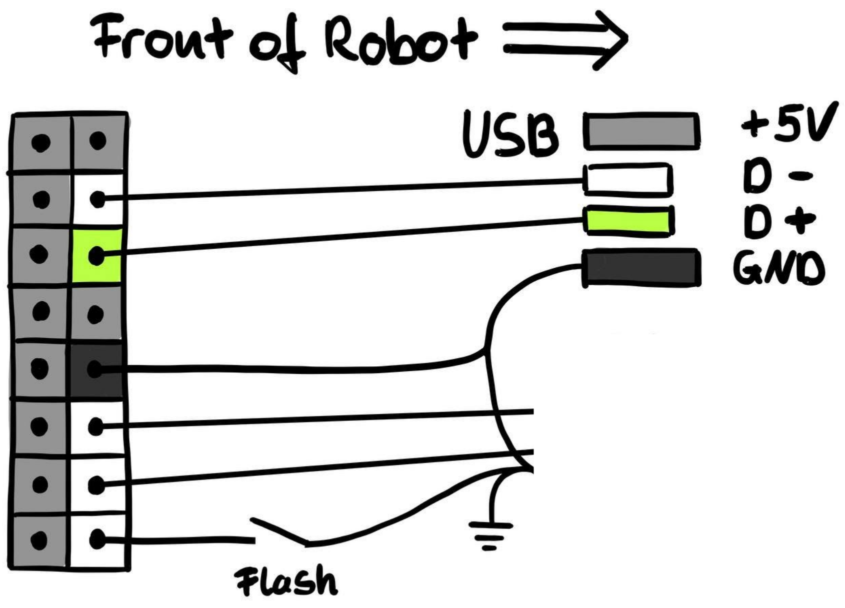
PhoenixSuit V1.10

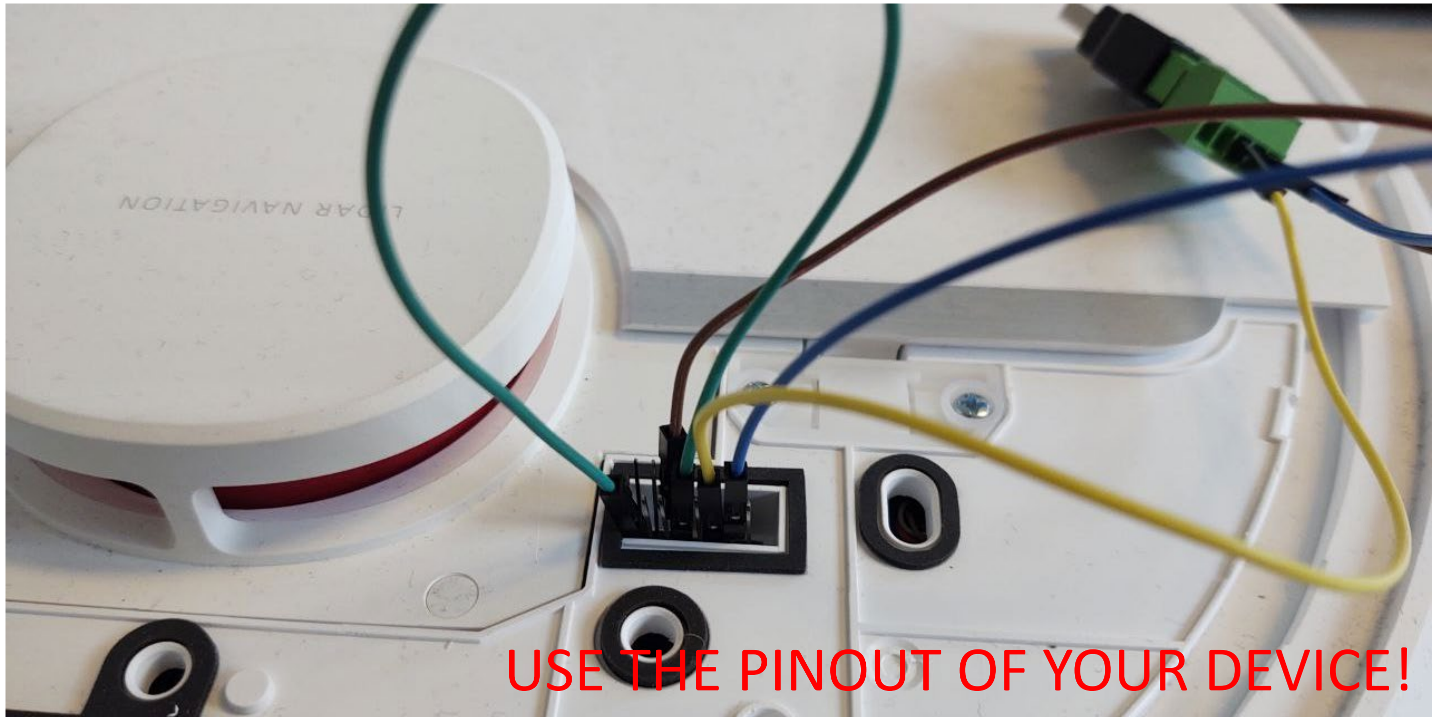
Root Step 2

- Boot the Robot in FEL mode
  - Connect MicroUSB to your computer (no USB hub, preferred USB 2.0 port)
  - Connect BOOT_SEL to GND
    - Use jumper wire (see 2a)
    - Press FEL button on custom adapter (see 2b)
    - Alternatively: if you have UART, press “2” while booting
  - Press the power button for 3 seconds
    - USB device should show up on your computer
      - Windows: Check Device Manager
      - Linux: lsusb
    - You might need to install the Phoenixsuit drivers located in “Drivers/AW_Driver”
      - (via Device Manager -> Unknown Device -> Update drivers)

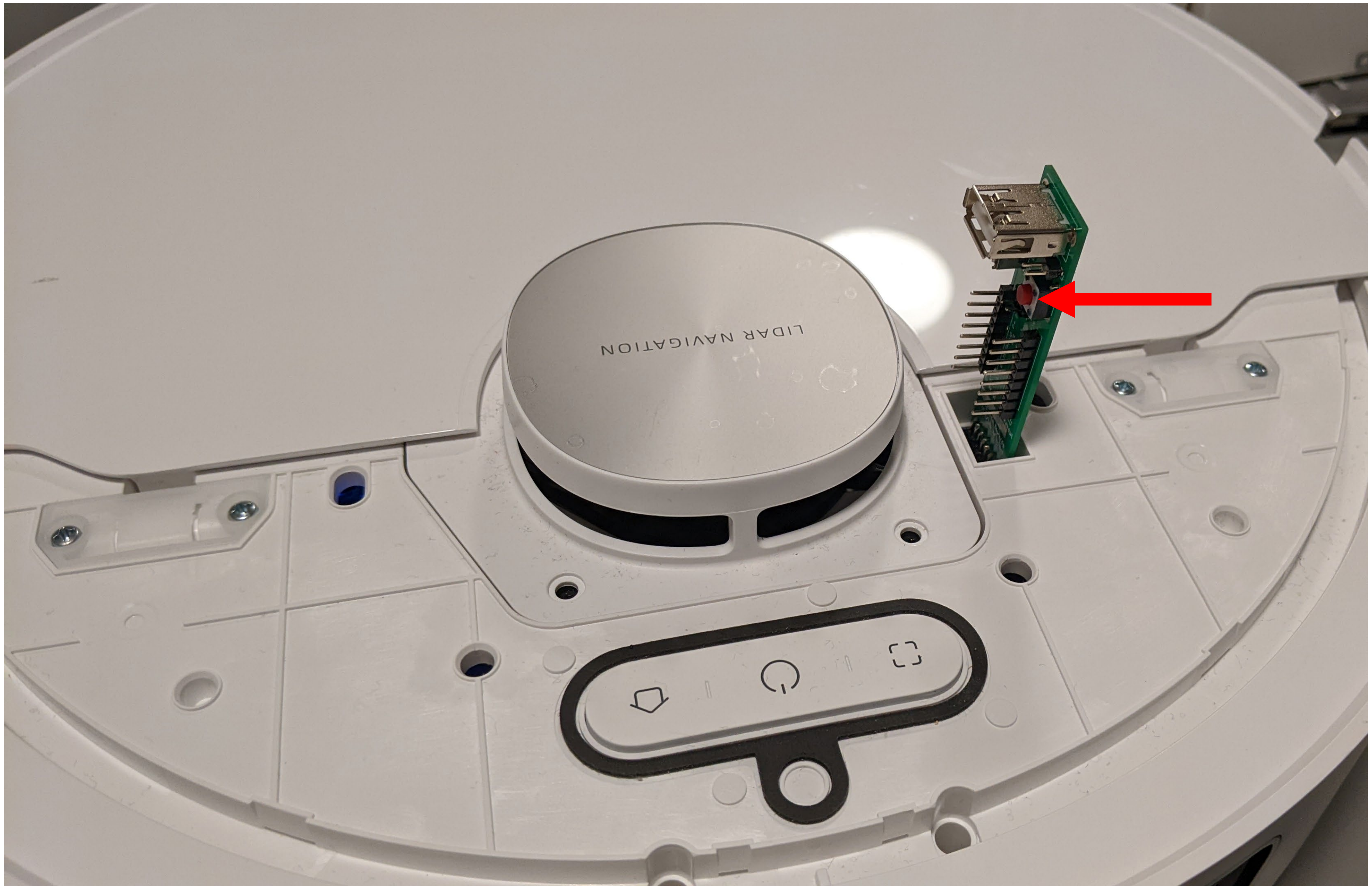
Root Step 2a

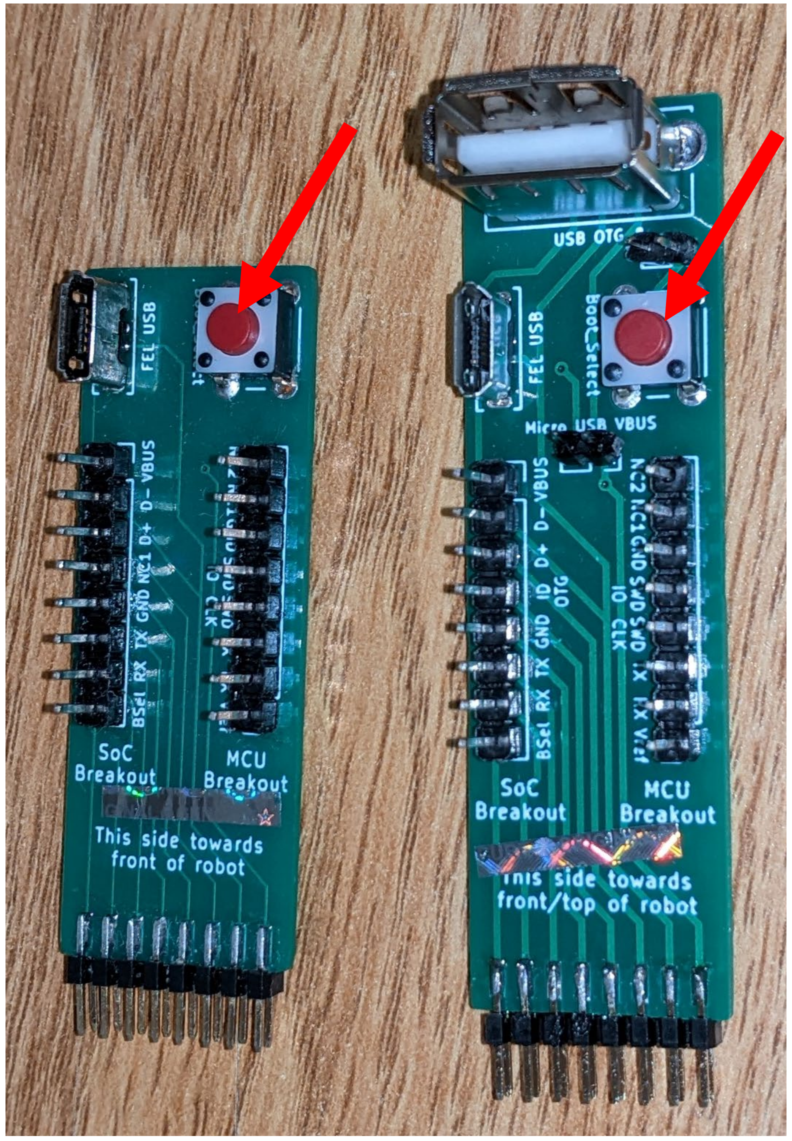
- Jumper wires: Ideally you have 2mm headers
  - Make sure that the USB connection stays stable before you trigger the update (check the Windows Device Manager)
  - 2.54 pitch jumper cables might fit but can cause issues





Root Step 2b

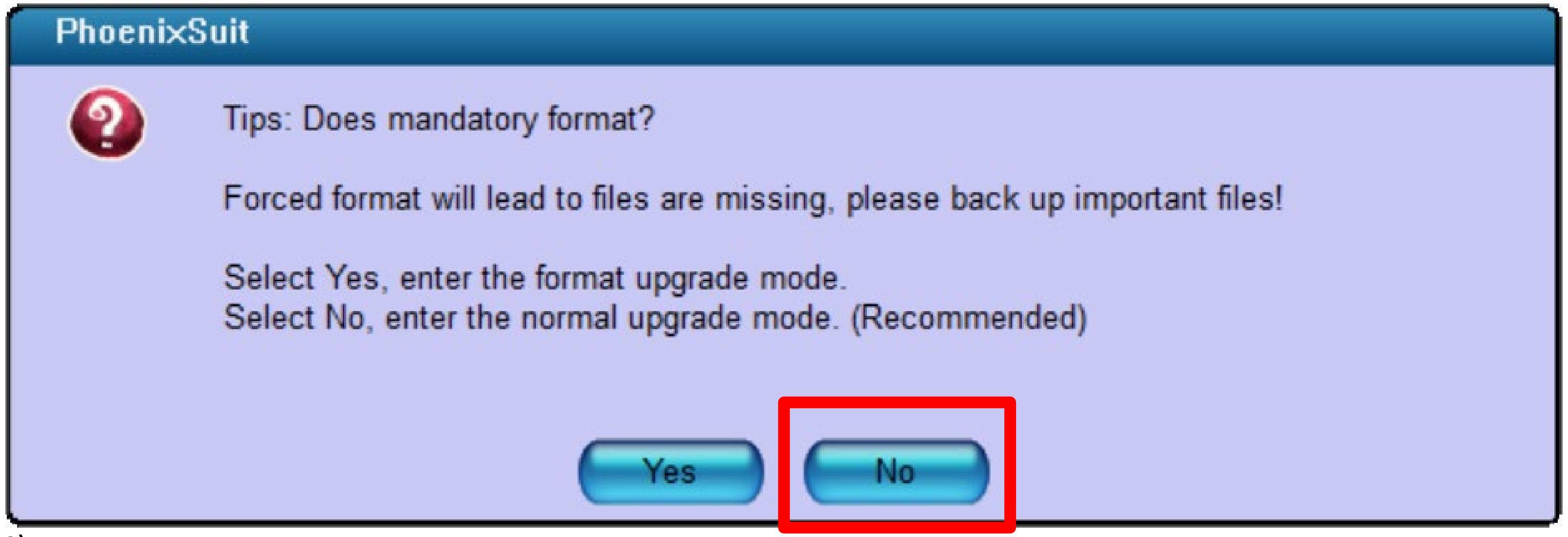

LIDAR NAVIGATION


FEL USB
USB OTG
Boot_Select
Micro USB VBUS
BSel RX TX GND NC1 D+ D- VBUS
BSel RX TX GND ID D+ D- VBUS
OTG
SoC Breakout
MCU Breakout
This side towards front of robot
This side towards front/top of robot

Root Step 3

- Disconnect BOOT_SEL before proceeding
- In Phoenixsuit
  - Click on Update
  - If asked about mandatory format, select "No" !
  - The flashing process will get stuck/fail -> that's okay and expected!

Make sure that FEL / Boot_SEL is disconnected!!



PhoenixSuit

Tips: Does mandatory format?

Forced format will lead to files are missing, please back up important files!

Select Yes, enter the format upgrade mode.
Select No, enter the normal upgrade mode. (Recommended)

Root Step 3 Troubleshooting

- Do not use USB hubs
- Use only USB 2.0 connections
- If the device is detected initially, but disappears:
  - Try to use a different USB port
  - If you use VMs: Make sure that the USB filter is correct

Root Step 4

- Device should reappear as an ADB device.
- Open command line (cmd) and run:

  # fastboot devices

  - should return "Android Fastboot   fastboot"

  # fastboot getvar config

  - should return "config: aabbccddeeff11223344556677889900“
- If you use Windows, the fastboot binary path is likely here:
  - C:\Program Files (x86)\Minimal ADB and Fastboot\fastboot.exe
- Go to https://builder.dontvacuum.me
  - generate a FEL firmware
  - supplying the string from above as the config value (aabbccddeeff11223344556677889900)
  - safe this string, as you will need it again in the future

Cheatsheet with commands here: https://builder.dontvacuum.me/nextgen/fastboot-cheatcheat.txt

Don’t panic if dustbuilder does not accept your config value. Go to https://check.builder.dontvacuum.me

Root Step 4a (for debugging)

- Run additionally these commands and save them:

  # fastboot getvar dustversion

  - should return “dustversion: 2024.xx.xx“

  # fastboot getvar ramsize

  - should return “ramsize: XXX000000000000ZXXXXXXXXXXXXXXX “
  - If Z=4 -> 1024MByte RAM, if Z=2 -> 512MByte RAM

  # fastboot getvar toc0hash

  - should return “toc0hash: aabbccddeeff11223344556677889900“

  # fastboot getvar toc1hash

  - should return “toc1hash: aabbccddeeff11223344556677889900“

  # fastboot getvar toc1version

  - should return “toc1version: U-Boot xxxx“

If Dustbuilder does not accept your config value, please provide all these information to us. This helps us to figure out what kind of settings we need to configure.

Upload this information and your generated files in 4b) to https://check.builder.dontvacuum.me (only if config value is unknown!)

Root Step 4b (for fallback)

- In case something goes wrong, this step helps to fix your device:
  - Linux/Windows:
    - fastboot get_staged dustx100.bin
    - fastboot oem stage1
    - fastboot get_staged dustx101.bin
    - fastboot oem stage2
    - fastboot get_staged dustx102.bin
  - Expected outputs:

```
Uploading 'dustx100.bin'                    OKAY [ 38.853s]
Finished. Total time: 38.854s
```

You should get ~400MBytes per file!

If your robot reboots in stage1, just skip the first step and continue with “fastboot oem stage1”

Provide us the files in case your robot is “soft-bricked”

Root Step 5

- Did you do 4b? If not, go back one page and do it!!
- After you get the firmware package, unpack the archive
  - Select the file “_dreame.XXX_phoenixsuit.img” as a firmware file in PhoenixSuit.exe
  - Power off the device
  - Repeat Steps 2 and 3 (with the new file)

Root Step 6

- copy the 8 character string from check.txt and run fastboot from your unpacked image folder:

```
# fastboot getvar config
# fastboot oem dust aabbccdd
```

  – this should return OKAY, if it does not, DO NOT PROCEED!

- Now we can start the rooting process:

```
# fastboot oem prep
```

  – this should return OKAY, if it does not, DO NOT PROCEED!

```
# fastboot flash toc1 toc1.img
```

  – this should return OKAY, if it does not, DO NOT PROCEED!

Root Step 7

- To actually root the device, run commands:

```
# fastboot flash boot1 boot.img
# fastboot flash boot2 boot.img
# fastboot flash rootfs1 rootfs.img
# fastboot flash rootfs2 rootfs.img
```

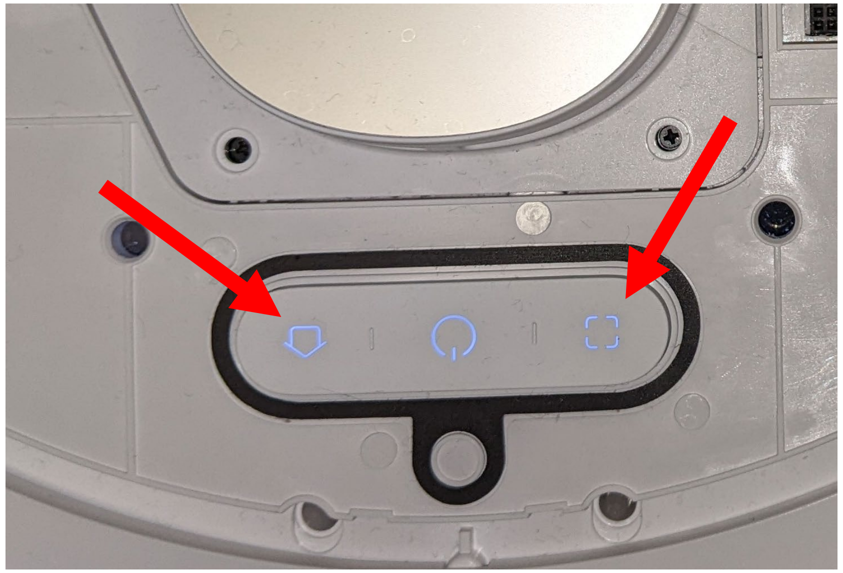
  - this should return OKAY. It might take 5-10 minutes for the commands to complete. Do not worry. Do not power off the device!
  - Ignore the "Invalid sparse file format at header magic“ error

Root Step 8

- We should be done now and need to restart the device:

```
# fastboot reboot
```

- The device should now reboot (the jingle should play)
- Press the outside 2 buttons to put the device in provisioning mode
- Connect to the robots WiFi, ssh to 192.168.5.1 (user:root)
  - Use SSH key or compute root password from SN



FIRMWARE CUSTOMIZATION

You have now installed a custom firmware ;)

Customization of firmware

- The firmware builder adds to hooks to the start process
  - /data/_root_postboot.sh (gets executed after boot)
- At each boot we check for the presence of the custom files
  - Safety measurement: In case something goes wrong, a factory reset will delete this files
- An example file can be found in /misc
- Your SSH keys are stored in /mnt/misc/authorized_keys
  - (this key will remain the same for any firmware update, factory resets)
  - (change the file manually if you need to change the key)

Valetudo installation

- Download valetudo binary to /data:
  - https://github.com/Hypfer/Valetudo/releases
  - Device dependent (check our website):
    - 1C, F9, D9, L10u: valetudo-armv7-lowmem
    - 1T, L10 Pro, Z10, L10sU, D10s Pro, D10s Plus, X40: valetudo-aarch64
  - Use scp / WinSCP to transfer file to /data
  - "wget https://github.com/Hypfer/Valetudo/releases/latest/download/valetudo-armv7-lowmem -O /data/valetudo"
- Make valetudo executable
  - "chmod +x /data/valetudo"
- Enable boot script
  - "cp /misc/_root_postboot.sh.tpl /data/_root_postboot.sh"
  - "chmod +x /data/_root_postboot.sh"
- Reboot

You might need to add "--no-check-certificate" if wget fails

Thank you for watching!

@dgi_DE

Website: dontvacuum.me