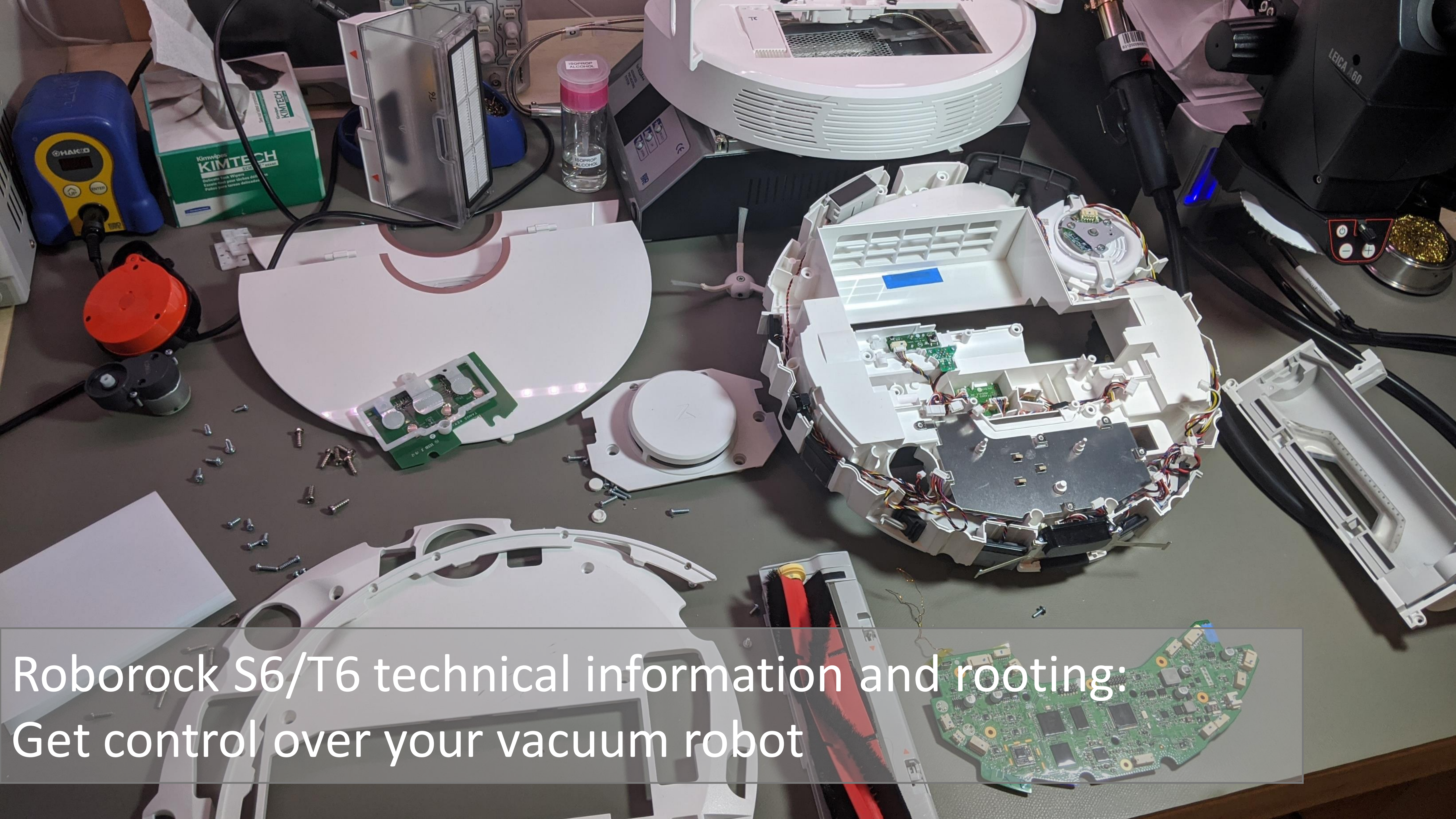


Roborock S6/T6 technical information and rooting: Get control over your vacuum robot

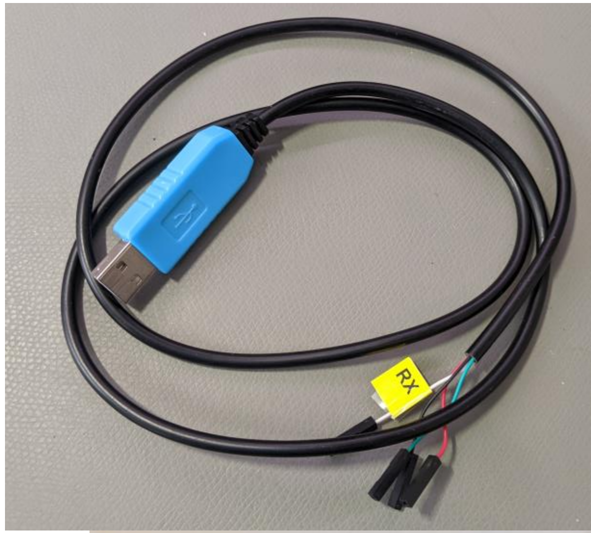
Before you continue: Please watch the whole video before you start your adventure

All commands and links are in the description

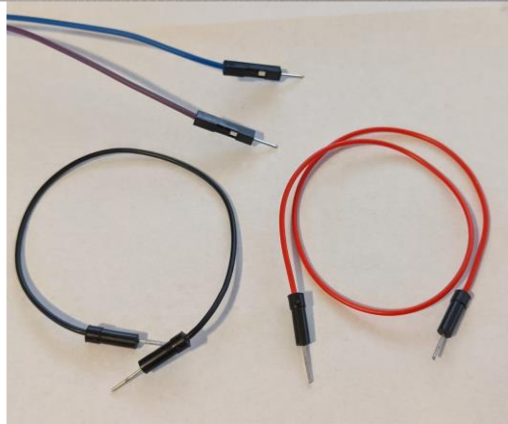
You might want to join the Telegram group

Tools required for root

- Screwdrivers
- UART-USB adapter (3.3V, also known as TTL adapter)
  - Typical chipsets:
    - FT232RL, FT232, PL2303TA or CP2102
  - Price ~10 USD/Euro



- Copper wires or 3x Breadboard Jumper Wires
- Tape



Software required

- Windows
  - Putty (for UART and SSH access)
  - WinSCP (to transfer files)

- Linux
  - Minicom (for UART)
  - SCP (should be already integrated in OpenSSH)

Additional requirements

- A Wi-Fi capable device (e.g. Notebook or Wi-Fi adapter)
- Alternatively:
  - Your robots needs to be provisioned (connected to Wi-Fi)
  - You need to know its IP address

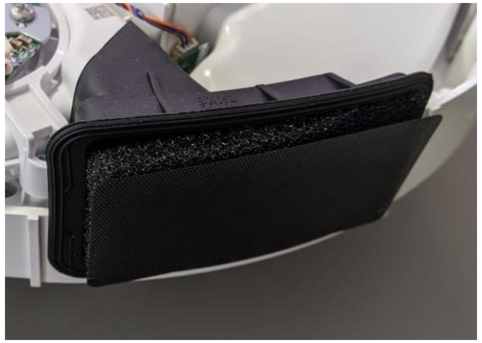
- Make sure that the battery of your robot is charged

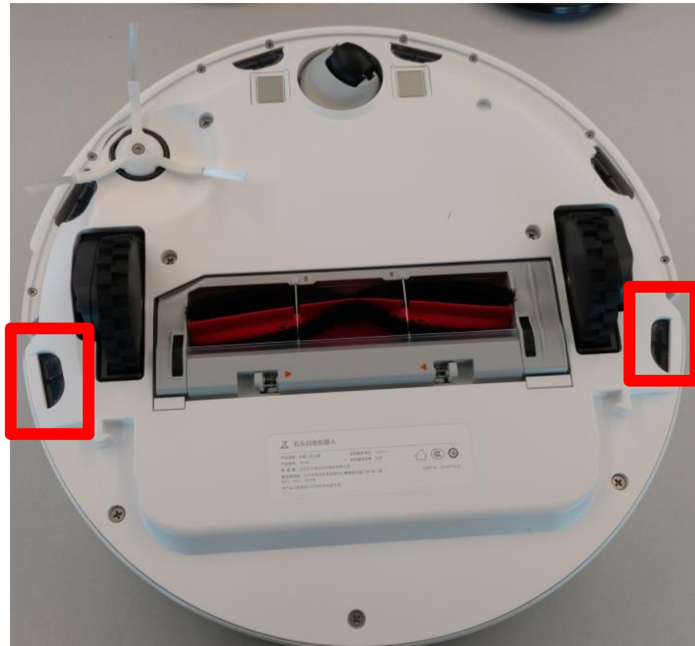
Why get root access?

- Remove geo-blocking (convert T6 into S6)
  - If device is a T6, it works only in mainland china
  - Change the region of the vacuum robot so that it works outside of mainland china
- 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

Differences to V1 and S5

- Hardware
  - Mostly the same
  - In comparison to S5:
    - Additional filter for fan (reduced noise)
    - 2 additional IR drop sensors
    - New type of wheels
    - New main brush
- Software/Configuration
  - Firmware is now encrypted and signed
    - Old update method does not work
    - Custom updates cannot be done over the network anymore
  - Configuration signed and bound to CPU ID
    - Region switch in roborock.conf is not possible anymore
  - TUYA integration as alternative to miOT cloud connection





Technical information

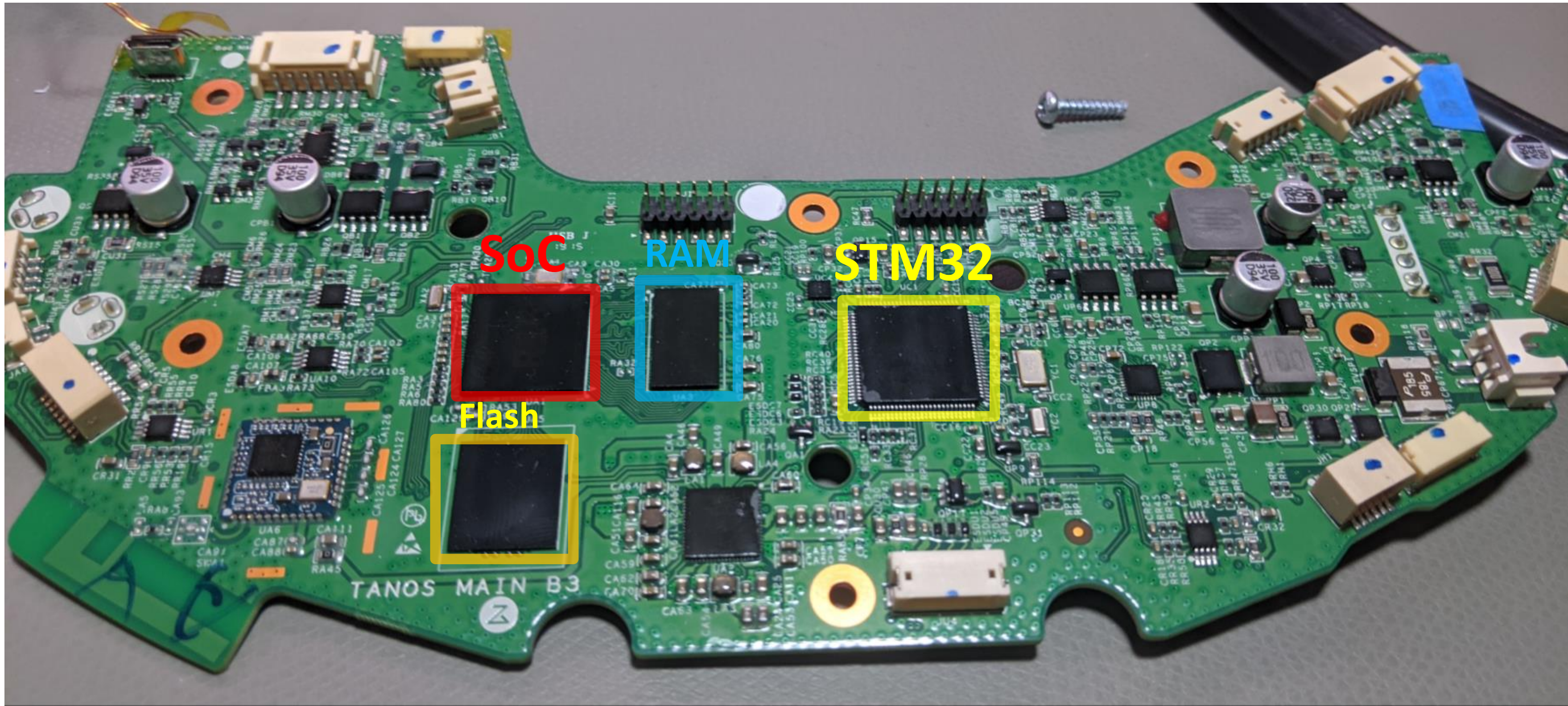
SoC: Allwinner R16 (Quadcore ARM)

Flash: 4 GByte eMMC

RAM: 512 Mbyte

MCU: STM32F103VET6

Wi-Fi: Realtek 8189es (2.4 GHz only)



Software

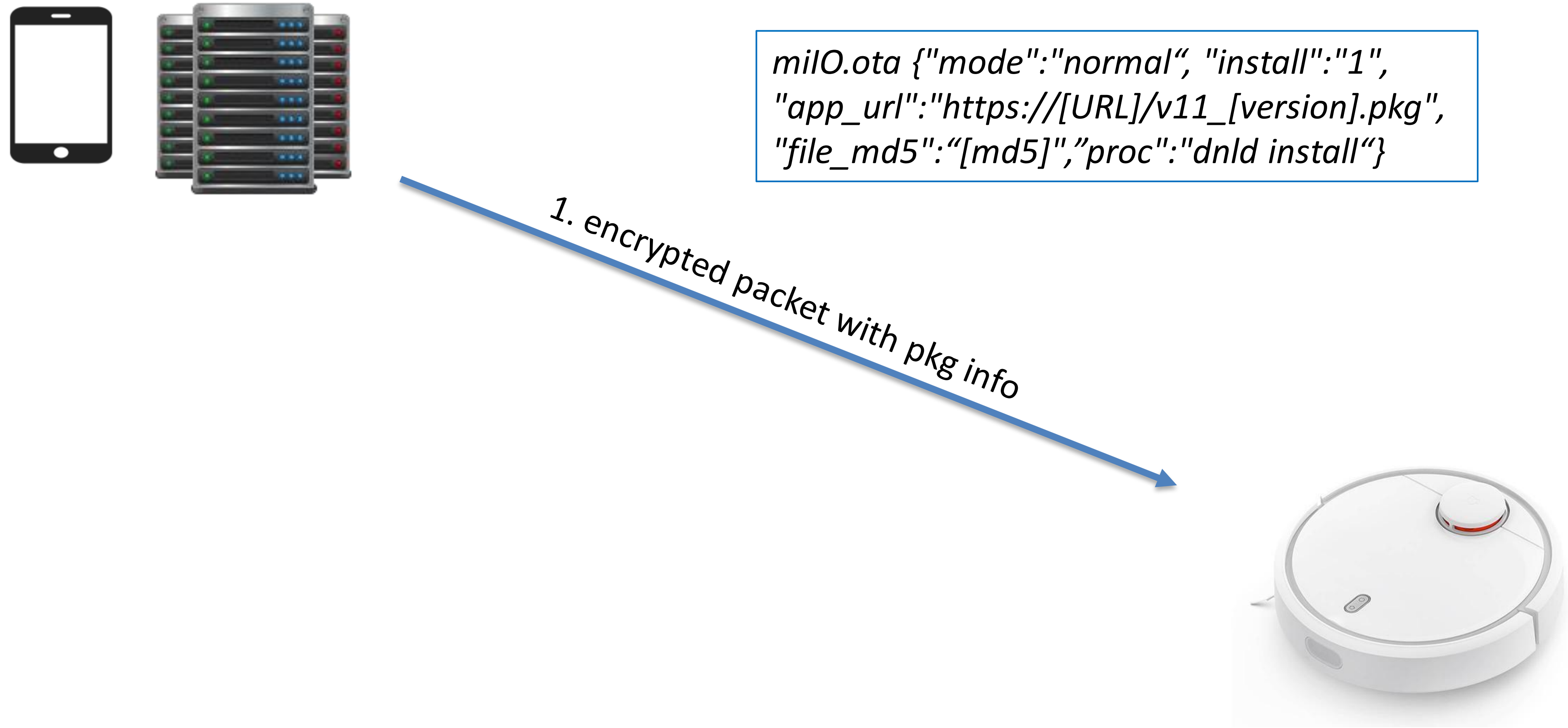
- Ubuntu 14.04.3 LTS (Kernel 3.4.xxx)
  - Depending on version: regular Ubuntu or stripped OS (>1048)
- Player 3.10-svn (relabeled as “rr_loader”)
  - Open-Source Cross-platform robot device interface & server
- Proprietary software (/opt/rockrobo)
  - AppProxy: controls device functionality (start, stop, map upload, etc.)
  - miIO-client/tuya-client: cloud communication interfaces
  - SysUpdate: responsible for system updates installation
  - Custom adbd-version
- SSH:
  - OpenSSH (for versions <=1048)
  - Dropbear (for newer versions)
- iptables firewall enabled
  - Blocks Port 22 (SSHd) + Port 6665 (player)
  - All IPv6 blocked

Dropbear only supports SCP, no SFTP

eMMC Layout

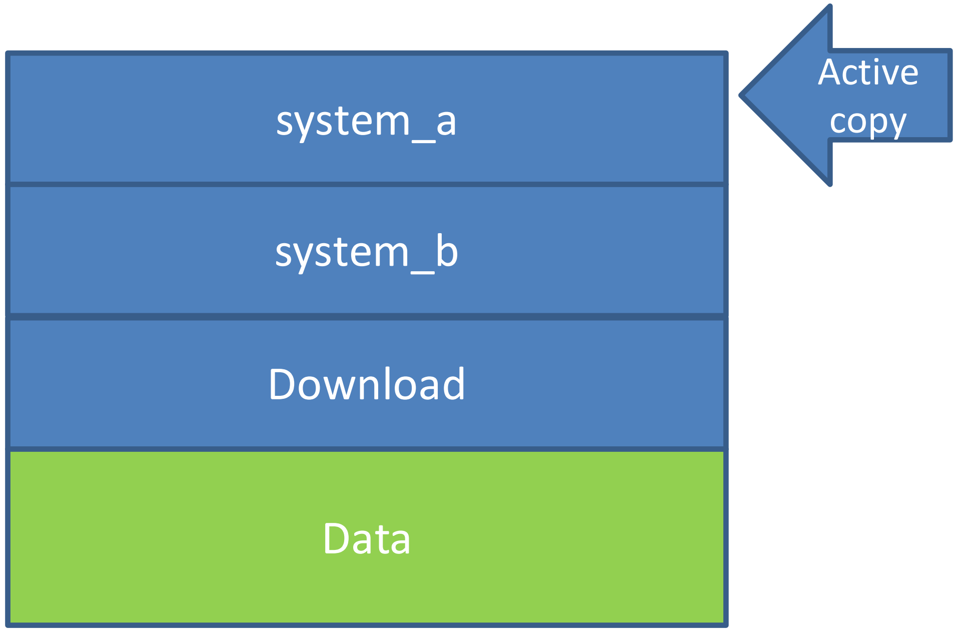
Label	Content	Size in MByte
boot-res	bitmaps & some wav files	8
env	uboot cmd line	16
app	device.conf (DID, key, MAC), adb.conf, rockrobo.conf (+sign) , vinda	16
recovery	fallback copy of OS	512
system_a	copy of OS (active by default)	512
system_b	copy of OS (passive by default)	512
Download	temporary unpacked OS update	528
reserve	config + calibration files	16
UDISK/Data	logs, maps	~1900

Default Update process



```
miIO.ota {"mode":"normal“, "install":"1",
"app_url":"https://[URL]/v11_[version].pkg",
"file_md5":“[md5]",”proc":"dnld install“}
```

Default Update process



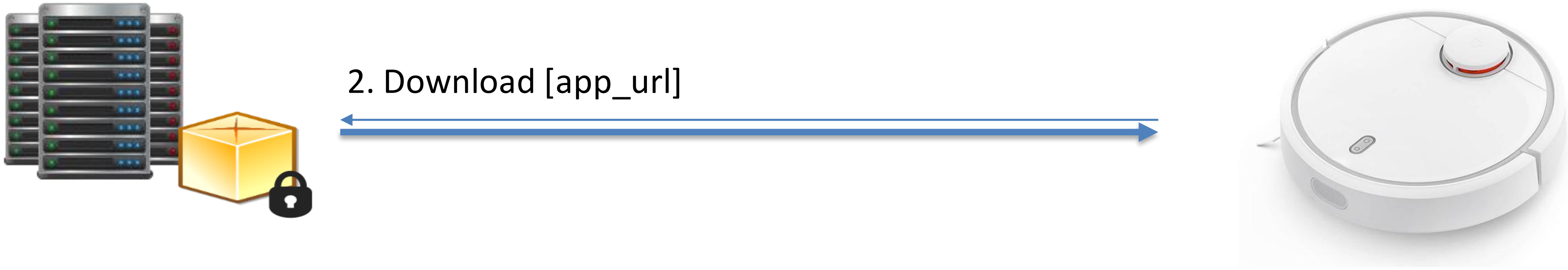
system_a

system_b

Download

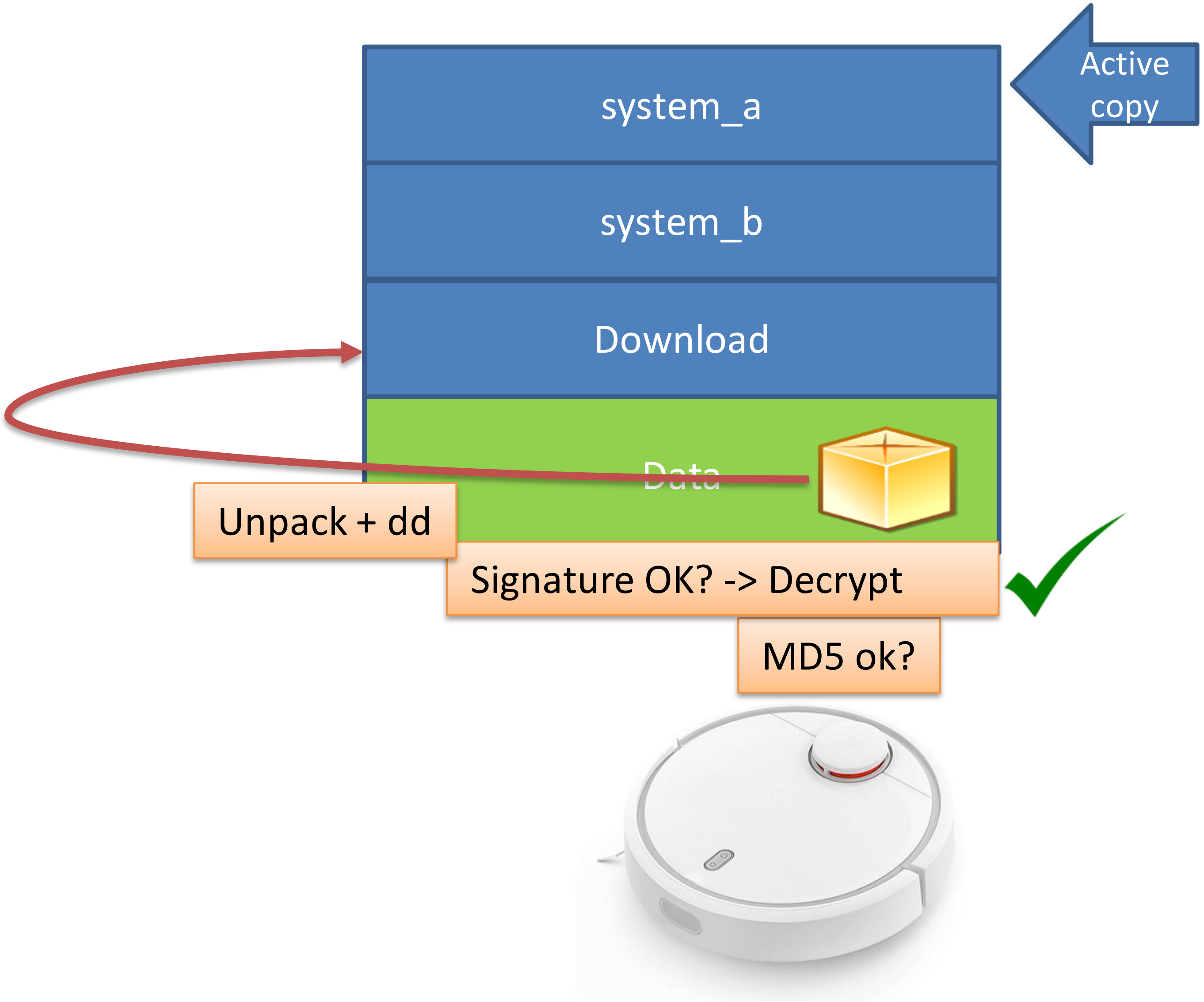
Data

Active copy



2. Download [app_url]

Default Update process



Active copy

system_a

system_b

Download

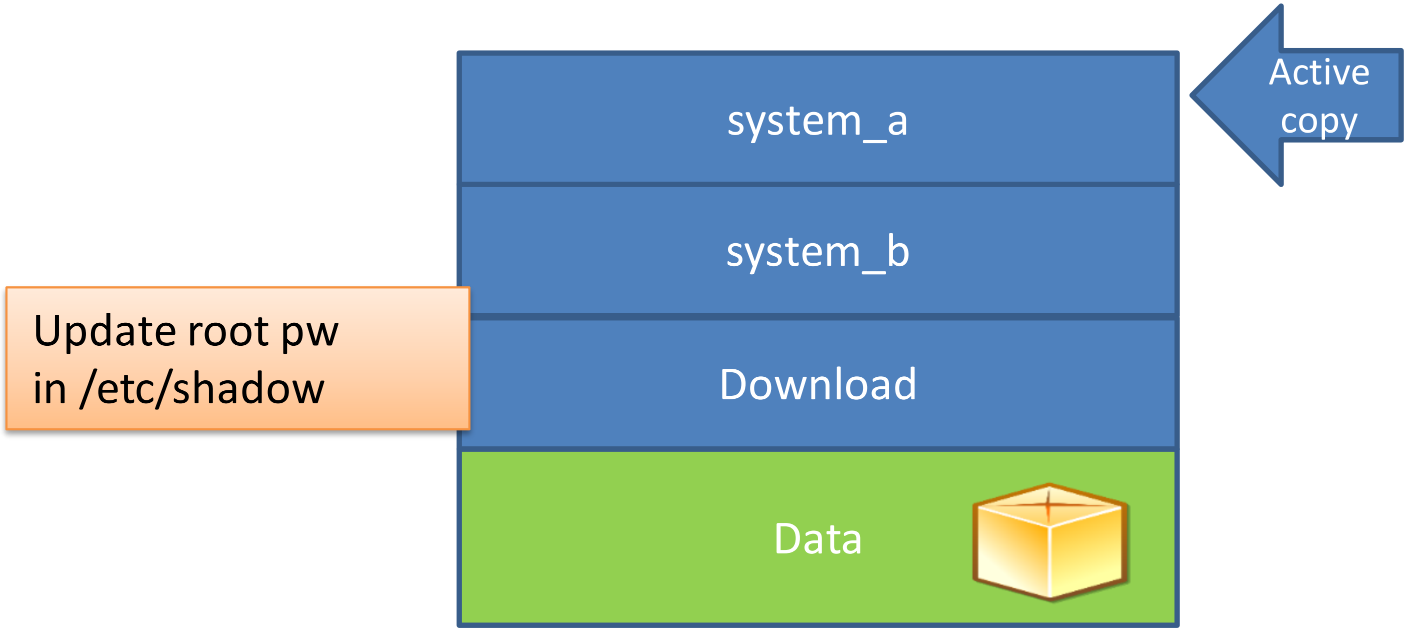
Data

Unpack + dd

Signature OK? -> Decrypt ✓

MD5 ok?

Default Update process



Active copy

system_a

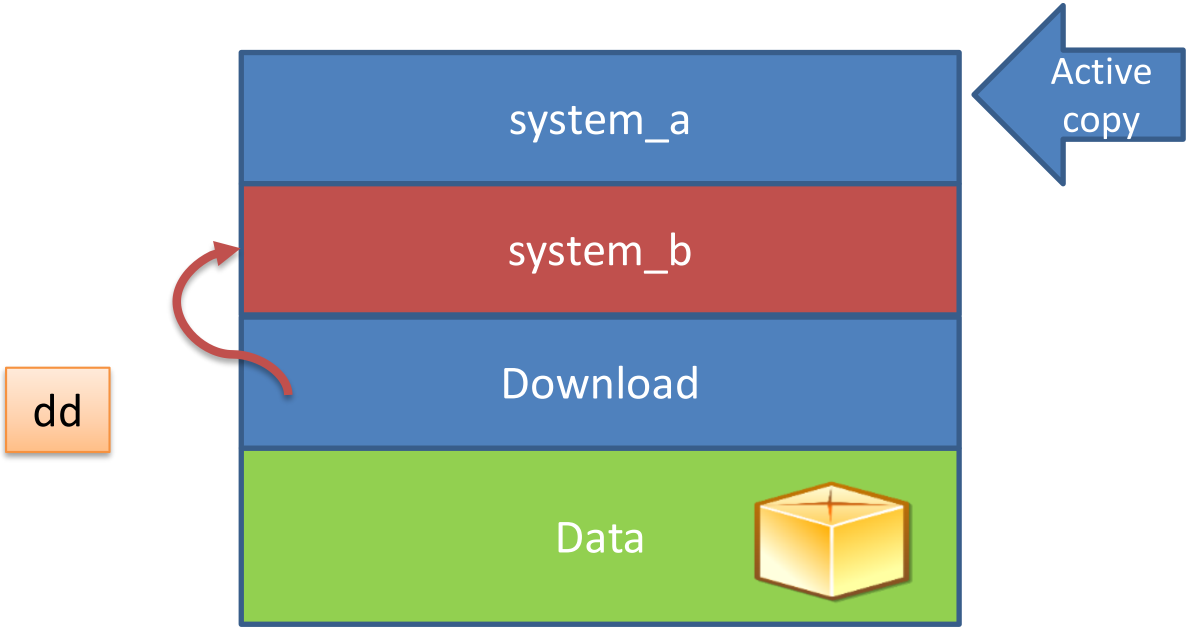
system_b

Update root pw in /etc/shadow

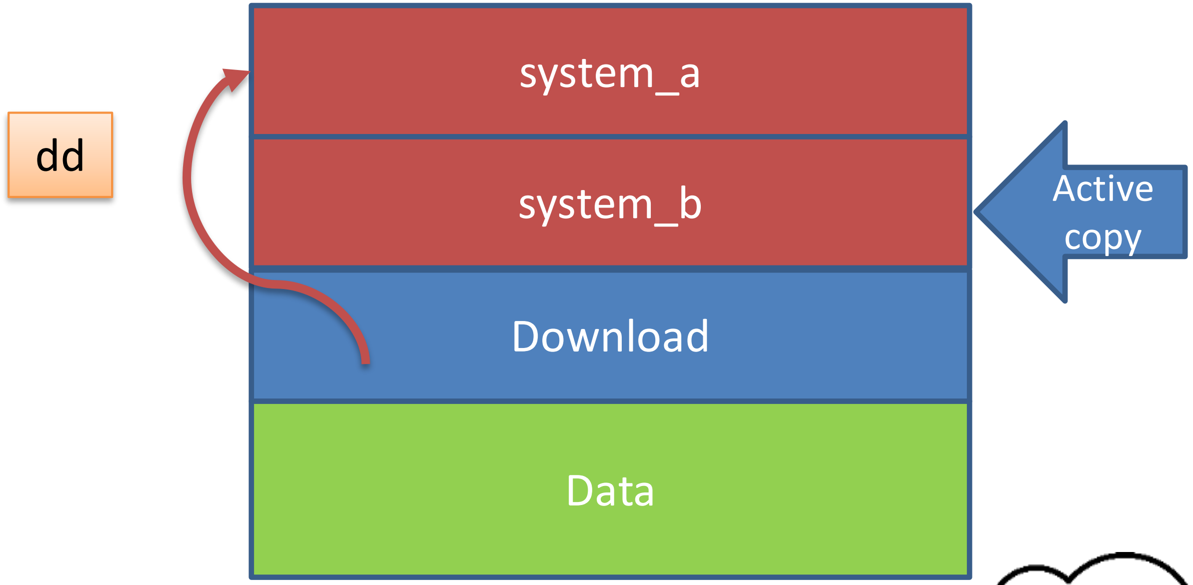
Download

Data

Default Update process



Default Update process



dd

system_a

system_b

Download

Data

Active copy

Disclaimer

- I take no responsibility for bricked devices.
- You will likely void your warranty by disassembling your device

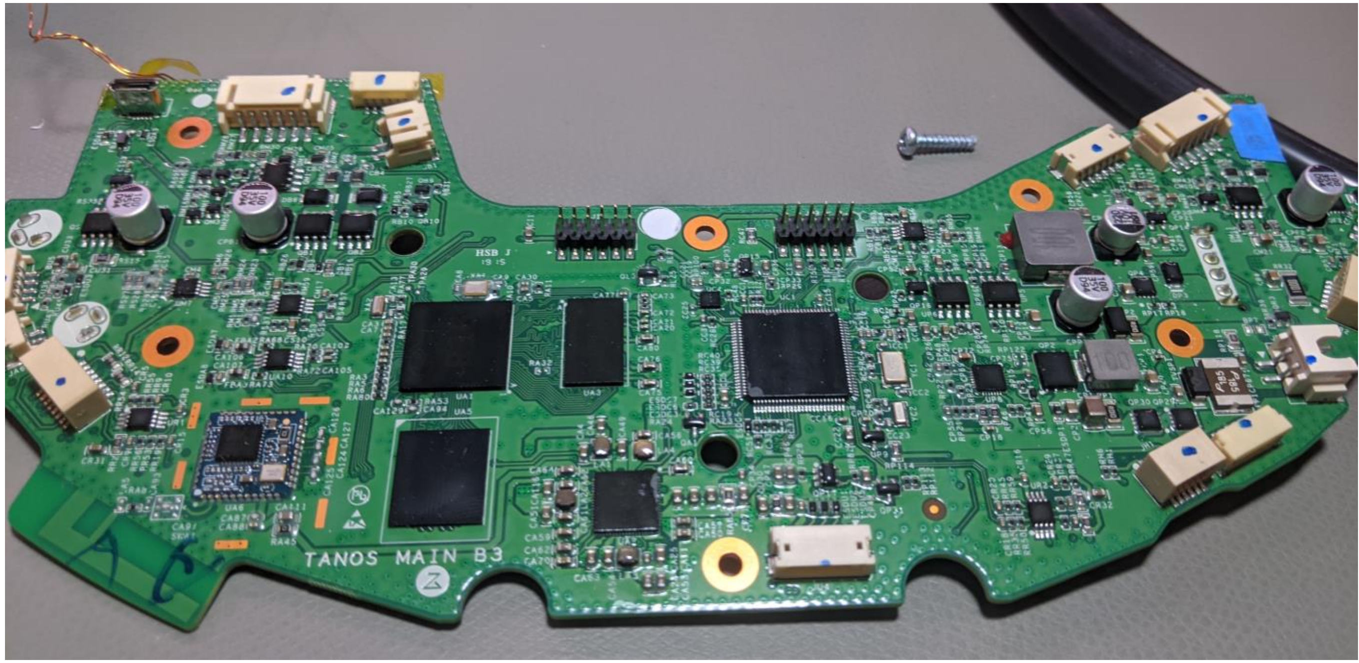
- Be very careful if you type copy commands
  - You can find all commands in the description, just copy + paste them

How to root

- Phase 1: disassembly of the device
  - needed to get access to test-pads on the PCB
  - Watch my Youtube video for the steps
- Phase 2: connect to the UART and enable SSH
  - Might require soldering or a second person
  - Extract root password via bootloader
  - Boot and login into Ubuntu, disable the firewall
  - Connect over SSH and enable permanent root
- Phase 3: Install custom firmware
  - Copy custom firmware over SSH and install in System_B
  - Reboot and install in System_A

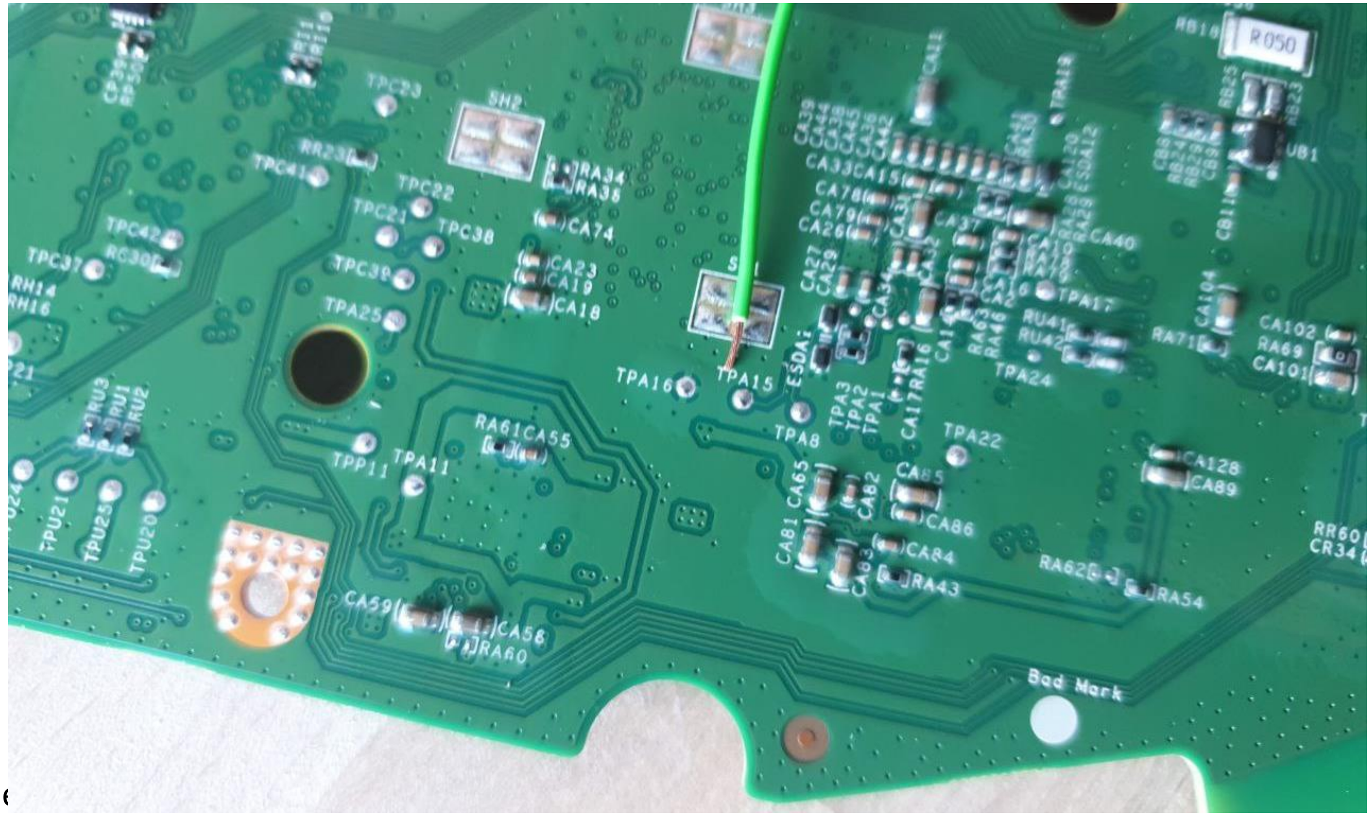
Phase 1

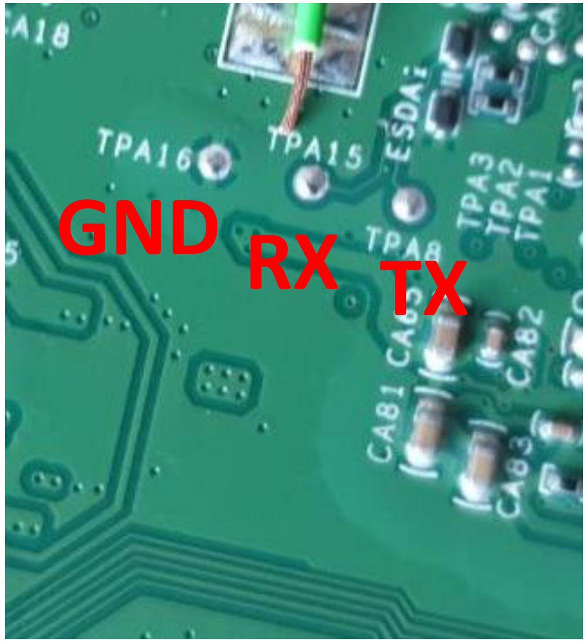
- At this point you should have disassembled the robot and should have the bare PCB
- Reconnect the button PCB



Phase 2: Step 1a

- Solderless method
  - You need a second person to hold the wires
  - Or, you can try to tape them in place

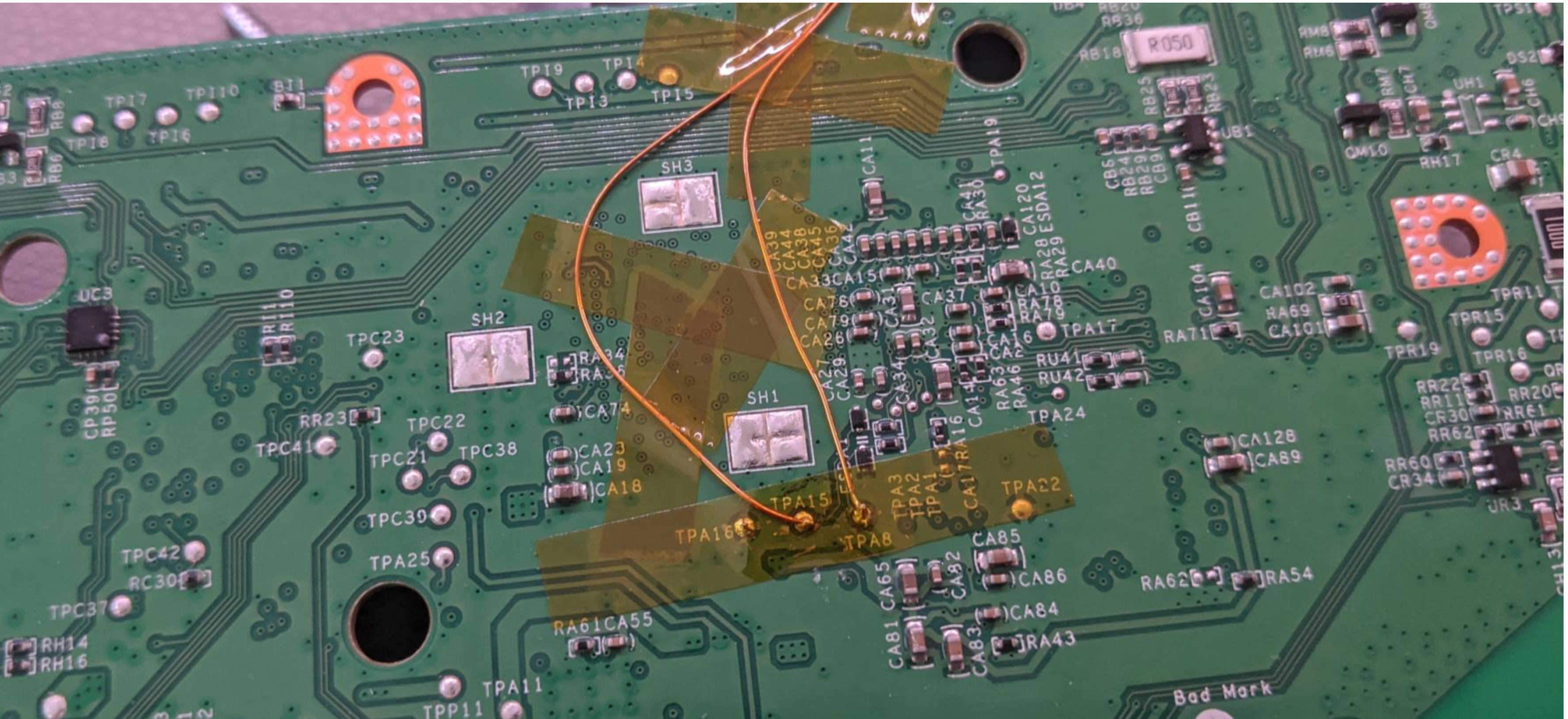


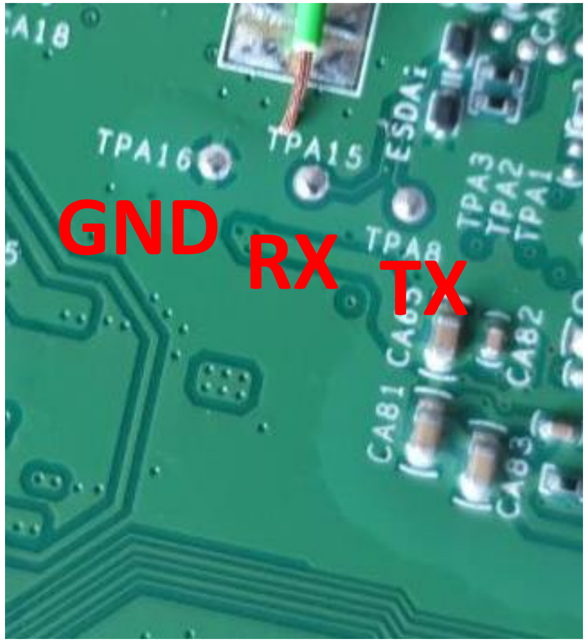


Hint:
TX is the output
RX is the input
GND can be also obtained from USB

Phase 2: Step 1b

- Solder method
  - Use tape to provide a strain relief for the wires before soldering
  - Route your cable around the ground pads

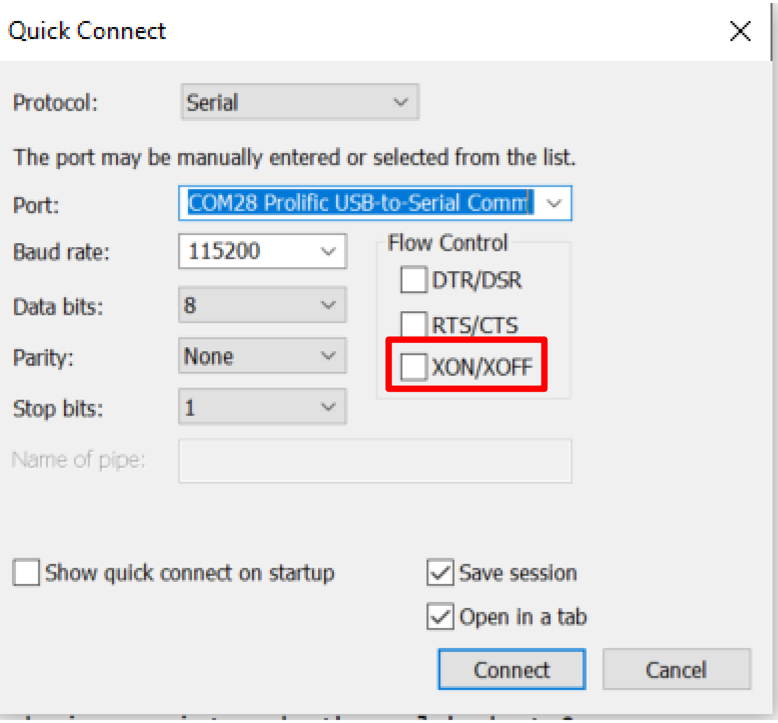




Hint:
TX is the output
RX is the input
GND can be also obtained from USB

Phase 2: Step 2

- Know where RX and TX on your adapter is
- Configure your UART program
  - Baud: 115200
  - Flow control: off (!)
- Test the settings without robot



Quick Connect

Protocol: Serial

The port may be manually entered or selected from the list.

Port: COM28 Prolific USB-to-Serial Comm

Baud rate: 115200

Data bits: 8

Parity: None

Stop bits: 1

Name of pipe:

Flow Control

□ DTR/DSR

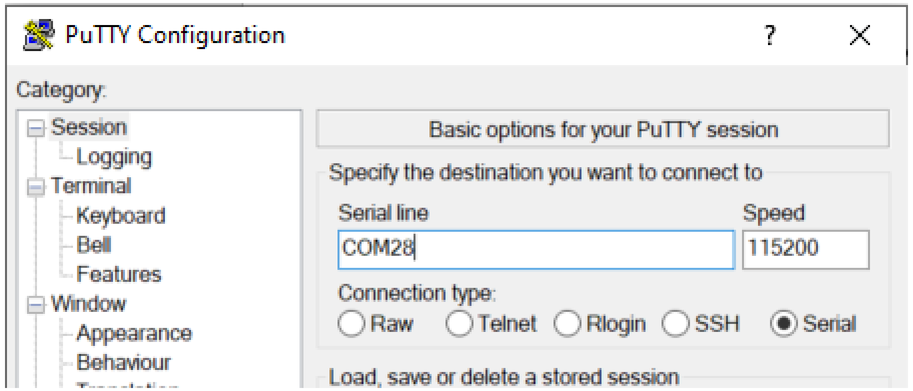
□ RTS/CTS

□ XON/XOFF

□ Show quick connect on startup

☑ Save session

☑ Open in a tab



PuTTY Configuration

Category:

Basic options for your PuTTY session

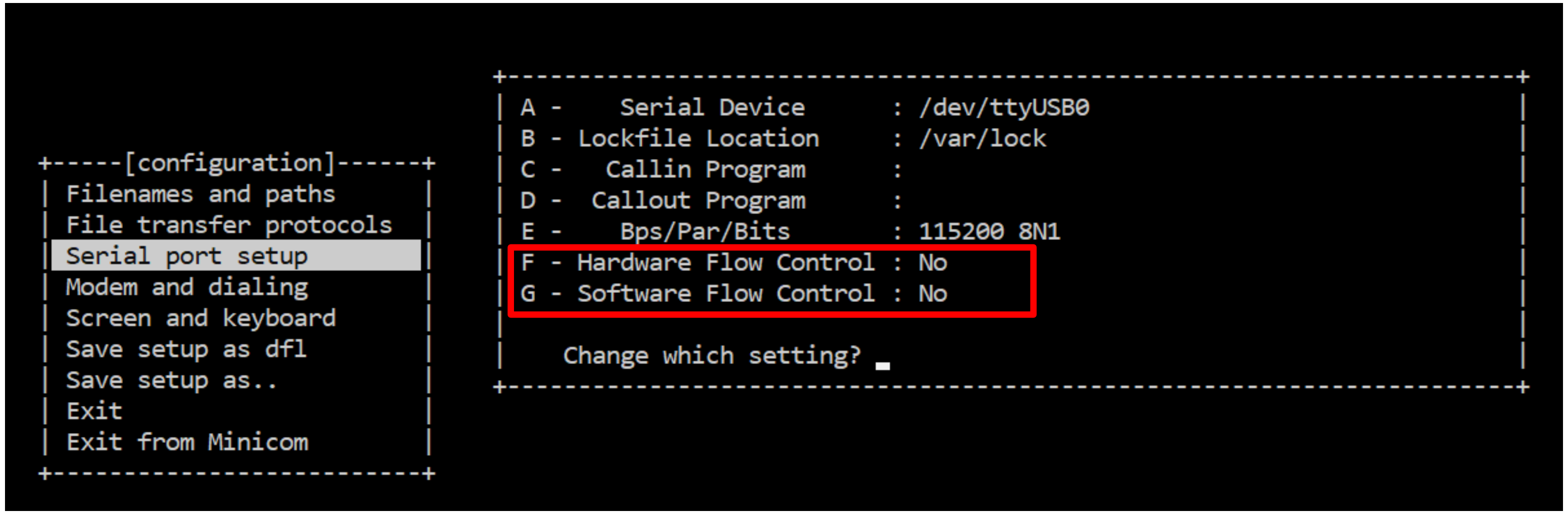
Specify the destination you want to connect to

Serial line: COM28

Speed: 115200

Connection type:
○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

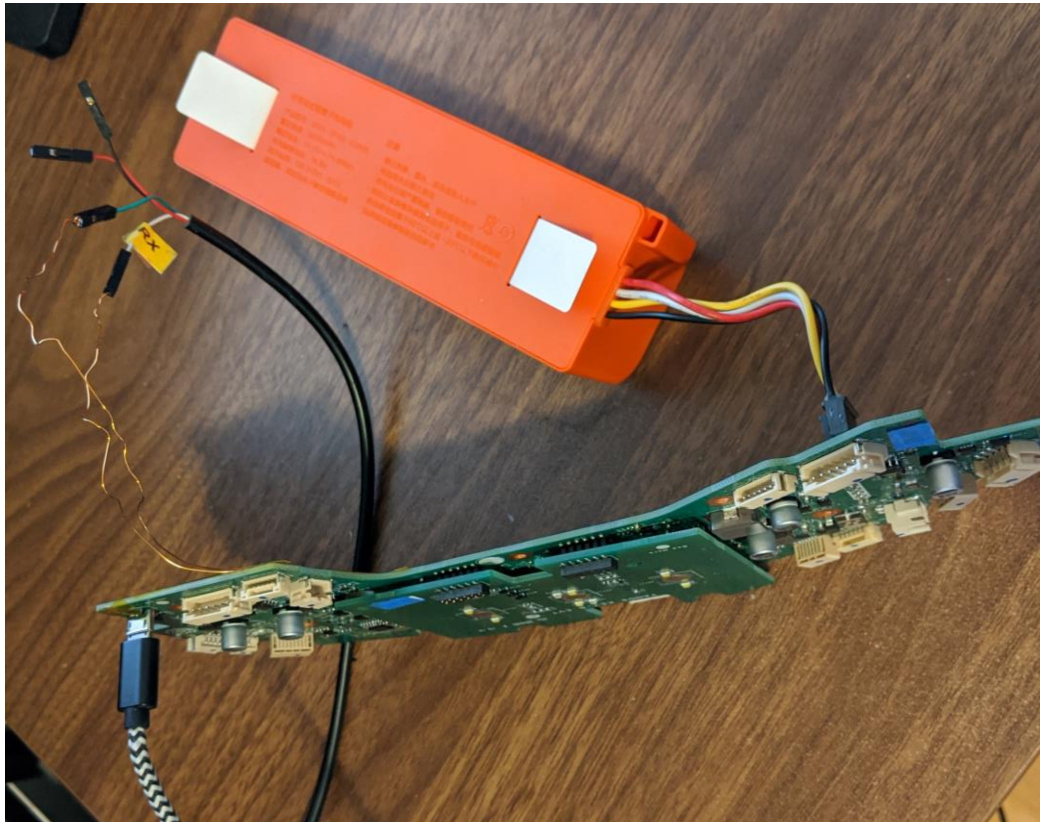
Load, save or delete a stored session



```
                          +-----------------------------------------------------------------------+
                          | A -    Serial Device      : /dev/ttyUSB0                              |
                          | B - Lockfile Location     : /var/lock                                 |
                          | C -   Callin Program      :                                           |
+-----[configuration]------+ D -  Callout Program      :                                           |
| Filenames and paths      | E -    Bps/Par/Bits       : 115200 8N1                                |
| File transfer protocols  | F - Hardware Flow Control : No                                        |
| Serial port setup        | G - Software Flow Control : No                                        |
| Modem and dialing        |                                                                       |
| Screen and keyboard      |    Change which setting?                                              |
| Save setup as dfl        +-----------------------------------------------------------------------+
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+--------------------------+
```

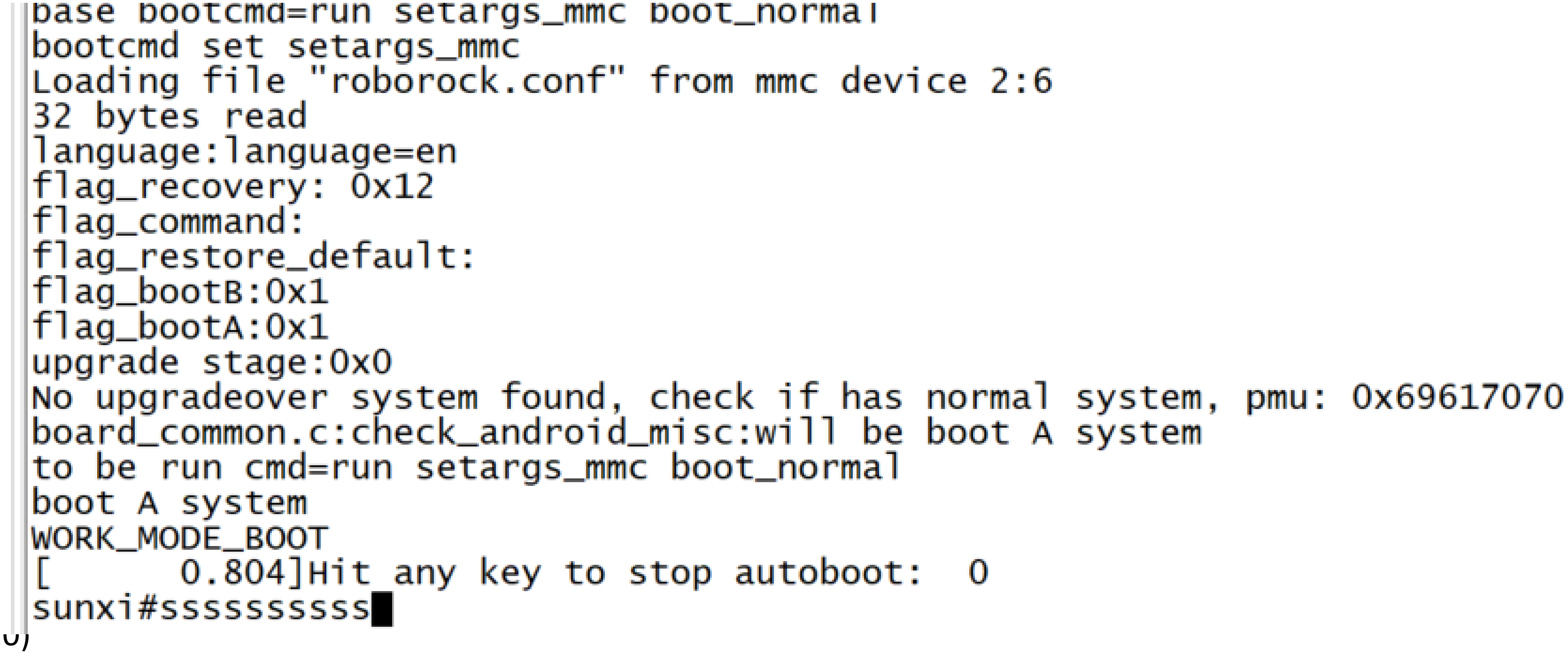
Phase 2: Step 3

- Connect battery to PCB
- Connect serial wires to PCB
  - If you connect a MicroUSB cable to the same computer, you only need 2 wires (TX, RX)
  - Do not connect 5V (red cable)!
  - Test for correct connection
    - Press middle button (<1s)
    - You should see some output



Phase 2: Step 4

- Inside the terminal program
  - Hold “s” key on your keyboard
  - At the same time: Press middle button for 3 seconds
  - We want to see this:



```
base bootcmd=run setargs_mmc boot_normal
bootcmd set setargs_mmc
Loading file "roborock.conf" from mmc device 2:6
32 bytes read
language:language=en
flag_recovery: 0x12
flag_command:
flag_restore_default:
flag_bootB:0x1
flag_bootA:0x1
upgrade stage:0x0
No upgradeover system found, check if has normal system, pmu: 0x69617070
board_common.c:check_android_misc:will be boot A system
to be run cmd=run setargs_mmc boot_normal
boot A system
WORK_MODE_BOOT
[      0.804]Hit any key to stop autoboot:  0
sunxi#ssssssssss
```

Phase 2: Step 5

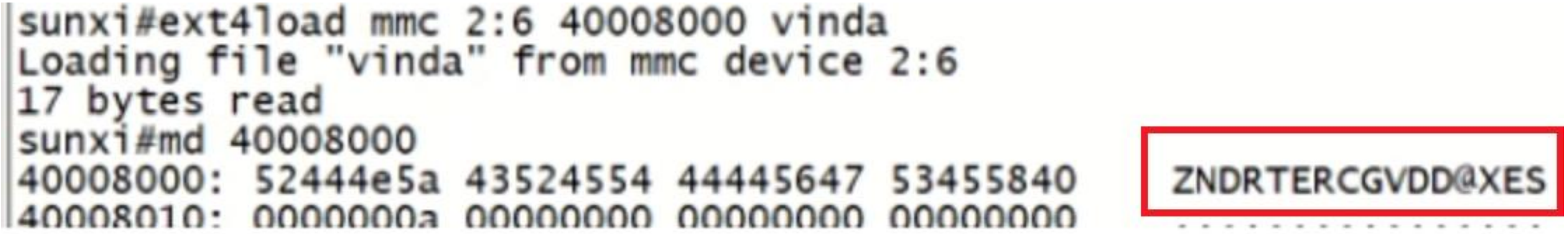
Commands for copy+paste: https://builder.dontvacuum.me/x6cheatsheet.txt

Paste works often over right-mouse click

- Type “ext4load mmc 2:6 40008000 vinda”
- Type “md 40008000”

  (if you are holding the cables, you may release them for now)

- Copy the string from the first line (16 characters)

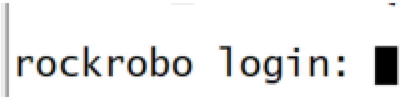


```
sunxi#ext4load mmc 2:6 40008000 vinda
Loading file "vinda" from mmc device 2:6
17 bytes read
sunxi#md 40008000
40008000: 52444e5a 43524554 44445647 53455840    ZNDRTERCGVDD@XES
40008010: 0000000a 00000000 00000000 00000000    ................
```

- Go to https://builder.dontvacuum.me/password.php
  - Paste the string there and get the root password

Phase 2: Step 6

- Connect to UART again (if you disconnected before)
- Type “run setargs_mmc boot_normal”
  - Your device will now boot into linux



```
rockrobo login: █
```

  - Use user “root” and the previously calculated root password
- After successful login: type “iptables -F”

  (if you are holding the cables, you may release them now)

- If you have soldered the UART cables, you may continue over serial, otherwise you can now connect via Wi-Fi and continue over SSH
- DO NOT RESTART/POWER OFF THE DEVICE

Phase 2: Step 7

Commands are in the description!

Paste works often over right-mouse click

- Make SSH access permanent

```
“sed -i -e '/    iptables -I INPUT -j DROP -p tcp --dport 22/s/^/#/g'
/opt/rockrobo/watchdog/rrwatchdoge.conf”
“sed -i -E 's/dport 22/dport 29/g' /opt/rockrobo/watchdog/WatchDoge”
“sed -i -E 's/dport 22/dport 29/g' /opt/rockrobo/rrlog/rrlogd”
```

- Patch recovery (so that SSH survives factory resets)

```
“mkdir /mnt/recovery”
“mount /dev/mmcblk0p7 /mnt/recovery”
“sed -i -e '/    iptables -I INPUT -j DROP -p tcp --dport 22/s/^/#/g'
/mnt/recovery/opt/rockrobo/watchdog/rrwatchdoge.conf”
“sed -i -E 's/dport 22/dport 29/g' /mnt/recovery/opt/rockrobo/watchdog/WatchDoge”
“sed -i -E 's/dport 22/dport 29/g' /mnt/recovery/opt/rockrobo/rrlog/rrlogd”
“umount /mnt/recovery”
```

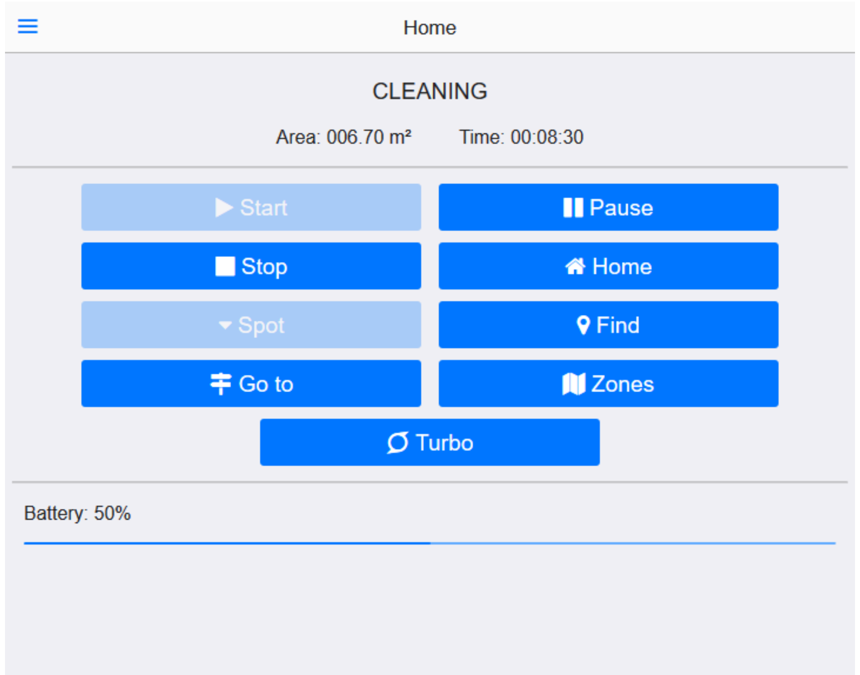
Phase 2: Step 8

You have now permanent root access!

At this point you can reassemble your device again ;)

Phase 3: Custom Firmware

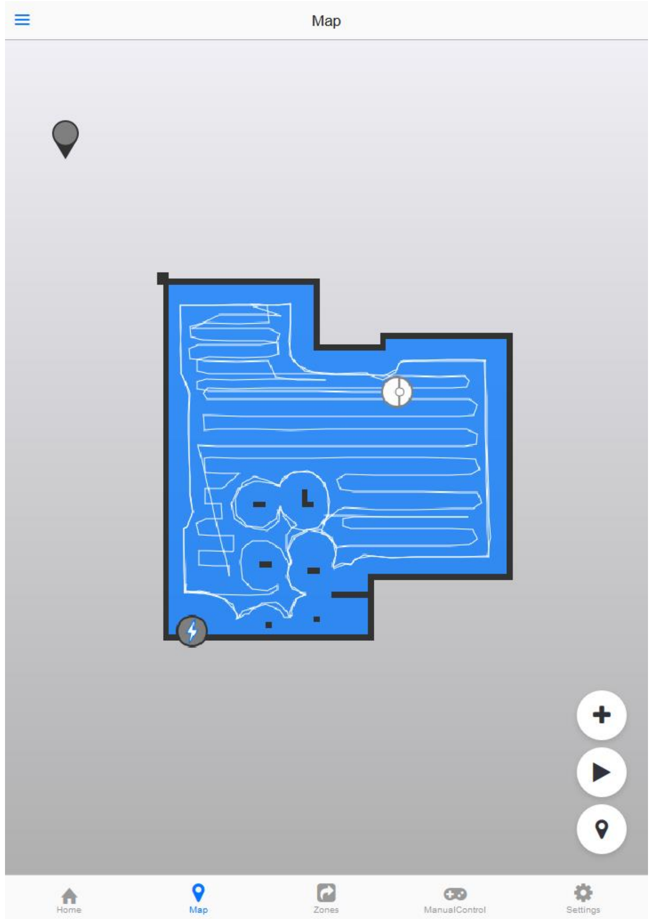
- A custom firmware enables you:
  - Change region of your device to bypass region lock
  - Run Valetudo, disconnect cloud completely
    - Region does not matter in this case



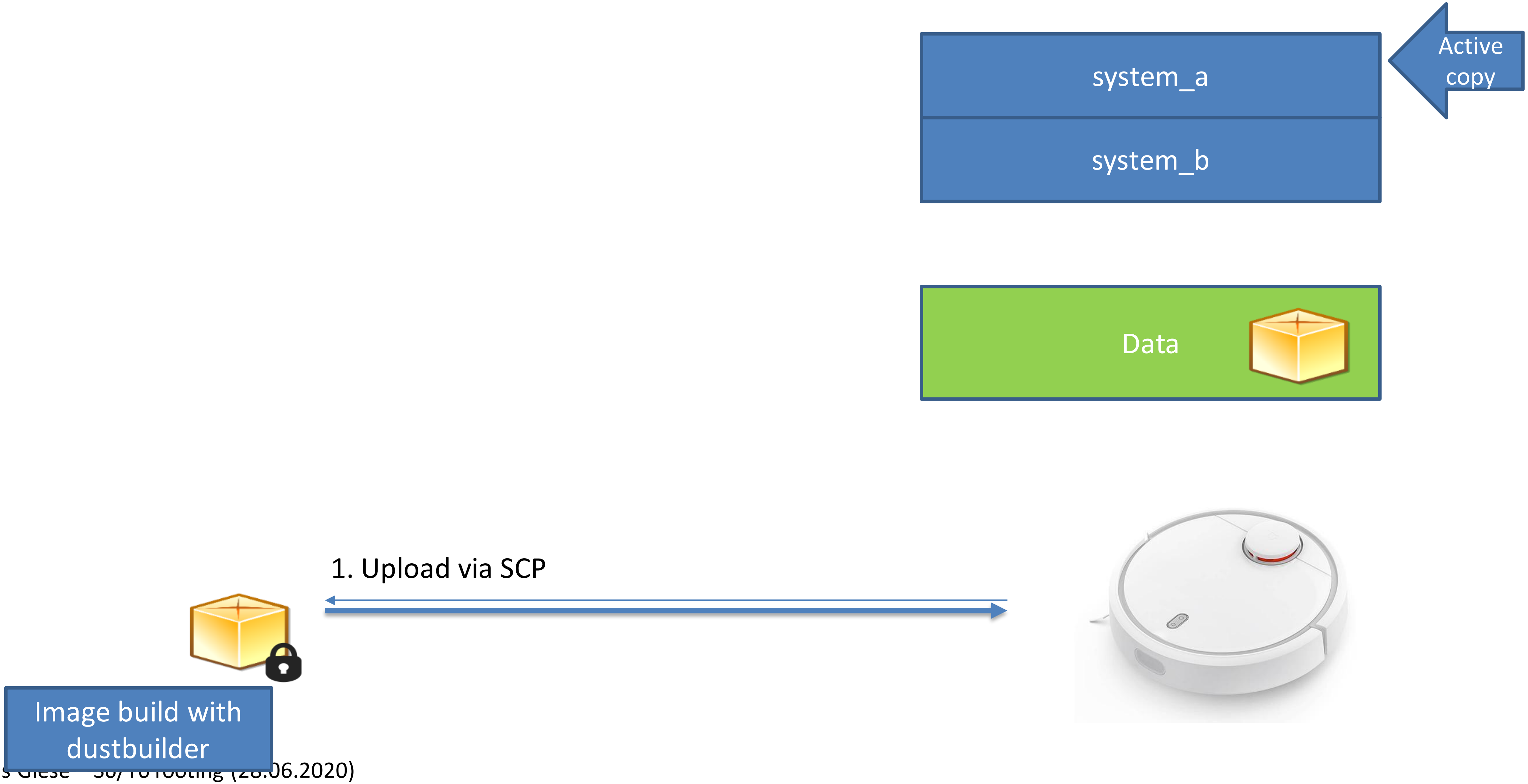
CLEANING

Area: 006.70 m²   Time: 00:08:30

Battery: 50%



Custom firmware installation strategy



system_a

system_b

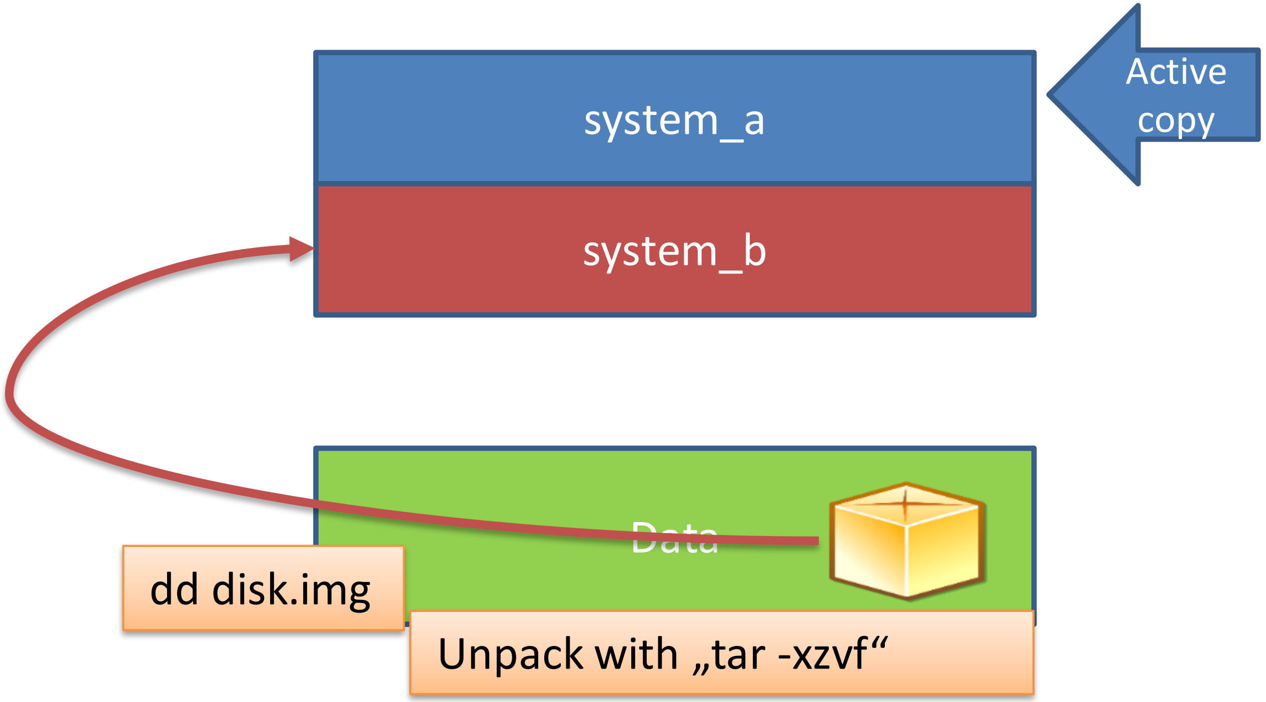
Active copy

Data

1. Upload via SCP

Image build with dustbuilder

Custom firmware installation strategy



Active copy

system_a

system_b

Data

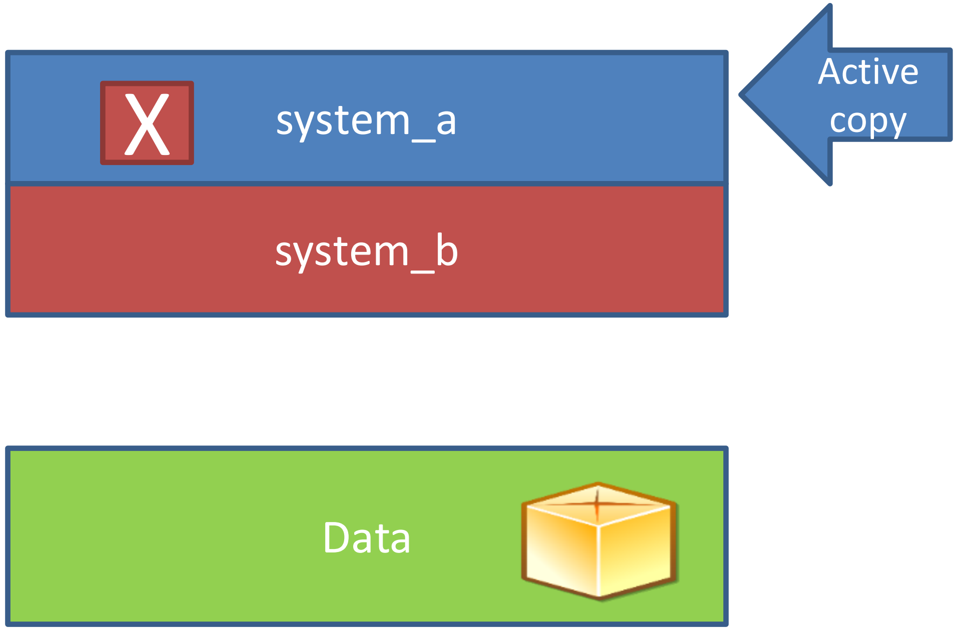
dd disk.img

Unpack with „tar -xzvf“

Custom firmware installation strategy

Mark System_a as „bad“

Mount and update root pw



X

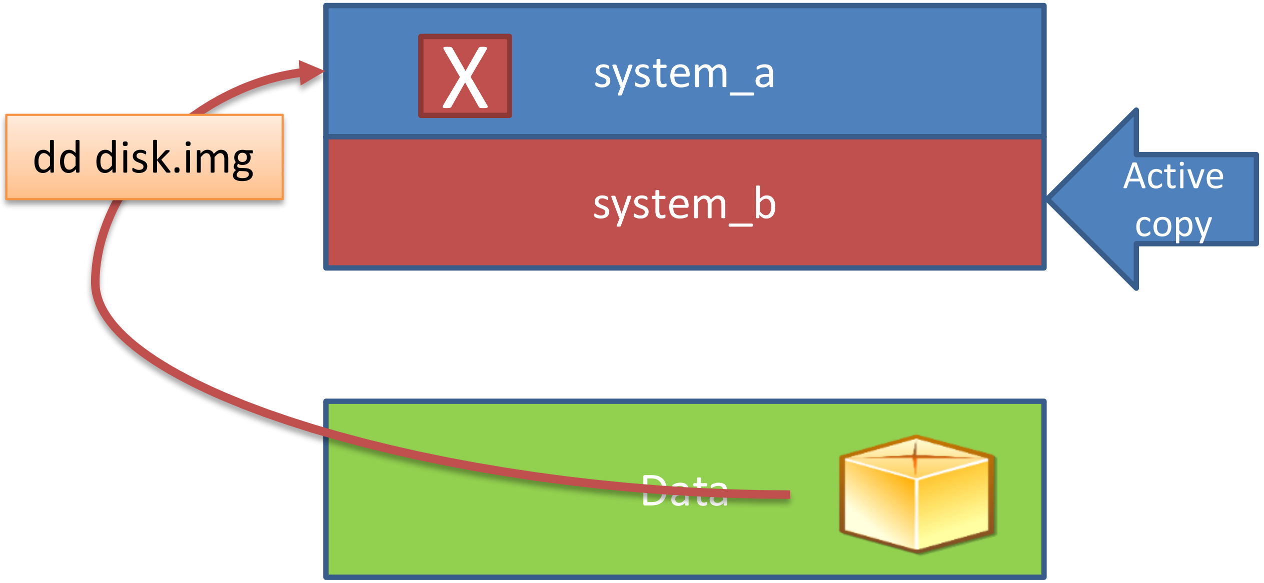
system_a

Active copy

system_b

Data

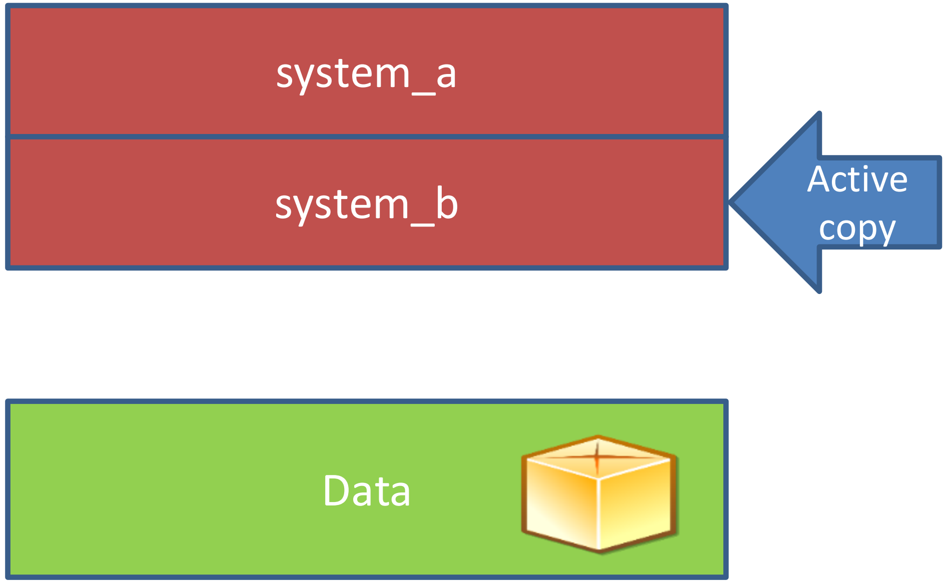
Custom firmware installation strategy



Custom firmware installation strategy

Mark System_a as „good“

Mount and update root pw



system_a

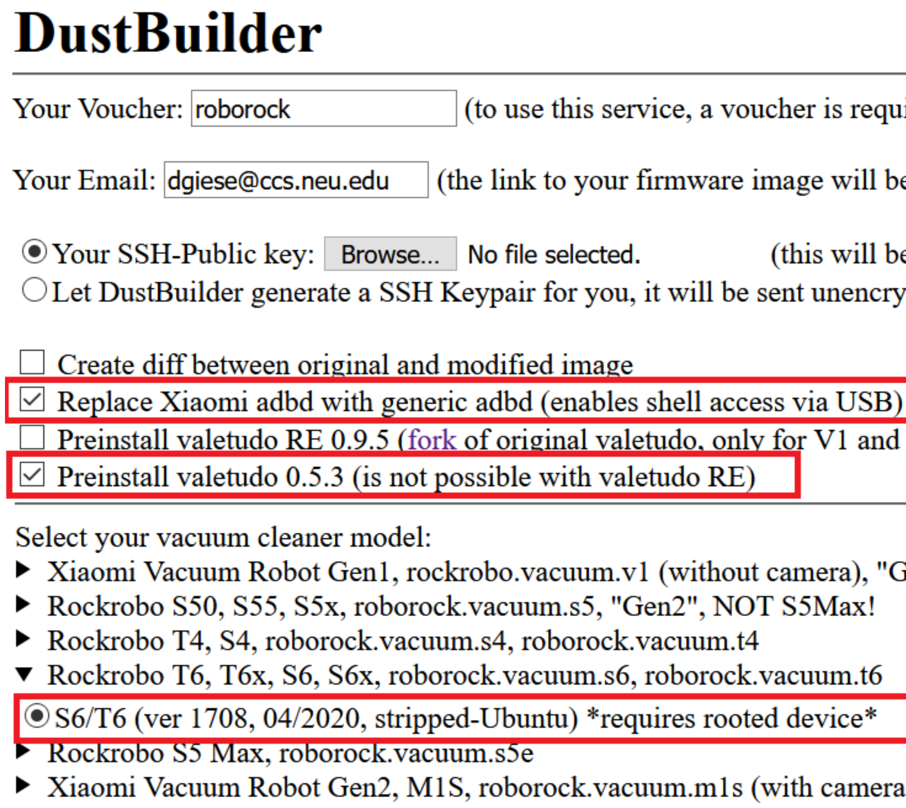
system_b

Active copy

Data

Phase 3: Step 1

- Go to https://builder.dontvacuum.me
  - Build a firmware for your S6/T6
  - Select adb, valetudo 0.5.x and S6/T6
  - Download the firmware



DustBuilder

Your Voucher: roborock (to use this service, a voucher is requi

Your Email: dgiese@ccs.neu.edu (the link to your firmware image will be

● Your SSH-Public key: Browse... No file selected. (this will be
○ Let DustBuilder generate a SSH Keypair for you, it will be sent unencry

☐ Create diff between original and modified image
☑ Replace Xiaomi adbd with generic adbd (enables shell access via USB)
☐ Preinstall valetudo RE 0.9.5 (fork of original valetudo, only for V1 and
☑ Preinstall valetudo 0.5.3 (is not possible with valetudo RE)

Select your vacuum cleaner model:
▸ Xiaomi Vacuum Robot Gen1, rockrobo.vacuum.v1 (without camera), "G
▸ Rockrobo S50, S55, S5x, roborock.vacuum.s5, "Gen2", NOT S5Max!
▸ Rockrobo T4, S4, roborock.vacuum.s4, roborock.vacuum.t4
▾ Rockrobo T6, T6x, S6, S6x, roborock.vacuum.s6, roborock.vacuum.t6
● S6/T6 (ver 1708, 04/2020, stripped-Ubuntu) *requires rooted device*
▸ Rockrobo S5 Max, roborock.vacuum.s5e
▸ Xiaomi Vacuum Robot Gen2, M1S, roborock.vacuum.m1s (with camera

Phase 3: Step 2

- Transfer the firmware package to /mnt/data
  - If using WinSCP make sure to select SCP and not SFTP
  - Under Linux you can use “scp”
  - User “root”, password was calculated in Phase 2 Step 5
- Connect over SSH as user “root”
  - “cd /mnt/data”
  - Run “tar -xzvf v11_001708.pkg”
  - “ls”
  - You should see the files disk.img, install_b.sh and install_a.sh

Phase 3: Step 3

- Patch System_B

  “cd /mnt/data”

  “bash install_b.sh”

  “reboot”

- Verify correct installation
  - Open your browser and access the IP address of the vacuum
  - You should see Valetudo
  - Try to SSH into your vacuum, the root password still works

- Patch System_A (after successful reboot)

  “cd /mnt/data”

  “bash install_a.sh”

  “reboot”

You have now installed a custom firmware ;)

Thank you for watching!

@dgi_DE

Website: dontvacuum.me