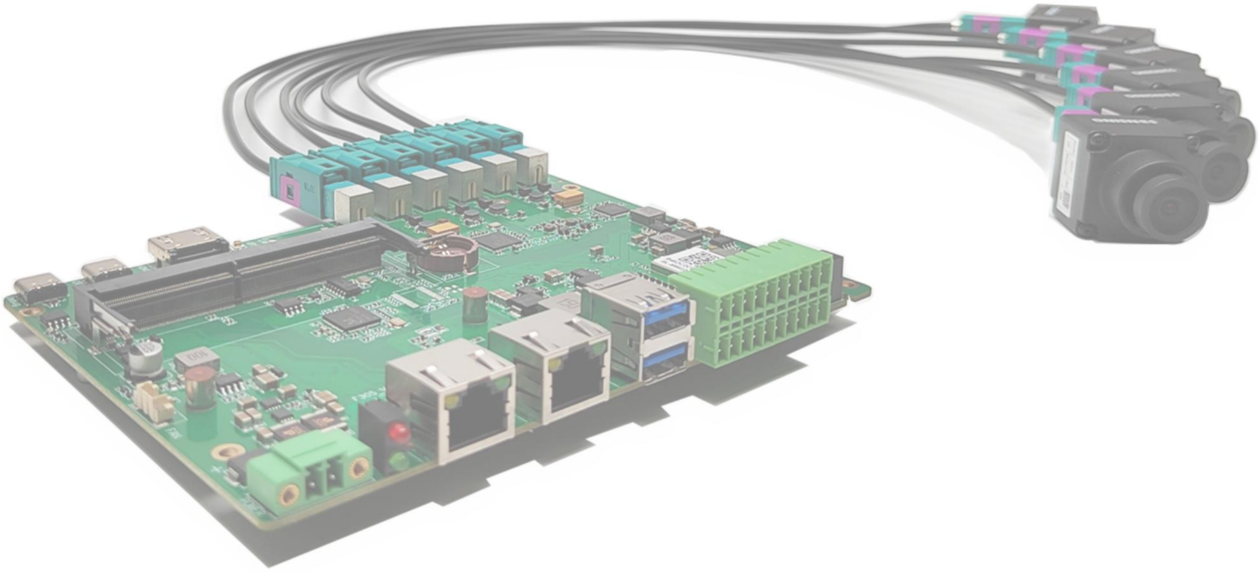
User manual

SG6C-ORNX-G2-F



Version 1.0

Disclaimers

SENSING reserves the right to edit/modify this document without any prior notice.

Catalogue

Document Revision of curriculum vitae

Rev Document version number	Date Date of revision	Description Description	Author Author
V1.0	2023/03/28	Initial release	Research and Development

Safety warnings and precautions for use

- **Safety instructions**

Before using this product, you must first consult this document to gain an initial understanding of the product and follow the safety instructions in the product's user manual to ensure your personal safety and to avoid damage to the equipment. The manufacturer is not responsible for any problems with the equipment or the safety of your life and property caused by its incorrect operation.

- **Supply voltage**

Carrier board supports input power range: 12-36V DC; current: 3A or more

- **Environmental requirements:**

Operating temperature: -20°C - 65°C

Ventilation requirements: The area around the installation of the calculation platform must be well ventilated.

- **Grounding requirements**

The power supply of the power adapter must be well earthed, in special scenarios it is necessary to install an earth screw to earth.

- **Static Protection**

Electronic components and circuits are sensitive to electrostatic discharge. Although we design our circuit board card products with anti-static protection for the main interfaces on the board, it is difficult to achieve anti-static safety protection for all components and circuits. It is therefore advisable to observe anti-static safety measures when handling any circuit board components. ESD safety measures include, but are not limited to, the following:

- ◆ The box should be placed in an anti-static bag during transport and storage until the installation and deployment of this carrier board card;
- ◆ The static electricity stored in the body should be discharged before the body comes into contact with the box: wear a discharge grounding wristband;
- ◆ Operate the box only within the safe area of the static discharge point;
- ◆ Avoid moving boxes in carpeted areas.

Explanation of nouns

POC	Power Over Coax
GMSL	Gigabit Multimedia Serial Links
FPDLINK	Flat Panel Display Link
SerDes	Serializer and Deserializer
D-PHY	The original version of the MIPI D-PHY was designed for

	500Mbits/s, while D is 500 in Roman numerals (Latin numerals), and C and M are 100 and 1000 in Roman numerals respectively, i.e. C and M in C-PHY and M-PHY.
C-PHY	C-PHYs may be used in channel-limited applications, hence the use of the character "C", 3-Phase symbol encoding technology, which allows each symbol to 2.28bits of data per symbol, 2.27 times more efficient than D-PHY using a 5-input transmission

Chapter 1 Introduction to the SG6C-ORNX-G2-F Carrier Board

The SG6C-ORNX-G2-F is a computing platform carrier board designed for NVIDIA Jetson Xavier/Orin NX series modules, and is also compatible with TX2-NX and Nano SOM core modules. PCIe expandable 4G, M.2 Key-B access to 5G modules, 2-way CAN bus, 2-way USB3.0 compatible with 2.0, etc. With long MTBF stable operation capability, it is widely used in low-speed automatic driving controller, logistics, security, transportation and other industries, and the end product form is rich for application in robots, drones, unmanned delivery vehicles, unmanned ships, intelligent law enforcement equipment, etc.

1.1 SG6C-ORNX-G2-F Carrier Board Features

- Support for NVIDIA Jetson Xavier/Orin NX (compatible with TX2-NX/Nano)
- Supports M.2 KEY M 2280 (PCIex4 NVMe SSD or Compute Stick)
- Supports 6 GMSL cameras (GMSL1 and GMSL2 compatible)
- Supports M.2 KeyB extension interface for access to 5G modules
- Supports mini-PCIe expansion interface for access to 4G modules
- Supports dual Gigabit RJ45 network interfaces
- Dual-band WIFI support and optional GPS support
- Supports a wide range of interfaces (e.g. USB/Serial/CAN/GPIO, etc.)

- Compatible with installation of different versions of NVIDIA JetPack SDK

1.2 SG6C-ORNX-G2-F Carrier Board Specifications

Product definition			
Type	Items	Description	Comment
SOM Core	Processor	NVIDIA Jetson Xavier NX (21TOPS)	
		NVIDIA Jetson Orin NX (100TOPS)	
		NVIDIA Jetson TX2-NX/Nano	
	LPDDR	8GB/16GB LPDDR5	
	HDD storage	Xavier NX with 16G eMMC 5.1 Orin NX without internal eMMC can use SSD	
Network Interface	Wired Network	Supports 1x Gigabit RJ45	RJ45
		Supports 1x Gigabit RJ45	PCIe to NIC
	WIFI network	WIFI 2.4G/5GHz compatible BT	Must link: BL_M8822CU
	4G network	Compatible with the MobileTech EC20 (mini-PCIe)	Uses USB 2.0 protocol

	5G networks	Compatible with RM500U-CN (M.2 Key B)	Uses USB 3.0 protocol
	GNSS	L76K module supports GPS/BeiDou-2.0M accuracy	UART is used (no module by default)

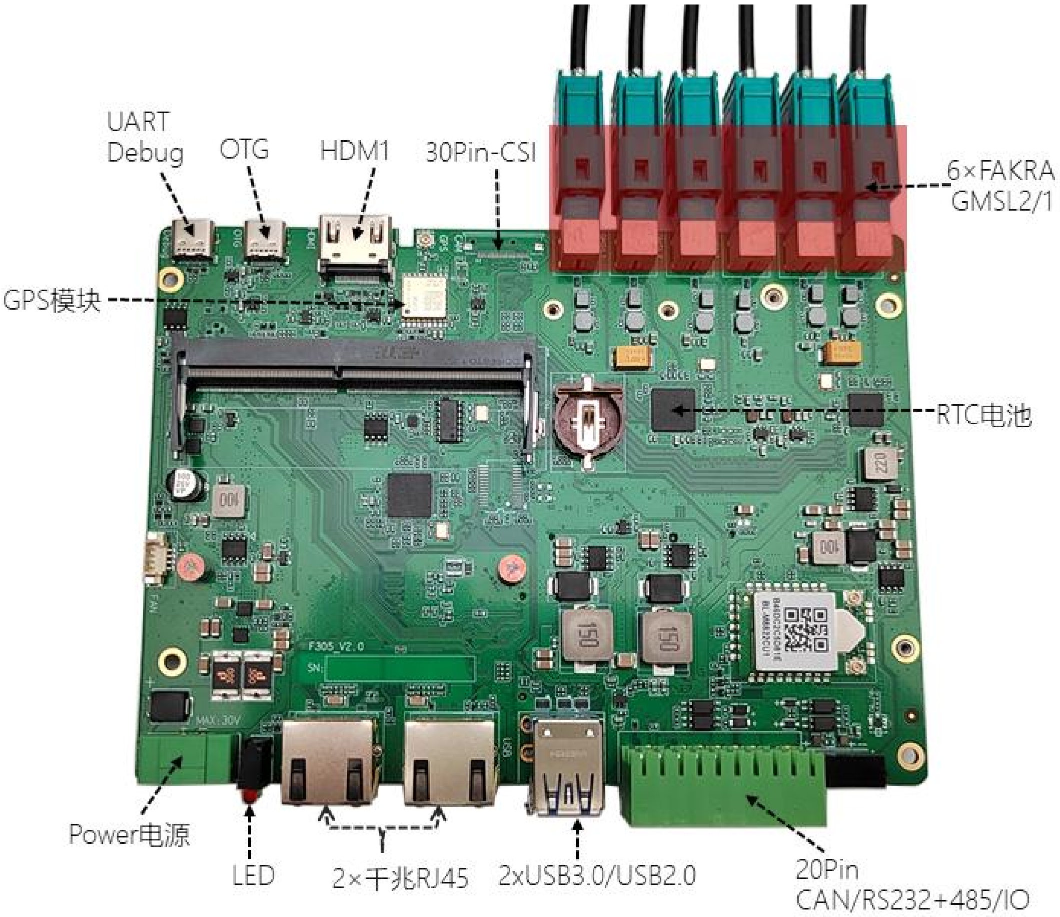
Functional interfaces	M.2 interface	M.2-Key M SSD support, length 2280	Can support 1TB
	USB port	2xUSB3.0 compatible with 2.0 TYPE-A 1xOTG USB2.0 TYPE-C (download) 1xUSB2.0 TYPE-C Debug (USB to UART Debug interface)	Supports USB 5V/1.5A per port, 2xUSB3.0 Hub RTS5411 TYPE-C Debug interface for accessing the device UART Debug information on the PC
	Camera interface	2-way GMSL FAKRA (1 x MAX9296)	Compatible with various serializers such as GMSL2/GMSL1
		4-way GMSL FAKRA (1 x MAX96712)	Compatible with various serializers such as GMSL2/GMSL1
		Supports one set of 4Lane CSI interfaces	30Pin ZIF connector

	20Pin Header	2xDI 3.3V TTL 2xDO 3.3V TTL 2x CAN_H/L (with isolation) 1x RS232 1x RS485 1x Sync signal (input to GMSL)	2x10Pin 2mm Pitch Phoenix Terminal
	T-Card	T-card interface, no hot-swap support	3.3V@1A (This T-card interface is not supported if Orin NX is used)
	RTC	1x RTC	3mAh 6.8mm X 2.1mm

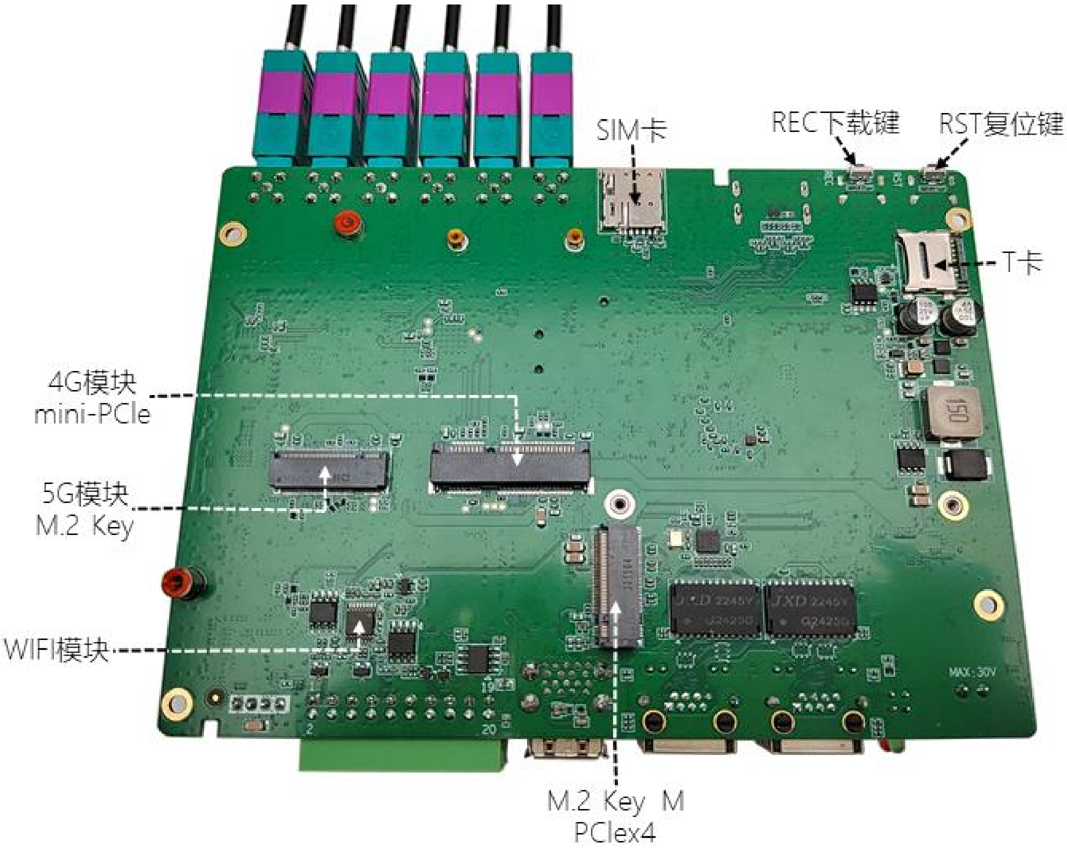
	FAN	4-Pin fan connector	Support PWM regulation (original fan) 5V power supply
	Display interface	HDMI 2.0	Horizontal
	Button button	Reset button	Hardware reset button
		Recover button	For download mode
	LED lights	Power indicator	Red light
		Status indicator	Green indicator, API available for control
	SIM card	Push-pull	Side
Power	Start up	Power on since	Power-up controlled by MCU

supply specifications	Shutdown	Does not support power-off delay	
	Electricity supply	12~30V DC	
	Power consumption	≤30W	Subject to change depending on peripherals
Complete machine form	Size	Length*width*height	155x124x24.5mm
	Heat Dissipation	Active cooling support	Fully compatible with NX kit fans
Temperature and humidity	Operating temperature	-20°C ~ +65°C	
	Storage temperature	-40°C ~ +85°C	

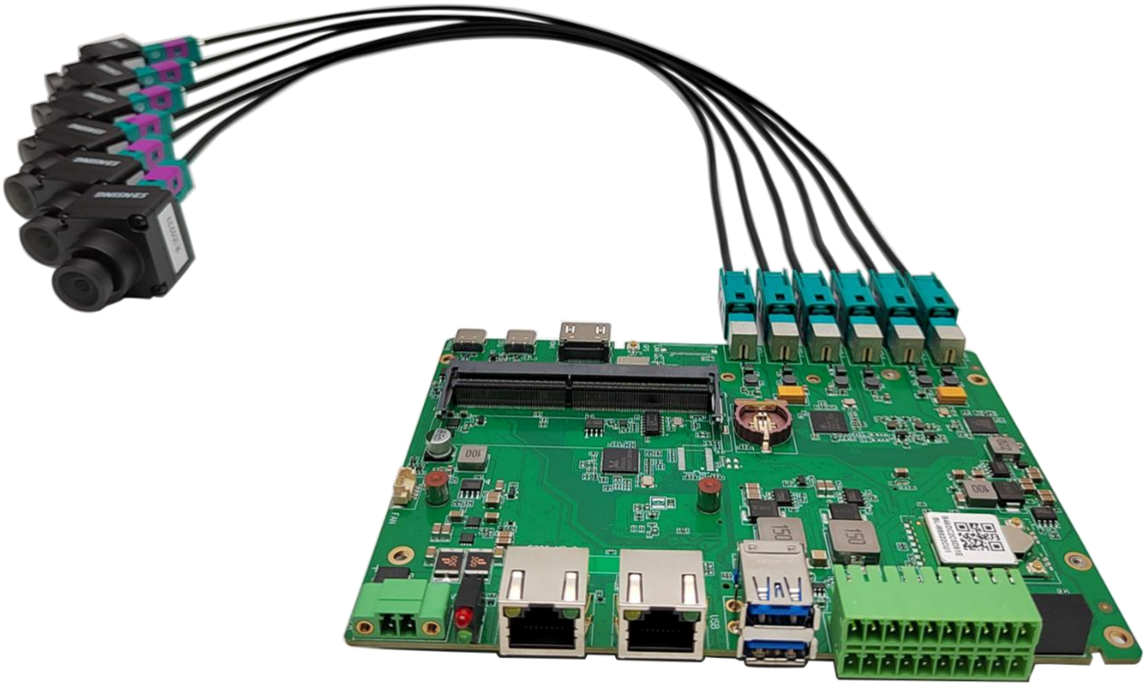
- Front interface view of the carrier board

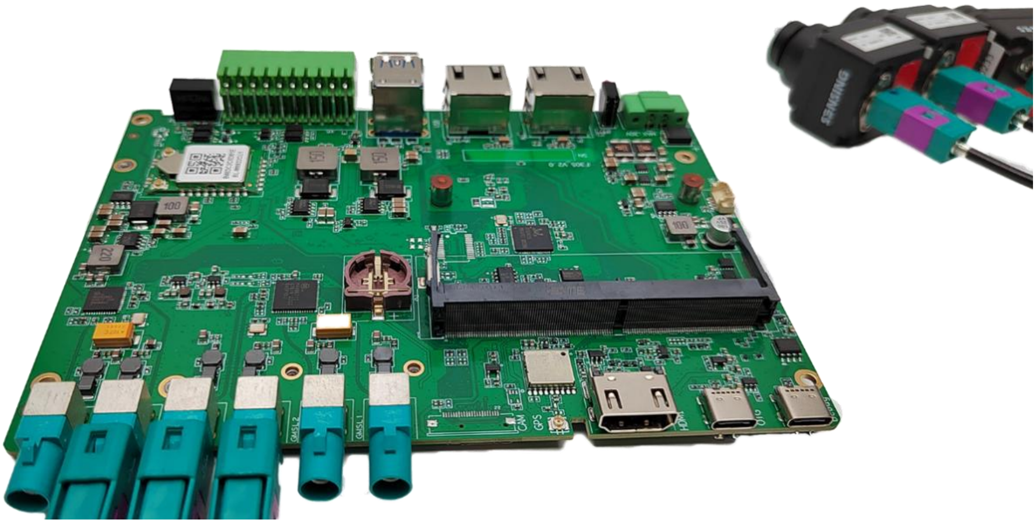


- Back of Carrier Board Interface Diagram



- Side view interface view of the carrier board





Chapter 2 SG6C-ORNX-G2-F Carrier board operating instructions

Reference link to Nvidia's official website:

https://docs.nvidia.com/jetson/archives/r35.2.1/DeveloperGuide/text/IN/QuickStart.html#in-quickstart

2.1 Setting up the environment

The SG6C-ORNX-G2-F carrier board is used with NVIDIA Jetson Xavier/Orin NX and other core boards. The driver package is developed on a specific Jetson Linux version and you need to make sure the Jetson Linux version matches before installing the driver. If it does not match, you need to re-flash the system, otherwise the system will not boot after installing the driver. NVIDIA Jetson Xavier/Orin NX supports two types of flushing methods, using SDK Manager and via the Flash.sh script. Here is how to use the Flash.sh script.

Preparation required:

- SG6C-ORNX-G2-F carrier board with NVIDIA Jetson Xavier/Orin NX module and SSD
- Computer with Ubuntu 18.04/20.04s 1
- USB TYPE-C data cable 1pc

2.1.1 Set RECOVERY download mode

The Nvidia Jetson platform software is upgraded via the USB interface and the Jetson device needs to be put into Recovery mode before the upgrade, which allows system updates, including the kernel Kernel, file system RootFS, JetPack SDK, etc.

To enter Recovery download mode in the off state, proceed as follows:

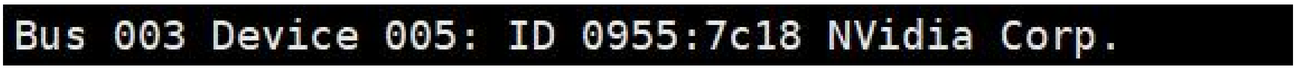
(1) Connecting a USB TYPE-C cable

Use a USB TYPE-C cable to connect the Jetson device to the Ubuntu Host, with one end connected to the TYPE-C port of the Jetson device and the other end connected to the USB port of the Ubuntu Host

(2) Pressing the REC download button (hold) and then powering up the system using the power supply paired with the carrier board;

(3) Wait for more than 5s and release all buttons to enter Recovery download mode

You can run the command lsusb on the Ubuntu Host to check for NVIDIA Corp devices to confirm that the Jetson device has successfully entered Recovery download mode. The USB VID/PID varies from Jetson module to Jetson module.



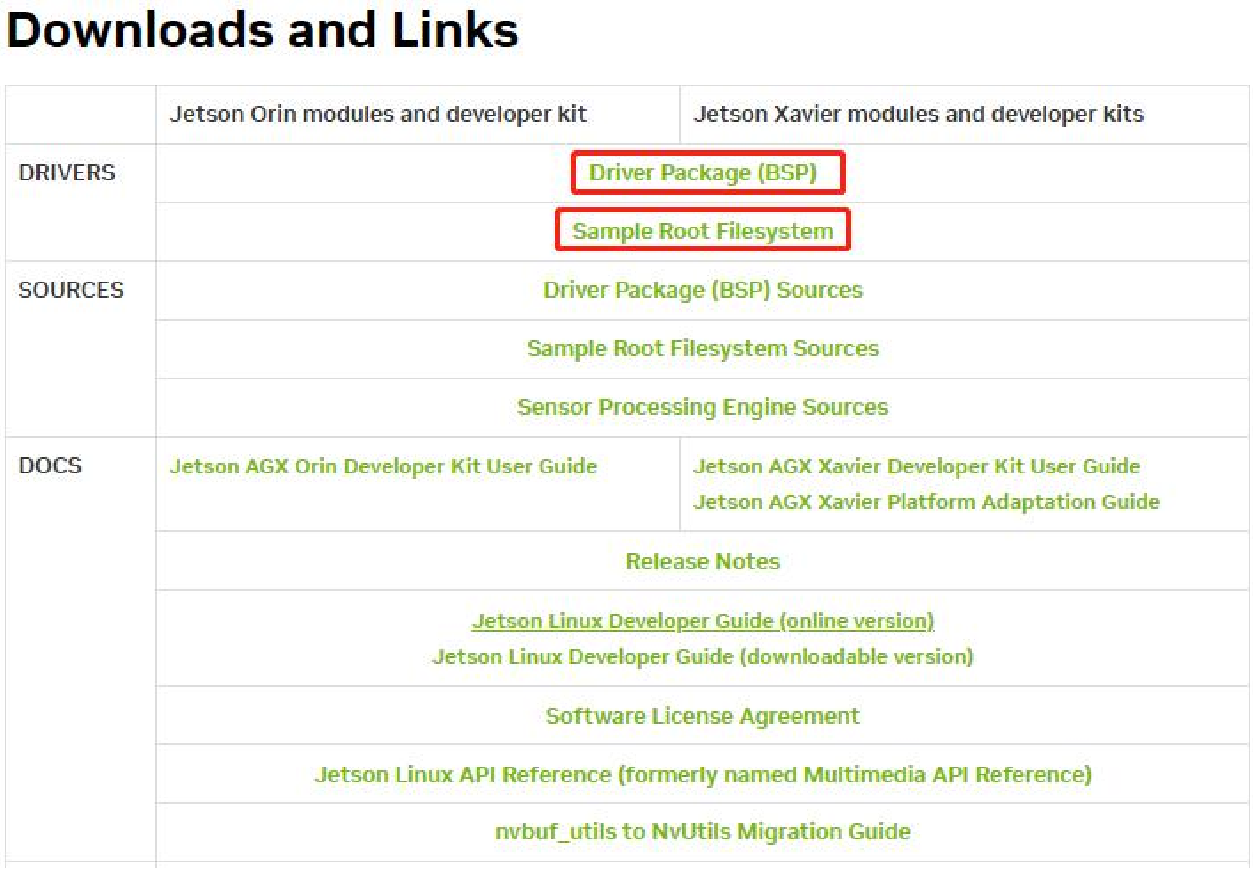
```
Bus 003 Device 005: ID 0955:7c18 NVidia Corp.
```

2.1.2 Use the flash.sh script to flush the machine

(1) Download the brush package

https://developer.nvidia.com/embedded/jetson-linux

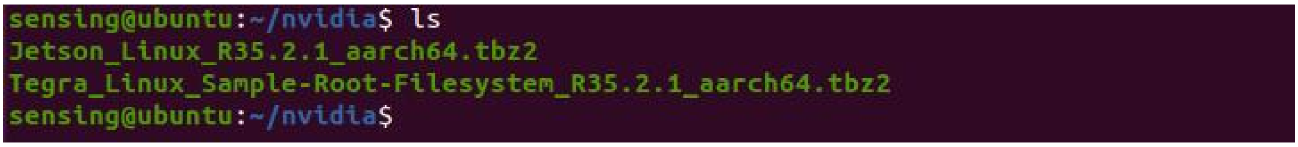
Depending on the driver version, select the corresponding Jetson Linux version, here JetPack 5.1 (Jetson Linux 35.2.1) is used as an example.



Downloads and Links

	Jetson Orin modules and developer kit	Jetson Xavier modules and developer kits
DRIVERS	Driver Package (BSP)	
	Sample Root Filesystem	
SOURCES	Driver Package (BSP) Sources	
	Sample Root Filesystem Sources	
	Sensor Processing Engine Sources	
DOCS	Jetson AGX Orin Developer Kit User Guide	Jetson AGX Xavier Developer Kit User Guide Jetson AGX Xavier Platform Adaptation Guide
	Release Notes	
	Jetson Linux Developer Guide (online version) Jetson Linux Developer Guide (downloadable version)	
	Software License Agreement	
	Jetson Linux API Reference (formerly named Multimedia API Reference)	
	nvbuf_utils to NvUtils Migration Guide	

Click on Driver Package to download Jetson_Linux_R35.2.1_aarch64.tbz2, click on Sample Root Filesystem to download Tegra_Linux_Sample-Root-Filesystem_R35.2.1_aarch64. tbz2 and copy it to the working directory of your Ubuntu Host.



```
sensing@ubuntu:~/nvidia$ ls
Jetson_Linux_R35.2.1_aarch64.tbz2
Tegra_Linux_Sample-Root-Filesystem_R35.2.1_aarch64.tbz2
sensing@ubuntu:~/nvidia$
```

(2) Unzip and install the brush package

Unzip the package, note that the filesystem needs to be extracted to the Linux_for_Tegr a/rootfs directory via sudo.

```
$ tar -jxvf Jetson_Linux_R35.2.1_aarch64.tbz2
$ sudo tar -jxvf Tegra_Linux_Sample-Root-Filesystem_R35.2.1_aarch64.tbz2 -C
Linux_for_Tegra/rootfs/
```

Execute the following script

```
$ cd Linux_for_Tegra/
$ sudo . /apply_binaries.sh
$ sudo . /tools/l4t_flash_prerequisites.sh
```

Note: If an error is reported during the execution of this script, follow the prompts to install the appropriate dependency package.

As the carrier board does not have an eeprom, the following files need to be modified

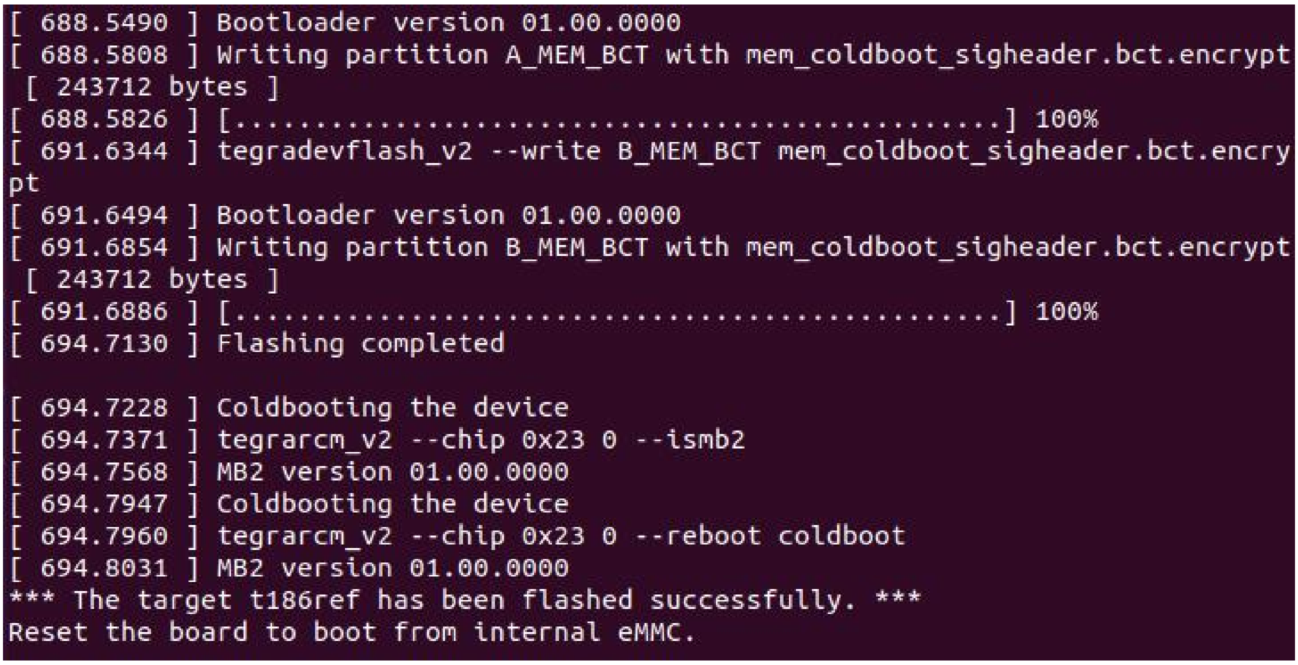
Linux for_Tegra/bootloader/t186ref/BCT/ tegra234-mb2-bct-misc-p3767-0000.dts

```
- cvb_eeprom_read size =<0x100>
+ cvb_eeprom_read size =<0x0>
```

(3) Refresh

Refer to section "2.1.1 Setting up Recovery Download Mode" to put the Jetson device into Recovery download mode. After confirming that the device is recognized with the lsusb command, execute the following command to refresh the device.

```
$ sudo . /tools/kernel_flash/l4t_initrd_flash.sh \
    --external-device nvme0n1p1 \
-c tools/kernel_flash/flash_l4t_external.xml \
-p "-c bootloader/t186ref/cfg/flash_t234_qspi.xml" \
--showlogs --network usb0 p3509-a02+p3767-0000 internal
```



```
[ 688.5490 ] Bootloader version 01.00.0000
[ 688.5808 ] Writing partition A_MEM_BCT with mem_coldboot_sigheader.bct.encrypt
 [ 243712 bytes ]
[ 688.5826 ] [................................................] 100%
[ 691.6344 ] tegradevflash_v2 --write B_MEM_BCT mem_coldboot_sigheader.bct.encry
pt
[ 691.6494 ] Bootloader version 01.00.0000
[ 691.6854 ] Writing partition B_MEM_BCT with mem_coldboot_sigheader.bct.encrypt
 [ 243712 bytes ]
[ 691.6886 ] [................................................] 100%
[ 694.7130 ] Flashing completed

[ 694.7228 ] Coldbooting the device
[ 694.7371 ] tegrarcm_v2 --chip 0x23 0 --ismb2
[ 694.7568 ] MB2 version 01.00.0000
[ 694.7947 ] Coldbooting the device
[ 694.7960 ] tegrarcm_v2 --chip 0x23 0 --reboot coldboot
[ 694.8031 ] MB2 version 01.00.0000
*** The target t186ref has been flashed successfully. ***
Reset the board to boot from internal eMMC.
```

After brushing, the Jetson device automatically reboots into the Linux system and follows the wizard to complete the configuration to access the desktop.

2.2 Application Notes

2.2.1 Network port applications

The carrier board has two Gigabit RJ45 network interfaces, one of which is the core module with its own Gigabit network, and the other is the PCIex1 expansion using the Realtek RTL8111 chip. Test network bandwidth and packet loss, etc. are as follows:

(1) Testing Tools

iperf or iperf3 (Windows or Linux)

sudo apt-get install iperf

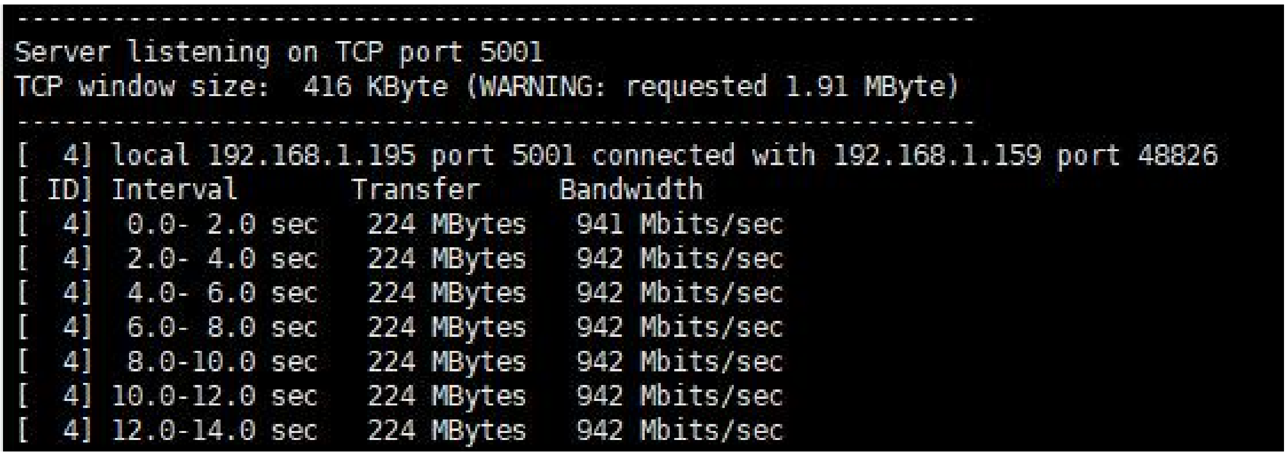
(2) Test network bandwidth, stable operation bandwidth up to 940Mbits / sec

PC computer (IP: 192.168.1.195)

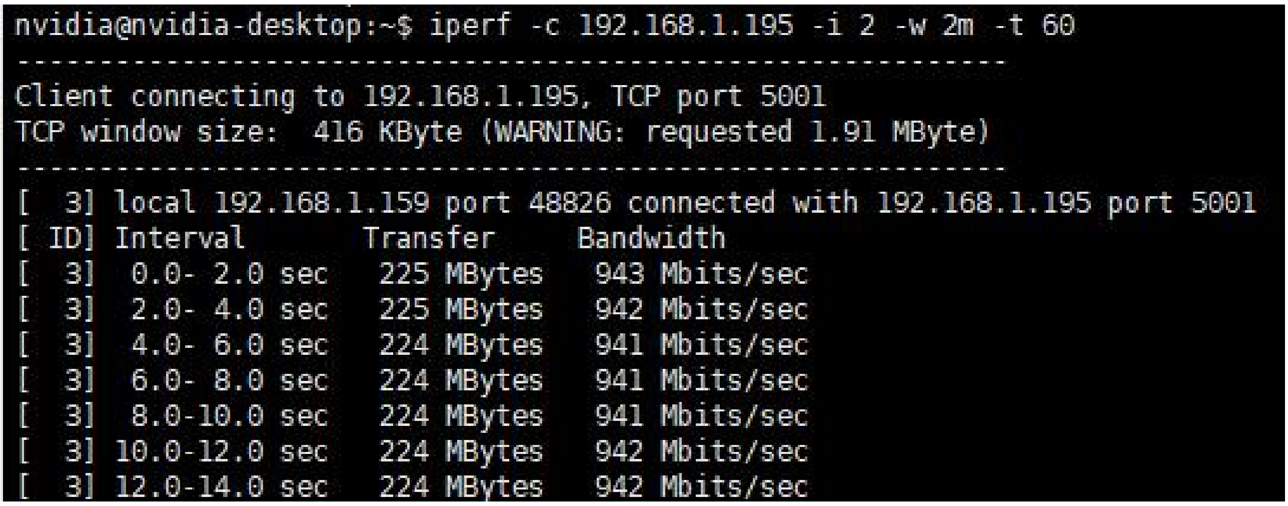
Carrier machine (IP: 192.168.1.159)

PC computer running as server: iperf -s -i 2 -w 2m (IP: 192.168.1.2)

Carrier board running as device side: iperf -c 192.168.1.2-i 2 -w 2m -t 60



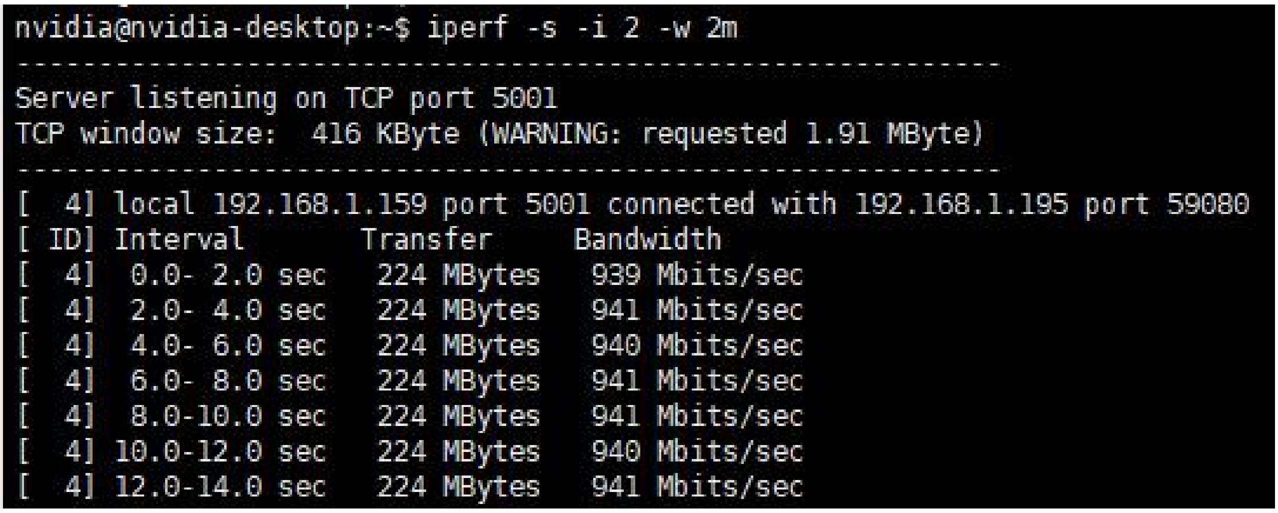
```
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  416 KByte (WARNING: requested 1.91 MByte)
------------------------------------------------------------
[  4] local 192.168.1.195 port 5001 connected with 192.168.1.159 port 48826
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0- 2.0 sec   224 MBytes   941 Mbits/sec
[  4]  2.0- 4.0 sec   224 MBytes   942 Mbits/sec
[  4]  4.0- 6.0 sec   224 MBytes   942 Mbits/sec
[  4]  6.0- 8.0 sec   224 MBytes   942 Mbits/sec
[  4]  8.0-10.0 sec   224 MBytes   942 Mbits/sec
[  4] 10.0-12.0 sec   224 MBytes   942 Mbits/sec
[  4] 12.0-14.0 sec   224 MBytes   942 Mbits/sec
```



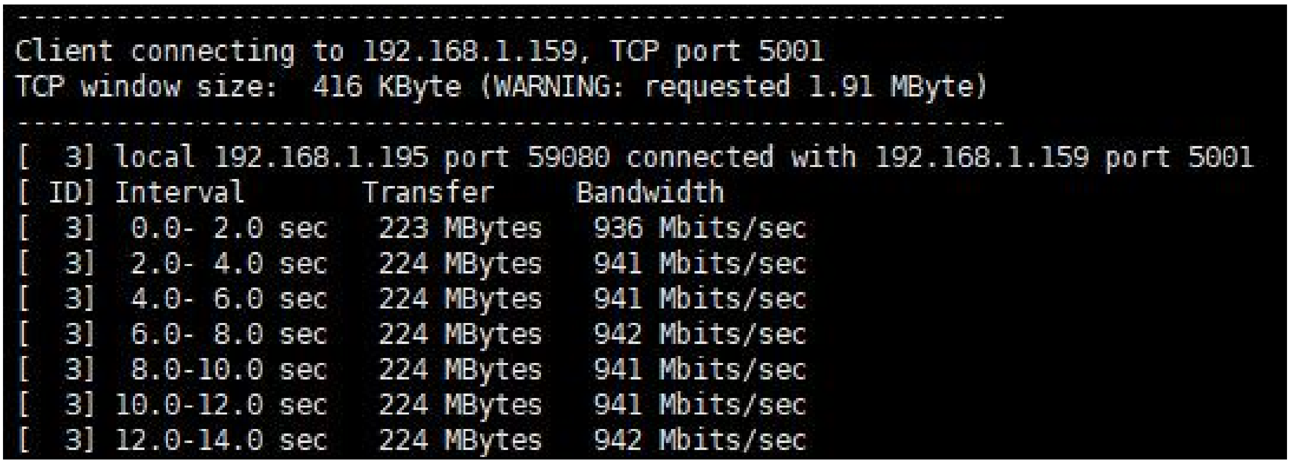
```
nvidia@nvidia-desktop:~$ iperf -c 192.168.1.195 -i 2 -w 2m -t 60
------------------------------------------------------------
Client connecting to 192.168.1.195, TCP port 5001
TCP window size:  416 KByte (WARNING: requested 1.91 MByte)
------------------------------------------------------------
[  3] local 192.168.1.159 port 48826 connected with 192.168.1.195 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 2.0 sec   225 MBytes   943 Mbits/sec
[  3]  2.0- 4.0 sec   225 MBytes   942 Mbits/sec
[  3]  4.0- 6.0 sec   224 MBytes   941 Mbits/sec
[  3]  6.0- 8.0 sec   224 MBytes   941 Mbits/sec
[  3]  8.0-10.0 sec   224 MBytes   941 Mbits/sec
[  3] 10.0-12.0 sec   224 MBytes   942 Mbits/sec
[  3] 12.0-14.0 sec   224 MBytes   942 Mbits/sec
```

Carrier board running as server: iperf -s -i 2 -w 2m (IP: 192.168.1.159)

PC computer as device side run: iperf -c 192.168.1.159-i 2 -w 2m -t 60



```
nvidia@nvidia-desktop:~$ iperf -s -i 2 -w 2m
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  416 KByte (WARNING: requested 1.91 MByte)
------------------------------------------------------------
[  4] local 192.168.1.159 port 5001 connected with 192.168.1.195 port 59080
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0- 2.0 sec   224 MBytes   939 Mbits/sec
[  4]  2.0- 4.0 sec   224 MBytes   941 Mbits/sec
[  4]  4.0- 6.0 sec   224 MBytes   940 Mbits/sec
[  4]  6.0- 8.0 sec   224 MBytes   941 Mbits/sec
[  4]  8.0-10.0 sec   224 MBytes   941 Mbits/sec
[  4] 10.0-12.0 sec   224 MBytes   940 Mbits/sec
[  4] 12.0-14.0 sec   224 MBytes   941 Mbits/sec
```

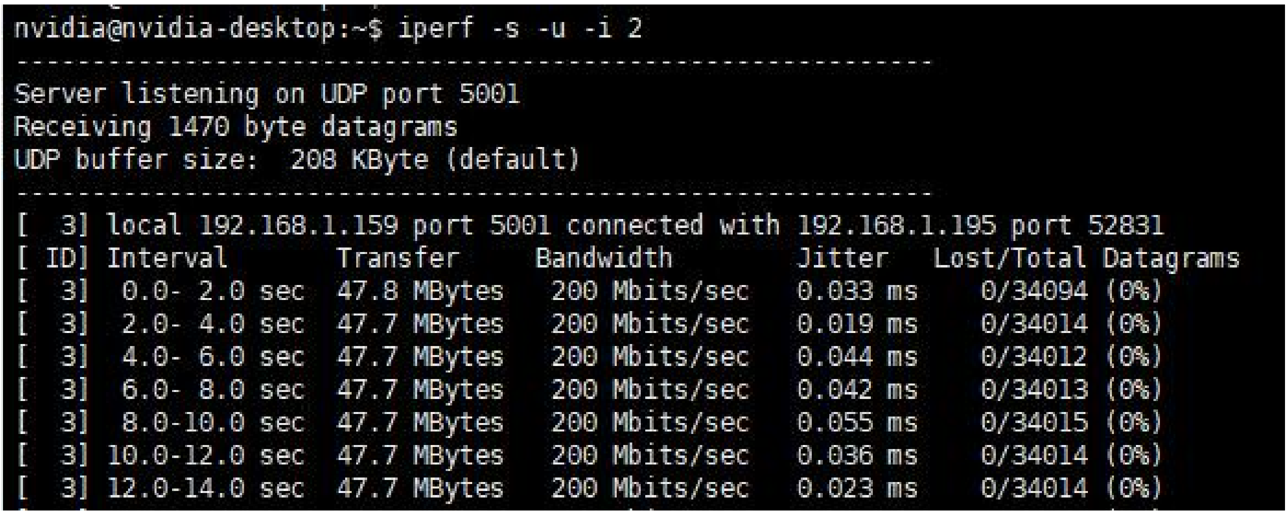


```
------------------------------------------------------------
Client connecting to 192.168.1.159, TCP port 5001
TCP window size:  416 KByte (WARNING: requested 1.91 MByte)
------------------------------------------------------------
[  3] local 192.168.1.195 port 59080 connected with 192.168.1.159 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 2.0 sec   223 MBytes   936 Mbits/sec
[  3]  2.0- 4.0 sec   224 MBytes   941 Mbits/sec
[  3]  4.0- 6.0 sec   224 MBytes   941 Mbits/sec
[  3]  6.0- 8.0 sec   224 MBytes   942 Mbits/sec
[  3]  8.0-10.0 sec   224 MBytes   941 Mbits/sec
[  3] 10.0-12.0 sec   224 MBytes   941 Mbits/sec
[  3] 12.0-14.0 sec   224 MBytes   942 Mbits/sec
```

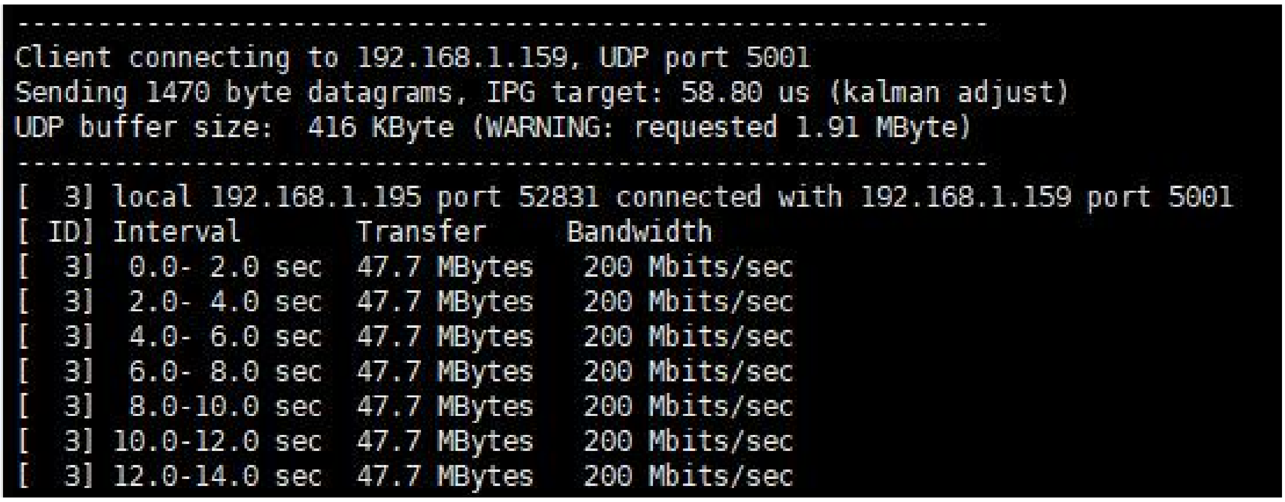
（3）Test jitter, packet loss rate

The carrier board runs as a server: iperf -s -u -i 2

PC computer as device side run: iperf -c 192.168.1.159 -i 2 -w 2m -u -b 200m -t60



```
nvidia@nvidia-desktop:~$ iperf -s -u -i 2
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 192.168.1.159 port 5001 connected with 192.168.1.195 port 52831
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 2.0 sec  47.8 MBytes   200 Mbits/sec   0.033 ms    0/34094 (0%)
[  3]  2.0- 4.0 sec  47.7 MBytes   200 Mbits/sec   0.019 ms    0/34014 (0%)
[  3]  4.0- 6.0 sec  47.7 MBytes   200 Mbits/sec   0.044 ms    0/34012 (0%)
[  3]  6.0- 8.0 sec  47.7 MBytes   200 Mbits/sec   0.042 ms    0/34013 (0%)
[  3]  8.0-10.0 sec  47.7 MBytes   200 Mbits/sec   0.055 ms    0/34015 (0%)
[  3] 10.0-12.0 sec  47.7 MBytes   200 Mbits/sec   0.036 ms    0/34014 (0%)
[  3] 12.0-14.0 sec  47.7 MBytes   200 Mbits/sec   0.023 ms    0/34014 (0%)
```



```
------------------------------------------------------------
Client connecting to 192.168.1.159, UDP port 5001
Sending 1470 byte datagrams, IPG target: 58.80 us (kalman adjust)
UDP buffer size:  416 KByte (WARNING: requested 1.91 MByte)
------------------------------------------------------------
[  3] local 192.168.1.195 port 52831 connected with 192.168.1.159 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 2.0 sec  47.7 MBytes   200 Mbits/sec
[  3]  2.0- 4.0 sec  47.7 MBytes   200 Mbits/sec
[  3]  4.0- 6.0 sec  47.7 MBytes   200 Mbits/sec
[  3]  6.0- 8.0 sec  47.7 MBytes   200 Mbits/sec
[  3]  8.0-10.0 sec  47.7 MBytes   200 Mbits/sec
[  3] 10.0-12.0 sec  47.7 MBytes   200 Mbits/sec
[  3] 12.0-14.0 sec  47.7 MBytes   200 Mbits/sec
```

2.2.2 4G Applications

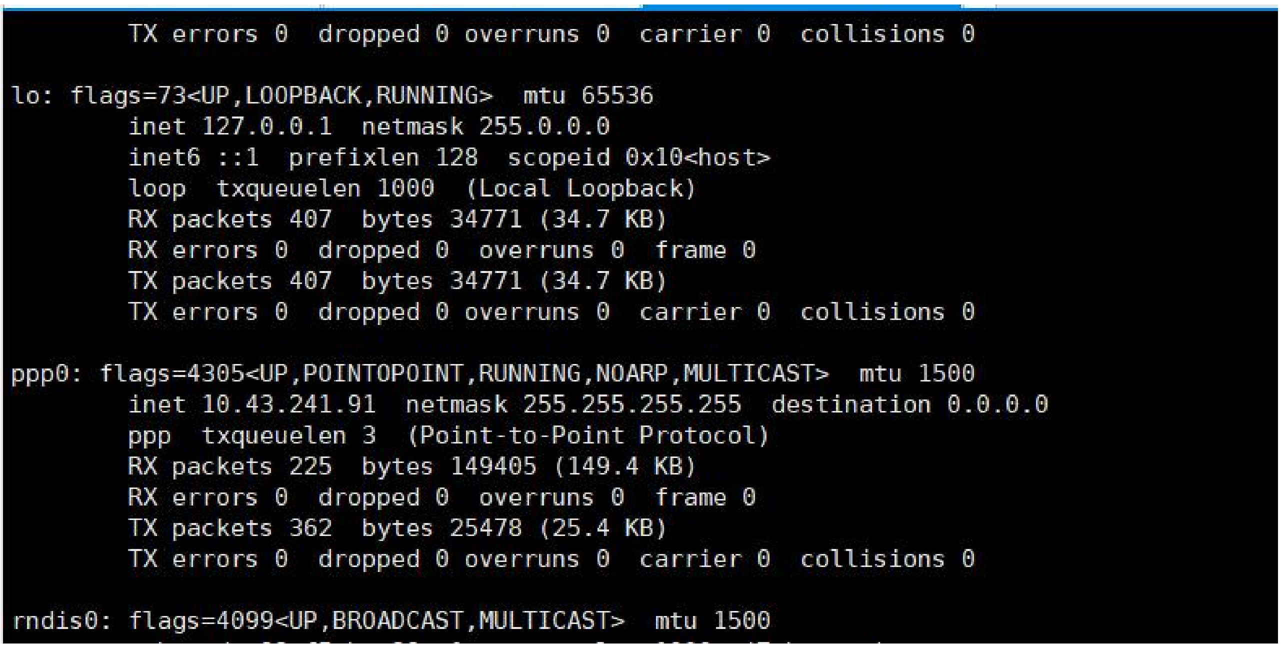
4G module (example: Mobile Farms EC20 module)

The USB device number is Bus 001 Device 005: ID 2c7c:0125

The 4G module uses ppp0 dialing method, you can follow the steps below to complete the dialing setup.

After inserting the 4G module and 4G sim card, click on "Network Settings" in the top right corner of the Ubuntu desktop, select "Mobile Broadband off" in the drop-down box, and select connect from the expanded options. After the installation prompt, select next to proceed and finally you will see connect, which means you are done dialing.

In the command line, you can use ifconfig to get the following figure, ppp0 assign ip successfully, you can do the Internet operation

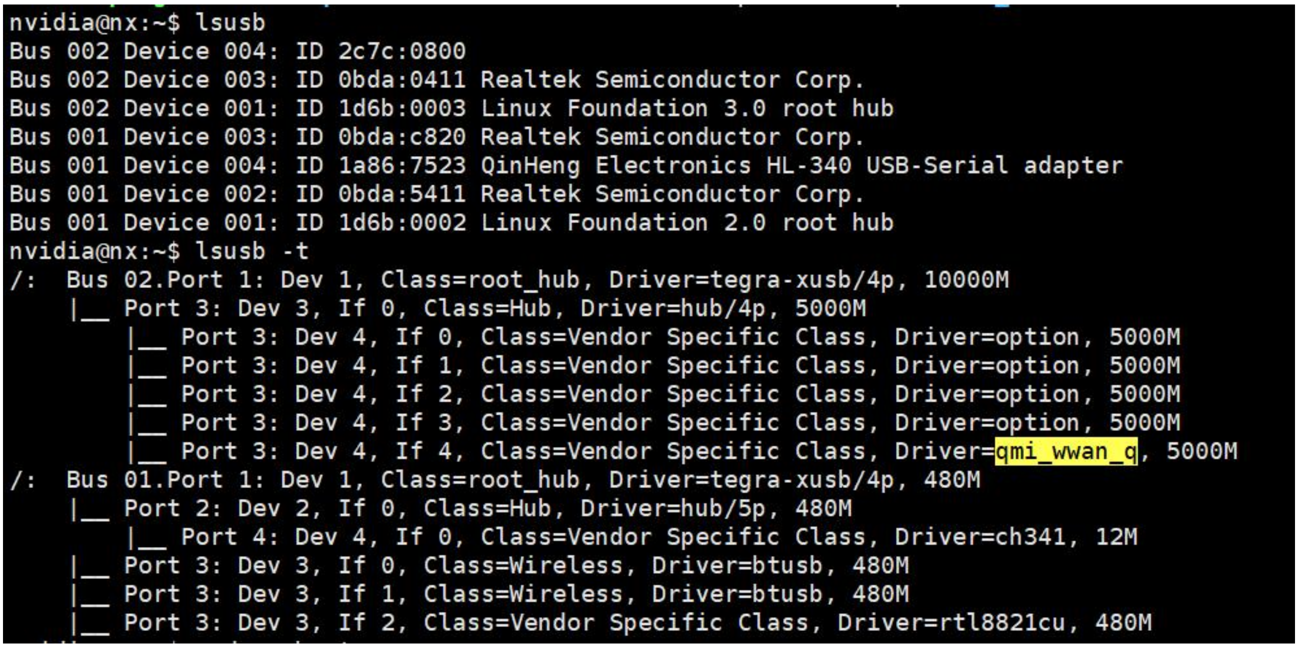


```
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 407  bytes 34771 (34.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 407  bytes 34771 (34.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ppp0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.43.241.91  netmask 255.255.255.255  destination 0.0.0.0
        ppp  txqueuelen 3  (Point-to-Point Protocol)
        RX packets 225  bytes 149405 (149.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 362  bytes 25478 (25.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

rndis0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
```

2.2.3 5G Applications

5G module (example: Mobile Far RM500U-CN module)

USB device number is Bus 002 Device 004: ID 2c7c:0800



```
nvidia@nx:~$ lsusb
Bus 002 Device 004: ID 2c7c:0800
Bus 002 Device 003: ID 0bda:0411 Realtek Semiconductor Corp.
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 0bda:c820 Realtek Semiconductor Corp.
Bus 001 Device 004: ID 1a86:7523 QinHeng Electronics HL-340 USB-Serial adapter
Bus 001 Device 002: ID 0bda:5411 Realtek Semiconductor Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
nvidia@nx:~$ lsusb -t
/:  Bus 02.Port 1: Dev 1, Class=root_hub, Driver=tegra-xusb/4p, 10000M
    |__ Port 3: Dev 3, If 0, Class=Hub, Driver=hub/4p, 5000M
        |__ Port 3: Dev 4, If 0, Class=Vendor Specific Class, Driver=option, 5000M
        |__ Port 3: Dev 4, If 1, Class=Vendor Specific Class, Driver=option, 5000M
        |__ Port 3: Dev 4, If 2, Class=Vendor Specific Class, Driver=option, 5000M
        |__ Port 3: Dev 4, If 3, Class=Vendor Specific Class, Driver=option, 5000M
        |__ Port 3: Dev 4, If 4, Class=Vendor Specific Class, Driver=qmi_wwan_q, 5000M
/:  Bus 01.Port 1: Dev 1, Class=root_hub, Driver=tegra-xusb/4p, 480M
    |__ Port 2: Dev 2, If 0, Class=Hub, Driver=hub/5p, 480M
        |__ Port 4: Dev 4, If 0, Class=Vendor Specific Class, Driver=ch341, 12M
    |__ Port 3: Dev 3, If 0, Class=Wireless, Driver=btusb, 480M
    |__ Port 3: Dev 3, If 1, Class=Wireless, Driver=btusb, 480M
    |__ Port 3: Dev 3, If 2, Class=Vendor Specific Class, Driver=rtl8821cu, 480M
```

The dial-up application adopts the dial-up tool provided by Mobile Farms (Quectel_QConnectManager_Linux_V1.6.4, which supports automatic replay after disconnection; if other modules are used, you need to contact the 5G supplier to provide relevant tools)

```
$ unzip Quectel_QConnectManager_Linux_V1.6.4.zip
$ cd Quectel_QConnectManager_Linux_V1.6.4
$ make
$ sudo . / quectel-CM
```

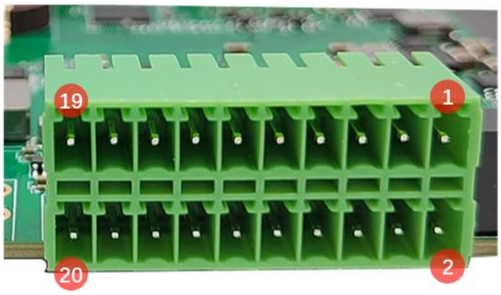
2.2.4 WIFI Applications

The carrier board has built-in USB WIFI module (brand: BILIAN), BILIAN: BL8821CU with Realtek RTL8821CU chip or BILIAN: BL8822CU with Realtek RTL8822CU chip, the specification of these two WIFI modules can be obtained by contacting technical support

USB device number:

Bus 001 Device 003: ID 0bda:c820 Realtek Semiconductor Corp.

2.2.5 20Pin multi-function interface

Pins	Function		Pins	Function
Pin1	RS485A		Pin2	Isolated DO/460
Pin3	RS485B		Pin4	Isolation DO/459
Pin5	GND		Pin6	Isolated GND
Pin7	RS232_TX		Pin8	Isolated DI/465
Pin9	RS232_RX		Pin10	Isolated DI/458
Pin11	GND		Pin12	GND

Pin13	NX_CAN_L		Pin14	SPI_CAN_L
Pin15	NX_CAN_H		Pin16	SPI_CAN_H
Pin17	GPIO12		Pin18	GND
Pin19	GPIO13		Pin20	5V/non-isolated

2.2.6 Serial RS232/RS485 Signal Definition

(1) Signal Definition

Interface Name	PHD2.0 pin number	Signal Definition	Interface Description
UART1 /dev/ttyTHS0	1	RS485_A	UAR0_TXD sends RS485 level
	3	RS485_B	UAR0_RXD receives RS485 level
	5	GND	Grounding
UART2 /dev/ttyUSB0	7	RS232_TX	RS232 send, USB to RS232
	9	RS232_RX	RS232 receiver, USB to RS232
	11	GND	Grounding

(2) UART interface configuration method

Open the corresponding device node under the /dev/ directory and set the baud rate, stop bits, parity bits, data bits, etc. You can use the stty command to configure the baud rate, stop bits, parity bits, data bits, etc. of the serial port. See the stty command description for details.

command example, please modify the information in < > to the serial port node number you want to adjust.

```
$ sudo stty -F /dev/ttyTHS<XXX> speed 115200 cs8 -parenb -cstopb -echo
```

Output data testing

```
$ sudo echo "uart debug" > /dev/ttyTHS<XXX>
```

Open a new terminal and use the following command to receive input data

```
$ sudo cat /dev/ttyTHS<XXX>
```

2.2.7 Camera Sync IO synchronization signal definition

Interface Name	PHD2.0 pin number	Signal Definition	Interface Description
Sync in1 sync interface	17	GPIO13 (NO: 419)	9296 Sync in signal
Sync in2 sync interface	19	GPIO12 (NO: 321)	96712 Sync in signal

Fixed frequency signal can be input to trigger the GMSL operation

2.2.8 CAN interface signal definition

(1) Signal Definition

Interface Name	PHD2.0 pin number	Signal Definition	Interface Description
CANL	13	NX_CAN_L	NX_CAN_L terminal
CANH	15	NX_CAN_H	NX_CAN_H terminal
CAN_L	14	CAN_FD_L	SPI to CAN_FD_L terminal
CAN_H	16	CAN_FD_H	SPI to CAN_FD_H terminal
Total GND	5/11/12/18	GND	Earth Signal

Notes:

CANL/H is the CAN controller that comes with the NX module

CAN_L/H for SPI to CAN controller (chip is MCP2518FD) CAN FD support

CAN device specific use method, refer to https://github.com/linux-can/can-utils cansend.c and candump.c in

(2) Shell script test commands

```
$ sudo modprobe can
$ sudo modprobe can_raw
$ sudo modprobe mttcan
$ sudo insmod /lib/modules/ $(uname -r)/kernel/drivers/net/can/spi/mcp25xxfd.ko
$ sudo ip link set can0 type can bitrate 200000 sjw 4 berr-reporting on loopback off
$ sudo ip link set up can0
$ sudo cansend can0 12345#abcdeabcde
$ sudo candump can0
$ sudo ip -details -statistics link show can0
```

(3) How to use CAN FD configuration

```
$ sudo modprobe can
$ sudo modprobe can_raw
$ sudo modprobe mttcan
$ sudo insmod /lib/modules/ $(uname -r)/kernel/drivers/net/can/spi/mcp25xxfd.ko
$ sudo ip link set can0 type can bitrate 200000 sjw 4 dbitrate 1000000 dsjw 4 berr-reporting on fd on
$ sudo ip link set up can0
$ sudo cansend can0 321##011
```

(4) The difference between CAN FD and CAN 2.0

```
$ sudo ip link set can0 type can bitrate 200000 dbitrate 1000000 berr-reporting on fd on
```

Where bitrate is the baud rate in can2.0 mode; dbitrate is the baud rate in can fd mode, according to the official documentation, this value can be configured up to 5M, general applications are best to use 2M.

```
$ sudo cansend can0 321##011
```

In the transmit command, there is an extra # between id and data, and the first byte (0) after ## is the value of canfd_frame.flags, ranging from 0 to F; the byte (11) after canfd_frame.flags is the first data, and up to 64 bytes can be transmitted at one time.

2.2.9 GPIO interface signal definition

GPIOs with isolated power supply method, 2 of them are input IO only (Pin2/Pin4) and 2 are used as output IO (Pin8/Pin10)

Interface Name	**PHD2.0 pin number**	**Signal Definition**	**Interface Description**
GPIO_222_3V3	2	GPIO222_3V3	GPIO0222(gpio460)
GPIO_220_3V3	4	GPIO220_3V3	GPIO0220(gpio459)
GPIO_197_3V3	8	GPIO197_3V3	GPIO0197(gpio465)
GPIO_226_3V3	10	GPIO226_3V3	GPIO226(gpio458)

An example of the use of the GPIO interface is as follows. Please change the information in <xxx > to the GPIO node designation you want to adjust:

```
# Switch to root user
$ sudo su -
$ cd /sys/class/gpio
$ echo gpio_num > export
```

Note:On JetPack5.x, exprot gpio_num generates a yyyy name instead of the pre-JetPack4.x gpionum form

# Set to high level (DO)

```
$ echo 1 > /sys/class/gpio/xxxx/value
```

# Set to low (DO)

```
$ echo 0 > /sys/class/gpio/xxx/value
```

# Read data (DI)

```
$ cat /sys/class/gpio/xxxx/value
```

# Set the input and output direction (DIRCTION)

```
$ echo in > /sys/class/gpio/xxxx/direction
$ echo out > /sys/class/gpio/xxx/direction
```

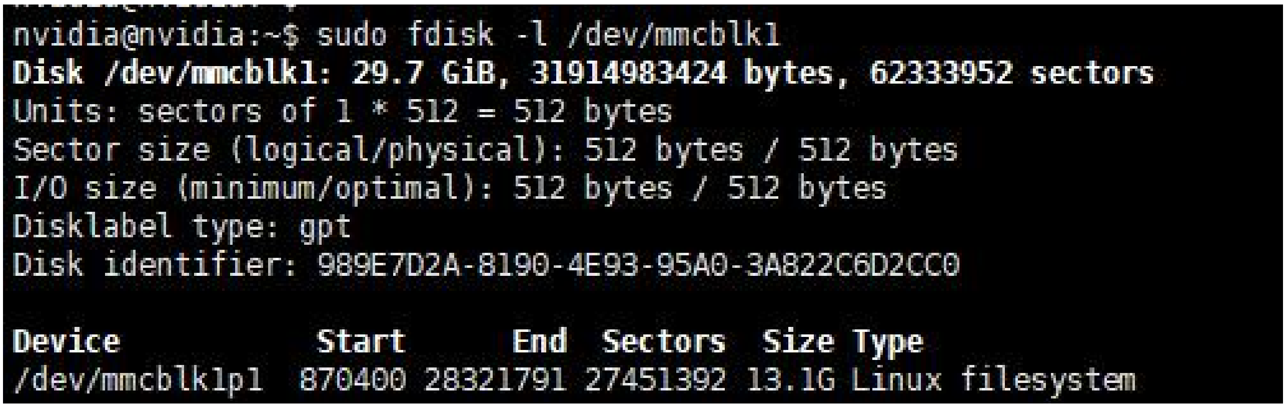
If you need to keep the configuration after shutdown, you can write the above commands to the /etc/rc.local file

2.2.10 Fan FAN

Fan (compatible with Xavier NX original fan), speed automatically adjusted according to temperature

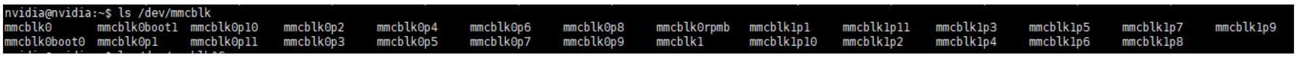
2.2.11 TF card applications

```
$ sudo fdisk -l /dev/mmcblk1
```



```
nvidia@nvidia:~$ sudo fdisk -l /dev/mmcblk1
Disk /dev/mmcblk1: 29.7 GiB, 31914983424 bytes, 62333952 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 989E7D2A-8190-4E93-95A0-3A822C6D2CC0

Device          Start      End  Sectors  Size Type
/dev/mmcblk1p1 870400 28321791 27451392 13.1G Linux filesystem
```

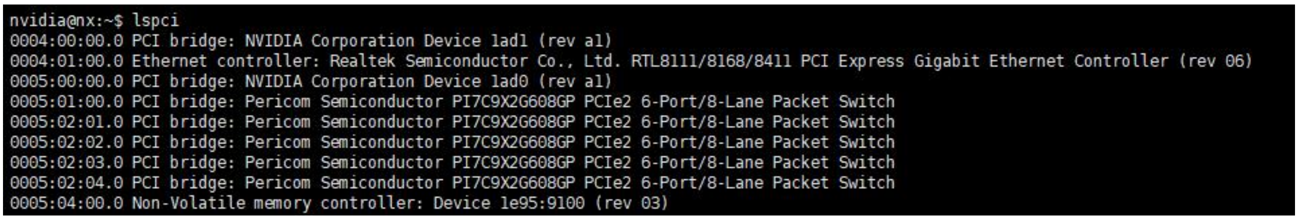
```
$ ls /dev/mmcblk*
```



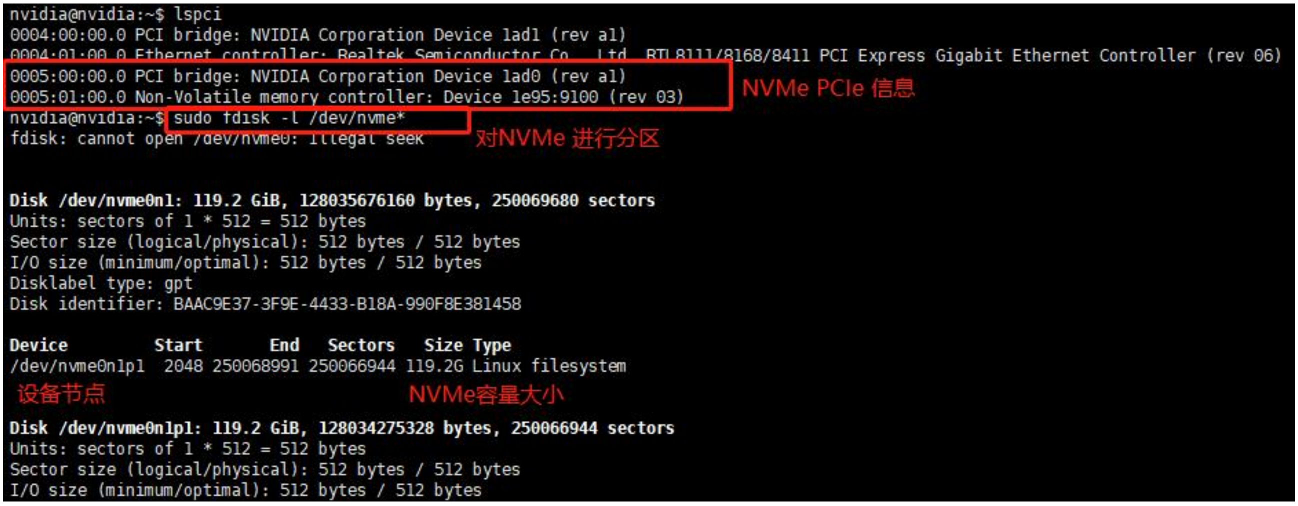
```
nvidia@nvidia:~$ ls /dev/mmcblk
mmcblk0       mmcblk0boot1  mmcblk0p10    mmcblk0p2     mmcblk0p4     mmcblk0p6     mmcblk0p8     mmcblk0rpmb   mmcblk1p1     mmcblk1p11    mmcblk1p3     mmcblk1p5     mmcblk1p7     mmcblk1p9
mmcblk0boot0  mmcblk0p1     mmcblk0p11    mmcblk0p3     mmcblk0p5     mmcblk0p7     mmcblk0p9     mmcblk1       mmcblk1p10    mmcblk1p2     mmcblk1p4     mmcblk1p6     mmcblk1p8
```

2.2.12 M.2 SSD Applications

SSD with M.2 Key M interface NVMe (communication protocol is PCIe x4, the interface is located on the back of the carrier board)



```
nvidia@nx:~$ lspci
0004:00:00.0 PCI bridge: NVIDIA Corporation Device 1ad1 (rev a1)
0004:01:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller (rev 06)
0005:00:00.0 PCI bridge: NVIDIA Corporation Device 1ad0 (rev a1)
0005:01:00.0 PCI bridge: Pericom Semiconductor PI7C9X2G608GP PCIe2 6-Port/8-Lane Packet Switch
0005:02:01.0 PCI bridge: Pericom Semiconductor PI7C9X2G608GP PCIe2 6-Port/8-Lane Packet Switch
0005:02:02.0 PCI bridge: Pericom Semiconductor PI7C9X2G608GP PCIe2 6-Port/8-Lane Packet Switch
0005:02:03.0 PCI bridge: Pericom Semiconductor PI7C9X2G608GP PCIe2 6-Port/8-Lane Packet Switch
0005:02:04.0 PCI bridge: Pericom Semiconductor PI7C9X2G608GP PCIe2 6-Port/8-Lane Packet Switch
0005:04:00.0 Non-Volatile memory controller: Device 1e95:9100 (rev 03)
```



```
nvidia@nvidia:~$ lspci
0004:00:00.0 PCI bridge: NVIDIA Corporation Device 1ad1 (rev a1)
0004:01:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller (rev 06)
0005:00:00.0 PCI bridge: NVIDIA Corporation Device 1ad0 (rev a1)
0005:01:00.0 Non-Volatile memory controller: Device 1e95:9100 (rev 03)
nvidia@nvidia:~$ sudo fdisk -l /dev/nvme*
fdisk: cannot open /dev/nvme0: Illegal seek


Disk /dev/nvme0n1: 119.2 GiB, 128035676160 bytes, 250069680 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: BAAC9E37-3F9E-4433-B18A-990F8E381458

Device          Start       End   Sectors   Size Type
/dev/nvme0n1p1   2048 250068991 250066944 119.2G Linux filesystem


Disk /dev/nvme0n1p1: 119.2 GiB, 128034275328 bytes, 250066944 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Set the M.2 Key M SSD as the file system disk

This solution is a third-party solution and its percentage stability is not guaranteed. The NVMe SSD drive can be used as general storage space or as a file system disk (rootfs and user area), i.e. the system boots up through EMMC (the DTB and kernel Image files are still stored in EMMC) after booting the kernel, file system, installed third-party tools, and user data The default storage is on the SSD.

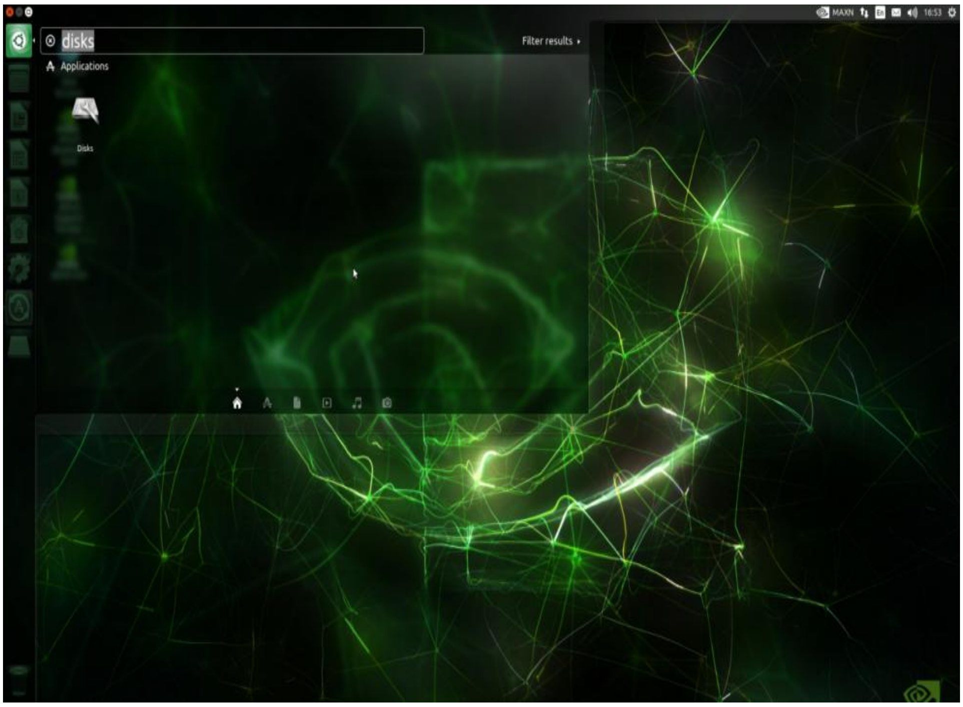
Step 1: Prepare the SSD and format it to GPT format

（1）Prepare M.2 Key M SSD (2280 size specification)

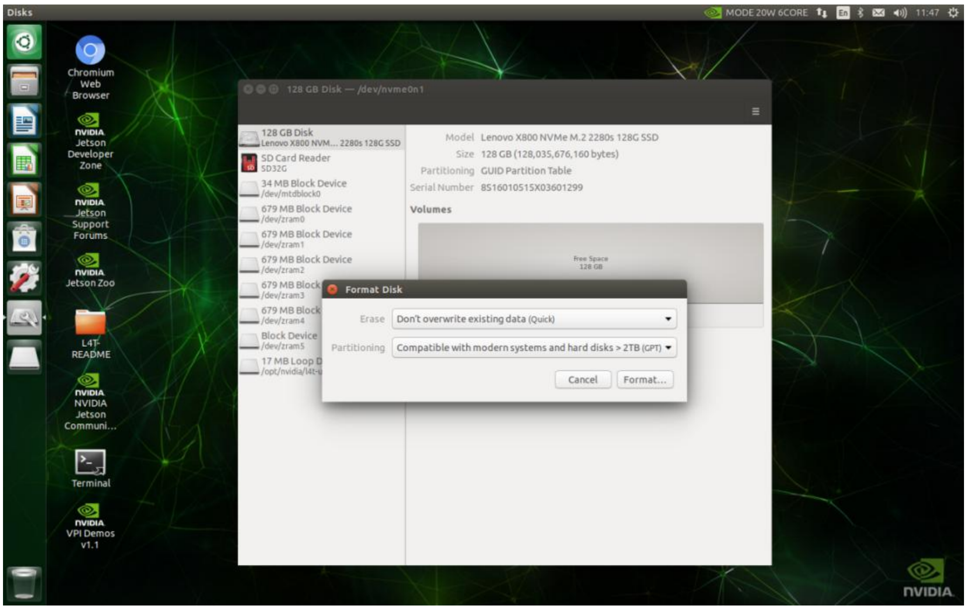
（2）Open the Disks tool that comes with Ubuntu 18.04, find the SSD hard drive you installed, and first press "Ctrl+F" to quickly format it.

（3）Operation refer to the following diagram (follow the following way to carry out the relevant operations, to avoid causing errors that lead to the device can not enter the desktop system)

a. Open the Disks tool at



b. Go to Disks, find the installed SSD, and then combine the keys "CTRL+F" to do a quick format of the drive, click Format



128 GB Disk — /dev/nvme0n1

128 GB Disk
Lenovo X800 NVM... 2280s 128G SSD
SD Card Reader
SD32G
34 MB Block Device
/dev/mtdblock0
679 MB Block Device
/dev/zram0
679 MB Block Device
/dev/zram1
679 MB Block Device
/dev/zram2
679 MB Block
/dev/zram3
679 MB Block
/dev/zram4
Block Device
/dev/zram5
17 MB Loop D
/opt/nvidia/l4t-u

Model Lenovo X800 NVMe M.2 2280s 128G SSD
Size 128 GB (128,035,676,160 bytes)
Partitioning GUID Partition Table
Serial Number 8S16010515X03601299

Volumes

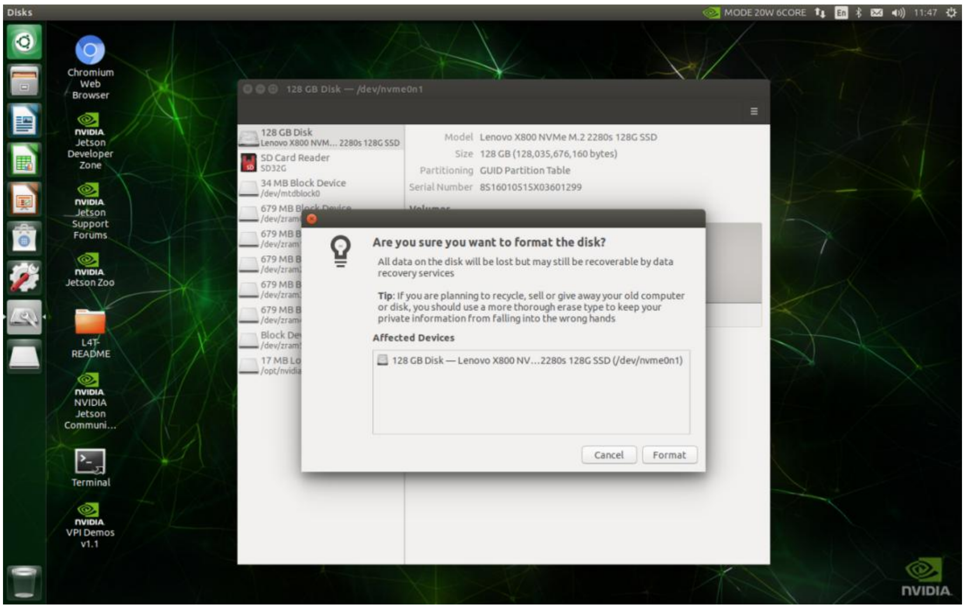
Free Space
128 GB

Format Disk

Erase: Don't overwrite existing data (Quick)
Partitioning: Compatible with modern systems and hard disks > 2TB (GPT)

Cancel | Format...

c. Continue clicking on Format



128 GB Disk — /dev/nvme0n1

Model Lenovo X800 NVMe M.2 2280s 128G SSD
Size 128 GB (128,035,676,160 bytes)
Partitioning GUID Partition Table
Serial Number 8S16010515X03601299

Are you sure you want to format the disk?

All data on the disk will be lost but may still be recoverable by data recovery services

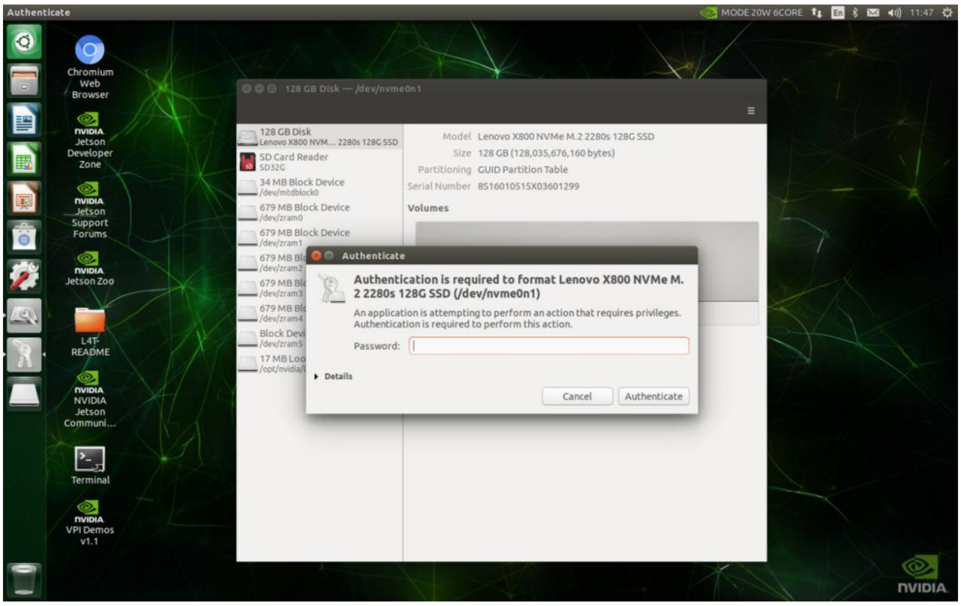
Tip: If you are planning to recycle, sell or give away your old computer or disk, you should use a more thorough erase type to keep your private information from falling into the wrong hands

Affected Devices

128 GB Disk — Lenovo X800 NV...2280s 128G SSD (/dev/nvme0n1)

Cancel | Format

d. Enter the NX user password (case sensitive)



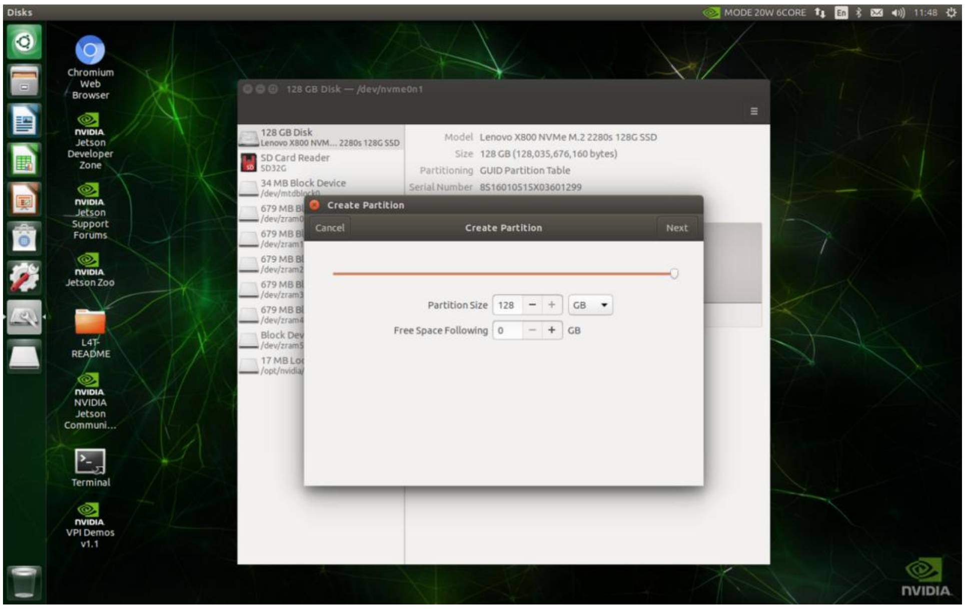
Model Lenovo X800 NVMe M.2 2280s 128G SSD
Size 128 GB (128,035,676,160 bytes)
Partitioning GUID Partition Table
Serial Number 8S16010515X03601299

Authentication is required to format Lenovo X800 NVMe M.2 2280s 128G SSD (/dev/nvme0n1)

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

Password:

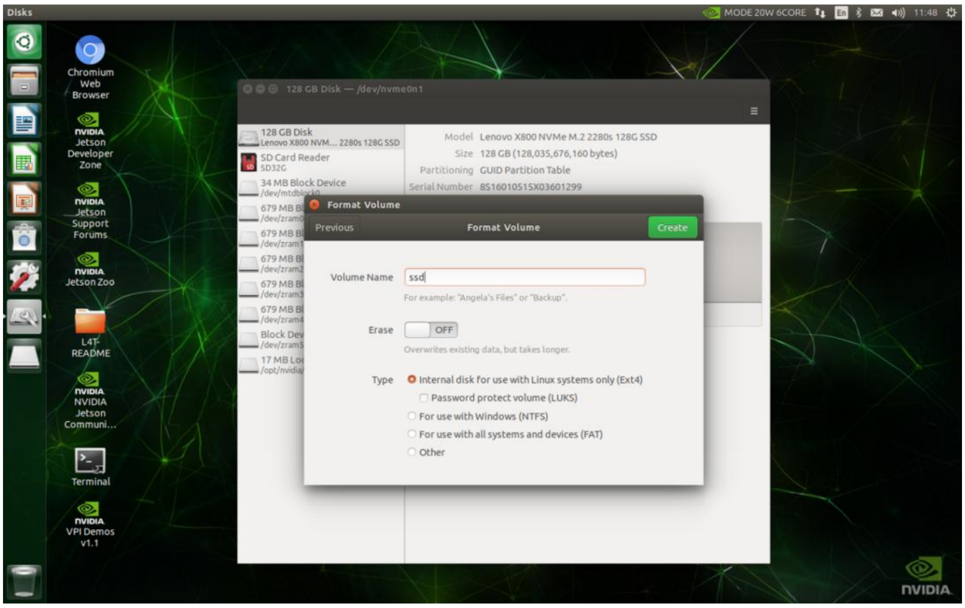
e. You can divide the size of the ratio yourself, the following figure is the default maximum partition, click next



Model Lenovo X800 NVMe M.2 2280s 128G SSD
Size 128 GB (128,035,676,160 bytes)
Partitioning GUID Partition Table
Serial Number 8S16010515X03601299

Partition Size 128 GB
Free Space Following 0 GB

f. Give the partition a name, leave all other options as default and click create



128 GB Disk — /dev/nvme0n1

128 GB Disk
Lenovo X800 NVM... 2280s 128G SSD

SD Card Reader
SD32G

34 MB Block Device
/dev/mtdblock0

679 MB Bl
/dev/zram0

679 MB Bl
/dev/zram1

679 MB Bl
/dev/zram2

679 MB Bl
/dev/zram3

679 MB Bl
/dev/zram4

Block Dev
/dev/zram5

17 MB Lo
/opt/nvidia/

Model Lenovo X800 NVMe M.2 2280s 128G SSD
Size 128 GB (128,035,676,160 bytes)
Partitioning GUID Partition Table
Serial Number 8S16010515X03601299

Format Volume

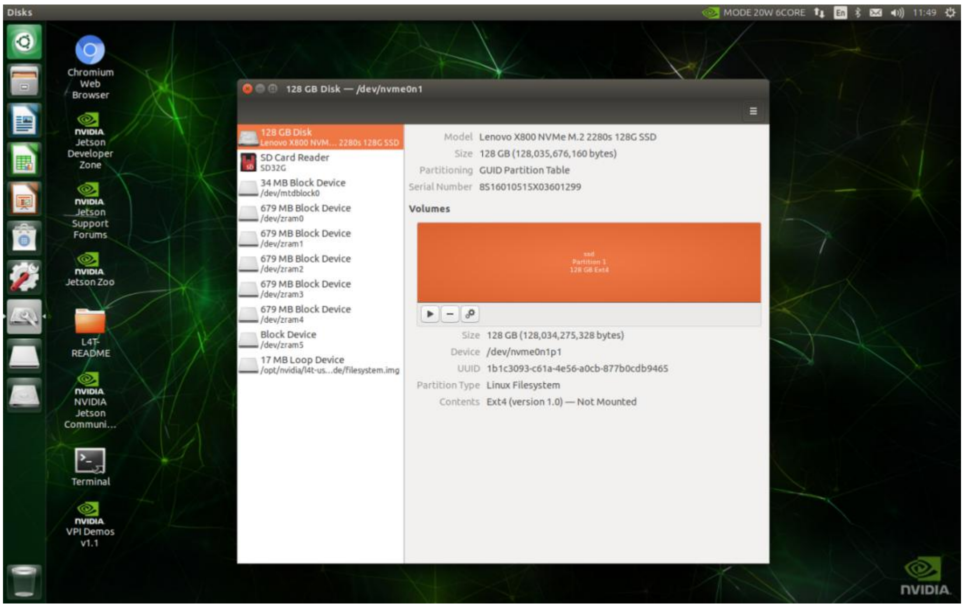
Previous Format Volume Create

Volume Name ssd
For example: "Angela's Files" or "Backup".

Erase OFF
Overwrites existing data, but takes longer.

Type
● Internal disk for use with Linux systems only (Ext4)
□ Password protect volume (LUKS)
○ For use with Windows (NTFS)
○ For use with all systems and devices (FAT)
○ Other

g. The screen shown below appears, click on the triangle symbol to mount



128 GB Disk — /dev/nvme0n1

128 GB Disk
Lenovo X800 NVM... 2280s 128G SSD

SD Card Reader
SD32G

34 MB Block Device
/dev/mtdblock0

679 MB Block Device
/dev/zram0

679 MB Block Device
/dev/zram1

679 MB Block Device
/dev/zram2

679 MB Block Device
/dev/zram3

679 MB Block Device
/dev/zram4

Block Device
/dev/zram5

17 MB Loop Device
/opt/nvidia/l4t-us...de/filesystem.img

Model Lenovo X800 NVMe M.2 2280s 128G SSD
Size 128 GB (128,035,676,160 bytes)
Partitioning GUID Partition Table
Serial Number 8S16010515X03601299

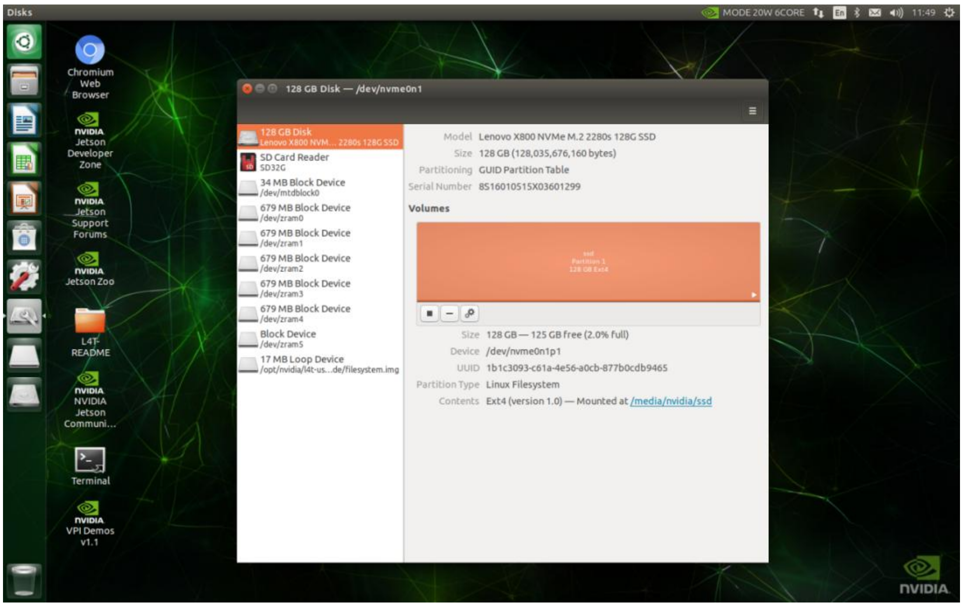
Volumes

ssd
Partition 1
128 GB Ext4

▶ −

Size 128 GB (128,034,275,328 bytes)
Device /dev/nvme0n1p1
UUID 1b1c3093-c61a-4e56-a0cb-877b0cdb9465
Partition Type Linux Filesystem
Contents Ext4 (version 1.0) — Not Mounted

h. The image below shows that the mount is successful



128 GB Disk — /dev/nvme0n1

128 GB Disk
Lenovo X800 NVM... 2280s 128G SSD

SD Card Reader
SD32G

34 MB Block Device
/dev/mtdblock0

679 MB Block Device
/dev/zram0

679 MB Block Device
/dev/zram1

679 MB Block Device
/dev/zram2

679 MB Block Device
/dev/zram3

679 MB Block Device
/dev/zram4

Block Device
/dev/zram5

17 MB Loop Device
/opt/nvidia/l4t-us...de/filesystem.img

Model Lenovo X800 NVMe M.2 2280s 128G SSD
Size 128 GB (128,035,676,160 bytes)
Partitioning GUID Partition Table
Serial Number 8S16010515X03601299

Volumes

Size 128 GB — 125 GB free (2.0% full)
Device /dev/nvme0n1p1
UUID 1b1c3093-c61a-4e56-a0cb-877b0cdb9465
Partition Type Linux Filesystem
Contents Ext4 (version 1.0) — Mounted at /media/nvidia/ssd

Step 2 : Convert system disk to EMMC built-in system

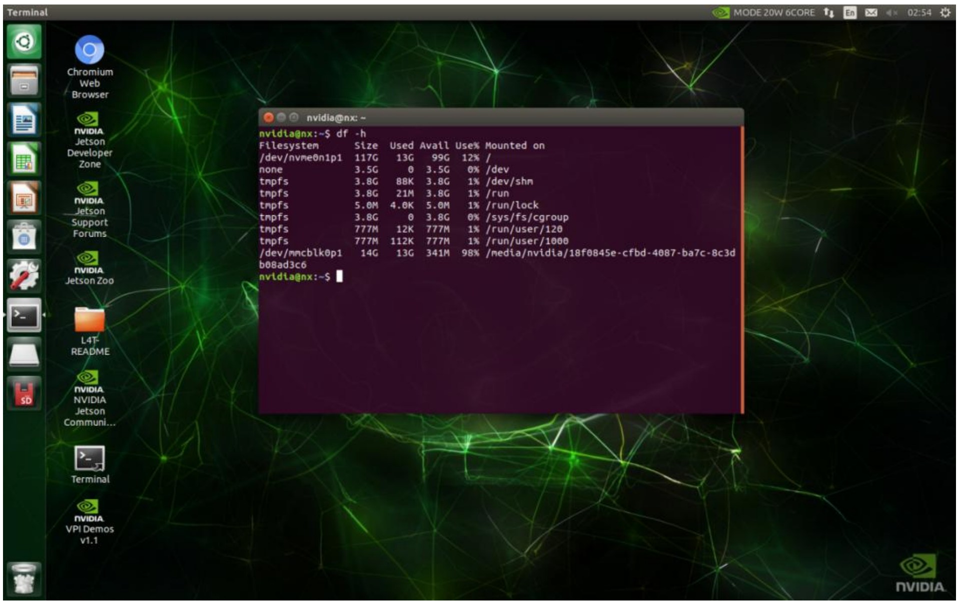
```
$ git clone    https://github.com/jetsonhacks/rootOnNVMe.git
```

Note: If git fails, you can go to the website and download the zip file and copy it to the device to unzip it

```
$ cd rootOnNVMe
$ . /copy-rootfs-SSD.sh (this takes a little longer, please be patient and wait for the execution to finish before executing the next step)
$ . /setup-service.sh
$ sudo reboot
```

Step 3: Inspection

Reboot the machine, open a terminal and type df -h to see if the hard disk device /dev/nvme0n1p1 is mounted on the /root directory, if not, re-run



```
nvidia@nx:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme0n1p1  117G   13G   99G  12% /
none            3.5G     0  3.5G   0% /dev
tmpfs           3.8G   88K  3.8G   1% /dev/shm
tmpfs           3.8G   21M  3.8G   1% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           3.8G     0  3.8G   0% /sys/fs/cgroup
tmpfs           777M   12K  777M   1% /run/user/120
tmpfs           777M  112K  777M   1% /run/user/1000
/dev/mmcblk0p1   14G   13G  341M  98% /media/nvidia/18f0845e-cfbd-4087-ba7c-8c3d
b08ad3c6
nvidia@nx:~$ 
```

2.2.13 GMSL Camera Application

GMSL camera application interface is based on V4L2 (abbreviation of Video for Linux, a driver framework for Linux video capture related devices), which provides a set of unified interface specifications for drivers and applications to facilitate application development.

GMSL Physical Interface Features

(1) Support the signal transmission of up to 15 meters coaxial cable, which can be extended by using Moriyun repeater.

(2) support different serializer camera, support output resolution of 720p, 1080p, 4K, 8M and other resolutions of the camera

Noun	Explanation	Remarks
Self-triggered	The camera is not controlled by the trigger signal, and the camera outputs images after powering up	This mode is generally supported by cameras. As long as the external does not give the trigger signal is working in this mode.
Synchronous Trigger	Refers to a camera controlled by a trigger signal that outputs an	Camera firmware support is required. Please check with the manufacturer for

	image and simultaneously shutters when triggered.	support when purchasing the camera. It also requires an external trigger signal.

GMSL camera driver installation and lighting

For a list of supported cameras and driver installation dots, please contact sales to obtain them.

☎ Contact us

SZ Sensing TECH.,LTD

8/F, Block B, Building 1, Wisdom Home, Baolong
Avenue, Baolong Street, Longgang District, Shenzhen

Tel/Phone. +86-755-28990915

Mail/Email: Sales@sensing-world.com