

# CM5-DUAL-ETH-BASE

From Waveshare Wiki

Jump to: navigation, search

## Overview

### Introduction

As the IO board for the Raspberry Pi Compute Module 5, CM5-DUAL-ETH-BASE is a baseboard that can be used with the Raspberry Pi Compute Module 5. It features powerful functionality with onboard 1 USB3.2 Gen1, 2 USB2.0, M.2 M KEY, and dual Ethernet ports for convenient use.

### Features and Precautions

1. Do not plug or unplug any device other than USB and HDMI when the device is powered on
2. Type-C can be used as a USB SLAVE interface to flash images
3. Onboard default 1 USB 3.2 Gen1 port
4. Supports 4 screens to display different contents, adds MIPI DSI driver, regardless of whether a screen is connected, the system will default to connect, the screen will display in split-screen mode (related to the system version)
5. Onboard M.2 M KEY interface, supports NVMe boot
6. The onboard BOOT button, you can press the BOOT button before powering on, and connect to the computer through Type-C to let the device enter the flashing mode
7. Do not connect other devices when using Type-C for flashing, otherwise insufficient

#### CM5-DUAL-ETH-BASE



(<https://www.waveshare.com/cm5-dual-eth-base.htm>)

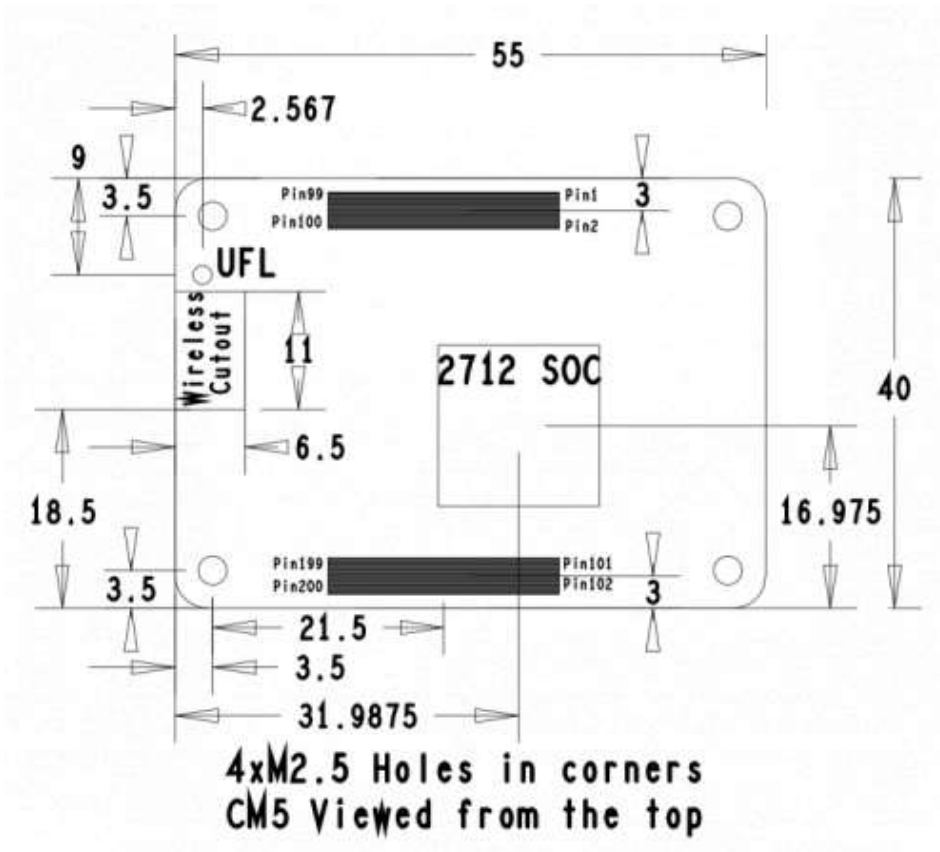
MIPI, 4G/5G, HDMI, USB2.0/3.0, I2C, I2S, SPI, UART, RPi, Ethernet

power supply will cause the device to be unrecognized

8. The USB2.0 ports have a total maximum power output of 2A, and the USB3.0 port has a maximum power output of 2A

## Dimensions

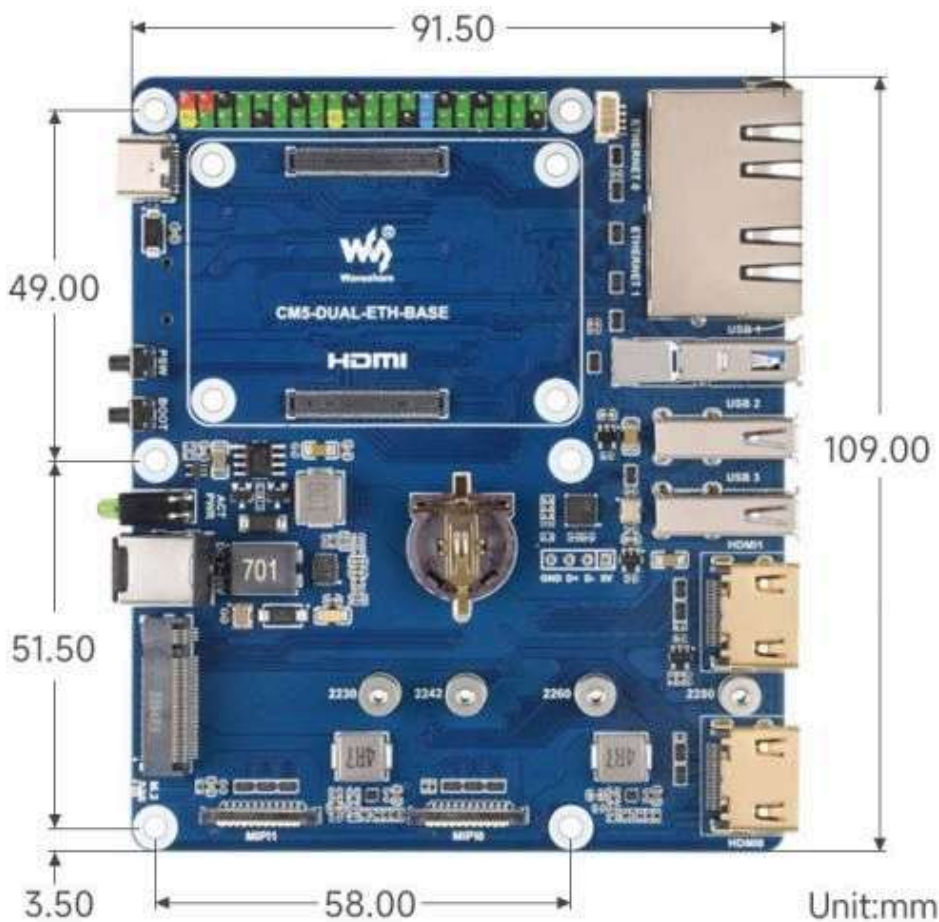
### Core Board Size



(/wiki/File:750px-

Compute-Module-5-1-1.png)

## Baseboard Size



(/wiki/File:CM5-DUAL-

ETH-BASE-details-size.jpg)

## Case Size

Stay tuned...

## Image Flashing

### How to Flash

Enter flashing mode

1. The onboard BOOT button, you can press the BOOT button before powering on, and connect to the computer through Type-C to let the device enter the flashing mode

Note that when connecting to the DC port, Type-C will output 5V voltage. Do not connect the DC power supply and connect the computer host via Type-C at the same time

Click here ([https://www.waveshare.com/wiki/Compute\\_Module\\_Burn\\_EMMC](https://www.waveshare.com/wiki/Compute_Module_Burn_EMMC)) for EMMC version

Click here ([https://www.waveshare.com/wiki/CM4\\_Burn\\_SD](https://www.waveshare.com/wiki/CM4_Burn_SD)) for LITE version

## NVME

### 1. Enable PCIE interface

```
PI5B defaults to not having the PCIE interface enabled. Add to /boot/firmware/config.txt:  
dtparam=pciex1
```

2. PCIE is gen2 by default, if you need to enable PCIE gen3, then add following to /boot/firmware/config.txt:

```
dtparam=pciex1_gen=3
```

3. After the modification, reboot the PI5, and the device can be recognized.

As shown in the figure below, SM2263 is identified as my SSD solid state drives, and the other PI5 is the RPI chip

```
pi@raspberrypi:~$ lspci  
0000:00:00.0 PCI bridge: Broadcom Inc. and subsidiaries Device 2712 (rev 21)  
0000:01:00.0 Non-Volatile memory controller: Silicon Motion, Inc. SM2263EN/SM2263XT SSD Controller (rev 03)  
0001:00:00.0 PCI bridge: Broadcom Inc. and subsidiaries Device 2712 (rev 21)  
0001:01:00.0 Ethernet controller: Device 1de4:0001
```

(/wiki/File:PCIE\_TO\_M.2\_HAT%2B\_W\_1.png)

4. Partition, skip this step if you have partitioned and formatted on other platforms (Note: Partitioning will delete all data on the SSD, please proceed with caution)

Lsblk This command is executed to view the disk (if you want to see the details, run the sudo fdisk -l command)

```
pi@raspberrypi:~ $ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
mmcblk0     179:0    0  29.7G  0 disk
├─mmcblk0p1 179:1    0   512M  0 part /boot/firmware
└─mmcblk0p2 179:2    0   29.2G  0 part /
nvme0n1     259:0    0 119.2G  0 disk
└─nvme0n1p1 259:1    0 119.2G  0 part
```

(/wiki/File:PCIe\_TO\_M.2\_HAT%2

B\_W\_2.png)

Partition:

sudo fdisk /dev/nvme0n1 The device number is the total device number, do not add p1, that is just a partition

How to use the partitioning tool fdisk:

n New partition

q Quit without saving

p Print the partition table

m Print the selection menu

D Delete the partition

w Save and exit

t Modify the ID number

Add the partition and execute n, then save and exit with w

## 5. Format

sudo mkfs. Execute the command and press Tab key, you will see a lot of different suffixes, and the different suffixes are the formats you need to format

```
pi@raspberrypi:~ $ sudo mkfs.
mkfs.bfs mkfs.cramfs mkfs.exfat mkfs.ext2 mkfs.ext3 mkfs.ext4 mkfs.fat mkfs.minix mkfs.msdos mkfs.ntfs mkfs.vfat
```

(/wiki/File:PCIe\_TO\_M.2\_HAT%2B\_W\_3.png)

If I want to format it in ext4 file format, then execute the command:

```
sudo mkfs.ext4 /dev/nvme0n1p1
```

Wait a moment, once all "done" appear as below, it indicates that the formatting is completed

```
pi@raspberrypi:~ $ sudo mkfs.ext4 /dev/nvme0n1p1
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 31258368 4k blocks and 7815168 inodes
Filesystem UUID: 1a84fb29-5460-475f-afb7-0a90271ef975
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done
```

(/wiki/Fi

le:PCIe\_TO\_M.2\_HAT%2B\_W\_4.png)

## 6. Mount

```
Create a mount directory
sudo mkdir toshiba
Mount the device
sudo mount /dev/nvme0n1p1 ./toshiba
Check the disk status
df -h
```

## Read/Write Test

---

Enter the directory where the disk is mounted

```
cd toshiba
```

- Free up the memory

```
sudo sh -c "sync && echo 3 > /proc/sys/vm/drop_caches"
```

- Copy Raspberry Pi memory content to the hard disk (write)

```
sudo dd if=/dev/zero of=./test_write count=2000 bs=1024k
```

```
pi@raspberrypi:~/toshiba $ sudo dd if=/dev/zero of=./test_write count=2000 bs=1024k
2000+0 records in
2000+0 records out
2097152000 bytes (2.1 GB, 2.0 GiB) copied, 3.78947 s, 553 MB/s
```

(/wiki/File:PCIe\_TO\_M.2\_HAT%2B\_W\_5.png)

- Copy the hard drive content to the Raspberry Pi memory (/etc/fstab read)

```
sudo dd if=./test_write of=/dev/null count=2000 bs=1024k
```

```
pi@raspberrypi:~/toshiba $ dd if=./test_write of=/dev/null count=2000 bs=1024k
2000+0 records in
2000+0 records out
2097152000 bytes (2.1 GB, 2.0 GiB) copied, 3.53634 s, 593 MB/s
```

(/wiki/File:Pcie-m2-6new.png)

- Note: The test results vary for different cards and environments. The Raspberry Pi is significantly affected. If you want to test accurate performance, use a PC for the test

## Auto Mount

---

Test shows there's no issue. If it's not required to be used as a system disk, but only for expanding the disk, set it to auto-mount

```
sudo nano /etc/fstab

#Add at the end
/dev/nvme0n1p1 /home/pi/toshiba ext4 defaults 0 0
#/dev/nvme0n1p1 is the device name, /home/pi/toshiba refers to mounting to a directory, e
xt4 is the file system type, defaults uses the default mount option
#Make the changes take effect (reboot only after testing, otherwise it will fail to mount
and boot)
sudo mount -a

#Then reboot
Check the device with lsblk
```

## NVMe SSD Boot

Boot the Raspberry Pi with a TF card first, mount and test it, and make sure the hardware can work properly

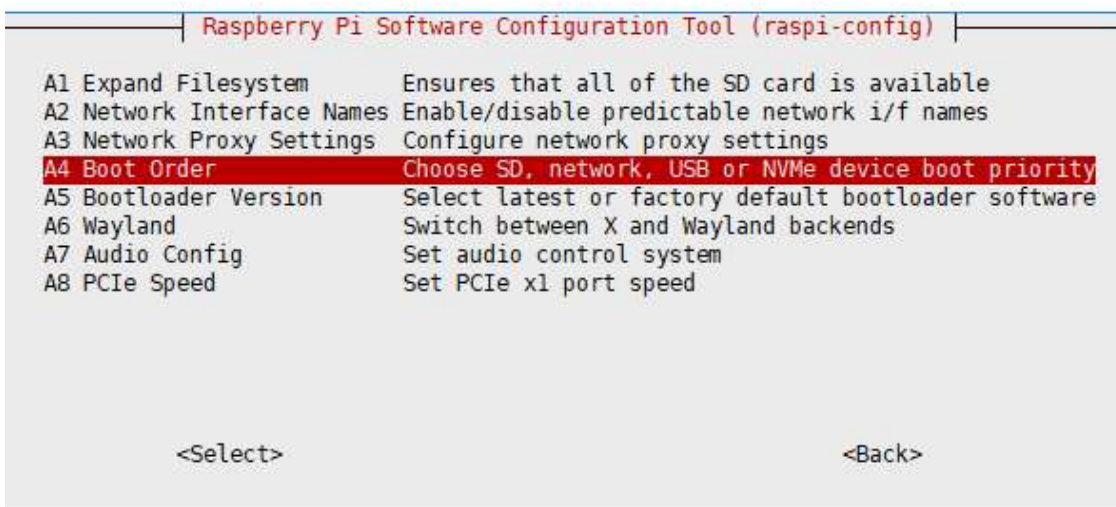
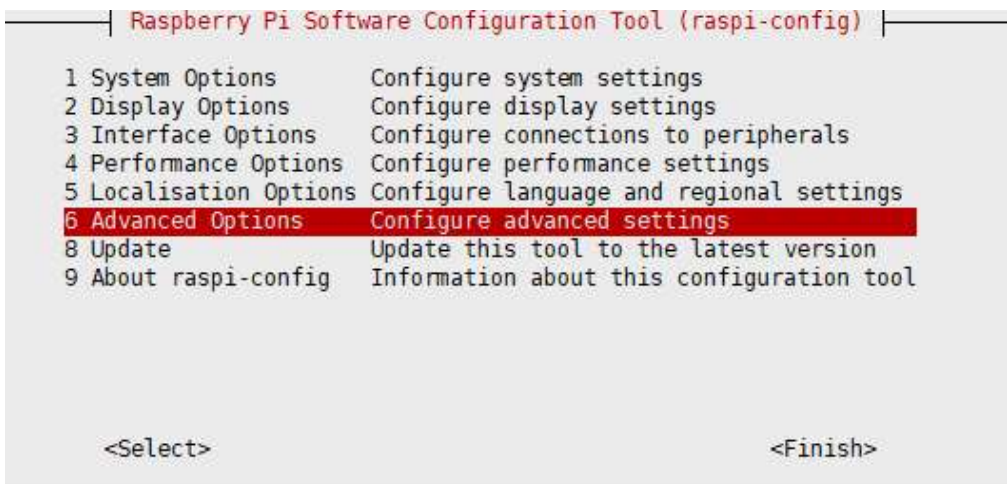
Choose one of the following methods

### Method 1

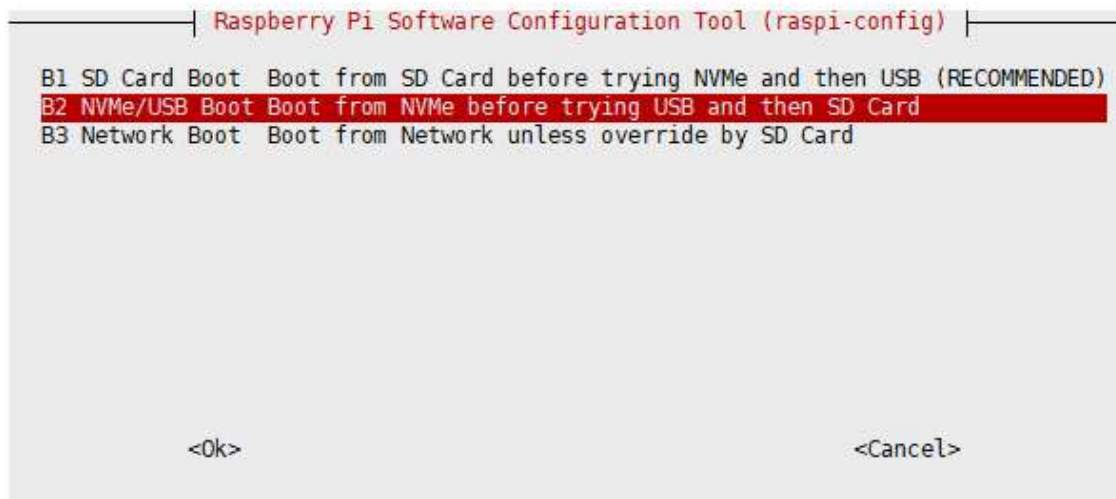
---

1. Run the following command:

```
sudo raspi-config
```



(/wiki/Fil



e:PCIE\_NVME\_BOOT-1.png)

## 2. Reboot Raspberry Pi

If you find you can't modify it multiple times, please reconnect to the network and then try to modify it (wait for the network to synchronize the time automatically), or modify the file after setting the correct time

- Flash the system to NVME, then connect the NVME to the expansion board, remove the TF card and power it on again

## Method 2

- Modify the BOOT\_ORDER in the Raspberry Pi boot loader configuration:

```
sudo rpi-eeeprom-config --edit
```

```
Modify BOOT_ORDER=0xf41 to BOOT_ORDER=0xf416
```

Value	Mode	Description
0x0	SD CARD DETECT	Try SD then wait for card-detect to indicate that the card has changed - deprecated now that 0xf (RESTART) is available.
0x1	SD CARD	SD card (or eMMC on Compute Module 4).
0x2	NETWORK	Network boot - See <a href="#">Network boot server tutorial</a>
0x3	RPIBOOT	RPIBOOT - See <a href="#">usbboot</a>
0x4	USB-MSD	USB mass storage boot - See <a href="#">USB mass storage boot</a>
0x5	BCM-USB-MSD	USB 2.0 boot from USB Type C socket (CM4: USB type A socket on CM4IO board). Not available on Raspberry Pi 5.
0x6	NVME	CM4 and Pi 5 only: boot from an NVMe SSD connected to the PCIe interface. See <a href="#">NVMe boot</a> for more details.
0x7	HTTP	HTTP boot over ethernet. See <a href="#">HTTP boot</a> for more details.
0xe	STOP	Stop and display error pattern. A power cycle is required to exit this state.
0xf	RESTART	Restart from the first boot-mode in the BOOT_ORDER field i.e. loop

(/wiki/File:PCIe\_TO\_M.2\_HAT%2B\_W\_6.png)

For more information, please refer to BOOT\_ORDER (<https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#raspberrypi-bootloader-configuration>)

- Reboot Raspberry Pi

If you find you can't modify it multiple times, please reconnect to the network and then try to modify it (wait for the network to synchronize the time automatically), or modify the file after setting the correct time

3. Flash the system to NVME, then connect the NVME to the expansion board, remove the TF card and power it on again

## Fan Control

The PI5 fan defaults to starting when the temperature reaches 50 degrees. If you want to control it at other temperatures, you can do so by adding specific content to `/boot/firmware/config.txt`, for example:

```
dtparam=fan_temp0=36000,fan_temp0_hyst=2000,fan_temp0_speed=90
dtparam=fan_temp1=40000,fan_temp1_hyst=3000,fan_temp1_speed=150
dtparam=fan_temp2=52000,fan_temp2_hyst=4000,fan_temp2_speed=200
dtparam=fan_temp3=58000,fan_temp3_hyst=5000,fan_temp3_speed=255
```

Among which:

<code>fan_temp0/1/2/3</code>	Indicates the temperature (36000 means 36°C)
<code>fan_temp0/1/2/3_speed</code>	Indicates the corresponding rotational speed (value up to 255)
<code>fan_temp0/1/2/3_hyst</code>	Indicates the hysteresis temperature

For more details, please click here (<https://github.com/raspberrypi/linux/blob/7af85d54e39733bb9a236b95ea5ed1ab8277d560/arch/arm/boot/dts/overlays/README#L265>)

Notice: There are only 4 temperatures: 0, 1, 2 and 3, and it's not possible to set other temperatures. The hysteresis temperature must not exceed the step range between two temperatures

## MIPI

Support dual MIPI, customers can freely choose CSI or DSI connections

The DSI screen is 800x480 resolution screen by default, please refer to the corresponding WIKI for other resolution screens

Note: A single camera can only connect to MIPI0

## DSI

```
#Add the following to the config.txt file:
sudo nano /boot/firmware/config.txt
# DSI0
dtoverlay=vc4-kms-dsi-7inch,dsi0
# DSI1
dtoverlay=vc4-kms-dsi-7inch,dsi1
```

## CSI

Add the following to the config.txt file:

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

Note: A single camera can only connect to MIPI0

Mode	CAM0 setup statement	CAM1 setup statement
OV9281	dtoverlay=ov9281,cam0	dtoverlay=ov9281,cam1
IMX290/IMX327	dtoverlay=imx290,clock-frequency=37125000,cam0	dtoverlay=imx290,clock-frequency=37125000,cam1
IMX378	dtoverlay=imx378,cam0	dtoverlay=imx378,cam1
IMX219	dtoverlay=imx219,cam0	dtoverlay=imx219,cam1
IMX477	dtoverlay=imx477,cam0	dtoverlay=imx477,cam1
IMX296	dtoverlay=imx296,cam0	dtoverlay=imx296,cam1
IMX708	dtoverlay=imx708,cam0	dtoverlay=imx708,cam1

Allow one connection to DSI and one connection to CSI, for example, use IMX219 to connect to MIPI0

For example, if you want to connect a DSI display to MIPI1, add the following to the config.txt file

```
dtoverlay=imx219,cam0
dtoverlay=vc4-kms-dsi-7inch,dsi1
```

Use the shortcut Ctrl+o to save the file, then press Enter



(/wiki/File:CM4-

NANO-B-CSI-03.png)

Use the shortcut Ctrl+x to exit the editor

Reboot the device

```
sudo reboot
```

## Test Camera

1. Enter the camera detection command, and you can see that the camera has been detected

```
rpicam-hello --list-cameras
```

```

pi@CM4-NANO-B:~$ libcamera-hello --list-cameras
[0:00:59.527001537] [1165] INFO Camera camera_manager.cpp:293 libcamera v0.0.0+3424-e68e0f1e
[0:00:59.581096979] [1172] WARN RPI raspberrypi.cpp:1202 Mismatch between Unicam and CamHelper for e
mbedded data usage!
[0:00:59.581715900] [1172] INFO RPI raspberrypi.cpp:1317 Registered camera /base/soc/i2c0mux/i2c@0/i
mx219@10 to Unicam device /dev/media3 and ISP device /dev/media0
Available cameras
-----
0 : imx219 [3280x2464] (/base/soc/i2c0mux/i2c@0/imx219@10)
  Modes: 'SRGGB10_CSI2P' : 640x480 1640x1232 1920x1080 3280x2464
        'SRGGB8' : 640x480 1640x1232 1920x1080 3280x2464

```

(/wiki/File:CM4-NANO-B-CSI-01.png)

## 2. Display the camera screen to the desktop

```
rpicam-hello -t 0
```

## 3. Take a photo

```
rpicam-jpeg -o test.jpg
```

## 4. Record a 10s video

```
rpicam-vid -t 10s -o test.h264
```

## Other commands

### Check if the camera is detected

```
rpicam-hello --list-cameras
```

### Open the corresponding camera

```
rpicam-hello --list-cameras 0
rpicam-hello --list-cameras 1
```

### Take a photo

```
rpicam-jpeg -o test.jpg
#You can add --camera to specify the camera
```

## Real-Time Clock (RTC)

There is no battery by default, and an additional RTC battery is required

### Software Debugging

The default device is /dev/rtc0

Regarding time, by inputting "date" in the command line, you can see the current time. Connecting the Raspberry Pi system to the network will automatically synchronize the time. If the default RTC device is connected and functioning properly, the RTC time will be updated after the automatic network synchronization

`sudo hwclock -r` Read the RTC time, if there are multiple RTC devices, you can use `-f` to select the corresponding devices (for example: `sudo hwclock -f /dev/rtc1 -r`)

## Hwclock

System clock -> Hardware clock (RTC)

```
sudo hwclock -w
```

Synchronize hardware clock (RTC) -> System clock

```
sudo hwclock -s  
#Need to turn off the network, or disable network time synchronization, otherwise it will  
be changed back
```

Set the hardware clock time (RTC):

```
sudo hwclock --set --date="9/8/2023 16:45:05"
```

View the hardware clock (RTC)

```
sudo hwclock -r
```

Show version information.

```
sudo hwclock --verbose
```

## Automated Wakeup

To support a low-power mode for wake-up alarms, add the configuration:

```
sudo -E rpi-eeeprom-config --edit
#Add the following 2 lines
POWER_OFF_ON_HALT=1
WAKE_ON_GPIO=0
#Restart the device after adding (if you connect to the serial port Log, you can see that t
here are update related logs)
sudo reboot
#You can use the following methods to test the function:
echo +600 | sudo tee /sys/class/rtc/rtc0/wakealarm
sudo halt or sudo poweroff
#10 minutes later, it will be awakened and restarted
```

## RTC Battery Charging

Note: Before adding this, make sure your RTC battery allows charging and check the maximum allowable voltage

```
sudo nano /boot/firmware/config.txt
#Add
dtparam=rtc_bbat_vchg=3000000
#Among these, 3,000,000 represents the maximum voltage. Charging will stop when it reaches
3V, and the charging will restart with a trickle charge when the voltage drops below 3V
```

## USB

There is one USB3.2 Gen1 port onboard, which can reach a rate of 5Gbps and a current limit of 2A

There are 2 USB2.0 Gen1 ports onboard which share a rate of 480Mbps, and the total current is limited to 2A

The above two current limits are independent

## Type-C

---

For flashing

Note that when connecting to the DC port, Type-C will output 5V voltage. Do not connect the DC power supply and connect the computer host via Type-C at the same time

The default is HOST mode when booting normally

## Encryption Chip (ATSHA204)

I2C address: 0x64

I2C bus: I2C1 (default GPIO2,3)

Onboard encryption chip, not enabled by default

For usage details, please refer to the datasheet and official library:

<https://github.com/MicrochipTech/cryptoauthlib>

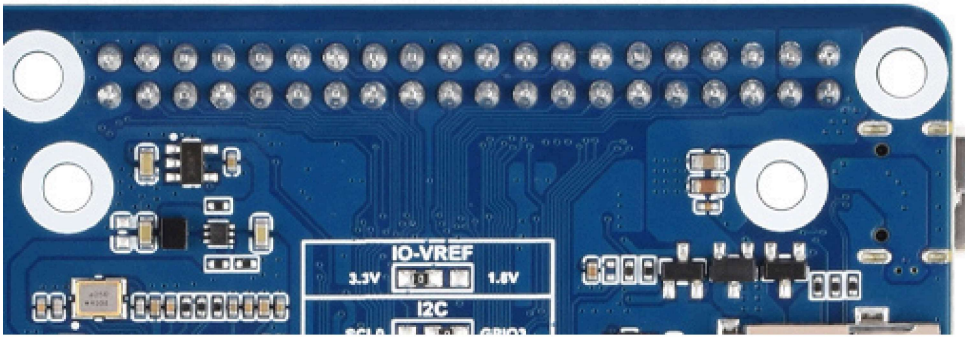
(<https://github.com/MicrochipTech/cryptoauthlib>)

## EEPROM (CAT24C32)

I2C address: 0x50 (default)

I2C bus: I2C1 (default GPIO2,3)

The I2C address can be switched through resistors



(/wiki/File:CM5-

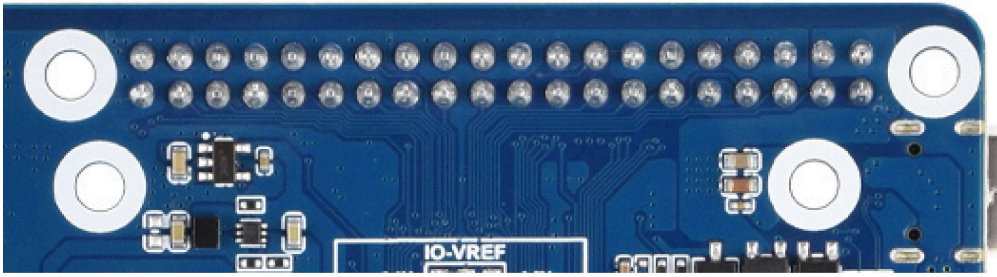
DUAL-ETH-BASE-EEPROM-1.png)

## I2C Bus Switching

---

The I2C bus can be switched through resistors (the switch will switch the EEPROM and the encryption chip together)

I2C1(GPIO2,3) (default)



(/wiki/File:CM5-DUAL-

ETH-BASE-I2C-1.png)

## Resources

### Datasheets and Schematic Diagram

---

CM5 Core board datasheet (<https://datasheets.raspberrypi.com/cm5/cm5-datasheet.pdf>)

CM5 Core board 3D (<https://pip.raspberrypi.com/categories/1096-design-files>)

CM5 Core introduction (<https://datasheets.raspberrypi.com/cm5/cm5-product-brief.pdf>)

CM5-DUAL-ETH-BASE Schematic diagram (<https://files.waveshare.com/wiki/CM5-DUAL-ETH-BASE/CM5-DUAL-ETH-BASE-Sch.pdf>)

CM5-DUAL-ETH-BASE-3D (<https://files.waveshare.com/wiki/CM5-DUAL-ETH-BASE/CM5-DUAL-ETH-BASE-3D.zip>)

### Related Links

---

## FAQ

**Question: Can the old versions of the Raspberry Pi operating system be**

## used with CM5?

### Answer:

CM5 requires the latest version of Raspberry Pi OS Bookworm (2024-11-19 or later).

# Support

## Technical Support

If you need technical support or have any feedback/review, please click the **Submit Now** button to submit a ticket, Our support team will check and reply to you within 1 to 2 working days. Please be patient as we make every effort to help you to resolve the issue.

Working Time: 9 AM - 6 PM GMT+8  
(Monday to Friday)

Submit Now (<https://service.waveshare.com/>)

*Retrieved from "<https://www.waveshare.com/w/index.php?title=CM5-DUAL-ETH-BASE&oldid=105468>  
(<https://www.waveshare.com/w/index.php?title=CM5-DUAL-ETH-BASE&oldid=105468>)"*

---