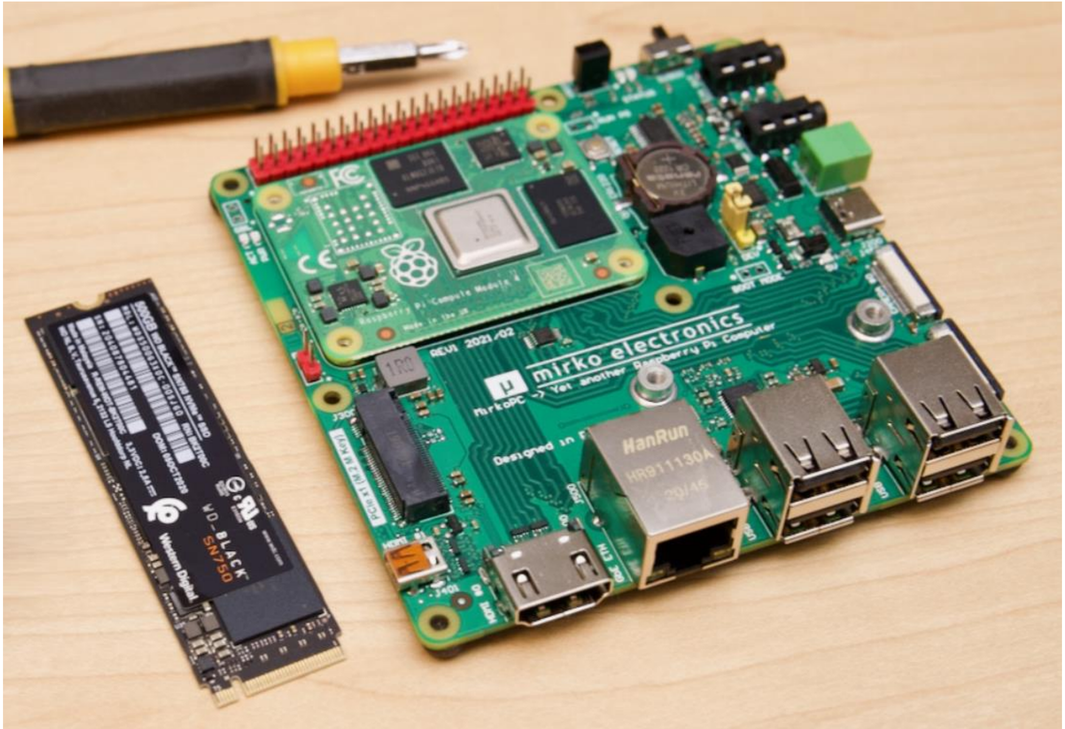
MirkoPC

Yet another Raspberry Pi Carrier Board



picture source: Jeff Geering blog (https://www.jeffgeerling.com/blog)

Document revision: **REV1.1**

Hardware revision: **REV1**

Table of Contents

1. Glossary

All abbreviations used in this document are listed as below in the following table:

CM4	Compute Module 4	Raspberry Pi SOM Module (Gen4)
RPi	Raspberry Pi	-
GPIO	General Purpose Input Output	General purpose pin of processor
NVMe	NVM (Non-Volatile Memory) Express	Logical-device interface specification for accessing a computer's non-volatile storage media attached via PCI Express (PCIe) bus
SSD	Solid-State Drive	Solid-state storage device that uses integrated circuits
MIPI	Mobile Industry Processor Interface	Serial interface for connecting between media device and host procesor
CSI-2	Camera Serial Interface (v2.0)	Serial interface for connecting between camera and a host processor
DSI	Display Serial Interface	Serial interface for connecting between display controller and a host processor
HDMI	High-Definition Multimedia Interface	Serial interface for trasmitting high-resolution video and audio data
USB	Universal Serial Bus	Industry standard for connection, communication and power supply between computers, peripherals and other computers
GbE	Gigabit Ethernet	Communication interface via Ethernet network standard with up to 1Gb/s transfer rate
IR	Infrared Receiver	Wireless interface for remote control
DAC	Digital-to-Analog Converter	Chip IC to convert digital data to analog signal
RTC	Real Time Clock	Dedicated chip IC for generating real-time clock and calendar with battery backup
FFC/FPC	Flat Flex Cable / Flexible Printed Circuits	flexible and flat connection system (cables and connectors) between PCB boards
IC	Integrated Circuit	-
CPU	Central Processing Unit	Chip IC that executes instructions comprising a computer program
RAM	Random Access Memory	Computer memory for read/write data, typically used to store working data and machine code
SBC	Single-Board Computer	-
SOM	System On Module	PCB board that integrates a system function in a single module (e.g. RPi CM4)
eMMC	Embedded MMC (MultiMediaCard)	Memory card standard used for solid-state storage in a small BGA IC package
BGA	Ball Grid Array	miniature surface-mount packaging (a chip carrier) used for integrated circuits
OS	Operating System	-
WiFi	Wireless Fidelity	Wireless network standard used for local area networking of devices and Internet access
BLE	Bluetooth Low Energy	Energy-efficient Bluetooth Standard
CLI	Command Line Interface	User terminal interface to execute text commands

2. Introduction

MirkoPC project was created by the single DIY maker (known as Mirko from Poland) as a non-commercial and non-official platform for Raspberry Pi Compute Module 4 (CM4).

MirkoPC motherboard was designed and dedicated for two different purposes:

- as a development board (carrier board) for CM4 modules (open frame),
- as microcomputer (micro PC) with dedicated 3D printed case (enclosure).

The project is mostly dedicated to Raspberry Pi users, technology enthusiasts, and passionists.

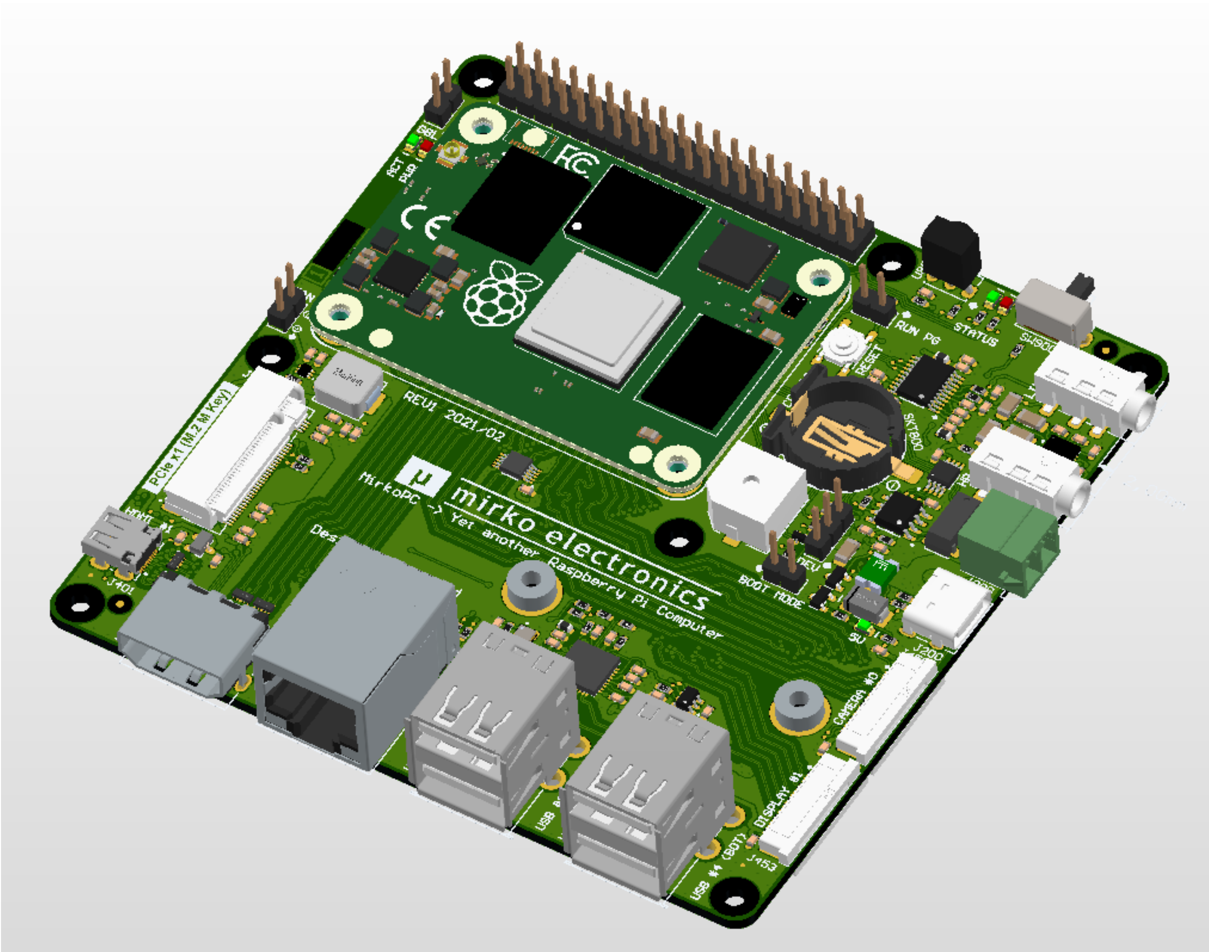


Fig. MirkoPC board (HW REV1) – Top view (with installed CM4 module)

The main motivation to create this project was to connect all the interfaces in the CM4 module available for the user and also to create a fully functional computer in a small form factor.

Compared to the Raspberry Pi 4B SBC (single-board computer) as well as the CM4-IO board, the MirkoPC project has additional features such as a full M.2 slot for the installation of the high-performance NVMe SSD disk, a high-quality HiFi audio output for connecting stereo or headphones, it is equipped with an IR receiver to control by using remote control, and also has typical PC accessories such as numerous USB ports, HDMI, RTC clock and buzzer.

3. Hardware block diagram

Figure, as below, shows the hardware block diagram for MirkoPC carrier board and illustrates the connections between CM4 module and its peripherals (e.g. audio and video ports, GPIO devices, USB ports and power supply circuit).

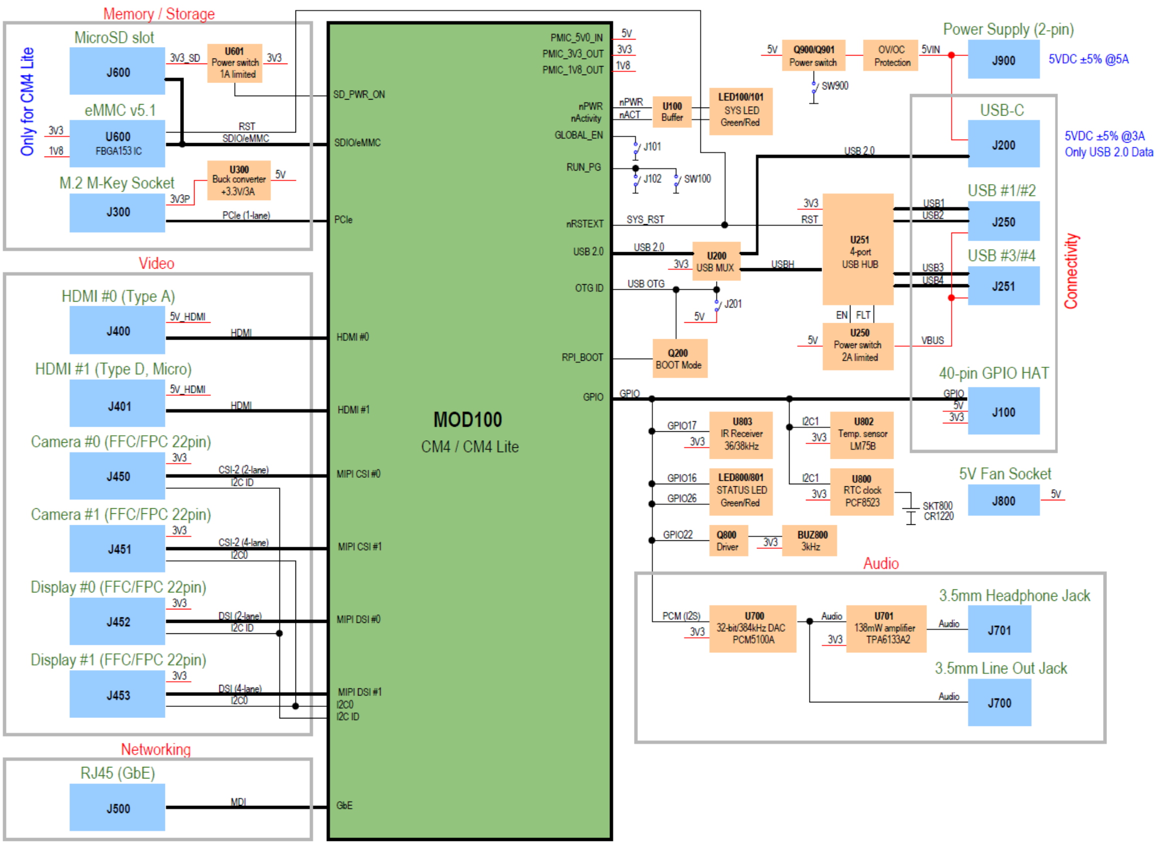


Fig. MirkoPC board – simplified block diagram

Main component and all the interfaces are connected to the Raspberry Pi Compute Module 4 (CM4).

Here you can find the block diagram in the PDF format:

https://github.com/mfolejewski/MirkoPC/blob/main/MirkoPC_block_schematic_draft_2021-03-24.pdf

4. Technical Specification

CPU

- BROADCOM BCM2711
- Quad core Cortex-A72 (ARM v8)
- 64-bit SoC @ 1.5GHz
- H.265 (HEVC, up to 4K@60Hz dec)
- H.264 (up to 1080p@60Hz, 1080p@30Hz enc)
- OpenGL ES3.0 graphics

MEMORY

- RAM up to 8GB LPDDR4-3200 (CM4 module)
- Flash up to 32GB eMMC 5.1 (CM4 module)

STORAGE

- M.2 2242/2280 M-Key socket with PCIe x1 Gen2 (NVMe SSD drive support)
- MicroSD slot (SDIO 4-bit mode)
- extra eMMC 5.1 support (BGA 153-pin chip)

NETWORKING

- WiFi 2.4GHz, 5GHz IEEE 802.11 b/g/n/ac (onboard CM4 module)
- Bluetooth 5.0 BLE (onboard CM4 module)
- 100/1000M Ethernet RJ45 with IEEE1588

CONNECTIVITY

- 4x USB 2.0 Type A ports
- HAT interface (28-pin GPIO Raspberry Pi standard)
- up to: 5x UART, 5x I2C, 5x SPI, 2x PWM (by GPIO port)
- up to: 2xSDIO, 1xDPI, 1x PCM, 3x GPCLK (by GPIO port)

VIDEO

- 2x HDMI 2.0 outputs (up to 4K@60Hz)
- 2x MIPI CSI-2 camera interfaces (22-pin FFC/FPC)
- 2x MIPI DSI display interfaces (22-pin FFC/FPC)

AUDIO

- 32-bit/384kHz stereo line output (100dB SNR)
- 128mW headphone output (DirectDrive, <0.01% THD typ.)

EXTRAS

- RTC clock with battery backup (CR1220)
- Temperature sensor (I2C)
- IR 38kHz infrared receiver
- 5VDC Fan power supply
- 3kHz buzzer
- 2x user status LEDs (green/red)
- Jumpers for Boot mode and Power Down
- RESET switch

POWER SUPPLY

- 5VDC standard, OV and OC protected
- USB-C @3A power supply connector
- 2-pin terminal block @5A
- power switch

OS

- 32- and 64-bit Linux-based OS (e.g. Raspbian, Ubuntu, OpenSUSE, Manjaro)

5. Applications

Mirko PC board has been designed for the 2 major applications:

- non-official development board (carrier board) for Raspberry Pi CM4 module,
- miniature PC (Mini PC).

There are other potential applications and dedicated purposes, for different industries, such as Telecommunication, Multimedia and Consumer electronics, Entertainment, Industrial automation, Networking, Computing, etc.:

- Media center box (Home media player)
- HiFi music server
- Micro server
- Video game console & Retro-gaming (RetroPie)
- Home automation controller
- Smart controller
- HMI panel
- NAS server
- BOINC client
- Cloud computing platform
- Machine learning and AI platform (Coral Edge TPU)
- and many more…

6. User interface

6.1. Introduction

MirkoPC carrier board has available to the user lot of different interfaces, such as: USB ports, GbE port, dedicated interfaces for connecting cameras and LCD displays, audio and video outputs, such as HDMI or standard 3.5mm jacks, IR receiver and also extra GPIO interface for connecting additional boards and extensions to create and development embedded systems.

6.2. Key components and interfaces

Below on the drawings, marked and described all the interfaces available for user.

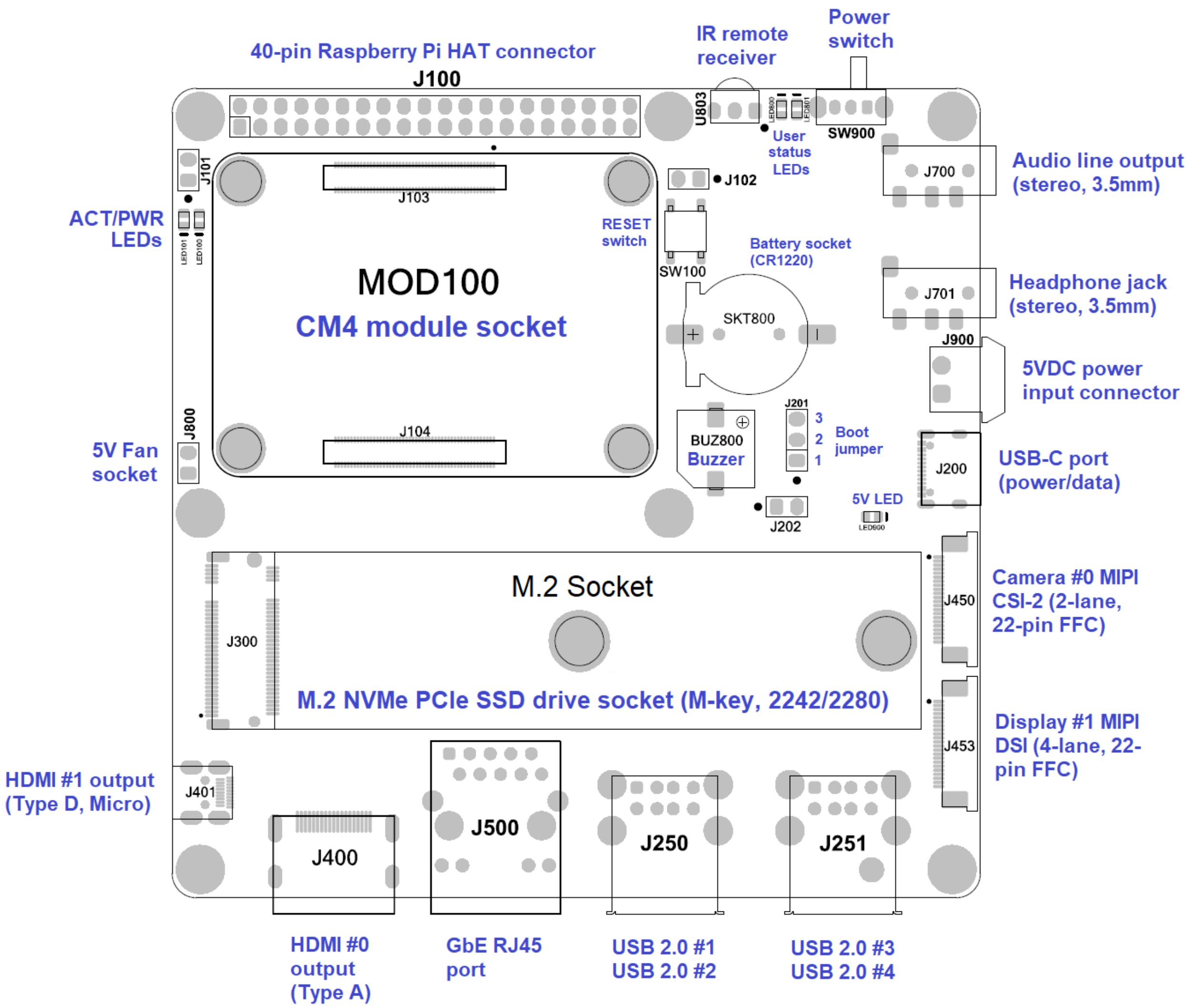


Fig. User interface – Top view

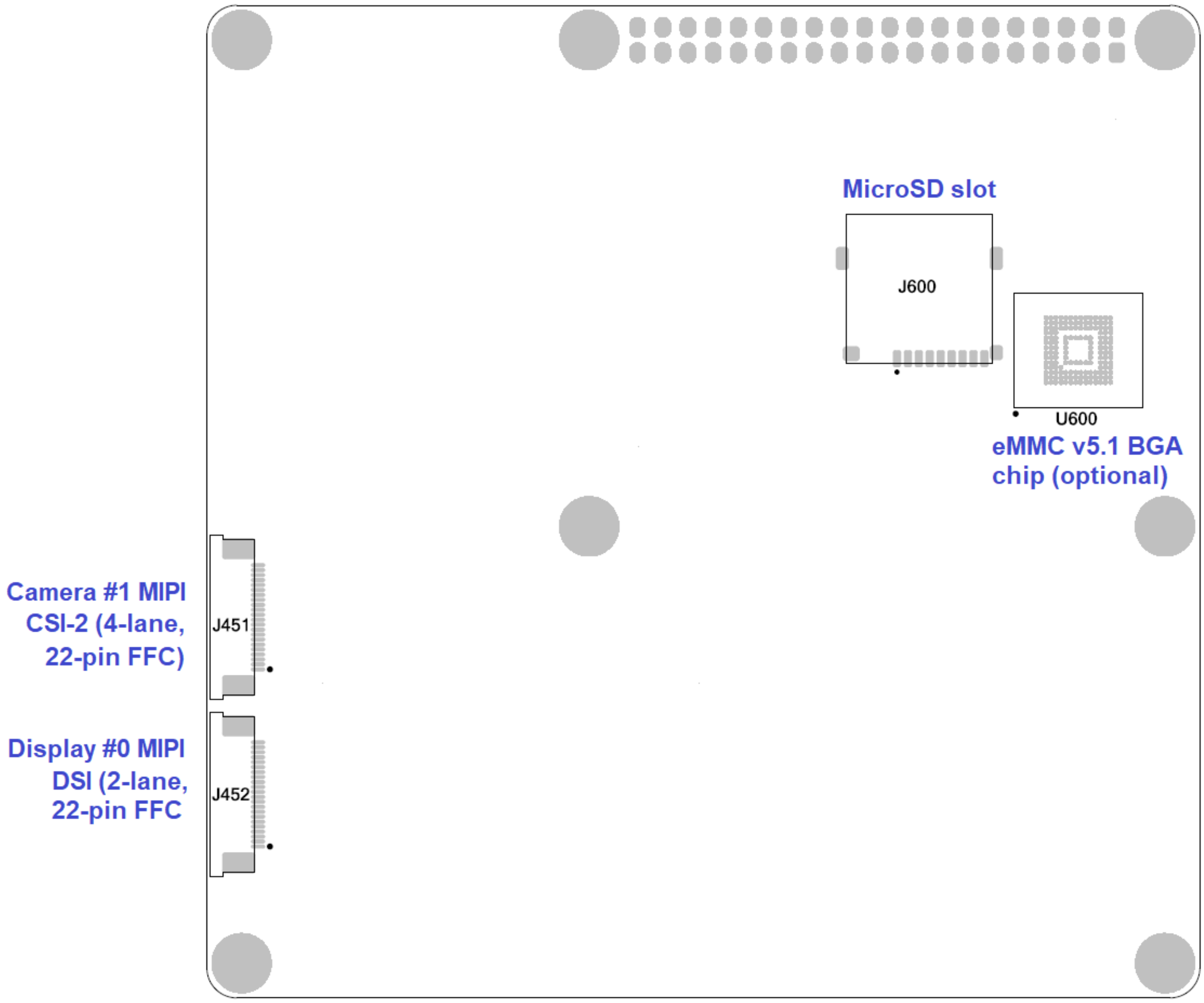


Fig. User interface – Bottom view

These drawings can be downloaded from:

https://github.com/mfolejewski/MirkoPC/tree/main/drawings

The following interfaces and ports are available on the top side of carrier board, grouped by. their functionalities:

- SOM socket:
    - MOD100 – socket for installing Raspberry Pi Compute Module 4 modules (CM4 or CM4 Lite),
- video interfaces:
    - J400 – HDMI 2.0 port #0 output, type A (full size connector), up to 4K UHD @60Hz,
    - J401 – HDMI 2.0 port #1 output, type D (MicroHDMI size), up to 4K UHD @60Hz,
    - J450 – camera #0 input, 2-lane MIPI CSI-2 standard, 22-pin FFC/FPC connector (0.5mm pitch),
    - J453 – display #1 output, 4-lane MIPI DSI standard, 22-pin FFC/FPC connector (0.5mm pitch),
- audio interfaces:
    - J700 – audio line output (stereo, up to 2.1Vrms, 100dB SNR), standard 3.5mm jack,
    - J701 – headphone outut (stereo, up to 138mW, THD+N < 0.01% typ.), standard 3.5mm jack,

- general purpose interfaces:
  - J250 – double USB 2.0 port (type A connectors), host mode, #1 and #2 ports,
  - J251 – double USB 2.0 port (type A connectors), host mode, #3 and #4 ports,
  - J200 – USB-C port with USB 2.0 interface (device mode),
  - J100 – 40-pin GPIO port (fully compatible with Raspberry Pi GPIO HAT standard),
- network interfaces:
  - J500 – gigabit Ethernet port (GbE, RJ45 connector), support IEEE1548 (PTP), without PoE standard support,
- storage drive port:
  - J300 – M.2 socket with M-Key, 2242 and 2280 form factors supported, compatible with standard NVMe PCIe SSD drives (SATA drives are not supported),
- power supply connectors:
  - J900 – 5VDC @5A input, 2-pin terminal block connector (5mm pitch),
  - J200 – standard USB-C port as power supply input, 5VDC @3A,
- LED signaling:
  - LED100 – power supply LED (red color), controlled by CM4 module (LED_nPWR pin),
  - LED101 – activity LED (green color), controlled by CM4 module (LED_nActivity pin),
  - LED800 – status LED #1 (green color), connected to CM4 GPIO16 pin,
  - LED801 – status LED #1 (red color), connected to CM4 GPIO26 pin,
  - LED900 – 5VDC power supply LED (red color),
- switches:
  - SW900 – power supply switch, slide type, angled, SPDT mode,
  - SW100 – RESET switch, TACT type (monostable), connected to CM4 RUN_PG pin,
- other accessories:
  - BUZ800 – buzzer, continous sound, 3kHz, controlled by CM4 module (GPIO22 pin),
  - SKT800 – CR1220 lithium coin battery socket (RTC clock battery backup),
  - U803 – IR receiver (36kHz infrared), for receiving data from standard remote controllers,
  - J800 – 2-pin socket (2.54 mm pitch pin header) for connecting 5VDC standard fan,
  - J201 – 3-pin config jumper (2.54 mm pitch pin header) for selecting boot/user mode,
  - J202 – 2-pin BOOT mode jumper (2.54 mm pitch pin header), not mounted,
  - J101 – 2-pin GLOBAL ENABLE jumper (2.54 mm pitch pin header), connected to CM4 module (GLOBAL_EN pin),
  - J101 – 2-pin RESET jumper (2.54 mm pitch pin header), connected to CM4 module (RUN_PG pin, similar as SW100), not mounted,

The following interfaces and ports are available on the bottom side of carrier board, grouped by. their functionalities:

- video interfaces:
  - J451 – camera #1 input, 4-lane MIPI CSI-2 standard, 22-pin FFC/FPC connector (0.5mm pitch),
  - J452 – display #0 output, 2-lane MIPI DSI standard, 22-pin FFC/FPC connector (0.5mm pitch),
- storage drive devices:
  - J600 – microSD card slot (4-bit SDIO mode), used only for OS image storage,
  - U600 – additional eMMC v5.1 memory chip (FPGA153 footprint compatible), only for CM4 Lite modules, not mounted (optionally mounted instead of using microSD card),

6.3. Power supply switch

Carrier board has dedicated power supply switch (slide switch, marked as SW900). To turn on the device that switch should be set in left position. Similar, to turn off the device and cut out the power supply from the device, switch should be set in the right position (see picture below).

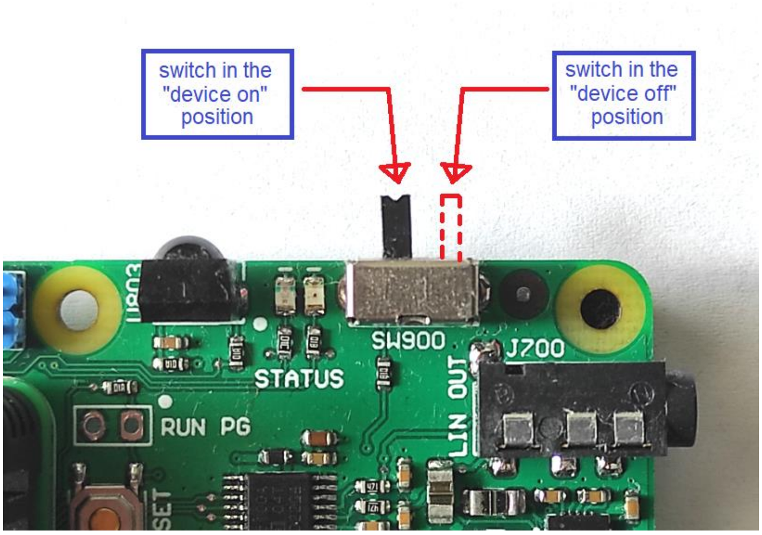


Fig. Possible power supply switch positions (SW900)

6.4. LED signaling

CM4 module control two separate LEDs to to indicate its current state (see figure below):

- LED100 (red color), marked as „PWR" on the PCB, driven by CM4 module and LED_nPWR line,
- LED101 (green color), marked as „ACT" on the PCB, driven by CM4 module and LED_nActivity line.

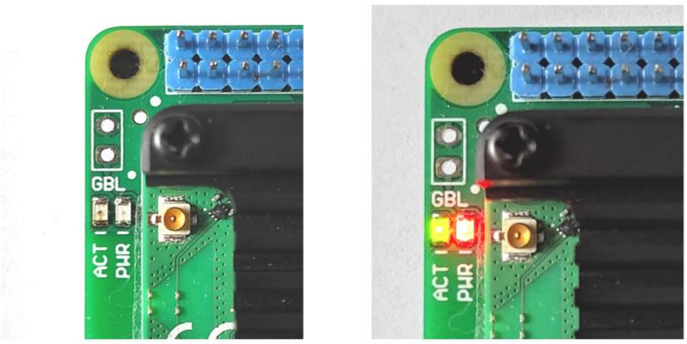


Fig. Possible ACT and PWR LED states (LED100 and LED101) – are off (left side), are on (right side)

When both LEDs are off, means that the OS is closed and the CM4 module is disabled (also when the power switch is in off position).

When both LEDs are on, means that the OS is running and the CM4 module is in normal operation mode (is working). When the „ACT" diode flash, signify access to the eMMC memory. If there is an error during booting this LED can flash to signal error patterns which can be decoded using the look up table. For more information please read the CM4 datasheet and visit the Raspberry Pi website.

6.5. 5VDC power supply indicator

To indicate that the carrier board is power supplied from nominal 5VDC voltage source, near to the USB-C connector (J200) is placed a dedicated LED diode (LED900, red color, marked as „5V” on the PCB). If the diode is on, this means that the 5VDC power supply is delivered to the board and the power switch (SW900) is in left position (device is power sourced and the OS is working).

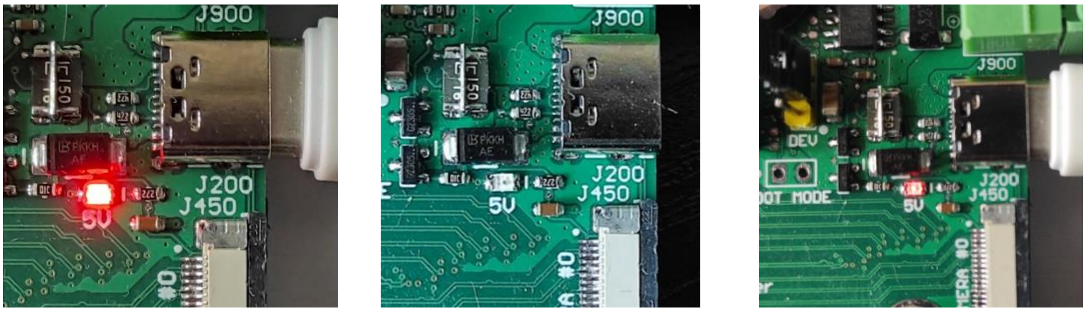


Fig. Possible LED900 states: is on (left side), is off (middle), glows dimly (righ side)

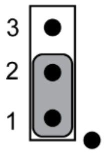
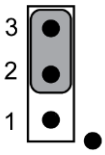
When the OS system is closed, and power switch (SW900) is in the right position or just simply power all source cables are disconnected, then the device is in off state (not working) and 5V LED is also off.

There is also the third possible state of the 5V LED - when the device is off, but is connected the HDMI cable, some voltage from the HDMI circuit can pull back to the carrier board 5V rail and then 5V LED can glows dimly. This normal effect and can be probably fixed in the next hardware revision.

6.6. BOOT Jumper

To select proper operating mode of the CM4 module is used the BOOT jumper (J201, marked as „DEV” on the PCB board).

There are two possible positions and configurations of the BOOT jumper:

J201 Jumper position	Mode	Description
	DEVICE (Flashing)	USB bus of the CM4 module is connected directly to the USB-C connector in device mode, and its embedded eMMC memory or microSD card (for CM4 Lite) is mounted as mass storage drive. This mode is dedicated for flashing eMMC memory or microSD card.
	BOOT (Normal)	USB bus of the CM4 module is connected to the 4-port USB hub (U251) and the USB-C data interface is disconnected (non active). After power up the CM4 will boot from the embedded eMMC memory or from microSD card (for CM4 Lite). This mode is dedicated to run Raspberry Pi OS.

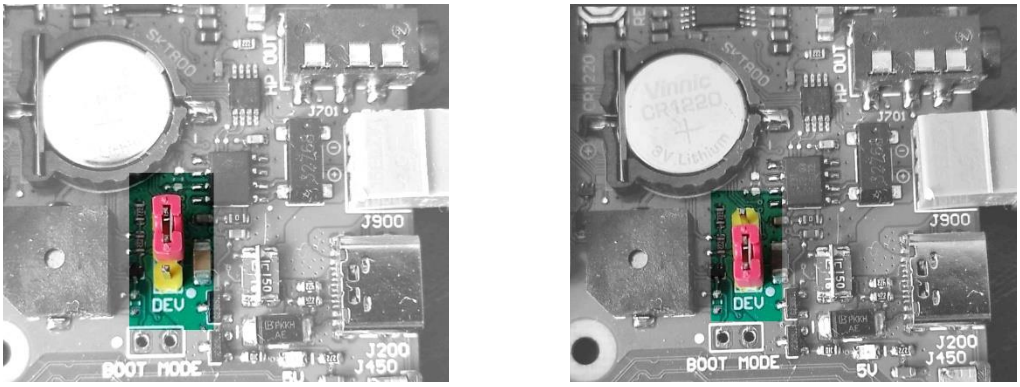


Fig. left side: BOOT mode (booting from eMMC or SD card), right side: DEVICE mode (for flashing eMMC or SD card)

For normal operation and boot from embedded eMMC memory (for CM4 module) or from microSD card (for CM4 Lite) user should set J201 jumper in the top position (shorted 2-3 pins).

For flashing the embedded eMMC memory (for CM4 module) or microSD card (for CM4 Lite) user should set J201 jumper in the bottom position (shorted 1-2 pins).

Note that the BOOT jumper should be changed only when the device is not powered. It is not recommended to change the position of the BOOT jumper while the device is running, as it may disconnect the USB interface between the CM4 module and the USB hub and may also cause unstable system operation.

There is no need to flash the microSD card only by using carrier board. User can also flash the OS image on the microSD card externally, by using PC computer and any USB card reader.

6.7. CM4 module mounting

The most important and most critical stage is the CM4 module installation and deinstallation.

User should install the CM4 module in dedicated socket (MOD100).

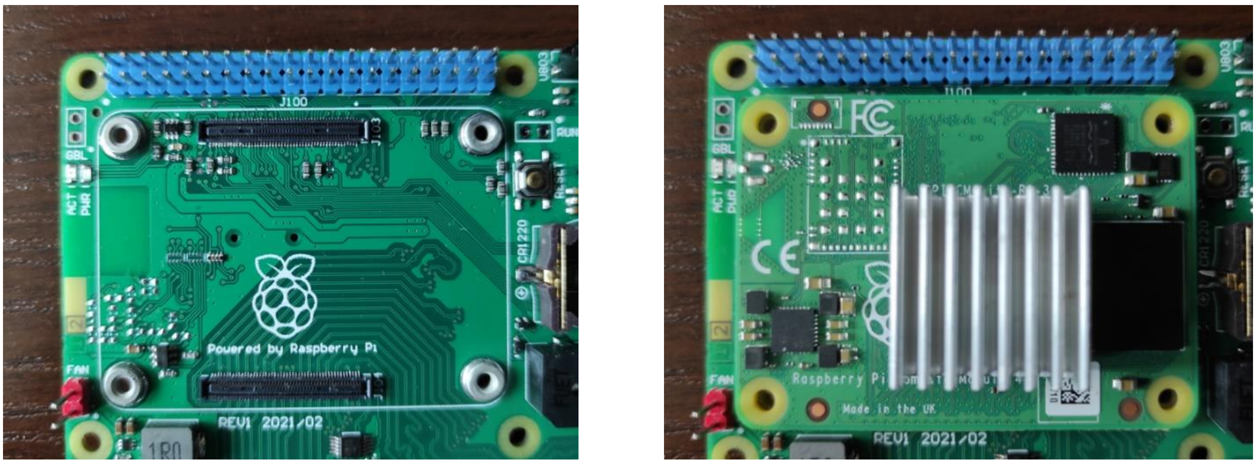


Fig. Socket for mount CM4 module (MOD100), without (on the left side) and with CM4 module placed (on the right side)

First, user should the CM4 module place in socket, and check if module is located correctly (see pictures above).

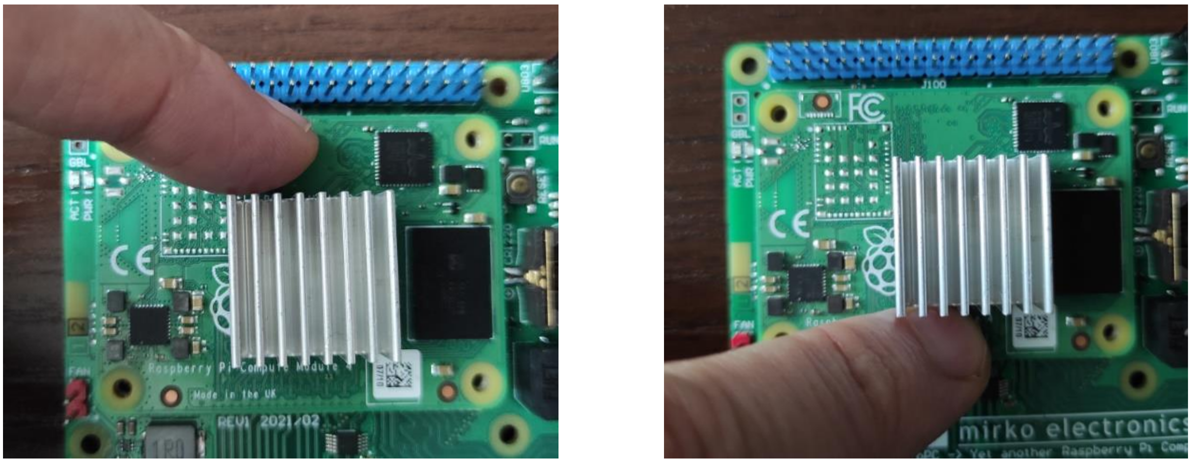


Fig. Process of manual mounting of the CM4 module

Next, user should then press down firmly on the module (preferably manually - by pressing with two fingers) where and at the same height as the first Mezzanine connector is located (J103 or J104) and then repeat in the second place (see pictures above).

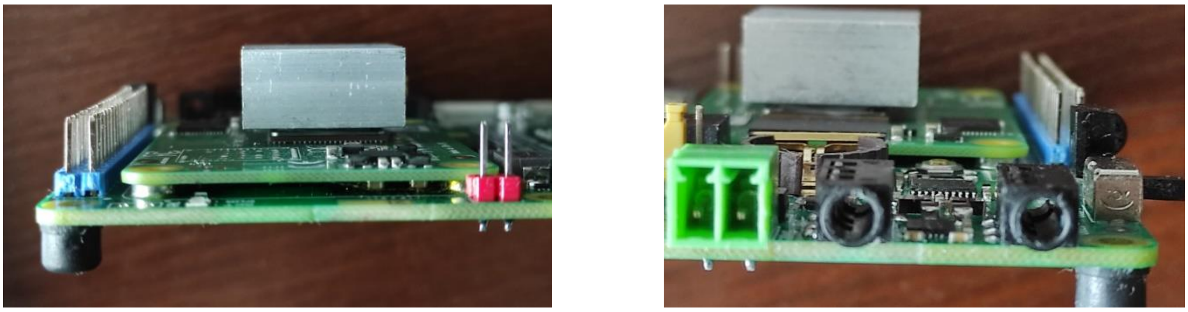


Fig. CM4 module mounting visual verification

At the end, user should check if the module is installed correctly, by checking if the PCB board of CM4 module is touching with SMD spacers in the four places and if both Mezzanine connectors are correctly inserted.

Manufacturer of the board to board connectors (microminiature mezzanine type) declares mating/unmating up to only 30 cycles.

Therefore, the number of possible installations of the CM4 module should be taken into account and the number of such installations should be limited to the necessary minimum due to the very delicate and susceptible to damage mezzanine connectors.

6.8. CM4 module disassembly

The process of disassembling the CM4 module should be carried out very carefully, without using excessive force.

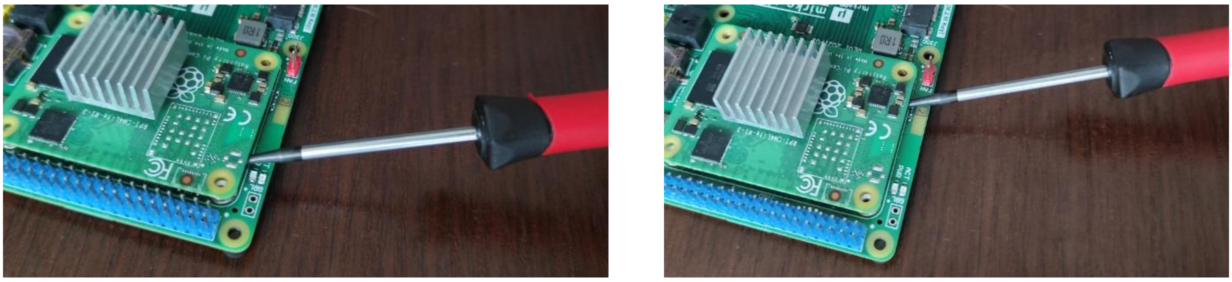


Fig. Recommended way to disassemble the CM4 module

The recommended method of disassembling the CM4 module should be performed using a long and flat screwdriver. Insert the screwdriver between the module and the carrier board close to the location of one of the mezzanine connectors, and then slowly pry the module up. Next, repeat the operation in location near to the second connector.

The number of CM4 module uninstallations should be limited to the minimum necessary.

It is not recommended to disassemble the CM4 module using only your hands, as there is a risk of excessively bending the PCB, which may damage the solder joints (e.g. BGA devices).

6.9. Turning on the carrier board

The correct sequence for powering on the board is as follows:

- make sure that the carrier board is not power sourced (at least power switch is in the right position or the power supply cable is disconnected - USC-C cable or 2-wire cable for the terminal block input power),
- for the CM4 lite module versions place the microSD card in the dedicated slot,
- set (optionally) the BOOT jumper in the required position (for flashing of for normal operation),
- connect at least the HDMI cable and connect to the monitor or display,
- when the CM4 is installed, first connect the power supply cable (USB-C cable or connect cable to 2-pin terminal block) and deliver 5VDC voltage with 1A minimum (for flashling) and 3A (for normal operation),
- next, change the power switch position from right to the left side,
- the 5V LED and also PWR/ACT LEDs will be turned on,
- next, the CM4 module will boot and the Raspberry Pi OS should start.

When you repeat this procedure again, you do not need to complete all the steps. You can skip the step of changing the BOOT jumper position, installing a microSD card, connecting a HDMI cable or power supply cable - if you only wanted to make minor changes into the setup (e.g. you connected only a MIPI camera / display, M.2 disk or just simply rebooted the OS).

6.10. Turning off the carrier board

The correct sequence for powering off the board is as follows:

- first, if the CM4 module is running the OS, shut down the OS by:
    - clicking Raspberry Pi logo -> Logout -> Shutdown,
    - executing CLI command:

```
sudo shurtdown -h now
```

- and wait until the PWR/ACT LEDs will be off (only if the CM4 module is running the OS),
- change the switch position from left to the right side,
- disconnect the power supply cable (USB-C or cable connected to the terminal block connector).

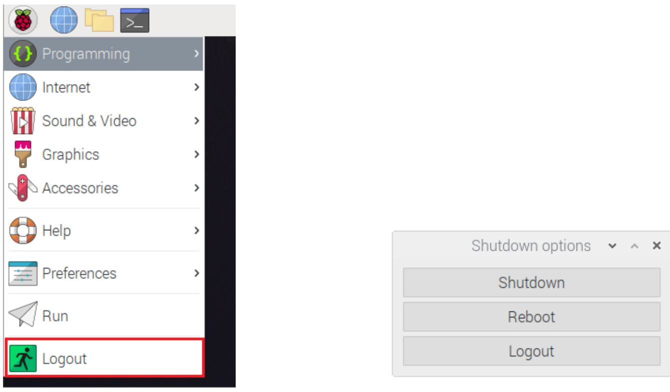


Fig. Remember to shut down the OS before turning off the power supply

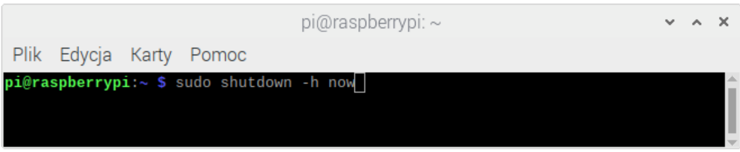


Fig. Another way to shut down the OS

Never turn off the carrier board using the power switch if the RPi OS system is still running and has not yet shut down properly. Sudden power off without shutting down the OS may damage the eMMC storage file system and can require a system reinstallation.

Likewise, never use the RESET button while the OS is running from safety reasons.

Before you will turn off the power always check if the ACT/PWR LEDs are on, this means, that in normal operation mode, OS system is running and it should be closed safely.

If you are during flashing (e.g. eMMC memory or micoSD card) always wait to complete that process.

if the CM4 module is in DEVICE mode, the ACT / PWR LEDs will not turn off automatically, you should first unmount the drive "Compute Module" from your operating system (e.g. Windows), and then turn off the power by changing the SW900 switch position.

Very important - remember that the following accessories should always be installed and removed when the carrier board power supply is turned off:

- CM4 module,
- M.2 card,
- MIPI CSI and DSI interfaces (cameras and displays).

Rest of the interfaces (such as USB devices, audio and video interfaces) can be connected and disconnected during OS is running.

6.11. USB ports

On the carrier board, there are the two kind of the USB ports, dedicated for user operation:

- 4x standard USB 2.0 ports (type A connectors, J250 and J251, available in host mode),
- USB-C port with implemented only USB 2.0 interface (J200 connector).

During normal operation (when the Raspberry Pi OS is working) the 4x USB type A ports working as normal USB host ports, which means user can connect to them all the standard USB accessories (such as USB keyboards, mouses, pendrives).

During normal operation the USB-C port is not available for user (is disconnected from CM4 module) and can be used in that mode only as 5V power supply source.

The USB-C port is active and available for user only in the DEVICE mode (see BOOT jumper settings) and only for flashing the ROM memory (onboard eMMC or microSD card). In that specific case, USB 2.0 interface from the USB-C port is connected directly to the USB port of the CM4 module configured as the DEVICE mode, and its ROM memory (embedded eMMC or microSD card) is mounted on the USB as mass storage device (seen as another disk drive).

6.12. Plug & play accessories

There are only the two kind ports are hot-pluggable and plug & play type (can be plug during powering the device and can be automatically detected and installed by OS after plug):

- gigabit Ethernet port (RJ45 connector, J500),
- 4x USB ports (type A connectors, J250 and J251).

During RPi OS running user can connect on-the-fly any standard USB devices and accessories, such as:

- keyboards, mouses,
- pendrives, memory card readers, external disk drives (HDD or SSD types),
- sound cards,
- GSM modems, wireless cards (e.g. WiFi),
- any standard USB devices, which can be detected and installed in Raspberry Pi OS (Linux compatible).

Audio jacks, such as line out and headphone out can be also connected and disconnected during normal operation and when the device is powered without any restrictions.

Other interfaces and ports cannot be installed and removed during the carrier board powered (even if they can be detected if OS running) and until power supply is turned off, due to this **may damage the hardware**.

These are, in particular, the following ports:

- CM4 module socket,
- M.2 socket,
- MIPI CSI and DSI interfaces (cameras and displays),
- HAT boards connected to the 40-pin GPIO port.

By default, HDMI ports should only be connected and disconnected when the device is powered off. HDMI ports require a explanation in a separate chapter.

6.13. HDMI ports

By the default configuration, HDMI ports should be connected only when the device is powered off and before the CM4 module will boot and OS system will start (they are detected and configured during OS booting). Otherwise, if the user connects the HDMI port only after system booted, then the video output will not be active (it will be disabled).

To change this default setting user should change the **config.txt** file and uncomment this line:

```
hdmi_force_hotplug=1
```

For more information, please read:

https://www.raspberrypi.org/documentation/configuration/config-txt/video.md

6.14. M.2 socket

The main purpose of the M.2 socket is to install standard SSD NVMe drives with the PCIe type interface (not the old generation SATA interface). User can install two different M.2 card types: 2242 or 2280 form factor.

You can also install cards other than SSDs (e.g. Coral TPU or GSM modules), however, before installing, check if your OS supports this device type.

Remarks:

1. M.2 cards and modules are supported only with PCIe interface implemented, and other types are not supported (especially with SATA interface). Before installation please check, which interface type has your M.2 card.
2. Supported M.2 card form factors: 2242 or 2280 with M- or B+M Key.
3. Only M.2 cards are supported, powered by the 3.3VDC standard. Before installation please check, which power supply standard has your M.2 card.
4. For the M.2 socket, the maximum possible power consumption is set to 3A, however, for long-term and reliable operation, it is recommended to use M.2 energy-efficient cards specified as e.g. 1A (or up to 1.5A).
5. M.2 card should be always mounted and deinstalled in socket only during the device is powered off.
6. List of the supported M.2 devices can be found here:
   https://pipci.jeffgeerling.com/

6.15. 5VDC power supply

MirkoPC carrier board should be supplied by using the nominal 5VDC voltage (range 4.95V to 5.20V), and specified as:

- at least 1.0A during flashning ROM memory (DEVICE mode) – for flashing user can use the standard USB-C cable as power input and data and connect to standard PC or laptop,
- max. 3.0A for USB-C connector during normal operation (specified by USB-C standard),
- max. 5.0A for 2-pin green socket, during normal operation (max. current not tested yet).

Device can be power supplied by using two different connectors:

- standard 2-pin terminal block socket (marked as J900),
- or USB-C connector (marked as J200).

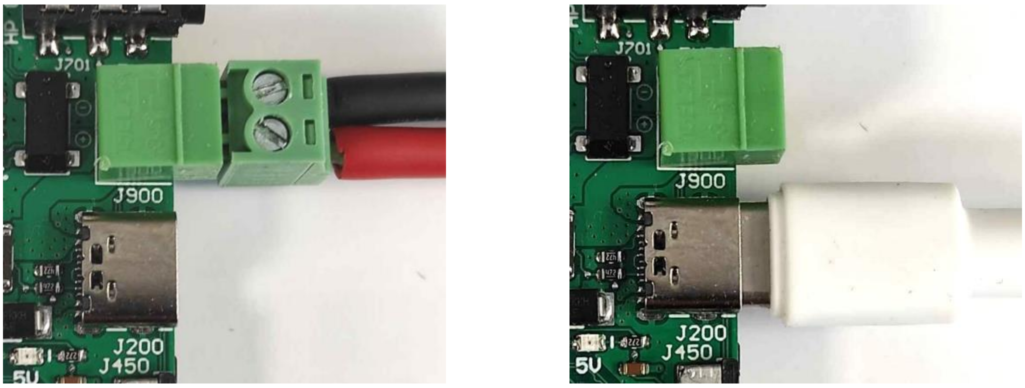


Fig. Possible methods power supply of the device

During flashing the CM4 module user should use only the USB-C port as power input (and also for data). Is not reccomended to use the 2-pin power port in DEVICE mode or/and during flashing CM4 module, due to for USB connection between CM4 module and USB host (PC) is required to use USB-C port.

It is not allowed to connect both power ports at the same time and power the device from two different power sources, due to the difficult to predict the current flow and possible damage to the device.

It should be remembered that during normal operation, when powering the device from a source with low current efficiency (e.g. a PC), voltage drops may appear on the 5V supply rail and this may cause OS warning messages to appear.

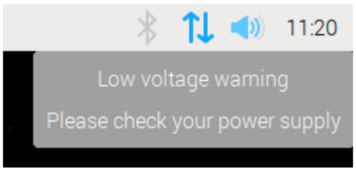


Fig. Occasional low voltage warnings due to low current efficiency of the power source

The above messages are not dangerous, they can only be a bit annoying. To avoid their appearance, the user can use a power source with high current efficiency (e.g. 5A) and a stable output voltage, or safely raise the supply voltage slightly to a value in the range of 5.05 - 5.15V.

For example, Xiaomi mobile chargers with a USB type A port provide an output voltage of 5.10V under a load of up to 3A and can be a good example of a stable power supply.

6.16. 5VDC Fan socket

MirkoPC support installation of a basic fan. User can connect to the dedicated 2-pin header socket (J800, 2.54mm pitch) a simple 5VDC fan.

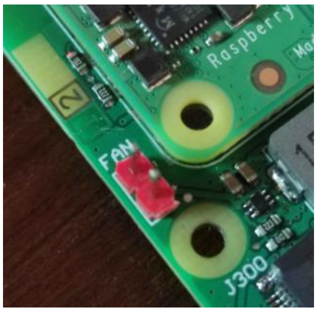


Fig. 5VDC fan socket (pin 1/plus marked by little white dot)

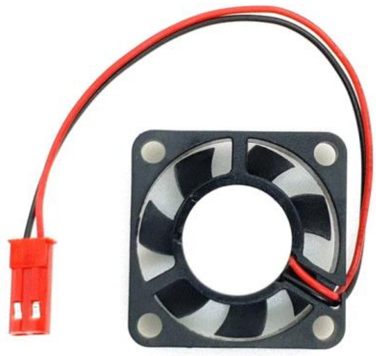


Fig. Example of basic fan that can be connected to the J800 fan socket

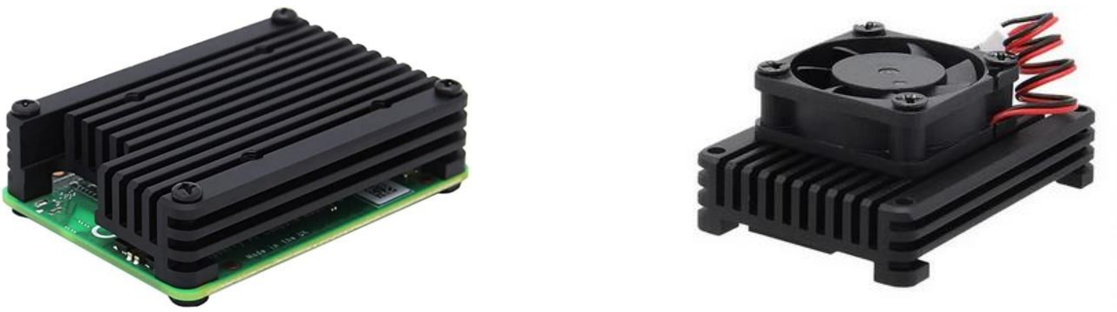


Fig. Recommended heatsink for CM4 module (left side) with optional fan (right side)

7. CM4 image flashing

MirkoPC support both CM4 modules:

- standard CM4 version with onboard embedded eMMC memory, and
- CM4 Lite version (with external connected microSD card or eMMC v5.1 BGA IC chip).

Before you will start:

- make sure that device is unpowered,
- the CM4 module is installed correctly,
- microSD card is placed (only required for the CM4 lite version),
- prepare PC computer (or laptop) with MS Windows OS and with installed applications:
    - RPiBoot.exe
    - Raspberry Pi Imager

Detailed instruction, how to download and flash the OS image into the eMMC or SDCard, is as following:

1. set BOOT jumper (J201) in the DEVICE position,
2. connect the USB-C cable (power/data) between carrier board (J200) and PC computer with installed Windows OS (e.g. laptop),
3. turn on the device,
4. 5V LED and CM4 LEDs ACT/PWR will be on,
5. run „RPiBoot.exe” application (for a while you can see executing of that application, as on the picture below),

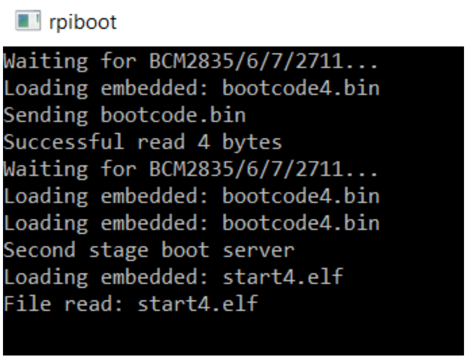


Fig. Executing „RPiBoot.exe” application

6. CM4 will be installed (exactly, its eMMC memory or the microSD card attached to it) as mass storage drive (as next disk drive),
7. next, run "Raspberry Pi Imager" application, select proper Raspberry Pi OS image, select also storage, which you would like to flash (CM4 module will be shown as mass storage drive), and press „Write” to download and flash OS image into the CM4 module (please see next pictures),

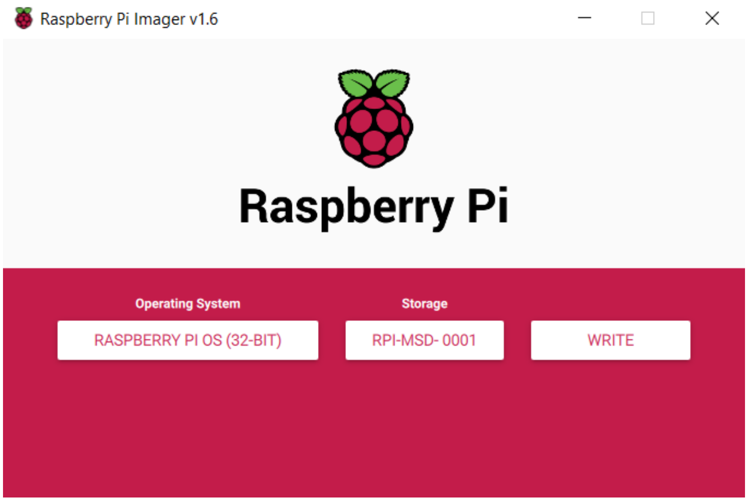


Fig. „Raspberry Pi Imager" application

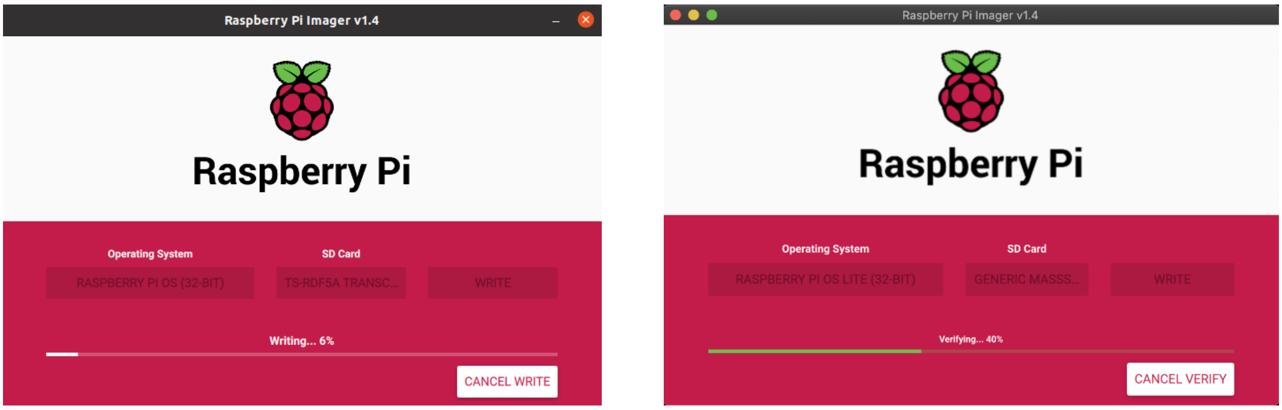


Fig. „Raspberry Pi Imager" application during writing and verifying the image

8. safely unmount the CM4 drive, turn off and turn on the power supply again and optionally make steps 9. to 11.,
9. run again the „RPiboot.exe" application,
10. open the CM4 disk, find and modify **config.txt** file (at least for adding USB hub support),
11. safely unmount the CM4 drive, turn off the device again,
12. set the BOOT jumper in the BOOT position (normal operation, boot from embedded memory and run installed RPi OS),
13. connect at least HDMI output to the display/monitor and connect dedicated power supply cable,
14. turn on again the device again.

More information you about flashing and to download rewuired software, can find here:

https://www.raspberrypi.org/documentation/hardware/computemodule/cm-emmc-flashing.md

https://www.raspberrypi.org/software/

8. First run

Before the device first run please read carefully chapters especially about CM4 mounting, BOOT jumper, power supply switch, turning on the power supply sequence, and how to flash the CM4 module.

Correct sequence of the first run is as following:

1. make sure that all the cables are unconnected and the power switch is in off position,
2. install CM4 or CM4 Lite module,
3. optionally mount microSD card (only for CM4 Lite module),
4. set the BOOT jumper in DEVICE position,
5. connect at least one HDMI port,
6. connect the USB-C port between the carrier board and PC, turn on the device and optionally process the flashing CM4 module operation (or if you have flashed CM4 module or prepared microSD card with RPi OS, just skip the steps about flashing and the DEVICE mode),
7. turn off and turn on the device again in the DEVICE mode and change at least the config.txt file for USB hub configuration and optionally make any changes that you require (e.g. for audio DAC configuration),
8. turn off the device, optionally install any needed interfaces, devices or accessories (e.g. M.2 NVMe disk, MIPI cameras or displays),
9. set jumper in the BOOT mode, and turn on the device again connected to the dedicated power source with enough current supply, and connect optionally USB mouse and keyboard for easier operation and setup,
10. wait a while until OS will boot,
11. make sure that Raspberry Pi OS is entirely up to date. This ensures that we will be utilizing all the latest software available. Please run CLI commands:

    ```
    sudo apt-get update
    sudo apt-get upgrade
    ```
12. make any required configuration (e.g. setup RTC clock, MIPI cameras and displays) and install any needed software.

9. Raspberry Pi OS configuration

After downloading and first boot the Raspberry Pi OS image some interfaces are disabled (not working) by using the default configuration (please see the table below). For e.g. the USB interface is disabled to save power by default on the CM4, so the USB hub (U251 device) connected to its USB ports (J250 and J251 connectors) are also disabled.

To activate turned off by default interfaces, Raspberry Pi OS image needs changing configuration by modifying system files or by using special commands to install drivers or apps.

Below you can find more details on how to configure OS image for a specific interface.

Remark: most of the described here tips are dedicated to the default Raspberry Pi OS (previously called Raspbian OS) and potentially may not be applicable for other OS images (e.g. LibreELEC, RiscOS, or Ubuntu) due to the differences in system files and configuration.

All the configurations described in this chapter should be used only for a basic setup (these are not the best and optimal settings) and allow users only to test all the functionalities, interfaces, and peripherals.

An advanced Raspberry Pi user can configure peripheral devices for their needs by using his best experience. For e.g. to achieve the best audio quality and setup for Audio DAC user should use high-resolution audio files and a media player, and also should configure the PCM interface to set the highest data resolution and sample rate.

Interface	Reference	Default configuration
USB Hub	J250/J251	Disabled (not working)
M.2 socket	J300	Enabled
MicroSD card slot	J600	Enabled
HDMI #0	J400	Enabled
HDMI #1	J401	Enabled
RJ45 (Ethernet)	J500	Enabled
Camera #0 (MIPI CSI)	J450	Disabled (not working)
Camera #1 (MIPI CSI)	J451	Disabled (not working)
Display #0 (MIPI DSI)	J452	Disabled (not working)
Display #1 (MIPI DSI)	J453	Disabled (not working)
USB-C (data)	J200	Enabled
Buzzer	BUZ800	Disabled (not working)
RTC clock	U800/SKT800	Disabled (not working)
Audio DAC	J700/J701	Disabled (not working)
Temperature sensor	U802	Disabled (not working)
IR remote sensor	U803	Disabled (not working)
Status LEDs	LED800/LED801	Disabled (not working)
5VDC Fan socket	J800	Enabled

Tab. Raspberry Pi OS - interfaces supported by default

9.1. USB Hub

By default the USB interface is disabled.

For most RPi OS systems (LibreELEC excluded), to enable it you need to configure device tree overlay by adding to the **config.txt** file:

```
#usb
dtoverlay=dwc2,dr_mode=host
```

References:

https://datasheets.raspberrypi.org/cm4/cm4-datasheet.pdf

https://github.com/raspberrypi/firmware/issues/1500

https://www.jeffgeerling.com/blog/2020/usb-20-ports-not-working-on-compute-module-4-check-your-overlays

For the LibreELEC OS to enable the USB interface is needed to download the latest nighly image (not the official release, e.g. 9.2.6) and to add to the **config.txt** file:

```
#usb
otg_mode=1
```

References:

https://forum.libreelec.tv/thread/23185-compute-module-4/

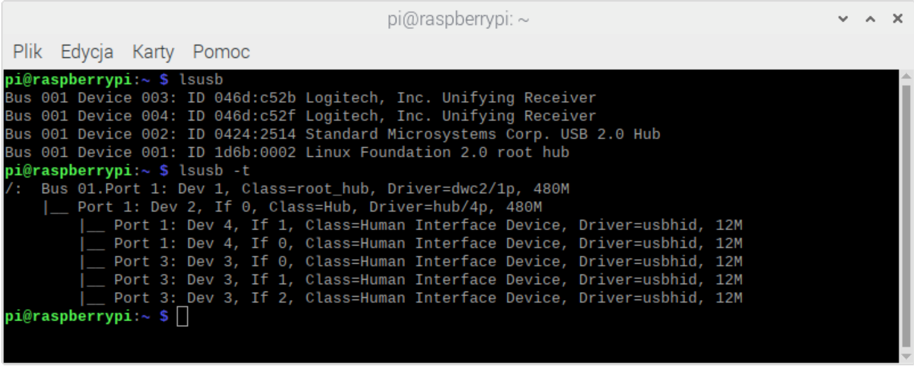
https://github.com/LibreELEC/LibreELEC.tv/issues/4722

You can list USB bus and all the devices connected by using commands:

```
lsusb
```

or

```
lsusb -t
```



References:

http://helloraspberrypi.blogspot.com/2013/11/lsusb-list-usb-buses-and-devices.html

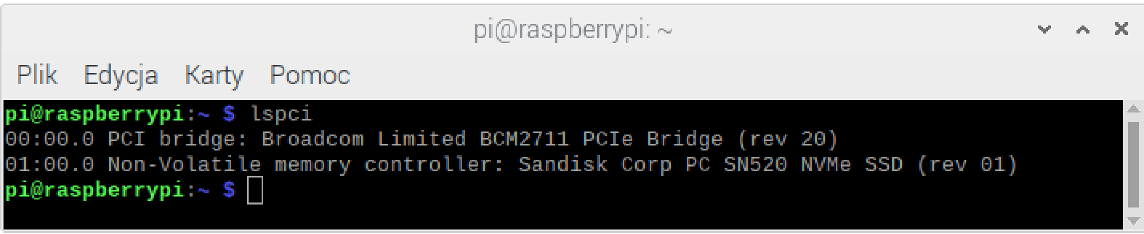
9.2. M.2 socket

9.2.1. Configuration

By default, standard M.2 SSD NVMe drives should be correctly detected and installed by Raspberry Pi OS during boot.

You can check and read hardware information of installed devices on the PCIe bus and installed in M.2 socket by using following command (below the picture is shown an example of a detected NVMe SSD drive from Sandisk, SN520 model):

```
lspci
```



9.2.2. NVMe drive as boot partition

Note that the NVMe SSD drive can be used as the boot partition, here are the references:

https://www.raspberrypi.org/documentation/hardware/raspberrypi/bootmodes/nvme.md

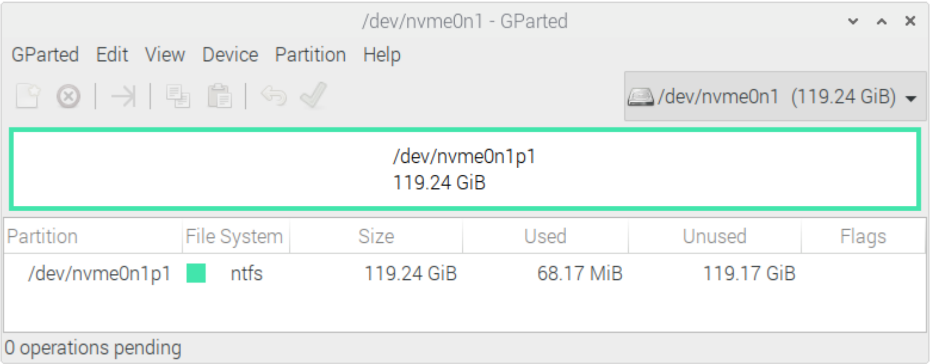
https://www.jeffgeerling.com/blog/2021/raspberry-pi-can-boot-nvme-ssds-now

https://www.youtube.com/watch?v=l5hDeibtQuo

9.2.3. NVMe drive mounting

A newly installed and non-formatted drive should be prepared and formatted by using "GParted" tool application and NTFS file system setup (see the picture below – an axample of the NVMe drive formatted as NTFS file system). Users can create as many different partitions as following their needs.

Then NVMe drive can be easily installed in Raspberry Pi OS as another drive.



9.3. RTC clock

As the RTC clock, MirkoPC board uses the IC PCF8523 chip from NXP Semiconductor (marked as U800 device).

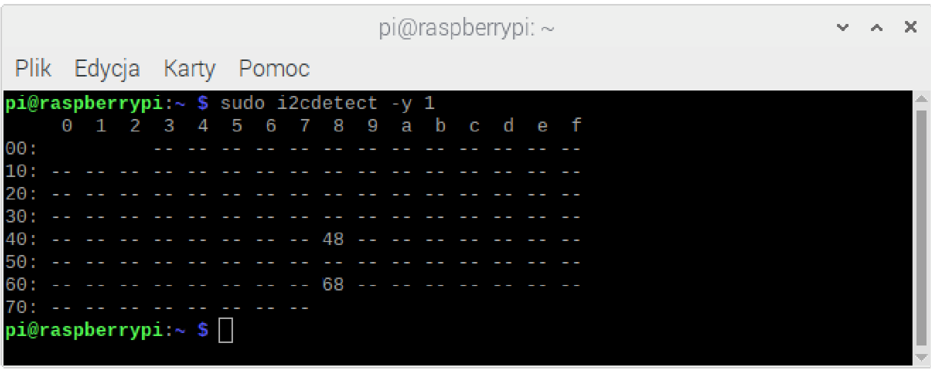
By default, the onboard embedded RTC clock is disabled.

RTC clock chip is connected to the I2C interface – I2C1 bus (used GPIO2 and GPIO3 pins).

You can simply check if the RTC clock is detected on the I2C1 bus by using command:

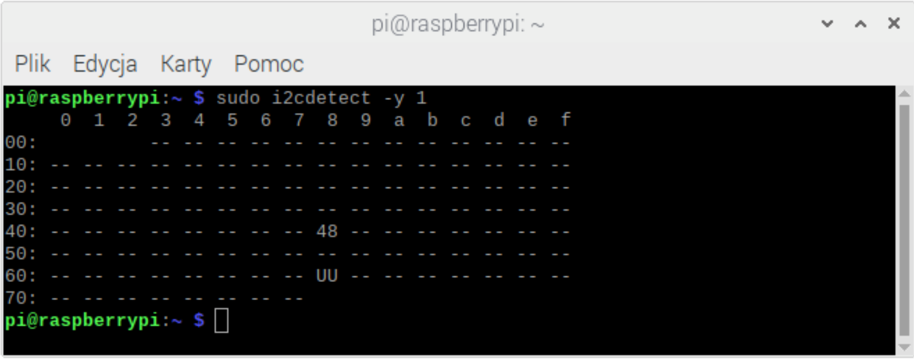
```
sudo i2cdetect -y 1
```

RTC clock should respond at the address 0x68, as below (present as number 0x68):



Next, the RTC clock should be installed by using one of the tutorials described below. After installation RTC clock should be present at the address 0x68 by string „UU" by using the same command:

```
sudo i2cdetect -y 1
```



```
pi@raspberrypi:~ $ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- 48 -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- UU -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
pi@raspberrypi:~ $
```

To check if the RTC clock is working properly with the battery backup, you can turn off the power supply from the board and wait a few minutes and then again turn on the power supply and check if the system time and date are correct. Remember that the Ethernet network (or Wireless connection) should be disconnected, due to OS can get and set the current time from the Internet.

References:

Here are the detailed tutorials on how to install and configure the RTC clock for Raspberry Pi OS:

https://pimylifeup.com/raspberry-pi-rtc/

https://learn.adafruit.com/adding-a-real-time-clock-to-raspberry-pi?view=all

9.4. IR receiver

On the MirkoPC board is mounted the IR receiver, located near to the power switch buton. That receiver is used for receiving data from standard IR remote controllers, attached to audio and video equipment, and can be used in many Raspberry Pi OS (e.g. LibreELEC, Volumio) to remote control and manage multimedia applications such as Kodi media player.

As the IR receiver, the carrier board uses a standard 36kHz device (marked as U803 device) and can receive IR data from different frequency range 36, 38 or even 40 kHz (depended on the maximum range of the remote control transmission).

By default, the onboard embedded IR sensor is disabled.

IR sensor is connected to the GPIO17 pin.

First, you should enable GPIO17 pin to receive the IR data – configure the device tree overlay by adding to the **config.txt** file:

```
#IR receiver

dtoverlay=gpio-ir,gpio_pin=17
```

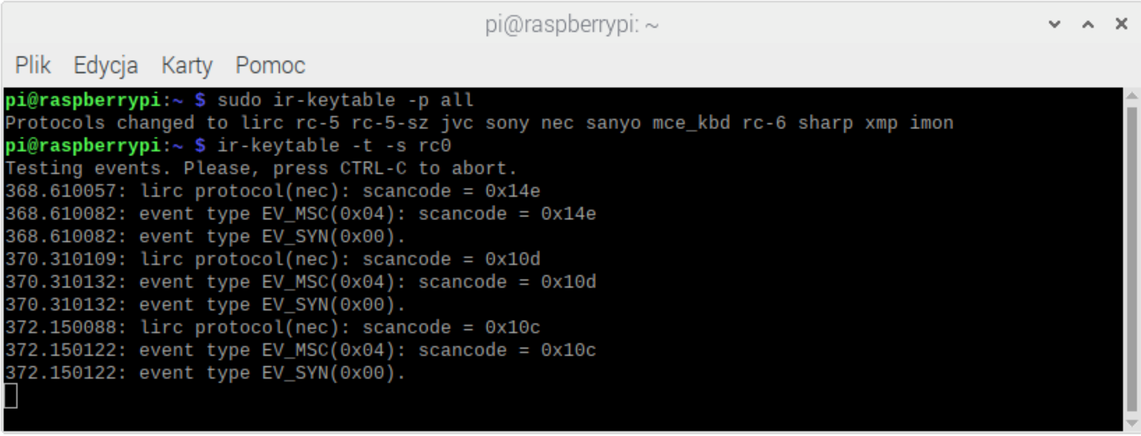
Next, reboot the OS and install IR receiver by using tutorial listed below:

https://blog.gordonturner.com/2020/05/31/raspberry-pi-ir-receiver/

By using commands, as following, you can use and test any standard remote controller and check if the IR sensor can receive and decode these data:

```
sudo ir-keytable -p all

ir-keytable -t -t rc0
```



In the picture above, you can see an example of testing the IR remote controller (decoded NEC frames) in Raspberry Pi OS.

9.5. Status LEDs

By default Status LEDs (LED800 and LED801) are not controlled by OS. To change the Status LEDs state user should change proper GPIO pins (GPIO16 or GPIO26) by control them to 0 or 1 logic state.

The easiest way to do that is to use dedicated Python script.

Here is an example of python script to test the Status LEDs and also the Buzzer by control assigned GPIO pins:

https://github.com/mfolejewski/MirkoPC/blob/main/python/gpio_test.py

9.6. Buzzer

By default Buzzer (BUZ800) is not controlled by OS. To change the Buzzer state user should change proper GPIO pin (GPIO22) by drive its to 0 or 1 logic state.

The easiest way to do that is to use dedicated Python script.

Here is an example of python script to test the Buzzer and also Status LEDs by control assigned GPIO pins:

https://github.com/mfolejewski/MirkoPC/blob/main/python/gpio_test.py

9.7. Temperature sensor

By default the temperature sensor (U802, LM75B IC device) is not controlled by OS. To read the current temperature from sensor user should read I2C registers from I2C1 bus.

The easiest way to do that is to use dedicated Python script.

Here is an example of python script to read the current temperature from dedicated onboard installed temperature sensor (U802):

https://github.com/mfolejewski/MirkoPC/blob/main/python/lm75_read.py

In the picture above, you can see the result of reading current temperature from dedicated sensor.

9.8. HDMI outputs

The default configuration of the Raspberry Pi OS (previously called as Raspbian OS) allows to work both HDMI outputs (#0 and #1 ports) in the same time.

If the OS configuration is changed, so that the port, eg HDMI # 1, will be disabled, to restore the configuration, modify the DTS file and compile it, and then reboot the system to update the changes.

To compile DTS file into dt-blob.bin file user should execute command, as following (an example):

```
sudo dtc -I dts -o /boot/dt-blob.bin /boot/hdmi.dts
```

Here you can find the DTS file, for access and activate both HDMI ports:

https://github.com/mfolejewski/MirkoPC/blob/main/DTS_files/hdmi.dts

References:

https://www.raspberrypi.org/documentation/configuration/device-tree.md

https://www.raspberrypi.org/documentation/configuration/hdmi-config.md

9.9. MIPI CSI-2 camera and DSI display

User can install up to 2x dedicated MIPI CSI-2 cameras and up to 2x dedicated MIPI DSI displays. These devices can be connected to the carrier board by using FFC/FPC 22-pin connectors:

- J450 – camera #0 input, 2-lane MIPI CSI-2 standard, 22-pin FFC/FPC connector (0.5mm pitch),
- J451 – camera #1 input, 4-lane MIPI CSI-2 standard, 22-pin FFC/FPC connector (0.5mm pitch),
- J452 – display #0 output, 2-lane MIPI DSI standard, 22-pin FFC/FPC connector (0.5mm pitch),
- J453 – display #1 output, 4-lane MIPI DSI standard, 22-pin FFC/FPC connector (0.5mm pitch).

Most of the market MIPI cameras and displays have 15-pin standard FFC/FPC connector interface. To connect these accessories to the MirkoPC carrier board user should use special, dedicated 15-22pin FFC/FPC flex cable or use dedicated 15 -> 22pin adapter.

By default all MIPI cameras and displays are disabled.

To turn on MIPI devices user should modify and compile the DTS file into dt-blob.file by using special command (example):

```
sudo dtc -I dts -o /boot/dt-blob.bin /boot/dsi0_cam1.dts
```

Here you can find the examples of the DTS files, for test, access and activate MIPI ports – DSI displays and CSI-2 cameras:

https://github.com/mfolejewski/MirkoPC/blob/main/DTS_files/disp0-cam2.dts

https://github.com/mfolejewski/MirkoPC/blob/main/DTS_files/disp1-cam2.dts

Remember, that there are only 3 HVS channels, so you can only ever run 3 displays at a time. Run DSI0, DSI1, HDMI0, and HDMI1 all simultaneously is therefore impossible.

The firmware is limited to 2 displays simultaneously as it reserves the 3rd HVS channel for the transposer (used for offscreen composition).

The original RPi OS firmware won't drive more than one DSI display, so user should looking at the ARM-side KMS/DRM display drivers, starting with this overlay:

https://github.com/raspberrypi/linux/blob/rpi-5.10.y/arch/arm/boot/dts/overlays/vc4-kms-dsi-7inch-overlay.dts

References:

https://www.raspberrypi.org/documentation/hardware/computemodule/cmio-camera.md

https://www.raspberrypi.org/documentation/configuration/camera.md

https://www.raspberrypi.org/documentation/configuration/pin-configuration.md

https://www.raspberrypi.org/documentation/raspbian/applications/camera.md

https://www.raspberrypi.org/documentation/configuration/device-tree.md

9.10. Audio DAC

As the the Audio DAC chip, MirkoPC board uses the IC PCM5100A from Texas Instruments (marked as U700 device).

By default the onboard embedded Audio DAC and headphone amplifier are disabled (not active).

Audio DAC is connected by using the CPU’s GPIO pins and configured as the PCM interface (and set in I2S mode), as following:

- GPIO18 as PCM CLK (I2S BCK signal),
- GPIO19 as PCM FS (I2S LRCK signal),
- and GPIO21 as PCM DOUT (I2S DAT signal).

To enable the PCM interface user should modify the **config.txt** file (and use „hifiberry-dac” device tree overlay):

1. disable the default audio out by commenting out line:

```
#dtparam=audio=on
```

2. turn on the I2S interface and configure device tree overlay by adding these lines:

```
#I2S DAC
dtparam=i2s=on
dtoverlay=hifiberry-dac
dtdebug=1
```

3. reboot OS.

You can test if audio DAC is configured and installed by using command:

```
sudo aplay -l
```

References:

https://www.hifiberry.com/docs/software/configuring-linux-3-18-x/

https://blog.himbeer.me/2018/12/27/how-to-connect-a-pcm5102-i2s-dac-to-your-raspberry-pi/

9.11. RTC Battery

If the RTC clock is properly configured in Raspberry Pi OS, then the battery backup no needs any additional steps.

As the battery backup should be used only the CR1220 lithium coin battery, and installed in the dedicated socket (marked as SKT800). Users should install the RTC battery only when the carrier board is not powered to avoid any untended risk of circuit shorts.

9.12. 5VDC Fan socket

For connecting Fan and for cooling the CM4 module (especially the main CPU) user can use a simple, dedicated 2-pin fan socket, which can deliver stable, 5VDC power supply for fan mounted on the CM4 module heatsink.

Fan socket is working all the time and not require a special OS config. On the PCB this connector has marked two pins for proper polarity and fan connecting (pins marked ad plus and minus).

Please read also the chapter: User interface -> 5VDC Fan socket

10. Raspberry Pi OS systems

User can download and install on the CM4 module a lot of 32- and 64-bit operating systems, such as the official Raspberry Pi OS (previously called Raspbian), highly modified and customized OS (DietPi, RISC OS) or dediated for special purposes: as media player (LibreELEC), as music player (Volumio, MoodeAudio) or game emulator (RetroPi).

OS	Status	Version	Interfaces	Issues
Raspberry Pi OS (32-bit) (previously Raspbian)	Tested	Release: 2021-03-04	100% tested	No
LibreELEC (KODI)	Tested	9.2.6	Not tested	USB hub issue (USB hub not working)
LibreELEC (KODI)	Tested	Nightly build 10.0, 20210331	Partially tested	USB hub working (required different setup)
DietPi	Partially tested	7.0.2	Partially tested	USB hub, audio DAC and M.2 socket are working
RISC OS Pi	Partially tested	?	Partially tested	USB hub is working
Volumio (music player)		2.873 2021-02-19	Partially tested	USB hub, audio DAC and M.2 socket are working
MoodeAudio (music player)	Not tested	-	-	Too long configuration, not tested
RuneAudio (music player)	Not tested	-	-	No official image for RPi4/CM4 yet
Manjaro ARM	Tested	21.02 (KDE Plasma)	Partially tested	USB hub, M.2 socket and audio DAC are working

Tab. Tested Raspberry Pi OS systems

11. Hardware status

So far all the interfaces were tested and almost all functionalities. In the table below, described current test status and all found issues.

ID	Object	Reference	Result	Remarks
#01	HDMI #0 – Full HD	J400	OK	-
#02	HDMI #1 – Full HD	J401	OK	-
#03	Gigabit Ethernet RJ45	J500	OK	-
#04	USB hub + USB ports (#1 -#4)	J250/J251	OK	-
#05	CM4 module	J103/J104	OK	-
#06	CM4 Lite module	J103/J104	OK	-
#07	CM4 with WiFi/BT module	J103/J104	OK	-
#08	ACT/PWR CM4 LEDs	LED100/LED101	OK	-
#09	RESET switch	SW100	OK	-
#10	RTC clock	U800	OK	-
#11	Battery backup for RTC clock	SKT800	OK	-
#12	User status LEDs	LED800/LED801	OK	-
#13	Audio DAC	U700/J700	OK	-
#14	Headphone amplifier	U701/J701	OK	-
#15	IR receiver	U803	OK	-
#16	Camera CSI #0	J450	OK	-
#17	Camera CSI #1	J451	OK	-
#18	Display DSI #0	J452	OK	-
#19	Display DSI #1	J453	OK	-
#20	MicroSD slot	J600	OK	little difficult MicroSD card installation
#21	eMMC BGA chip	U600	OK	-
#22	M.2 socket (SSD NVMe drive)	J300	OK	dedicated 3.3V power rail requires further testing
#23	M.2 socket – 2242/2280 sizes	J300	OK	-
#24	M.2 socket – Coral TPU module	J300	FAIL	not supported yet by Raspberry Pi OS
#25	5V fan socket	J800	OK	-
#26	Buzzer	BUZ800	OK	-
#27	Boot jumper	J201	OK	-
#28	5V status LED	LED900	OK	-
#29	5V input (terminal block)	J900	OK	possible drop voltages on 5V power rail
#30	USB-C (as power supply)	J200	OK	possible drop voltages on 5V power rail
#31	USB-C (data)	J200	OK	CC pins issue (see Raspberry Pi 4B issue)
#32	Power switch	SW900	OK	-
#33	HAT GPIO connector	J100	0%	require testing
#34	HDMI – dual monitor mode	J400/J401	OK	-
#35	HDMI – 4K UHD mode	J400/J401	OK	-

Tab. Performed hardware tests

All the carrier board interfaces were successfully tested. So far, no major problems with the operation of interfaces have been noticed and found. Confirmed and verified that all the PCB footprints are OK and also the electrical schematic is also correct (there are a few minor issues, but they are not so important).

12. GPIO Interface (HAT connector)

12.1. Introduction

MirkoPC has 40-pin GPIO connector (J100), electrical and mechanical compatible with the Raspberry Pi 4B GPIO header standard and is backwards compatible with all previous Raspberry Pi boards with a 40-way header.

That connector has connected 28 GPIO lines of the Broadcom BCM2711 CPU and it can be used to connect any standard HAT compatible board.

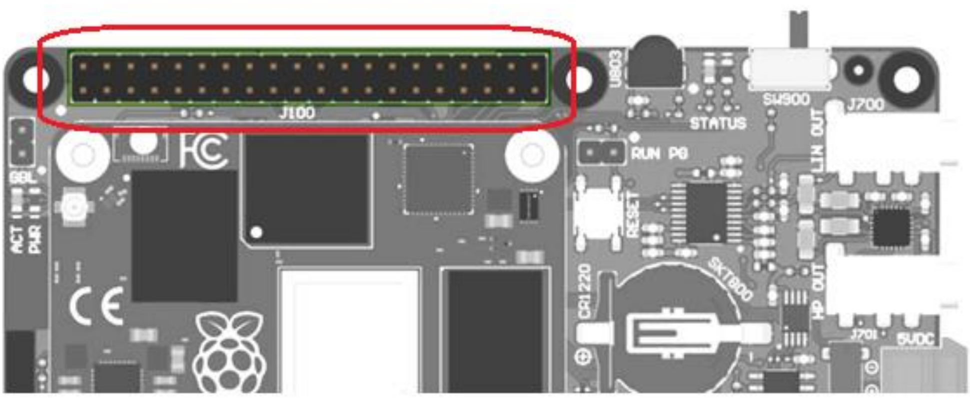


Fig. GPIO interface connector (J100) – red marked

12.2. GPIO pin assigment

On the pictures below, you can find out how the GPIO pins are connected and in which order to the pin header connector (HAT interface).

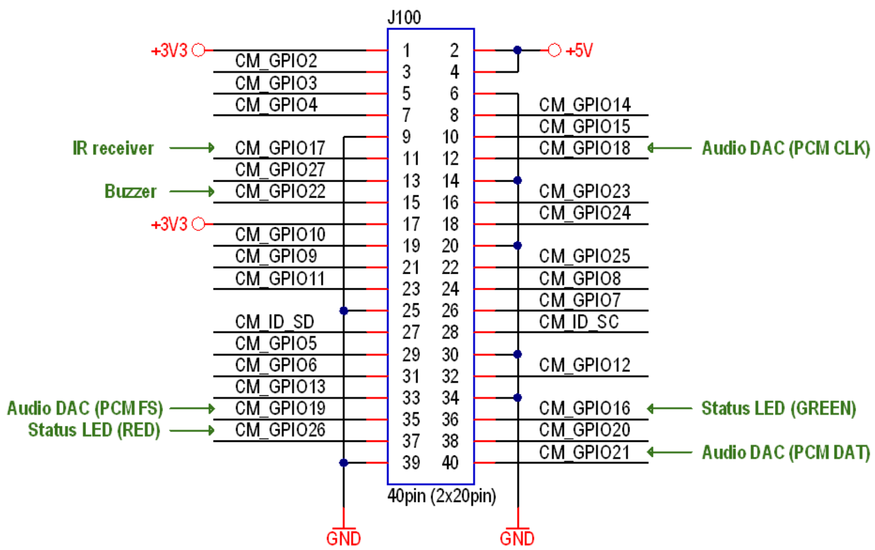


Fig. GPIO pin assigment of the J100 connector (schematic)

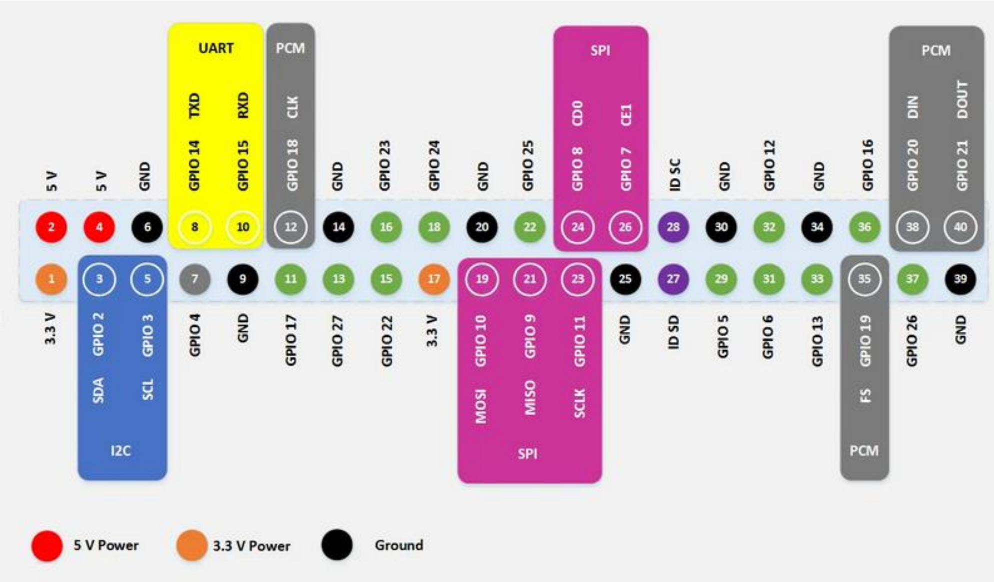


Fig. GPIO pin assigment of the J100 connector (legend)

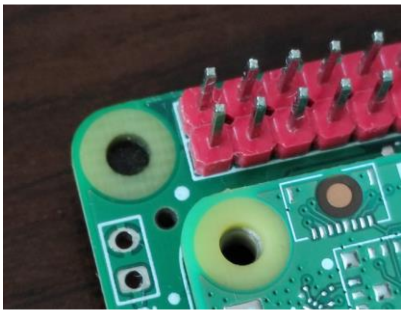


Fig. GPIO header pin #1 marked as little white dot

12.3. Reserved GPIO pins

Some GPIO pins are reserved and used for internal purpose, such as for interfacing audio DAC (PCM interface), I2C1 bus, and a few GPIO lines as basic digital input or output to control components, such as buzzer, status LEDs and IR sensor receiver.

GPIO pin	Used as interface	Used to control
GPIO2	I2C1 SDA	RTC clock (U800), temperature sensor (U802)
GPIO3	I2C1 SCL	
GPIO18	PCM CLK (BCK)	Audio DAC (U700)
GPIO19	PCM FS (LRCK)	
GPIO21	PCM DOUT (DAT)	
GPIO17	Digital input	IR receiver (remote control) – U803
GPIO22	Digital output	Buzzer (BUZ800)
GPIO16	Digital output	Status LED (Green) – LED800
GPIO26	Digital output	Status LED (Red) – LED801

Tab. Used GPIO lines for internal purpose

12.4. How to disconnect reserved GPIO pins

By removing some component user can disconnect and free the GPIO pins which are used for internal usage (e.g. for Audio DAC, status LEDs, buzzer).

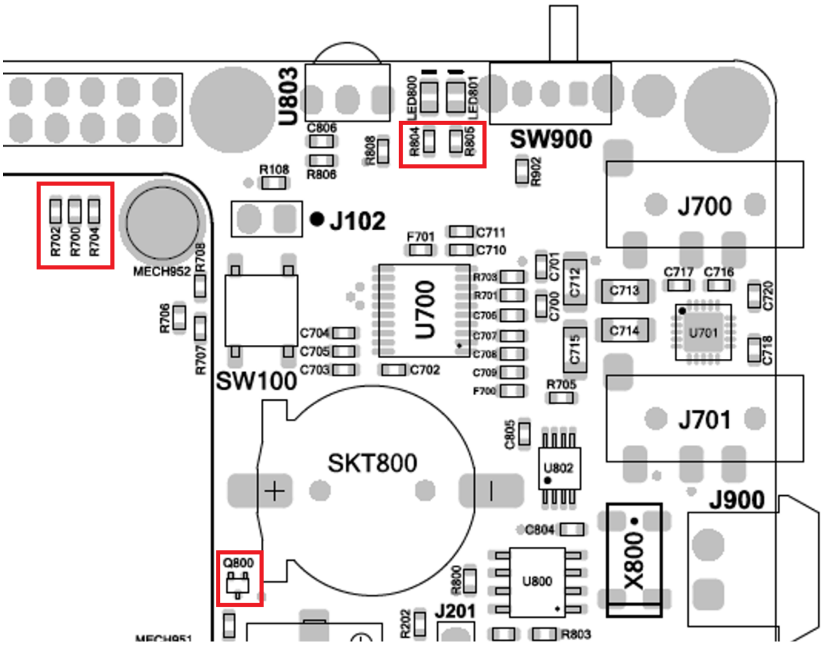


Fig. Assembly drawing with marked components connected to the used GPIO pins

By desoldering R700, R702 and R704 resistors user can free the GPIO pins used for Audio DAC (GPIO18, GPIO19, and GPIO21). In the similar way user can unmount R804 and R805 resistors to unconnect status LEDs (LED800 and LED801) and free the GPIO16 and GPIO26 pins.

By removing Q800 transistor user can free the GPIO22 pin, which is used for buzzer control.

By desoldering the U803 IR receiver user can free the GPIO17 pin.

It seems that this method is not so easy and require special equipment (solder station), so for the next hardware (REV2) will be added dedicated components (copper fuses) to easily disconnect needed GPIO pins.

References:

Chapter 5.1:
https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf

Chapter 2.5: https://datasheets.raspberrypi.org/cm4/cm4-datasheet.pdf

GPIO pinout: https://pinout.xyz/#

13. Known issues

13.1. Issue #1

Description - CC pins

Workaround

13.2. Issue #2

Description – voltage drops

Workaround

13.3. Issue #3

Description – SD card slot location

Workaround

13.4. Issue #4

Description - M.2 max. current

Workaround

14. Troubleshooting

14.1. Issue #1

Description

Workaround

14.2. Issue #2

Description

Workaround

14.3. Issue #3

Description

Workaround

14.4. Issue #4

Description

Workaround

15. Detailed assembly drawings

Here you can find the assembly drawing, where all the carrier board components are marked by using unique designators. You can check where on the PCB board are located specific SMD and THT components.

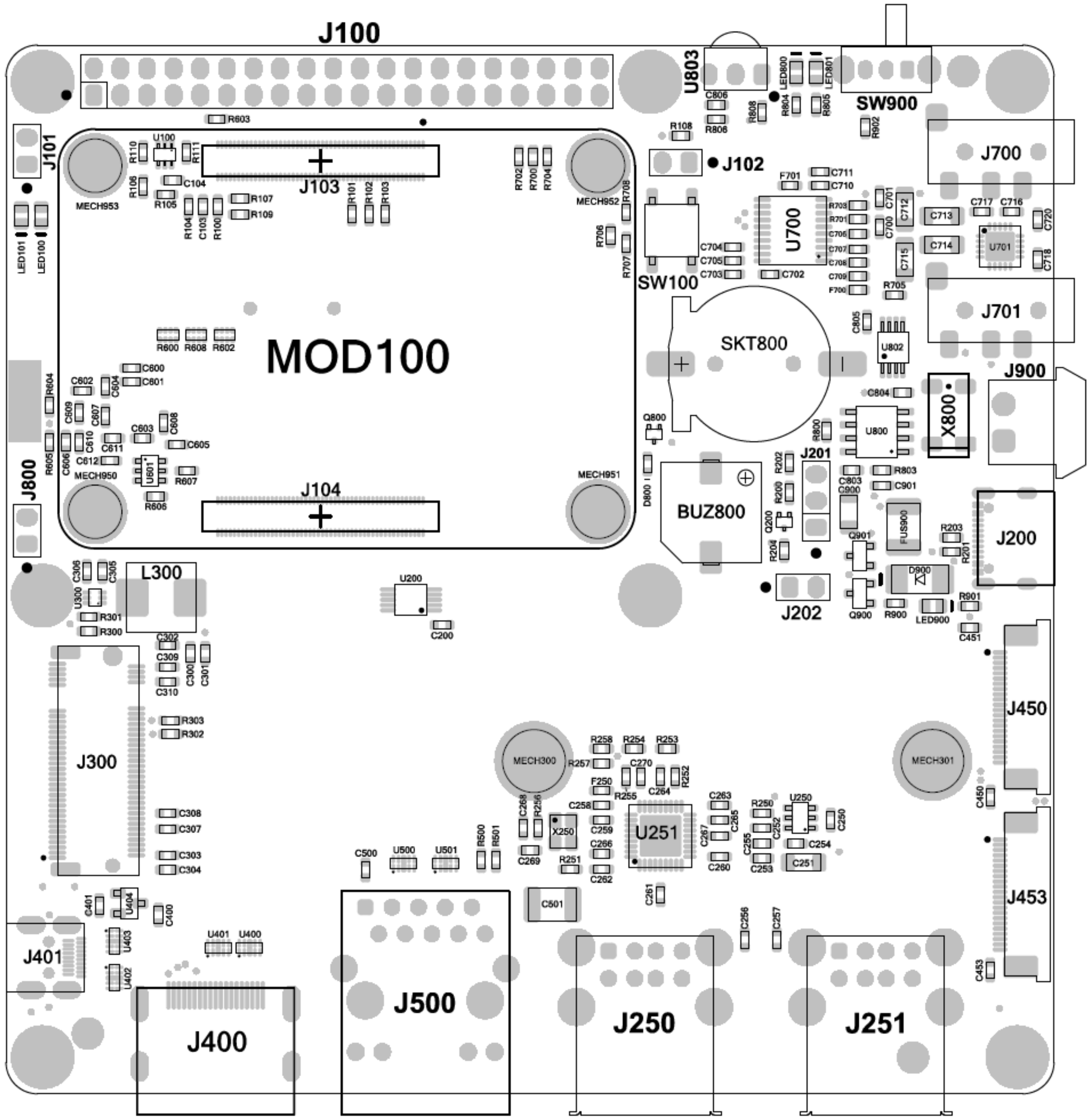


Fig. Assembly drawing – Top view (SMD and THT components)

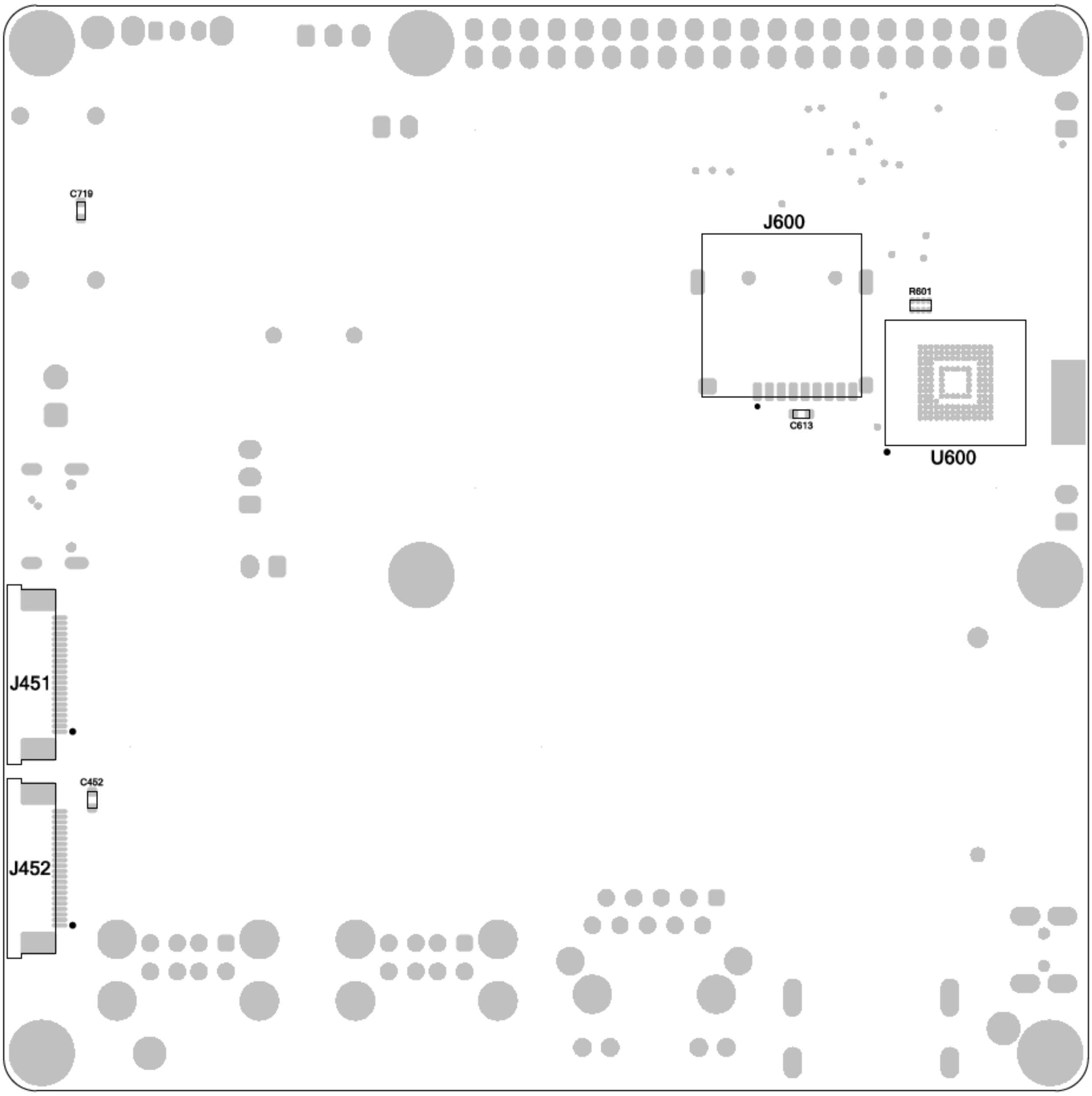


Fig. Assembly drawing – Bottom view (SMD components)

These drawings can be downloaded in the PDF format from:

https://github.com/mfolejewski/MirkoPC/blob/main/MirkoPC_REV1_Assembly_20210327.pdf

16. Mechnical dimensions

PCB outline has 100x100mm. The most important dimensions are shown in the figure below. There are listed the positions of the user interface (connectors).

Connectors such as: HDMI #0, RJ45, USB and audio connectors are placed 2mm from the edge of the board.

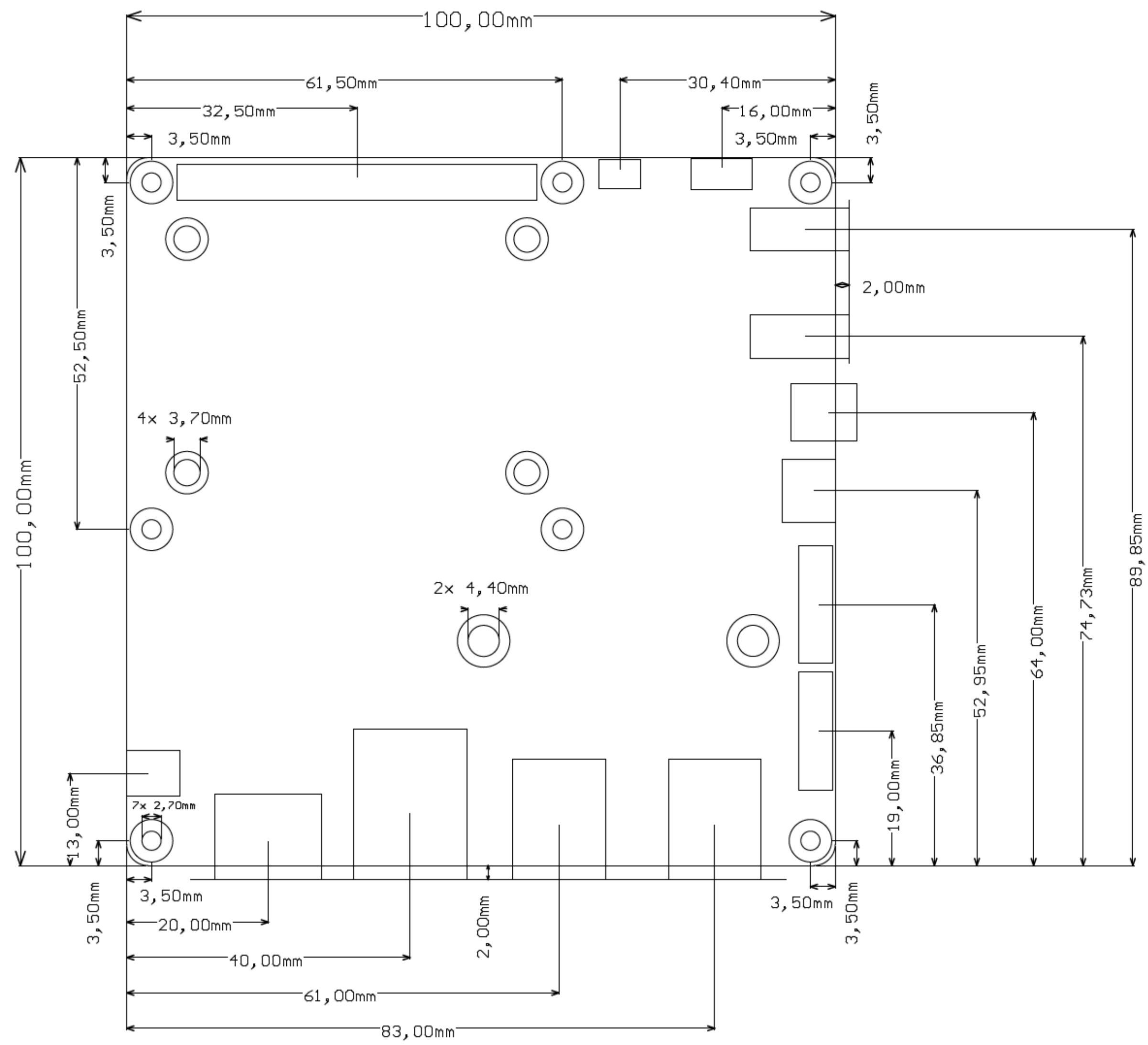


Fig. Mechanical dimensions

This mechnical drawing can be downloaded in the PDF and DXF formats from:

https://github.com/mfolejewski/MirkoPC/blob/main/MirkoPC_REV1_Mechanical_dimensions_20210407.DXF

https://github.com/mfolejewski/MirkoPC/blob/main/MirkoPC_REV1_Mechanical_dimensions_20210406.pdf

17. M.2 tested devices

So far, for slot M.2 the following cards have been successfully tested, as specified in the table below.

M.2 device	Manufacturer	Model	Capacity	Size	Result
SSD NVMe drive	Western Digital (WD)	PC SN520 (SDAPMUW-128G-1101)	128GB	2242 B+M Key	OK
SSD NVMe drive	Kingston Technology	RBU-SNS8154P3/256GJ1	256GB	2280 B+M Key	OK
SSD NVMe drive	Western Digital (WD)	WD_BLACK SN750 (WDS500G3X0C)	500GB	2280 M-Key	OK
ML/AI module	Coral (Google)	Coral M.2 TPU Accelerator	-	2280 B+M Key	Only detected on PCIe bus, not supported (drivers)

Tab. Tested M.2 cards

Due to still is the existing issue with the PCIe drivers, Coral TPU modules are currently not supported by CM4 module. Here you can find more informations:

https://www.raspberrypi.org/forums/viewtopic.php?t=293248

https://www.raspberrypi.org/forums/viewtopic.php?f=98&t=294924

https://github.com/google-coral/edgetpu/issues/280

18. M.2 possible extensions

Due to MirkoPC has implemented standard PCIe x1 interface on the M.2 socket, so it is theoretically possible to connect any devices and cards in the M.2 PCIe standard in 2242 or 2280 form factor (except for NVMe SSDs not tested in practice), such as:

- Google Coral TPU module for AI/ML applications (see MSI-X issue with Raspberry Pi),
- GSM modems (e.g. 3G/4G/LTE),
- FPGA devices.

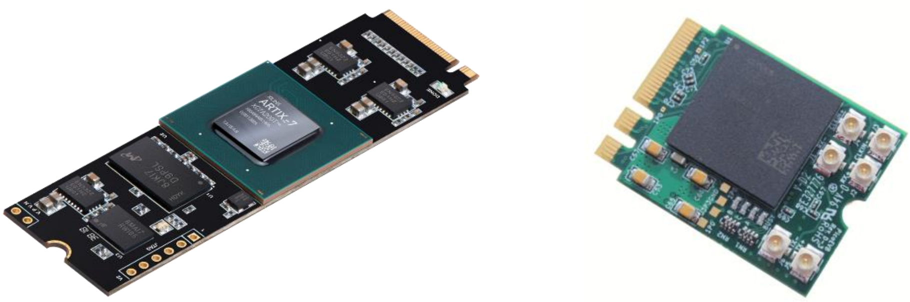


Fig. FPGA devices made as a standard M.2 PCIe cards

19. Future plans (REV2)

During manufacturing first batch (REV1, beta), bring up the hardware and testing by the first users, a few ideas and potential modifications emerged that could be introduced to the second revision of the project (REV2).

There are also some minor issues that need to be fixed or require to modify the existing hardware.

Here is the current list of possible changes:

ID	Type	Subject	Description
#01	Issue	USB-C interface	USB-C interface and Configuration Channel (*CC*) pins (reference: the Raspberry Pi 4B issue with USB-C)
#02	Change	Buzzer	there are only a few simple uses for the buzzer (e.g. warnings), during normal operation audio messages can be generated by using PCM or HDMI interface. This component is quite expensive, and require noticeable space on the PCB (can be replaced by other more useful feature).
#03	Change	Micro HDMI (#1)	For REV1 as HDMI connectors were used HDMI Type A (#0) and MicroHDMI Type D (#1). Second connector has few issues: another location (than other user ports), harder to solder, required cable adapter MicroHDMI -> HDMI. Possible changes: change both MicroHDMI + HDMI to two vertical HDMI connectors (more expensive), or to double stacked HDMI connectors (more expensive), or to two MicroHDMI connectors, or just remove this MicroHDMI port.
#04	Change	Power supply switch	Current power switch uses quite old technique (slide switch) and not currently used in typical mini SBC computers (obsolete feature). Possible idea to redesign that switch: use tactile, angled switch with dedicated controller (push-button controller) or MCU. Benefits and extra features: turn on the device by single click, turn off the device by holding button for a few sec (from safety reasons) and generate interrupt (warning), turn off system 5V rail after RPi OS closed, control a status LED (stand-by mode), turn on the device by using IR remote control, etc.
#05	Change	5VDC input	Consider to change system 5VDC input power rail to 12VDC due to the possible drop voltages and in some cases too low input voltage from external power supply (e.g. from PC laptop -> ~4.9V). Disadvantage to change: dedicated power supply required (12V), not possible to supply by standard USB-C cable or by using typical mobile chargers (5V).
#06	Change	M.2 spacers	Consider to change M.2 SMD spacers from M3 to M2.5 sizes.
#07	Change	MicroSD slot	Consider to change MicroSD slot and place near to the PCB Edge (with possible eMMC chip removing).
#08	Remove	eMMC BGA	Consider to remove extra eMMC BGA chip due the CM4 modules have embedded eMMC memory onboard and for the CM4 lite version MicroSD card can be used.
#09	Change	CM4 spacers	Consider not mounting SMD spacers (by default) for CM4 due to quite hard disassembly process.
#10	Issue	Polymer fuse	Consider to change or to remove 3A polimer fuse, due to too high voltage drop on the system 5V power rail.
#11	Issue	5V power rail	Improve copper track and PCB vias on the 5V power rail by changing track width, add extra vias, unmask copper track (soldermask) to decrease series resistance.
#12	Add	5V power rail	Add extra bypass capacitors.
#13	Change	3V3 buck (M.2)	Consider to change buck regulator for 3.3V @3A power rail for M.2 socket with improved thermal managment (QFN/DFN package and high-efficiency buck regulator, e.g. > 95%).
#14	Add	GPIO config	Add configuration resistors (0603) on the uses GPIO lines to disconnect these line ondemand (for GPIO lines used for PCM interface, LED status, buzzer and IR sensor).
#15	Add	Fan controller	Add optional and dedicated fan controller 5V or 12V (e.g. similar as EMC2301 on the CM4-IO board).
#16	Remove	BOOT jumper	Consider to remove not used BOOT MODE jumper (J202).
#17	Remove	RESET switch	Consider to remove RESET switch (SW100). During OS running and from safety reasons RUN_PG signal is blocked, so the RESET switch is not used and can be removed.

#18	Remove	RUN_PG jumper	Consider to remove RUN_PG jumper (J102) – not often used feature.
#19	Add	Status LED pipe	Consider to add light pipe for status LEDs.
#20	Add	SIM slot	Consider to add Nano SIM slot to the M.2 socket (for support GSM cards).
#21	Change	Mechnical	Consider to change connectors placement near to the PCB edge regarding mechanical design of the enclosure (3mm instead of 2mm?).
#22	Add	ICT Testpoints	On the bottom side add ICT testpoints for mass production and testing process.
#23	Add	Assembly variants	Consider to add 3 different assembly variants: Lite (without audio/M.2/only 2xUSB), Standard (without M.2/headphone amp) and Pro (all features assembled)
#24	Change	PCB Marking	Change PCB marking (silkscreen) – remove „mirko electronics” brand, add MirkoPC icon.
#25	Issue	HDMI voltage pull back	When the device is off (power switch in right position), user can observe that 5V LED (LED900) glows dimly. Some voltage from the HDMI oputput can pull back to the carrier board. It does not cause any problems or damage except the glowing LED diode. Consider to protect the board from that issue.
#26			
#27			

20. Project info

More information, technical details and the current status about the project you can find on the internet.

Below are listed most important sources.

Hackaday.io (project description):

https://hackaday.io/project/177626-mirkopc-cm4-carrier-board

Twitter account (current project status):

https://twitter.com/Mirek34416541

Github (technical details):

https://github.com/mfolejewski/MirkoPC

21. Help & support

If you have any questions, are looking for support, or are troubleshooting problems while using that motherboard, here are some helpful links and support channels.

Contact with project Author (DM):

Twitter -> https://twitter.com/Mirek34416541

Hackaday.io -> https://hackaday.io/adsp531

email -> mfolejewski@gazeta.pl

Raspberry Pi Compute Module datasheet:

https://datasheets.raspberrypi.org/cm4/cm4-datasheet.pdf

Raspberry Pi CM4-IO board datasheet:

https://datasheets.raspberrypi.org/cm4io/cm4io-datasheet.pdf

Raspberry Pi 4B datasheet:

https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf

Raspberry Pi forum (all categories):

https://www.raspberrypi.org/forums/

Raspberry Pi forum -> Compute Modules:

https://www.raspberrypi.org/forums/viewforum.php?f=98&sid=26785cbc106ae43fbe012ab00fb874c5

The Raspberry Pi Foundation webpage:

https://www.raspberrypi.org/documentation/

https://www.raspberrypi.org/help/

Jegg Geerling blog (Raspberry Pi OS configuration):

https://www.jeffgeerling.com/blog

22. News media and press releases

You can find more information about MirkoPC project on the internet. So far several press releases in the media about the project have been published. The most important are listed below.

Until now, most of this media news about the project was published thanks to the well-known IT enthusiast Jeff Geering by a demonstration of the new feature of booting a Raspberry Pi CM4 module from an NVMe disk based on the MirkoPC board.

Jeff Geerling blog and Youtube channel:

https://pipci.jeffgeerling.com/boards_cm/mirkopc.html

https://www.youtube.com/watch?v=4Womn10v71s

https://www.jeffgeerling.com/blog/2021/raspberry-pi-can-boot-nvme-ssds-now

https://github.com/geerlingguy/raspberry-pi-pcie-devices/issues/101

LinuxGizmos.com:

http://linuxgizmos.com/open-spec-piunora-and-mirkopc-carriers-for-rpi-cm4-feature-m-2-for-nvme/

Tom's Hardware:

https://www.tomshardware.com/news/raspberry-pi-compute-module-4-nvme

Hackaday:

https://hackaday.com/2021/03/29/nvme-boot-finally-comes-to-the-pi-compute-module-4/

Hackster.io:

https://www.hackster.io/news/this-lets-you-boot-compute-module-4-with-nvme-sdds-b1ff4c39a4eb

Raspberry Pi Projects:

https://raspberrypiprojects.com/raspberry-pi-boots-off-an-nvme-ssd-natively/

23. Document revision history

Date	Revision	Changes
2021-04-09	Draft	Initial revision (beta)
2021-04-14	1.0	First release
2021-04-15	1.1	Described: - chapter: CM4 image flashing, - chapter: First run, Changed: User interface chapter,
	1.2	
	1.3	

24. Trademarks

Raspberry Pi and the Raspberry Pi logo and other Raspberry Pi brands are trademarks of Raspberry Pi Foundation. For additional information about Raspberry Pi trademarks, please refer to:

https://www.raspberrypi.org/trademark-rules/

or if you have any further questions, please contact with:

trademarks@raspberrypi.org

All product names, logos, brands, trademarks and registered trademarks are property of their respective owners. All company, product and service names used in this document are for identification purposes only. Use of these names, trademarks and brands does not imply endorsement.