KAMAMI

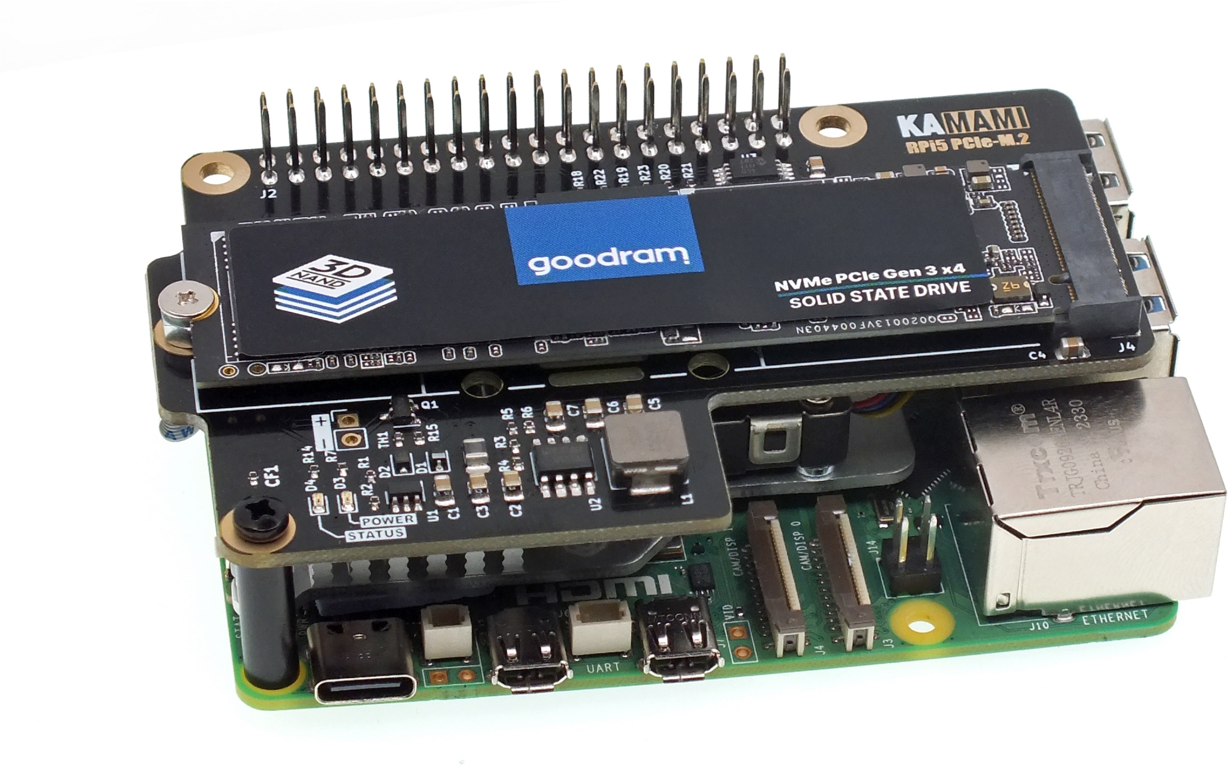
KAmodRPi5 PCIe-M.2

Rev. 20241026155347
Źródło: https://wiki.kamamilabs.com/index.php/KAmodRPi5_PCIe-M.2

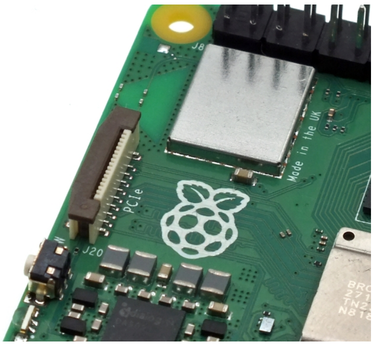
Spis treści

Description

KAmodRPi5 PCIe-M.2 - NVME M.2 Drive Adapter for Raspberry Pi 5



The Raspberry Pi 5 computer provides a **PCI Express** interface, making it compatible with many of the latest computer technology solutions. To maintain the compact dimensions of the Raspberry series, the interface was led out to a non-standard, miniature FFC connector with 16 pins and a 0.5 mm pitch, marked on the board as **PCIe**.



The new connector allows you to connect modern SSD drives that support the fast and reliable NVME protocol. This solution makes Raspberry Pi an ideal multimedia center or file server while maintaining small size and energy efficiency.

The interface used works in the PCI Express Gen2 x1 version and allows you to connect an NVME drive with an **M.2 M-KEY** type connector to the Raspberry Pi 5 computer. This requires the use of the **KAmodRPi5 PCIe-M.2** adapter.

For this set to work correctly, the latest version of software for the Raspberry Pi 5 computer is required.
Only after all updates have been completed can you install the operating system image on the NVME M.2 drive and set the boot sequence to boot from the NVME M.2 drive. All necessary steps, in the order they are performed, are described in the instructions.

Basic parameters

- The adapter is designed for easy installation on the Raspberry Pi 5 board
- The connection to the Raspberry Pi 5 computer is via a 40-pin GPIO connector and a 16/0.5 FFC ribbon (optionally, it is possible to connect only using FFC ribbon)
- It takes up the typical space reserved for standard HAT expansion modules
- It allows you to connect additional HAT modules to the 40-pin GPIO connector
- It allows you to connect an NVME drive with an M.2 type connector (also called NGFF)
- Communication via the PCI Express Gen2 x1 interface (optionally also Gen3 x1)
- It allows you to mount a 2230, 2242, 2260 or 2280 size drive
- It provides a drive power supply voltage of 3.3V and a maximum current of 3A (current efficiency depends on the power of the power supply supplying power to the Raspberry Pi 5)
- It draws power from the Raspberry computer Pi 5, so you should use a power supply for Raspberry Pi 5 with the appropriate power (recommended - at least 5.1V, 3.0A, 15W)
- LED indicators signaling correct power supply and disk activity
- Dimensions 56x85 mm (56x89 mm in the middle, where there is a hole for the mounting screw of the 2280 disk)
- The design of the adapter does not block the possibility of using a radiator with a fan dedicated to Raspberry Pi 5
- **Not every NVME M.2 disk is compatible with the Raspberry Pi 5 computer**

Tested and working disk models include:

- Goodram PX500 (256 GB, 2280)
- Samsung 980 (500 GB, 2280)
- Samsung PM991a (256GB, 2242)
- Patriot P310 (240GB, 2280)

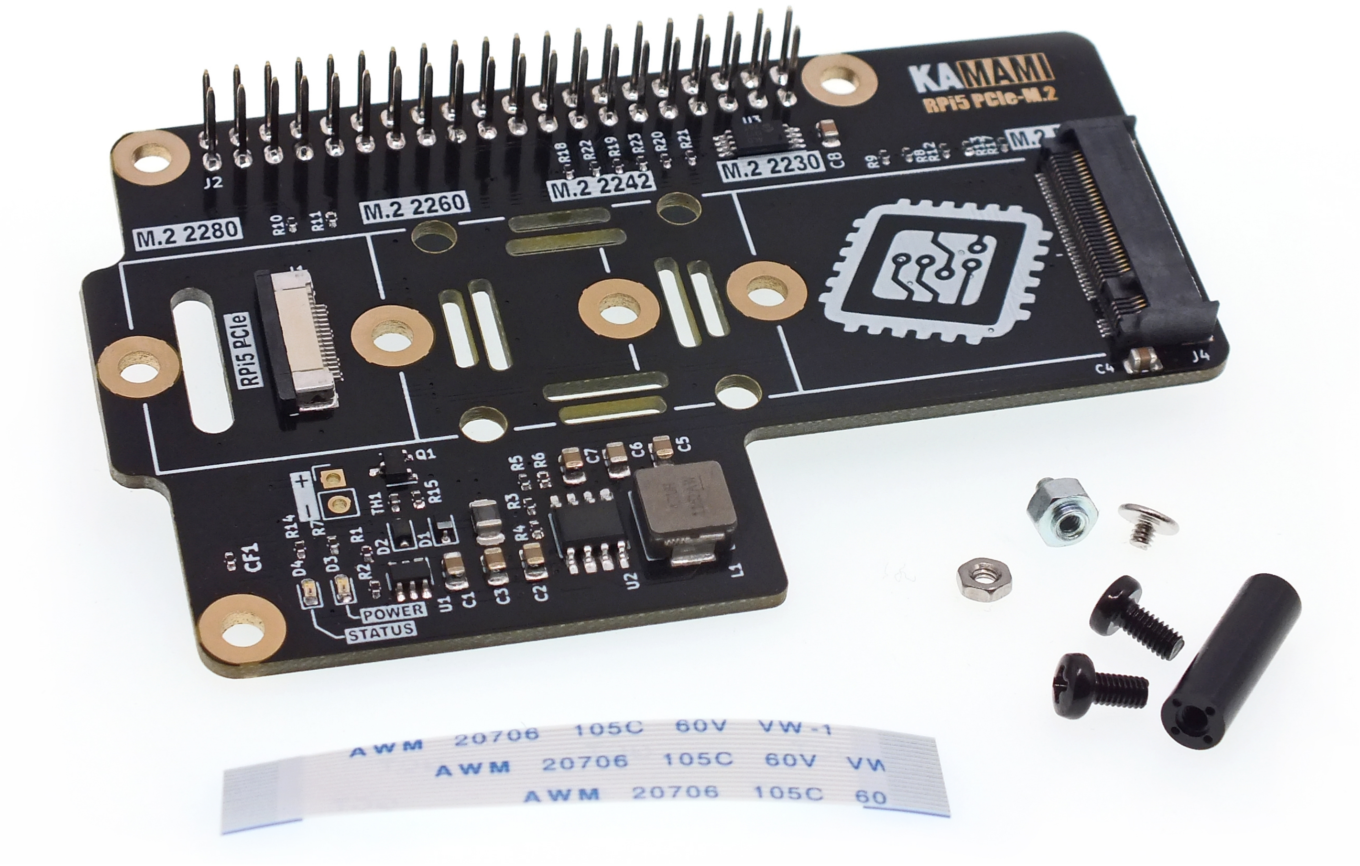
Incompatible drives (work only as additional memory, not suitable for system boot):

- Goodram PX600 (256GB, 2280)
- WD SN530 (256GB, 2242)

Standard equipment

Code	Description
• **KAmodRPi5 PCIe-M.2 Adapter** • **1 x FPC Ribbon** • **1 x Mounting Kit**	• Assembled and started module • FFC 16/0.5 tape • Set ensuring stable attachment of the adapter (2 M2 screws, 1 spacer), M.2 disk mounting screw

• NVME M.2 disk is not included in the set
• Raspberry Pi 5 is not included in the set



Description of the most important elements

J1 connector - RPi5 PCIe

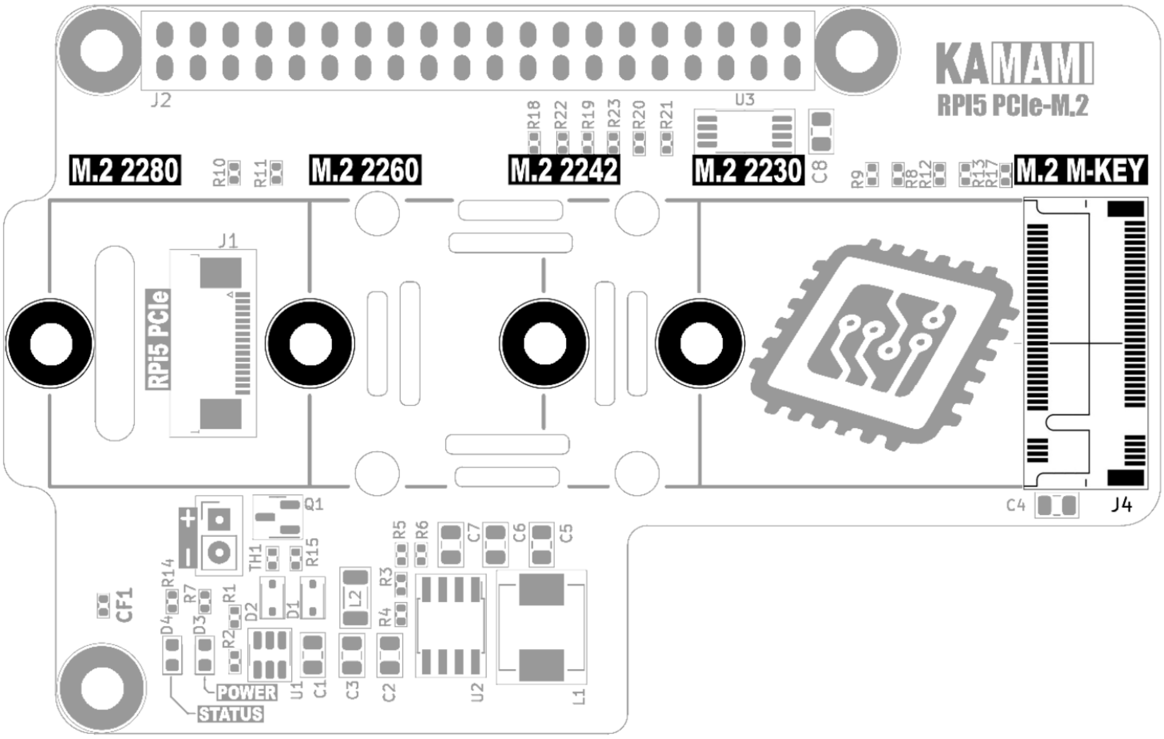
Connector type	Function
FFC 16 pins, 0.5 mm pitch	• Connects the PCIe interface on the Raspberry Pi 5 board to the KAmod RPi5 PCIe-M.2 adapter • Responsible for data transfer, but also provides power to the adapter (the adapter can work when connected to the Raspberry Pi 5 only with the ribbon cable, if the drive does not draw more than 1A) • There is a hole near the connector through which the connecting ribbon cable should be passed • Connects the NVME M.2 drive to the PCI Express Gen2 x1 interface

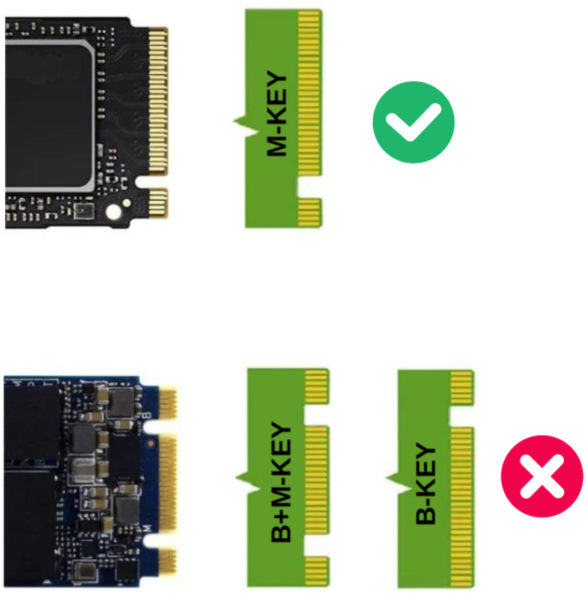
J2 connector - GPIO

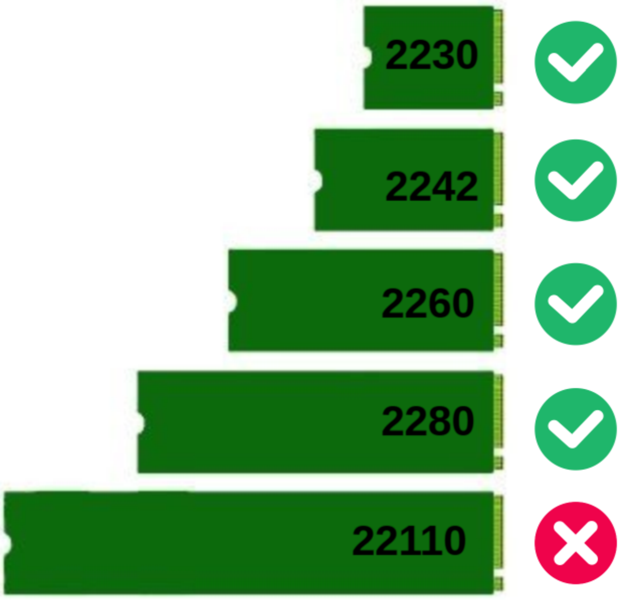
Connector type	Function
Goldpin male-female connector, high	• Connects the KAmod RPi5 PCIe-M.2 adapter to the Raspberry Pi 5 - creates electrical connections and mechanical mounting • Allows you to connect additional mods that extend the capabilities of the Raspberry Pi 5 - does not block the functionality of the GPIO connector • Provides power to the adapter from the Raspberry Pi 5 computer

J4 - M2 M-KEY connector

Connector type	Function
M.2 (NGFF)	• Allows you to connect an NVME drive with an M.2 connector, with the so-called "M" type key (M-KEY) • Set of holes allows for stable mounting of drives in sizes 2230, 2242, 2260 or 2280 • Provides drive power supply voltage of 3.3 V and maximum current of 3 A • Connects NVME M.2 drive to PCI Express Gen2 x1 interface

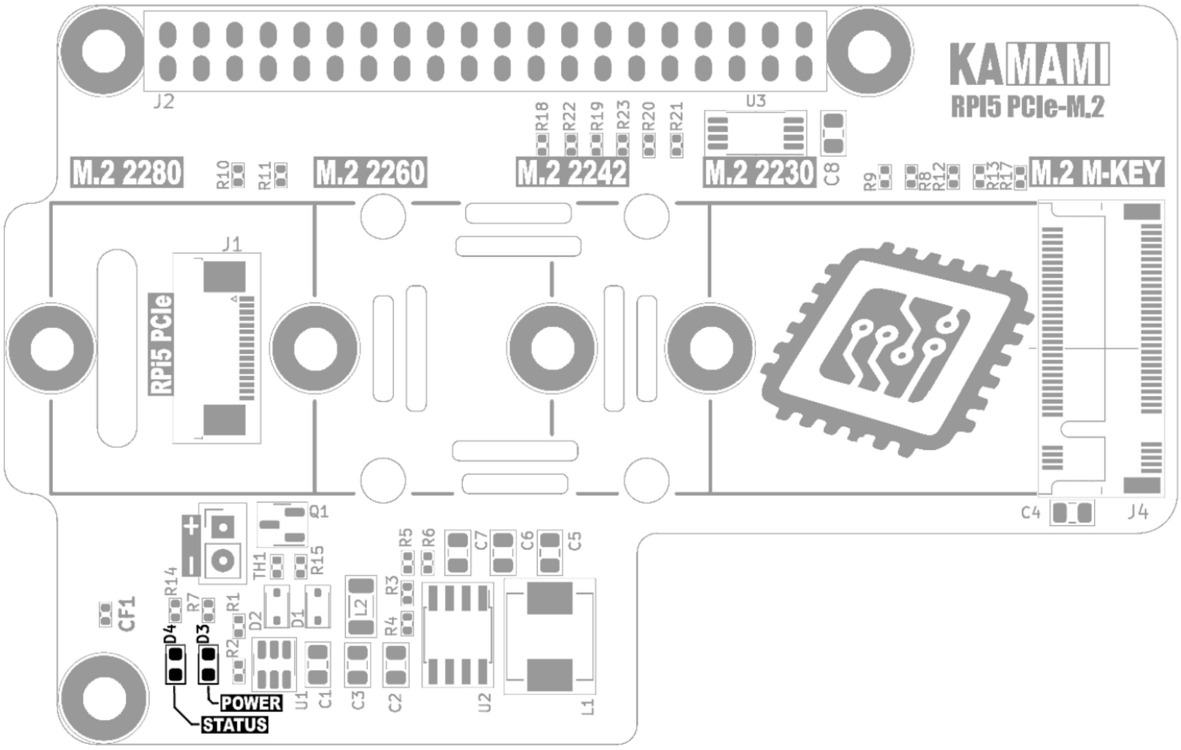






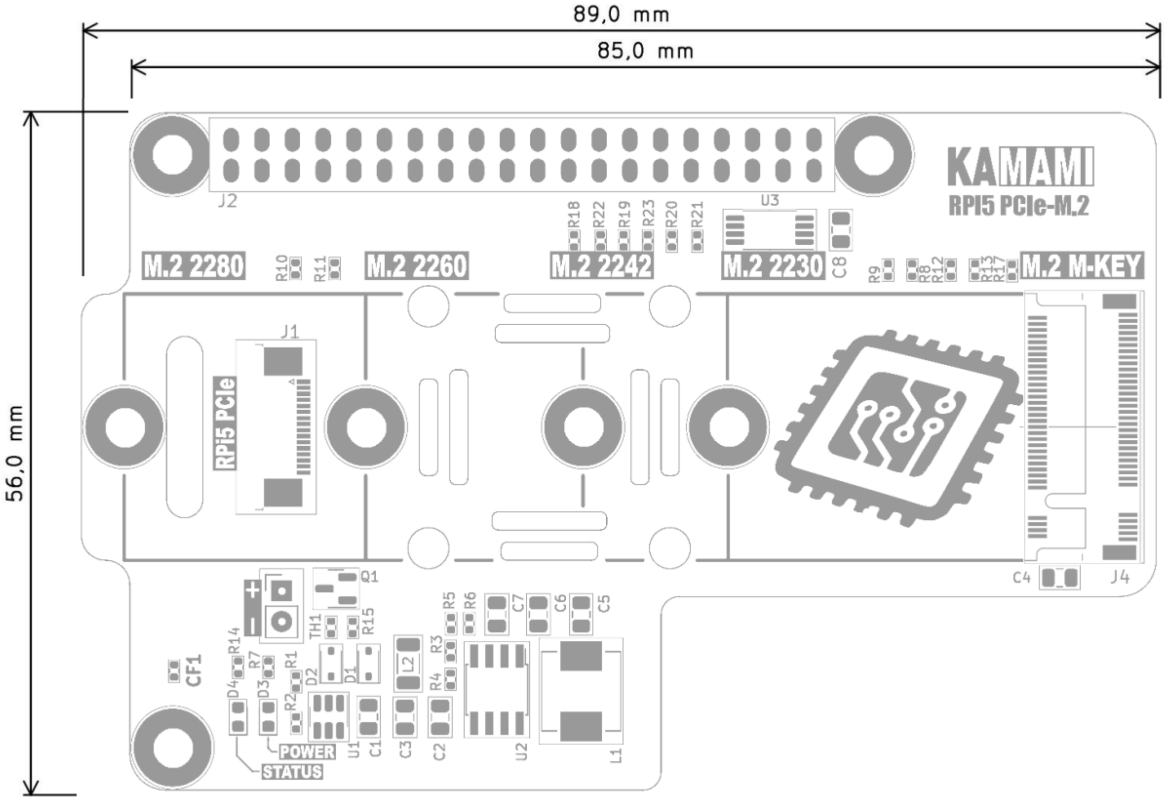
LED indicators

Label	Function
D3 - POWER	• A clear glow indicates correct power parameters of the NVME M.2 drive
D4 - STATUS	• When the light is on, it indicates that the NVME M.2 drive is being read/written/transferred



Dimensions

Dimensions 56x85 mm and match the dimensions of the Raspberry Pi 5 computer. In the middle of the board there is a hole for the 2280 disk mounting screw, at this point the dimensions are 56x89 mm.



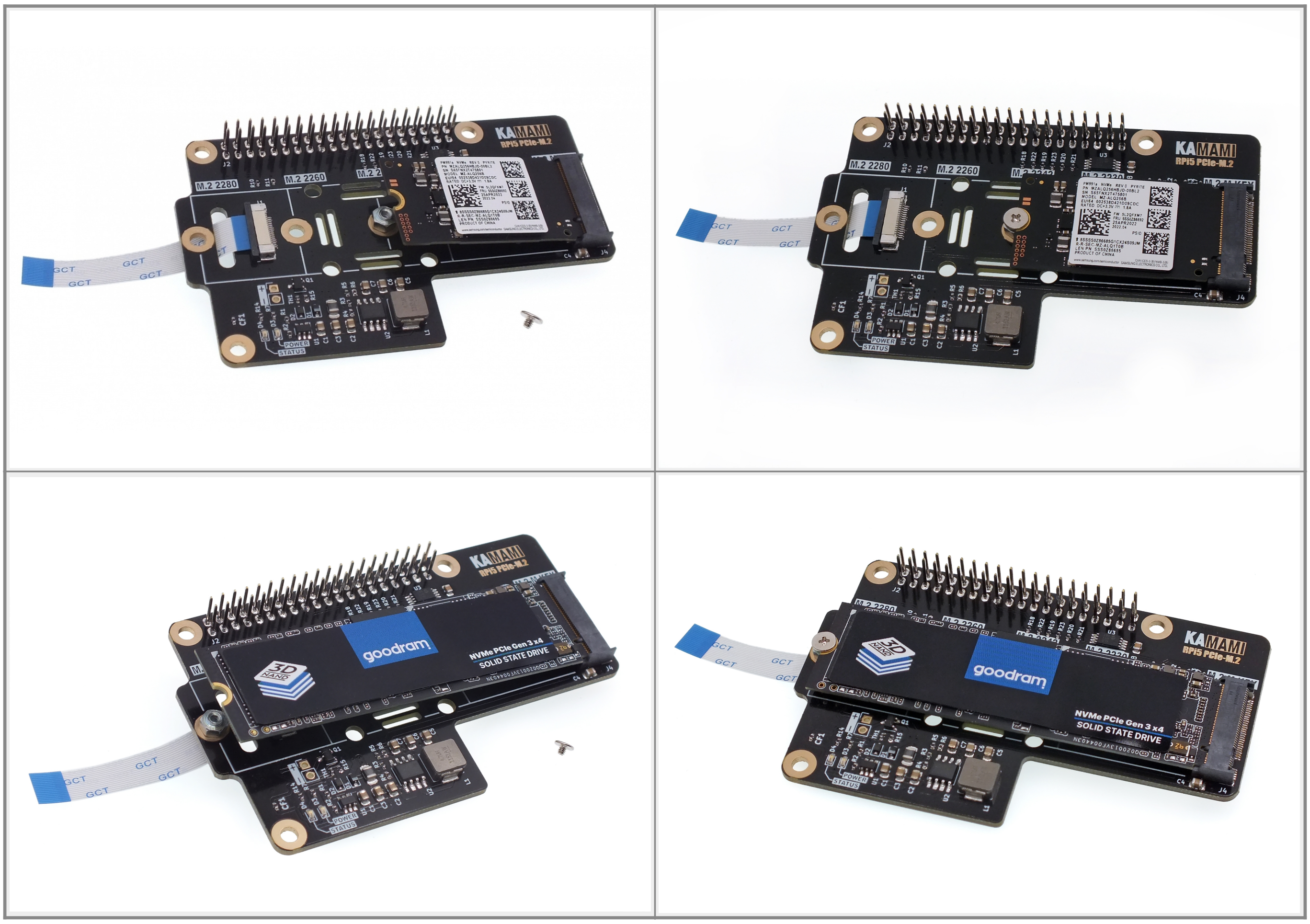
Installing an NVME M.2 Drive in the KAmod RPi5 PCIe-M.2 adapter

Attention! Connecting and disconnecting the NVME M.2 drive should only be done when the Raspberry Pi 5 computer is turned off and disconnected from the power supply.

First, determine the drive size (not capacity) - 2230, 2242, 2260 or 2280 drives are accepted. Once you know the drive size, install the male-female screw securing the M.2 drive in the hole corresponding to the given drive size and tighten it lightly with a nut from the bottom. The holes in the board are slightly larger to allow for positioning the screw in the optimal location for the given drive.

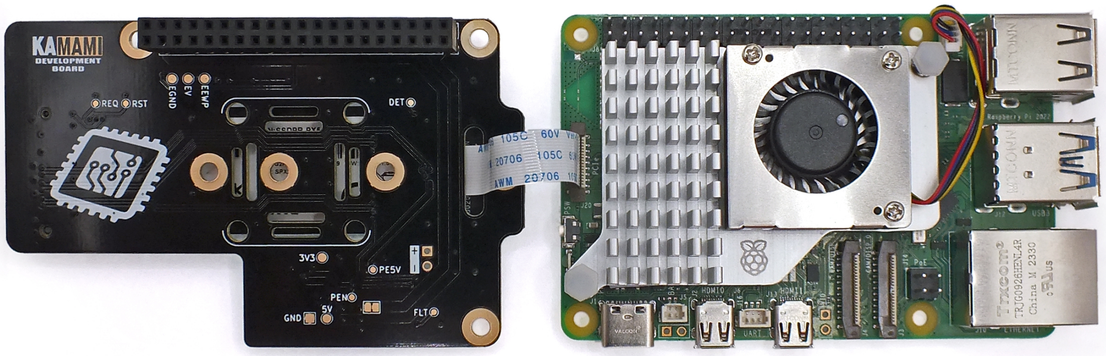
In the second step, connect the FFC 16/0.5 ribbon to the J1 connector (blue markers up) and tighten the lock, and put the other end of the ribbon through the hole at the J1 connector (RPi5 PCIe).

Now you can insert the NVME M.2 drive into the J4 connector (M.2 M-KEY), so that the opposite side of the drive protrudes slightly from the board. Then, gently press the drive onto the mounting screw (and if necessary) move the screw slightly so that it precisely fits into the notch on the drive. Finally, screw in the top screw and tighten the nut a little more from the bottom.

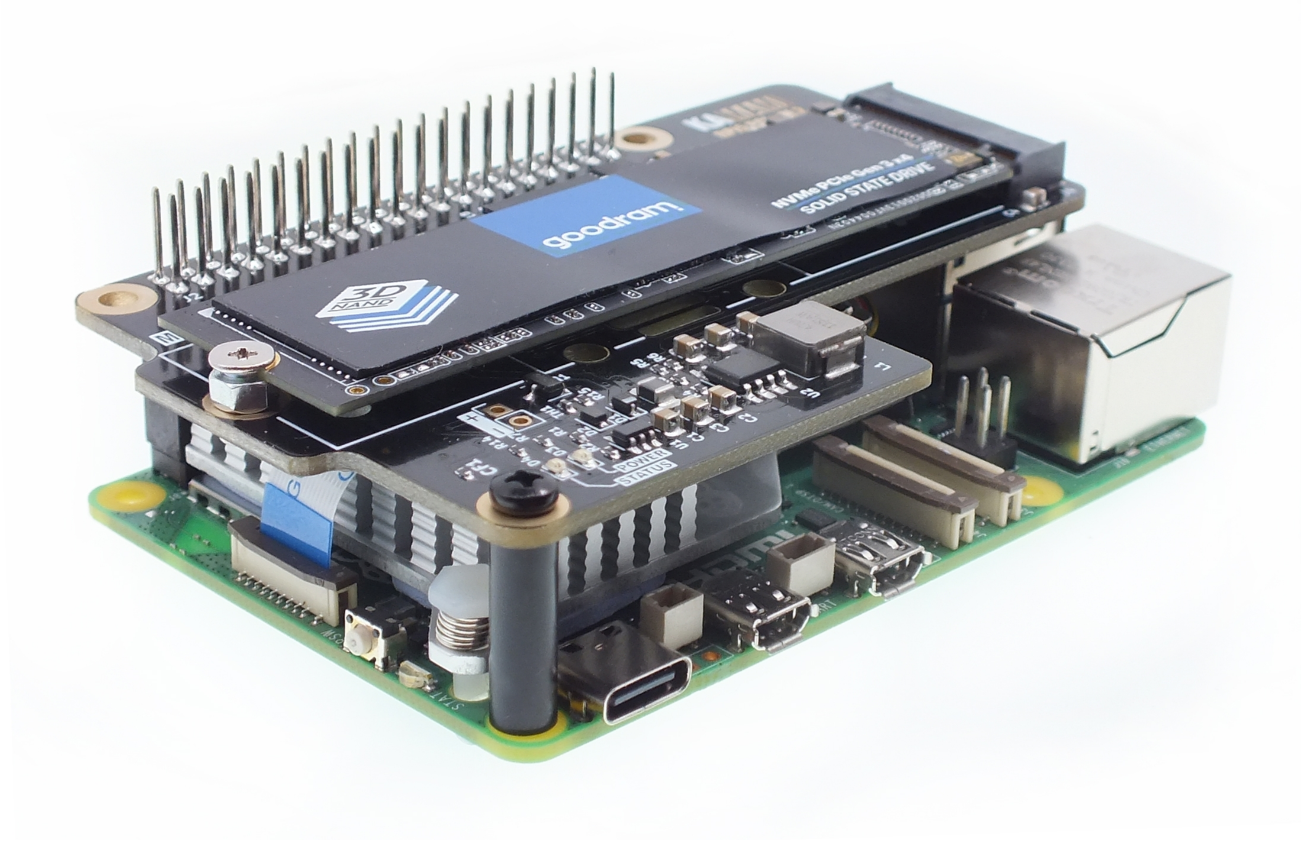


Mounting the KAmod RPi5 PCIe-M.2 adapter on the Raspberry Pi 5 board

The KAmod RPi5 PCIe-M.2 adapter with the ribbon cable mounted should be placed next to the Raspberry Pi 5. Now, you can easily attach the FFC ribbon cable to the PCIe connector on the Raspberry Pi 5 board.



After attaching the ribbon cable, the adapter should be placed above the Raspberry Pi 5 computer, and then the lower side of the J2 connector should be placed on the 40-pin GPIO connector of the computer. Finally, it is worth installing a 16 mm high sleeve at the USBC connector, which will ensure a stable connection of the whole.



Software update process

Performing a full software update involves 3 stages:

- updating the **operating system**,
- updating the **EEPROM memory** content,
- updating the **raspi-config** tool.

Performing all the necessary activities requires a working Raspberry Pi 5 computer with an operating system running, e.g. from a microSD memory card and Internet access. Detailed instructions for preparing the operating system on the memory card can be found in the chapter **Preparing a memory card with an operating system for Raspberry Pi 5**.

Operating system update

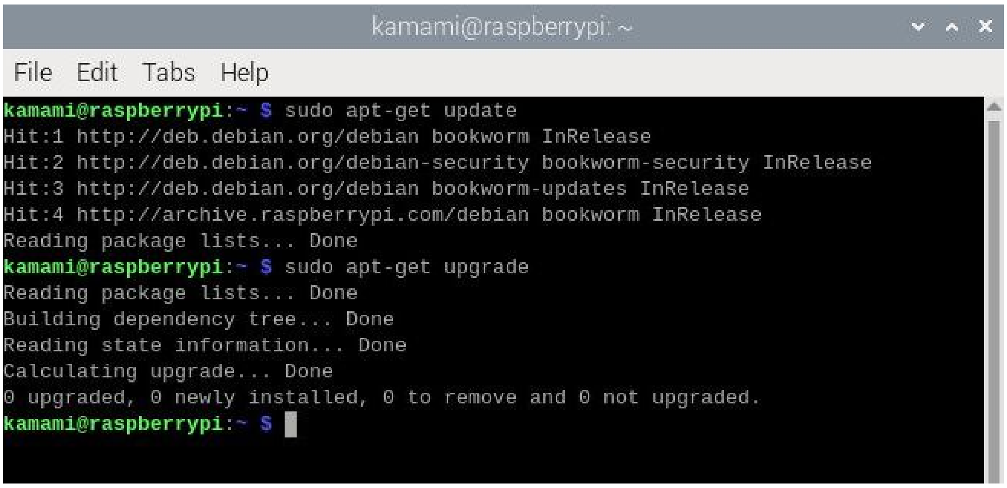
We start Raspberry Pi 5 with the operating system prepared earlier on the memory card (at this stage the KAmod adapterRPi5 PCIE-M.2 should be dismantled). The first boot of the system may take a little longer than usual.
After the system desktop is displayed, open the console window (Terminal), e.g. using the key combination *Ctrl+Alt+T* and enter:

```
sudo apt-get update
```

and then:

```
sudo apt-get upgrade
```

Completion of all actions may take several minutes, depending on the number of components that require updating (the console window may display many more messages than in the example below). Confirm any questions by pressing *Y*(Yes).



```
kamami@raspberrypi:~ $ sudo apt-get update
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://deb.debian.org/debian-security bookworm-security InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Hit:4 http://archive.raspberrypi.com/debian bookworm InRelease
Reading package lists... Done
kamami@raspberrypi:~ $ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
kamami@raspberrypi:~ $
```

Finally, restart the system, e.g. by entering the command:

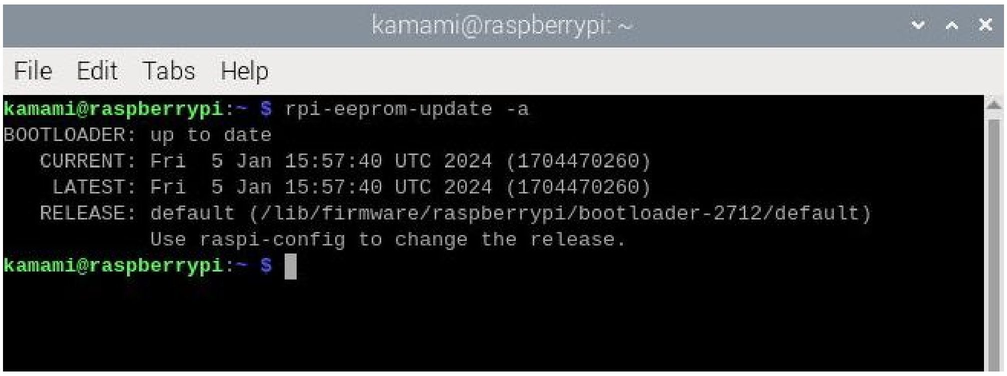
```
sudo reboot
```

Updating the EEPROM memory content

Open the console window (Terminal), e.g. using the key combination *Ctrl+Alt+T* and enter:

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

The console window may display more messages than in the example below if the content requires updating. Confirm any questions by pressing *Y* (Yes).



```
kamami@raspberrypi:~ $ rpi-eeprom-update -a
BOOTLOADER: up to date
   CURRENT: Fri  5 Jan 15:57:40 UTC 2024 (1704470260)
    LATEST: Fri  5 Jan 15:57:40 UTC 2024 (1704470260)
   RELEASE: default (/lib/firmware/raspberrypi/bootloader-2712/default)
            Use raspi-config to change the release.
kamami@raspberrypi:~ $
```

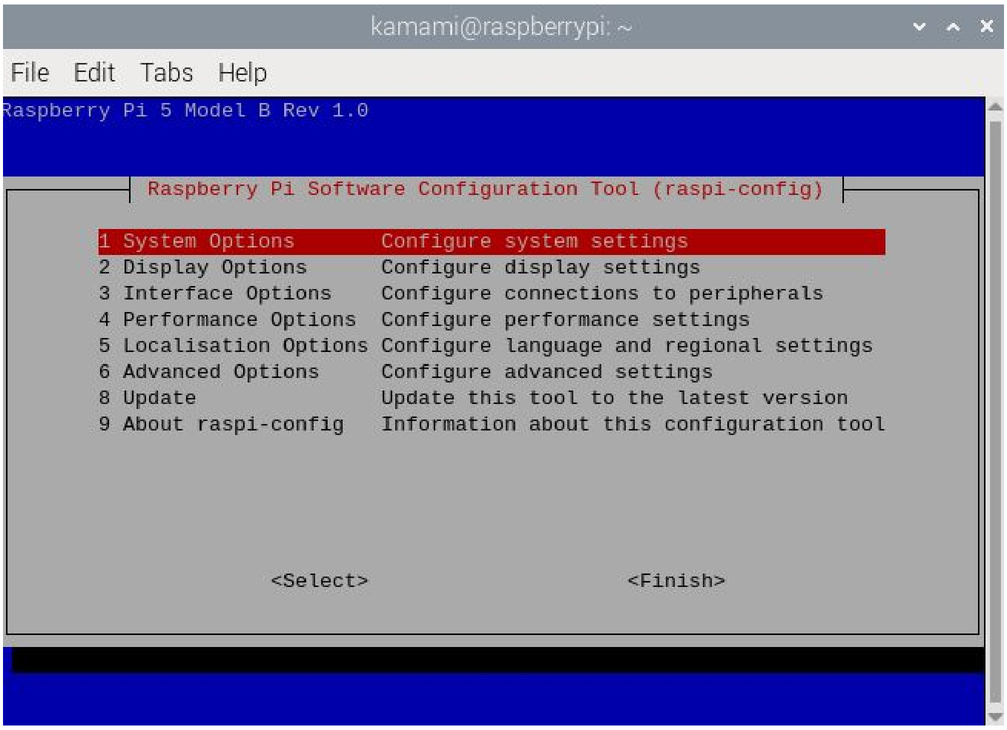
After completing the actions, it is necessary to restart the system, which we can invoke, for example, by entering the command:

```
sudo reboot
```

Updating the raspi-config tool

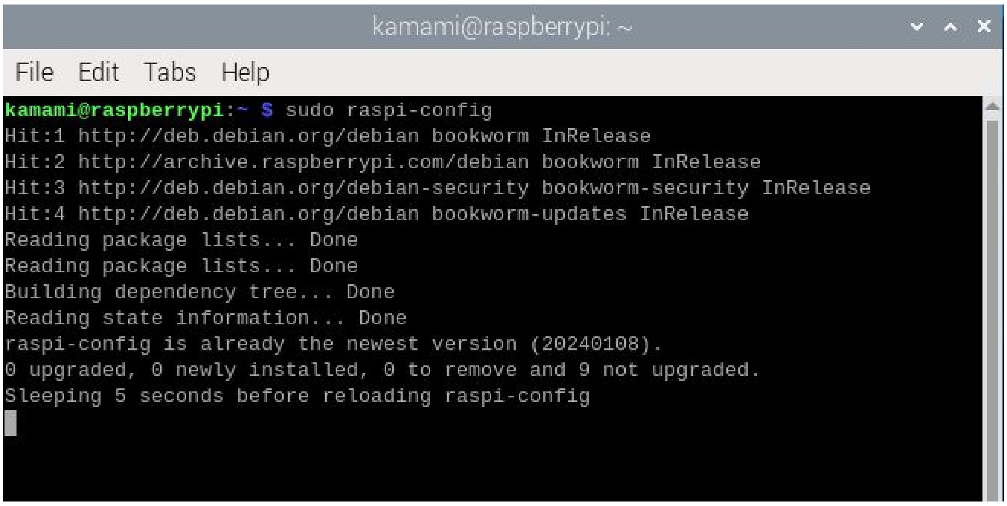
Open the console window (Terminal), for example, using the key combination *Ctrl+Alt+T* and enter:

```
sudo raspi-config
```



```
Raspberry Pi 5 Model B Rev 1.0

        Raspberry Pi Software Configuration Tool (raspi-config)

  1 System Options       Configure system settings
  2 Display Options      Configure display settings
  3 Interface Options    Configure connections to peripherals
  4 Performance Options  Configure performance settings
  5 Localisation Options Configure language and regional settings
  6 Advanced Options     Configure advanced settings
  8 Update               Update this tool to the latest version
  9 About raspi-config   Information about this configuration tool

               <Select>                     <Finish>
```

Using the arrow keys we indicate the line: **8 Update**. We wait for all actions to finish. After a while, the window will open again.



```
kamami@raspberrypi:~ $ sudo raspi-config
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://archive.raspberrypi.com/debian bookworm InRelease
Hit:3 http://deb.debian.org/debian-security bookworm-security InRelease
Hit:4 http://deb.debian.org/debian bookworm-updates InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
raspi-config is already the newest version (20240108).
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
Sleeping 5 seconds before reloading raspi-config
```

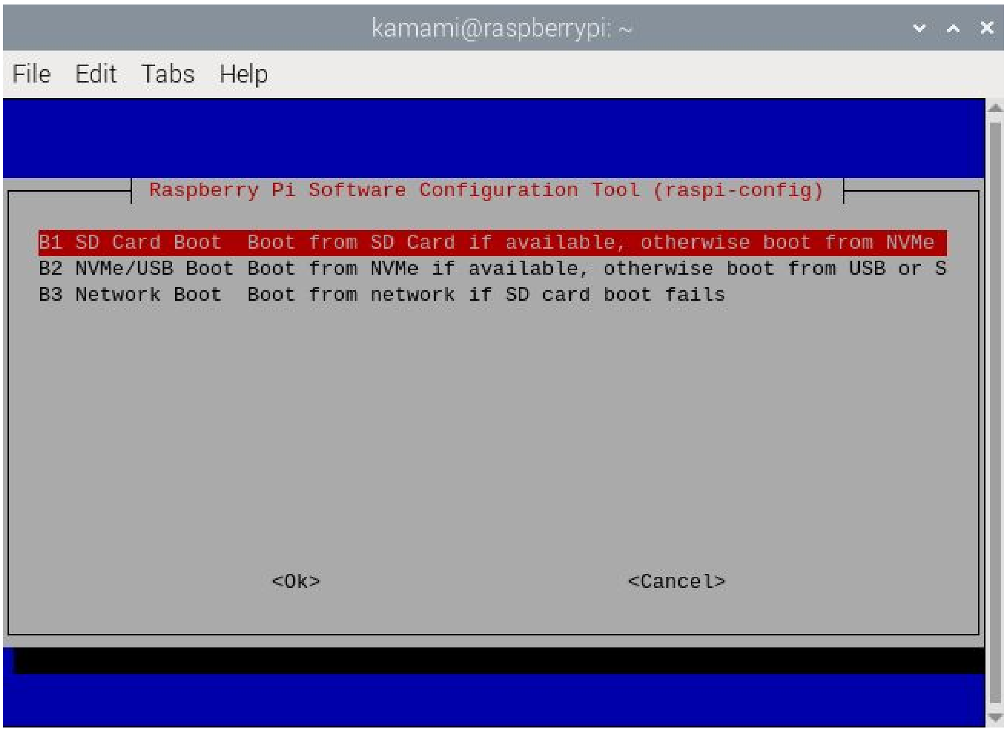
Setting the boot sequence

The boot sequence determines from which media (memory card, USB stick, NVME disk) and in what order Raspberry will start the operating system.

The sequence can be configured using the raspi-config tool, which we run by typing in the console:

```
sudo raspi-config
```

Now we select the line: ***6 Advanced Options***, and then ***A4 Boot Order***. There are three options to choose from, as shown in the screenshot below:



```
Raspberry Pi Software Configuration Tool (raspi-config)

B1 SD Card Boot  Boot from SD Card if available, otherwise boot from NVMe
B2 NVMe/USB Boot Boot from NVMe if available, otherwise boot from USB or S
B3 Network Boot  Boot from network if SD card boot fails

                <Ok>                          <Cancel>
```

Option one: ***B1 SD Card Boot*** - causes the system to be loaded from the memory card first, and if the card is not detected, the system will be loaded from the NVME drive. This is a safe setting and should be selected.

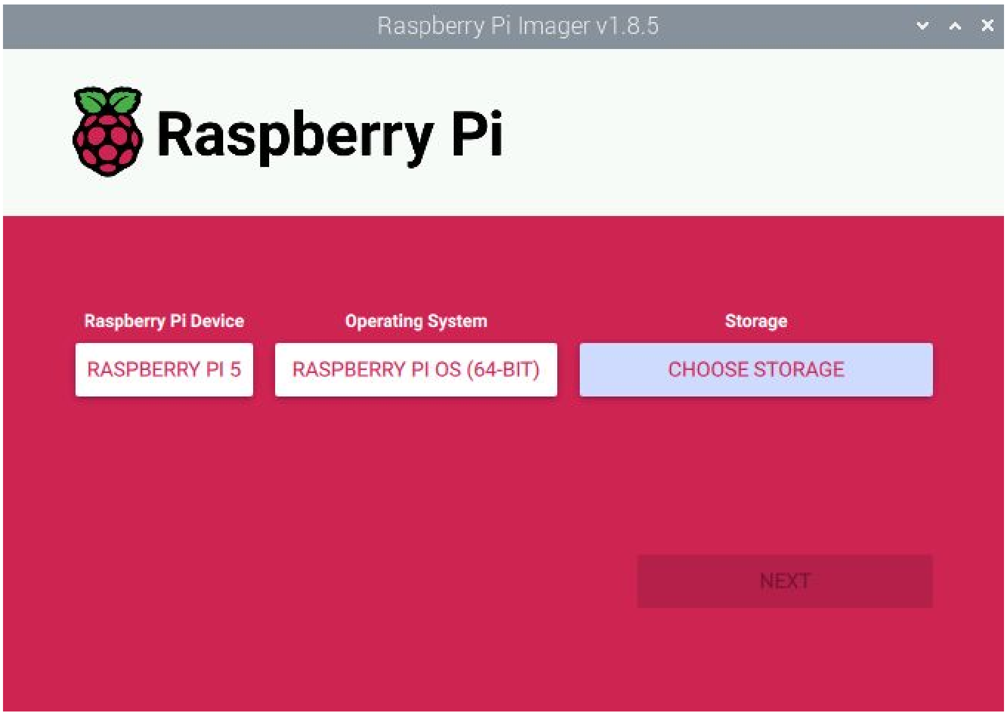
Confirm the changes, and in the initial window select the command ***<Finish>*** (at the bottom of the window). When asked to restart the system, select ***<Yes>***'.

Installing the system on the NVME M.2 drive

Install the KAmod RPi5 PCIe-M.2 adapter with the mounted NVME M.2 drive. Detailed instructions for these steps can be found in the chapter **Mounting the KAmod RPi5 PCIe-M.2 adapter on the Raspberry Pi 5 board *and*** *Mounting the NVME M.2 drive in the KAmod RPi5 PCIe-M.2 adapter.*

We start Raspberry Pi 5, open the console window (Terminal), e.g. using the key combination *Ctrl+Alt+T* and enter:

```
sudo rpi-imager
```



In the window that appears, select:

- computer model (Raspberry Pi Device): ***RASPBERRY PI 5***,
- operating system (Operating System): ***RASPBERRY PI OS (64-BIT)***,
- disk (Storage): here we indicate the NVME M.2 disk, which was mounted in the KAmod RPi5 PCIe-M.2 adapter.

The further process of installing the operating system image is similar to that of the memory card and has been described in detail in the chapter '*Preparing a memory card with an operating system for Raspberry Pi 5*.

Starting the system from the NVME M.2 drive

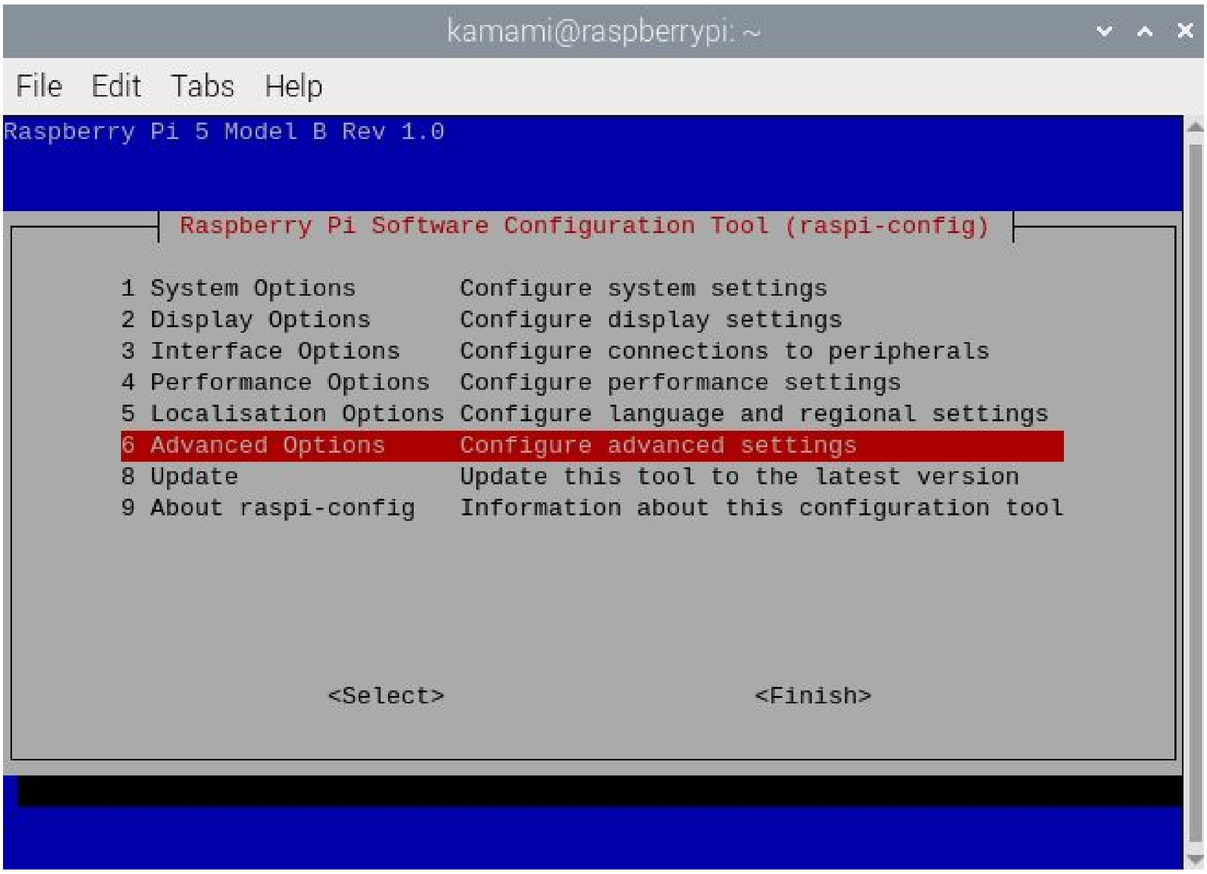
When we have a mounted NVME M.2 drive with an installed operating system image, before starting the Raspberry Pi 5 computer, we remove the memory card from the dedicated slot - it will no longer be needed. The memory card can be removed/mounted only when the computer is turned off and disconnected from the power supply.

The first start of the new operating system will take a little longer than the next ones, but after a while the desktop of the ready-to-work Raspbian installed on the NVME M.2 drive will appear. This is a new operating system and does not contain the changes we made to the system on the memory card. Therefore, we need to update the operating system and the raspi-config tool, as described earlier. We do not need to update the EEPROM content - it has not changed after changing the operating system.

Finally, using the raspi-config tool, set the boot sequence to boot from the NVME disk, which will speed up the system startup a bit, because the attempt to read from the memory card will be skipped.

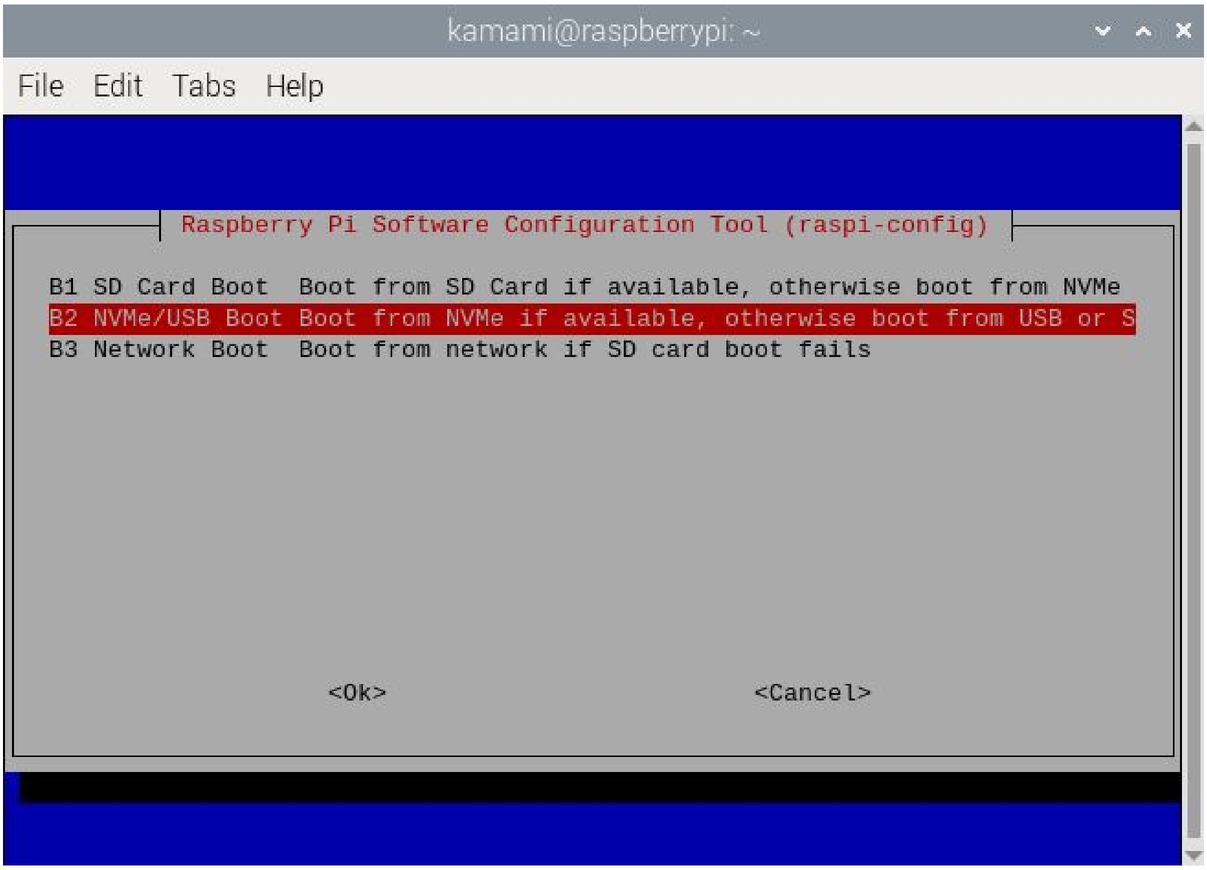
Open the console window (Terminal), e.g. using the key combination *Ctrl+Alt+T* and enter:

```
sudo raspi-config
```



```
Raspberry Pi 5 Model B Rev 1.0

       Raspberry Pi Software Configuration Tool (raspi-config)

      1 System Options       Configure system settings
      2 Display Options      Configure display settings
      3 Interface Options    Configure connections to peripherals
      4 Performance Options  Configure performance settings
      5 Localisation Options Configure language and regional settings
      6 Advanced Options     Configure advanced settings
      8 Update               Update this tool to the latest version
      9 About raspi-config   Information about this configuration tool

                   <Select>                     <Finish>
```

Now select the line: ***6 Advanced Options***, and then ***A4 Boot Order***. We select the option: ***B2 NVMe/USB Boot***, which causes the system to first load from the NVME disk.



```
Raspberry Pi Software Configuration Tool (raspi-config)

B1 SD Card Boot   Boot from SD Card if available, otherwise boot from NVMe
B2 NVMe/USB Boot Boot from NVMe if available, otherwise boot from USB or S
B3 Network Boot   Boot from network if SD card boot fails

                  <Ok>                          <Cancel>
```

We confirm the changes, and in the initial window select the command ***<Finish>*** (at the bottom of the window). When asked to restart the system, select ***<Yes>***'.

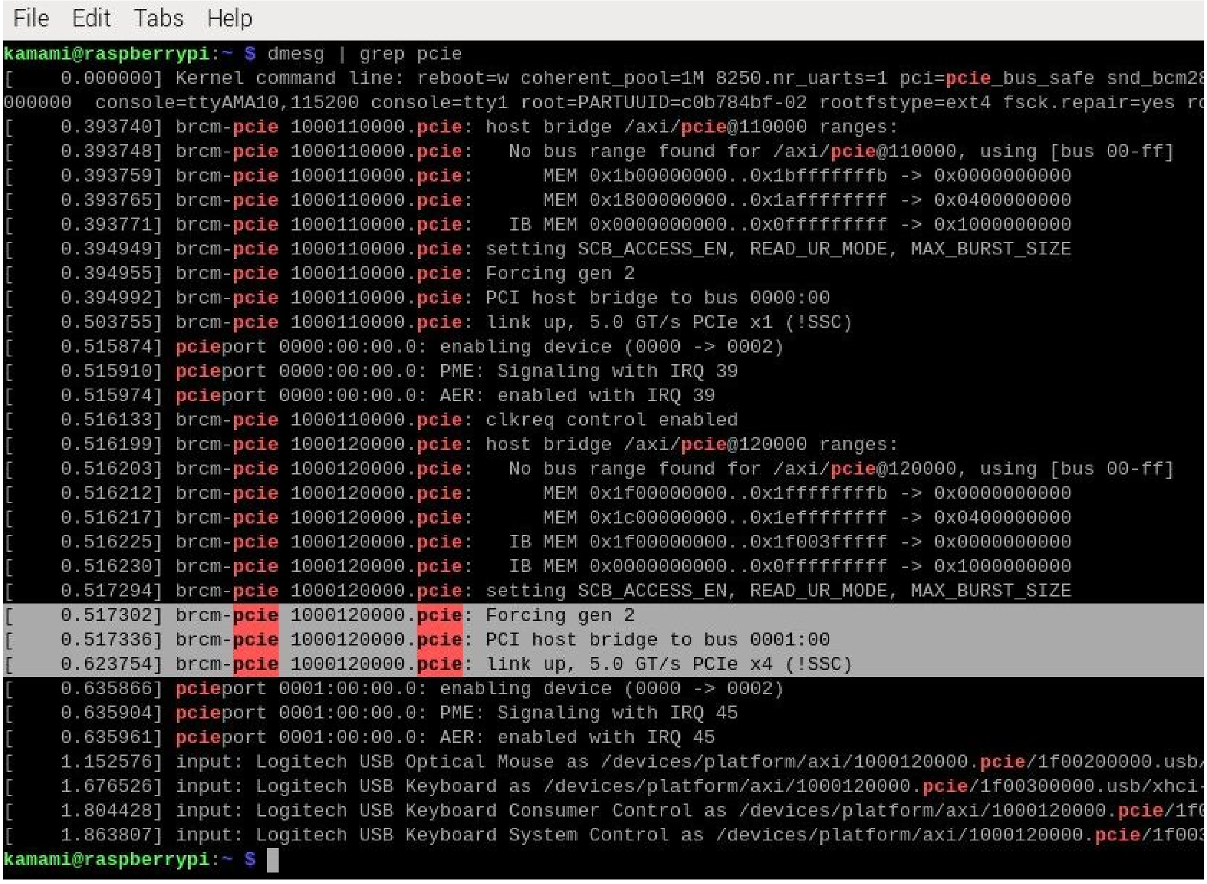
Additional information

The NVME M.2 disk is not visible in the system

In a situation where the disk is not visible in the system, e.g. the rpi-imager program does not have the attached disk among the available media, you should first check the correct operation of the PCIe interface. This can be done by analyzing the system boot messages. You should open the console window (Terminal), e.g. using the key combination *Ctrl+Alt+T* and enter:

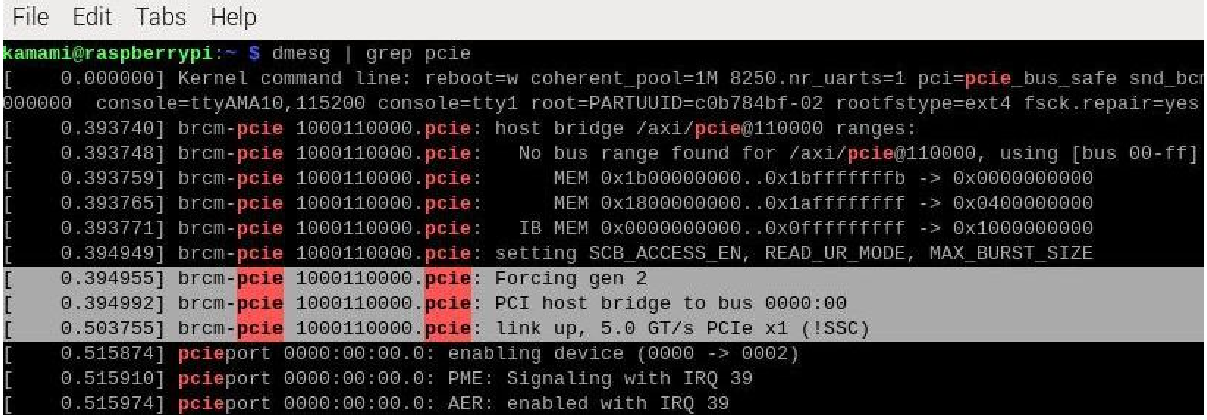
```
dmesg | grep pcie
```

You will see content similar to that in the screenshot below:



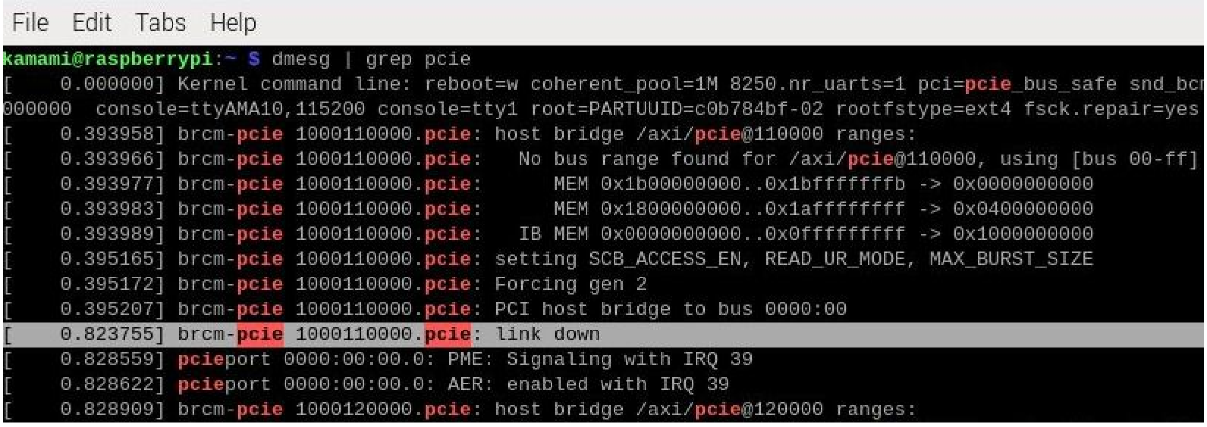
```
kamami@raspberrypi:~ $ dmesg | grep pcie
[    0.000000] Kernel command line: reboot=w coherent_pool=1M 8250.nr_uarts=1 pci=pcie_bus_safe snd_bcm28
000000  console=ttyAMA10,115200 console=tty1 root=PARTUUID=c0b784bf-02 rootfstype=ext4 fsck.repair=yes ro
[    0.393740] brcm-pcie 1000110000.pcie: host bridge /axi/pcie@110000 ranges:
[    0.393748] brcm-pcie 1000110000.pcie:   No bus range found for /axi/pcie@110000, using [bus 00-ff]
[    0.393759] brcm-pcie 1000110000.pcie:      MEM 0x1b00000000..0x1bfffffffb -> 0x0000000000
[    0.393765] brcm-pcie 1000110000.pcie:      MEM 0x1800000000..0x1affffffff -> 0x0400000000
[    0.393771] brcm-pcie 1000110000.pcie:   IB MEM 0x0000000000..0x0fffffffff -> 0x1000000000
[    0.394949] brcm-pcie 1000110000.pcie: setting SCB_ACCESS_EN, READ_UR_MODE, MAX_BURST_SIZE
[    0.394955] brcm-pcie 1000110000.pcie: Forcing gen 2
[    0.394992] brcm-pcie 1000110000.pcie: PCI host bridge to bus 0000:00
[    0.503755] brcm-pcie 1000110000.pcie: link up, 5.0 GT/s PCIe x1 (!SSC)
[    0.515874] pcieport 0000:00:00.0: enabling device (0000 -> 0002)
[    0.515910] pcieport 0000:00:00.0: PME: Signaling with IRQ 39
[    0.515974] pcieport 0000:00:00.0: AER: enabled with IRQ 39
[    0.516133] brcm-pcie 1000110000.pcie: clkreq control enabled
[    0.516199] brcm-pcie 1000120000.pcie: host bridge /axi/pcie@120000 ranges:
[    0.516203] brcm-pcie 1000120000.pcie:   No bus range found for /axi/pcie@120000, using [bus 00-ff]
[    0.516212] brcm-pcie 1000120000.pcie:      MEM 0x1f00000000..0x1ffffffffb -> 0x0000000000
[    0.516217] brcm-pcie 1000120000.pcie:      MEM 0x1c00000000..0x1effffffff -> 0x0400000000
[    0.516225] brcm-pcie 1000120000.pcie:   IB MEM 0x1f00000000..0x1f003fffff -> 0x0000000000
[    0.516230] brcm-pcie 1000120000.pcie:   IB MEM 0x0000000000..0x0fffffffff -> 0x1000000000
[    0.517294] brcm-pcie 1000120000.pcie: setting SCB_ACCESS_EN, READ_UR_MODE, MAX_BURST_SIZE
[    0.517302] brcm-pcie 1000120000.pcie: Forcing gen 2
[    0.517336] brcm-pcie 1000120000.pcie: PCI host bridge to bus 0001:00
[    0.623754] brcm-pcie 1000120000.pcie: link up, 5.0 GT/s PCIe x4 (!SSC)
[    0.635866] pcieport 0001:00:00.0: enabling device (0000 -> 0002)
[    0.635904] pcieport 0001:00:00.0: PME: Signaling with IRQ 45
[    0.635961] pcieport 0001:00:00.0: AER: enabled with IRQ 45
[    1.152576] input: Logitech USB Optical Mouse as /devices/platform/axi/1000120000.pcie/1f00200000.usb/
[    1.676526] input: Logitech USB Keyboard as /devices/platform/axi/1000120000.pcie/1f00300000.usb/xhci-
[    1.804428] input: Logitech USB Keyboard Consumer Control as /devices/platform/axi/1000120000.pcie/1f0
[    1.863807] input: Logitech USB Keyboard System Control as /devices/platform/axi/1000120000.pcie/1f003
kamami@raspberrypi:~ $
```

You can see information about the PCIe x4 interface, which is the internal interface of the Raspberry Pi 5 computer and is not available to the user. There should also be information about the PCIe x1 interface, which is responsible for connecting to the NVME M.2 drive:



```
kamami@raspberrypi:~ $ dmesg | grep pcie
[    0.000000] Kernel command line: reboot=w coherent_pool=1M 8250.nr_uarts=1 pci=pcie_bus_safe snd_bc
000000  console=ttyAMA10,115200 console=tty1 root=PARTUUID=c0b784bf-02 rootfstype=ext4 fsck.repair=yes
[    0.393740] brcm-pcie 1000110000.pcie: host bridge /axi/pcie@110000 ranges:
[    0.393748] brcm-pcie 1000110000.pcie:   No bus range found for /axi/pcie@110000, using [bus 00-ff]
[    0.393759] brcm-pcie 1000110000.pcie:      MEM 0x1b00000000..0x1bfffffffb -> 0x0000000000
[    0.393765] brcm-pcie 1000110000.pcie:      MEM 0x1800000000..0x1affffffff -> 0x0400000000
[    0.393771] brcm-pcie 1000110000.pcie:   IB MEM 0x0000000000..0x0fffffffff -> 0x1000000000
[    0.394949] brcm-pcie 1000110000.pcie: setting SCB_ACCESS_EN, READ_UR_MODE, MAX_BURST_SIZE
[    0.394955] brcm-pcie 1000110000.pcie: Forcing gen 2
[    0.394992] brcm-pcie 1000110000.pcie: PCI host bridge to bus 0000:00
[    0.503755] brcm-pcie 1000110000.pcie: link up, 5.0 GT/s PCIe x1 (!SSC)
[    0.515874] pcieport 0000:00:00.0: enabling device (0000 -> 0002)
[    0.515910] pcieport 0000:00:00.0: PME: Signaling with IRQ 39
[    0.515974] pcieport 0000:00:00.0: AER: enabled with IRQ 39
```

In a situation where the PCIe x1 interface is not properly started, we will find an entry similar to this:



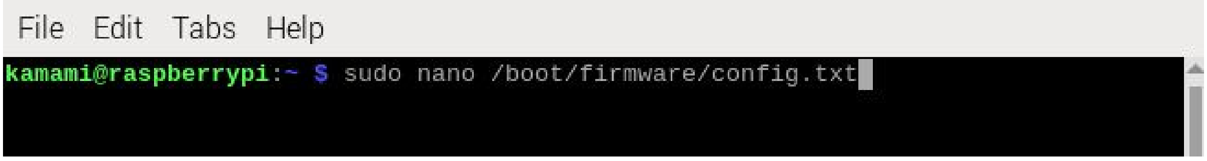
```
kamami@raspberrypi:~ $ dmesg | grep pcie
[    0.000000] Kernel command line: reboot=w coherent_pool=1M 8250.nr_uarts=1 pci=pcie_bus_safe snd_bc
000000  console=ttyAMA10,115200 console=tty1 root=PARTUUID=c0b784bf-02 rootfstype=ext4 fsck.repair=yes
[    0.393958] brcm-pcie 1000110000.pcie: host bridge /axi/pcie@110000 ranges:
[    0.393966] brcm-pcie 1000110000.pcie:   No bus range found for /axi/pcie@110000, using [bus 00-ff]
[    0.393977] brcm-pcie 1000110000.pcie:      MEM 0x1b00000000..0x1bfffffffb -> 0x0000000000
[    0.393983] brcm-pcie 1000110000.pcie:      MEM 0x1800000000..0x1affffffff -> 0x0400000000
[    0.393989] brcm-pcie 1000110000.pcie:   IB MEM 0x0000000000..0x0fffffffff -> 0x1000000000
[    0.395165] brcm-pcie 1000110000.pcie: setting SCB_ACCESS_EN, READ_UR_MODE, MAX_BURST_SIZE
[    0.395172] brcm-pcie 1000110000.pcie: Forcing gen 2
[    0.395207] brcm-pcie 1000110000.pcie: PCI host bridge to bus 0000:00
[    0.823755] brcm-pcie 1000110000.pcie: link down
[    0.828559] pcieport 0000:00:00.0: PME: Signaling with IRQ 39
[    0.828622] pcieport 0000:00:00.0: AER: enabled with IRQ 39
[    0.828909] brcm-pcie 1000120000.pcie: host bridge /axi/pcie@120000 ranges:
```

The message "*Link down*" means that connecting the drive failed.

In this situation, it is worth modifying the contents of the configuration file *config.txt.* In the console, type:

```
sudo nano /boot/firmware/config.txt
```

(in earlier versions of the operating system, the file *config.txt* was placed directly in the /boot directory)

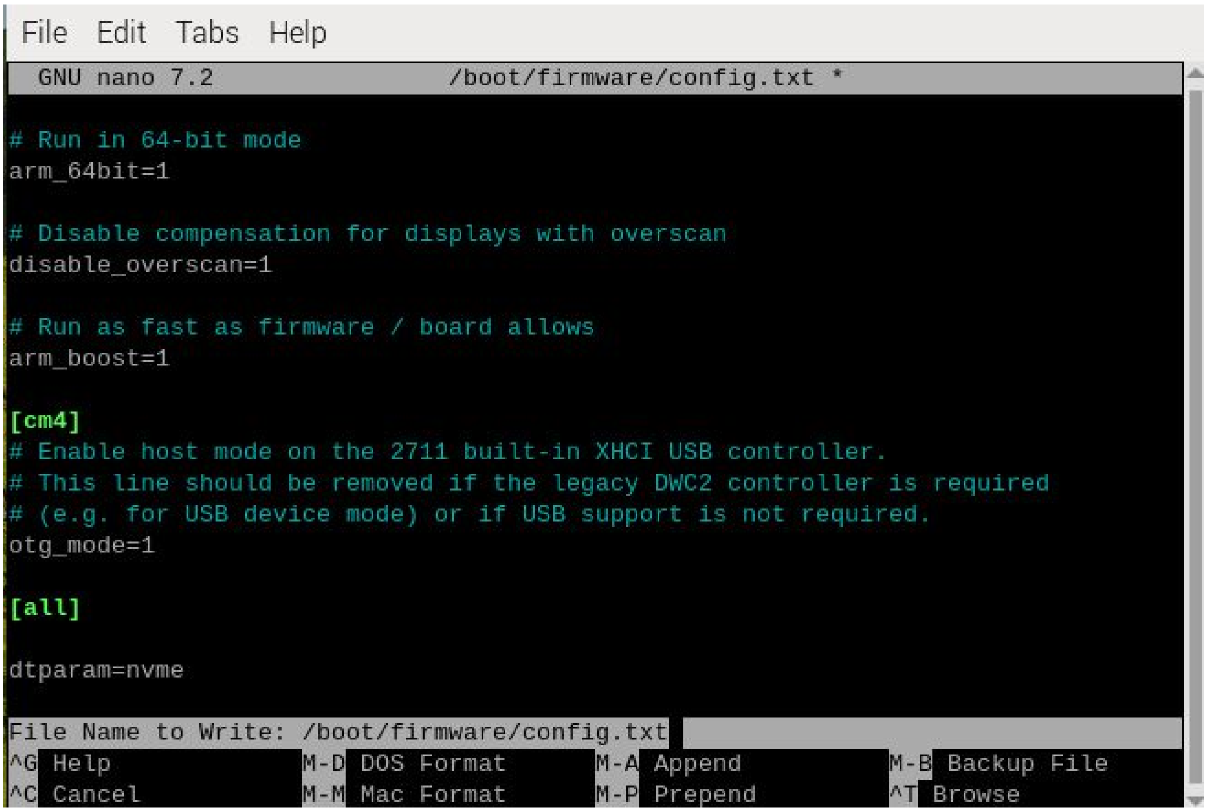


```
kamami@raspberrypi:~ $ sudo nano /boot/firmware/config.txt
```

At the end of the file (scroll to the bottom with the arrows), add the line:

```
dtparam=nvme
```

Then save the changes using the *Ctrl+O* keys, close the editor using the *Ctrl+X* keys, and restart the system.



```
 GNU nano 7.2                 /boot/firmware/config.txt *

# Run in 64-bit mode
arm_64bit=1

# Disable compensation for displays with overscan
disable_overscan=1

# Run as fast as firmware / board allows
arm_boost=1

[cm4]
# Enable host mode on the 2711 built-in XHCI USB controller.
# This line should be removed if the legacy DWC2 controller is required
# (e.g. for USB device mode) or if USB support is not required.
otg_mode=1

[all]

dtparam=nvme

File Name to Write: /boot/firmware/config.txt
^G Help              M-D DOS Format       M-A Append           M-B Backup File
^C Cancel            M-M Mac Format       M-P Prepend          ^T Browse
```

If these actions do not work, it means that the connected drive is not compatible with the Raspberry Pi 5 computer.

Increasing the speed of the PCIe interface

The PCIe interface of the Raspberry Pi 5 computer starts by default in **gen 2** mode, which allows communication with a maximum throughput of 5 GT/s (Gigatransfers per second). There is a way to start **gen 3** mode, which offers a throughput of up to 8 GT/s. To do this, modify the contents of the configuration file *config.txt.*

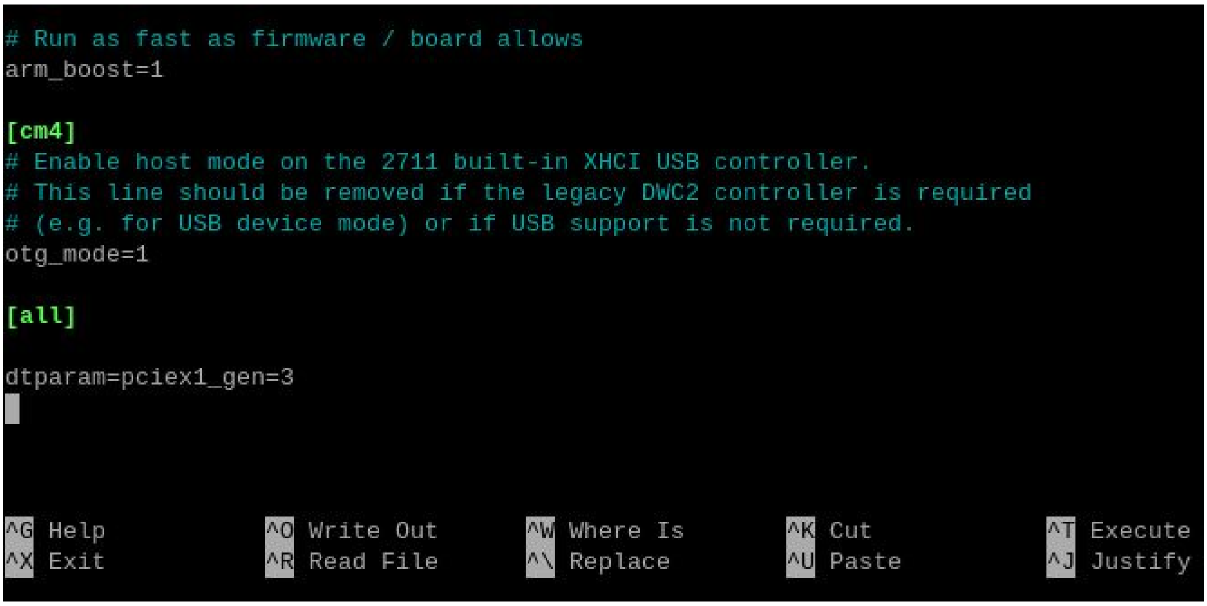
In the console, type:

```
sudo nano /boot/firmware/config.txt
```

(in earlier versions of the operating system, the file *config.txt* was placed directly in the /boot directory).

At the end of the file (scroll down with the arrows) add the line:

```
dtparam=pciex1_gen=3
```



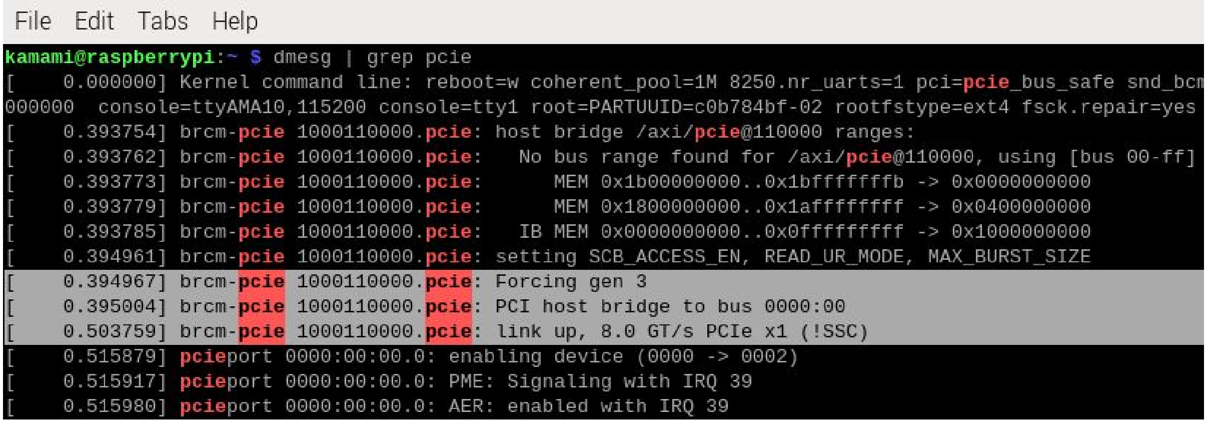
```
# Run as fast as firmware / board allows
arm_boost=1

[cm4]
# Enable host mode on the 2711 built-in XHCI USB controller.
# This line should be removed if the legacy DWC2 controller is required
# (e.g. for USB device mode) or if USB support is not required.
otg_mode=1

[all]

dtparam=pciex1_gen=3
```

Then save the changes using the *Ctrl+O* keys, close the editor using the *Ctrl+X* keys and restart the system.

To check if the modification worked, you can analyze the system boot messages. Open the console window (Terminal), e.g. using the *Ctrl+Alt+T* key combination and enter:

```
dmesg | grep pcie
```

You will see content similar to the one in the screenshot below:



```
kamami@raspberrypi:~ $ dmesg | grep pcie
[    0.000000] Kernel command line: reboot=w coherent_pool=1M 8250.nr_uarts=1 pci=pcie_bus_safe snd_bcm
000000  console=ttyAMA10,115200 console=tty1 root=PARTUUID=c0b784bf-02 rootfstype=ext4 fsck.repair=yes
[    0.393754] brcm-pcie 1000110000.pcie: host bridge /axi/pcie@110000 ranges:
[    0.393762] brcm-pcie 1000110000.pcie:   No bus range found for /axi/pcie@110000, using [bus 00-ff]
[    0.393773] brcm-pcie 1000110000.pcie:      MEM 0x1b00000000..0x1bfffffffb -> 0x0000000000
[    0.393779] brcm-pcie 1000110000.pcie:      MEM 0x1800000000..0x1affffffff -> 0x0400000000
[    0.393785] brcm-pcie 1000110000.pcie:   IB MEM 0x0000000000..0x0fffffffff -> 0x1000000000
[    0.394961] brcm-pcie 1000110000.pcie: setting SCB_ACCESS_EN, READ_UR_MODE, MAX_BURST_SIZE
[    0.394967] brcm-pcie 1000110000.pcie: Forcing gen 3
[    0.395004] brcm-pcie 1000110000.pcie: PCI host bridge to bus 0000:00
[    0.503759] brcm-pcie 1000110000.pcie: link up, 8.0 GT/s PCIe x1 (!SSC)
[    0.515879] pcieport 0000:00:00.0: enabling device (0000 -> 0002)
[    0.515917] pcieport 0000:00:00.0: PME: Signaling with IRQ 39
[    0.515980] pcieport 0000:00:00.0: AER: enabled with IRQ 39
```

You can see the entries: *„Forcing gen 3”* and *„Link up, 8.0 GT/s PCIe x1”*, which means that the modification was successful. However, **this does not guarantee full system stability under all conditions.**

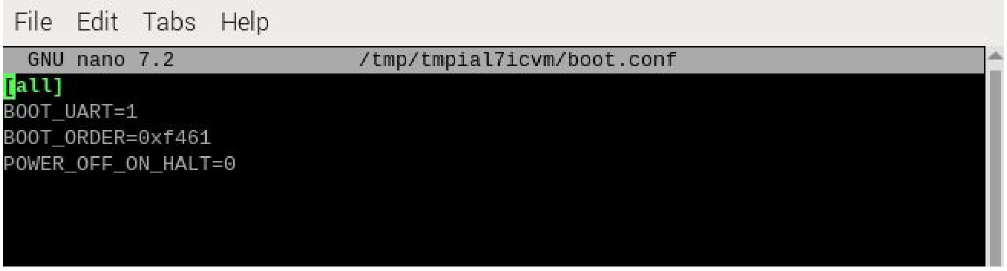
Setting the boot sequence by editing the contents of the EEPROM memory

If for some reason we are unable to update the Raspberry Pi 5 computer software, some of the settings related to system bootu from the NVME disk may be unavailable. In such a situation, you can set the appropriate sequence by directly editing the contents of the EEPROM memory of the Raspberry Pi 5 computer.

Open the console window (Terminal), e.g. using the key combination *Ctrl+Alt+T* and enter:

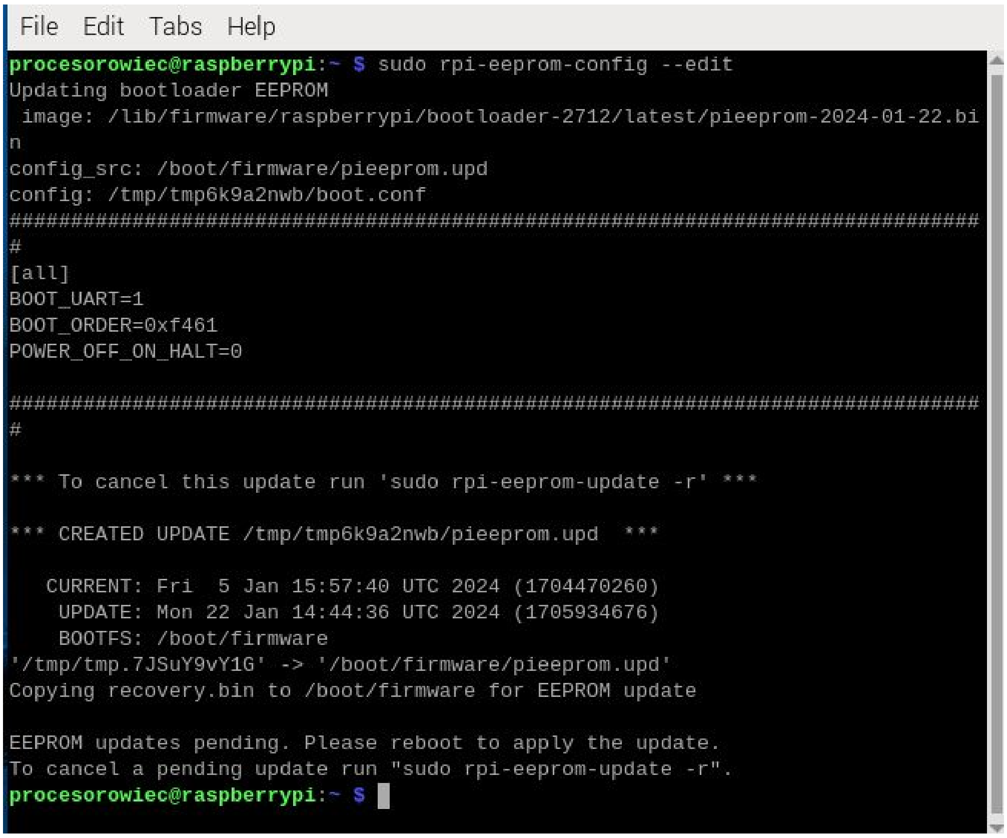
```
sudo rpi-eeprom-config --edit
```

Replace the line starting with "*BOOT_ORDER=*" with "*BOOT_ORDER=0xf461*".



```
 GNU nano 7.2                  /tmp/tmpial7icvm/boot.conf
[all]
BOOT_UART=1
BOOT_ORDER=0xf461
POWER_OFF_ON_HALT=0
```

This entry sets a safe sequence that causes the system to load from the NVME M.2 disk if no memory card is detected. Then save the changes using the *Ctrl+O* keys, close the editor using the *Ctrl+X* keys and restart the system.



```
procesorowiec@raspberrypi:~ $ sudo rpi-eeprom-config --edit
Updating bootloader EEPROM
 image: /lib/firmware/raspberrypi/bootloader-2712/latest/pieeprom-2024-01-22.bi
n
config_src: /boot/firmware/pieeprom.upd
config: /tmp/tmp6k9a2nwb/boot.conf
################################################################################
#
[all]
BOOT_UART=1
BOOT_ORDER=0xf461
POWER_OFF_ON_HALT=0

################################################################################
#

*** To cancel this update run 'sudo rpi-eeprom-update -r' ***

*** CREATED UPDATE /tmp/tmp6k9a2nwb/pieeprom.upd  ***

   CURRENT: Fri  5 Jan 15:57:40 UTC 2024 (1704470260)
    UPDATE: Mon 22 Jan 14:44:36 UTC 2024 (1705934676)
    BOOTFS: /boot/firmware
'/tmp/tmp.7JSuY9vY1G' -> '/boot/firmware/pieeprom.upd'
Copying recovery.bin to /boot/firmware for EEPROM update

EEPROM updates pending. Please reboot to apply the update.
To cancel a pending update run "sudo rpi-eeprom-update -r".
procesorowiec@raspberrypi:~ $
```

Creating a partition on a new NVME drive

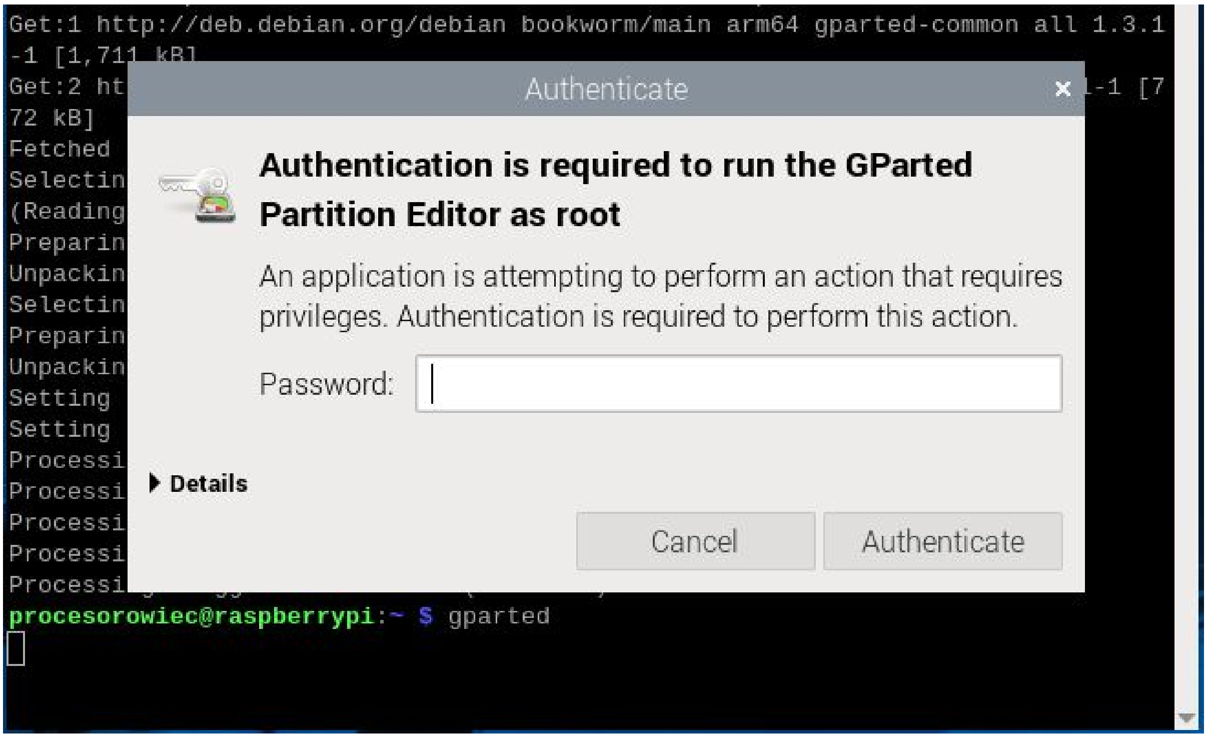
The NVME M.2 drive does not have to be the main drive on which the operating system will be installed - it can act as a secondary drive for storing large amounts of data. When a brand new NVME M.2 drive is connected to the Raspberry Pi, it will be necessary to create a partition to make it visible in the file system.
Raspberry Pi 5 allows you to install the GParted software tool, which is used to create partitions and other related operations, and is operated via an intuitive graphical interface. To install GParted, open a console window (Terminal), e.g. using the key combination *Ctrl+Alt+T* and enter the following command to start the installation:

```
sudo apt install gparted -y
```

After the installation is complete, enter the command:

```
gparted
```

and in the window that appears, enter the password to access the system:



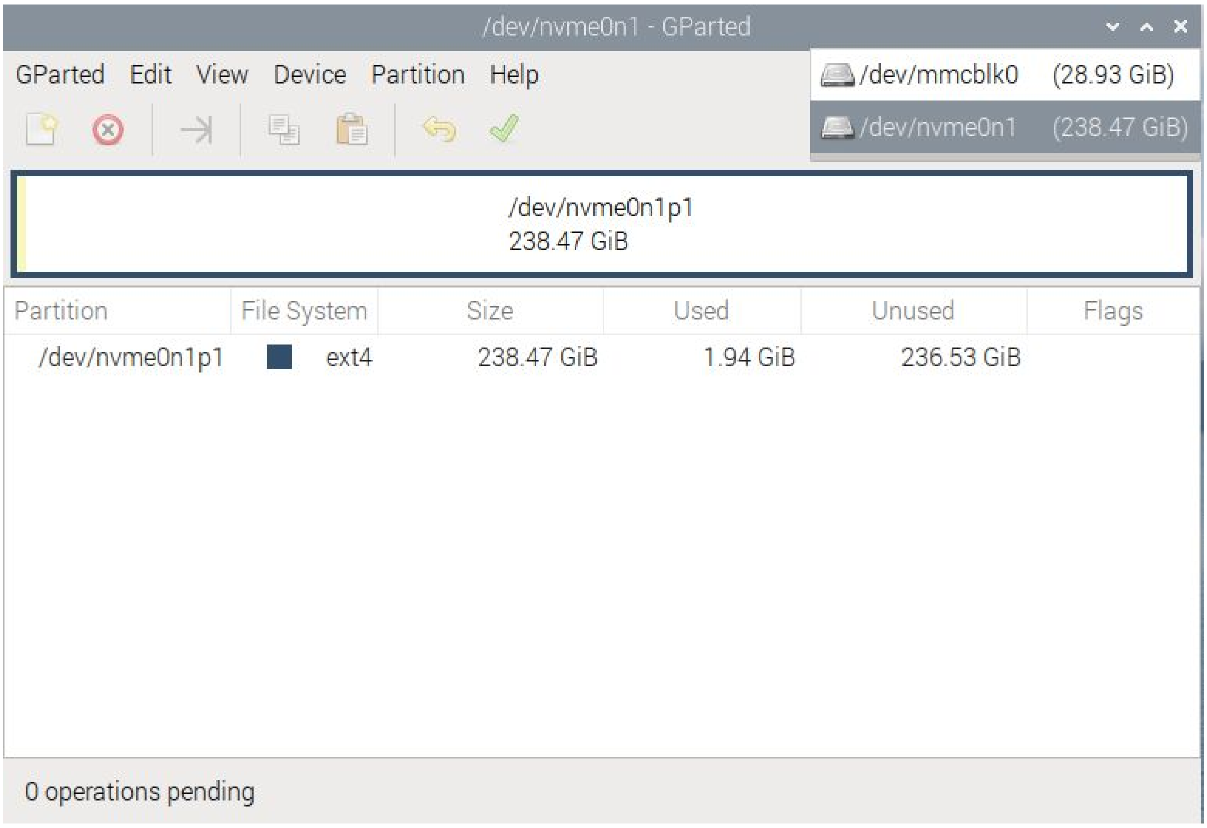
```
Get:1 http://deb.debian.org/debian bookworm/main arm64 gparted-common all 1.3.1
-1 [1,711 kB]
Get:2 ht                                                               1-1 [7
72 kB]
Fetched
Selectin
(Reading
Preparin
Unpackin
Selectin
Preparin
Unpackin
Setting
Setting
Processi
Processi
Processi
Processi
Processi
procesorowiec@raspberrypi:~ $ gparted
```

Authentication is required to run the GParted Partition Editor as root

An application is attempting to perform an action that requires privileges. Authentication is required to perform this action.

Password:

The main program window is shown in the screenshot below. The interface is clear and intuitive:

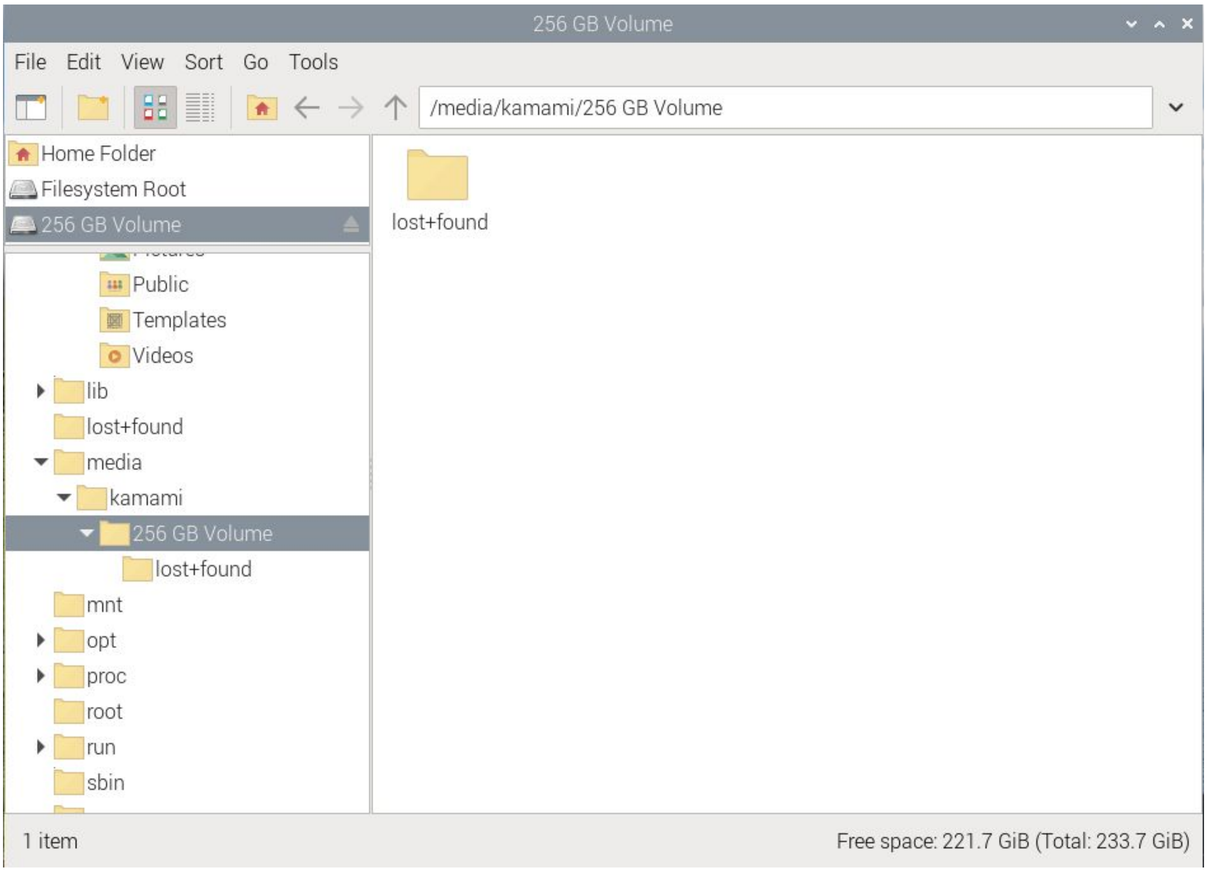


Partition	File System	Size	Used	Unused	Flags
/dev/nvme0n1p1	🟦 ext4	238.47 GiB	1.94 GiB	236.53 GiB	

0 operations pending

In the upper right corner there is a field to select the disk on which the operations will be performed, in the *Partition* tab there are the most important commands, such as creating a new partition - *New*. All operations must be confirmed with the green button.

Once the partition is created, it will become visible in the file manager, similar to the screenshot below:



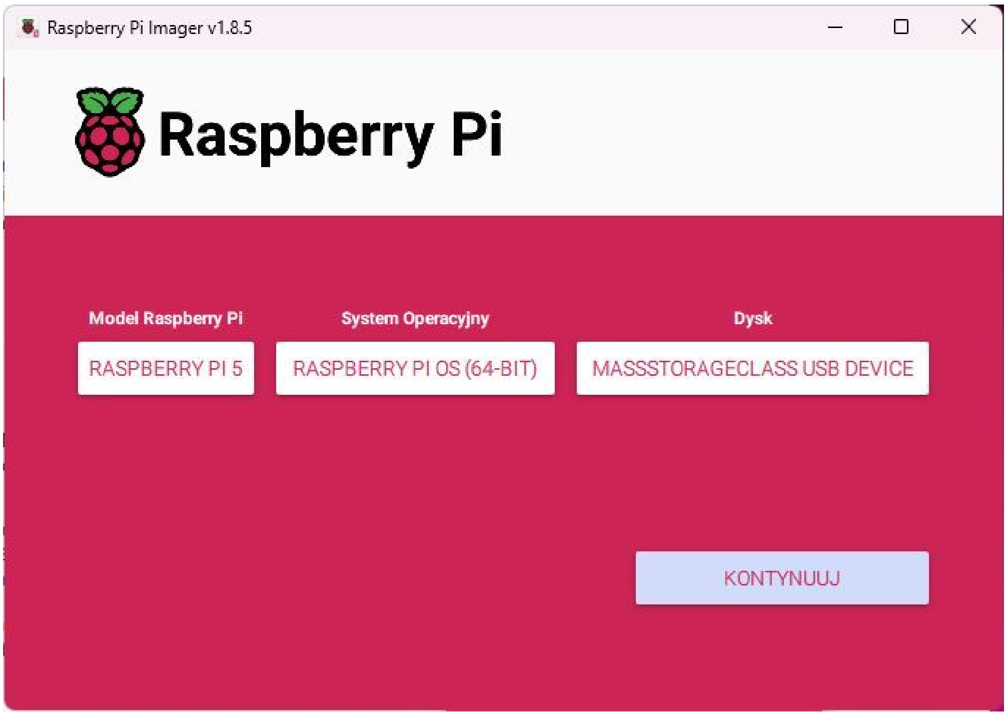
/media/kamami/256 GB Volume

1 item

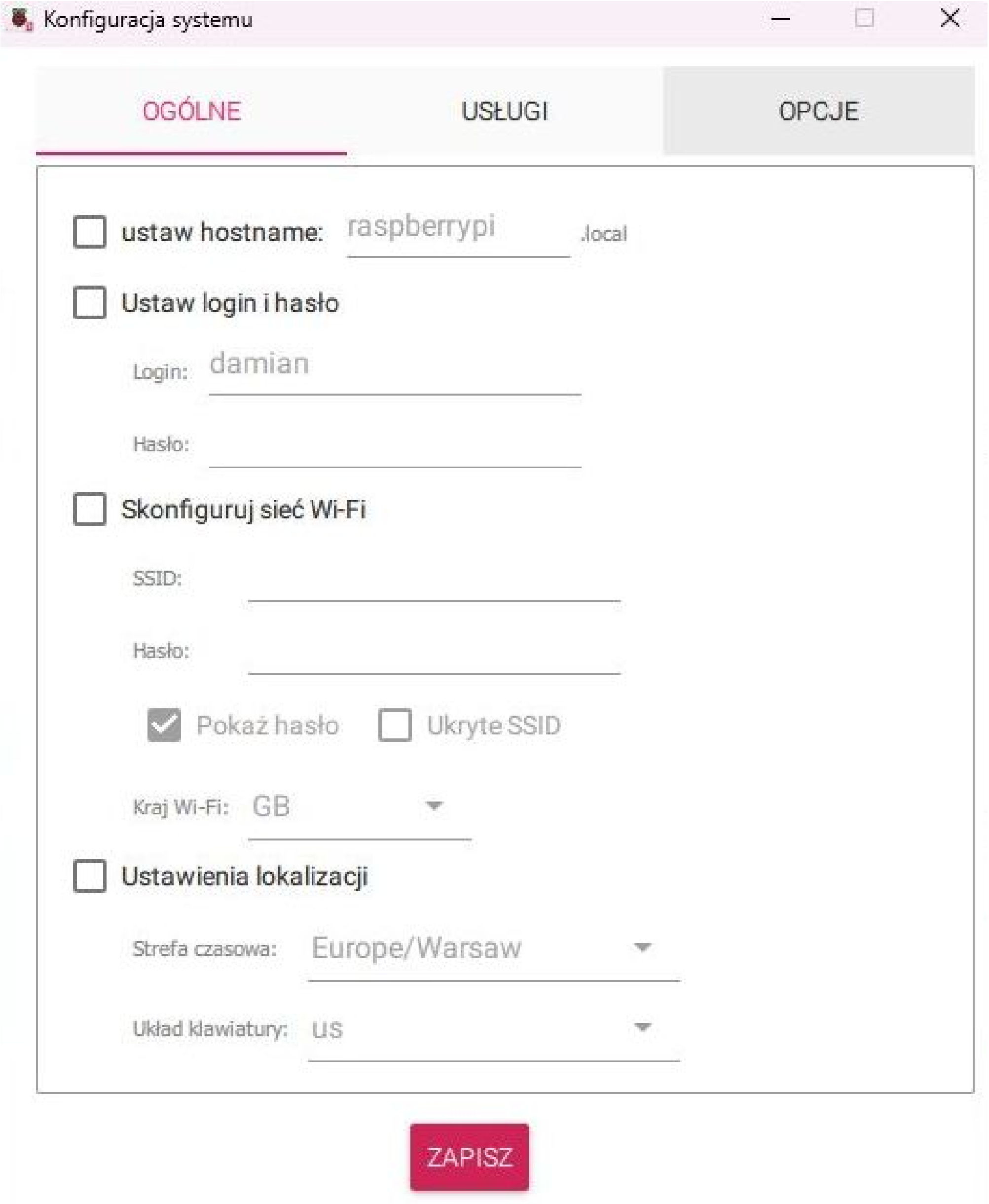
Free space: 221.7 GiB (Total: 233.7 GiB)

Preparing a memory card with an operating system for Raspberry Pi 5

The software that allows you to prepare the memory card can be downloaded from https://www.raspberrypi.com/software/.
After installing and running it, select:
computer model: *RASPBERRY PI 5*,
operating system: *RASPBERRY PI OS (64-BIT)*,
disk: here we indicate the memory card (or other memory, e.g. USB pendrive).



Click *CONTINUE*, and then select *EDIT SETTINGS*.
Now it is worth setting the login and password for access to the installed system, which is necessary to maintain its security. Then you need to provide the Wi-Fi access parameters (if you are going to use this form of internet access) and fill in the information about the location.



Konfiguracja systemu

□ ustaw hostname: raspberrypi .local

□ Ustaw login i hasło

Login: damian

Hasło:

□ Skonfiguruj sieć Wi-Fi

SSID:

Hasło:

☑ Pokaż hasło □ Ukryte SSID

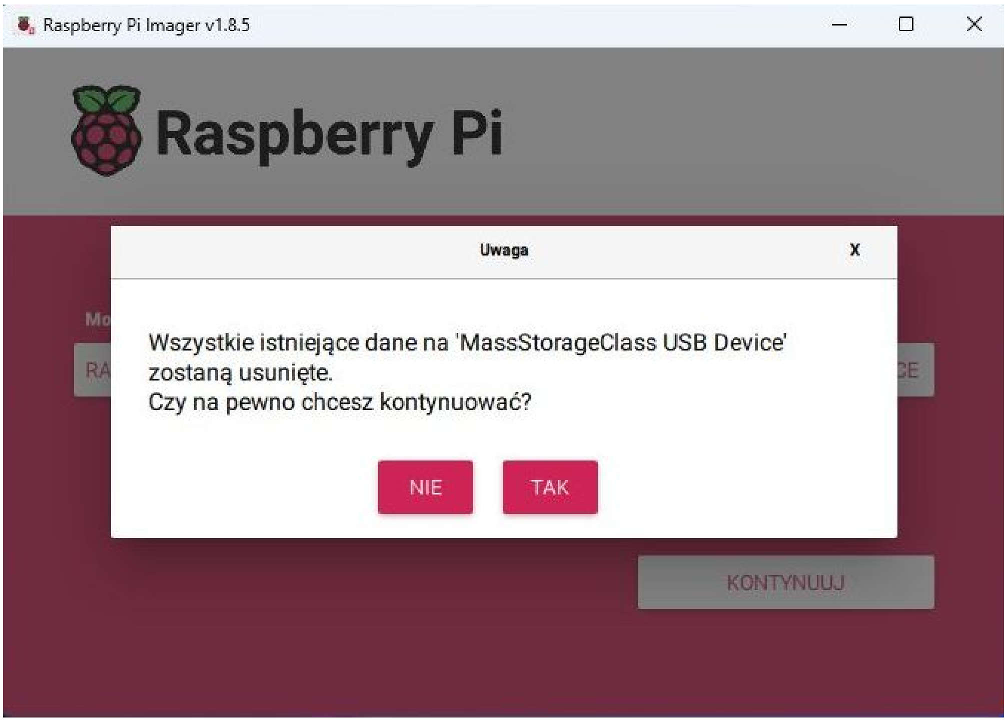
Kraj Wi-Fi: GB

□ Ustawienia lokalizacji

Strefa czasowa: Europe/Warsaw

Układ klawiatury: us

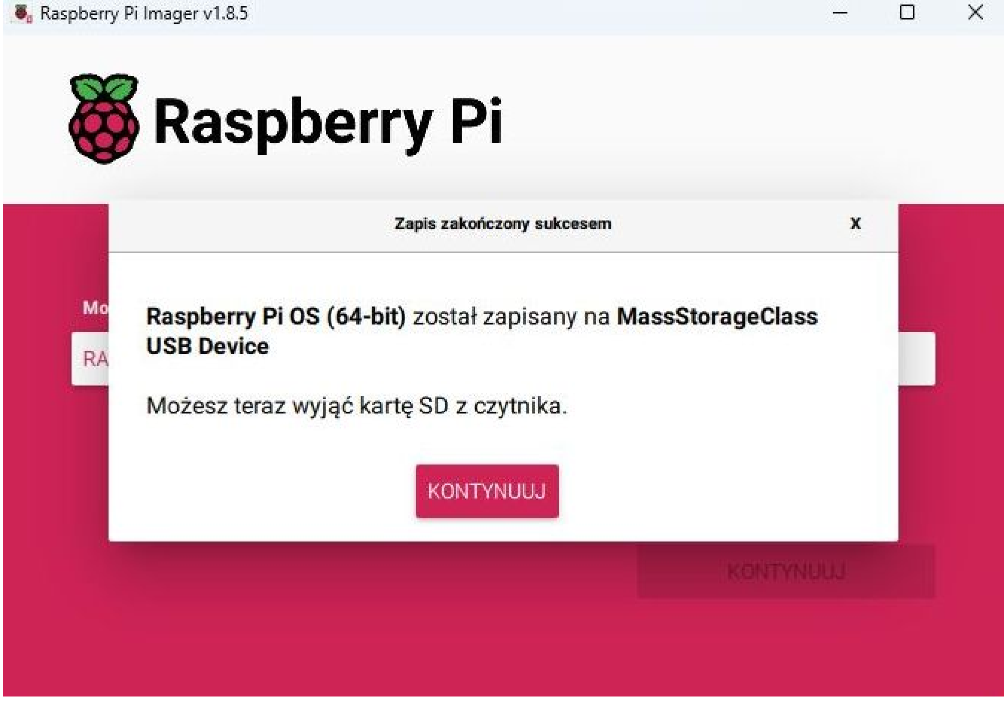
To apply the settings, click *SAVE* and in the next window confirm *YES*. A warning will be displayed that all data currently on the memory card will be deleted. Click *YES* if you agree to this and want to continue.



Uwaga

Wszystkie istniejące dane na 'MassStorageClass USB Device' zostaną usunięte.
Czy na pewno chcesz kontynuować?

Installing the system on the memory card will take from a few to a dozen or so minutes. After the actions are completed, if everything went well, a corresponding message will be displayed and the memory card with the operating system will be ready.



Zapis zakończony sukcesem

Raspberry Pi OS (64-bit) został zapisany na **MassStorageClass USB Device**

Możesz teraz wyjąć kartę SD z czytnika.

Additional features of the KAmod RPi5 PCIe-M.2 adapter

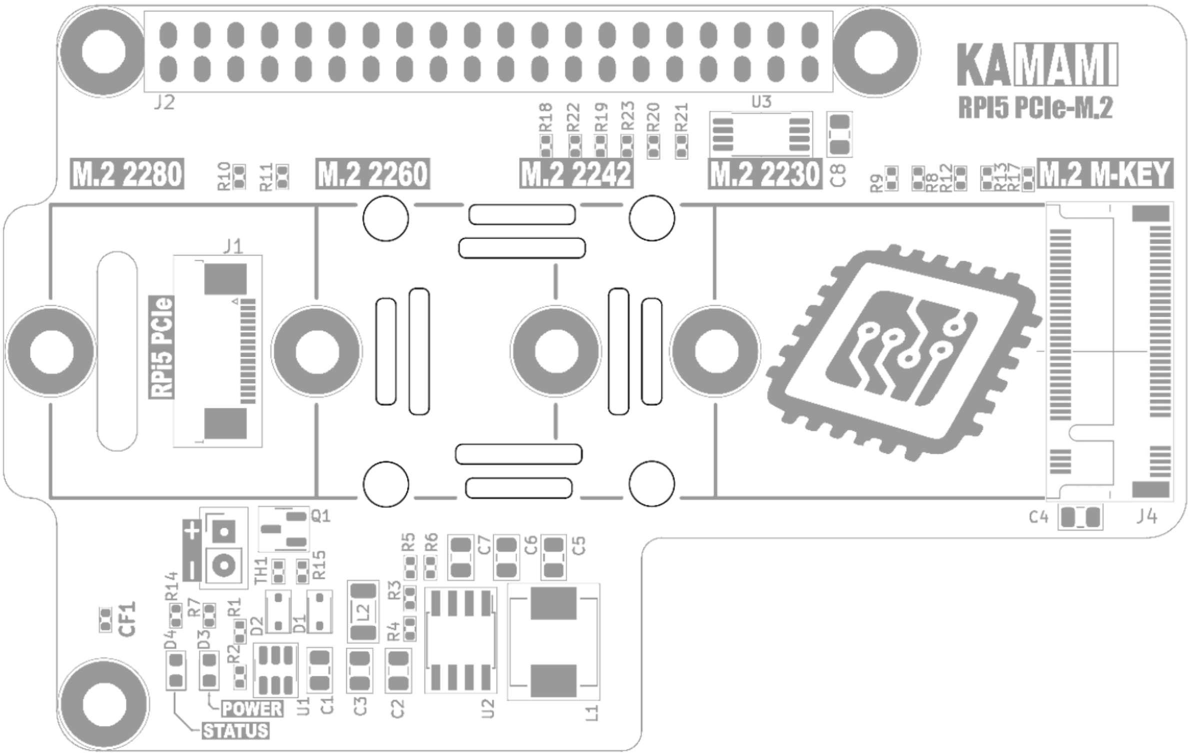
Connection using only the ribbon cable

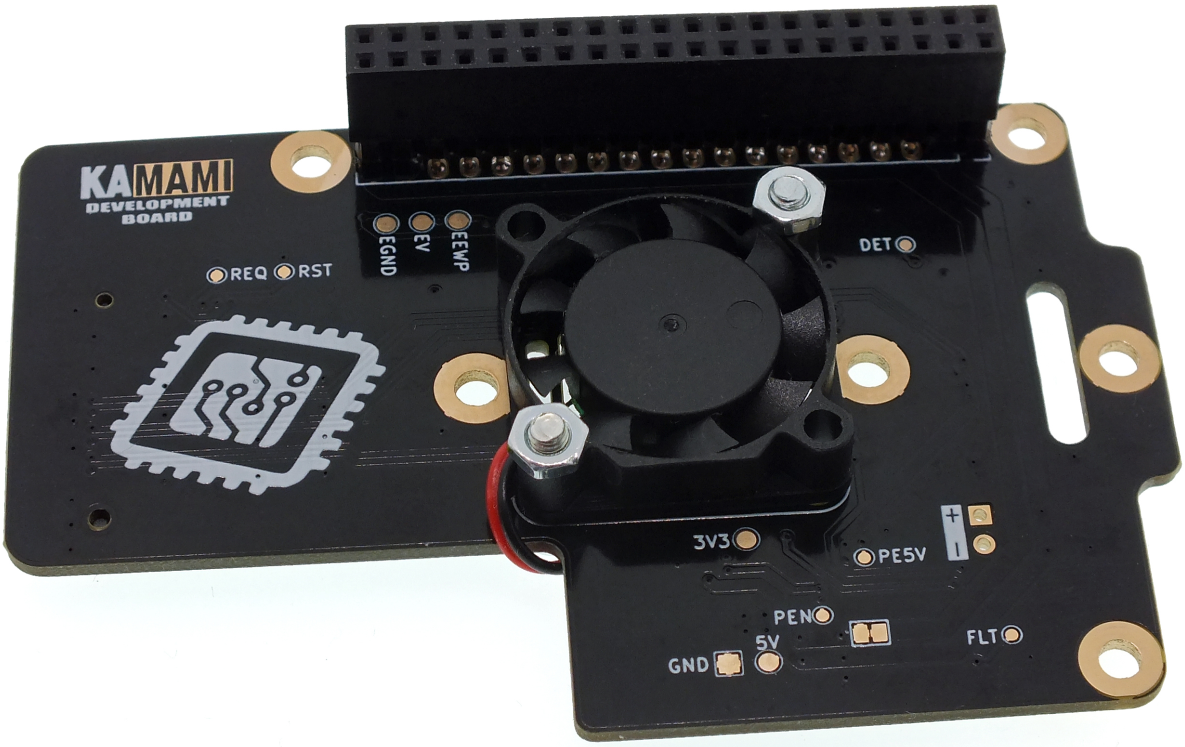
The KAmod RPi5 PCIe-M.2 adapter can work properly when connected to the Raspberry Pi 5 using only the FFC 16/0.5 ribbon cable. The only condition that must be met is that the current consumption of the NVME M.2 drive cannot exceed 1 A.

The M.2 standard requires a power supply with a voltage of 3.3 V and a current of up to 3 A. The rated current consumption, provided by the drive manufacturers, applies to work with the full PCI Express GenX **x4** interface, at maximum drive performance, while when working with the **x1** interface and lower performance, the current consumption is lower. In practice, many drives can operate correctlyaw with a current limit of 1 A with the PCI Express Gen2 x1 interface used in the Raspberry Pi 5.

Fan installation

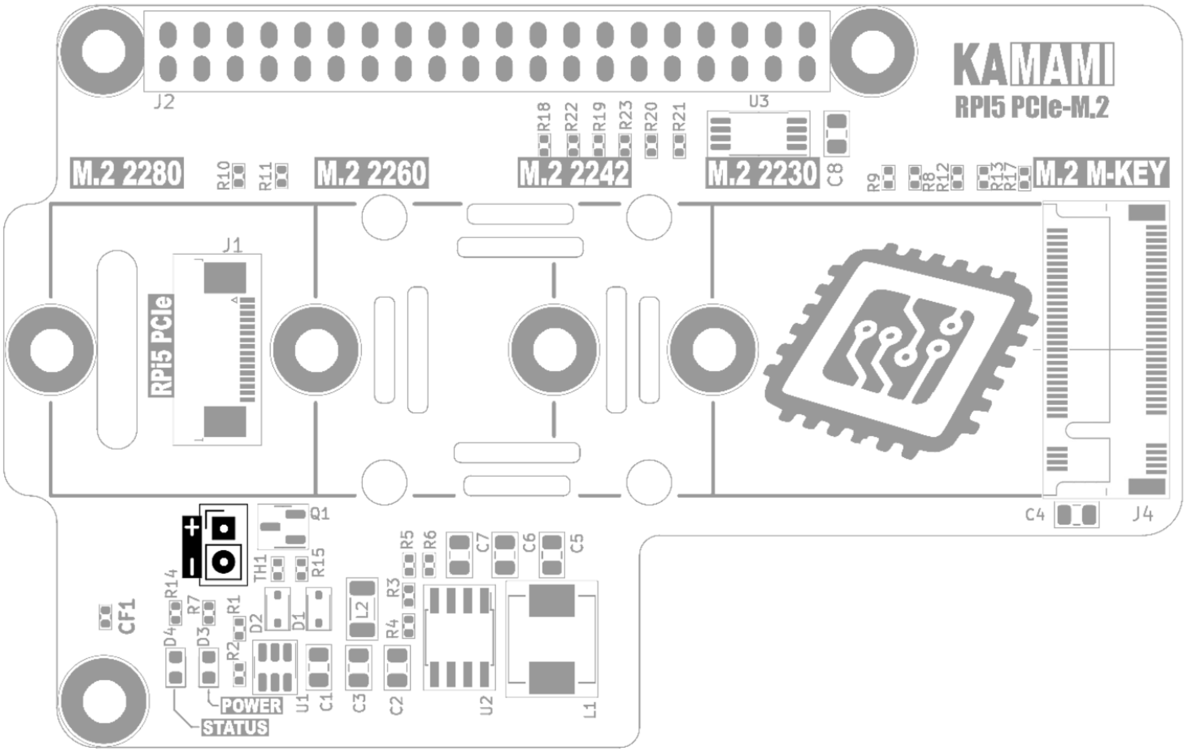
The KAmod RPi5 PCIe-M.2 adapter board has oblong holes that improve the circulation of air cooling the Raspberry Pi 5 computer. The design of the adapter does not block the possibility of using a radiator with a fan dedicated to the Raspberry Pi 5. If additional cooling is not installed, it is possible to mount a 25x25 mm fan on the bottom side of the adapter board. The board has holes for placing appropriate screws.




KAMAMI
DEVELOPMENT
BOARD
REQ
RST
EGND
EV
EEWP
DET
3V3
PE5V
+
-
PEN
5V
GND
FLT

Fan power supply

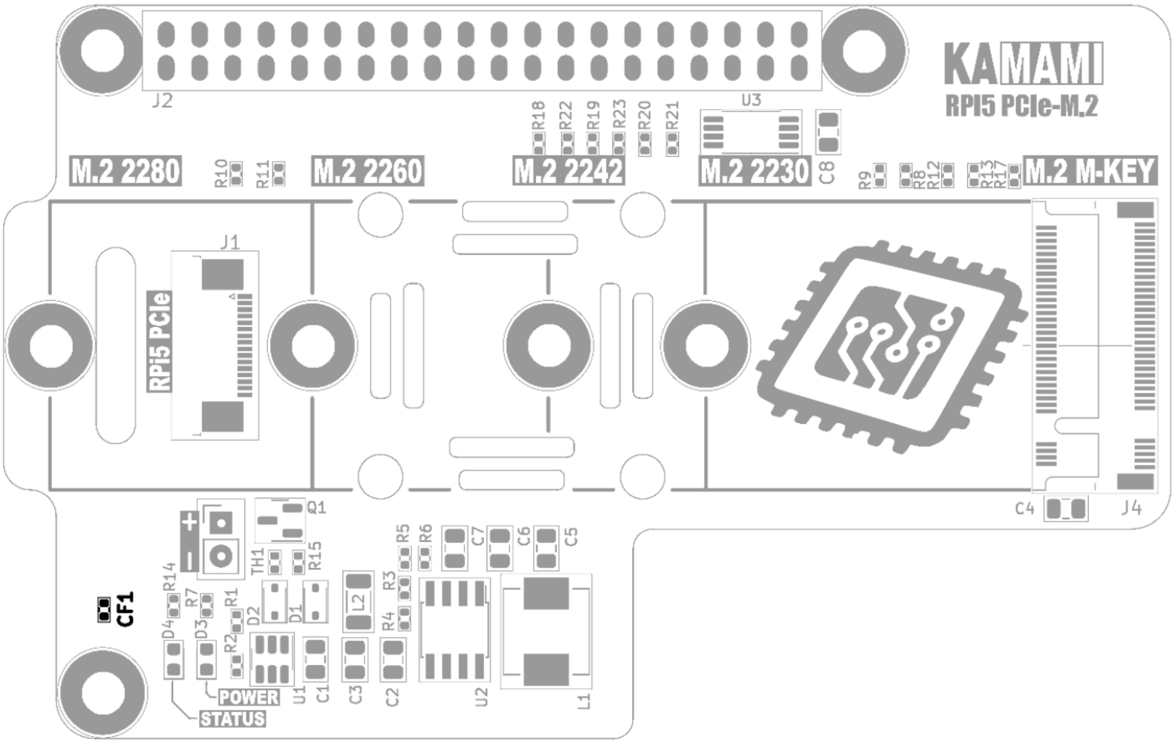
There is a connector on the board that can power an additional fan. It provides voltage that is regulated by temperature - the higher the adapter temperature, the closer the voltage will be to 5 V. Additionally, the fan will be completely turned off after the computer is turned off (in standby mode).
The maximum fan current cannot exceed 0.3 A.



Constant power supply for NVME disk=

The adapter supplies power to the NVME M.2 disk continuously - the disk is powered even when the Raspberry Pi 5 computer is turned off (in standby mode). Some disks allow the operating system to start correctly only in this configuration.

The KAmod RPi5 PCIe-M.2 adapter also offers another configuration - the disk power will be turned off when the Raspberry Pi 5 is turned off (in standby mode). This solution allows you to reduce the power consumption of the entire set in standby mode, but may disrupt the correct startup of the system. To enable power saving mode, remove the element marked CF1 from the board.



External links

- CAD model (STEP)

btc

BTC Korporacja
05-120 Legionowo
ul. Lwowska 5
tel.: (22) 767-36-20
faks: (22) 767-36-33
e-mail:
sprzedaz@kamami.pl
https://kamami.pl

Zastrzegamy prawo do wprowadzania zmian bez uprzedzenia.
Oferowane przez nas płytki drukowane mogą się różnić od prezentowanej w dokumentacji, przy czym zmianom nie ulegają jej właściwości użytkowe.
BTC Korporacja gwarantuje zgodność produktu ze specyfikacją.
BTC Korporacja nie ponosi odpowiedzialności za jakiekolwiek szkody powstałe bezpośrednio lub pośrednio w wyniku użycia lub nieprawidłowego działania produktu.
BTC Korporacja zastrzega sobie prawo do modyfikacji niniejszej dokumentacji bez uprzedzenia.