

Preparation before use

You can refer to AI HAT+ - Raspberry Pi Documentation <https://www.raspberrypi.com/documentation/accessories/ai-hat-plus.html#ai-hat-plus>

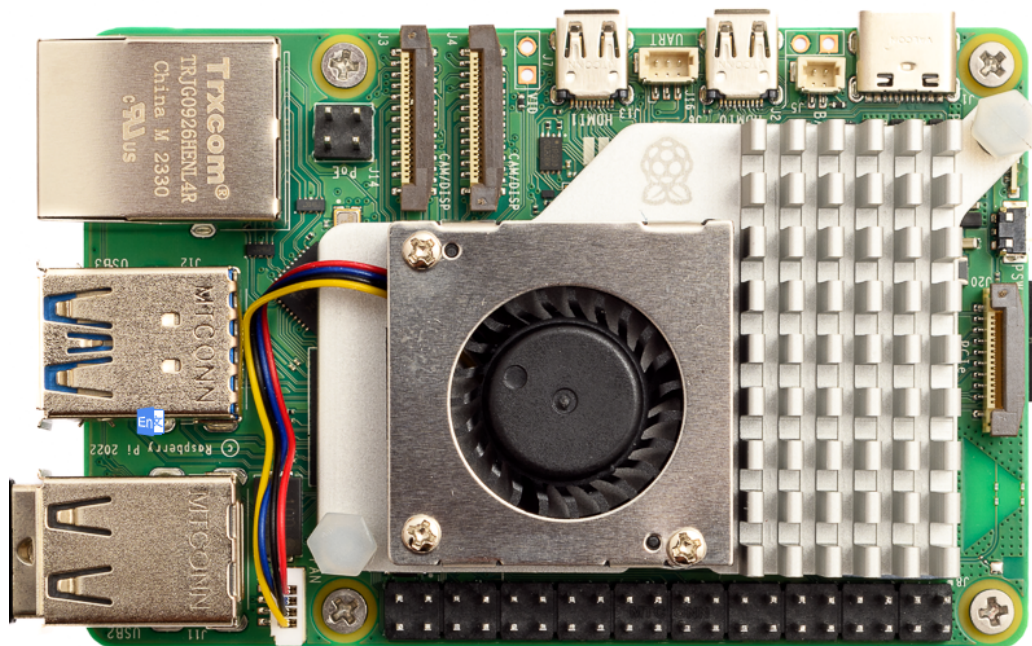
In this section, learn how to set up a Raspberry Pi 5 with the AI HAT+.

Required Content

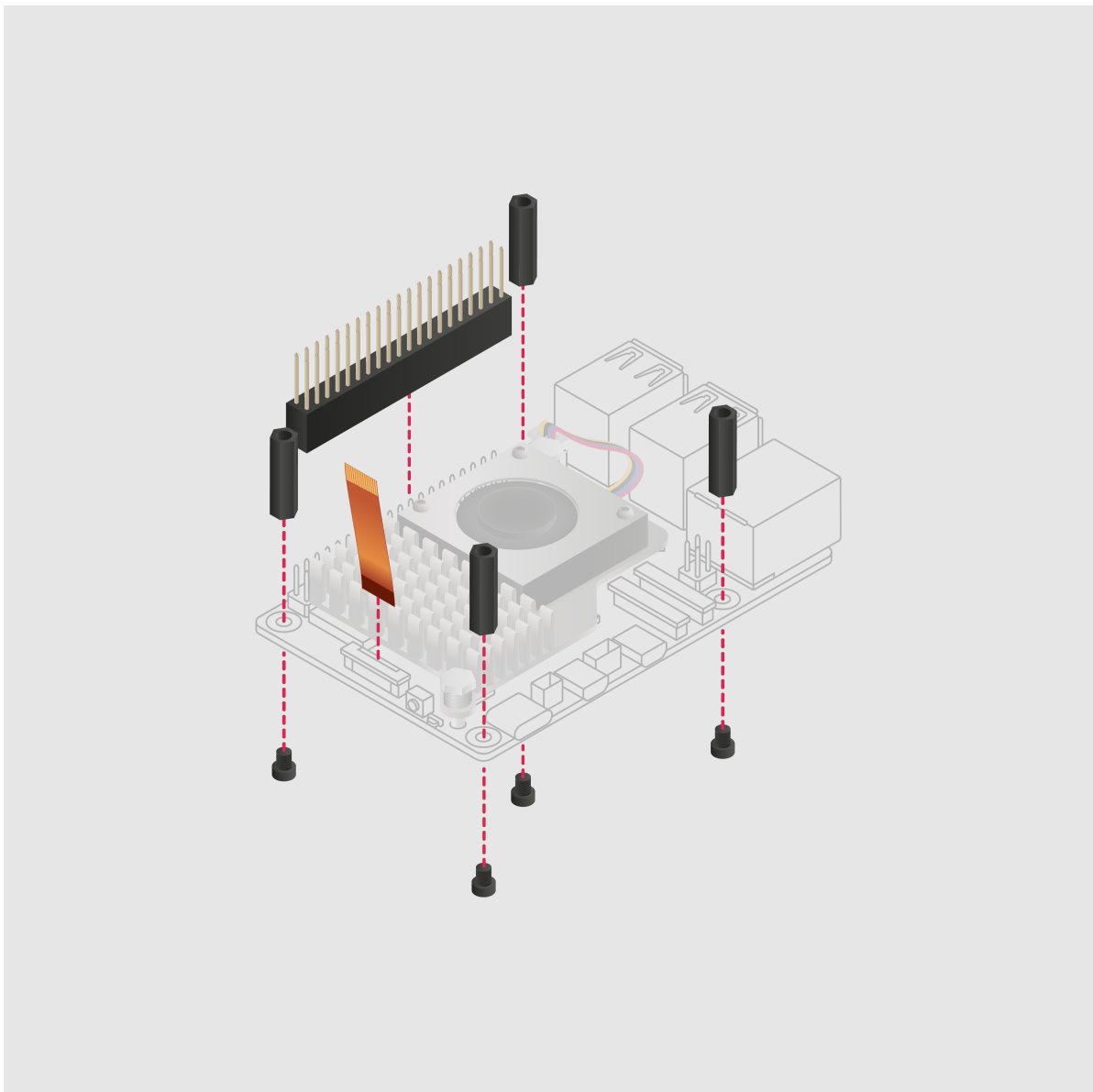
- Raspberry Pi 5
- AI HAT+
- Thermal pad (included in the kit)
- Optional: Heat sink
- Optional: Official Raspberry Pi camera (e.g., Camera Module 3 or High Quality Camera)
- Optional: USB camera

Hardware:

We used a Raspberry Pi 5 model with the official active cooler and a 27W USB-C power supply. We recommend using the official USB-C power supply or one that supports the Raspberry Pi 5 Power Delivery protocol to ensure the motherboard can provide sufficient power to the AI HAT+.



Install the spacer using the four screws provided. Firmly press the GPIO 40pin header onto the bottom of the Raspberry Pi GPIO pins; confirm that all pins are seated in place. Disconnect the cable from the AI HAT+ and insert the other end into the PCIe port of the Raspberry Pi. Lift the cable brackets from both sides and insert the cable with the copper contacts facing inward, toward the USB port. Once the cable is fully and evenly inserted into the PCIe port, lower the cable brackets from both sides to secure the cable in place.



The installation is as shown below:



For CSI camera connection, please refer to the Raspberry Pi 5 camera case.

Software:

Note: The SD card of the Raspberry Pi 5 needs to contain a bootable image. Please refer to the image burning in the Raspberry Pi 5 tutorial we provided

The interface that appears after successful startup



UPDATE SYSTEM

Note: (The following steps are only required for DIY in your own environment. If you directly use the image we provide, you do not need to rebuild it)

```
sudo apt update
sudo apt full-upgrade
```

```
pi@raspberrypi:~ $ sudo apt update
Hit:1 https://mirrors.aliyun.com/docker-ce/linux/debian bookworm InRelease
Hit:2 http://deb.debian.org/debian bookworm InRelease
Hit:3 http://deb.debian.org/debian-security bookworm-security InRelease
Hit:4 http://archive.raspberrypi.com/debian bookworm InRelease
Hit:5 http://deb.debian.org/debian bookworm-updates InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
11 packages can be upgraded. Run 'apt list --upgradable' to see them.
pi@raspberrypi:~ $ sudo apt full-upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libcamera0.1 libraspberrypi0 libssl1.1 libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  gir1.2-gtk-3.0 gtk-update-icon-cache gtk2-engines-pixbuf libgtk-3-0
  libgtk-3-common libgtk2.0-0 libgtk2.0-bin libgtk2.0-common pipanel
  raspberrypi-sys-mods wpasupplicant
11 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 23.3 MB of archives.
```

This will update your system to the latest Raspberry Pi core, which includes AI HAT+ driver support.

Next, [make sure your Raspberry Pi firmware is up to date](#). Run the following command to see what firmware you are running:

```
sudo rpi-eeprom-update
```

If you see a date of December 6, 2023 or later, proceed to the next step. If you see a date earlier than December 6, 2023, run the following command to open the Raspberry Pi Configuration CLI:

```
sudo raspi-config
```

Under **Advanced Options** > **Bootloader Version**, select **Latest**. Then exit **raspi-config** using **Finish** or the **Esc** key.

Run the following command to update the firmware to the latest version:

```
sudo rpi-eeprom-update -a
```

Then, reboot

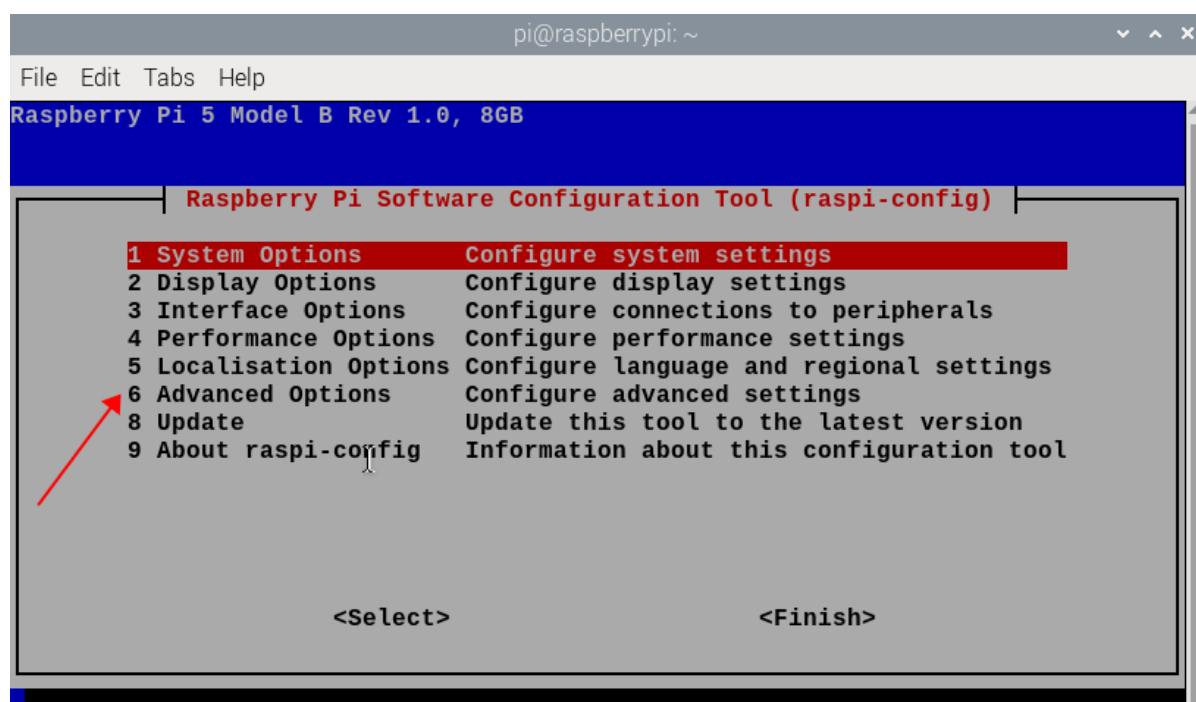
```
sudo reboot
```

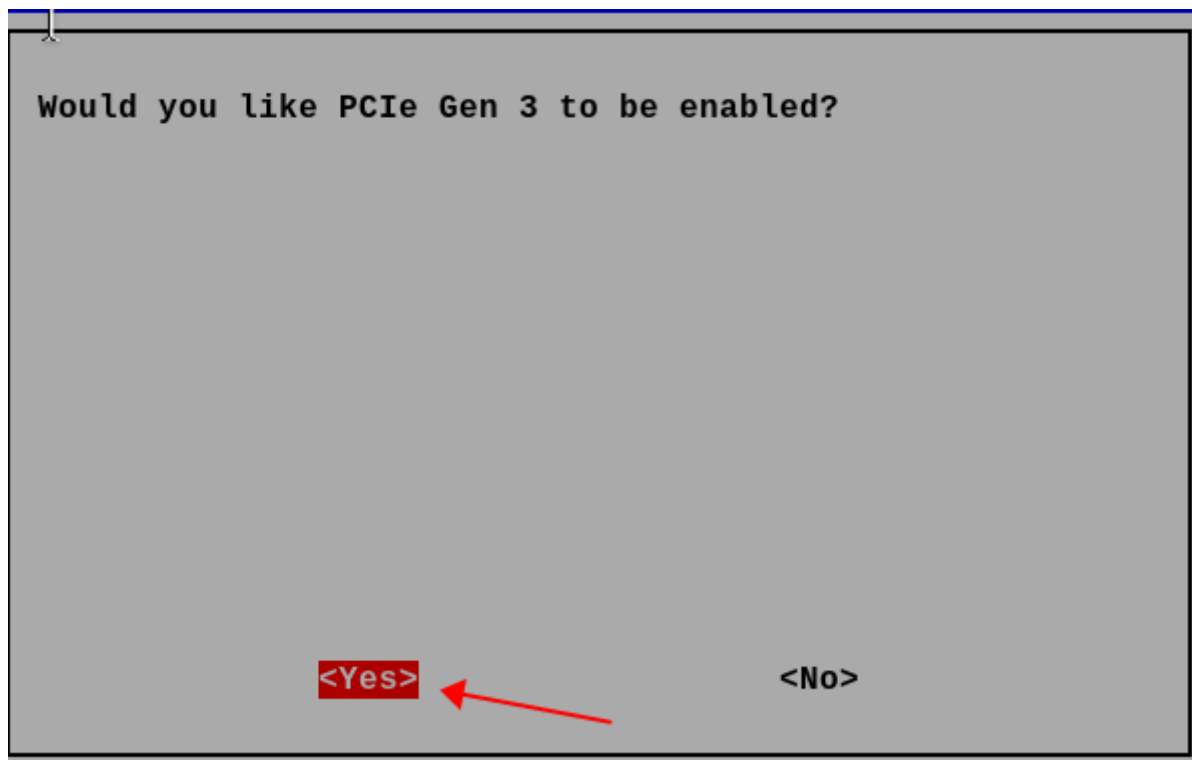
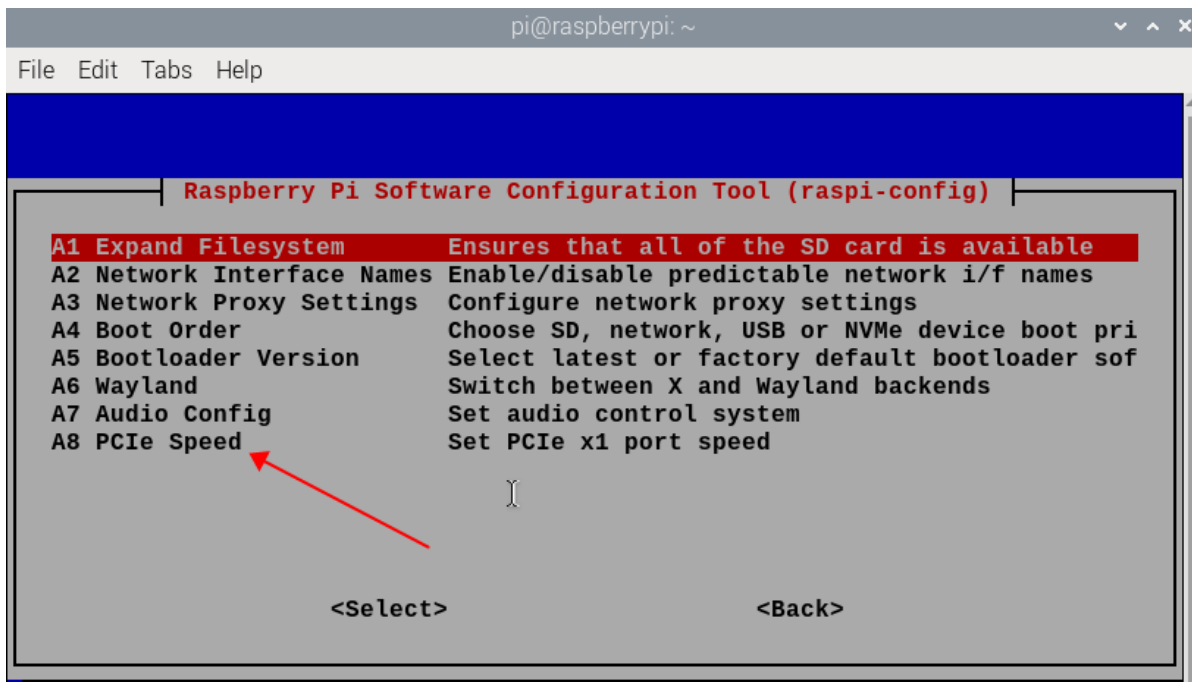
Set PCIe to Gen3

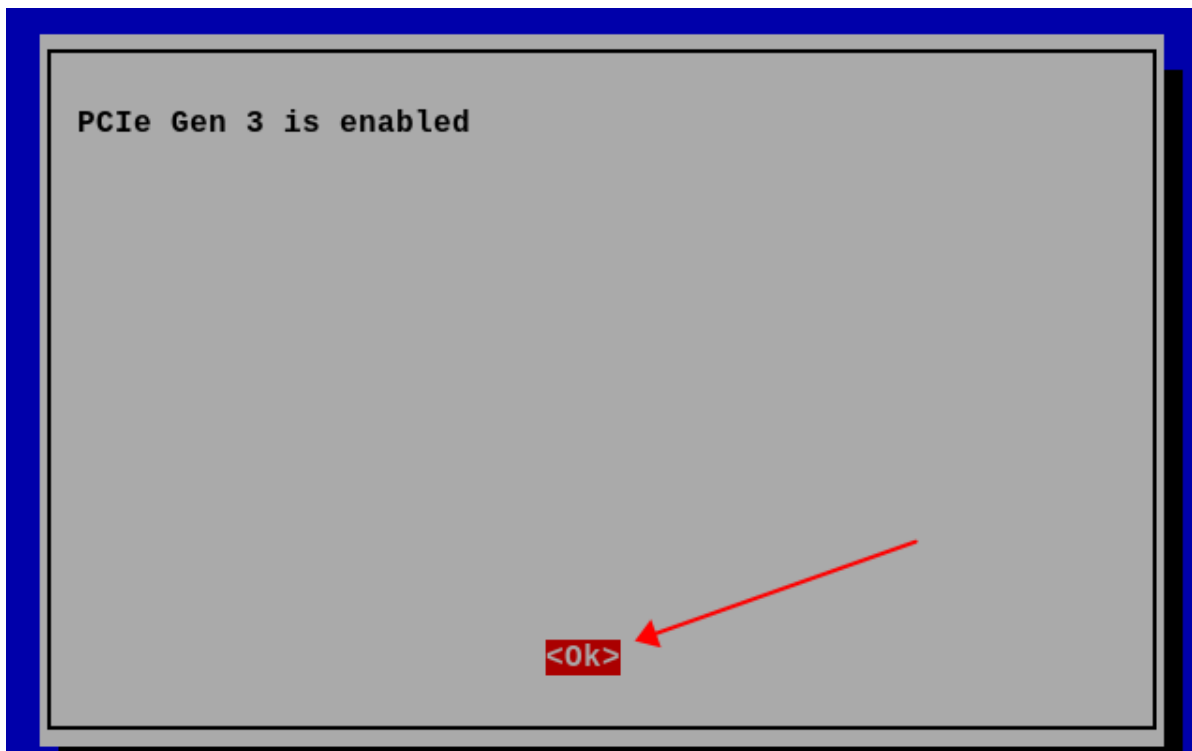
To achieve optimal performance of the AI HAT+ device, it is necessary to set PCIe to Gen3. While using Gen2 is an option, it will result in reduced performance.

Open the Raspberry Pi Configuration Tool:

```
sudo raspi-config
```







Select option 6 Advanced options and then option A8 PCIe speed. Select Yes to enable PCIe Gen 3 mode. Click Finish to exit.

Install hailo software

Install all necessary software to make the Raspberry Pi AI HAT+ work properly. To do this, run the following command from a terminal window:

```
sudo apt install hailo-all
```

The following picture shows that the installation has been successful

```
pi@raspberrypi:~ $ sudo apt install hailo-all
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
hailo-all is already the newest version (3.28.2+1).
The following packages were automatically installed and are no longer required:
  libcamera0.1 libraspberrypi0 libssl1.1 libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi:~ $
```

This will install the following software components:

- Hailo firmware
- HailoRT runtime software. More information available.
- Hailo post-processing software demo stage.

Reboot your Raspberry Pi.

```
sudo reboot
```

Test the TAPPAS Core installation by running the following command:

Hailotools:

```
gst-inspect-1.0 hailotools
```


Expected results:

```
Plugin Details:
  Name      hailotools
  Description hailo tools plugin
  Filename   /lib/aarch64-linux-gnu/gstreamer-1.0/libgsthailotools.so
  Version    3.28.2
  License    unknown
  Source module  gst-hailo-tools
  Binary package  gst-hailo-tools
  Origin URL    https://hailo.ai/

hailoaggregator: hailoaggregator - Cascading
hailocounter: hailocounter - postprocessing element
hailocropper: hailocropper
hailoexportfile: hailoexportfile - export element
hailoexportzmq: hailoexportzmq - export element
hailofilter: hailofilter - postprocessing element
hailogallery: Hailo gallery element
hailograytonv12: hailograytonv12 - postprocessing element
hailoimportzmq: hailoimportzmq - import element
hailomuxer: Muxer pipeline merging
hailonv12togray: hailonv12togray - postprocessing element
hailonvalve: HailoNValve element
hailooverlay: hailooverlay - overlay element
hailoroundrobin: Input Round Robin element
hailostreamrouter: Hailo Stream Router
hailotileaggregator: hailotileaggregator
hailotilecropper: hailotilecropper - Tiling
hailotracker: Hailo object tracking element
```

Hailonet:

```
gst-inspect-1.0 hailo
```

Expected results:

```
pi@raspberrypi:~ $ gst-inspect-1.0 hailo
Plugin Details:
  Name      hailo
  Description hailo gstreamer plugin
  Filename   /lib/aarch64-linux-gnu/gstreamer-1.0/libgsthailo.so
  Version    1.0
  License    unknown
  Source module  hailo
  Binary package  GStreamer
  Origin URL    http://gstreamer.net/

hailodevicestats: hailodevicestats element
hailonet: hailonet element
synchailonet: sync hailonet element

3 features:
+-- 3 elements
```

If found or not, try deleting the GStreamer registry: `hailo` `hailotools`

```
rm ~/.cache/gstreamer-1.0/registry.aarch64.bin
```

Known Issues

The following issues should be handled by the TAPPAS Core installation deb, but if you encounter them, you can fix them manually.

PCIe Page Size Issues

Some hosts do not support certain PCIe descriptor page sizes. If you receive an error like:

```
[HailoRT] [error] CHECK_AS_EXPECTED failed - max_desc_page_size given 16384 is bigger than hw max desc page size 4096
```

Make sure that exists and contains the following line: `/etc/modprobe.d/hailo_pci.conf`

```
options hailo_pci force_desc_page_size=4096
```

Check the configuration:

```
cat /etc/modprobe.d/hailo_pci.conf
# expected result:
options hailo_pci force_desc_page_size=4096
```

Unable to allocate memory in static TLS block

In some cases (especially aarch64), you may encounter the following error, causing some GStreamer plugins to not load correctly. The error message is:

```
bash
(gst-plugin-scanner:67): GStreamer-WARNING **: 12:20:39.178: Failed to load plugin '/usr/lib/aarch64-linux-gnu/gstreamer-1.0/libgstlibav.so': /lib/aarch64-linux-gnu/libgomp.so.1: cannot allocate memory in static TLS block
```

This should be fixed by adding the following to your file: `~/.bashrc`

```
echo 'export LD_PRELOAD=/usr/lib/aarch64-linux-gnu/libgomp.so.1' >> ~/.bashrc
```

If you have already encountered this error, you can fix it by running the following command:

```
export LD_PRELOAD=/usr/lib/aarch64-linux-gnu/libgomp.so.1
rm ~/.cache/gstreamer-1.0/registry.aarch64.bin
```