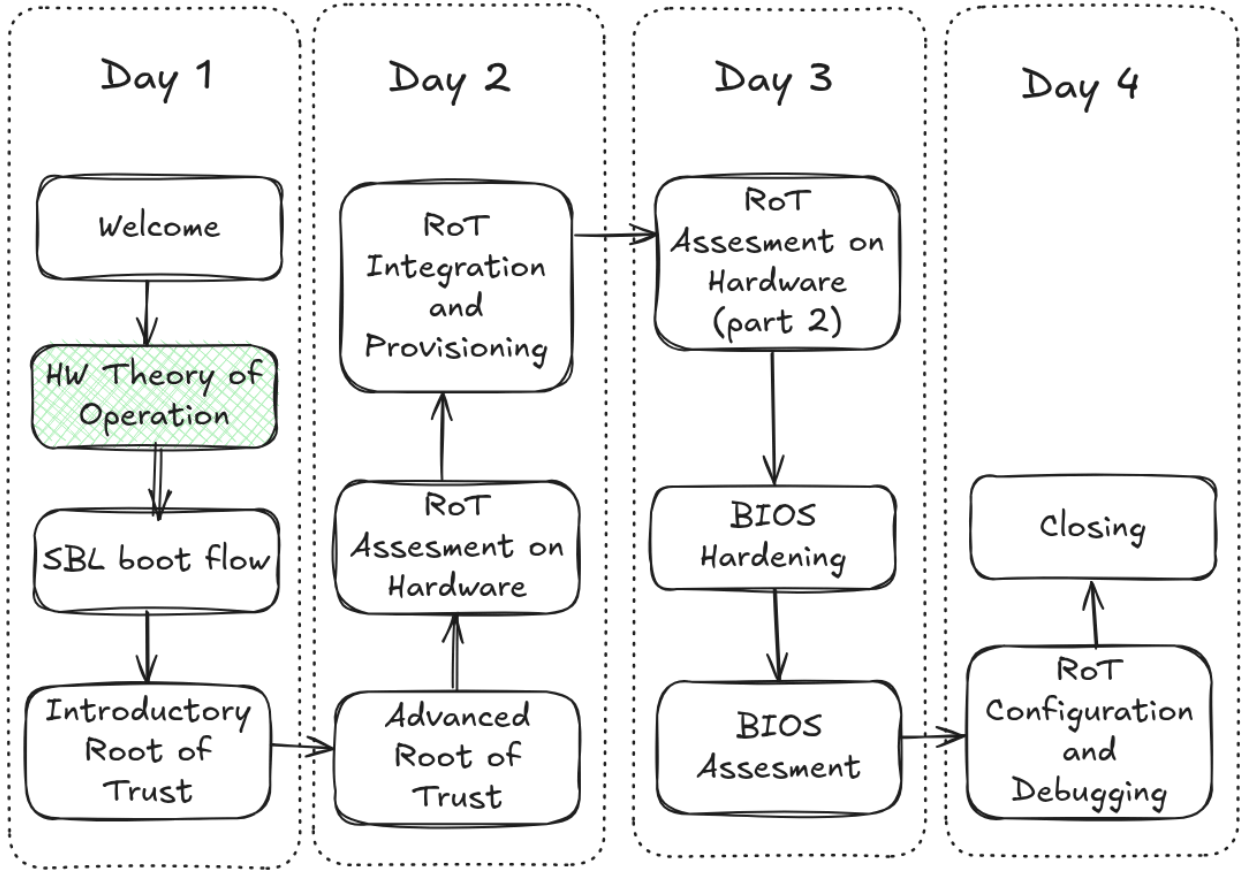
DS09SBL: Dasharo TrustRoot Training

Hardkernel Odroid-H4+

Hardware Theory of Operation

Where we are in the course



Goals of the Presentation

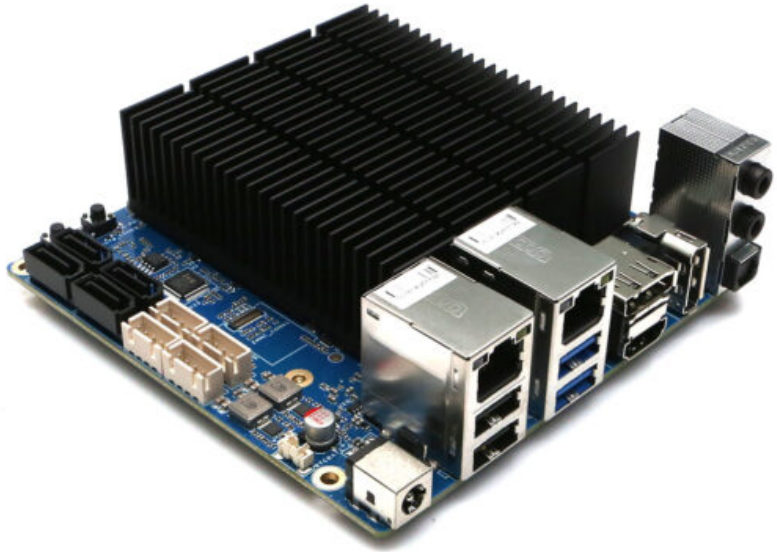
In this presentation, we aim to:

- Demonstrate how to connect, power on, and confirm successful boot of Odroid-H4+ (aka *theory of operation*).
- Explain briefly *recovery methods* for broken firmware on Odroid-H4+ and how to apply them.

These skills will help us safely and with confidence perform experiments related to Intel Root of Trust and Slim Bootloader.

Requirements

- Basic understanding of embedded hardware.



ODROID-H4+

- **CPU:** Intel® Processor N97
- **CPU Architecture:** *Alder Lake-N*
- **Cores/Threads:** 4C4T
- **TDP:** 12W
- **Memory:** DDR5 4800MT/s (max 48GB)
- **GPU:** Intel UHD Graphics 500/1200MHz (base/burst)
- **PCIe/NVMe:** Gen3, 4 Lanes
- **USB:** 2x2.0, 2x3.0
- **Ethernet:** 3x2.5GbE
- **SATA:** 4 ports
- **Certification:** FCC/CE/KC/RoHS

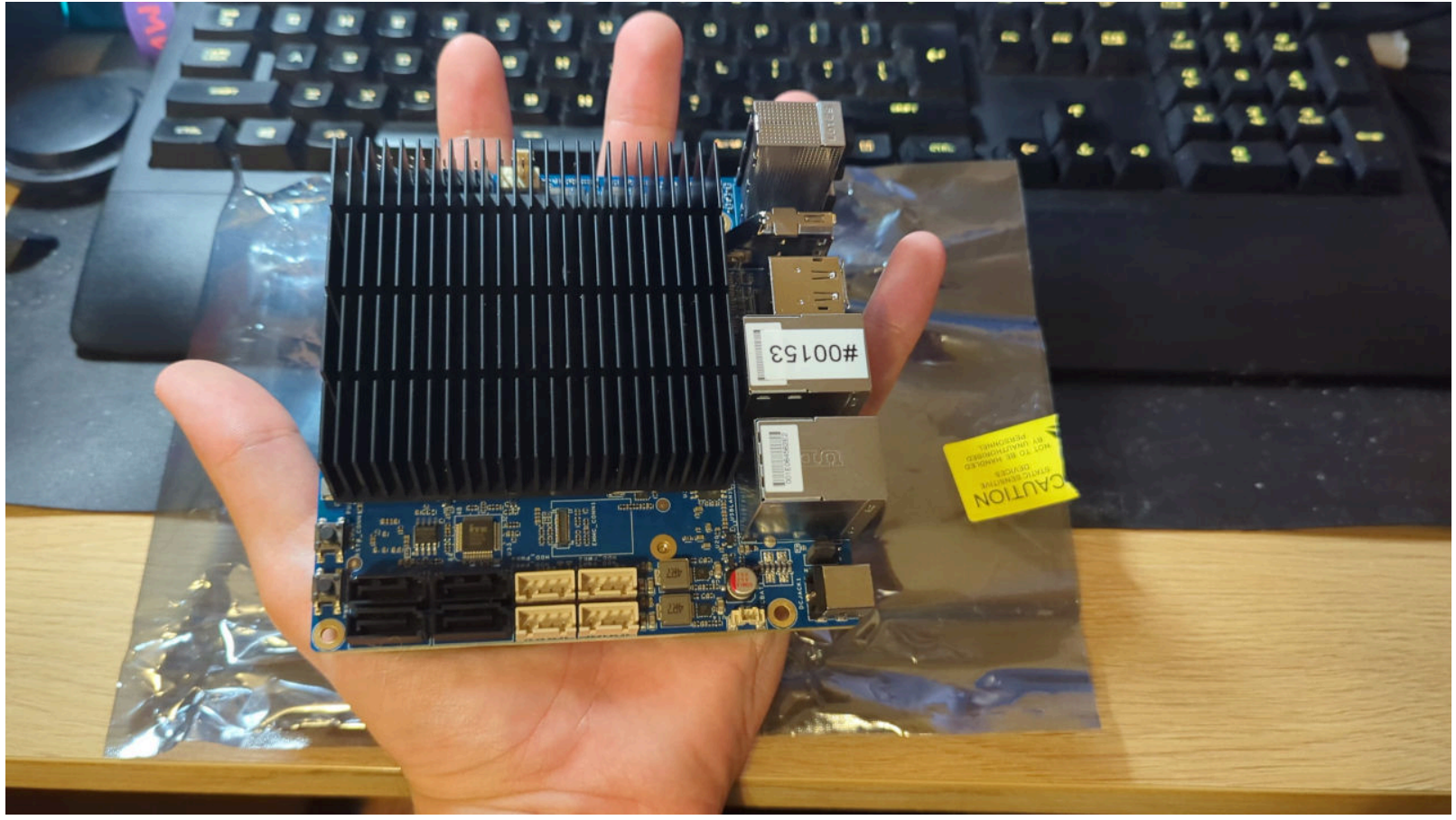
- Hardkernel is a well-known hardware vendor based in South Korea.
- After PC Engines (Swiss router manufacturer that is now defunct), they are likely the only vendor offering x86 hardware with schematics. This is crucial for open-source firmware development and highly valuable for low-level hackers.
- The modern spec can be used for production or homelab workloads once its use as a training platform has been exhausted.

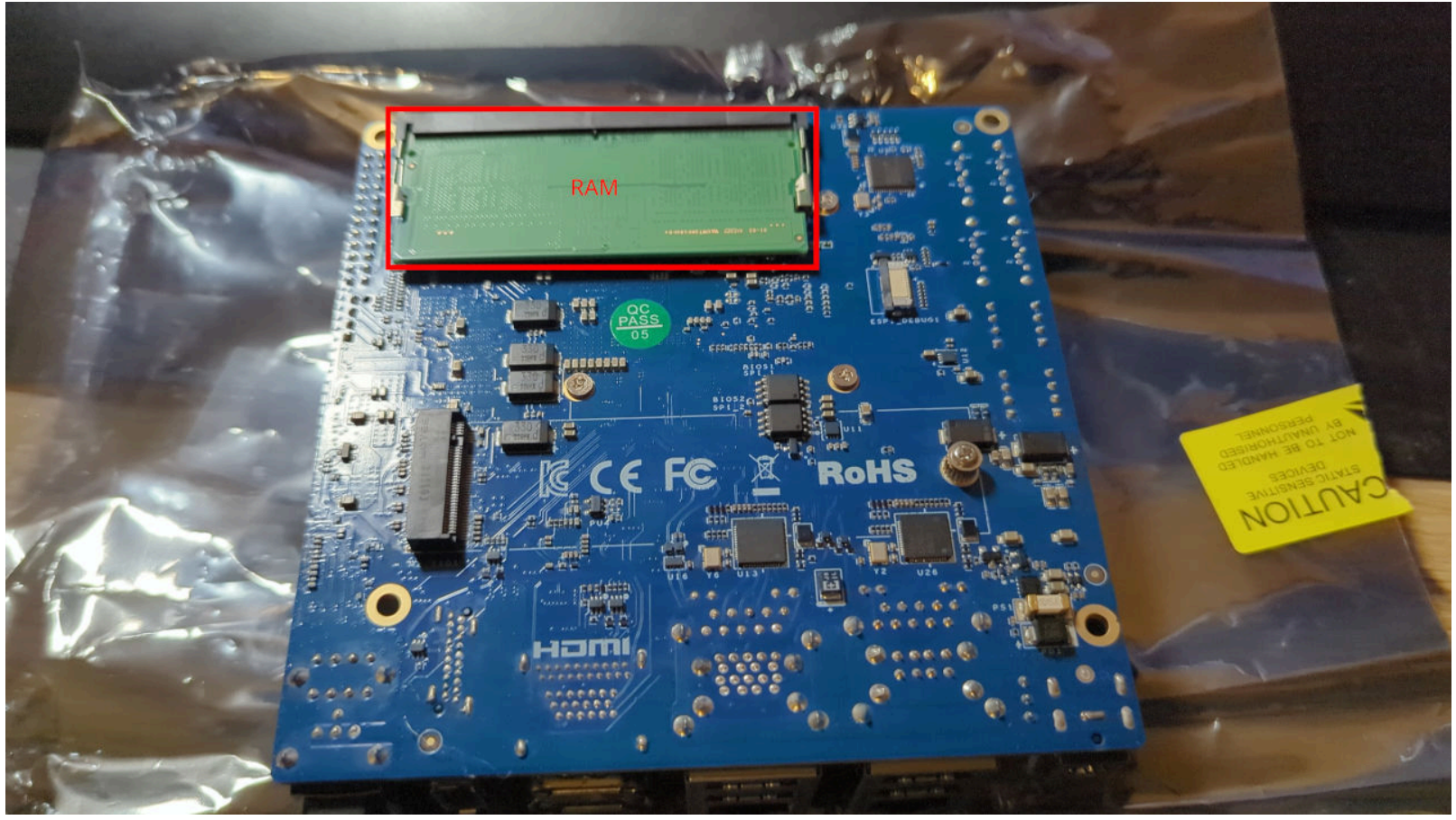
Practice #102 Exercise #0

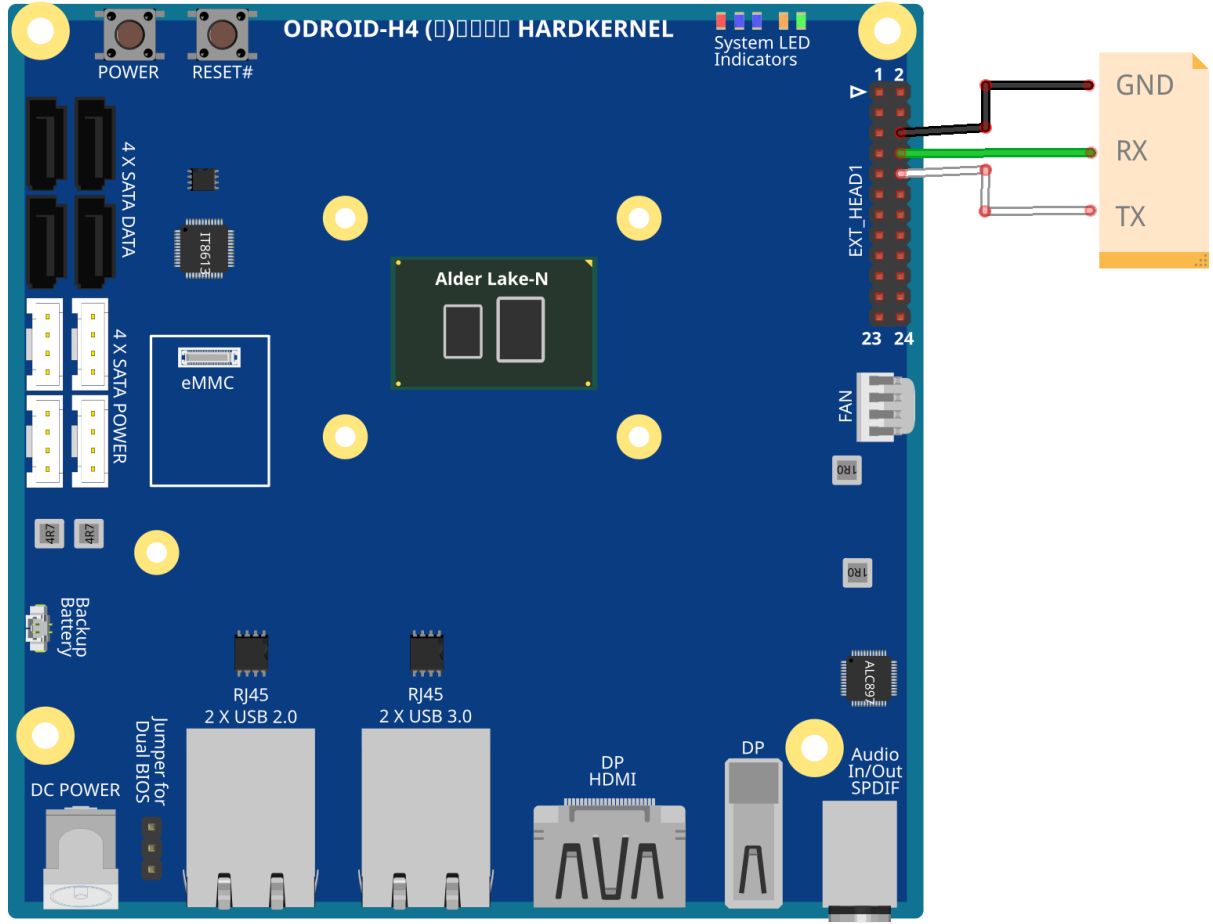

15V 4A
Power Supply
UK


ODROID
HD
HIGH DEFINITION MUL

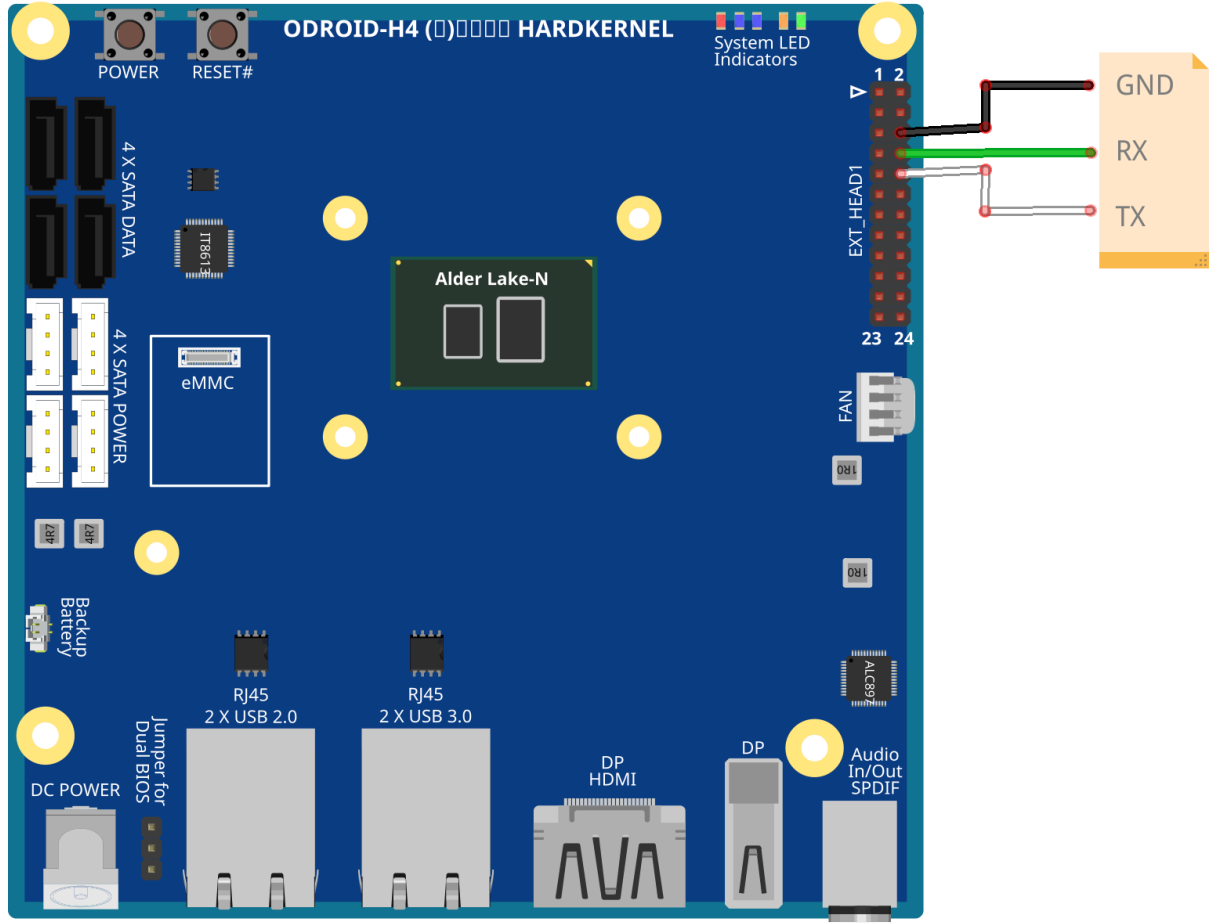

#00153


#00153
CAUTION

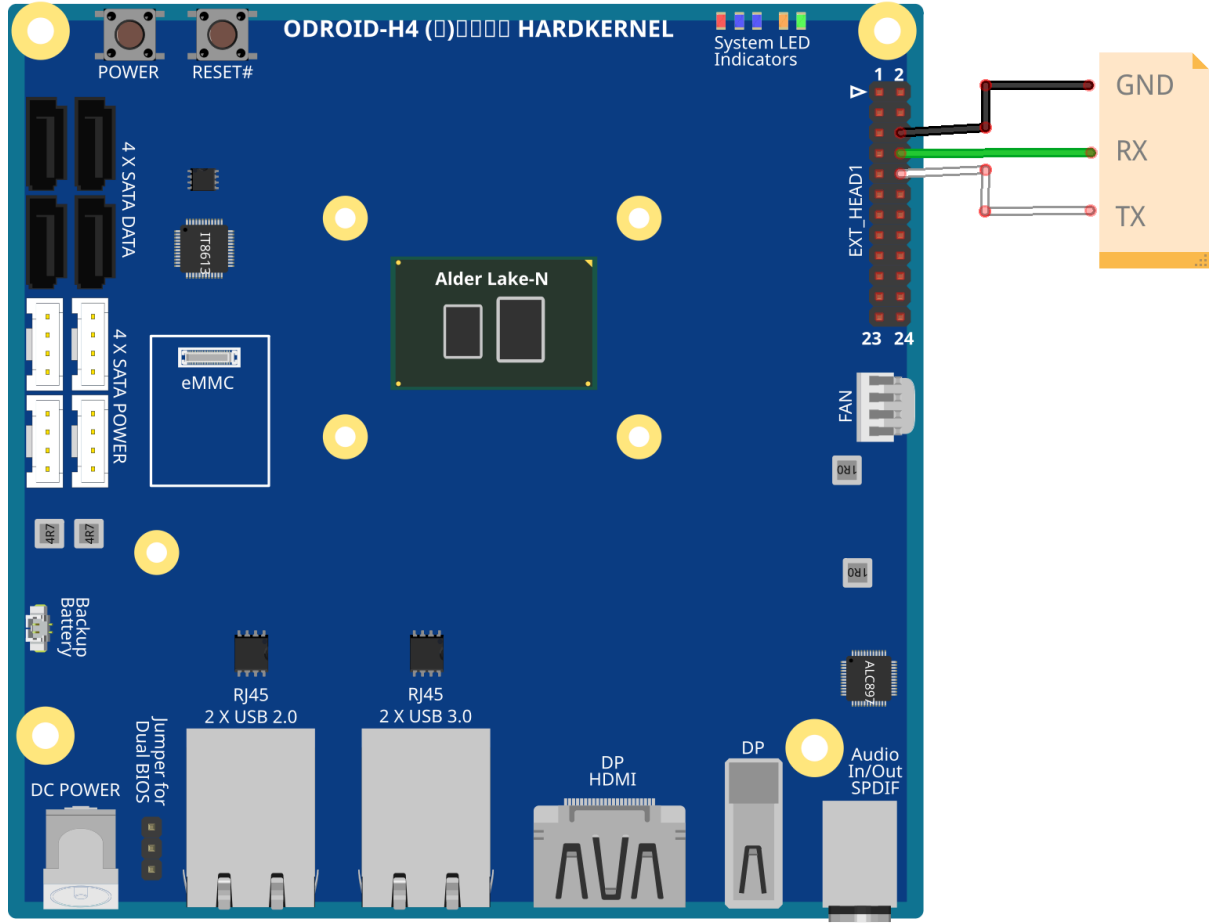

RAM
QC PASS 05
RoHS
HDMI
CAUTION
STATIC SENSITIVE DEVICES
NOT TO BE HANDLED BY UNAUTHORISED PERSONNEL


ODROID-H4 (□)□□□□ HARDKERNEL
POWER
RESET#
System LED Indicators
4 X SATA DATA
4 X SATA POWER
eMMC
Alder Lake-N
EXT_HEAD1
1 2
23 24
GND
RX
TX
FAN
Backup Battery
RJ45
2 X USB 2.0
RJ45
2 X USB 3.0
DP
HDMI
DP
Audio In/Out SPDIF
DC POWER
Jumper for Dual BIOS

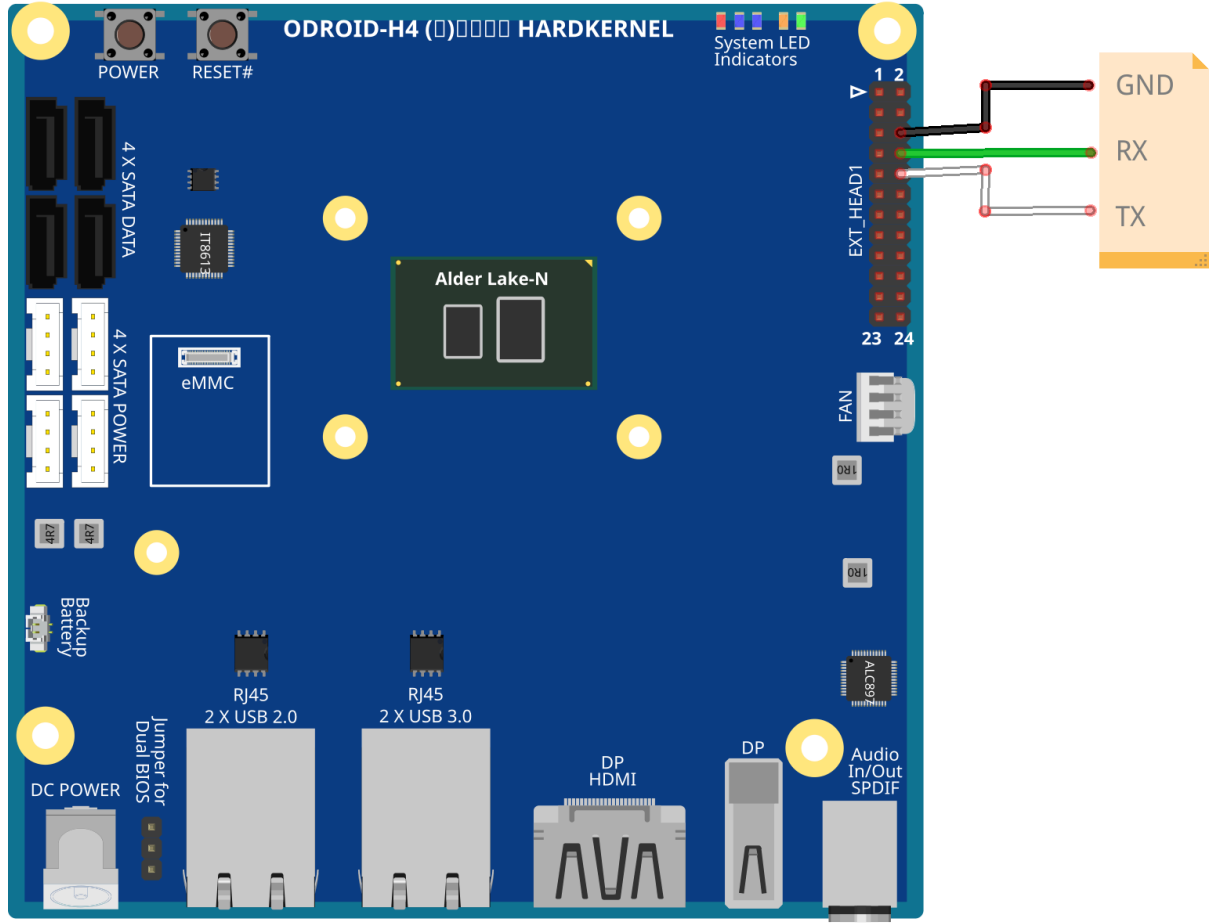
fritzing


ODROID-H4 (□)□□□□ HARDKERNEL
POWER
RESET#
System LED Indicators
4 X SATA DATA
4 X SATA POWER
eMMC
Alder Lake-N
EXT_HEAD1
1 2
23 24
GND
RX
TX
FAN
Backup Battery
RJ45
2 X USB 2.0
RJ45
2 X USB 3.0
DP
HDMI
DP
Audio In/Out SPDIF
DC POWER
Jumper for Dual BIOS

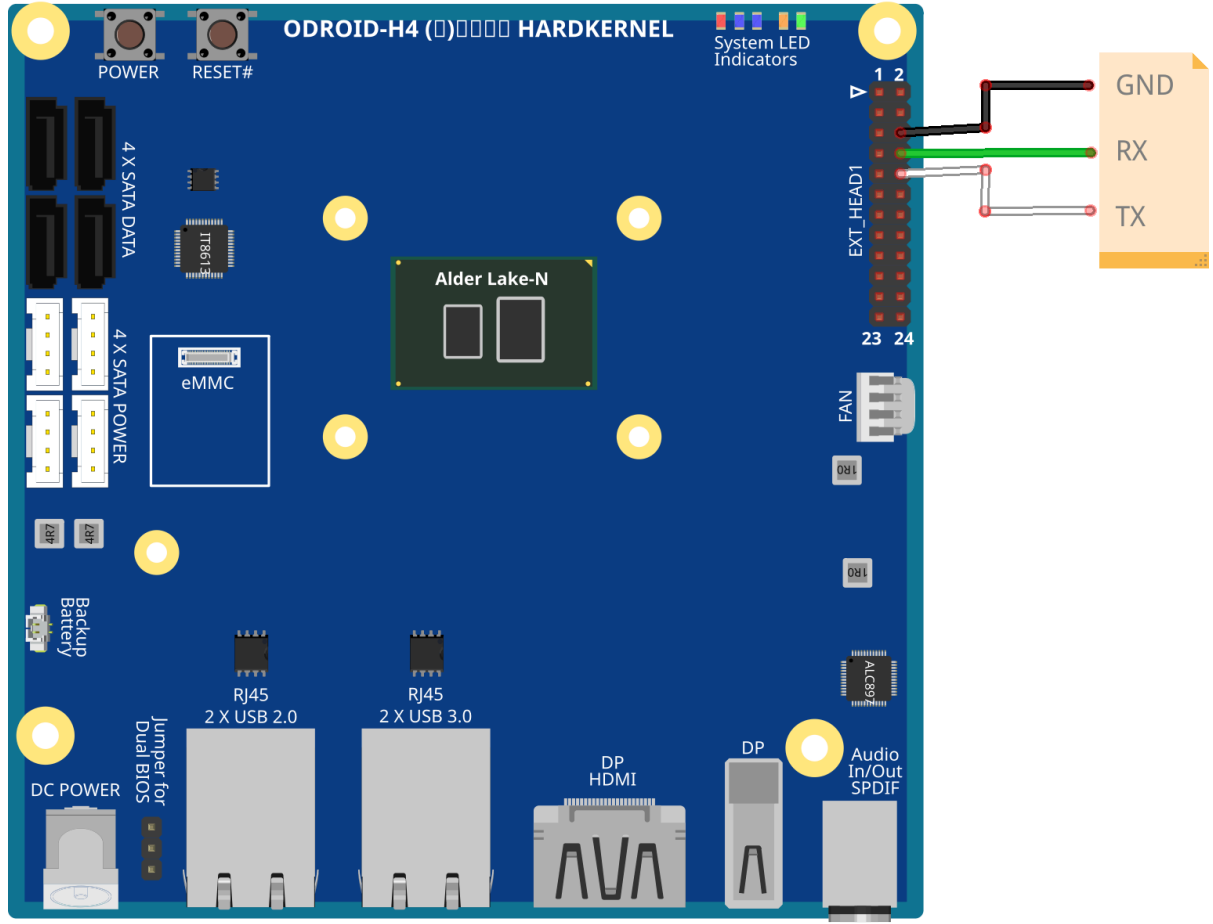
fritzing


ODROID-H4 (□)□□□□ HARDKERNEL
POWER
RESET#
System LED Indicators
4 X SATA DATA
4 X SATA POWER
eMMC
Alder Lake-N
EXT_HEAD1
1 2
23 24
GND
RX
TX
FAN
Backup Battery
RJ45
2 X USB 2.0
RJ45
2 X USB 3.0
DP
HDMI
DP
Audio In/Out SPDIF
DC POWER
Jumper for Dual BIOS

fritzing


ODROID-H4 (□)□□□□ HARDKERNEL
POWER
RESET#
System LED Indicators
4 X SATA DATA
4 X SATA POWER
eMMC
Alder Lake-N
EXT_HEAD1
1 2
23 24
GND
RX
TX
FAN
Backup Battery
RJ45
2 X USB 2.0
RJ45
2 X USB 3.0
DP
HDMI
DP
Audio In/Out SPDIF
DC POWER
Jumper for Dual BIOS

fritzing


ODROID-H4 (□)□□□□ HARDKERNEL
POWER
RESET#
System LED Indicators
4 X SATA DATA
4 X SATA POWER
eMMC
Alder Lake-N
EXT_HEAD1
1 2
23 24
GND
RX
TX
FAN
Backup Battery
RJ45
2 X USB 2.0
RJ45
2 X USB 3.0
DP
HDMI
DP
Audio In/Out SPDIF
DC POWER
Jumper for Dual BIOS

fritzing

Practice #102 Exercise #1

To verify if we receive Odroid-H4+ UEFI BIOS output on our serial let's attach USB-to-TTL serial cable to VM.

```
minicom -D /dev/ttyUSB0 -c on -b 115200 -C odroid-serial.log
```

Options explained:

- `-c on/off` - color. Set to `on` to enable colors in `minicom`
- `b` - specify the baud rate.
- `-C filename` - capture file. Opens a capture file at startup.

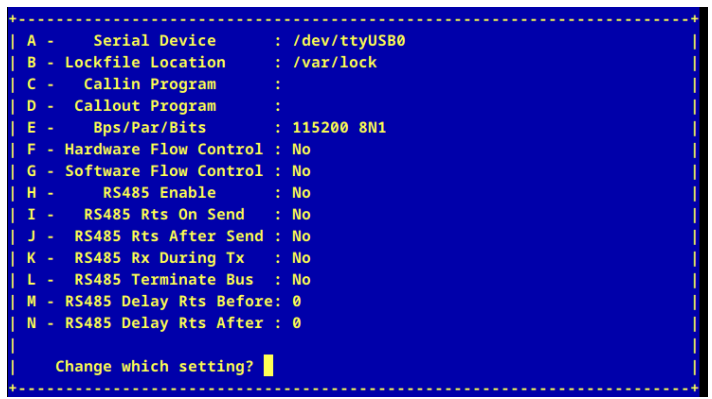
Useful `minicom` shortcuts:

- `Ctrl` + `a` `z` - command summary
- `Ctrl` + `a` `x` - exit `minicom`
- `Ctrl` + `a` `o` - configure `minicom`
- `Ctrl` + `a` `c` - clear screen

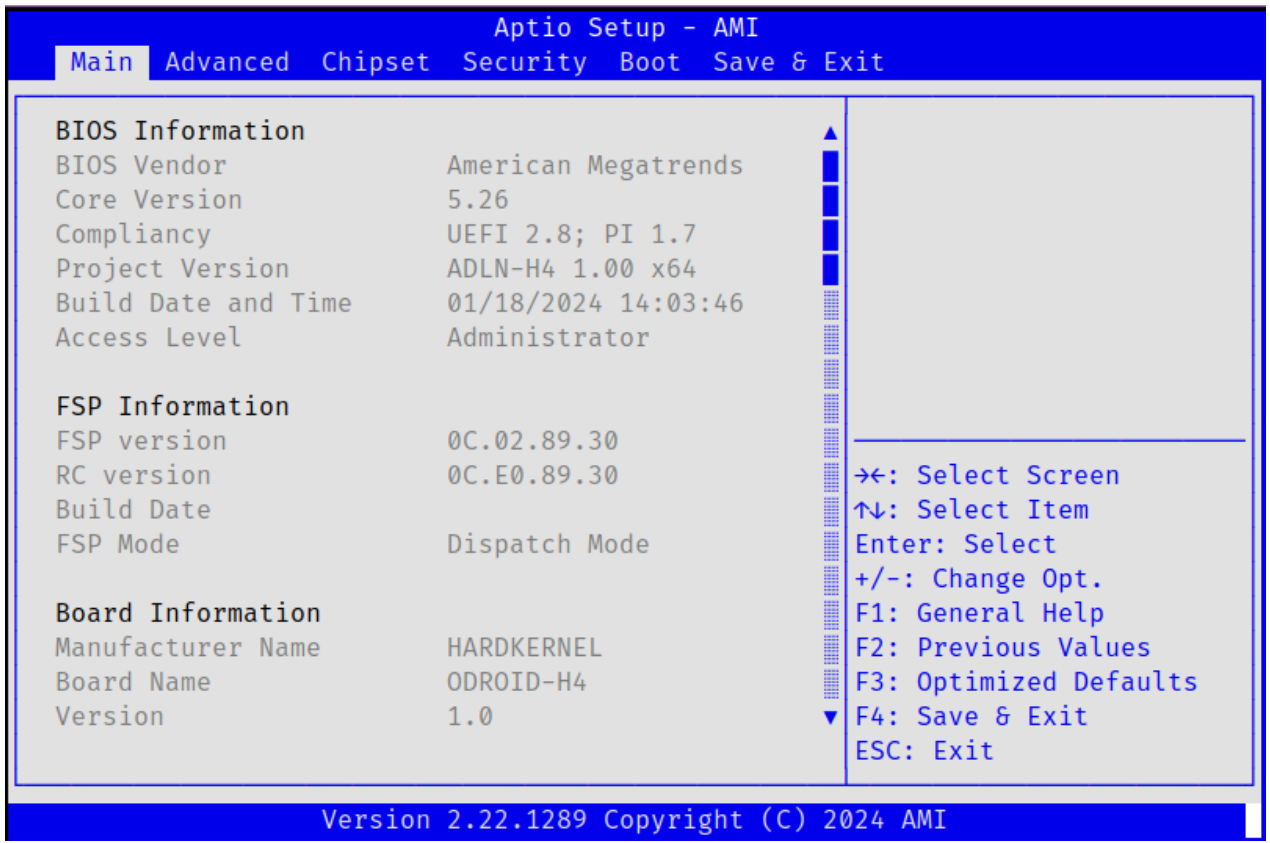
Hint

Disable "Hardware Flow Control" otherwise you may have issues in sending and receiving data.

- Go to `Ctrl` + `a` `o` and select `Serial port setup`
- Use `f` to disable `Hardware Flow Control`
- While leaving remember to `Save setup as dfl`



Let's push `POWER` and observe output. Depending on the terminal, it may look slightly different.



No output on serial diagnostics

- Serial Port Console Redirection misconfigured
  - make sure wires are correctly connected (TX and RX),
  - go to linked documentation and follow the instructions,
  - ask instructor.
- Stuck in limbo
  - sometimes switching between BIOSes, especially when flashing the whole firmware image or accidentally touching ME, may lead to a seemingly unrecoverable state (e.g. switching BIOS_EN jumper doesn't help, no LEDs lit),
  - disconnect power (ACPI G3),
  - if it doesn't help disconnect CMOS battery,

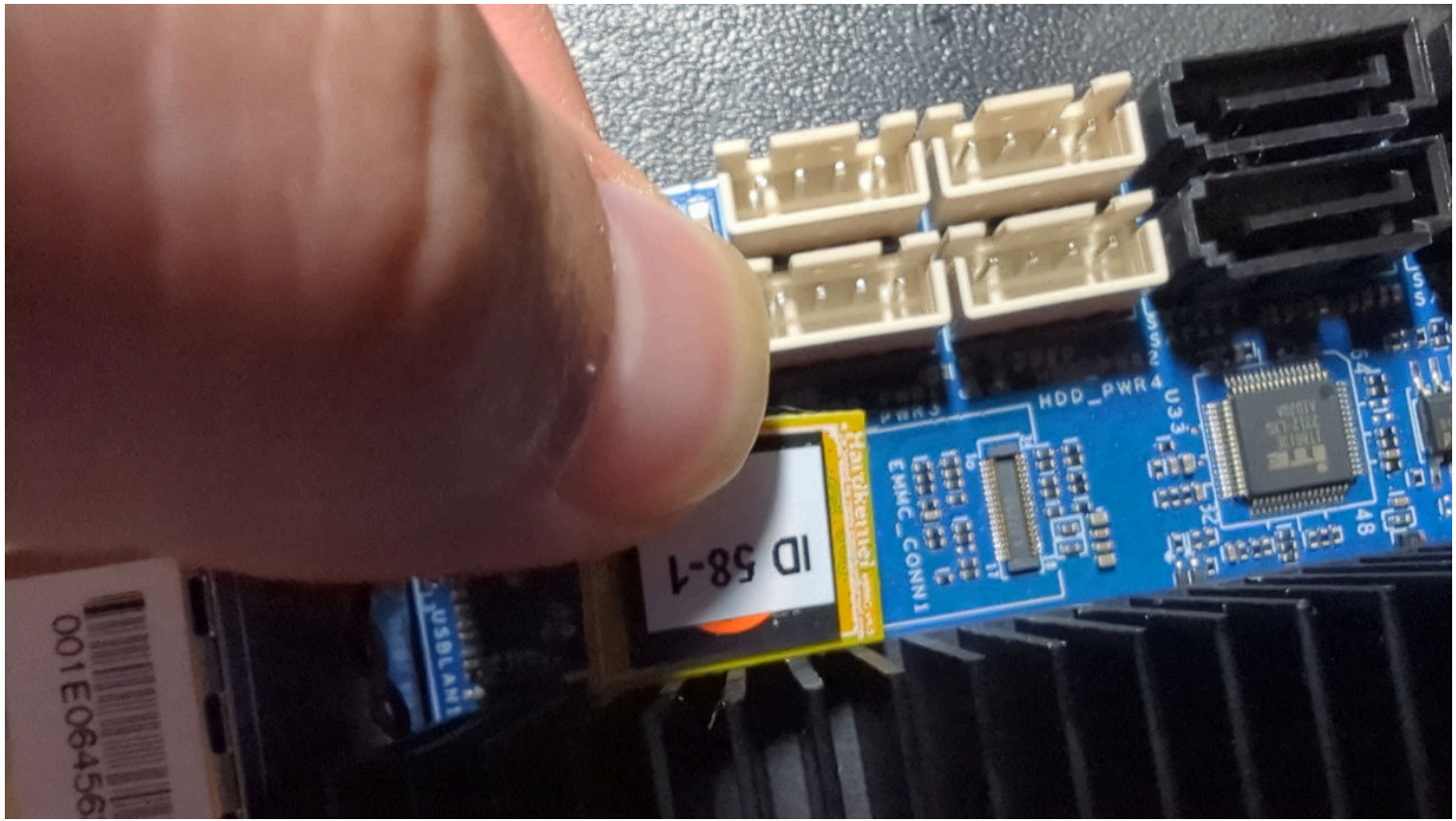
AMI BIOS issues

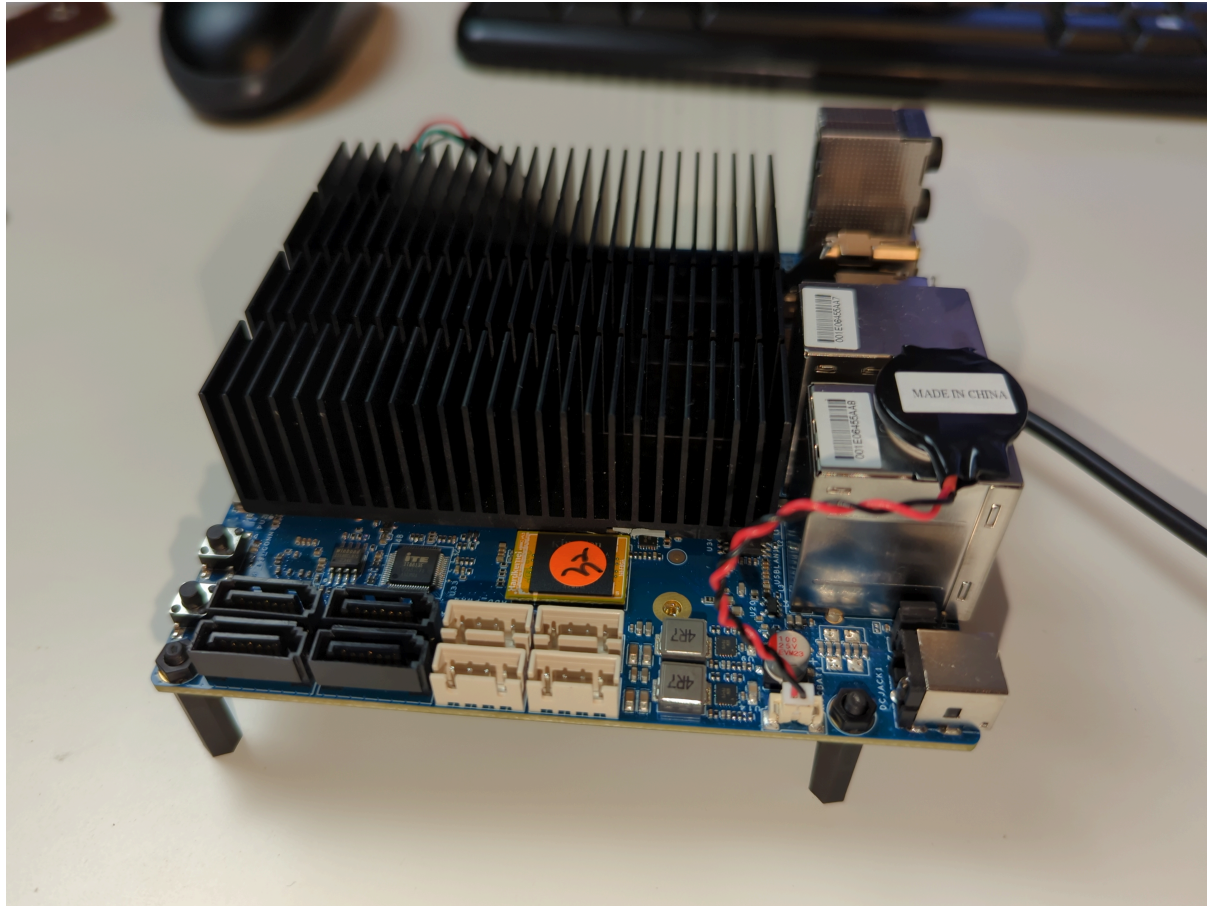
- USB is not always detected.
- `DEL` and `ESC` do not always work over serial.
- UEFI Secure Boot has some misbehavior.

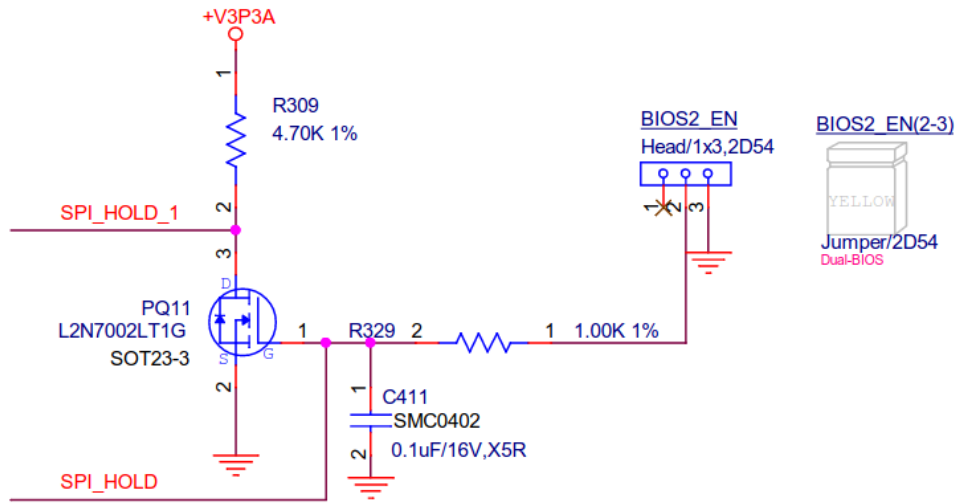
Ubuntu (eMMC) recovery

In case Ubuntu becomes damaged/unbootable instructor should have backup eMMCs with original Ubuntu Server installed. Please inform them so it can be replaced.

Connect eMMC and coin battery, if not yet connected.




MADE IN CHINA
4R7
4R7



Notice

This board has two BIOS SPI flashes that can be switched using a `BIOS_EN` jumper. We do not need to be overly concerned about recovery; we can always switch to the flash with functioning firmware if something goes awry.

Practice #102 Exercise #2

First, we confirm we can boot from both BIOS SPIs.

- Note the position of BIOS jumper.
- Boot to GRUB menu.
- Power off using the button.
- Change the jumper.
- Boot to GRUB menu.
- Only when you confirm that you can continue.

Second, we boot Dasharo Tools Suite (DTS) from USB to perform a BIOS backup.

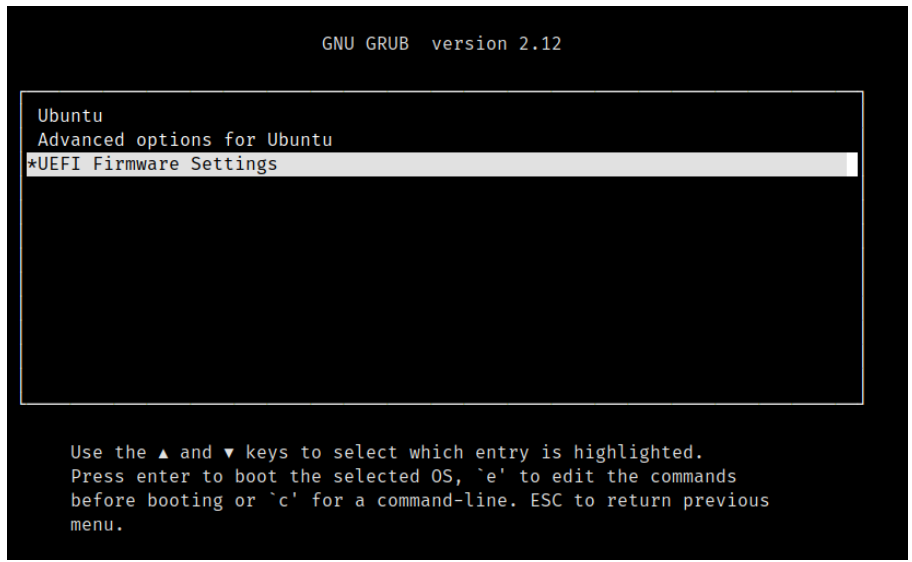
> **Quote**
>
> Dasharo Tools Suite (DTS) is a set of tools running in a minimal Linux environment to deploy, update, and maintain firmware on Dasharo-supported devices. For example, it can be used to update the firmware on a device or run the initial deployment, even when no OS is currently installed.

DTS is built using Yocto. `meta-dts` layer is MIT-licensed and available on Github.

Every experiment and research should start with gathering information about the platform and BIOS backup, which is one of the DTS features called HCL Report.

Warning

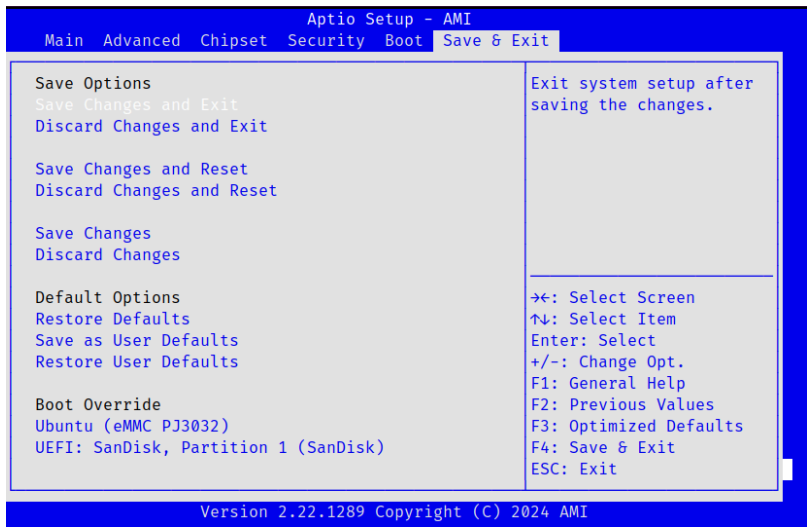
Because AMI BIOS issues with `Del` or `Esc` over serial we have to workaround it by selecting GRUB menu position `UEFI Firmware Settings`:



Warning

Because of another AMI BIOS "feature" USB is not always detected. It was noticed especially when using warm boot (`reboot`) from the system. If you experience that please:

- try to reboot,
- try to cold boot (power off aka ACPI G2/S5, or disconnect power ACPI G3).



> **Hint**
>
> If the serial is mangled, just redraw the serial using `Ctrl` + `L`, then hit `Enter`, and DTS will redraw the menu.

If USB was detected correctly, you can boot into DTS.

➡️ Use `1) Dasharo HCL report` to create HCL report

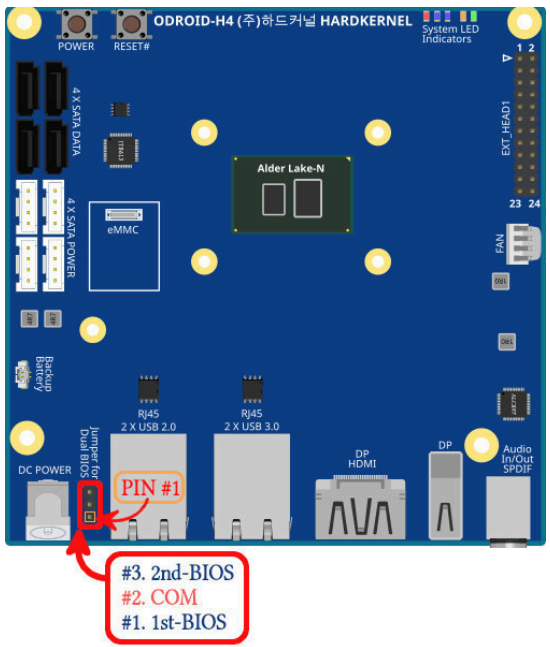
```
Please note that the report is not anonymous, but we will use it only for
backup and future improvement of the Dasharo product. Every log is encrypted
and sent over HTTPS, so security is assured.
If you still have doubts, you can skip HCL report generation.

What is inside the HCL report? We gather information about:
  - PCI, Super I/O, GPIO, EC, audio, and Intel configuration,
  - MSRs, CMOS NVRAM, CPU info, DIMMs, state of touchpad, SMBIOS and ACPI tables,
  - Decoded BIOS information, full firmware image backup, kernel dmesg,
  - IO ports, input bus types, and topology - including I2C and USB,

You can find more info about HCL in docs.dasharo.com/glossary
Do you want to support Dasharo development by sending us logs with your hardware configuration? [N/y]
```

When asked to send logs, hit `Enter`.

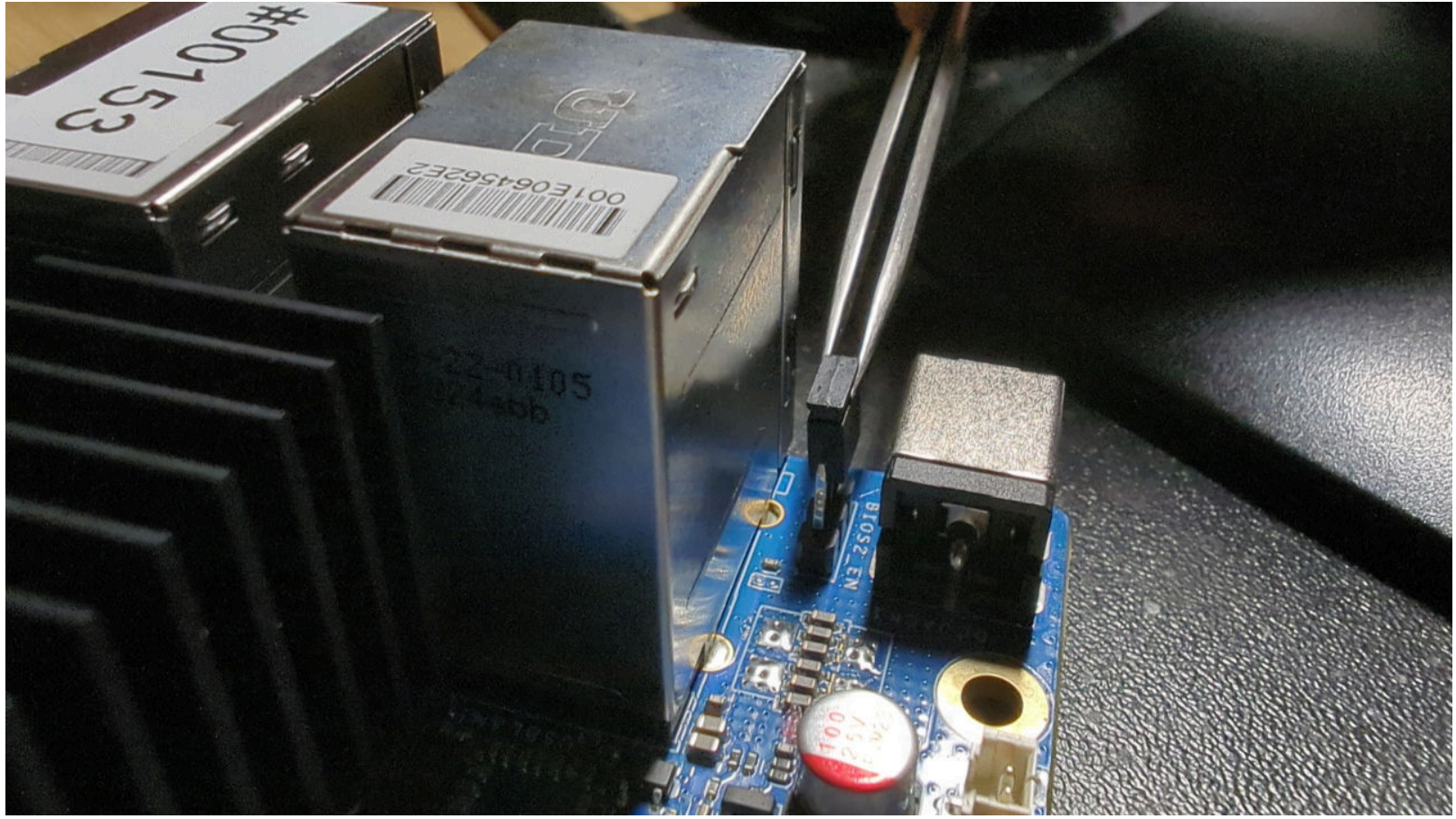
The second question concerns logs' contribution to the Linux Hardware project, we also hit `Enter`.

Recovery

- Disconnect the power cable.
- Change jumper position to use different BIOS



- The power platform is on after which the backup BIOS should boot.


#00153
001E06456262
BIOS_EN
100
25V

- Boot into DTS
- Enter Shell (`S`)

> **Danger**
>
> **Move the jumper position to the previous position i.e. one with damaged BIOS. Failing to do this may lead to breaking your platform.**

- Now we will flash the original, unmodified backup of our BIOS

```
flashrom -p internal -w logs/rom.bin --ifd -i bios
```

- After flashing BIOS finishes, you can reboot, which should result in booting to your original recovered BIOS

- Boot into DTS
- Enter Shell (`S`)

> **Danger**
>
> **Move the jumper position to the previous position i.e. one with damaged BIOS. Failing to do this may lead to breaking your platform.**

- Now we will flash the original, unmodified backup of our BIOS

```
flashrom -p internal -w logs/rom.bin --ifd -i bios
```

- After flashing BIOS finishes, you can reboot, which should result in booting to your original recovered BIOS

- Boot into DTS
- Enter Shell (`S`)

> **Danger**
>
> **Move the jumper position to the previous position i.e. one with damaged BIOS. Failing to do this may lead to breaking your platform.**

- Now we will flash the original, unmodified backup of our BIOS

```
flashrom -p internal -w logs/rom.bin --ifd -i bios
```

- After flashing BIOS finishes, you can reboot, which should result in booting to your original recovered BIOS

- Boot into DTS
- Enter Shell (`S`)

> **Danger**
>
> **Move the jumper position to the previous position i.e. one with damaged BIOS. Failing to do this may lead to breaking your platform.**

- Now we will flash the original, unmodified backup of our BIOS

```
flashrom -p internal -w logs/rom.bin --ifd -i bios
```

- After flashing BIOS finishes, you can reboot, which should result in booting to your original recovered BIOS

- Boot into DTS
- Enter Shell (`S`)

> **Danger**
>
> **Move the jumper position to the previous position i.e. one with damaged BIOS. Failing to do this may lead to breaking your platform.**

- Now we will flash the original, unmodified backup of our BIOS

```
flashrom -p internal -w logs/rom.bin --ifd -i bios
```

- After flashing BIOS finishes, you can reboot, which should result in booting to your original recovered BIOS

- Boot into DTS
- Enter Shell (`S`)

> **Danger**
>
> **Move the jumper position to the previous position i.e. one with damaged BIOS. Failing to do this may lead to breaking your platform.**

- Now we will flash the original, unmodified backup of our BIOS

```
flashrom -p internal -w logs/rom.bin --ifd -i bios
```

- After flashing BIOS finishes, you can reboot, which should result in booting to your original recovered BIOS

Quiz #3

Quiz #3

What is the most important operation before hacking boot firmware?

Quiz #3

What is the most important operation before hacking boot firmware?

- Backup

Quiz #3

What is the most important operation before hacking boot firmware?

- Backup

Where is boot firmware kept on Odroid-H4+?

Quiz #3

What is the most important operation before hacking boot firmware?

- Backup

Where is boot firmware kept on Odroid-H4+?

- SPI,

Quiz #3

What is the most important operation before hacking boot firmware?

- Backup

Where is boot firmware kept on Odroid-H4+?

- SPI,

How many SPIs Odroid-H4+ has?

Quiz #3

What is the most important operation before hacking boot firmware?

- Backup

Where is boot firmware kept on Odroid-H4+?

- SPI,

How many SPIs Odroid-H4+ has?

- Two,

Quiz #3

What is the most important operation before hacking boot firmware?

- Backup

Where is boot firmware kept on Odroid-H4+?

- SPI,

How many SPIs Odroid-H4+ has?

- Two,

How to switch between those two?

Quiz #3

What is the most important operation before hacking boot firmware?

- Backup

Where is boot firmware kept on Odroid-H4+?

- SPI,

How many SPIs Odroid-H4+ has?

- Two,

How to switch between those two?

- Using BIOS2_EN jumper,

Conclusion

In this lecture, we learned about the following:

- How to connect, power on, and confirm successful boot of Odroid-H4+.
- The recovery methods in case of broken firmware on Odroid-H4+ and how to apply them.
- How to communicate with the platform over serial.

These skills will help us perform experiments related to Intel Root of Trust and Slim Bootloader safely and with high confidence.

Questions?