Software Setup

1. Enable PCIe interface:

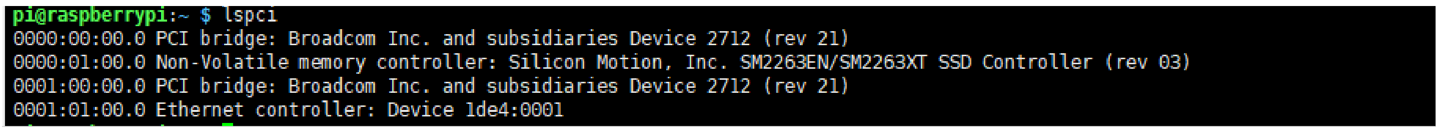
Connect the hardware and the PCIE interface will automatically open as the latest system detects the hardware. If it does not open, you can execute: add "dtparam=pciex1" in the /boot/firmware/config.txt

2. PCIE selects GEN2 mode by default. If PCIE gen3 is needed, you can add the following content in /boot/firmware/config.txt:

```
dtparam=pciex1_gen=3
```

3. Reboot the Pi5 after modifying, and then the device can be recognized.

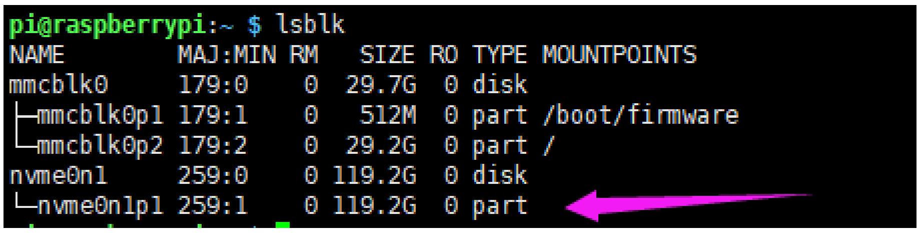
As shown below, SM2263 is the recognized solid-state drive that I use, and the other one is the RPI chip for PI5:



```
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
```

4. Partitioning: If partitioning and formatting have already been performed on another platform, skip this step. Caution: Partitioning and formatting will erase all data on the SSD, so proceed with caution.

lsblk #see the disk (execute "sudo fdisk -l" for more details)



```
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
```

Partition

sudo fdisk /dev/nvme0n1   #dev is the total device number, do not add "p1", just one partition

How do use fdisk

n New partition

q Exit without saving

p Print partition table

m Print selection menu

d Delete partition

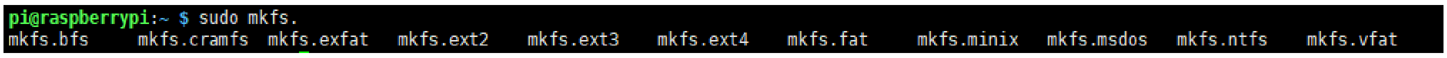
w Save and exit

t Modify ID

Add the partition and execute "n", and then press "w" to save and exit.

5. Format:

sudo mkfs. #Then, pressing Tab will display various file extensions. Each extension corresponds to a format you may want to format the drive into

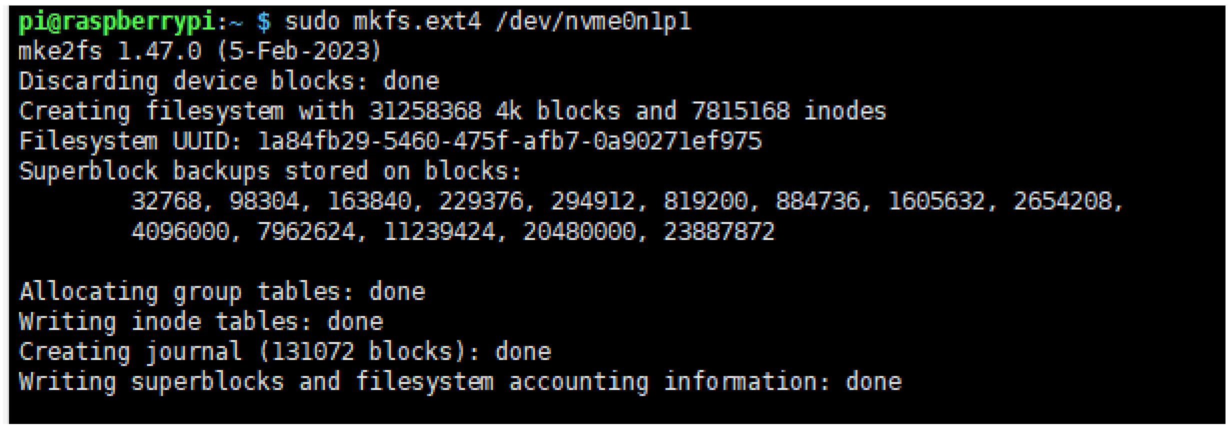


```
pi@raspberrypi:~ $ sudo mkfs.
mkfs.bfs     mkfs.cramfs  mkfs.exfat   mkfs.ext2    mkfs.ext3    mkfs.ext4    mkfs.fat     mkfs.minix   mkfs.msdos   mkfs.ntfs    mkfs.vfat
```

If I need to format it in "ext4" format, execute:

sudo mkfs.ext4 /dev/nvme0n1p1

Wait for a moment, when "done" appears for all, it means the formatting is complete.



```
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
```

6. Mount:

Create the mounting directory:

sudo mkdir toshiba

Mount the device

sudo mount /dev/nvme0n1p1 ./toshiba

Check disk status

df -h

Read/Write Test

Enter the directory to mount the disk:

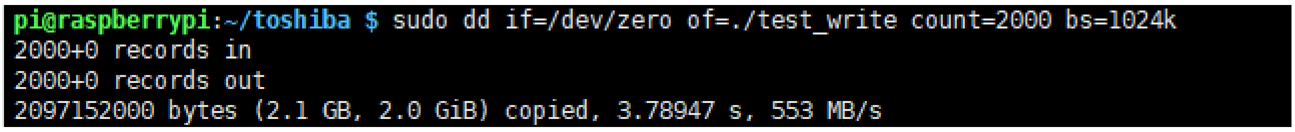
cd toshiba

- Release the caches:

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

- Copy the Raspberry Pi memory to the hard flash driver (write).

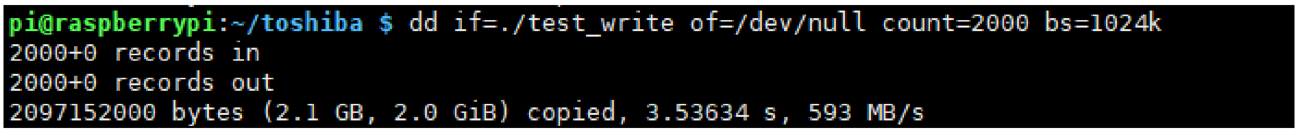
```
sudo dd if=/dev/zero of=./test_write count=2000 bs=1024k
```



```
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
```

- Copy the contents of the hard drive to the Raspberry Pi's memory (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
```

- Please note that using this product with different cards in different environments, the results may vary. When working with the Raspberry Pi, its performance will be significantly affected.

Auto Mount

If there are no issues with the test and the disk is not needed as a system disk, only for expanding disk usage, set up automatic mounting.

```
sudo nano /etc/fstab


#Add the following content at the end:

/dev/nvme0n1p1  /home/pi/toshiba  ext4  defaults  0  0

#/dev/nvme0n1p1 device name, /home/pi/toshiba mount to the directory, ext4 is the file system type, defaults means using the default mounting options

#Reboot to take effect (Please make sure there are no issues before rebooting, otherwise it can not be booted without mounting)

sudo mount -a


#And then reboot

Check the device through lsblk
```

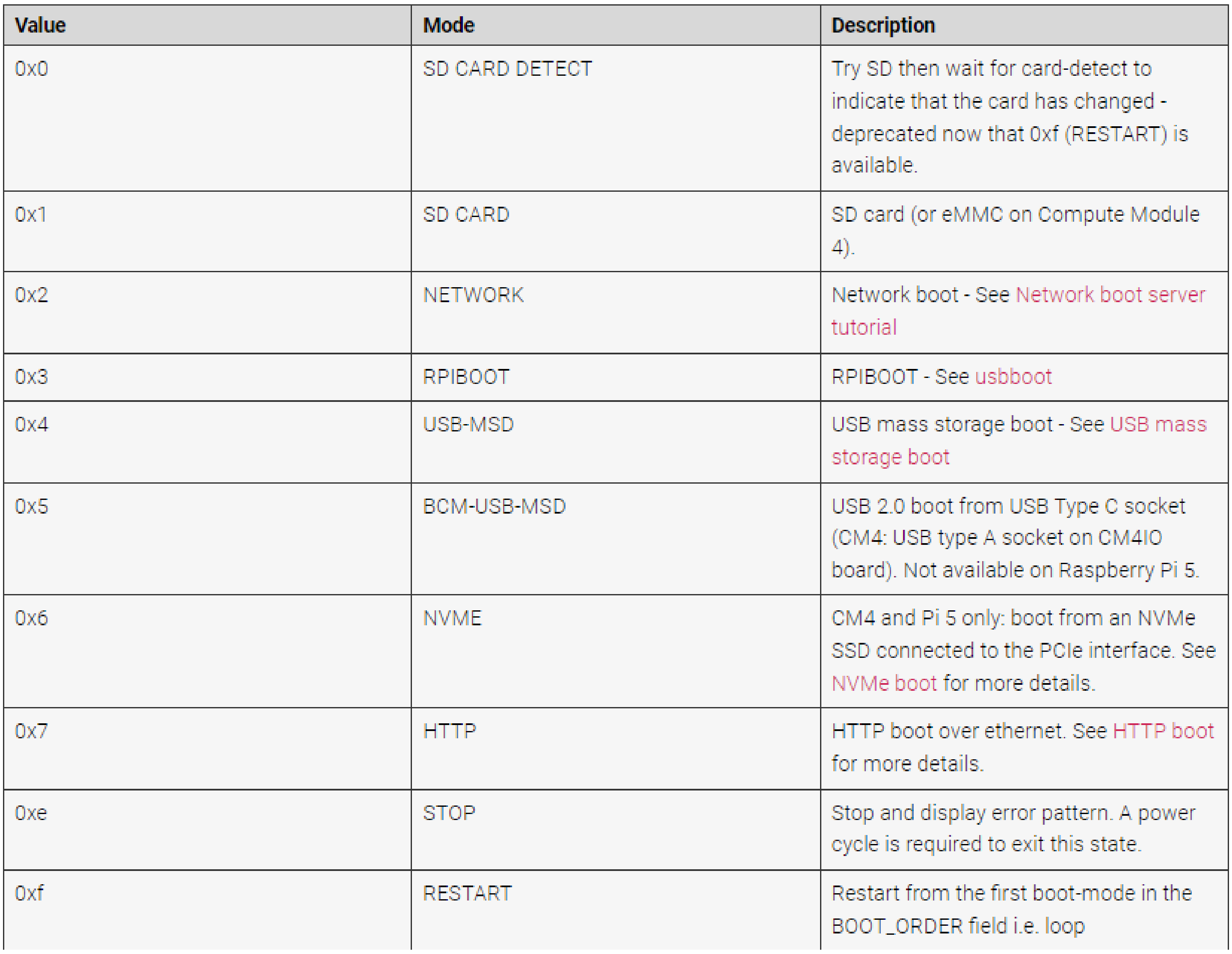
Booting from NVMe SSD

1: First, you can use an SD card to boot the Raspberry Pi, just test it to make sure the hardware can work properly.

2: Use the SD card to boot the Raspberry Pi and modify the config file, modify BOOT_ORDER:

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

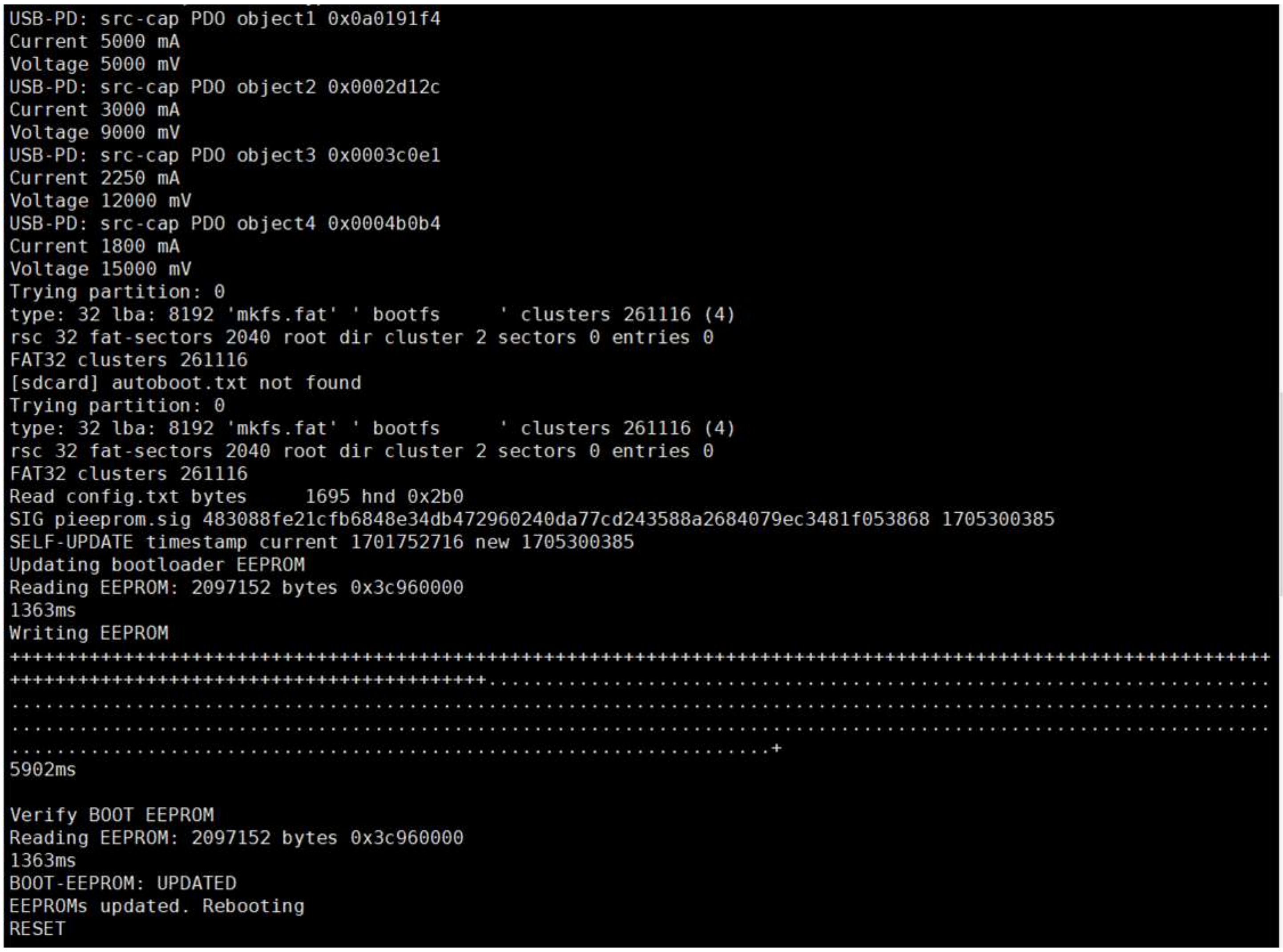
Modify BOOT_ORDER=0xf41 as BOOT_ORDER=0xf416



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 Network boot server tutorial
0x3	RPIBOOT	RPIBOOT - See usbboot
0x4	USB-MSD	USB mass storage boot - See USB mass storage boot
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 NVMe boot for more details.
0x7	HTTP	HTTP boot over ethernet. See HTTP boot 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

For more details, you can refer to BOOT_ORDER

3: Reboot the Raspberry Pi, and you can see the following content in serial port log during start-up:



```
USB-PD: src-cap PDO object1 0x0a0191f4
Current 5000 mA
Voltage 5000 mV
USB-PD: src-cap PDO object2 0x0002d12c
Current 3000 mA
Voltage 9000 mV
USB-PD: src-cap PDO object3 0x0003c0e1
Current 2250 mA
Voltage 12000 mV
USB-PD: src-cap PDO object4 0x0004b0b4
Current 1800 mA
Voltage 15000 mV
Trying partition: 0
type: 32 lba: 8192 'mkfs.fat' ' bootfs     ' clusters 261116 (4)
rsc 32 fat-sectors 2040 root dir cluster 2 sectors 0 entries 0
FAT32 clusters 261116
[sdcard] autoboot.txt not found
Trying partition: 0
type: 32 lba: 8192 'mkfs.fat' ' bootfs     ' clusters 261116 (4)
rsc 32 fat-sectors 2040 root dir cluster 2 sectors 0 entries 0
FAT32 clusters 261116
Read config.txt bytes     1695 hnd 0x2b0
SIG pieeprom.sig 483088fe21cfb6848e34db472960240da77cd243588a2684079ec3481f053868 1705300385
SELF-UPDATE timestamp current 1701752716 new 1705300385
Updating bootloader EEPROM
Reading EEPROM: 2097152 bytes 0x3c960000
1363ms
Writing EEPROM
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++.....................................................................
...............................................................................................................
...............................................................................................................
...................................................................+
5902ms

Verify BOOT EEPROM
Reading EEPROM: 2097152 bytes 0x3c960000
1363ms
BOOT-EEPROM: UPDATED
EEPROMs updated. Rebooting
RESET
```

That means the modification is successful.

If you fail after trying several times, you can connect it to the network before modify again (wait for network time synchronization), or set the correct time before modifying the file.

4: Flash the system to NVME, and then connect to the board, remove the SD card, and power it on again.