Getting started with OAK4-DEV-KIT R2D0

You have received the OAK4-DEV-KIT, which is meant for evaluation and research of the new RVC4 SoM platform from Luxonis. It incorporates the vast majority of the features that are built in on our SoM. Here are the very first steps needed in order to get your board up and running.

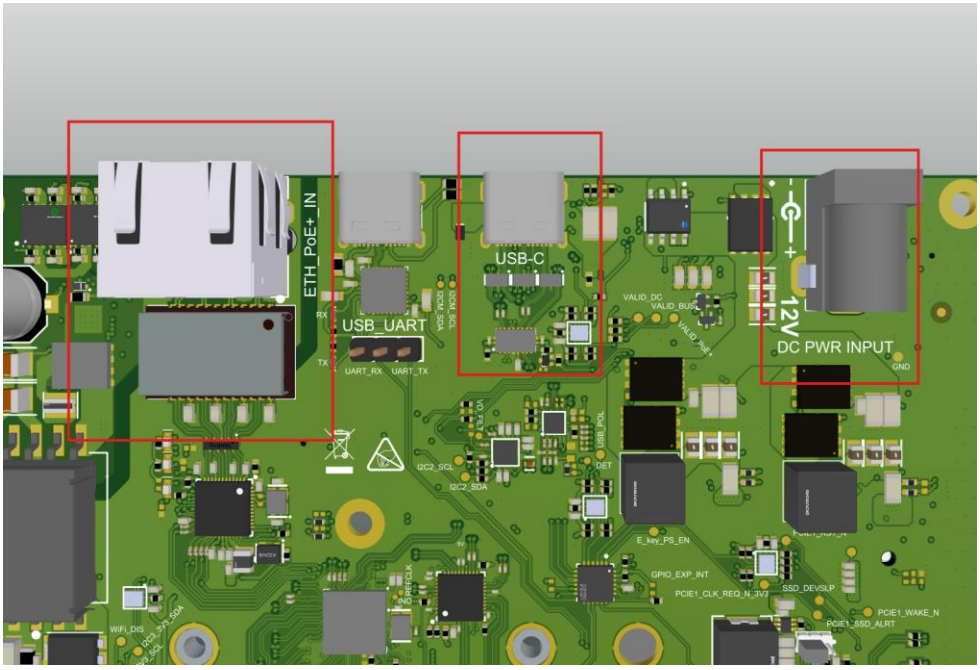
Powering the DEV-KIT

DEV-KIT can be powered from 3 different power sources:

- **DC PWR INPUT** barrel jack, where you need an AC-DC adapter (12 V), that should be included in the shipment. This one has the highest priority among all of them.
- **USB-PD**, that should be plugged to the Type C port marked with USB-C on the PCB. This port has second priority, which means, the device will be powered from this source if there is no DC PWR INPUT power present or it does not have a sufficient voltage range.
- **PoE+ input.** DEV-KIT can be also powered from PoE+ (30W) capable PSE. This power source is the last one in priority list, which means the device will be powered from this one, if neither of the previous ones is plugged in or within specified voltage range.
  Note: USB-C can be used in data mode and device powered from PoE+ simultaneously.

There are also other combinations possible in regard to data-power combination:

- Powered from DC PWR INPUT and data over USB-C
- Powered from DC PWR INPUT and data over ETH
- Powered from USB PD and data over ETH (non PoE switch/router)



Cooling of the SoM

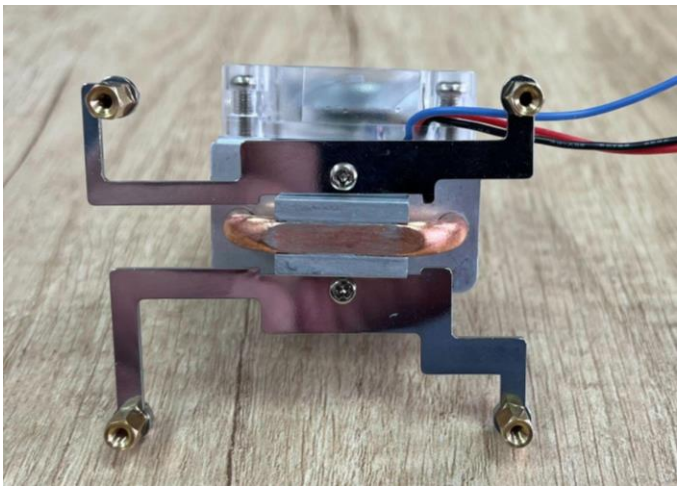
In order to get maximum possible performance, you should assemble a device with the cooler for the main CPU, that is placed on the SoM (black PCB). Here are the instructions on how to do it in order to install it properly and not make any damage.

Note: Device can be run even without this cooler, but this will result in poorer performance due to CPU cooling not being sufficient. The device will not get damaged, only throttled.

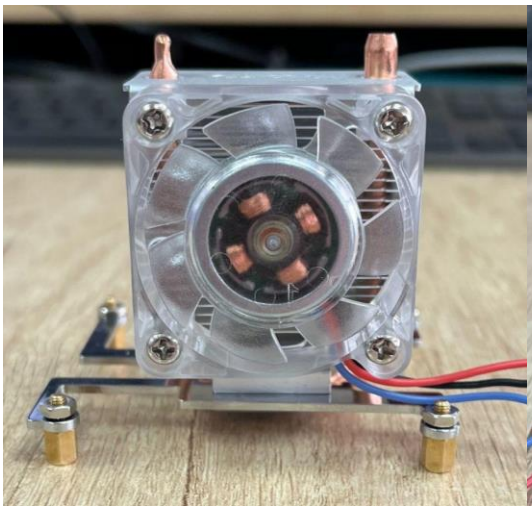
Assemble cooler as per following picture:

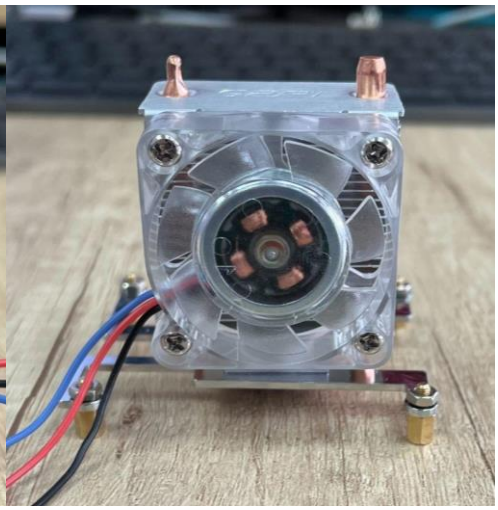
1. Screw in spacers to the brackets. The hole should be facing downwards. Use spacers/nuts included in the bag.
   Note: On some DEV-KITs spacer might already be populated on the PCB, so you can skip this step.
2. Screw brackets to the main “tower” like in the picture below.

Note: Take care of the bracket shape and position.



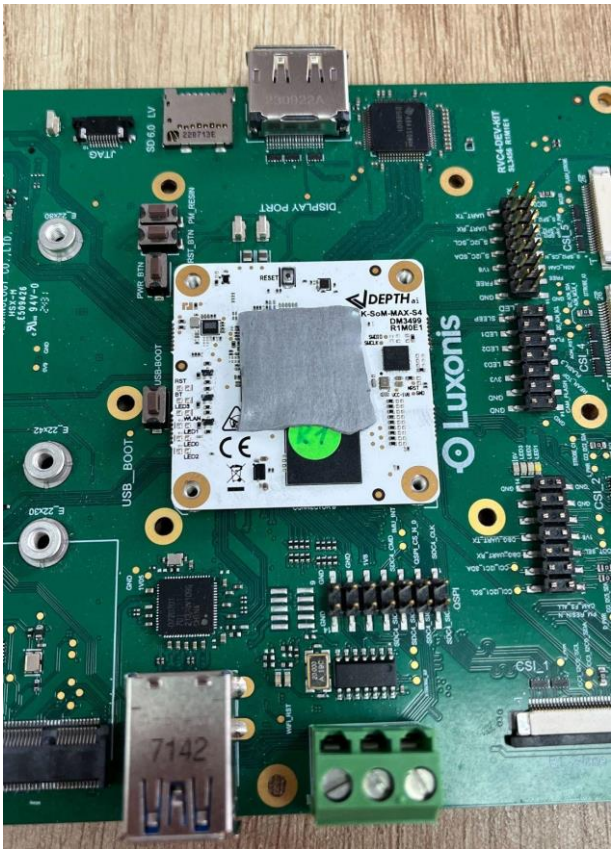
Unscrew and rotate fan in CW (clock wise) for 90 deg. Use a screwdriver that is included in the cooling tower box or you can also use your own if you have it.
Reference pictures before-after rotating fan:





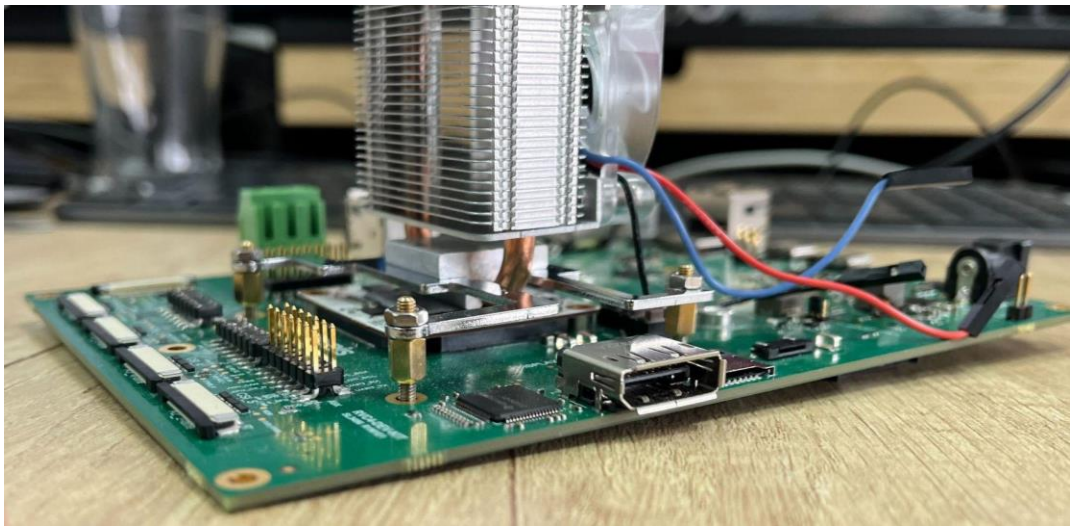
This is how cooler should look like from above, if all the steps are completed successfully:

Take DEV-KIT and added TIM (ThermalInterfaceMaterial) and place it on the SoM like in the picture below:
Note: SoM in your case has a different colour and shape of the TIM might vary a bit. Some details of the PCB might also be different. Spacer for the cooler might already be populated as well. Nothing to worry about.

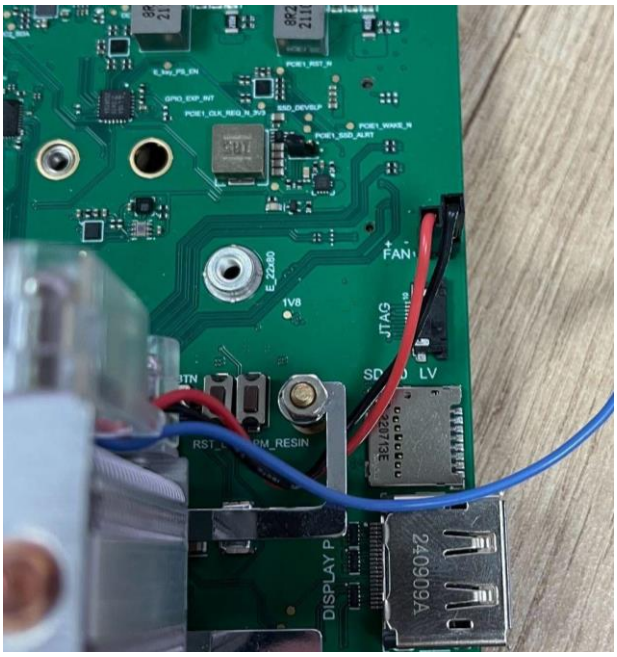


Take a cooler “tower” and place it on the PCB.
Note: take care of orientation. Orientation should be as in the picture.

Take Phillips screws from the cooler box and screw the brass spacers from the bottom side of the DEV-KIT. Some force might be needed in order to screw in screws. The result is shown in the picture.
Note: turn screws only ~2 turns, otherwise you can damage the PCB. If spacers are pre-populated on the PCB, you can tighten screws.

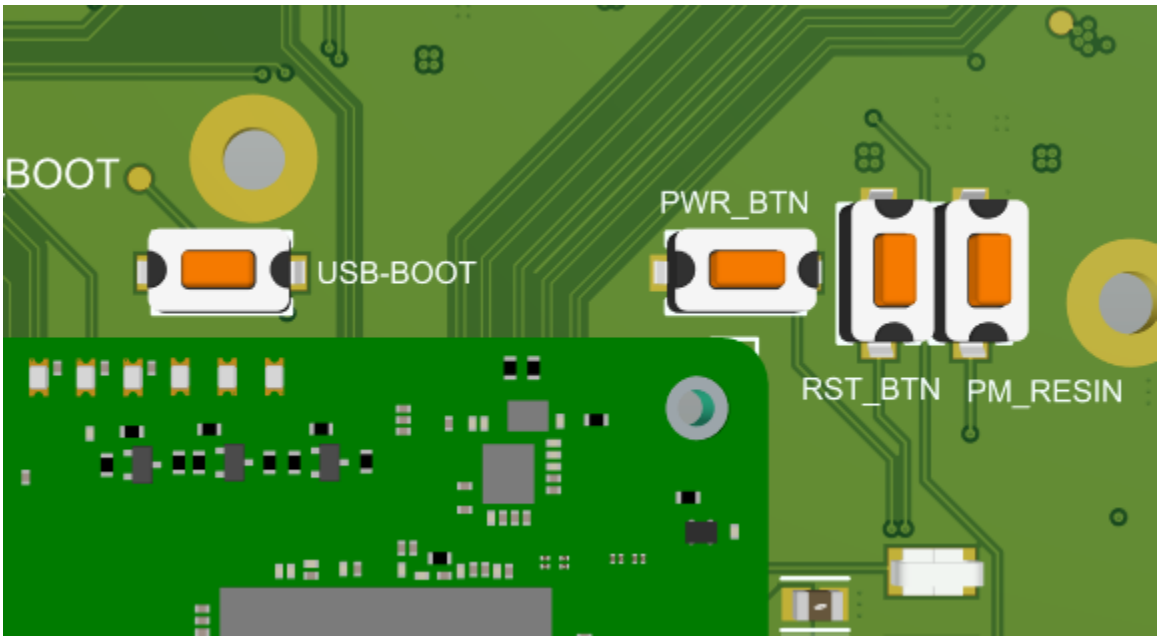


Plug in the red (+5V) and black (- ; GND) wire to the 5V header on the DEV-KIT like in the picture below to power the cooling fan. The fan will turn on as soon as the board is powered from one of the power sources. Leave blue wire unconnected.
Note: Please note the colour and position of the wires.



On-board buttons

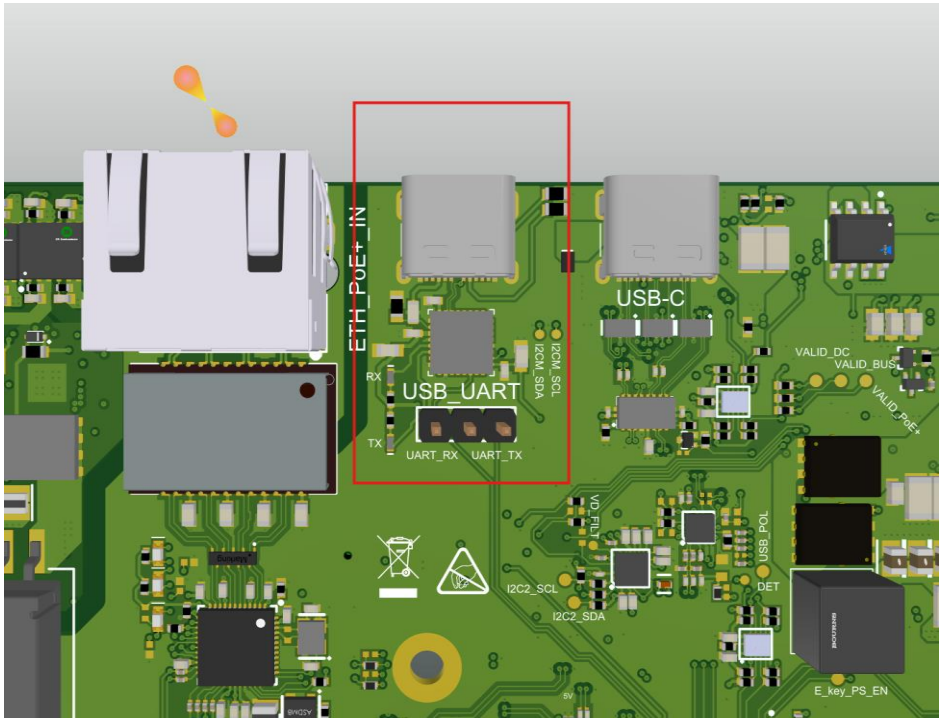
There are 4 buttons on the board that are important for the user.



- ***USB-BOOT*** puts the device into flashing mode. Hold this button, insert USB-C cable and one of the 3 power sources. Hold for an additional 3 seconds after power is applied. This will put SoM into BOOT mode, where you can flash it.
- ***PWR_BTN*** puts board into fast boot mode.
- ***RST_BTN*** No specific function.
- ***PM_RESIN*** No Specific function.

Serial debug

OAK4-DEV-KIT also incorporates a serial debug function for SoM with on-board USB-serial converter.
Serail debug is reachable over Type-C connector between USB-C and RJ45 connector.

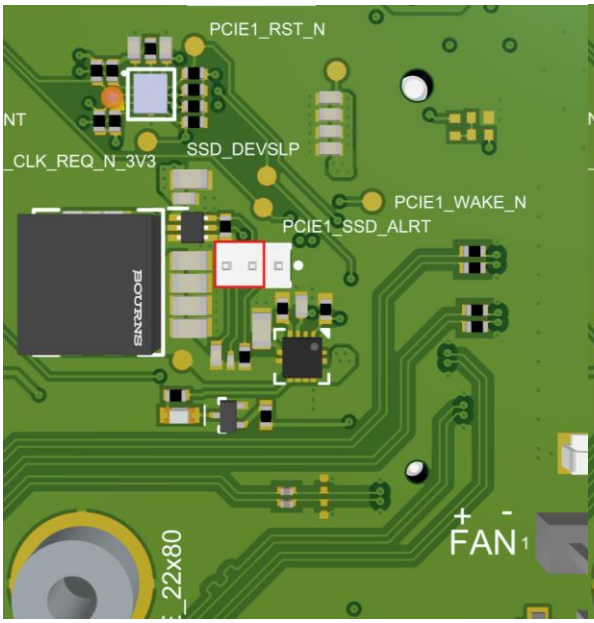


Serial debug works on 115200 baud, 8N1 configuration. If some other connection is needed, a 3 pin header can be used, which is placed under the USB-UART text. Voltage level is 1.8V.
Note: In case using header for UART strictly use 1.8V level, otherwise you will damage SoM.

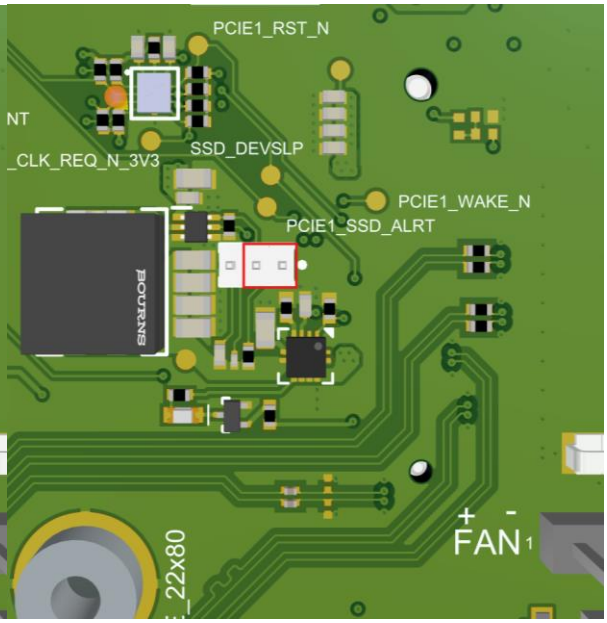
M.2 M-key power supply rail voltage

Power supply voltage for the M-key slot can be adjusted in two levels, 3.3V and 3.8V. The red rectangle marks the position of the jumper.

3.3V

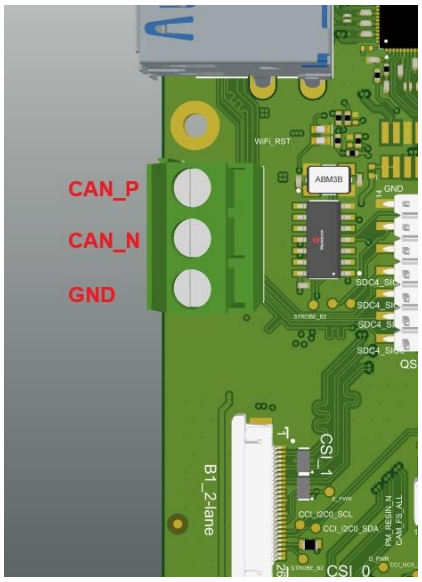


3.8V



CAN pinout

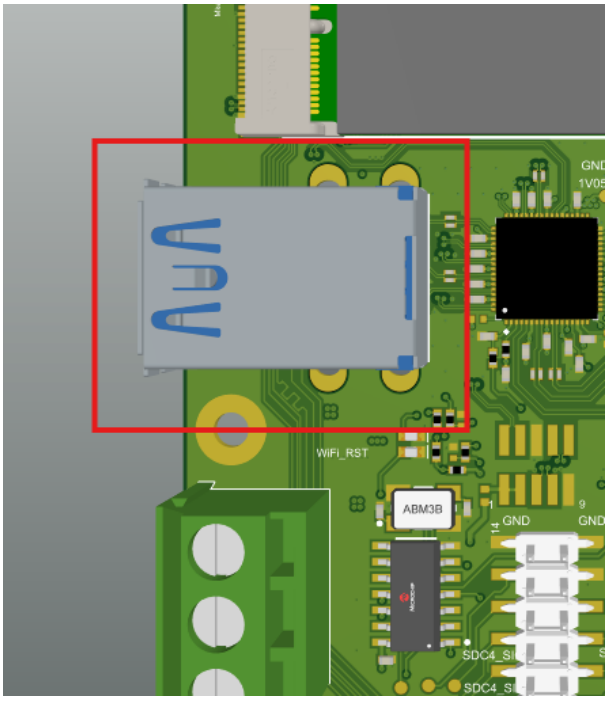
CAN connector (screw-in, 5.08mm pitch) has the following pinout:



Lines are terminated on the board with 120 OHM.

USB-A connector

There are 2 USB-A connectors on the board that are connected to the on-board USB hub. Both connectors support 3.0 speed and can deliver up to at least 500 mA per channel. There is no USB PD at these ports.



Connecting to the board

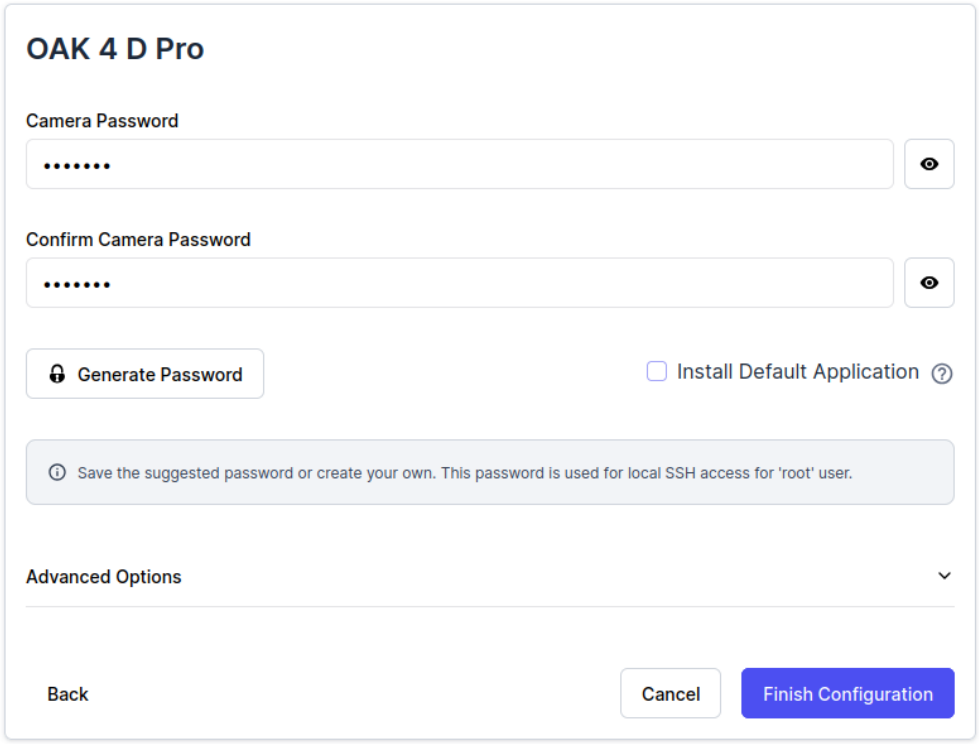
You can access the board either via network or via USB-C using ADB or via console (debug UART using the debug USB-C port or UART pins).

Prerequisites

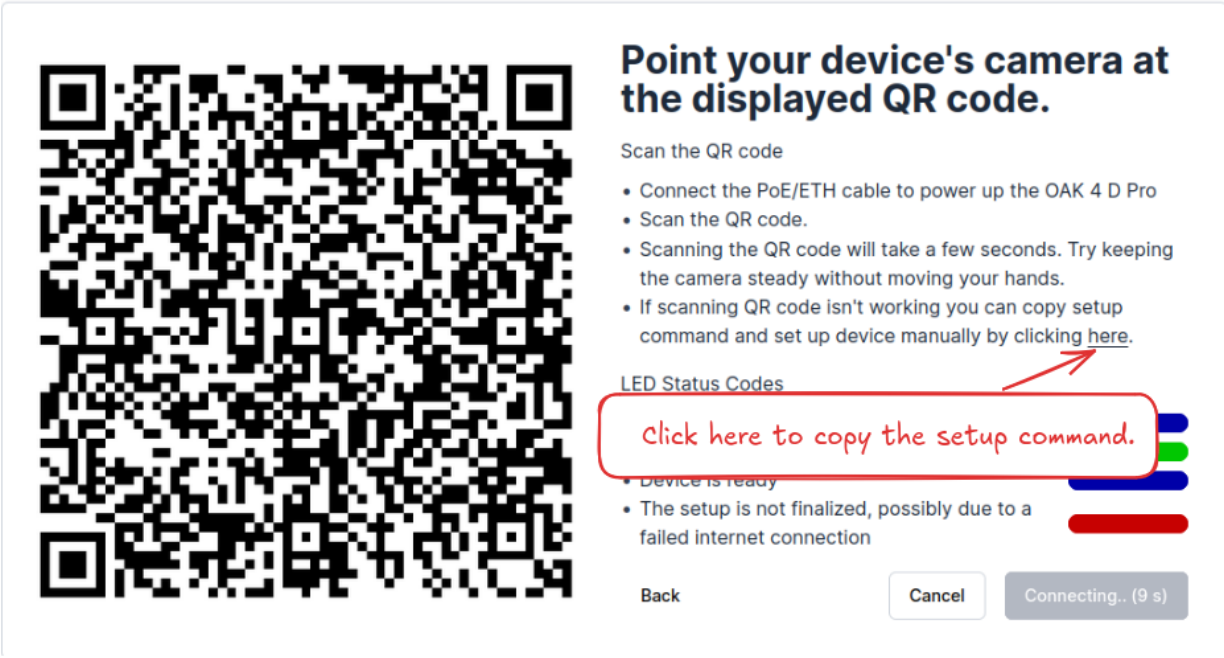
You will need a USB-C data cable to connect to the device via USB-C and ADB. You will need to have Android Debug Bridge (adb) client program installed on your system. For how to install adb on your system, please consult the following tutorial: Installing ADB on different systems.
After the USB cable has been properly connected to the USB-C port (not the USB-C debug port), the device has been powered up and booted the device should become visible using the “adb devices” command.
You can then use the adb program to connect, transfer files and configure the device

To set up the password for the device you should visit https://setup.luxonis.com/ , select the OAK 4 D Pro device, then click on “Setup Locally”.

Configure the password for the root user:



Click Finish Configuration



A QR code will be shown on the screen - click the highlighted “here” text to copy the command to set up the device via USB. Connect a USB-C cable between your computer and device and run the setup command via adb: adb shell <paste your setup command here>

Connecting via Console

You should use a Terminal program such as PuTTY, GTKTerm, screen or similar program to connect to the DEV-KIT console.
You can connect using any USB-C cable to the debug USB-C port, or using an external USB serial adapter and connecting to the UART pins on the board.
In either case you should set your terminal parameters to 115200 bits per second, with 8 data bits and one parity bit format (8N1).

Connecting via Ethernet

The board will try to obtain an IP address from the local DHCP server if connected to a network using an Ethernet cable. You can use your DHCP server or router information, nmap or similar program to find the board address on the local network, or you can look it up via Console or Adb connection:

```
/ $ ifconfig
enp5s0f1  Link encap:Ethernet  HWaddr 44:A9:2C:3F:6C:24
       inet addr:10.12.113.220  Bcast:10.12.113.255  Mask:255.255.255.0
       inet6 addr: fe80::17a6:3dbd:c826:2f3/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:55085 errors:0 dropped:11074 overruns:0 frame:0
       TX packets:191 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:4672647 (4.4 MiB)  TX bytes:14716 (14.3 KiB)
       Interrupt:122

```

After you have found the device address, you can use Secure Shell (ssh) client program to connect to the device:

```
$ ssh root@10.12.113.220

```

Starting additional services

CAN Bus

CAN is not started at boot, as all the required parameters are not known to us.
To start the CAN bus operation execute following code:

```
ip link set can0 type can bitrate 500000 # or your CANBus bitrate
ip link set can0 up

```

You can use the cansend and candump commands to quickly test the can link operation.