

# Hack the Gadget!

Daniel Maslowski



Unter  
ROBVL  
Bedingungen



# Agenda



Hands-on Hardware Hacking

Bringing up your device



Unter  
ROBOT  
Bedingungen



# Hands-on Hardware Hacking



Unter  
ROBVL  
Bedingungen



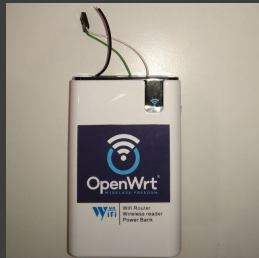
# Things I hack on



Unter  
ROBOT  
Bedingungen



# Things I hack on



- media players
- TV boxes
- NVRs / DVRs
- cameras
- routers
- wireless storages
- laptops, desktops, SBCs



Unter  
ROBOT  
Bedingungen



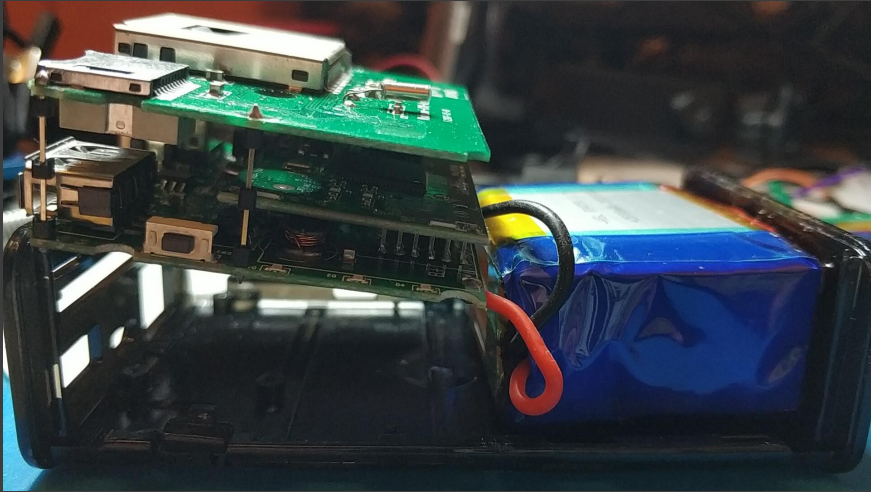
PCB mess



Unter  
ROBVL  
Bedingungen



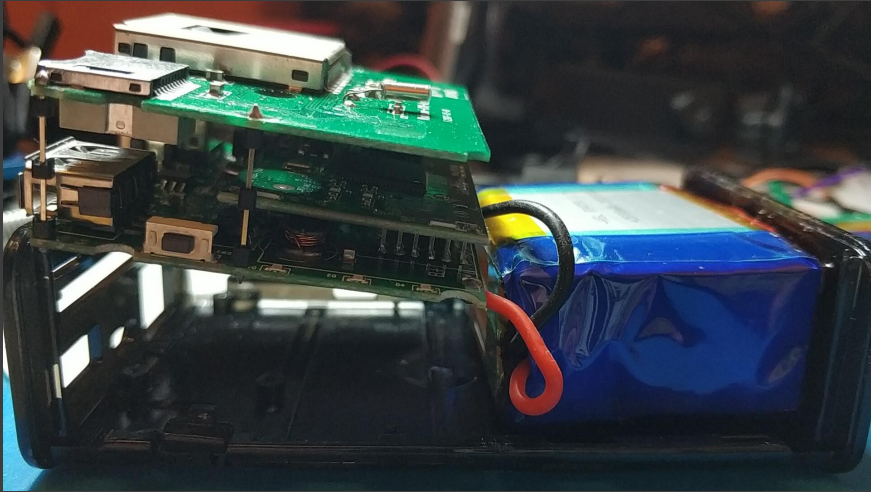
# PCB mess



Unter  
ROBOT  
Bedingungen



# PCB mess



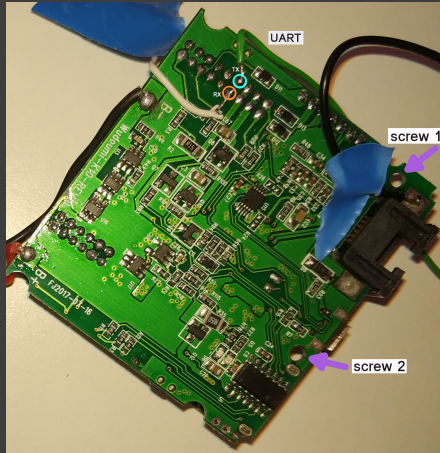
WHERE IS THE UART?!



Unter  
ROBOT  
Bedingungen



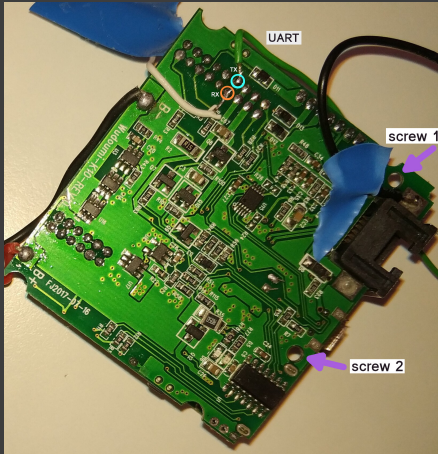
# PCB mess resolved



Unter  
ROBOT  
Bedingungen



# PCB mess resolved



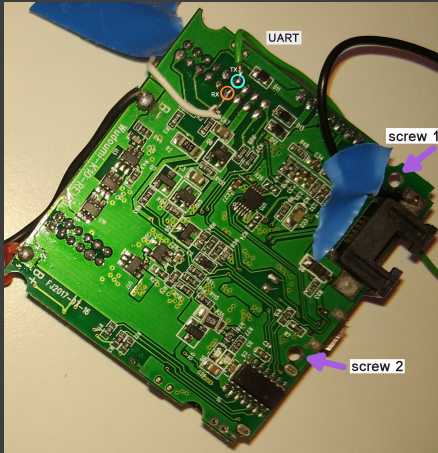
checked pins with multimeter  
▶ measured voltage



Unter  
ROBVL  
Bedingungen



# PCB mess resolved

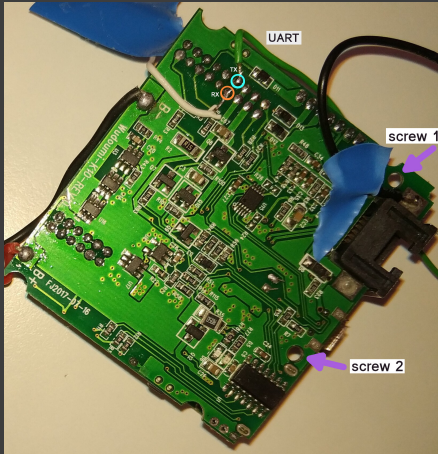


- checked pins with multimeter
  - measured voltage
- all voltages max 3V
  - attach USB serial RX
  - got nothing, no what?





# PCB mess resolved

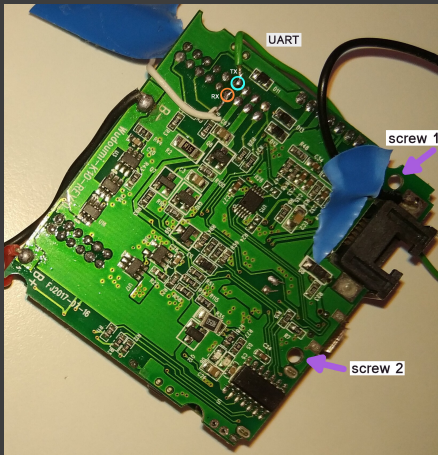


- checked pins with multimeter
  - measured voltage
- all voltages max 3V
  - attach USB serial RX
  - got nothing, no what?
- read about other products
  - OpenWrt forum rocks





# PCB mess resolved



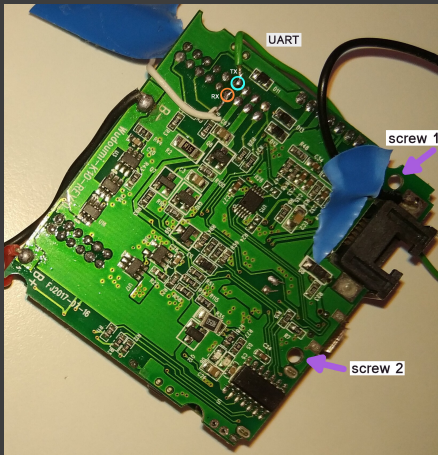
- checked pins with multimeter
  - measured voltage
- all voltages max 3V
  - attach USB serial RX
  - got nothing, no what?
- read about other products
  - OpenWrt forum rocks
- set the baud rate to 57600
  - got output -> TX

```
MtAsicSetPreTbtt(): bss_idx=0, PreTBTT timeout = 0xf0
ap_ftkd> Initialize FT KDP Module...
Main bssid = 00:9a:d5:51:52:46
<==== rt28xx_init, Status=0
@@@ ed_monitor_exit : <===
@@@ ed_monitor_exit : <===
mt7628_set_ed_cca: TURN OFF EDCCA mac 0x10618 = 0xd7083f0f, EDCCA_Status=0
WiFi Startup Cost (ra0): 3.224s
CTRL-A Z for help | 57600 8N1 | NOR | Minicom 2.8 | VT102 | Offline | ttyACM0
```





# PCB mess resolved



- checked pins with multimeter
  - measured voltage
- all voltages max 3V
  - attach USB serial RX
  - got nothing, no what?
- read about other products
  - OpenWrt forum rocks
- set the baud rate to 57600
  - got output -> TX

```
MtAsicSetPreTbtt(): bss_idx=0, PreTBTT timeout = 0xf0
ap_ftkd> Initialize FT KDP Module...
Main bssid = 00:9a:d5:51:52:46
<==== rt28xx_init, Status=0
@@@ ed_monitor_exit : ===>
@@@ ed_monitor_exit : <===
mt7628_set_ed_cca: TURN OFF EDCCA mac 0x10618 = 0xd7083f0f, EDCCA_Status=0
WiFi Startup Cost (ra0): 3.224s
CTRL-A Z for help | 57600 8N1 | NOR | Minicom 2.8 | VT102 | Offline | ttyACM0
```

- RX is likely nearby
  - poked around -> bingo!
  - typing echoed back





# Car Media Player



Roll over image to zoom in



## Portable Wireless Carplay Android Car Stereo 7 Inch HD Touchscreen Car MP5 Player with Mirrorlink Remote Control FM Radio USB 12 LED Camera

Brand: wepeculior

€113<sup>96</sup>

Prices for items sold by Amazon include VAT. Depending on your delivery address, VAT may vary at Checkout. For other items, please see [details](#).

Brand	Wepeculior
Connectivity technology	Bluetooth, Auxilliary, Wi-Fi, USB
Controller type	Android
Compatible devices	Smartphone, Speaker
Connector Type	USB Typ A, 3,5 mm Klinke
Audio output mode	Stereo
Control method	Touch

### About this item

- **Mirror Link:** This full touch screen car radio supports Mirror Link for iOS and Android smartphones. You can sync maps, movies etc. on the large 7 inch screen. The full touch HD display with a resolution of 1024 x 600 provides you with a clear and responsive viewing experience. Equipped with a remote control, it offers you a more convenient experience.



Unter  
ROBOT  
Bedingungen



# Car Media Player



Roll over image to zoom in



Portable Wireless Carplay Android Car Stereo 7 Inch HD Touchscreen Car MP5 Player with Mirrorlink Remote Control FM Radio USB 12 LED Camera

Brand: wepeculior

€113<sup>96</sup>

Prices for items sold by Amazon include VAT. Depending on your delivery address, VAT may vary at Checkout. For other items, please see [details](#).

Brand	Wepeculior
Connectivity technology	Bluetooth, Auxilliary, Wi-Fi, USB
Controller type	Android
Compatible devices	Smartphone, Speaker
Connector Type	USB Typ A, 3,5 mm Klinke
Audio output mode	Stereo
Control method	Touch

#### About this item

- Mirror Link: This full touch screen car radio supports Mirror Link for iOS and Android smartphones. You can sync maps, movies etc. on the large 7 inch screen. The full touch HD display with a resolution of 1024 x 600 provides you with a clear and responsive viewing experience. Equipped with a remote control, it offers you a more convenient experience.

## Product details



CPU F133



1 Gbit DRAM

memory.



Memory

None



1024 x 600

screen

resolution.



7 inch HD


screen size



Unter  
ROBVI  
Bedingungen



# Car Media Player



Portable Wireless Carplay Android Car Stereo 7 Inch HD Touchscreen Car MP5 Player with Mirrorlink Remote Control FM Radio USB 12 LED Camera  
Brand: wepeculor

€113<sup>96</sup>

Prices for items sold by Amazon include VAT. Depending on your delivery address, VAT may vary at Checkout. For other items, please see [details](#).

Brand	Wepeculor
Connectivity technology	Bluetooth, Auxiliary, Wi-Fi, USB
Controller type	Android
Compatible devices	Smartphone, Speaker
Connector Type	USB Typ A, 3,5 mm Klinke
Audio output mode	Stereo
Control method	Touch

**About this item**

- Mirror Link: This full touch screen car radio supports Mirror Link for iOS and Android smartphones. You can sync maps, movies etc. on the large 7 inch screen. The full touch HD display with a resolution of 1024 x 600 provides you with a clear and responsive viewing experience. Equipped with a remote control, it offers you a more convenient experience.

Roll over image to zoom in

## Product details

- CPU F133
- 1 Gbit DRAM memory.
- Memory None
- 1024 x 600 screen resolution.
- 7 inch HD screen size

Some of those details are lies: the F133 (Allwinner SoC aka D1s) only has 512 Mbit DDR2 DRAM in-package, or 64MiB. Is 1024 x 600 really HD? ...

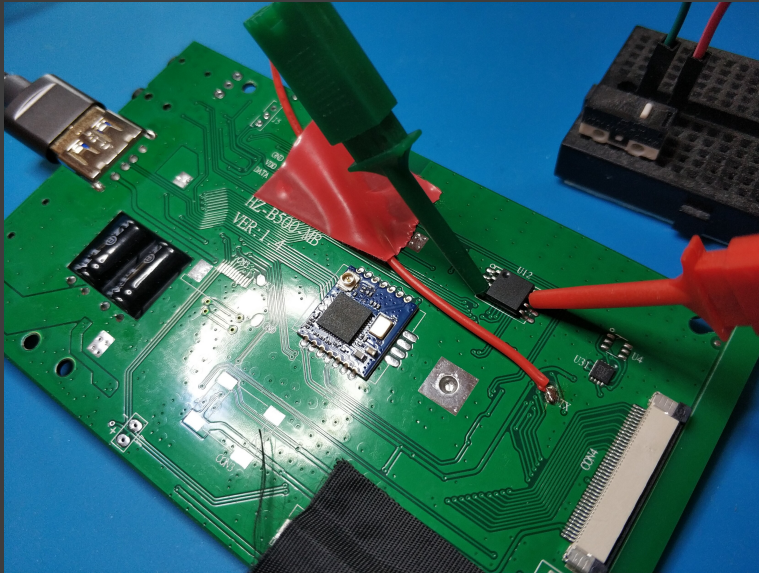
<https://www.amazon.de/-/en/Portable-Wireless-Carplay-Touchscreen-Mirrorlink/dp/B0C23SNRTC>



Unter  
ROBOT  
Bedingungen



# Probes and wires





# LCD bare metal demo



<https://github.com/orangecms/d1rgb/tree/cmp-hack>

(forked from <https://github.com/adamgreig/d1rgb>)



Unter  
ROBVT  
Bedingungen



# Development setup



Unter  
ROBOTik  
Bedingungen



# Development setup



Unter  
LABOR  
Bedingungen



# Projects focusing on products



Unter  
ROBOT  
Bedingungen



# Projects focusing on products

OpenWrt, pfSense/OPNsense



routers, network gear, WiFi



excellent OpenWrt wiki



Unter  
ROBVL  
Bedingungen



# Projects focusing on products

## OpenWrt, pfSense/OPNsense



routers, network gear, WiFi



excellent OpenWrt wiki

## OpenIPC



(network) cameras



lots of tooling, tutorials, etc



Unter  
ROBOT  
Bedingungen



# Projects focusing on products

## OpenWrt, pfSense/OPNsense



routers, network gear, WiFi



excellent OpenWrt wiki

## OpenIPC



(network) cameras



lots of tooling, tutorials, etc

## OpenBMC, u-bmc



board management controllers



remote OOB management



Unter  
ROBVL  
Bedingungen



# Projects focusing on products

## OpenWrt, pfSense/OPNsense



routers, network gear, WiFi



excellent OpenWrt wiki

## OpenIPC



(network) cameras



lots of tooling, tutorials, etc

## OpenBMC, u-bmc



board management controllers



remote OOB management

## Start a new one - pick u-root and cpu



<https://github.com/u-root/cpu>



<https://github.com/orangecms/arm-cpu>



<https://github.com/u-root/sidecore>





Bringing up your device



Unter  
ROBOT  
Bedingungen



# Firmware vs OS

## U-Boot

- 🐼 configs in `configs/` - they determine the ARCH themselves
- 🐼 device trees in `arch/$ARCH/dts/`
- 🐼 boards in `board/$VENDOR/` - emphasis on SoC, but not consistently

## Linux

- 🐼 configs in `arch/$ARCH/configs/` - \$ARCH must be provided by user
- 🐼 device trees in `arch/$ARCH/boot/dts/[$VENDOR/]`
- 🐼 board is described by firmware *and* own DTB, merged at runtime





# Hardware Description: Device Tree



Unter  
ROBVL  
Bedingungen



# Hardware Description: Device Tree



**devicetree**  
.org

Standardization in progress; current version: 0.4



Unter  
ROBOT  
Bedingungen



# Hardware Description: Device Tree



**devicetree**  
.org

Standardization in progress; current version: 0.4

A DT must have a memory node - provided by firmware, usually.

<https://devicetree-specification.readthedocs.io/en/latest/chapter3-devicenodes.html>



Unter  
ROBOT  
Bedingungen



# Hardware Description: Device Tree



**devicetree**  
.org

Standardization in progress; current version: 0.4

A DT must have a memory node - provided by firmware, usually.

<https://devicetree-specification.readthedocs.io/en/latest/chapter3-devicenodes.html>

Arm timer frequency must also be in DT, as I learned.

I simply put them in the kernel's DT, so I can do firmware without DT augmentation.

<https://lore.kernel.org/linux-arm-kernel/25965de3-cc82-7fe6-6b3d-5754c329ac07@suse.de/>



Unter  
ROBOT  
Bedingungen



# Kernel hacking



Unter  
ROBVL  
Bedingungen



# Kernel hacking

## Early output

- 🧠 find **indicators** to see how far you get
- 🧠 in early asm, direct MMIO on serial for single char output
- 🧠 `arch/$ARCH/kernel/head.S`
  - ▶ be careful with registers - they have special meaning in early asm
  - ▶ doing a `b1` will mess up the return address!
  - ▶ `debug.S` *really handy*, can print 2,4,8-digit hex values and ASCII





# Kernel hacking

## Early output

- 🧠 find **indicators** to see how far you get
- 🧠 in early asm, direct MMIO on serial for single char output
- 🧠 `arch/$ARCH/kernel/head.S`
  - ▶ be careful with registers - they have special meaning in early asm
  - ▶ doing a `bl` will mess up the return address!
  - ▶ `debug.S` *really handy*, can print 2,4,8-digit hex values and ASCII

## Logs, logs, logs!

- 🧠 earlycon, figure it out <https://falstaff.agner.ch/2015/10/17/linux-earlyprintkearlycon-support-on-arm/>
  - ▶ for 8250/16550: `earlycon=uart,mmio32,$UARTBASE_ADDR`
- 🧠 `loglevel=8,initcall_debug`, kernel config options





# A little userland

```
build-arm32.sh
```

```
#!/bin/sh
```

```
set -e
```

```
export GOARCH=arm
```

```
CPIO="/tmp/u-root-$GOARCH.cpio"
```

```
# build a root fs using the embedded template
```

```
go run . -uroot-source . -o "$CPIO" embedded
```

```
# https://github.com/u-root/u-root/#compression
```

```
xz --check=crc32 -9 --lzma2=dict=1MiB --stdout "$CPIO" | \
```

```
dd conv=sync bs=512 of="$CPIO.xz"
```



Unter  
ROBVI  
Bedingungen



# Getting stuck



Unter  
ROBVL  
Bedingungen



# Getting stuck

```
/# cat /sys/kernel/debug/devices_deferred  
1c50000.ethernet      platform: wait for supplier  
                       /soc/i2c@1c2ac00/pmic@34/regulators/dc1sw
```



Unter  
ROBOT  
Bedingungen



# Getting stuck

```
/# cat /sys/kernel/debug/devices_deferred  
1c50000.ethernet      platform: wait for supplier  
                      /soc/i2c@1c2ac00/pmic@34/regulators/dc1sw
```

In this case, I missed describing the power supply.



Unter  
ROBOT  
Bedingungen



# Getting stuck

```
/# cat /sys/kernel/debug/devices_deferred  
1c50000.ethernet      platform: wait for supplier  
                        /soc/i2c@1c2ac00/pmic@34/regulators/dc1sw
```

In this case, I missed describing the power supply.

It was a wrong guess anyway. More later.



Unter  
ROBOT  
Bedingungen



Device Tree is nice, but...



Unter  
ROBOT  
Bedingungen



Device Tree is nice, but...

The DT *could* be checked at build time!



Unter  
ROBVL  
Bedingungen



# Device Tree is nice, but...

The DT *could be checked at build time!*

Unless... the firmware is expected to provide (part of) it.

How about fallbacks?



Unter  
ROBOT  
Bedingungen



# Device Tree is nice, but...

The DT *could be checked at build time!*

Unless... the firmware is expected to provide (part of) it.

How about fallbacks?

## Solving Devicetree Issues, part 3.0

Frank Rowand at ELCE 2016

<https://www.youtube.com/watch?v=BDS6Hydtsx8>

[https://www.elinux.org/images/archive/e/e5/20161014033717!Dt\\_deb  
ugging\\_part\\_3.pdf](https://www.elinux.org/images/archive/e/e5/20161014033717!Dt_debugging_part_3.pdf)



Unter  
ROBOT  
Bedingungen



# Device Tree is nice, but...

The DT *could be checked at build time!*

Unless... the firmware is expected to provide (part of) it.

How about fallbacks?

## Solving Devicetree Issues, part 3.0

Frank Rowand at ELCE 2016

<https://www.youtube.com/watch?v=BDS6Hydtsx8>

[https://www.elinux.org/images/archive/e/e5/20161014033717!Dt\\_deb  
ugging\\_part\\_3.pdf](https://www.elinux.org/images/archive/e/e5/20161014033717!Dt_debugging_part_3.pdf)

Some great ideas which never landed upstream. Anyone?



Unter  
ROBOT  
Bedingungen



# Living the lie



Unter  
ROBOT  
Bedingungen



# Living the lie

Device Tree is a tree - but your hardware is **not**!



Unter  
ROBVL  
Bedingungen



# Living the lie

Device Tree is a tree - but your hardware is **not**!

Clocks, interrupts, GPIO pins, power supplies are all across.



Unter  
ROBVL  
Bedingungen



# Living the lie

Device Tree is a tree - but your hardware is **not**!

Clocks, interrupts, GPIO pins, power supplies are all across.

Some entries in DT are just loose strings or references, e.g., phy-supply.



Unter  
ROBOT  
Bedingungen



# Living the lie

Device Tree is a tree - but your hardware is **not**!

Clocks, interrupts, GPIO pins, power supplies are all across.

Some entries in DT are just loose strings or references, e.g., phy-supply.

[https://elinux.org/Device\\_Tree\\_Mysteries#Phandle](https://elinux.org/Device_Tree_Mysteries#Phandle)



Unter  
ROBOT  
Bedingungen



# Living the lie

Device Tree is a tree - but your hardware is **not**!

Clocks, interrupts, GPIO pins, power supplies are all across.

Some entries in DT are just loose strings or references, e.g., phy-supply.

[https://elinux.org/Device\\_Tree\\_Mysteries#Phandle](https://elinux.org/Device_Tree_Mysteries#Phandle)

I've started building a device tree visualizer! :-)



Unter  
ROBOT  
Bedingungen









Unter  
LABOR  
Bedingungen



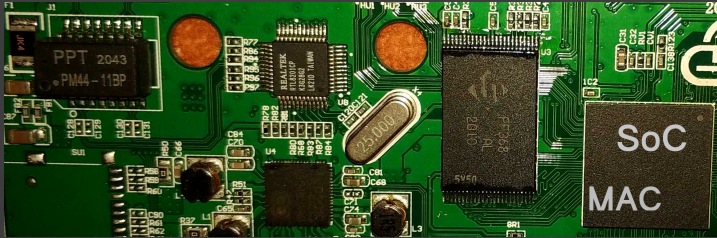
# Tracing Components



Unter  
ROBOTik  
Bedingungen



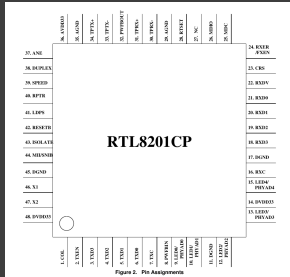
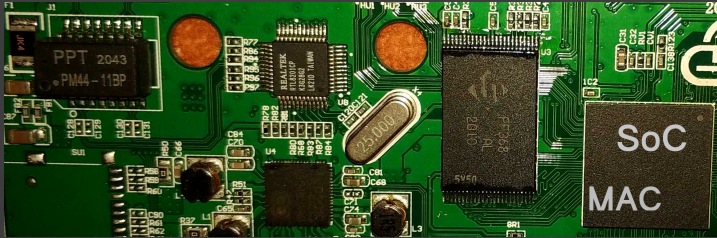
# Tracing Components



Unter  
ROBOT  
Bedingungen



# Tracing Components



Unter  
ROBOT  
Bedingungen



# Tracing Components

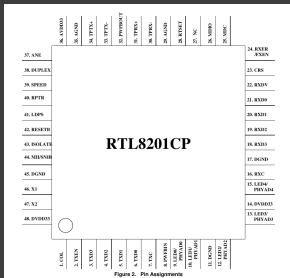
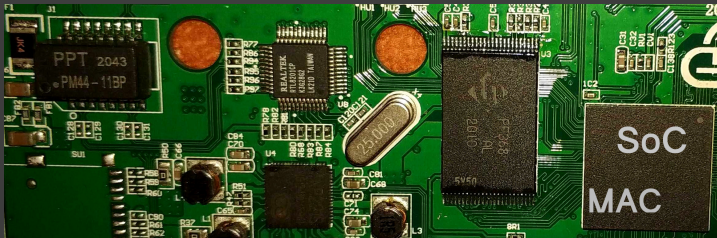
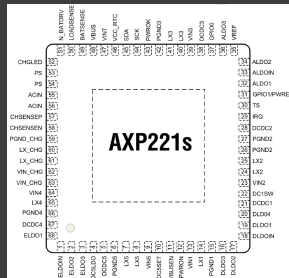


Figure 2. Pin Assignments



SoC platforms may use PMICs to supply power to components.





Thank you! :)



Unter  
ROBOTik  
Bedingungen



# Follow Me



Daniel Maslowski

<https://github.com/orangecms>

<https://twitter.com/orangecms>

<https://mastodon.social/@cyrevolt>

<https://youtube.com/@cyrevolt>

<https://twitch.tv/cyrevolt>

<https://metaspora.org/hack-the-gadget-labortage2023.pdf>

License: CC BY 4.0 <https://creativecommons.org/licenses/by/4.0/>



Unter  
ROBOT  
Bedingungen