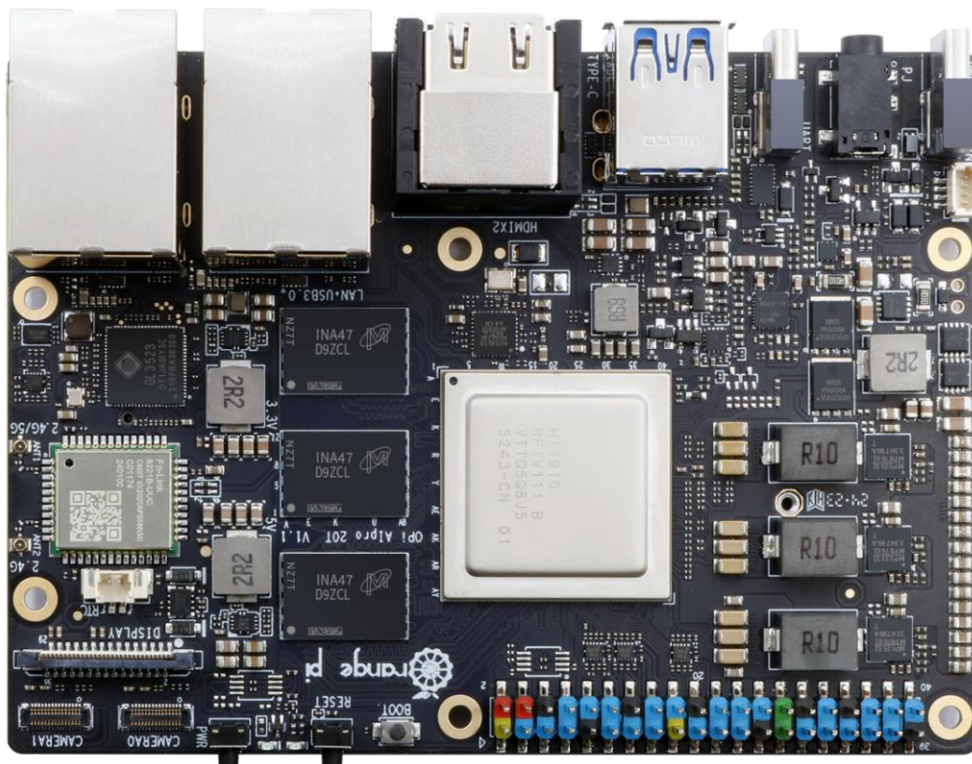




Orange Pi AI Pro 20T

User Manual





Catalogue

1. Introduction to Development Board Parameters	1
1. 1. Introduction to Development Board	1
1. 2. Hardware specifications of the development board	1
1. 3. Top and bottom level views of the development board	3
1. 4. Interface details diagram of development board	4
2. Introduction to using the development board	5
2. 1. Prepare the necessary accessories	5
2. 2. Download the image of the development board and related materials	10
2. 3. Instructions for using the three dial switches to control the startup device	11
2. 4. Method of burning Linux images to TF cards	12
2. 4. 1. Method of burning Linux image to TF card based on Windows PC	12
2. 4. 2. Method of burning Linux images to TF cards based on Ubuntu PC	20
2. 5. Method of burning Linux images to eMMC	24
2. 6. Method of burning Linux images to NVMe SSDs	30
2. 7. How to Flash a Linux Image to a SATA SSD	36
2. 8. Steps to Boot the Development Board	42
2. 9. Debugging the use of serial ports	44
2. 9. 1. Connection instructions for using the debug serial port via Type-C USB interface	45
2. 9. 2. Connection instructions for using the debug serial port via uart0 in the 40 pin interface	45
2. 9. 3. Ubuntu Platform Debugging Serial Port Usage	47
2. 9. 4. Windows platform debugging serial port usage	51
2. 10. Notes on the use of WIFI Bluetooth antennas	54
3. Instructions for using the Ubuntu Xfce desktop system	56
3. 1. Supported Ubuntu image types and kernel versions	56



3. 2.	Adaptation of Linux system functions	56
3. 3.	Linux system login instructions	58
3. 3. 1.	Logging in to the desktop of a Linux system	58
3. 3. 2.	Default login and password for Linux systems	59
3. 4.	On-Board LED Test Description	60
3. 5.	Network connectivity testing	60
3. 5. 1.	Ethernet port test	60
3. 5. 2.	WIFI connection test	62
3. 5. 3.	Methods for setting static IP addresses	68
3. 6.	SSH remote login development board	75
3. 6. 1.	SSH remote login development board under Ubuntu	75
3. 6. 2.	SSH remote login development board under Windows	76
3. 7.	HDMI interface usage instructions	78
3. 7. 1.	Instructions for displaying Linux desktop via HDMI	78
3. 7. 2.	How to use the HDMI interface to display images and play audio	78
3. 8.	How to use Bluetooth	80
3. 9.	USB interface test	82
3. 9. 1.	Type-C USB3.0 interface Host mode usage instructions	82
3. 9. 2.	Instructions for using the Type-C USB3.0 interface in Device mode ...	83
3. 9. 3.	Connect a USB mouse or keyboard to test	85
3. 9. 4.	USB camera test	86
3. 9. 5.	USB Audio Test	87
3. 10.	Audio Test	89
3. 10. 1.	Headphone jack audio playback test	89
3. 10. 2.	HDMI audio playback test	90
3. 10. 3.	Headphone MIC recording test	90
3. 11.	40 Pin interface pin function description	91
3. 12.	40 pin interface GPIO, I2C, UART, SPI, PWM and CAN test	93
3. 12. 1.	40 pin GPIO port test method	93
3. 12. 2.	40 pin SPI loopback test	96
3. 12. 3.	40 Pin I2C Test	97



3. 12. 4.	40 pin UART test	99
3. 12. 5.	40 pin PWM test	101
3. 12. 6.	Test method for 40 pin CAN	103
3. 13.	Installation and usage methods of wiringOP	111
3. 13. 1.	Method for installing wiringOP	111
3. 13. 2.	Method of using wiringOP to control 40pin GPIO	112
3. 14.	WiringOP hardware PWM usage method	114
3. 14. 1.	Method of setting PWM using the gpio command of WiringOP	115
3. 14. 2.	Usage of PWM Test Program	120
3. 15.	wiringOP-Python installation and usage instructions	121
3. 15. 1.	wiringOP-Python installation method	122
3. 15. 2.	40pin GPIO Port Test	124
3. 15. 3.	40pin SPI test	126
3. 15. 4.	40pin I2C Test	128
3. 15. 5.	40pin UART Test	131
3. 16.	How to upload files to the Linux system of the development board	133
3. 16. 1.	How to upload files from Ubuntu PC to the Linux system of the development board	133
3. 16. 2.	How to upload files from Windows PC to the Linux system of the development board	136
3. 17.	How to use the cooling fan	141
3. 18.	Setting method of AI CPU and control CPU	143
3. 19.	How to set swap memory	144
3. 20.	Methods for testing MindSpore	145
3. 21.	Using ascend hardware accelerated ffmpeg	146
3. 21. 1.	Using compiled deb software packages	146
3. 21. 2.	Building from Source Code	148
3. 21. 3.	Application Scenarios	152
3. 22.	Method for installing kernel header files	160
3. 23.	Method of installing ZFS	161
3. 24.	How to shut down and restart the development board	164



4. Experience AI application examples (openEuler)	165
4. 1. Login to Jupyter Lab	165
4. 2. Methods for freeing memory	167
4. 3. Running object detection samples	168
4. 4. Running Text Recognition Sample	171
4. 5. Running Target Classification Example	173
4. 6. Running Image Exposure Enhancement Sample	176
4. 7. Running Image Style Migration Sample	178
4. 8. Running Image Classification Sample	181
4. 9. Running FCN Image Semantic Segmentation Sample	183
4. 10. Running Implementation Image Conversion Sample	186
5. Experience AI application examples (Ubuntu)	189
5. 1. Login to jupyter Lab	189
5. 2. Methods for freeing memory	192
5. 3. Methods for Running Online Reasoning Cases	193
5. 3. 1. Running a Simple Deep Learning Model	193
5. 3. 2. Running ResNet50 Image Classification Sample	196
5. 3. 3. Running Vision Transformer Image Classification Sample	198
5. 3. 4. Running FCN Image Semantic Segmentation Sample	202
5. 3. 5. Running ShuffleNet Image Classification Sample	204
5. 3. 6. Running SSD Object Detection Sample	207
5. 3. 7. Running RNN to Implement Emotion Classification Example	209
5. 3. 8. Running LSTM+CRF Sequence Annotation Sample	211
5. 3. 9. Running GAN Image Generation Sample	213
5. 3. 10. Running DCGAN to Generate Manga Avatar Sample	216
5. 3. 11. Running Pix2Pix to Implement Image Conversion Sample	219
5. 3. 12. Example of Running Diffusion Model	221
5. 3. 13. Running ResNet50 Transfer Learning Sample	223
5. 4. Method for Running llm Large Language Model	226
5. 4. 1. qwen1.5-0.5b	227



5. 4. 2. Tinyllama	229
5. 4. 3. DeepSeek-R1-Distill-Qwen-1.5B	231
5. 5. Methods for Running Offline Reasoning Cases	233
6. Installation of AI Application Environment (OpenHarmony)	234
6. 1. Installation of inference environment	234
6. 2. Python Environment Installation	238
6. 3. Installing the toolkit package	238
6. 4. Kernels Environment Installation	242
7. MindSDK User Guide	243
7. 1. Vision SDK Visual Analysis	243
7. 1. 1. Installation and Deployment	243
7. 1. 2. Usage	244
8. Instructions for using the Linux kernel source package	249
8. 1. Requirements for the compilation host system	249
8. 2. Install the cross-compilation toolchain and dependent packages	251
8. 3. Download and decompress the Linux kernel source package	253
8. 4. How to compile and validate kernel image files	254
8. 5. How to compile and validate kernel DTB files	256
9. Instructions for using the Linux image compilation script	257
9. 1. Requirements for the compilation host system	257
9. 2. Things you need to prepare to make a Linux image	259
9. 3. Download the source code package of the Linux image compilation script	259
9. 4. How to create a minimal image	261
9. 5. How to create a complete image	264
9. 6. Methods for making compression and expansion images	266
10. Appendix	269
10. 1. User Manual Update History	269
10. 2. Image Update History	270



1. Introduction to Development Board Parameters

1.1. Introduction to Development Board

The Orange Pi AI Pro 20T development board is a high-performance AI development board meticulously crafted by Orange Pi in collaboration with Huawei. It is equipped with a Ascend AI processor and can provide 20TOPS INT8 computing power. The memory is available in two versions: 12GB and 24GB. It can achieve various data analysis and inference calculations such as images and videos, and can be widely used in scenarios such as education, robotics, and drones.

1.2. Hardware specifications of the development board

Orange Pi AI Pro 20T Development Board Hardware Specifications	
Ascend AI processor	4-core 64 bit Arm processor+AI processor
AI computing power	• Semi precision (FP16) : 10 TFLOPS • Integer precision (INT8) : 20 TOPS
Memory	• type: LPDDR4X • capacity: 12GB or 24GB
storage	• 32MB SPI Flash onboard • Micro SD card slot • EMMC socket: can be connected to external eMMC modules • M. 2 M-Key interface: can connect to NVMe SSD or SATA SSD with 2280 specifications
Ethernet	2 PCIe 2.5G Ethernet ports (RTL8125BG)
Wi-Fi+ Bluetooth	• Supports 2.4G and 5G dual band WIFI • BT4.2 • Module: Ouzhitong 6221BUUC
USB	• 3 USB 3.0 Host interfaces • 1 Type-C USB3.0 OTG interface

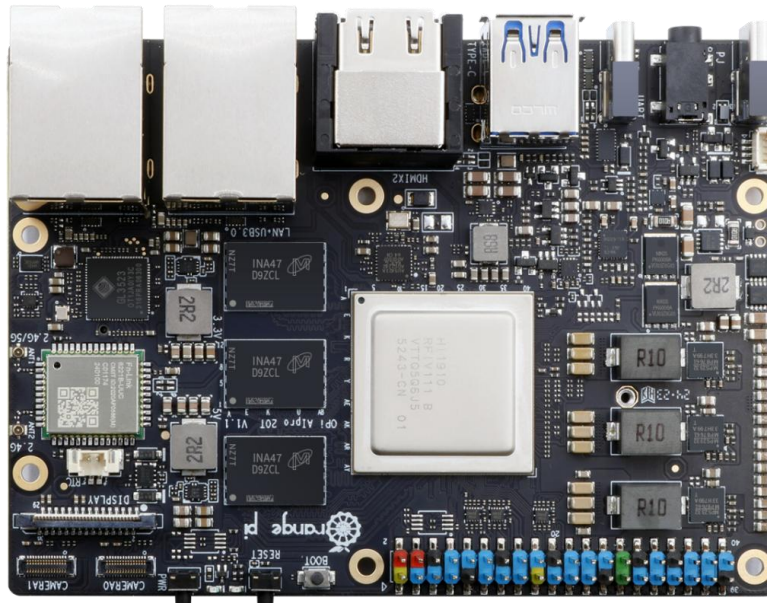


camera	Two MIPI CSI 4 Lane interfaces
display	• 2 HDMI interfaces • 1 MIPI DSI 4 Lane interface
audio frequency	• 1 3.5mm headphone jack, supporting audio input and output • 2 HDMI audio outputs
40 pin expansion port	Used to expand interfaces such as UART, I2C, SPI, PWM, and GPIO
Keys	1 reset button, 1 shutdown button, 1 upgrade button
DIP switch	Used to control SD card, eMMC, and SSD boot options
source	Supports Type-C power supply, 20V PD-65W adapter
LED lights	1 power indicator light and 1 software controllable indicator light
Fan interface	4pin, 0.8mm spacing, used for connecting to 12V fan, supporting PWM control
Battery interface	2 pins, 2.54mm spacing, used for connecting 3 series of batteries, supporting fast charging
RTC	2 pins, 1.25mm spacing, used for connecting RTC batteries
Debug UART	Debugging serial port for Micro USB interface
Supported operating systems	Ubuntu 22.04 and openEuler 22.03
Introduction to appearance specifications	
Product size	115*83mm
weight	120g
 rangePi™ is a registered trademark of Shenzhen Xunlong Software Co., Ltd	

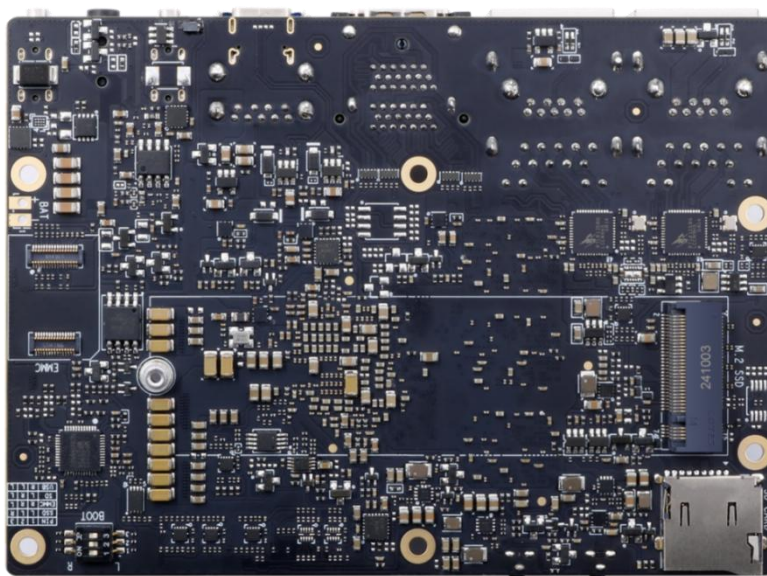


1. 3. Top and bottom level views of the development board

Top view:



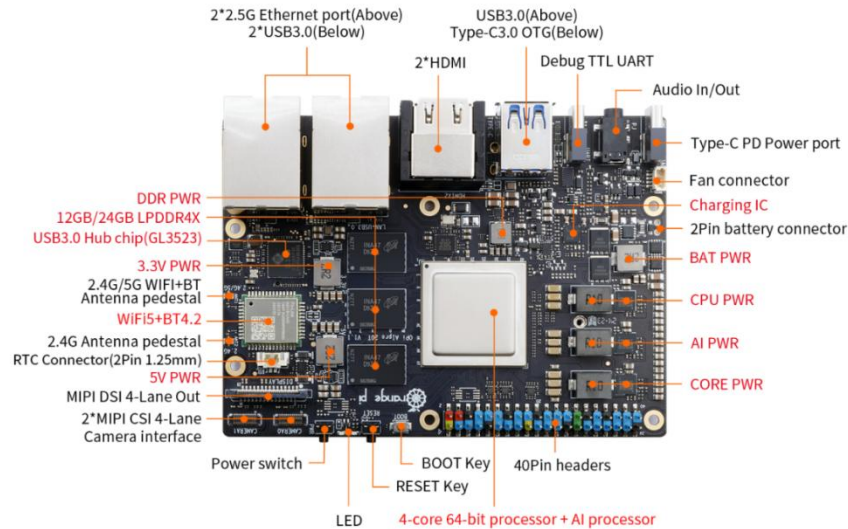
Bottom view:



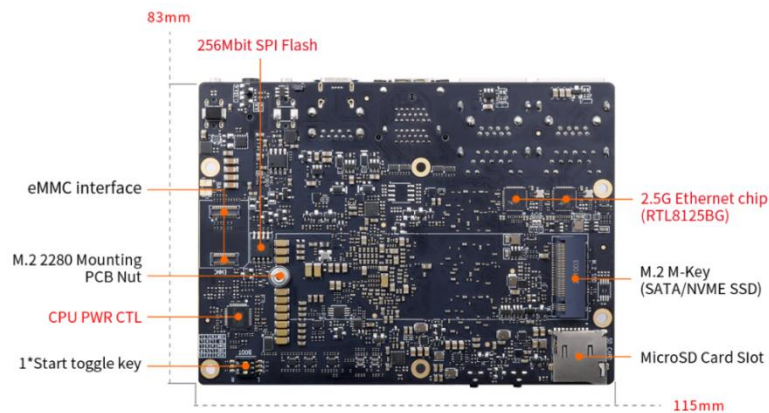


1. 4. Interface details diagram of development board

Product display



Top View



Bottom View



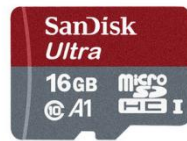
2. Introduction to using the development board

2. 1. Prepare the necessary accessories

1) TF card, a class 10 or higher high-speed flash card with a minimum capacity of 32GB.

We strongly recommend using a TF card with a capacity of 64GB or higher. 小 32GB

SanDisk 闪迪



2) TF card reader, used for reading and writing TF cards.



3) HDMI to HDMI connection cable, used to connect the development board to an HDMI monitor or TV for display.



4) Type-C to USB3.0 adapter cable, used for Type-C interface connection to USB3.0 storage devices.



5) Type-C interface data cable, used for debugging serial port, Type-C USB interface Device and other functions.



6) 10.1-inch MIPI screen (same as the 10.1-inch LCD screen used in the RK3588 series development board)



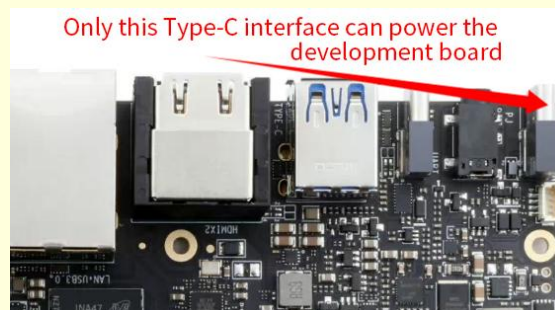
7) HDMI interface display



8) Power supply, Type-C interface 20V PD-65W adapter.



There are three Type-C interfaces on the development board, and the one near the PWM fan interface is the PD power interface. The other two Type-C interfaces cannot provide power to the development board. Please do not connect them incorrectly.



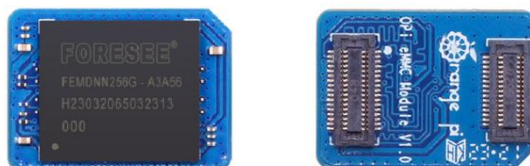
9) M.2 M-Key 2280 malformed PCIe NVMe SSD.



10) M.2 M-Key 2280 specification SATA SSD.



11) eMMC Module



12) Mouse and keyboard with USB interface.



13) USB camera.

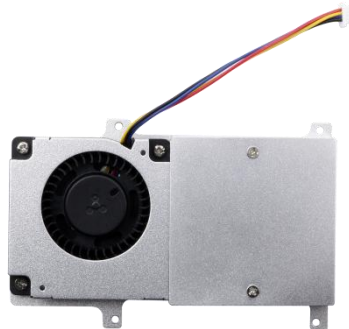


14) Matched metal shell.

15) Ethernet cable, used to connect the development board to the Internet.



16) 12V PWM cooling fan.



The position of the PWM fan interface on the development board is shown in the following figure:





17) RTC battery, with a 2-pin interface and a 1.25mm spacing.



The location of the RTC battery interface on the development board is shown in the following figure:



18) An X64 computer with Ubuntu 22.04 and Windows operating system installed.

1	Ubuntu22.04 PC	Optional, used to compile Linux source code
2	Windows PC	Used to burn Ubuntu and OpenEuler images

2.2. Download the image of the development board and related materials

The link to the development board data download page is as follows:

[http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro\(20t\).html](http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro(20t).html)



Installation information for
shell and radiator

[Downloads](#)



User Manual

[Downloads](#)



Schematic

[Downloads](#)



Official Tools

[Downloads](#)



Mechanical

[Downloads](#)



Linux Source code

[Downloads](#)

Official Images



Ubuntu Image

[Downloads](#)



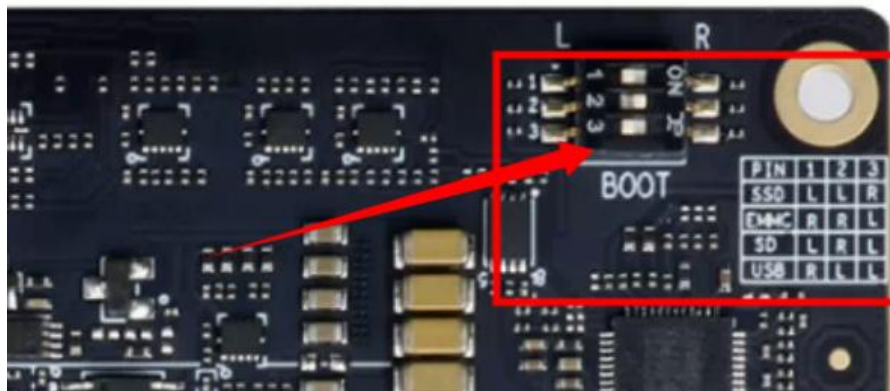
openEuler

openeuler

[Downloads](#)

2.3. Instructions for using the three dial switches to control the startup device

The Linux system of the development board supports booting from TF cards, eMMC, and SSDs (supports NVMe SSDs and SATA SSDs), **but USB booting is not supported**. The specific device to boot from is controlled by the three dial switches on the back of the development board.



All three dial switches support two setting states, left and right, so there are a total of eight setting states. Currently, the development board only uses three of them. The startup devices corresponding to different settings are shown in the table below:

DIP switch 1	DIP switch 2	DIP switch 3	Corresponding startup device
Left	Left	Right	SATA SSD 和 NVMe SSD
Right	Right	Left	eMMC
Left	Right	Left	TF card

Note that the setting status of the DIP switch corresponding to the boot mode of SATA SSD and NVMe SSD is the same. These two startup methods are automatically distinguished by the level of the M2-TYPE pin.

Also, please note that after switching the dial switch, you must unplug and unplug the power again to power on and off for the new boot device option to take effect. Resetting the system through the reset button on the development board will not allow the new configuration of the dial switch to take effect.

2. 4. Method of burning Linux images to TF cards

2. 4. 1. Method of burning Linux image to TF card based on Windows PC

Note that the Linux image referred to here specifically refers to the Ubuntu or openEuler image downloaded from the data download page of the development board.



2.4.1.1. Method of burning Linux images using Win32Diskimager

1) Firstly, prepare a TF card with a capacity of 32GB or larger (**it is recommended to use a TF card with a capacity of 64GB or higher**). The transfer speed of the TF card must be at or above **level 10**. It is recommended to use TF cards from brands such as SanDisk.

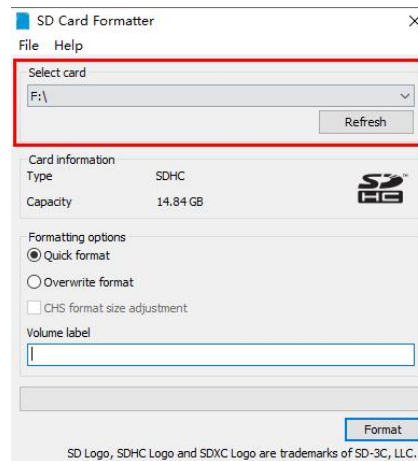
2) Then insert the TF card into the card reader, and then insert the card reader into the computer.

3) Then format the TF card.

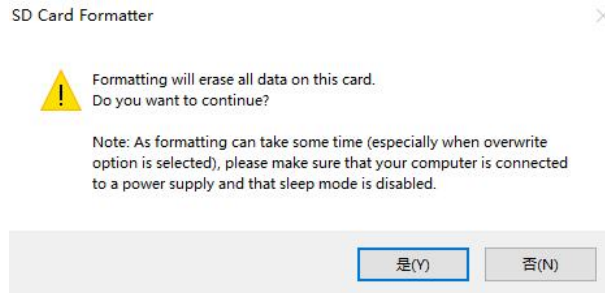
- a. You can use the **SD Card Formatter** software to format TF cards, and its download link is

https://www.sdcard.org/downloads/formatter/eula_windows/SDCardFormatterv5_WinEN.zip

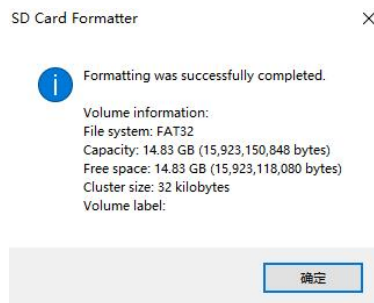
- b. After downloading, simply unzip and install it, and then open the software.
- c. If the computer only has a TF card inserted, the "**Select card**" column will display the drive letter of the TF card. If the computer has multiple USB storage devices inserted, you can select the drive letter corresponding to the TF card through the drop-down menu.



- d. Then click on "**Format**", a warning box will pop up before formatting, and selecting "**Yes (Y)**" will start formatting.



- e. After formatting the TF card, the message shown in the following image will pop up. Click OK to proceed.



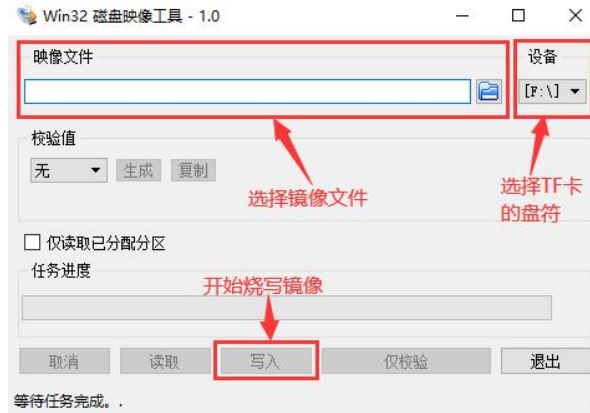
4) Download the compressed file of the Linux operating system image that you want to burn from [the download page of the development board](#), and then use decompression software to decompress it. In the decompressed file, the file ending with ".img" is the operating system image file.

5) Burn a Linux image to a TF card using **Win32Diskimager**.

- a. The download page for Win32Diskimager is

<http://sourceforge.net/projects/win32diskimager/files/Archive/>

- b. After downloading, simply install it directly. The interface of Win32Diskimager is shown below.
- First, select the path of the image file.
 - Then confirm that the drive letter of the TF card matches the one displayed in the "**Device**" column.
 - Finally, click "**Write**" to start burning.



- c. After the image writing is completed, click the "Exit" button to exit, and then you can unplug the TF card and insert it into the development board to start.

After the burning is completed, if the system pops up a Windows prompt window, this is a normal phenomenon and not a burning error. Please select 'Cancel' at this time, do not select 'Format Disk', otherwise the burned image will be formatted.



Attention, please ensure that the dip switch is turned to the TF card start position before starting the system. Please refer to the section on the instructions for using the two dip switches that control the starting device for instructions on how to use the dip switch.

2. 4. 1. 2. Method of burning Linux images using balanaEtcher

1) Firstly, prepare a TF card with a capacity of 32GB or larger (**it is recommended to use a TF card with a capacity of 64GB or higher**). The transfer speed of the TF card must be at or above **level 10**. It is recommended to use TF cards from brands such as SanDisk.

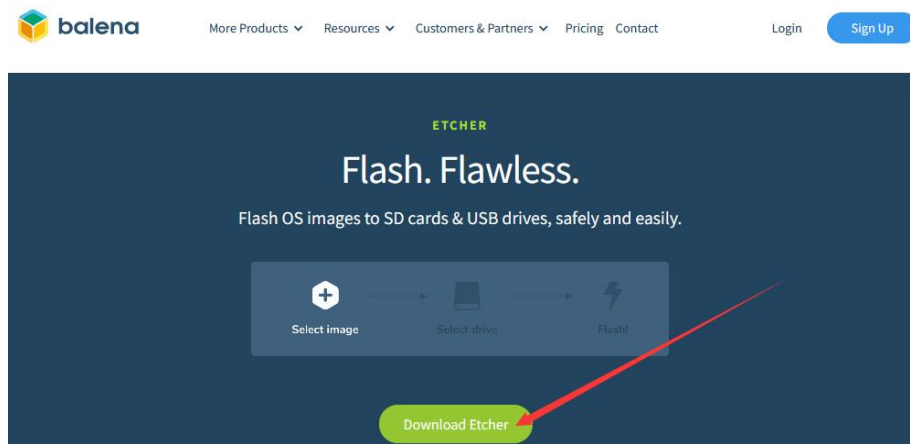


2) Then insert the TF card into the card reader, and then insert the card reader into the computer.

3) Download the Linux image compressed file you want to burn from the [data download page of the development board](#).

4) Then download the software for burning Linux images - **balenaEtcher**, download link:
<https://www.balena.io/etcher/>

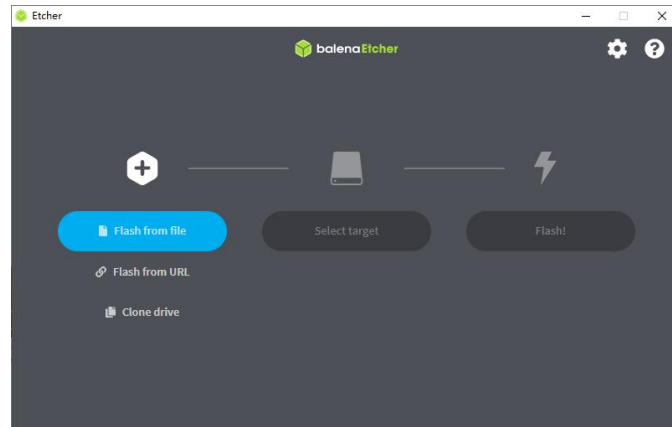
5) After entering the BalenaEtcher download page, clicking the green download button will jump to the software download location.



6) Then you can choose to download the portable version of balenaEtcher, which does not require installation and can be used by double clicking.



7) The balenaEtcher interface after opening is shown in the following figure:



When opening balenaEtcher, if prompted with the following error:

Attention

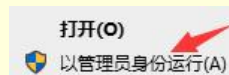
Something went wrong. If it is a compressed image, please check that the archive is not corrupted.

User did not grant permission.

Cancel

Retry

Please select balenaEtcher and right-click, then choose to run as administrator.

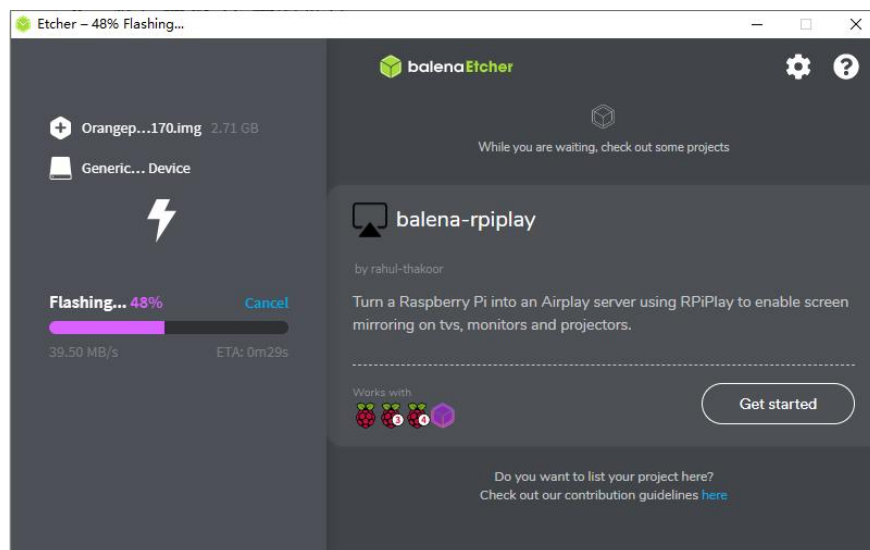


8) The specific steps for burning a Linux image using balenaEtcher are as follows:

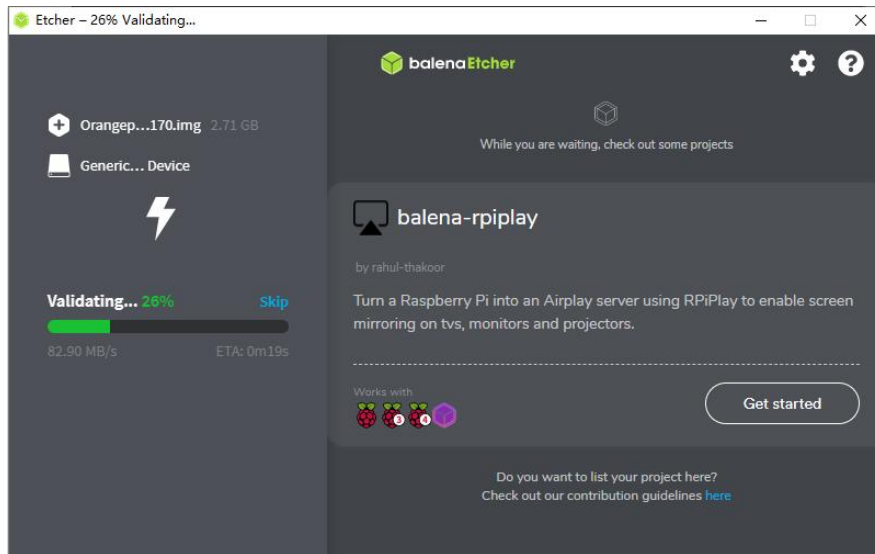
- Firstly, select the path of the Linux image file to be burned.
- Then select the drive letter of the TF card.
- Finally, clicking Flash will start burning the Linux image onto the TF card.



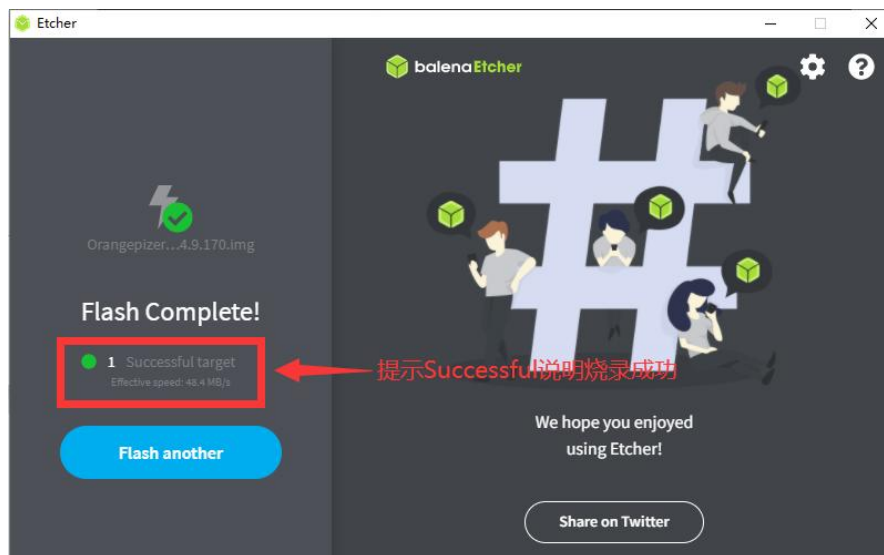
9) The interface displayed during the process of burning a Linux image by balenaEtcher is shown in the following figure. In addition, the progress bar displaying purple indicates that the Linux image is being burned to the TF card.



10) After the Linux image is burned, balenaEtcher will also verify the image burned to the TF card by default to ensure that there are no problems during the burning process. As shown in the following figure, a green progress bar indicates that the image has been burned and balenaEtcher is verifying the burned image.



11) After successful burning, the display interface of balenaEtcher is shown in the following figure. If a green indicator icon is displayed, it indicates that the image burning is successful. At this time, you can exit balenaEtcher, then unplug the TF card and insert it into the TF card slot of the development board for use.



After the burning is completed, if the system pops up a Windows prompt window, this is a normal phenomenon and not a burning error. Please select 'Cancel' at this time, do not select 'Format Disk', otherwise the burned image will be formatted.



Attention, please ensure that the dip switch is turned to the TF card start position before starting the system. Please refer to the section on the instructions for [using the three dip switches that control the starting device for instructions on how to use them](#).

2. 4. 2. Method of burning Linux images to TF cards based on Ubuntu PC

Note that the Linux image referred to here specifically refers to the Ubuntu or openEuler image downloaded from the development board's download page.

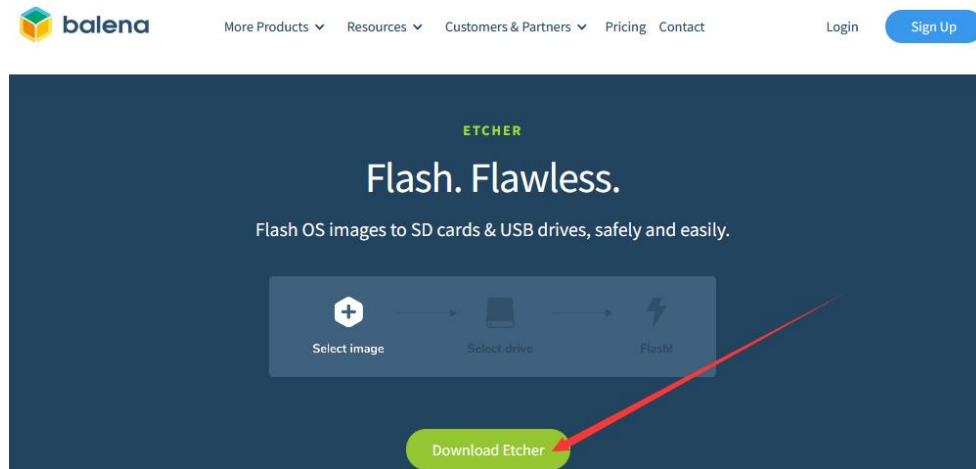
1) Firstly, prepare a TF card with a capacity of 32GB or larger (**it is recommended to use a TF card with a capacity of 64GB or higher**). The transmission speed of the TF card must be **class10 or above**. It is recommended to use TF cards from brands such as Sandisk.

2) Then insert the TF card into the card reader, and then insert the card reader into the computer.

3) Download balenaEtcher software from:

<https://www.balena.io/etcher/>

4) After entering the BalenaEtcher download page, clicking the green download button will redirect you to the software download location.



5) Then choose to download the Linux version of the software.

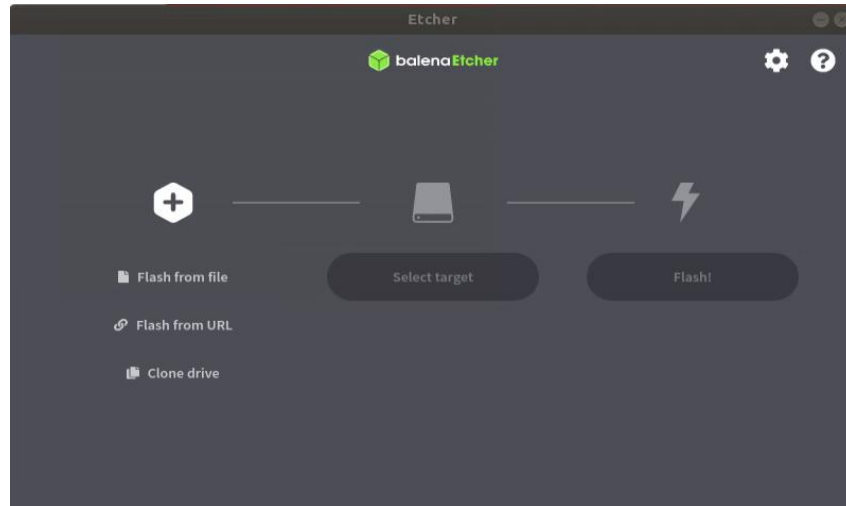
DOWNLOAD

Download Etcher

ASSET	OS	ARCH	
ETCHER FOR WINDOWS (X86 X64) (INSTALLER)	WINDOWS	X86 X64	Download
ETCHER FOR WINDOWS (X86 X64) (PORTABLE)	WINDOWS	X86 X64	Download
ETCHER FOR WINDOWS (LEGACY 32 BIT) (X86 X64) (PORTABLE)	WINDOWS	X86 X64	Download
ETCHER FOR MACOS	MACOS	X64	Download
ETCHER FOR LINUX X64 (64-BIT) (APPIMAGE)	LINUX	X64	Download
ETCHER FOR LINUX (LEGACY 32 BIT) (APPIMAGE)	LINUX	X86	Download

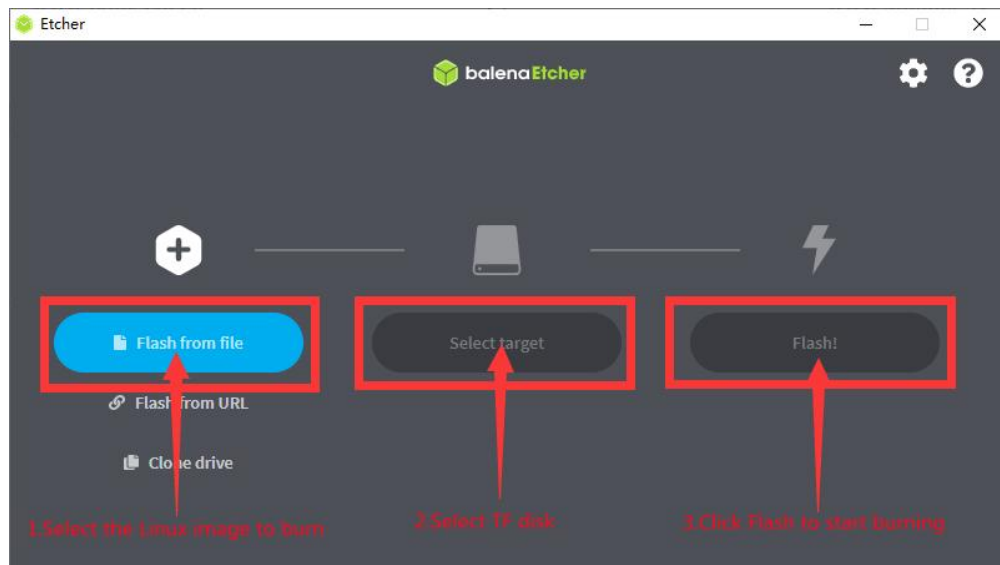
6) Download the Linux image file compression package you want to burn from the [download page of the development board](#).

7) Then double-click balenaEtcher x.x.x-x64.AppImage on the graphical interface of Ubuntu PC to open balenaEtcher. The interface after opening balenaEtcher is shown in the following figure:

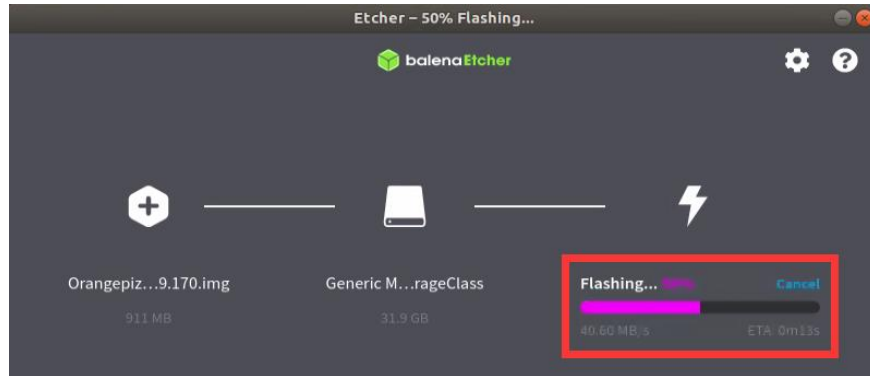


8) The specific steps for burning a Linux image using balenaEtcher are as follows:

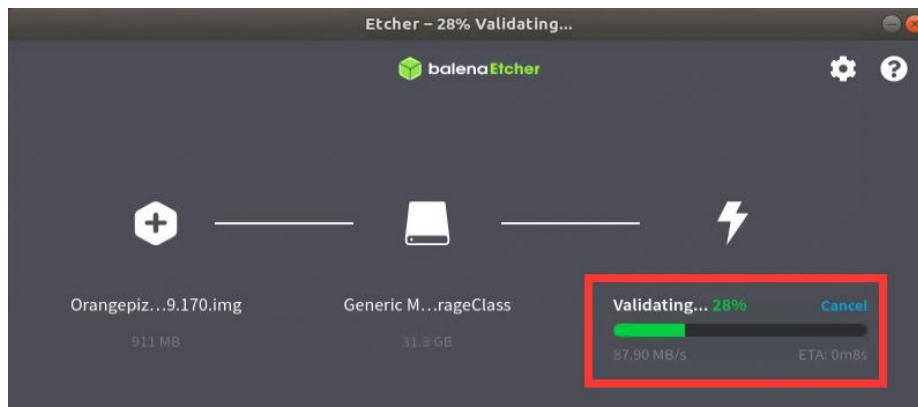
- a. Firstly, select the path of the Linux image file to be burned.
- b. Then select the drive letter of the TF card.
- c. Finally, clicking Flash will start burning the Linux image to the TF card.



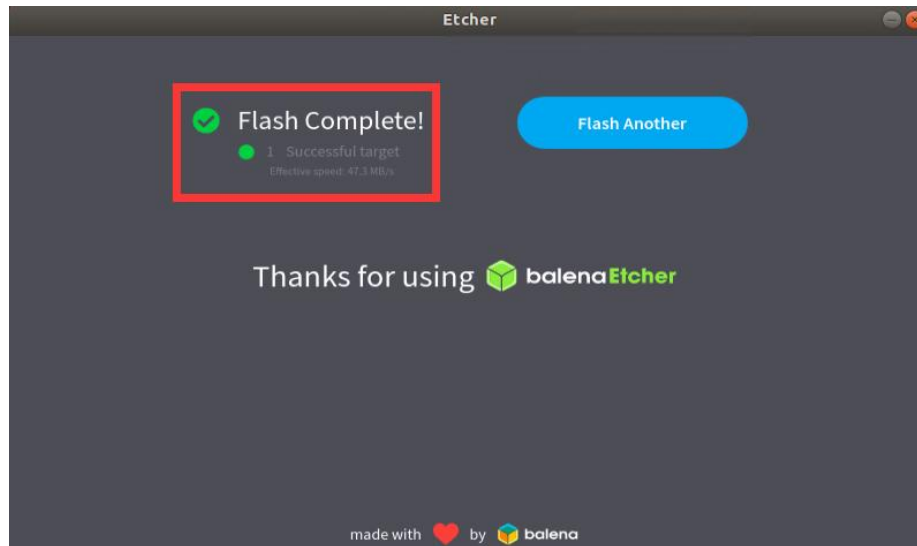
9) The interface displayed during the process of burning a Linux image with BalenaEtcher is shown in the following figure. In addition, the progress bar displays purple to indicate that the Linux image is being burned to the TF card.



10) After the Linux image is burned, BalenaEtcher defaults to verifying the images burned to the TF card to ensure that there are no issues during the burning process. As shown in the figure below, a green progress bar indicates that the image has been burned and BalenaEtcher is verifying the burned image.



11) After the successful burning is completed, the display interface of balenaEtcher is shown in the following figure. If a green indicator icon is displayed, it indicates that the image burning is successful. At this point, you can exit balenaEtcher, then unplug the TF card and insert it into the TF card slot of the development board for use.



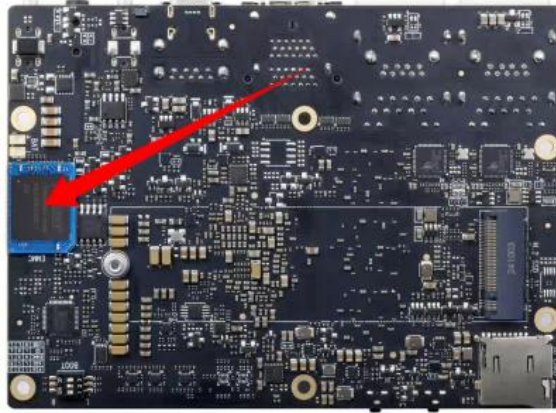
Note that before starting the system, please ensure that the DIP switch is turned to the TF card startup position. Please refer to the instructions in the section on the use of the [three DIP switches that control the startup device for instructions on how to use them](#).

2. 5. Method of burning Linux images to eMMC

Note that the Linux image referred to here specifically refers to the Ubuntu or openEuler image downloaded from the development board's download page.

1) The development board has reserved an extension interface for the eMMC module. Before burning the system to eMMC, it is necessary to purchase an eMMC module that matches the eMMC interface of the development board. Then install the eMMC module onto the development board. The matching eMMC module and the method of inserting the development board are as follows:





2) Burning Linux images into eMMC requires the use of a TF card, so first, you need to burn the Linux image onto the TF card, and then use the TF card to launch the development board and enter the Linux system. The method of burning Linux images to TF cards can be found in the section on the method of [burning Linux images to TF cards](#).

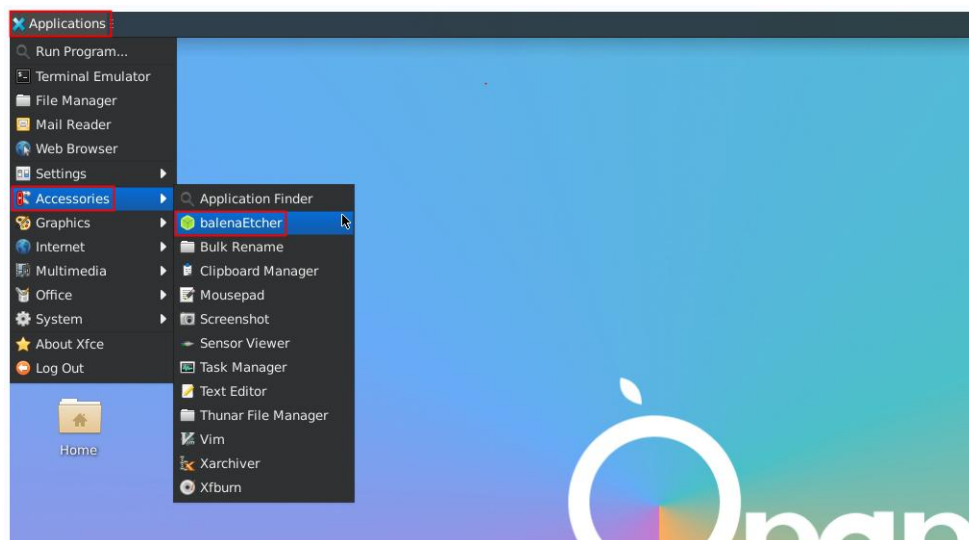
3) After starting the Linux system of the TF card, please confirm that the eMMC module has been recognized by the Linux system of the development board. If the eMMC module recognizes it properly, using the **fdisk -l** command under root users can display the capacity information of the eMMC module.

```
(base) root@orangepiaipro-20t:~# fdisk -l
.....
Disk /dev/mmcblk0: 28.91 GiB, 31037849600 bytes, 60620800 sectors
.....
```

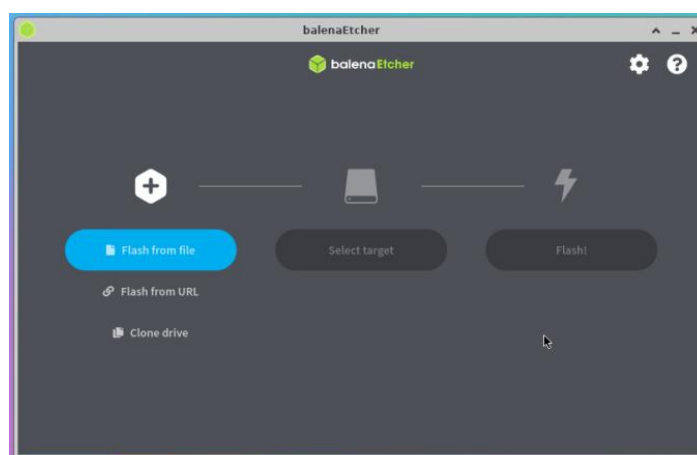
4) Then upload the compressed file of the Linux image to be burned to the Linux system of the TF card.

Note that Linux image compressed packages using the xz format do not need to be decompressed, as balenaEtcher will automatically decompress them when burning.

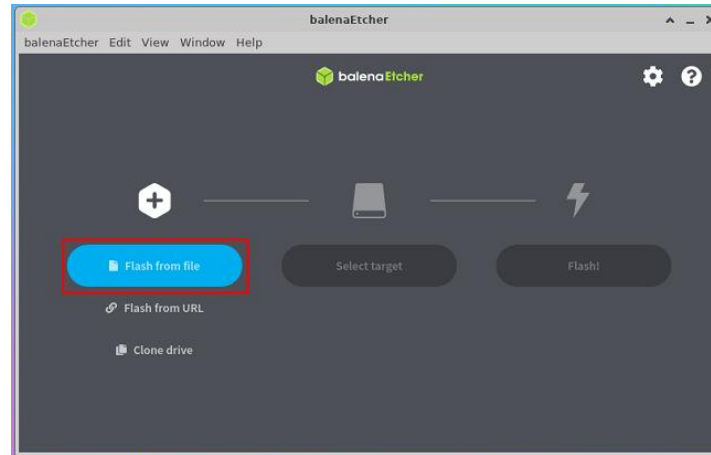
5) Then you can start using balenaEtcher software to burn images into eMMC. BalenaEtcher has been pre installed in the Linux system, and the opening method is as follows:



6) The interface after opening BalenaEtcher is as follows:

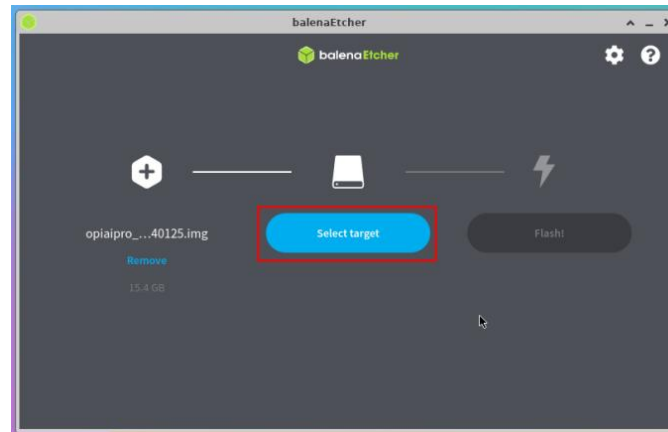


7) Then click on **Flash from file** to select the image file you previously uploaded and want to burn.

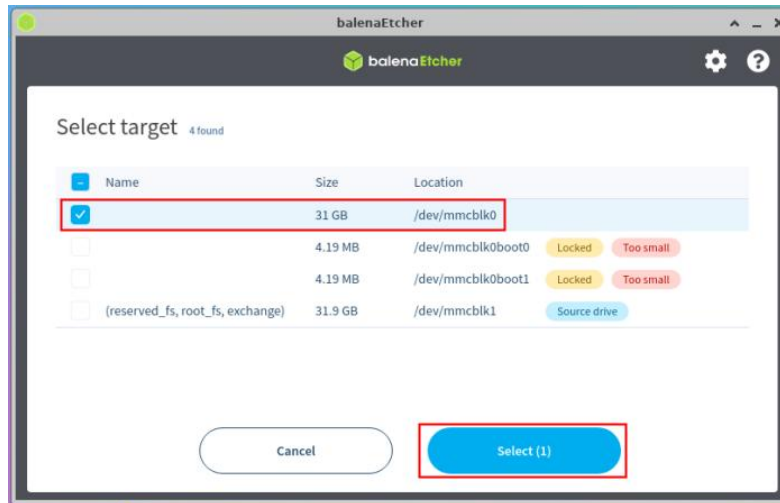


If there is a prompt indicating no permission when opening a Linux image file, use the command `sudo chmod 777 image file name` to add permissions to the image file.

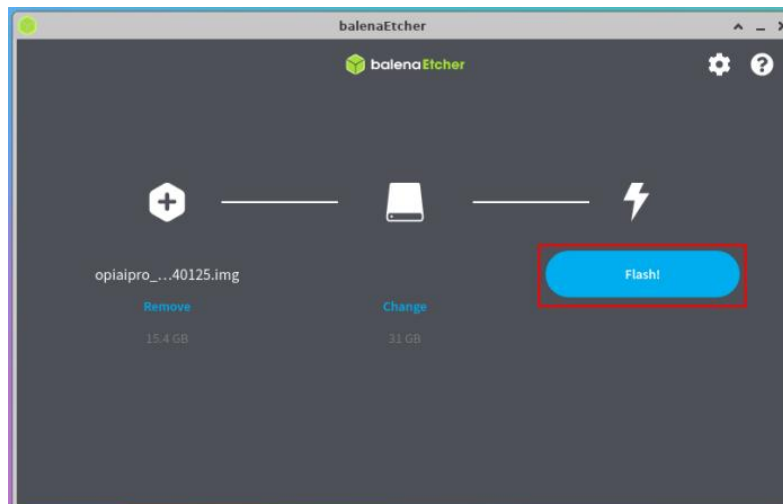
8) Then click on **Select target**



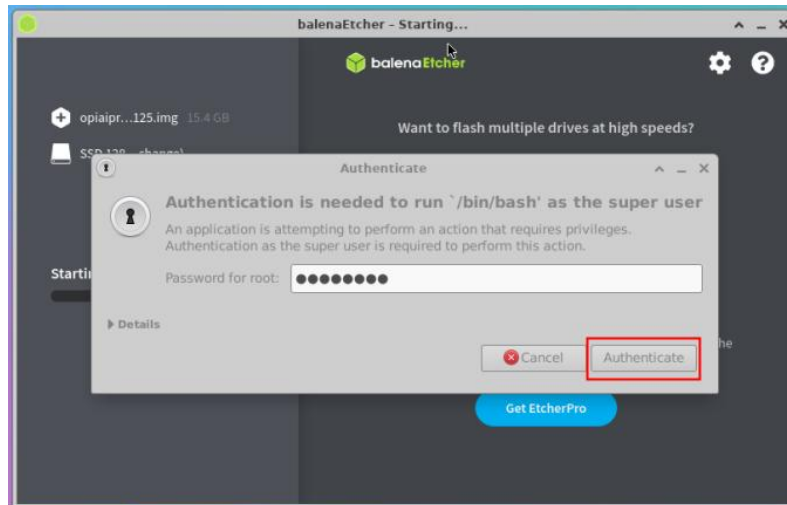
9) Then select the `/dev/mmcblk0` option corresponding to eMMC, and click **Select**.



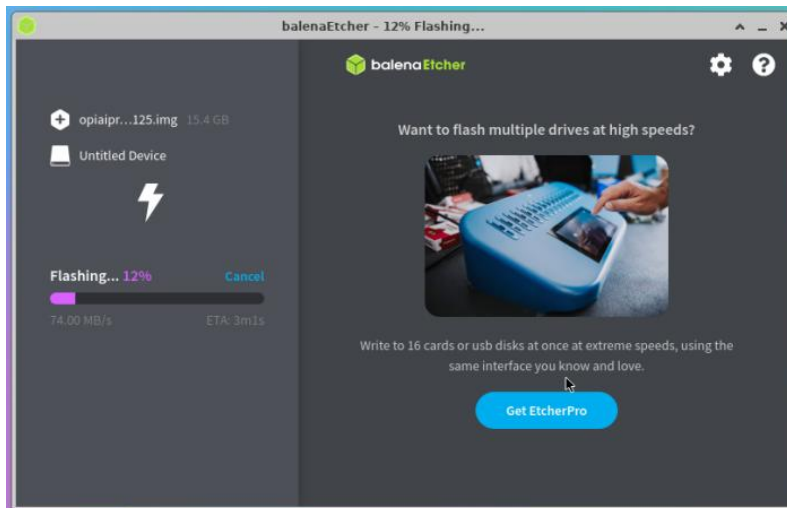
10) Then click on **Flash!** Start burning.



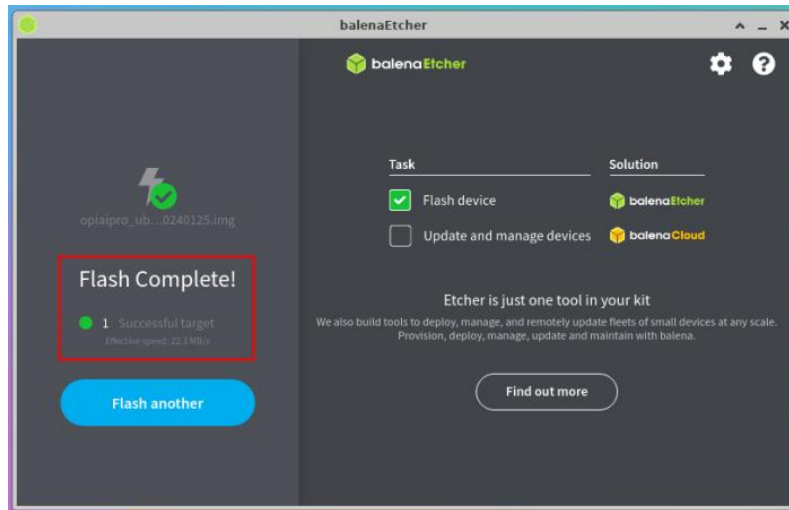
11) Then enter the password for the Linux system: **Mind@123** .



12) Then you will actually start burning Linux images into eMMC.



13) The display after burning the Linux image is as follows:



14) At this point, you can shut down the Linux system, then unplug the TF card and disconnect the Type-C power. Turn the three dial switches to the corresponding positions for eMMC startup, and then reinsert the Type-C power supply to start the Linux system in eMMC.

Note that before starting the system, please ensure that the DIP switch is turned to the eMMC start position. Please refer to the instructions in the section on [the use of the three DIP switches that control the startup device for instructions on how to use them](#).

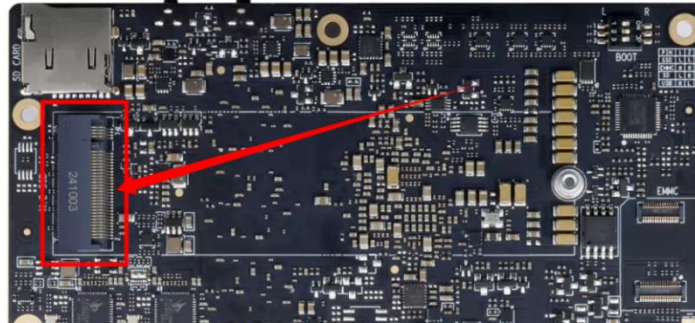
2. 6. Method of burning Linux images to NVMe SSDs

Note that the Linux image referred to here specifically refers to the Ubuntu or openEuler image downloaded from the development board's download page.

1) Firstly, a 2280 specification NVMe SSD needs to be prepared, and the PCIe specification supported by the M.2 slot of the development board is PCIe 3.0x4. Both PCIe 3.0 and PCIe 4.0 NVMe SSDs are usable, but the highest speed of PCIe 4.0 SSDs is only PCIe 3.0x4. SSDs of other specifications such as 2242 can also be used, but they cannot be fixed.



2) Then insert the NVMe SSD into the M.2 interface of the development board and secure it in place.



3) 烧 Burning a Linux image to an NVMe SSD requires the use of a TF card, so the first step is to burn the Linux image to the TF card, and then use the TF card to boot the development board into the Linux system. The method of **burning Linux images to TF cards** can be found in the section on the method of burning Linux images to TF cards.

4) After starting the Linux system to access the TF card, please confirm that the NVMe SSD has been recognized by the Linux system of the development board. If the NVMe SSD recognizes it properly, you can use the `sudo fdisk -l` command to see information related to **nvme**.

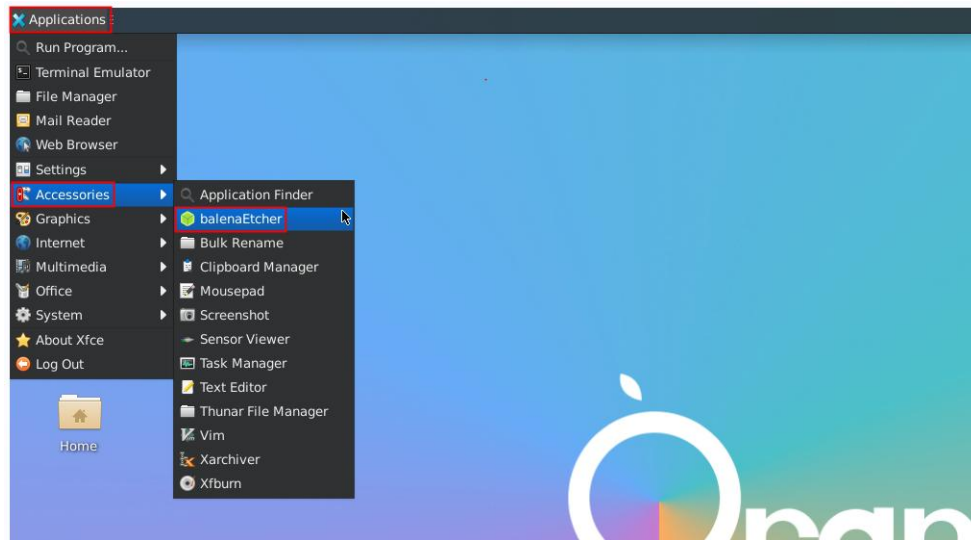
```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo fdisk -l | grep "nvme0n1"
Disk /dev/nvme0n1: 238.47 GiB, 256060514304 bytes, 500118192 sectors
.....
```

5) Then upload the compressed file of the Linux image to be burned to the Linux system of the TF card.

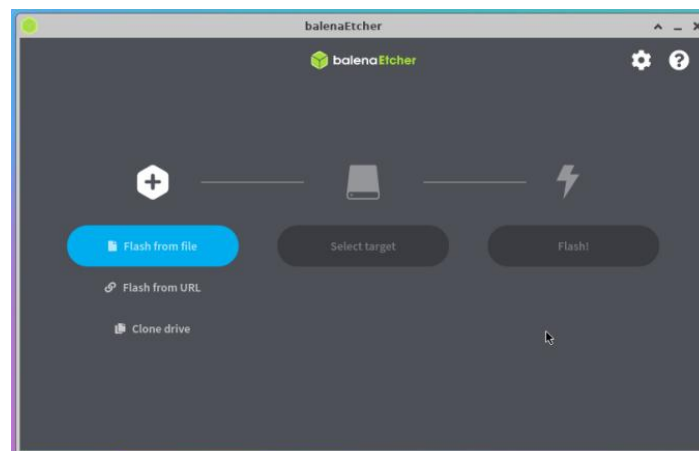
Note that Linux image compressed packages using the xz format do not need to be decompressed, as balenaEtcher will automatically decompress them when burning.



6) Then you can start using balenaEtcher software to burn the image to the SSD. BalenaEtcher has been pre installed in the Linux system, and the opening method is as follows:

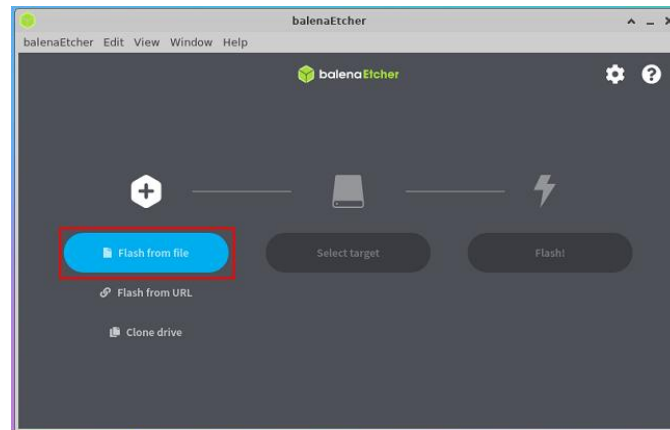


7) The interface after opening BalenaEtcher is as follows:

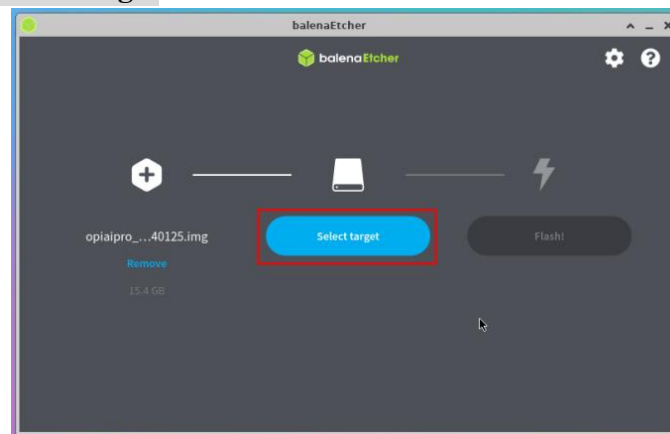


If there is a prompt indicating no permission when opening a Linux image file, use the command `sudo chmod 777 image file name` to add permissions to the image file.

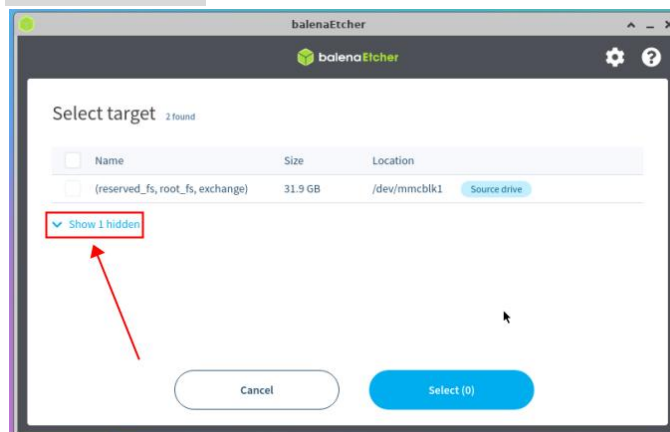
8) Then click on **Flash from file** to select the image file you previously uploaded and want to burn.



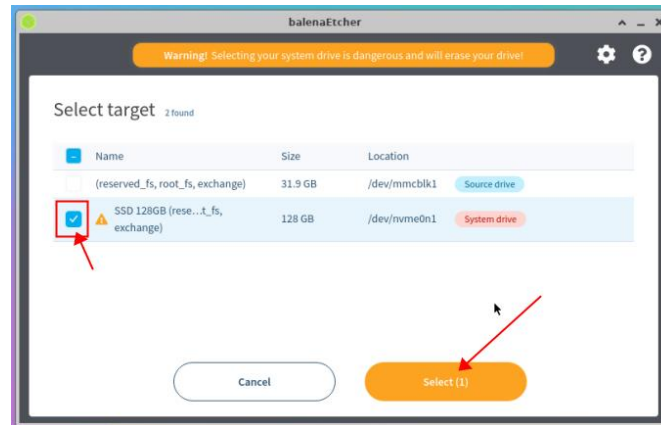
9) Then click on **Select target**.



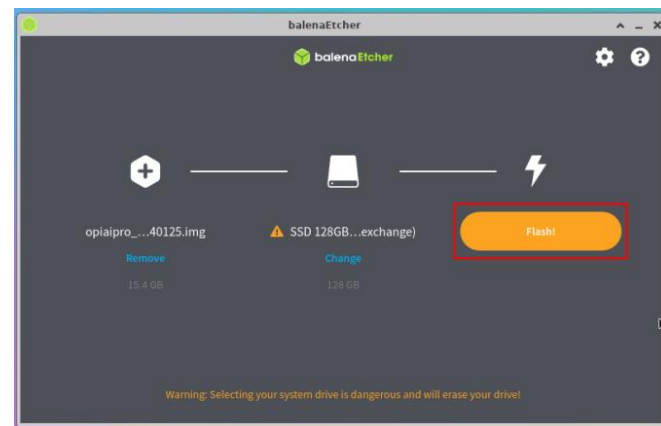
10) Then click on **Show 1 hidden**.



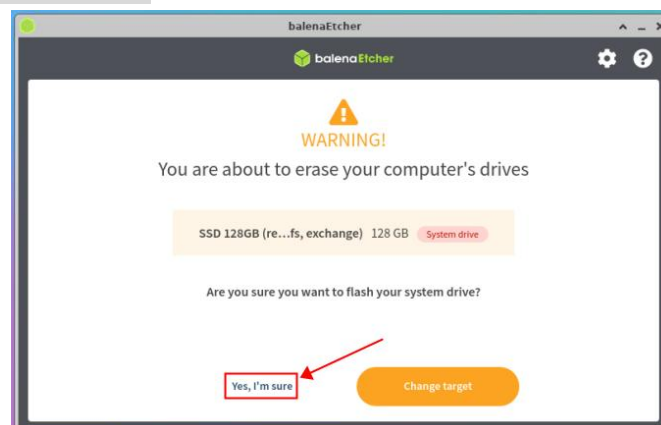
11) Then select the option corresponding to the SSD, and click **Select**.



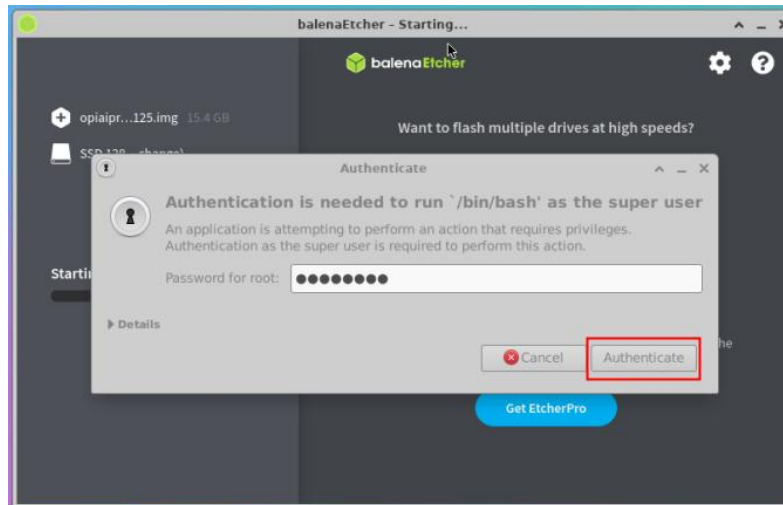
12) Then click on **Flash!** Start burning.



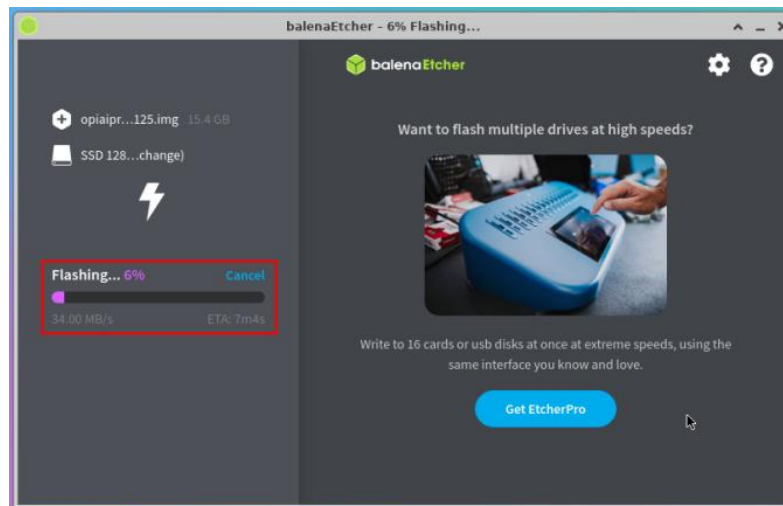
13) Then select **Yes, I'm sure**.



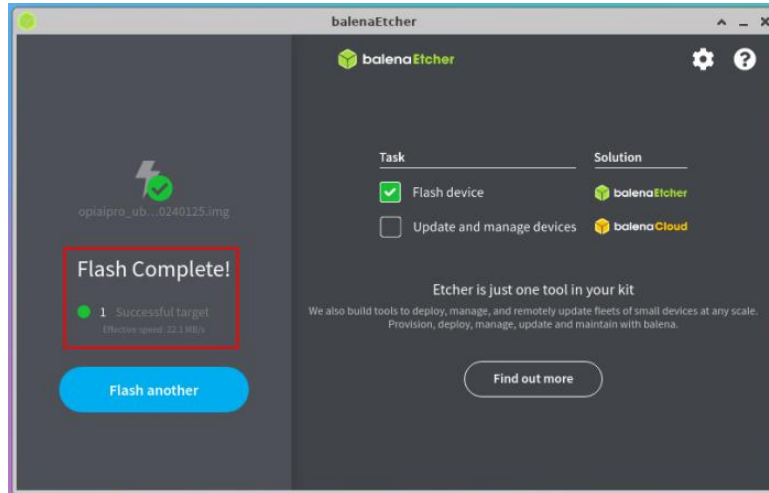
14) Then enter the password for the Linux system: **Mind@123**。



15) Then you will actually start burning the Linux image into the SSD.



16) The display after burning the Linux image is as follows:



17) At this point, you can shut down the Linux system, then unplug the TF card and disconnect the Type-C power. Turn the DIP switch to the corresponding position for SSD startup, and then reinsert the Type-C power supply to start the Linux system in the SSD.

Note that before starting the system, please ensure that the DIP switch is set to the SSD startup position. Please refer to the instructions in the section on the use of the [three DIP switches that control the startup device for instructions on how to use them](#).

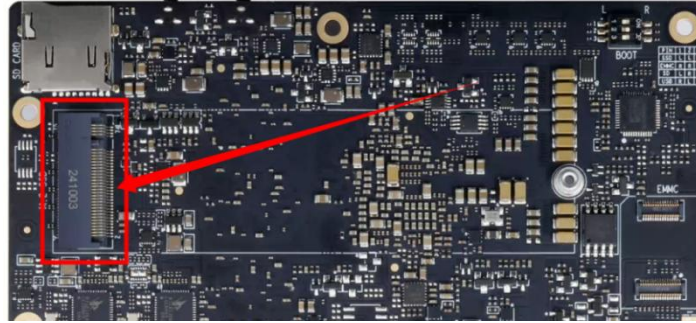
2. 7. How to Flash a Linux Image to a SATA SSD

Note: The Linux image referred to here specifically means the Ubuntu or openEuler image downloaded from the development board's resource download page.

1) First, prepare an M.2 2280-sized SATA SSD. Other sizes like 2242 can also be used, but they cannot be fixed in place.



2) Then, insert the SATA SSD into the M.2 slot on the development board and secure it.



3) To flash the Linux image to the SATA SSD, you need to use a TF card. First, flash the Linux image to the TF card, then use the TF card to boot the development board into the Linux system. For instructions on how to flash the Linux image to the TF card, please [refer to the section on how to flash a Linux image to a TF card](#).

4) After booting into the Linux system from the TF card, you first need to update the **dt.img** file corresponding to the SATA. The steps are as follows:

- a. First, navigate to the **/opt/opi_test/sata** directory.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ cd /opt/opi_test/sata
```

- b. Then, run the **update.sh** script to update the dt.img file corresponding to the SATA.

```
(base) HwHiAiUser@orangepiaipro-20t:/opt/opi_test/sata$ sudo ./update.sh
```

- c. After running the **update.sh** script, the Linux system will automatically restart to apply the configuration changes.
- d. If everything goes smoothly, after re-entering the Linux system from the TF card, the SATA SSD should be recognized.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo fdisk -l | grep "/dev/sd"
```

```
Disk /dev/sda: 238.47 GiB, 256060514304 bytes, 500118192 sectors
```

```
.....
```

The development board's M.2 interface supports both NVMe SSDs and SATA SSDs, but it's a mutually exclusive choice between the two. The default configuration in the Linux system's dt.img` file is set to PCIe. If you have connected a SATA SSD to the M.2 interface, you need to manually switch to the SATA configuration in the `dt.img` file to use it properly.

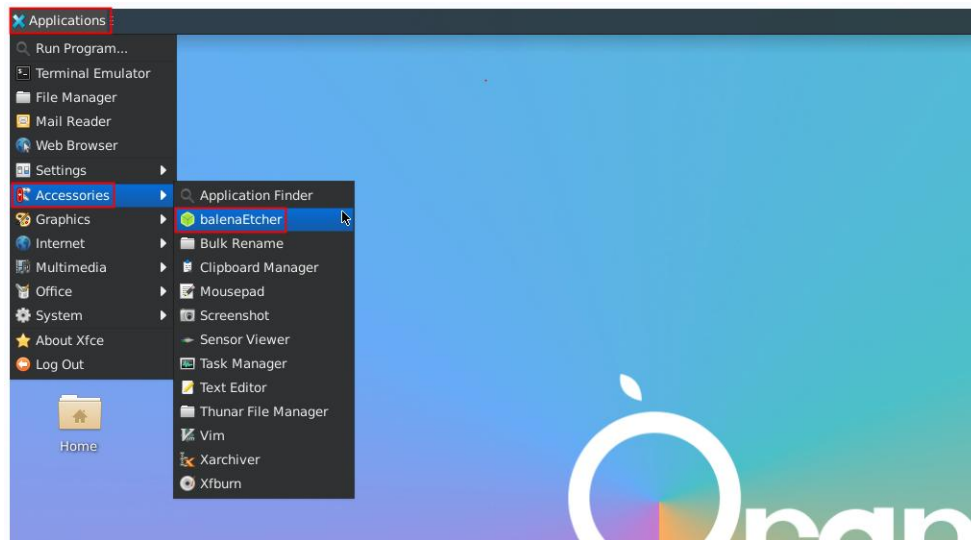
5) Next, upload the compressed Linux image file package to the Linux system on the TF



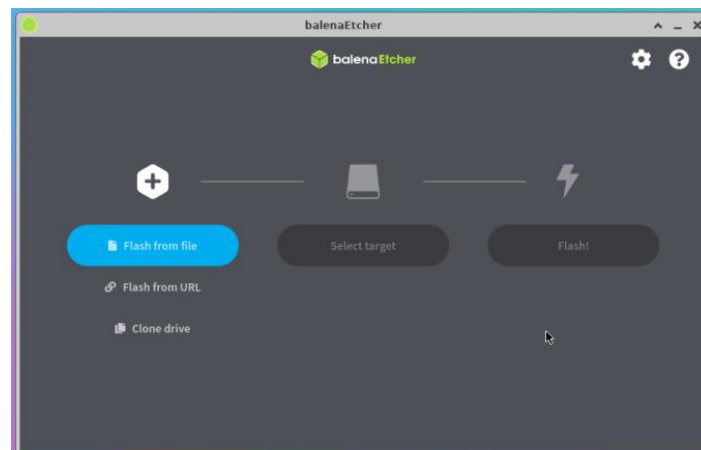
card.

Note: The Linux image compressed in xz format does not need to be uncompressed; balenaEtcher will automatically handle the decompression during the flashing process.

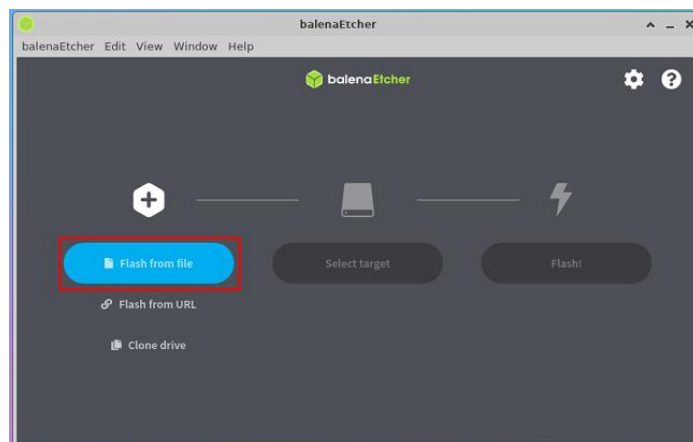
6) Now you can start using the balenaEtcher software to flash the image to the SSD. balenaEtcher is already pre-installed in the Linux system. To open it, follow these steps:



7) The interface of balenaEtcher after opening looks like this:

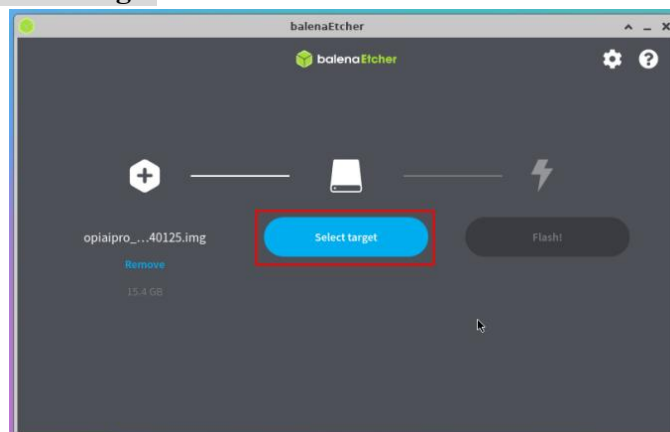


8) Next, click on **Flash from file** to select the Linux image file that was previously uploaded for burning.

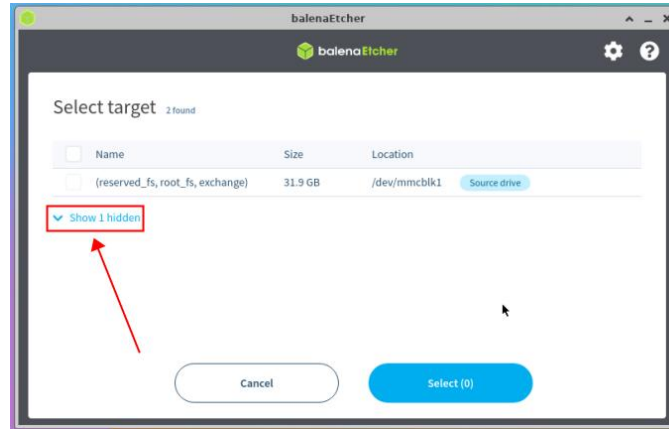


If you encounter permission issues when opening the Linux image file, use the command `sudo chmod 777 filename`` to grant the necessary permissions to the image file.

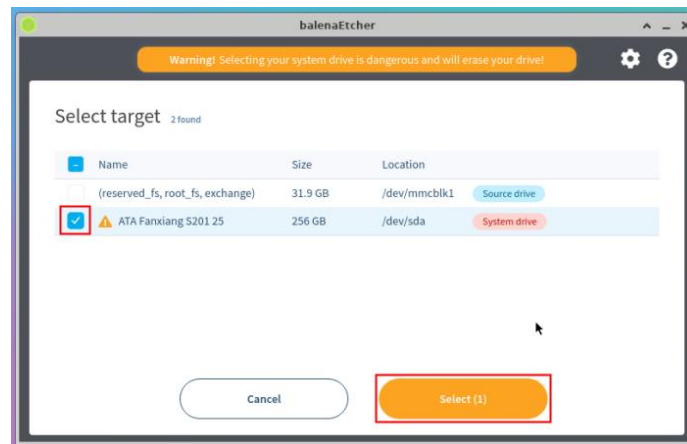
9) Then click on **Select target**.



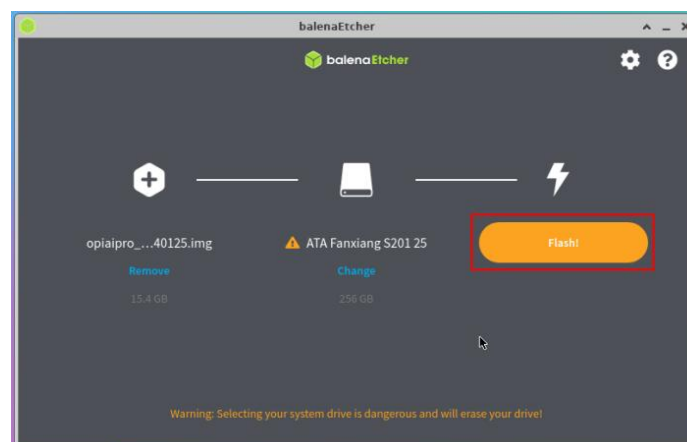
10) Then click on **Show 1 hidden**.



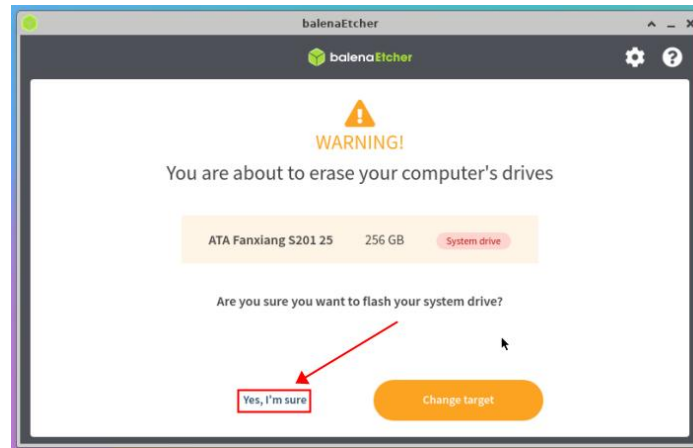
11) Next, choose the option corresponding to the SSD, then click **Select**



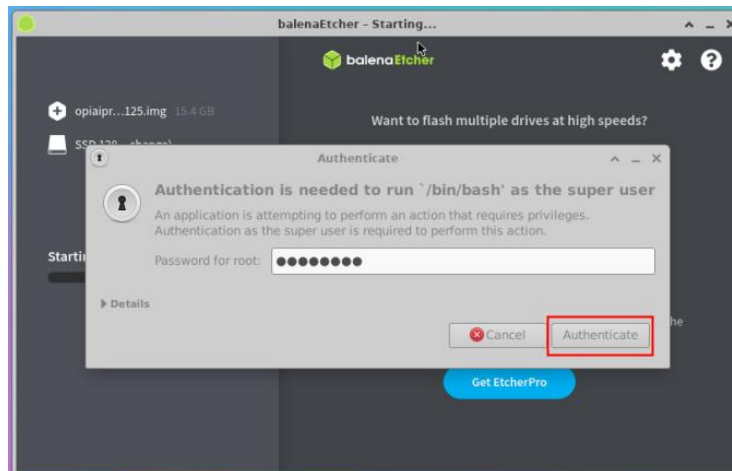
12) Then click **Flash!** to start the flashing process.



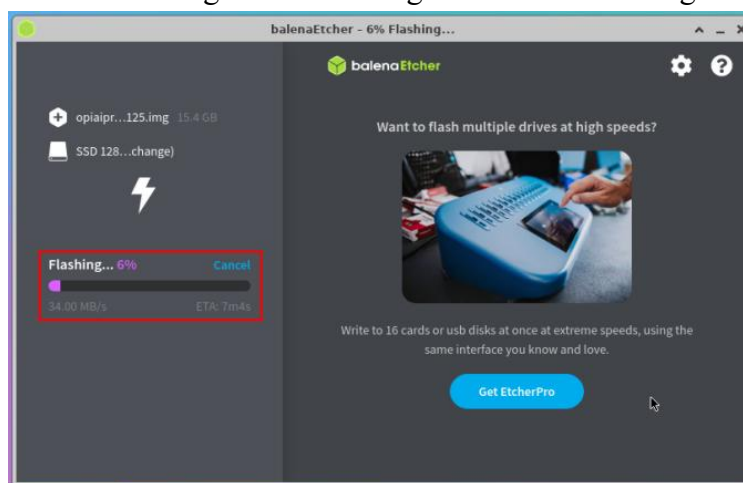
13) And then select **Yes, I'm sure**.



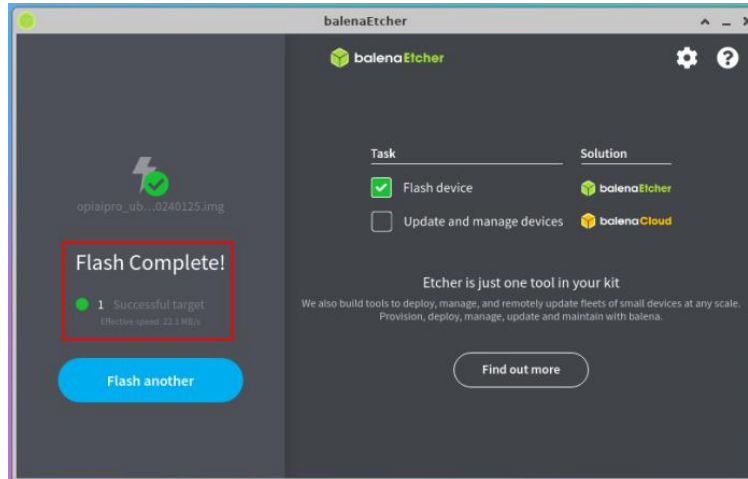
14) Then enter the password for the Linux system: **Mind@123**.



15) Then the process of flashing the Linux image to the SSD will begin.



16) After the Linux image is successfully flashed, the display will look like this:



17) After flashing is complete, it is necessary to burn the SATA version of `dt.img` to the SATA SSD, as the provided images default to PCIe configuration. The specific command is as follows:

```
sudo dd if=/opt/opi_test/dt_img/dt_drm_sata.img of=/dev/sda count=4096 seek=114688 bs=512
```

Please note, in the command above, the "/dev/sda" after the "of=" parameter represents the device node name for the SSD. Please modify it according to your actual situation.

18) Now you can shut down the Linux system, remove the TF card, and disconnect the Type-C power. Switch the dip switch to the position corresponding to SSD startup, then reconnect the Type-C power to start the Linux system from the SSD.

Please ensure that before starting the system, the dip switch is set to the position for SSD startup. Refer to [the instructions in section 3 on how to use the three dip switches to control the startup device](#).

2. 8. Steps to Boot the Development Board

1) Insert the TF card, eMMC module, or SSD with the flashed image into the corresponding slot on the development board.

2) Then set the dip switch to the correct position.



3) The development board has two HDMI ports (**currently only HDMI0 supports displaying the Linux desktop; functionality for HDMI1 to display the Linux desktop requires software updates**). To display the Linux desktop, connect the HDMI0 port of the development board to an HDMI monitor.

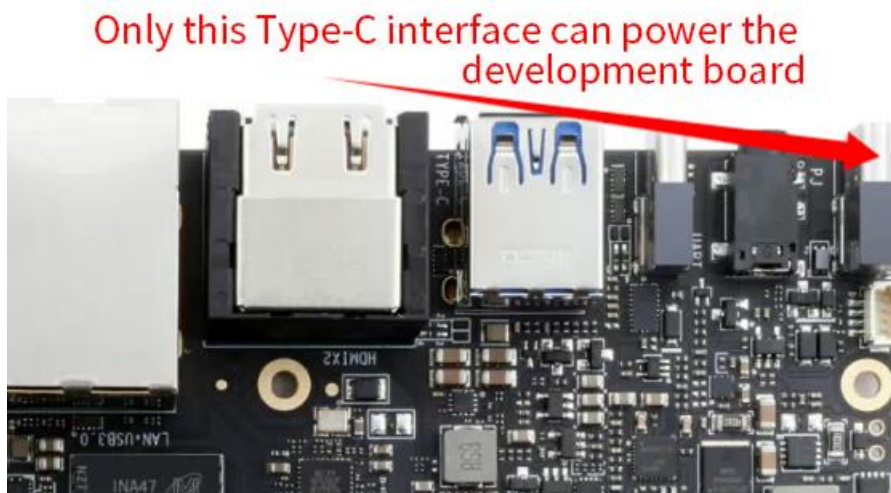


HDMI0 interface location

4) The development board has USB ports where you can connect a USB mouse and keyboard to control the development board.

5) The development board has two 2.5G Ethernet ports that can be plugged into a network cable for Internet access.

6) Then you need to connect a 20V PD-65W Type C connector power supply, the location of the power connector is shown below:



7) Then turn on the power adapter switch, if everything is fine, wait for some time, the HDMI monitor will be able to see the login interface of the Linux system.

8) If you want to view the output information of the system through the debugging serial

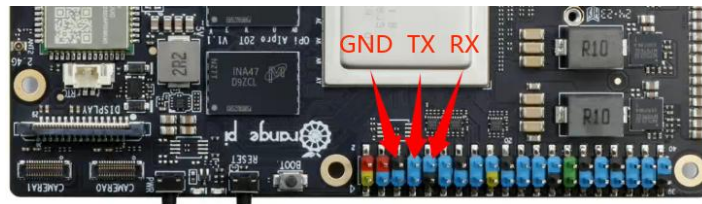


port, please use the serial port cable to connect the board to the computer, please [refer to the section of debugging the use of the serial port for the serial port connection method](#).

2. 9. Debugging the use of serial ports

The board uses uart0 as the debug serial port by default. Note that the tx and rx pins of uart0 are connected to two places at the same time, so there are two ways to use the debug serial port:

- 1) The tx and rx pins of uart0 are connected to pins 8 and 10 of the 40 pin extension interface. This method requires the preparation of a 3.3v USB to TTL module and the corresponding Dupont cable, and then you can use the debugging serial port function of the development board normally.



- 2) The tx and rx pins of uart0 are also connected to the CH343P chip of the development board, and then through the CH343P chip to the Type-C USB interface. This method only requires a Type-C USB interface cable to connect the development board to the computer's USB port can begin to use the development board debugging serial port function, without the need to purchase a USB to TTL module. This method is the recommended method.





Please also note that the two methods above can only be used in one of two ways, so please do not use them at the same time.

2. 9. 1. Connection instructions for using the debug serial port via Type-C

USB interface

1) First of all, you need to prepare a data cable with Type-C USB port.



2) Then plug the Type-C USB port end into the Type-C USB port on the development board.

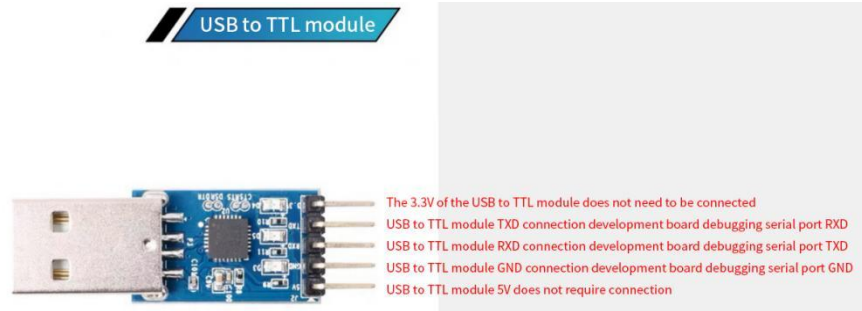


3) Plug the other end of the cable into the USB port of your computer.

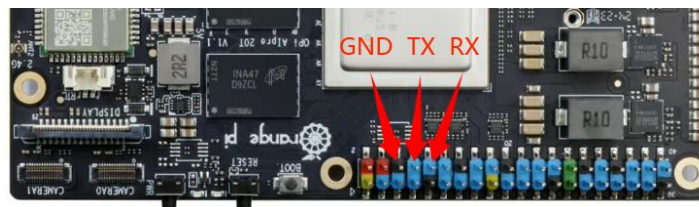
2. 9. 2. Connection instructions for using the debug serial port via uart0

in the 40 pin interface

1) First you need to prepare a 3.3v USB to TTL module, then insert the USB port end of the USB to TTL module into the USB port of your computer.



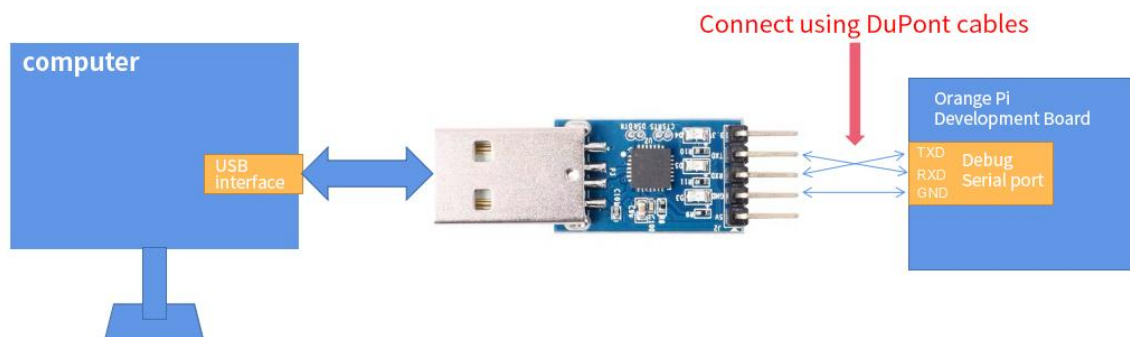
2) The correspondence between the GND, TX and RX pins of the development board's debug serial port is shown below:



3) USB to TTL module GND, TX and RX pins need to be connected to the debug serial port of the development board via Dupont cable.

- a. The GND of the USB to TTL module is connected to the GND of the development board.
- b. The RX of the USB to TTL module is **connected to the TX of the development board.**
- c. The TX of the USB to TTL module is **connected to the RX of the development board.**

4) The schematic diagram of the USB to TTL module connecting the computer and the development board is shown below:



Schematic diagram of USB to TTL module connecting computer and Orange Pi development board



The TX and RX of the serial port need to be cross-connected, if you don't want to carefully distinguish the order of TX and RX, you can connect the TX and RX of the serial port casually first, and then exchange the order of TX and RX if there is no output from the test serial port, so that there is always an order that is right.

2. 9. 3. Ubuntu Platform Debugging Serial Port Usage

Linux can be used under the serial port debugging software there are many, such as putty, minicom, etc., the following demonstrates the use of putty.

1) First of all, [please follow the instructions in the subsection of Connecting the serial port via 40 pin uart0 or Connecting the serial port via Type-C USB port](#) (please choose one of the two methods according to your own situation) to connect the board to the computer, if the serial module is recognized properly, you can see the corresponding device name under `/dev` in the Ubuntu PC. If the serial module is recognized correctly, you can see the corresponding device name under `/dev` in Ubuntu PC, please remember this device name, it will be used when setting up the serial software later.

- a. If uart0 in the 40 pin interface is used the device name displayed is typically `/dev/ttyUSB0`.

```
test@test:~$ ls /dev/ttyUSB*  
/dev/ttyUSB0
```

- b. If using a Type-C USB interface the device name displayed is typically `/dev/ttyACM0`.

```
test@test:~$ ls /dev/ttyACM*  
/dev/ttyACM0
```

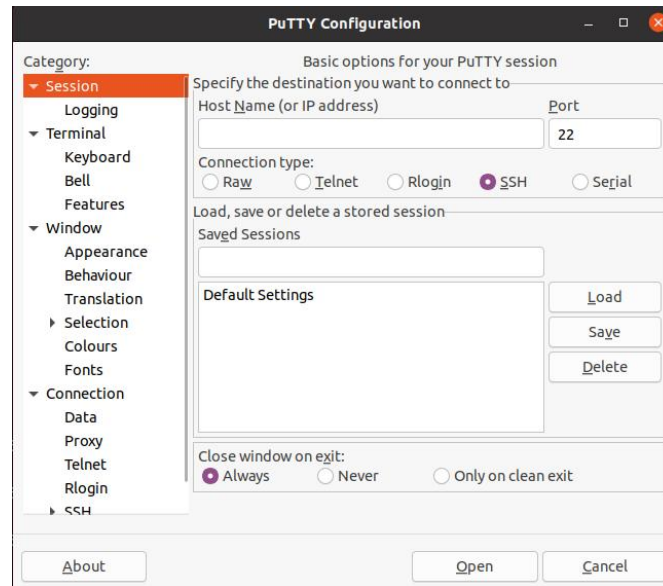
2) Then use the following command to install putty on your Ubuntu PC.

```
test@test:~$ sudo apt-get update  
test@test:~$ sudo apt-get install -y putty
```

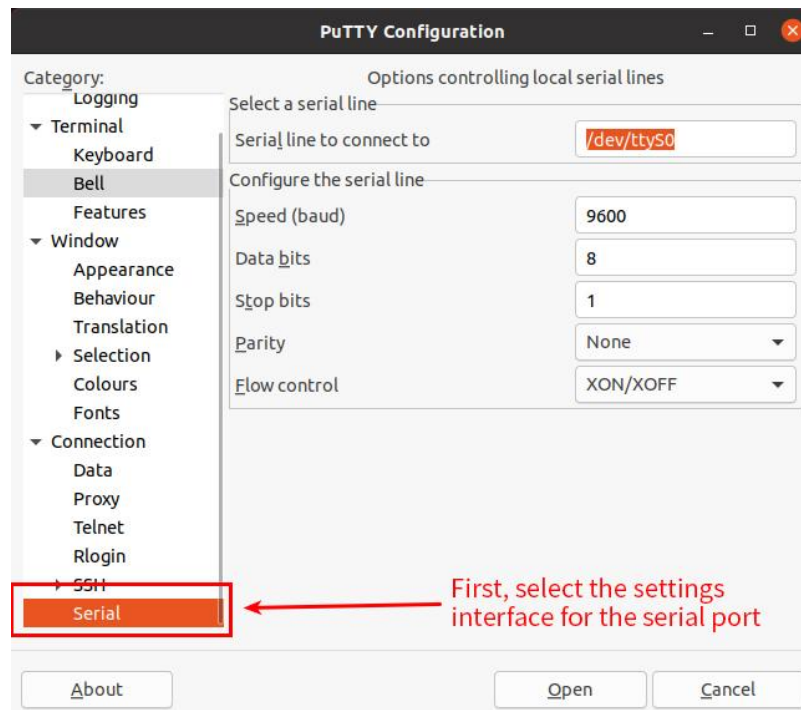
3) Then run putty, **remember to add sudo privileges.**

```
test@test:~$ sudo putty
```

4) The following screen will pop up after executing the putty command.

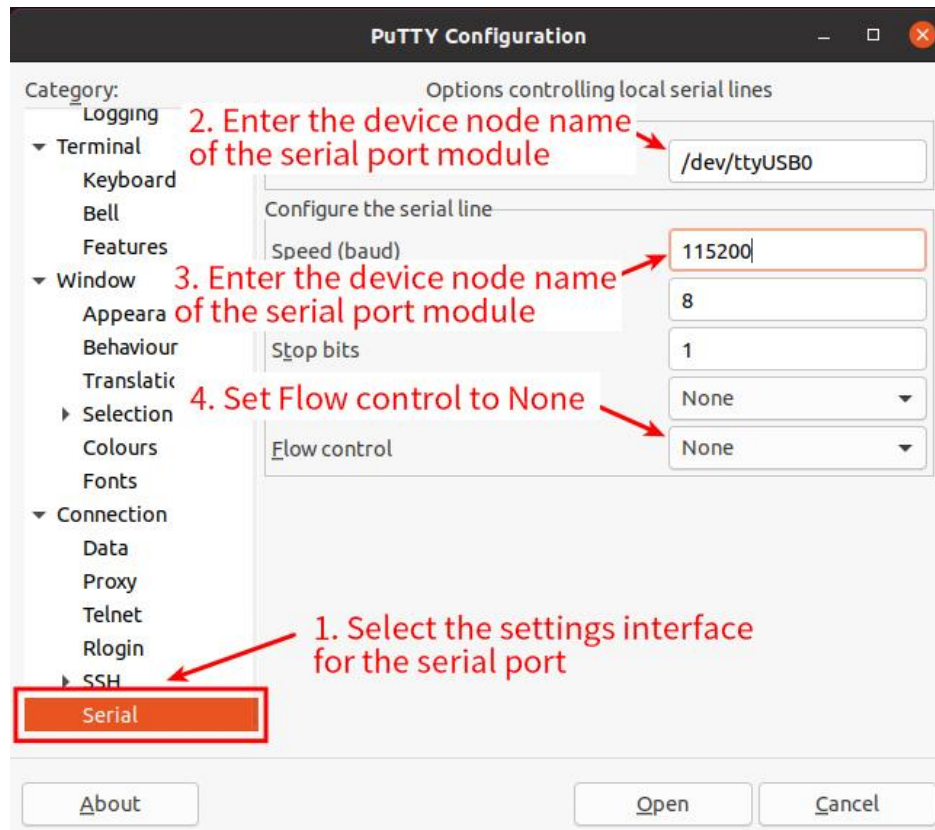


5) First select the setting screen of serial port.



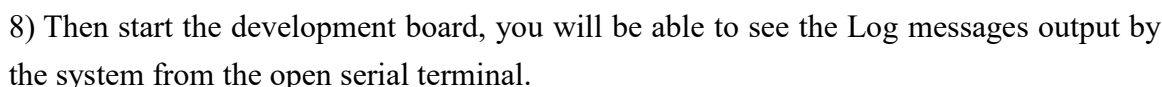
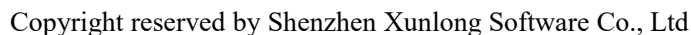
6) Then set the parameters of the serial port.

- Set the **Serial line to connect to** to `/dev/ttyUSB0` or `/dev/ttyACM0` (please modify according to the actual situation).
- Set **Speed(baud)** to **115200** (the baud rate of the serial port).
- Set **Flow control** as **None**.



7) After setting up the serial interface, go back to the Session interface.

- a. First select the **Connection type** as **Serial**.
- b. Then click the **Open** button to connect the serial port.





9) When you see the login screen, you can use the following account and password to login to your Linux system.

Account	Password
root	Mind@123
HwHiAiUser	Mind@123

2. 9. 4. Windows platform debugging serial port usage

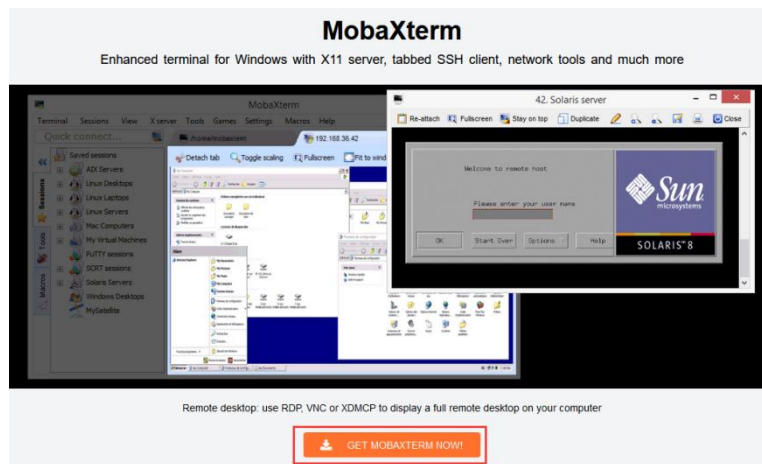
Windows can use a lot of serial debugging software, such as SecureCRT, MobaXterm, etc., the following demonstrates the use of MobaXterm, this software has a free version, no need to purchase a serial number can be used.

1) First download MobaXterm.

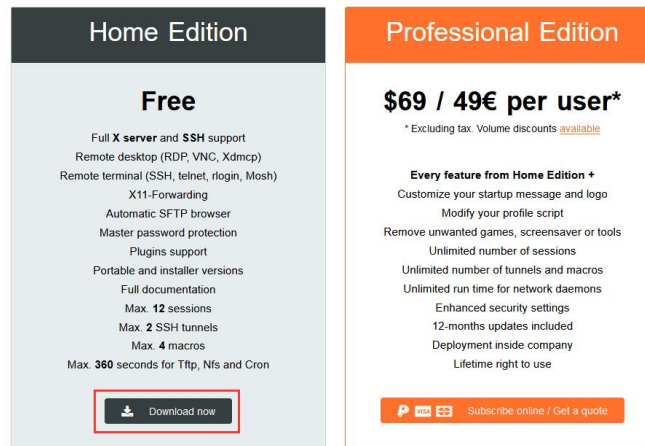
a. Download MobaXterm at the following URL:

<https://mobaxterm.mobatek.net/>

b. Go to the MobaXterm download page and click **GET XOBATERM NOW!**



c. Then choose to download the Home version.



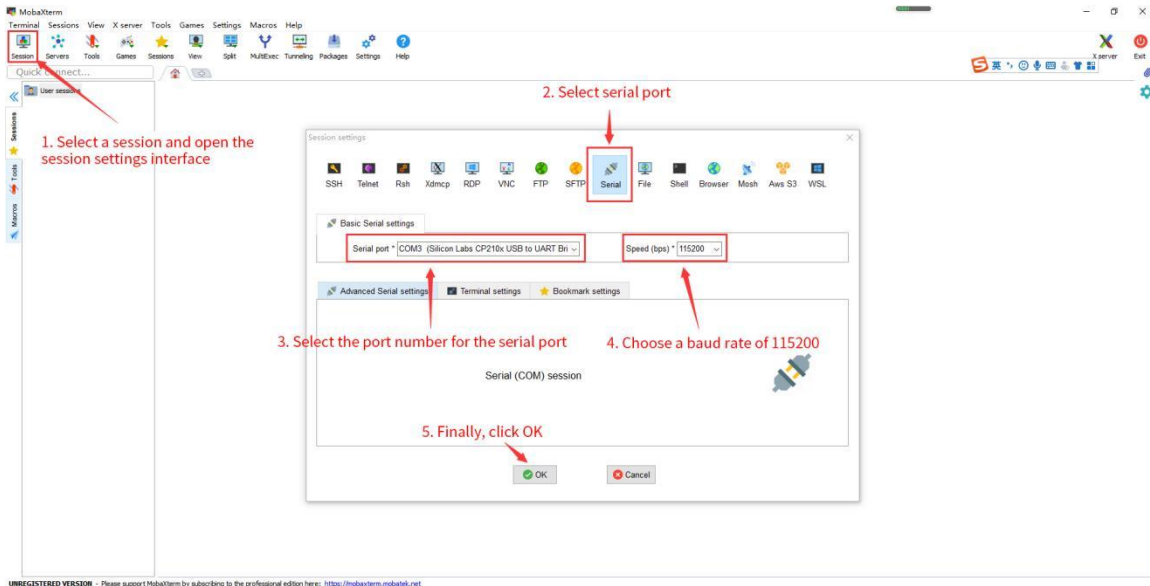
- d. Then choose the Portable version, after downloading, no need to install, directly open it can be used.



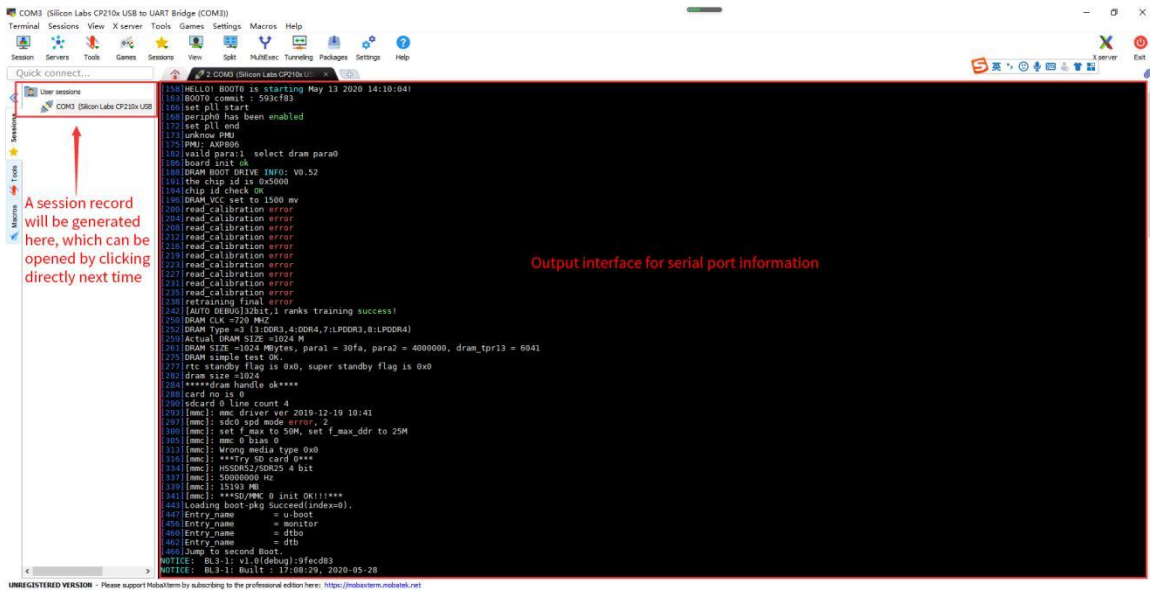
- 2) After downloading, use the decompression software to unzip the downloaded package, you can get the executable software of MobaXterm, and then double-click to open it.

MobaXterm_Personal_23.6.exe	2023/12/21 6:15	应用程序	16,556 KB
CygUtils.plugin	2023/12/21 5:08	PLUGIN 文件	17,748 KB
CygUtils64.plugin	2023/12/21 5:08	PLUGIN 文件	11,723 KB

- 3) After opening the software, set the serial port connection as follows:
- Open the session setup screen.
 - Select the serial port type.
 - Select the port number of the serial port (select the corresponding port number according to the actual situation), if you can't see the port number, please use 360 Driver Master to scan and install the driver of USB to TTL serial port chip.
 - Select the baud rate of the serial port as **115200**.
 - Finally, click the "OK" button to complete the setup.



4) Click the "OK" button to enter the following interface, at this time to start the development board will be able to see the output information of the serial port.



5) When you see the login screen, you can use the following account and password to login to your Linux system.

Account	Password
root	Mind@123
HwHiAiUser	Mind@123



2. 10. Notes on the use of WIFI Bluetooth antennas

The WIFI Bluetooth antenna for the development board is shown below on the left:



Antenna * 1



Development board * 1

The proper installation of the WIFI Bluetooth antenna is shown below:

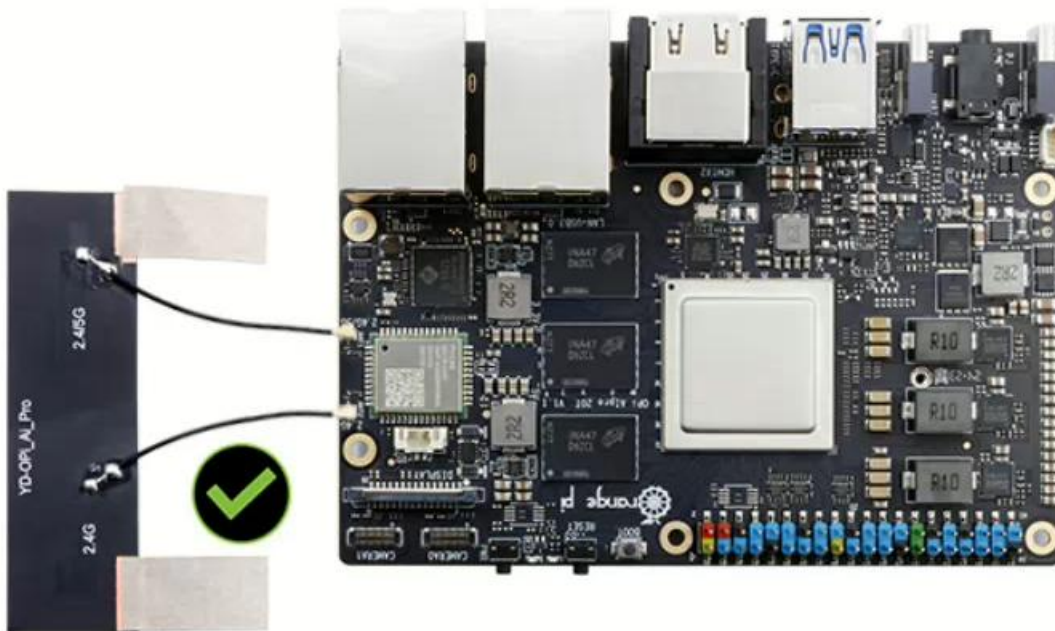
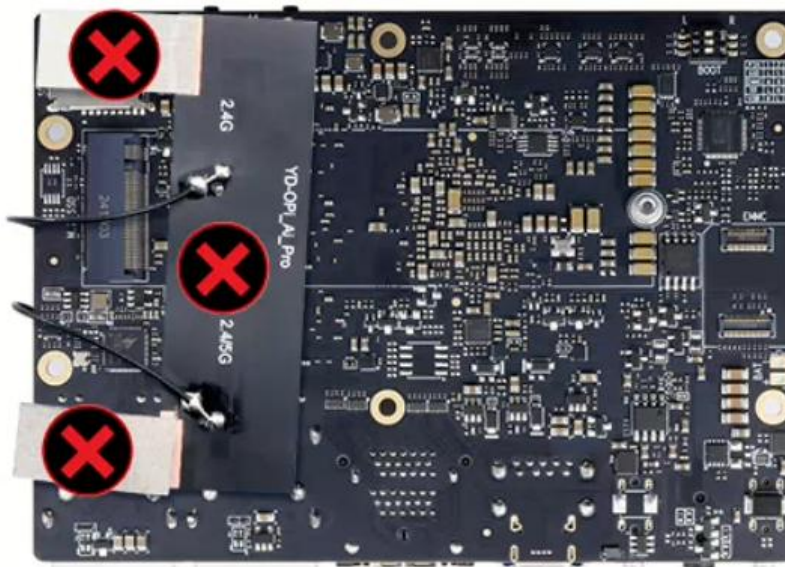


Diagram of correct installation and placement of antennas.

After installing the antenna, please do not attach the antenna to the back of the development board as shown in the figure below, and also the conductive cloth on



the antenna should not be next to the development board, otherwise the board may be burned.



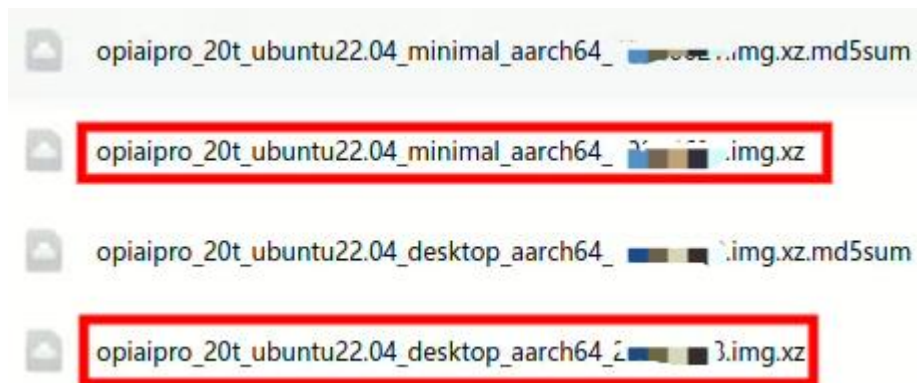


3. Instructions for using the Ubuntu Xfce desktop system

After going to the download link of the Ubuntu image you can see the two ubuntu images as shown in the image below, the difference between them is:

1) The **minimal** image is a mirror with only the most basic features, like Linux desktop, CANN and AI sample code are not pre-installed. This image is only recommended for developers who want to customize and install Linux desktop and AI related software from scratch.

2) The **desktop** image comes pre-installed with Linux desktop, CANN, AI sample code and a series of test programs. If you want to use the functions of the development board normally, please use this image. The contents of this chapter are based on the desktop image.



3. 1. Supported Ubuntu image types and kernel versions

Linux image types	kernel version	desktop version
Ubuntu 22.04 - Jammy	Linux5.10	Support

3. 2. Adaptation of Linux system functions

Function	Yes or No	Linux kernel driver
HDMI0 1080p display	OK	OK
HDMI0 4K display	NO	NO



HDMI0 Audio	OK	NO
HDMI1 Display	OK	NO
HDMI1 Audio	OK	NO
headphone playback	OK	NO
Headphone MIC Recording	OK	NO
TypeC USB3.0 Host	OK	OK
TypeC USB3.0 Device	OK	OK
USB3.0 Host x 3	OK	OK
2.5G network port x 2	OK	OK
2.5G network port light	OK	OK
WIFI	OK	OK
Bluetooth	OK	OK
Type-C USB debug serial port	OK	OK
Reset Button	OK	OK
Power On/Off Button	OK	OK
BOOT burn-in button	OK	OK
MIPI camera 0	NO	NO
MIPI camera 0	NO	NO
MIPI LCD Display	NO	NO
Power indicator	OK	OK
Software controllable LED lights	OK	OK
fan port	OK	OK
Battery Connector	OK	OK
RTC interface	OK	OK
TF card startup	OK	OK
TF Card Boot Recognition eMMC	OK	OK
TF card boot recognizes NVMe SSDs	OK	OK
TF card boot recognizes NVMe SSDs	OK	OK
eMMC boot	OK	OK
SATA SSD boot	OK	OK
NVMe SSD boot	OK	OK
3 dip switches	OK	OK

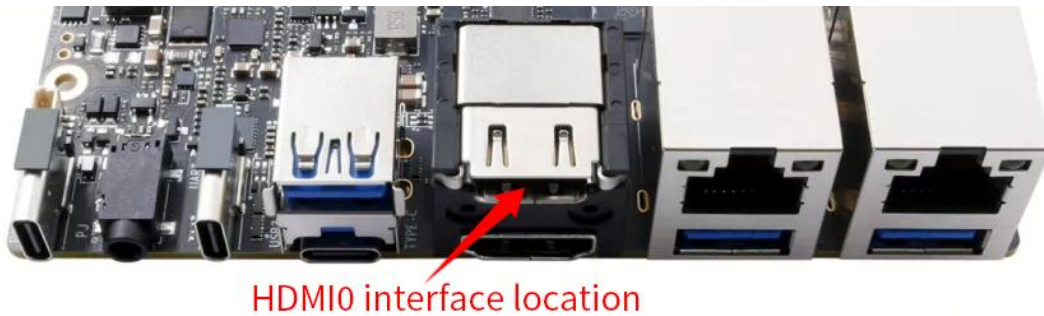


40 pin-Debug Serial Port	OK	OK
40 pin-GPIO	OK	OK
40 pin-UART	OK	OK
40 pin-SPI	OK	OK
40 pin-I2C	OK	OK
40 pin-PWM	OK	NO

3. 3. Linux system login instructions

3. 3. 1. Logging in to the desktop of a Linux system

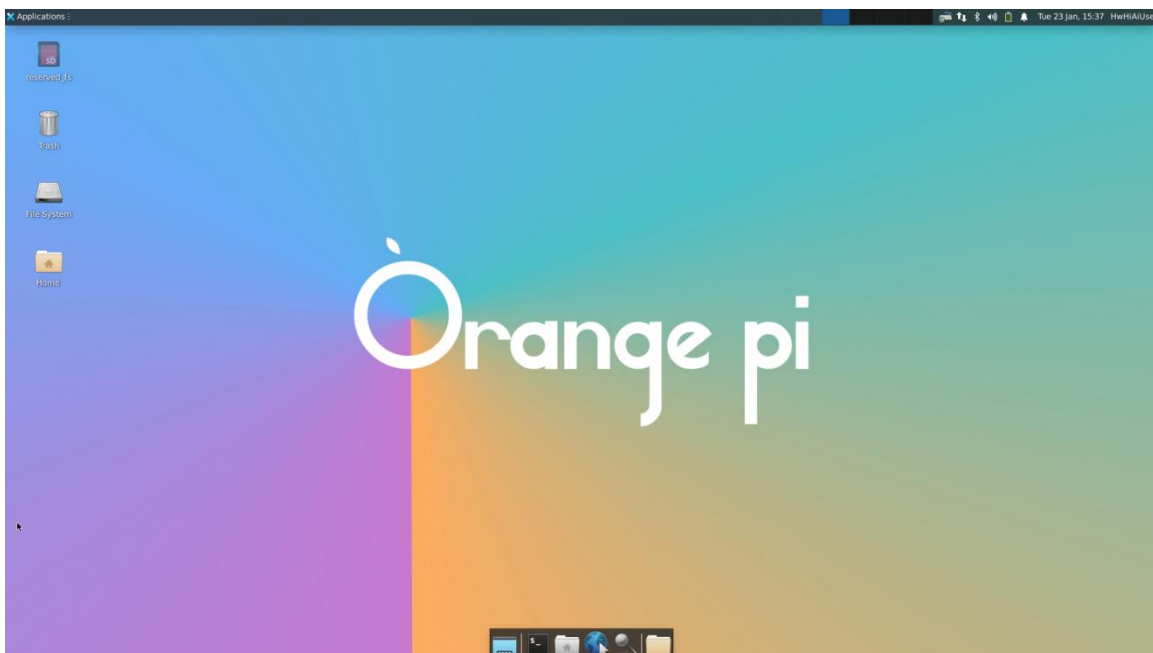
The development board has two HDMI ports, currently only HDMI0 support to display the desktop of Linux system, HDMI1 still need to wait for software updates. If you want to display the desktop of Linux system, please connect the HDMI0 interface of the development board to the HDMI monitor.



After the development board is powered on and turned on, you need to wait for some time before the HDMI monitor displays the login interface of the Linux system, which is shown in the following figure:



The default login user of the Linux desktop system is **HwHiAiUser**, and the login password is **Mind@123**. Currently, there is no channel to open the login for the root user. The desktop of the Linux system displayed after successful login is shown in the following figure:



3. 3. 2. Default login and password for Linux systems

Account number	Password
----------------	----------



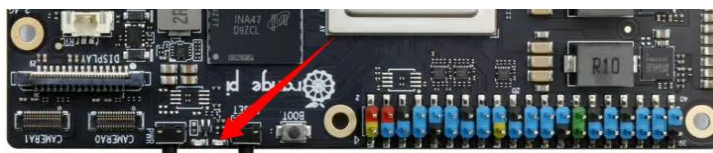
root	Mind@123
HwHiAiUser	Mind@123

When entering the password prompts an error, or there is a problem with the ssh connection, please note that as long as you are using the Linux image provided by Orange Pi, **please do not suspect that the password above is incorrect**, but look for other reasons.

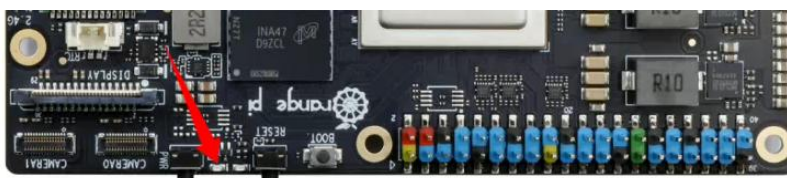
3. 4. On-Board LED Test Description

There are two green LED on the development board that function as shown below:

1) The green light near the reset button: This green light is a power indicator, which is controlled by hardware and cannot be controlled by software. As long as the board is connected to the Type-C power supply and powered on, this green light will be lit.



2) Green light near the on/off button: This green light is controlled by **GPIO4_19** to light up or down, and can be used as an indicator for SATA hard disk or other desired purposes. It is lit by default in current Linux releases. When you see this light on, it at least indicates that the Linux kernel has booted.



3. 5. Network connectivity testing

3. 5. 1. Ethernet port test

1) The development board has two 2.5G Ethernet ports, and the test method is the same for both ports. First, plug one end of the cable into the Ethernet port of the development



board, the other end of the cable into the router, and make sure the network is open.

2) The system will automatically assign an IP address to the Ethernet port via **DHCP** after startup.

3) The command to view the IP address in the Linux system of the development board is shown below:

```
(base) HwHiAiUser@orangepiaipro-20t:~$ ip addr show
.....
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether c0:74:2b:fe:3b:5a brd ff:ff:ff:ff:ff:ff
    inet 10.31.3.195/16 brd 10.31.255.255 scope global dynamic noprefixroute eth0
        valid_lft 43171sec preferred_lft 43171sec
    inet6 fe80::12fd:9a09:ed32:75d3/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether c0:74:2b:fe:3b:5b brd ff:ff:ff:ff:ff:ff
    inet 10.31.3.192/16 brd 10.31.255.255 scope global dynamic noprefixroute eth1
        valid_lft 43167sec preferred_lft 43167sec
    inet6 fe80::c274:2bff:fefe:3b5b/64 scope link
        valid_lft forever preferred_lft forever
.....
```

4) Test network connectivity commands are shown below, if you can ping through Baidu or other URLs that the development board's network connection is normal, **ping** command can be interrupted by the **Ctrl+C** shortcut to run.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ ping www.baidu.com -I eth0 #其中一个网口
(base) HwHiAiUser@orangepiaipro-20t:~$ ping www.baidu.com -I eth1 #另一个网口
PING www.a.shifen.com (183.2.172.185) from 192.168.2.100 eth0: 56(84) bytes of data.
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=1 ttl=52 time=10.0 ms
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=2 ttl=52 time=9.77 ms
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=3 ttl=52 time=9.94 ms
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=4 ttl=52 time=9.94 ms
^C
```



```
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 9.770/9.931/10.065/0.105 ms
```

3. 5. 2. WIFI connection test

Do not connect to WIFI by modifying the `/etc/network/interfaces` configuration file, there will be problems connecting to a WIFI network for use in this way.

3. 5. 2. 1. Connecting to WIFI via the nmcli command

1) Login to the Linux system first, there are three ways as below:

- a. If the board is connected to a network cable, you can log in to the Linux system remotely via ssh.
- b. If the development board is connected to a debugging serial port, you can log in to the Linux system using the serial terminal.
- c. If the development board is connected to an HDMI display, you can log in to the Linux system through the terminal of the HDMI display.

1) Then use the **nmcli dev wifi** command to scan for WIFI hotspots around you.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ nmcli dev wifi
```

2) Then use the **nmcli** command to connect to the scanned WIFI hotspot where:

- a. **wifi_name** should be the name of the WIFI hotspot you want to connect to.
- b. **wifi_passwd** needs to be replaced with the password of the WIFI hotspot you want to connect to.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo nmcli dev wifi connect wifi_name password
wifi_passwd
Device 'wlan0' successfully activated with 'cf937f88-ca1e-4411-bb50-61f402eef293'.
```

3) Through **ip addr show wlan0** You can view the IP address of the wifi by using the ip addr show wlan0 command.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ ip a s wlan0
4: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
    link/ether 54:f2:9f:7b:ba:36 brd ff:ff:ff:ff:ff:ff
    inet 10.31.2.93/16 brd 10.31.255.255 scope global dynamic noprefixroute wlan0
```



```
valid_lft 43191sec preferred_lft 43191sec
inet6 fe80::5297:7036:a33c:bb93/64 scope link noprefixroute
valid_lft forever preferred_lft forever
```

4) The connectivity of a wifi network can be tested by using the **ping** command. The **ping** command can be interrupted by the **Ctrl+C** shortcut.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ ping www.orangepi.org -I wlan0
PING www.orangepi.org (123.57.147.237) from 10.31.2.93 wlan0: 56(84) bytes of data.
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=1 ttl=53 time=47.1 ms
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=2 ttl=53 time=44.3 ms
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=3 ttl=53 time=45.0 ms
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=4 ttl=53 time=71.0 ms
^C
--- www.orangepi.org ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 44.377/51.902/71.082/11.119 ms
```

3.5.2.2. Method of connecting to WIFI via nmtui graphical method

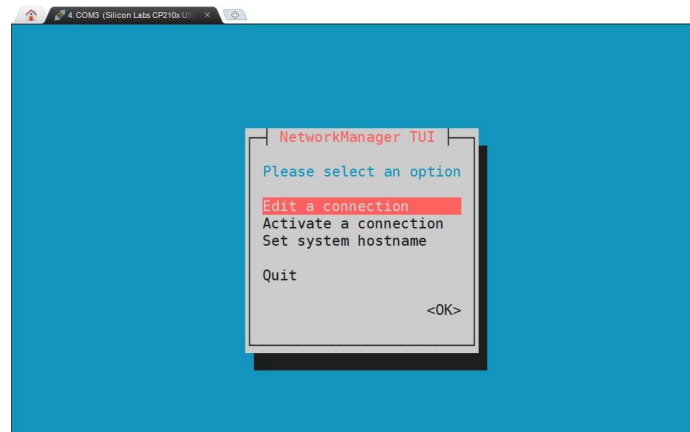
1) Login to the Linux system first, there are three ways as below:

- a. If the board is connected to a network cable, you can log in to the Linux system remotely via ssh.
- b. If the development board is connected to a debugging serial port, you can log in to the Linux system using the serial port terminal.
- c. If the development board is connected to a HDMI display, you can log in to the Linux system through the HDMI display terminal.

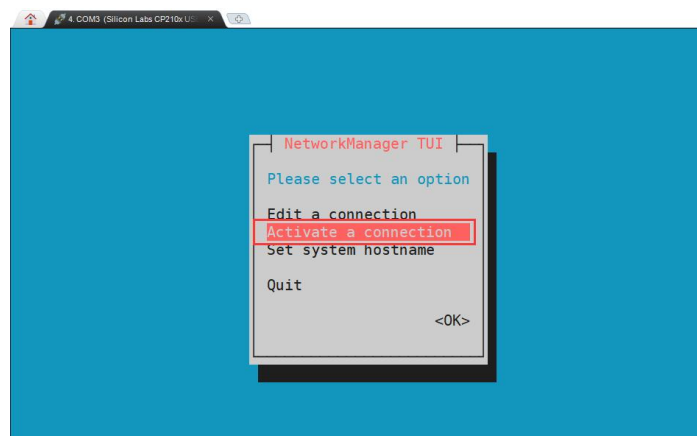
2) Then enter nmtui command in the command line to open the interface of wifi connection.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo nmtui
```

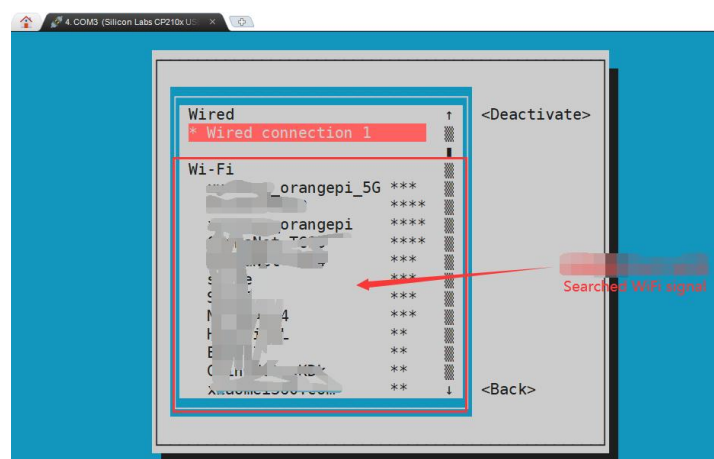
3) Entering the nmtui command opens the screen as shown below:



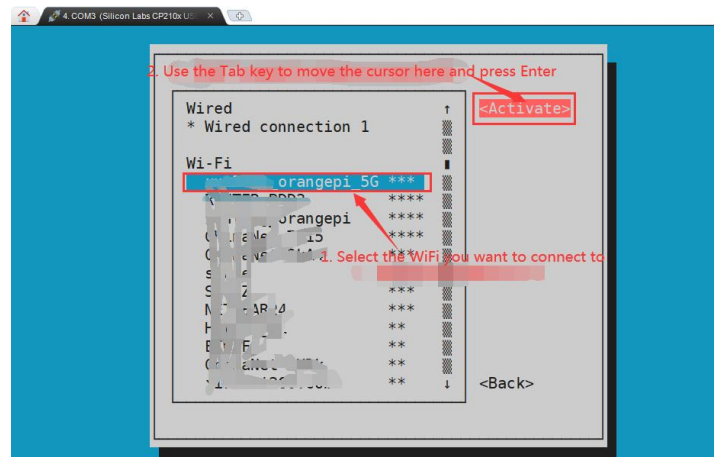
4) Select **Activate a connect** and enter.



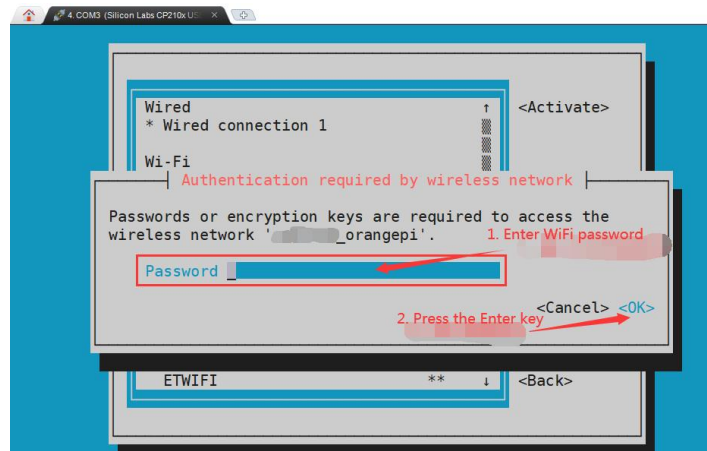
5) Then you can see all the searched WIFI hotspots.



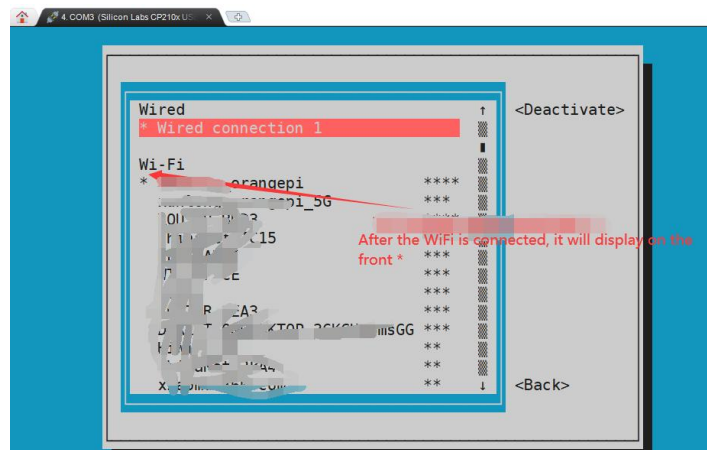
6) Select the WIFI hotspot you want to connect to, then use the Tab key to move the cursor to **Activate** and press Enter.



7) Then the dialog box for entering password will pop up, enter the corresponding password in **Password** and enter to start connecting to WIFI.



8) A "*" will be displayed in front of the connected WIFI name after successful WIFI connection.





9) The **ip a s wlan0** command allows you to view the IP address of the wifi.

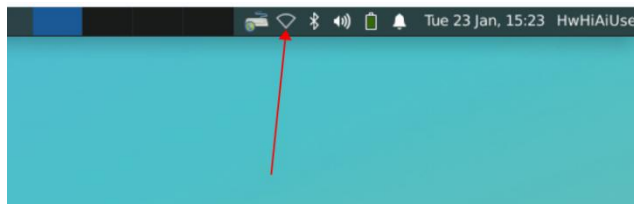
```
(base) HwHiAiUser@orangepiaipro-20t:~$ ip a s wlan0
4: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
    link/ether 54:f2:9f:7b:ba:36 brd ff:ff:ff:ff:ff:ff
    inet 10.31.2.93/16 brd 10.31.255.255 scope global dynamic noprefixroute wlan0
        valid_lft 43003sec preferred_lft 43003sec
    inet6 fe80::5297:7036:a33c:bb93/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

10) The connectivity of a wifi network can be tested using the **ping** command. The **ping** command can be interrupted from running by the **Ctrl+C** shortcut.

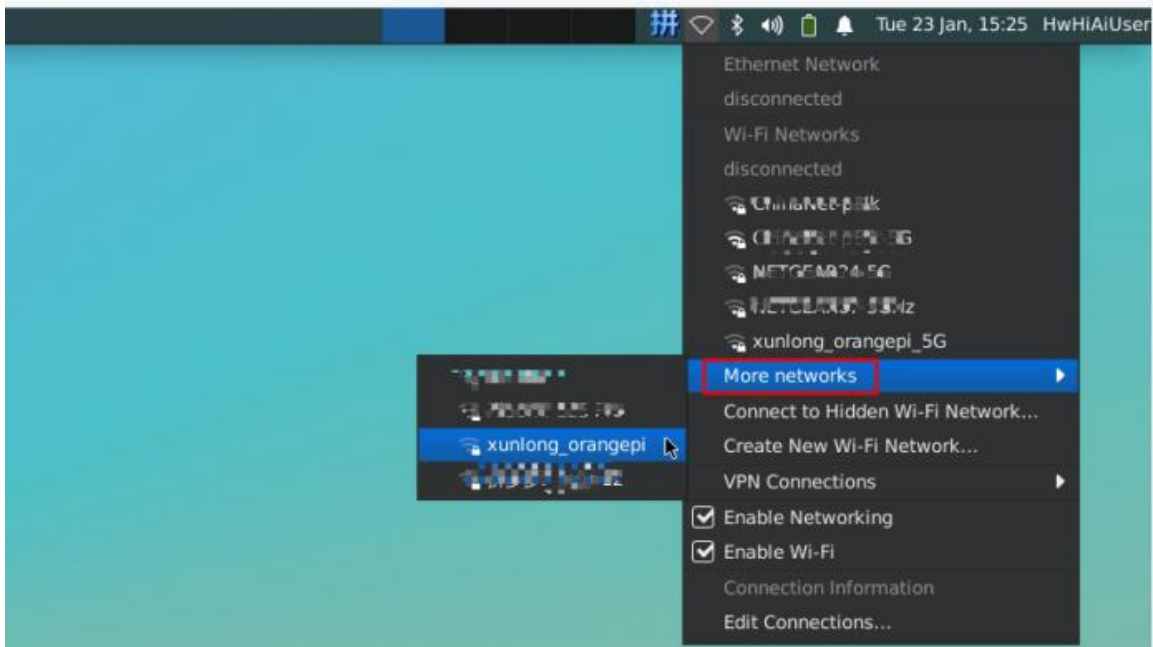
```
(base) HwHiAiUser@orangepiaipro-20t:~$ ping www.orangeapi.org -I wlan0
PING www.orangeapi.org (123.57.147.237) from 10.31.2.93 wlan0: 56(84) bytes of data.
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=1 ttl=53 time=47.1 ms
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=2 ttl=53 time=44.3 ms
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=3 ttl=53 time=45.0 ms
64 bytes from 123.57.147.237 (123.57.147.237): icmp_seq=4 ttl=53 time=71.0 ms
^C
--- www.orangeapi.org ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 44.377/51.902/71.082/11.119 ms
```

3. 5. 2. 3. Test Methods for Desktop Images

1) First click on the Network Configuration icon in the top right corner of your desktop.



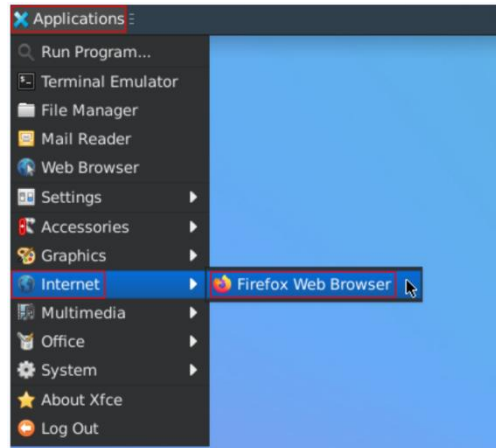
2) Click More networks in the pop-up drop-down box to see all the scanned **More networks**, and then select the WIFI hotspot you want to connect.



3) Then enter the password of the WIFI hotspot and click **Connect** to start connecting to the WIFI.



4) After connecting the WIFI, you can open the browser to check if you can access the internet, the entrance of the browser is shown below:



5) If you can open other web pages after opening the browser, it means the WIFI connection is normal.



OrangePi Alpro (20T)



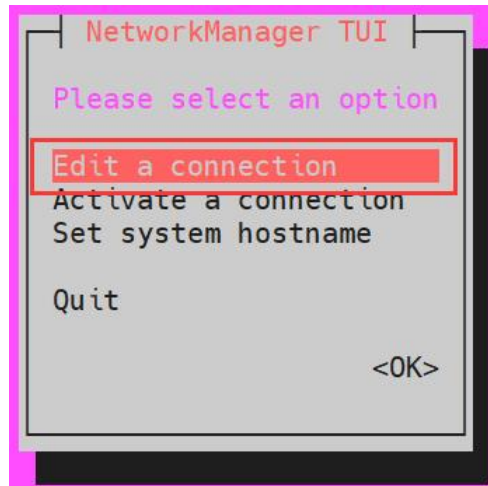
3. 5. 3. Methods for setting static IP addresses

3. 5. 3. 1. Using the nmtui command to set a static IP address

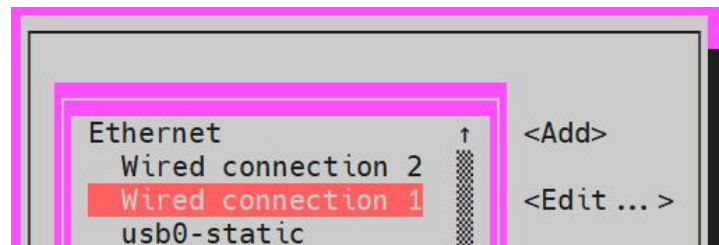
1) First run the **nmtui** command.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ nmtui
```

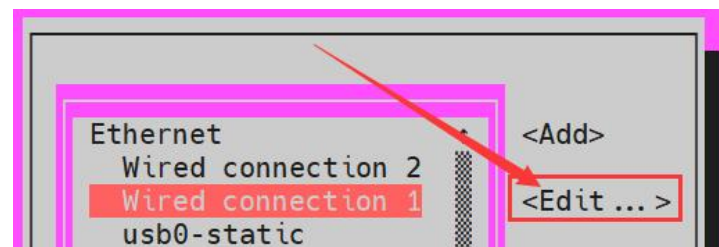
2) Then select **Edit a connection** and press enter.



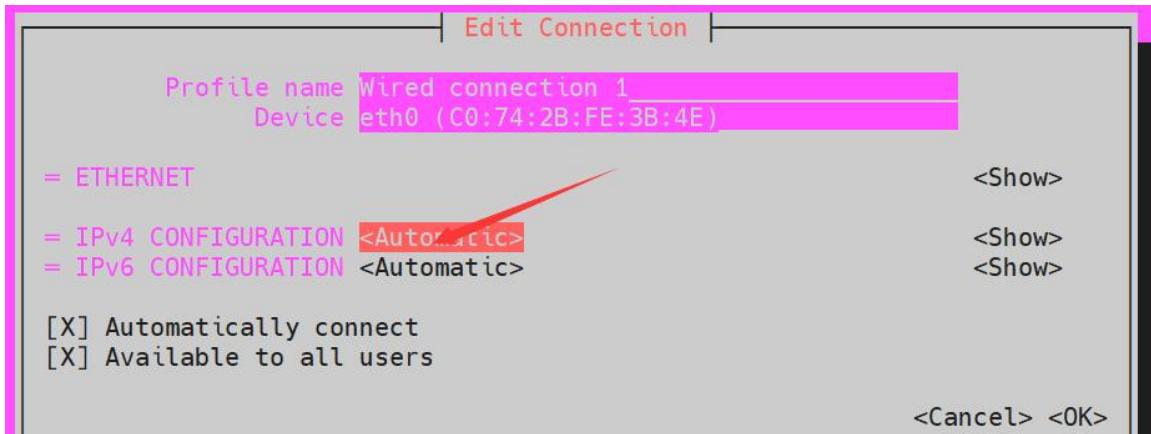
3) Then select the network interface where you need to set the static IP address, for example, to set the static IP address of the 2.5G Ethernet interface select **Wired connection 1** or **Wired connection 2**, and for the virtual usb0 network port of the Type-C interface select **usb0-static**.



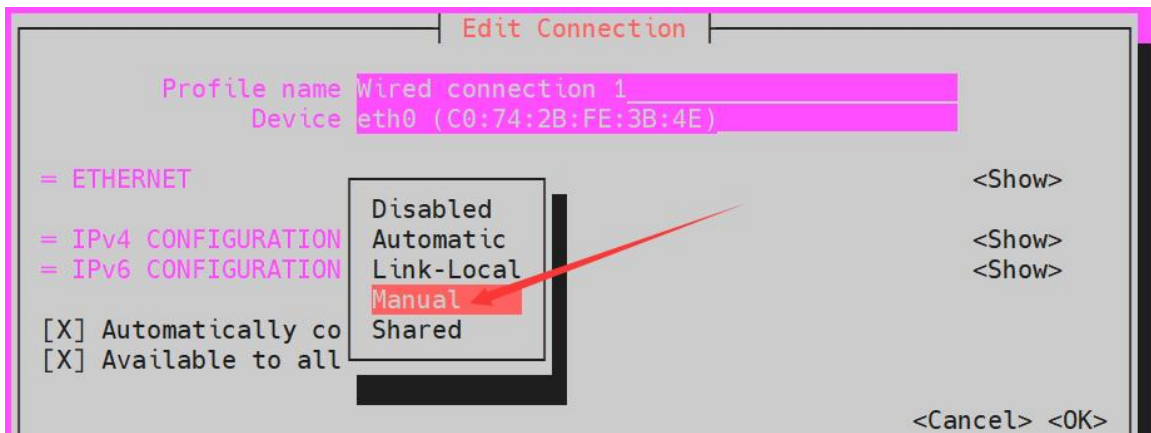
4) Then select the network port you want to set up, then select **Edit** via **Tab** key and press Enter.



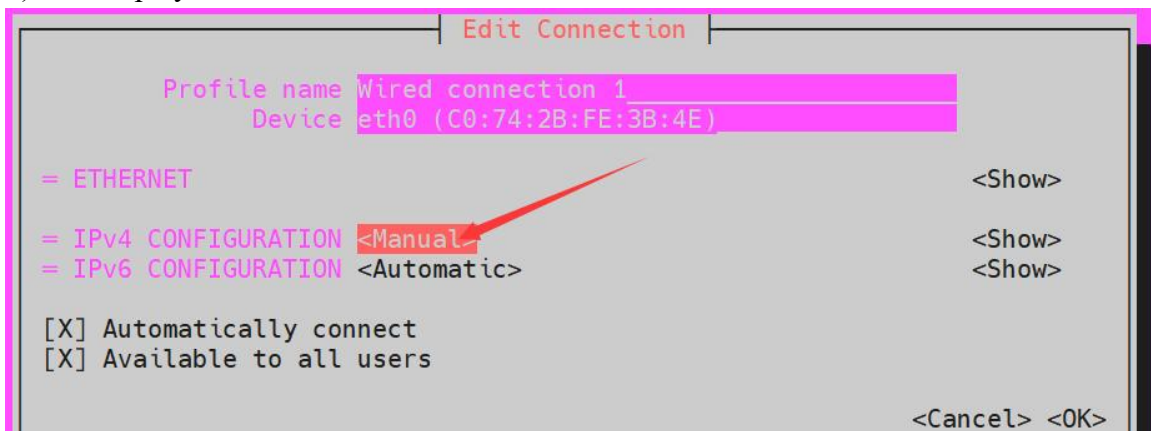
5) Then move the cursor to the **<Automatic>** position for IPv4 configuration by using the Tab key as shown below.



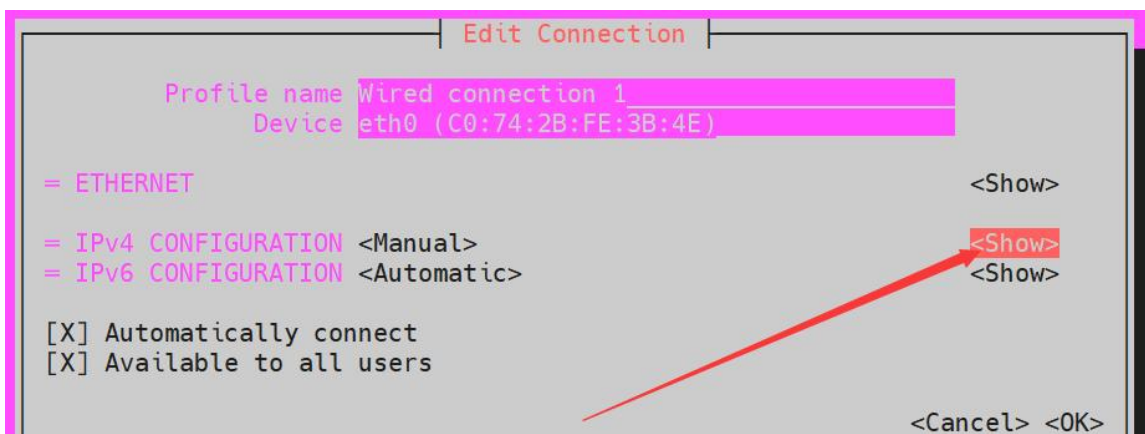
6) Then enter, select **Manual** by using the up and down arrow keys, and then enter to confirm.



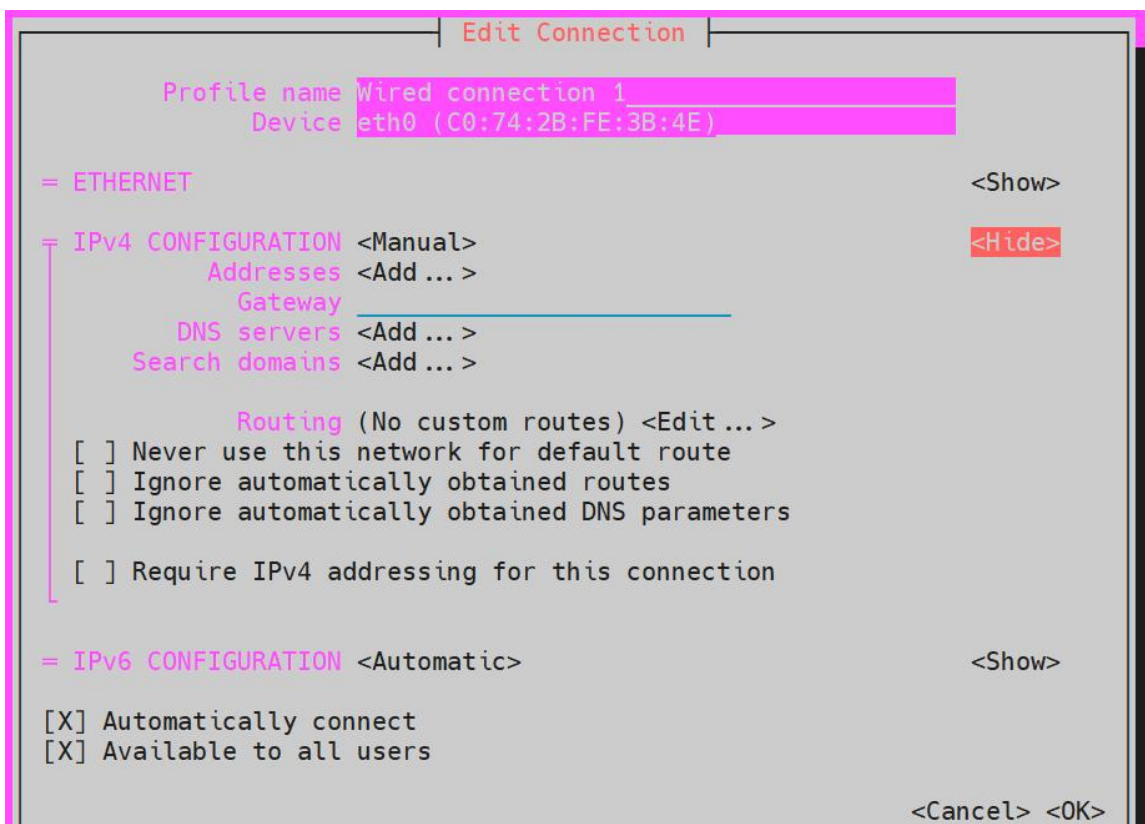
7) The display after selection is shown below:



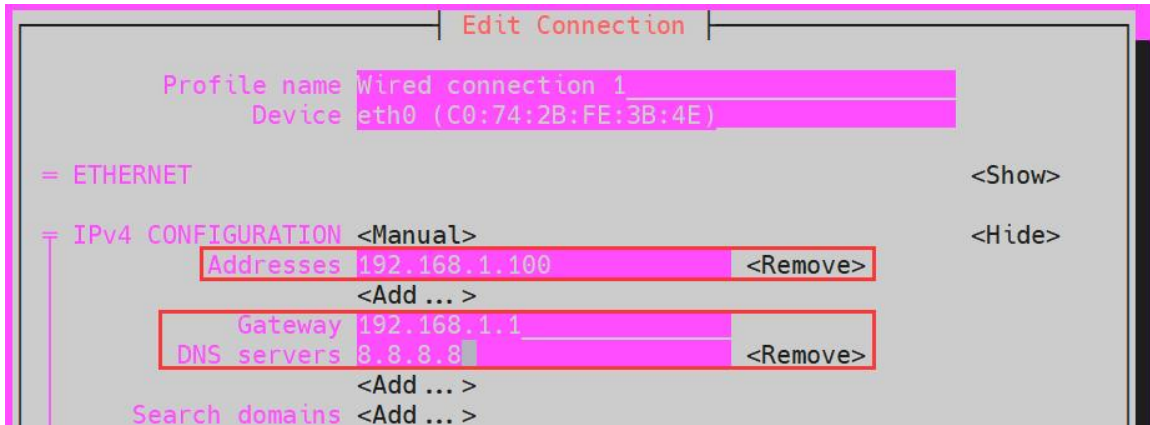
8) Then move the cursor to **<Show>** by using the Tab key.



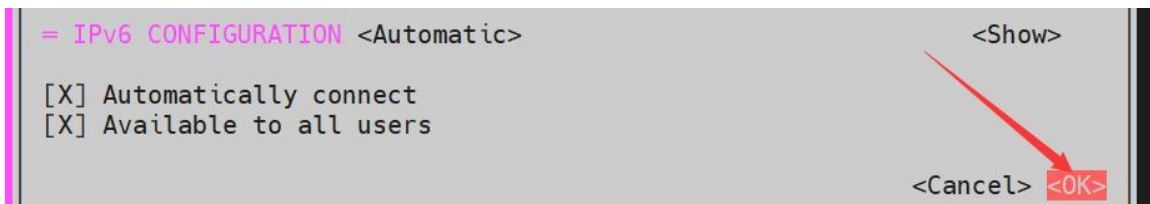
9) Then enter, the following setting interface will pop up after enter.



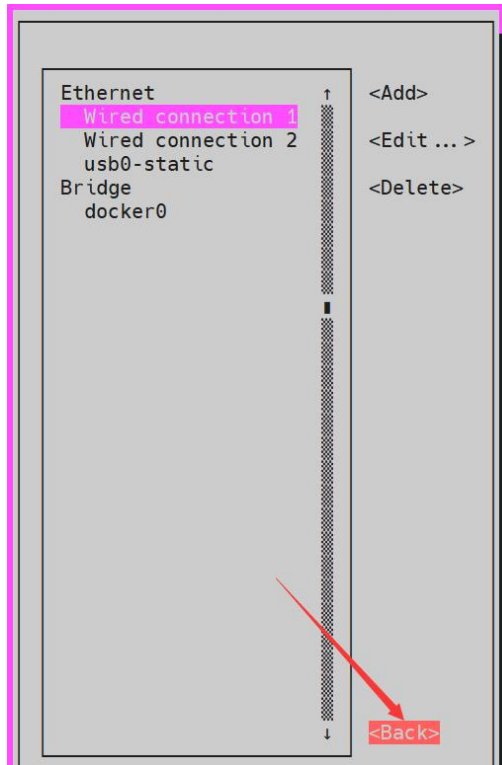
10) Then you can set the IP address (Addresses), gateway (Gateway) and DNS server address in the location shown in the figure below (**there are many other setup options in there, please explore on your own**), please set it up according to your own specific needs, and the values set in the figure below are just an example.



11) Move the cursor to **<OK>** in the lower right corner after setting, then enter to confirm.



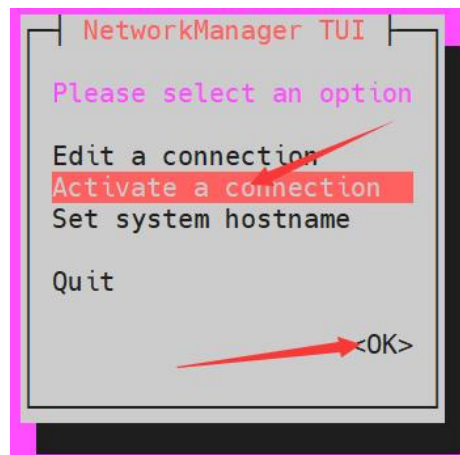
12) Then click **<Back>** to go back to the previous selection screen.



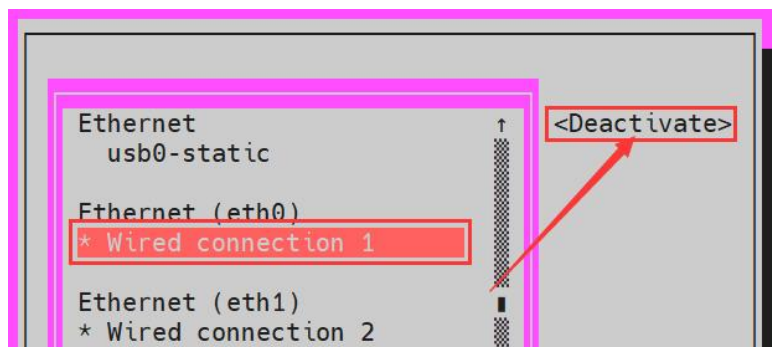
13) Then select **Activate a connection** and move the cursor to **<OK>** and finally click



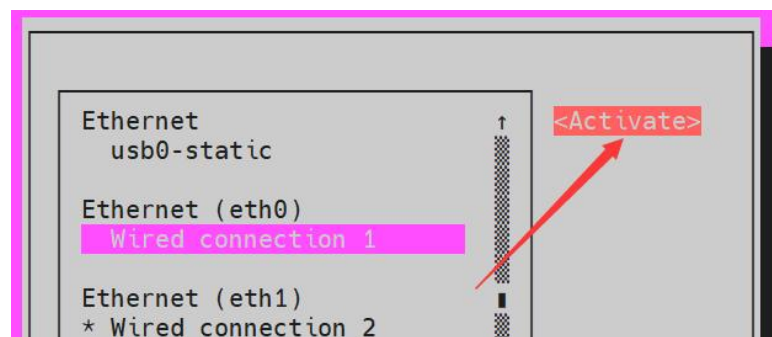
enter.



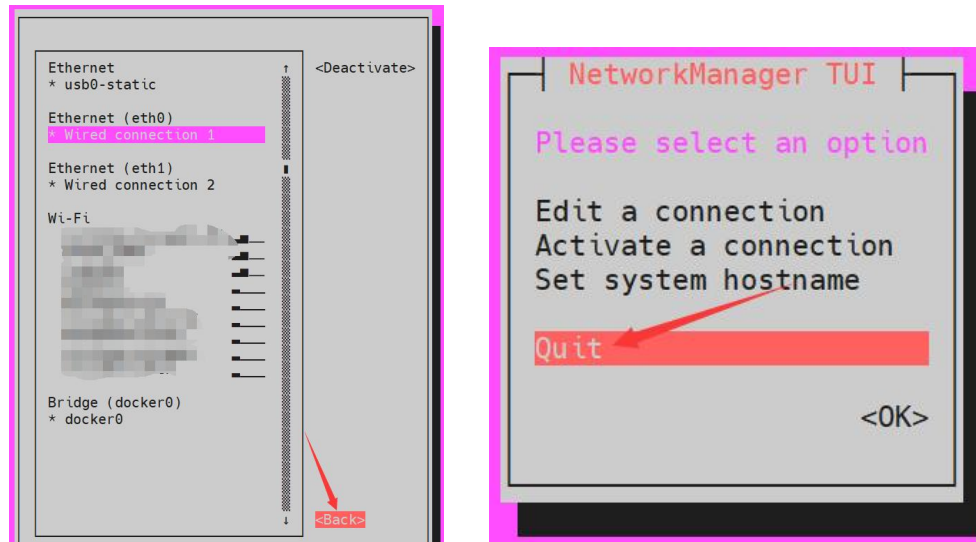
14) Then select the network interface that needs to be set, such as **Wired connection 1**, and then move the cursor to **<Deactivate>** and press enter to disable **Wired connection 1**.



15) Then re-select and enable **Wired connection 1**, so that the static IP set earlier will take effect.



16) You can then exit the nmtui with the **<Back>** and **Quit** buttons.



17) Then make sure that the network cable is connected, and then through the **ip addr show**, you can see that the IP address of the corresponding network port has been changed to a static IP address.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ ip a s eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether c0:74:2b:fe:3b:4e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.100/24 brd 192.168.1.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::70f3:e511:7706:7aa5/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

3. 5. 3. 2. Using the nmcli command to set a static IP address

1) Use the nmcli command to set the static IP address to **192.168.1.100** and the gateway to **192.168.1.1** with the command:

- a. Set static IP address for eth0 on 2.5G port

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo nmcli con add type ethernet ifname
eth0 con-name eth0-static ip4 192.168.1.100/24 gw4 192.168.1.1
```

- b. Set static IP address for eth1 on 2.5G port

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo nmcli con add type ethernet ifname
eth1 con-name eth1-static ip4 192.168.1.100/24 gw4 192.168.1.1
```



c. Setting the static IP address of Type-C's usb0 network port

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo nmcli con add type ethernet ifname
usb0 con-name usb0-static ip4 192.168.1.100/24 gw4 192.168.1.1
```

2) Make sure that the network cable is connected, and then use the **ip addr show eth0** command to see that the IP address has been set to the desired value. (Please change eth1 and usb0 to the corresponding commands)

```
(base) HwHiAiUser@orangepiaipro-20t:~$ ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether c0:74:2b:fe:3b:4e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.100/24 brd 192.168.1.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::d626:a148:dda0:8a9c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

3. 6. SSH remote login development board

By default, Linux systems enable ssh remote login and allow the root user to log in. Before logging in through ssh, you must first ensure that the Ethernet or WiFi network is connected, and then use the **ip addr** command or check the router to obtain the IP address of the development board.

3. 6. 1. SSH remote login development board under Ubuntu

1) Get the IP address of the development board.

2) Then you can remotely log in to the Linux system through the ssh command

```
test@test:~$ ssh root@192.168.2.xxx          (Need to be replaced with the IP address of the
development board, do not copy it)
root@192.168.2.xx's password:                (Enter the password here. The default password is
Mind@123)
```

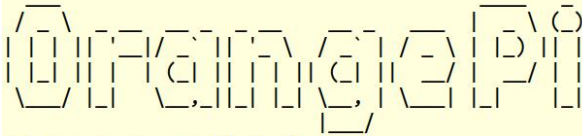
Please note that when you enter the password, **the specific content of the password will not be displayed on the screen**. Please do not think that there is any malfunction. Just press Enter after entering it.



If the connection is refused, as long as you are using the image provided by Orange Pi, **do not doubt whether the Mind@123 password is incorrect, but look for other reasons.**

3) After successfully logging into the system, the display is as shown below:

```
orangePi@orangePi-M600:~$ ssh root@192.168.2.100
root@192.168.2.100's password:
```



Welcome to Orange Pi Ai Pro

This system is based on Ubuntu 22.04 LTS (GNU/Linux 5.10.0+ aarch64)

This system is only applicable to individual developers and cannot be used for commercial purposes.

By using this system, you have agreed to the Huawei Software License Agreement.

Please refer to the agreement for details on <https://www.hiascend.com/software/protocol>

Last login: Fri Jan 26 16:55:15 2024 from 192.168.2.220

-bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)

(base) root@orangepiaipro:~#

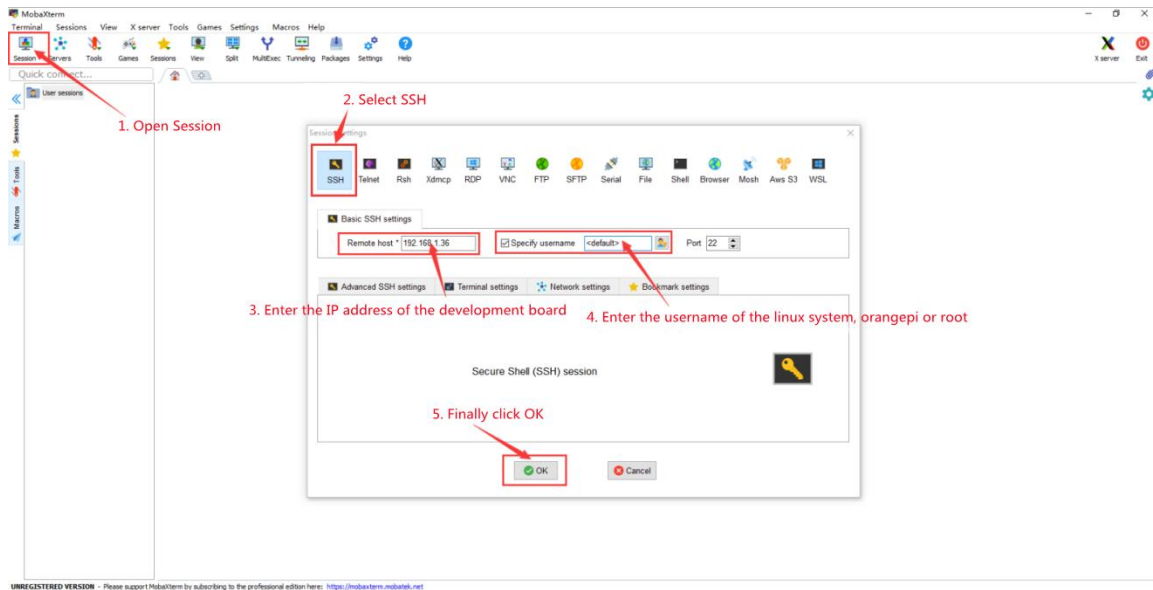
3. 6. 2. SSH remote login development board under Windows

1) First, obtain the IP address of the development board.

2) Under Windows, you can use MobaXterm to remotely log in to the development board.

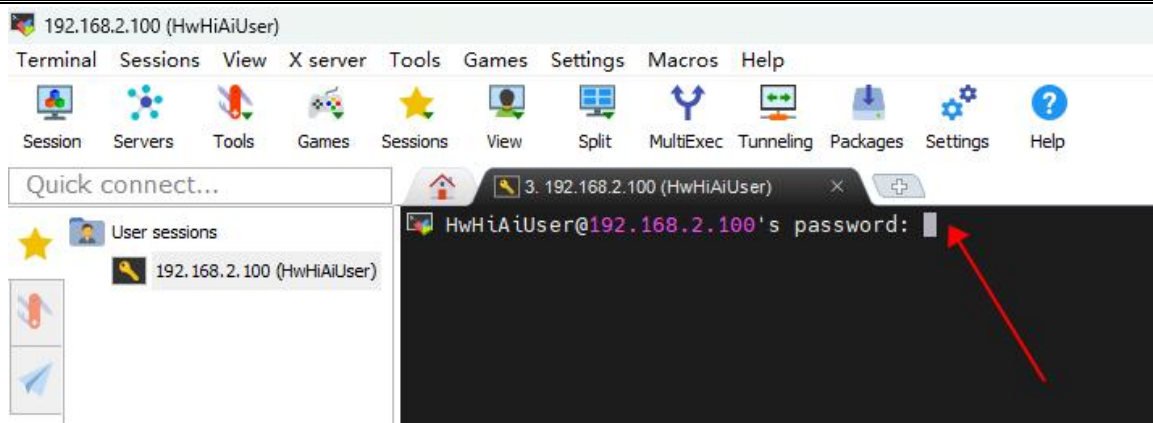
First, create a new ssh session.

- a. Open **Session**.
- b. Then select **SSH** in **Session Setting**.
- c. Then enter the IP address of the development board in **Remote host**.
- d. Then enter the Linux system user name **root** or **HwHiAiUser** in **Specify username**.
- e. Finally, click **OK**.

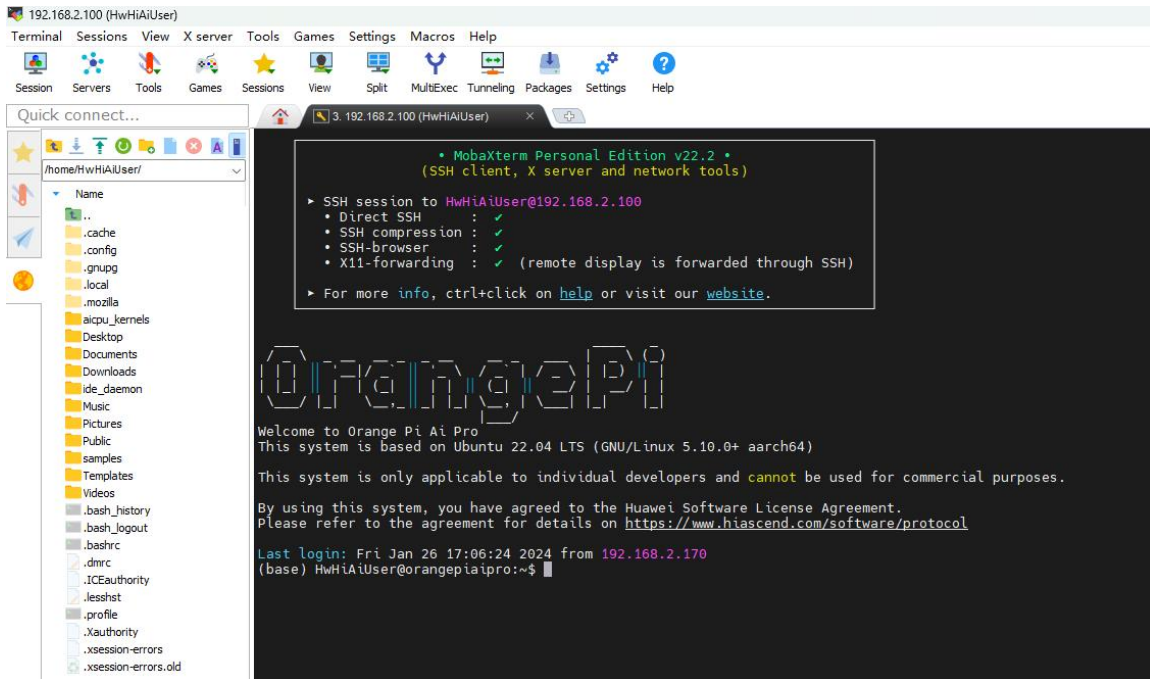


3) You will be prompted to enter a password. The default passwords for **root** and **HwHiAiUser** are both **Mind@123**.

Please note that when you enter the password, the specific content of the password will not be displayed on the screen. Please do not think that there is any malfunction. Just press Enter after entering it.



4) The display after successfully logging into the system is as shown below



3. 7. HDMI interface usage instructions

3. 7. 1. Instructions for displaying Linux desktop via HDMI

The development board has two HDMI2.0 interfaces. Currently, only HDMI0 supports displaying the Linux system desktop. For detailed instructions on how to display the Linux system desktop through HDMI0, please refer to the instructions in the section [How to log in to the Linux system desktop](#).

3. 7. 2. How to use the HDMI interface to display images and play audio

When the Linux desktop system is shut down, HDMI0 and HDMI1 can also be used to output images in NVR secondary development scenarios.

The method to test HDMI0 outputting a picture is as follows:

- 1) First connect the HDMI0 port to the HDMI display.
- 2) Then switch to the root user and enter the path where the HDMI0 test program is located.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo -i
(base) root@orangepiaipro-20t:~# cd /opt/opi_test/hdmi0_pic
```



```
(base) root@orangepiaipro-20t:/opt/opi_test/hdmi0_pic# ls
sample_hdmi test.sh update_dt.sh ut_1920x1080_nv12.yuv
```

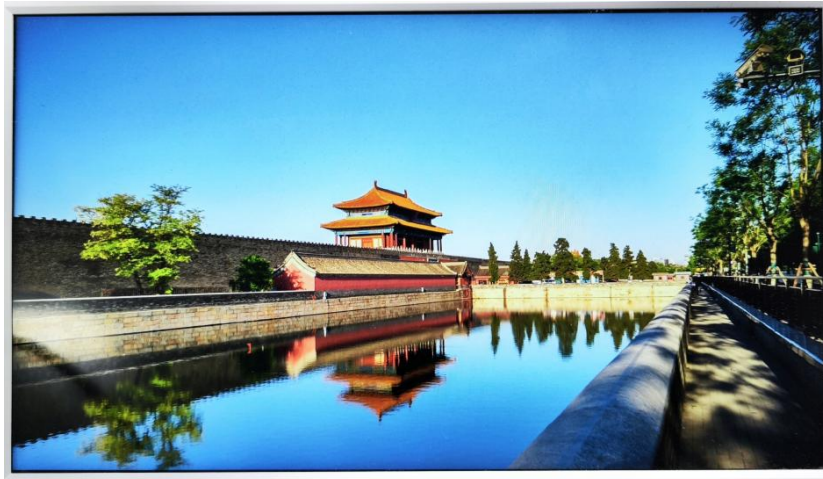
3) Then run the **update_dt.sh** script to update dt.img (turn off the drm configuration and turn on the vdp configuration). After the **update_dt.sh** script is run, the Linux system will automatically restart.

```
(base) root@orangepiaipro-20t:/opt/opi_test/hdmi0_pic# ./update_dt.sh
```

4) After restarting, enter the path where the HDMI0 test program is located again, and then run the **test.sh** script to play a picture to the HDMI display (10 seconds by default), and at the same time play an audio to the HDMI display. If the HDMI display supports audio playback, you can also hear the sound.

```
(base) root@orangepiaipro-20t:/opt/opi_test/hdmi0_pic# ./test.sh
```

5) The picture displayed by HDMI is as follows:



The method of testing HDMI1 to output a picture is the same as HDMI0, except that the path of the test program is:

```
/opt/opi_test/hdmi1_pic
```

After testing the HDMI playback image, the method to switch back to HDMI0 to display the Linux system desktop is as follows:

1) First switch to the root user and enter the **/opt/opi_test/hdmi_desktop** directory.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo -i
(base) root@orangepiaipro-20t:~# cd /opt/opi_test/hdmi_desktop
(base) root@orangepiaipro-20t:/opt/opi_test/hdmi_desktop# ls
```




```
update_dt.sh
```

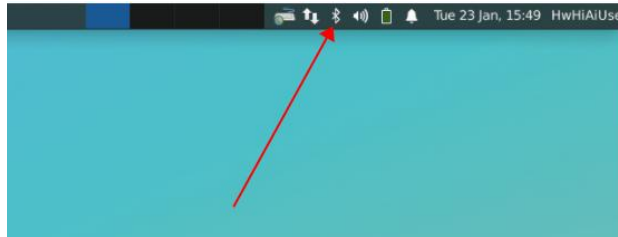
2) Then run the **update_dt.sh** script to update dt.img (turn off the vdp configuration and turn on the drm configuration). After the **update_dt.sh** script is run, the Linux system will automatically restart.

```
(base) root@orangepiaipro-20t:/opt/opi_test/hdmi_desktop# ./update_dt.sh
```

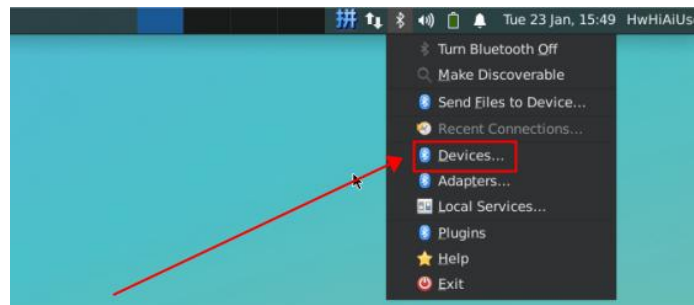
3) After restarting, you can see that the HDMI monitor will display the desktop of the Linux system.

3. 8. How to use Bluetooth

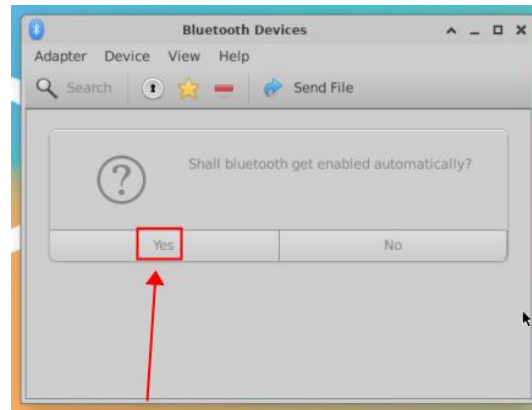
1) Click the Bluetooth icon in the upper right corner of the desktop



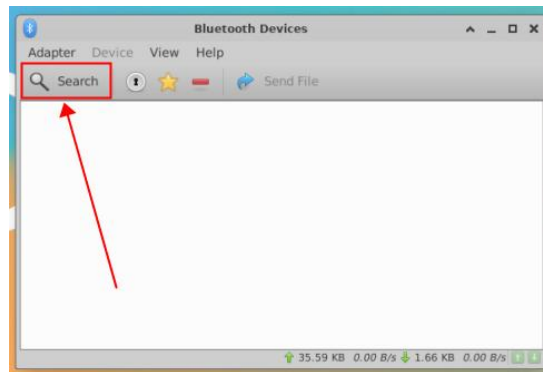
2) Then open the configuration interface of the Bluetooth device



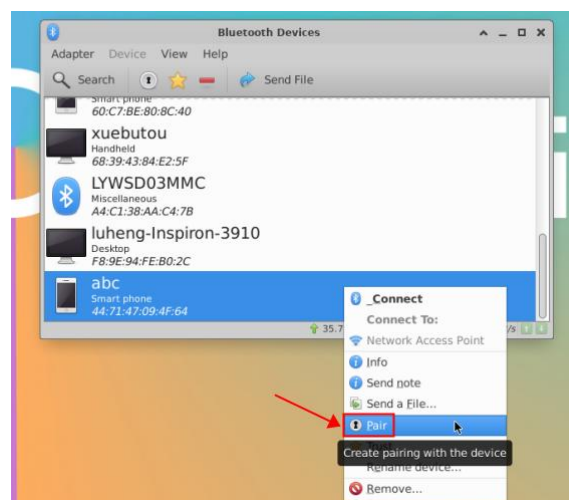
3) Then select **Yes** in the following interface



4) Click **Search** to start scanning for surrounding Bluetooth devices.



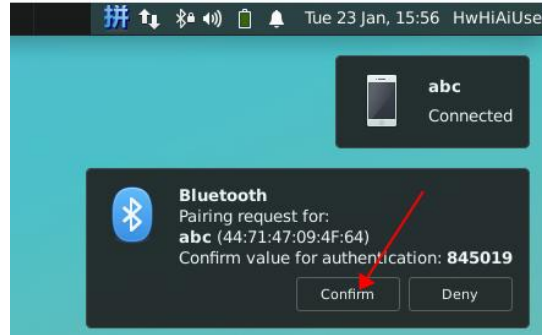
5) Then select the Bluetooth device you want to connect to, and right-click the mouse to pop up the operation interface for this Bluetooth device. Select **Pair** to start pairing. The demonstration here is pairing with an Android phone.



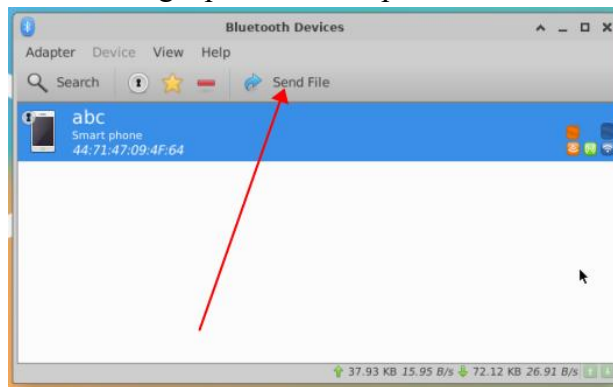
6) During pairing, a pairing confirmation box will pop up in the upper right corner of the



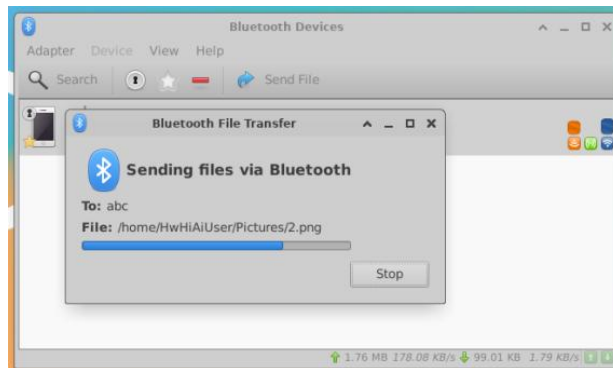
desktop. Select **Confirm**. Confirmation is also required on the phone.



7) After pairing with the phone, you can select the paired Bluetooth device and then select **Send a File** to start sending a picture to the phone.



8) The interface for sending pictures is as follows:



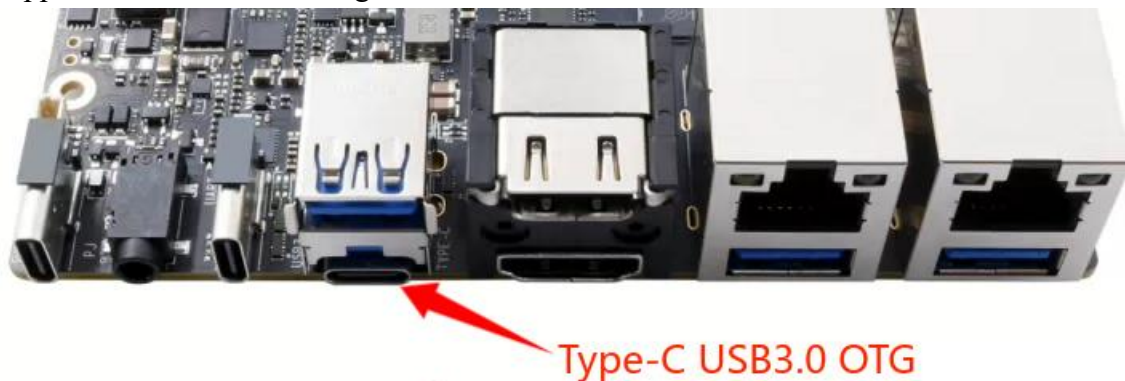
3. 9. USB interface test

3. 9. 1. Type-C USB3.0 interface Host mode usage instructions

The development board has a Type-C USB3.0 OTG interface, which is located as



shown in the figure below. This interface supports both Host mode and Device mode, and supports automatic switching between the two modes.

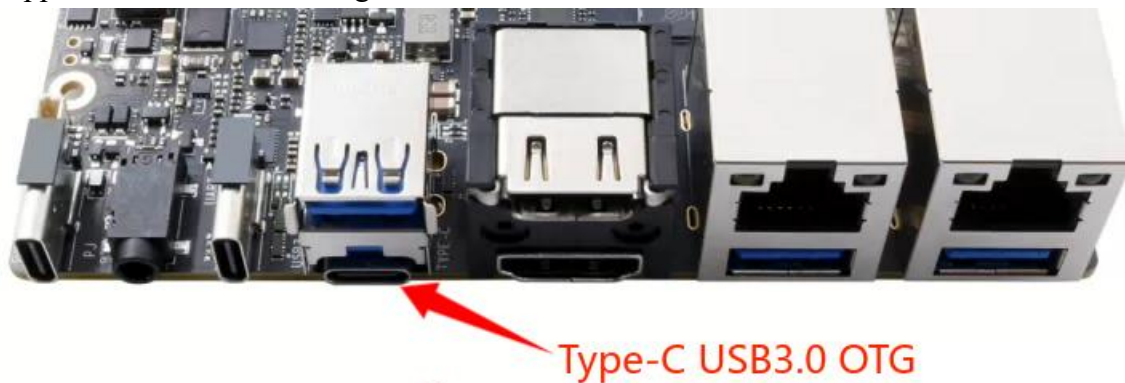


When testing the host function of the Type-C interface, you can use the Type-C to USB adapter cable shown in the figure below to connect a USB storage device or a USB device such as a mouse and keyboard.



3. 9. 2. Instructions for using the Type-C USB3.0 interface in Device mode

The development board has a Type-C USB3.0 OTG interface, which is located as shown in the figure below. This interface supports both Host mode and Device mode, and supports automatic switching between the two modes.



By default, the Linux system sets the Type-C Device as a virtual network port. The



steps to test the Device network port function are as follows:

1) Use the Type-C cable shown in the figure below to connect the development board to the Ubuntu 22.04 computer.



2) The virtual network port of Type-C is usb0. The Linux system sets a static IP address of **192.168.0.2** by default. The current IP address can be viewed through the `ifconfig usb0` command. For how to modify the static IP address of usb0, please refer to the instructions in the section [How to set a static IP address](#).

```
(base) root@orangepiaipro-20t:~# ifconfig usb0
usb0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.0.2 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::9391:bd5c:450a:857e prefixlen 64 scopeid 0x20<link>
    ether 36:2c:1f:7c:03:f1 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

3) After connecting the Type-C interface of the development board to the Ubuntu computer using a Type-C cable, if you use the `lsusb` command to see the following device information, it means the connection is normal.

```
orangepi@orangepi:~$ lsusb | grep Huawei
Bus 001 Device 022: ID 12d1:107e Huawei Technologies Co., Ltd. P10 smartphone
```

4) You will see an additional network port device in the Ubuntu computer.

```
orangepi@orangepi:~$ ifconfig
.....
```



```
enx1e71fffb0420: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 1e:71:ff:fb:04:20 txqueuelen 1000 (Ethernet)
    RX packets 38 bytes 3910 (3.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 95 bytes 22687 (22.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
.....
```

5) Then you can set a static IP address for this network port in the Ubuntu computer. You can use the `nmtui/nmcli` command to set it, or you can use the `ifconfig` command to temporarily set the IP address. (Please modify the network port name in the following command according to the actual situation)

```
orangepi@orangepi:~$ sudo ifconfig enx1e71fffb0420 192.168.0.3
```

6) Then you can use the `ping` command to test whether you can ping the development board.

```
orangepi@orangepi:~$ ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=1.14 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.148 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.170 ms
```

3.9.3. Connect a USB mouse or keyboard to test

The development board has three USB3.0 HOST interfaces for connecting the mouse and keyboard. Insert the USB mouse and keyboard into the USB 3.0 interface of the development board, and then connect the HDMI0 interface of the development board to the HDMI display. After the HDMI display displays the desktop of the Linux system, you can use the mouse and keyboard to operate the Linux system.





3.9.4. USB camera test

1) First, insert the USB camera into the USB3.0 HOST interface of the development board.



2) Then use the **v4l2-ctl** command to see the device node information of the USB camera is **/dev/video0**

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo apt-get update
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo apt-get install -y v4l-utils
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo v4l2-ctl --list-devices
Q8 HD Webcam: Q8 HD Webcam (usb-xhci-hcd.3.auto-1):
    /dev/video0
    /dev/video1    #This is used to collect metadata, ignore it for now.
    /dev/media0
```

Note that the l in v4l2 is a lowercase letter l, not the number 1.

In addition, the serial number of the video is not always video0, please refer to the actual one you see.

3) Use fswebcam to test the USB camera

a. Install fswebcam

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo apt-get update
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo apt-get install -y fswebcam
```

b. After installing fswebcam, you can use the following command to take photos

- a) The -d option is used to specify the device node of the USB camera
- b) --no-banner is used to remove the watermark of the photo
- c) The -r option is used to specify the resolution of the photo
- d) -S option is used to set the number of frames to skip ahead
- e) ./image.jpg is used to set the name and path of the generated photo



```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo fswebcam -d /dev/video0 --no-banner -r 1280x720 -S 5 ./image.jpg
```

- c. Then you can directly open and view the captured pictures on the Linux desktop via the HDMI display.

4) Use the built-in USB Camera sample code to test the USB camera.

- a. First enter the path of the USB Camera sample code.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo -i
(base) root@orangepiaipro-20t:~# cd /opt/opi_test/USBCamera
(base) root@orangepiaipro-20t:/opt/opi_test/USBCamera# ls
main  main.cpp  readme.md
```

- b. Then run the following command to take a photo using the USB camera:

```
(base) root@orangepiaipro-20t:/opt/opi_test/USBCamera# ./main /dev/video0
```

- c. After successful running, a **out.yuv** file with yuyv422 format and 1280*720 resolution will be generated in the USB Camera sample directory.

```
(base) root@orangepiaipro-20t:/opt/opi_test/USBCamera# ls
main  main.cpp  out.yuv  readme.md
```

- d. Then use the following command in the Linux desktop to view the contents of the **out.yuv** file.

```
(base) root@orangepiaipro-20t:/opt/opi_test/USBCamera# ffplay -pix_fmt yuyv422 -video_size 1280*720 out.yuv
```

3. 9. 5. USB Audio Test

- 1) First, you need to prepare a headset with a USB interface with recording function.



- 2) Then plug the USB headset into the USB port of the developer.

- 3) Then use the **arecord -l** command to view the recording device number, as shown in the output below, where the 0 in **card 0** indicates that the recording device number is 0.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ arecord -l
```




```
**** List of CAPTURE Hardware Devices ****
```

```
card 0: Audio [AB13X USB Audio], device 0: USB Audio [USB Audio]
```

```
Subdevices: 1/1
```

```
Subdevice #0: subdevice #0
```

4) Then enter the path of the USB audio test code.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo -i
```

```
(base) root@orangepiaipro-20t:~# cd /opt/opi_test/USBAudio
```

```
(base) root@orangepiaipro-20t:/opt/opi_test/USBAudio# ls
```

```
Readme.md main main.c
```

5) Then use the following command to record an audio clip using the USB audio device. **0** represents the recording device number, which needs to be filled in according to the actual situation.

```
(base) root@orangepiaipro-20t:/opt/opi_test/USBAudio# ./main plughw:0
```

6) After the recording is finished, enter **over** in the terminal interface to exit the recording.

```
(base) root@orangepiaipro-20t:/opt/opi_test/USBAudio# ./main plughw:0
```

```
Start record!
```

```
over          #Enter over to end recording audio.
```

```
(base) root@orangepiaipro-20t:/opt/opi_test/USBAudio#
```

7) After the recording is successful, an audio file **audio.pcm** will be generated in the USBAudio sample directory.

```
(base) root@orangepiaipro-20t:/opt/opi_test/USBAudio# ls *.pcm
```

```
audio.pcm
```

8) Then make sure you can see the USB audio device using the **aplay -l** command.

```
(base) root@orangepiaipro-20t:/opt/opi_test/USBAudio# aplay -l
```

```
**** List of PLAYBACK Hardware Devices ****
```

```
card 0: Audio [AB13X USB Audio], device 0: USB Audio [USB Audio]
```

```
Subdevices: 0/1
```

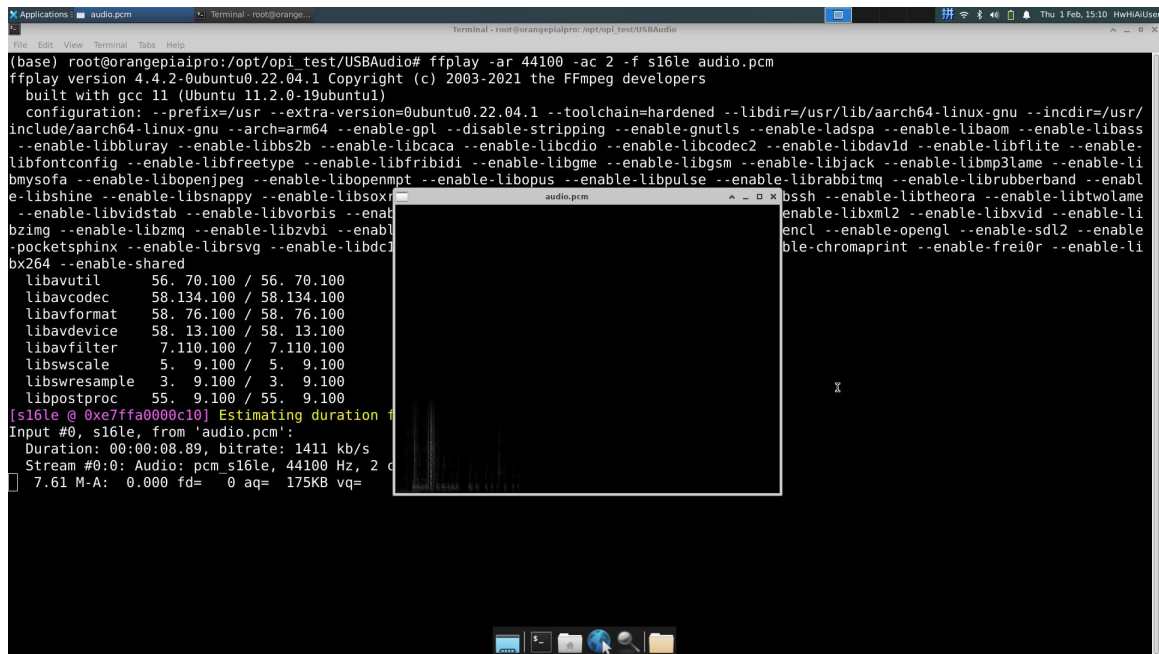
```
Subdevice #0: subdevice #0
```

9) Then in the Linux system desktop, use the following command to play the recorded



audio to the USB headset.

```
(base) root@orangepiaipro-20t:/opt/opi_test/USBAudio# ffplay -ar 44100 -ac 2 -f s16le audio.pcm
```



3. 10. Audio Test

The Linux kernel does not have ALSA audio drivers for headphones and HDMI. These drivers are still under development. Currently, you can only use audio sample codes to test the audio playback of headphones and HDMI and the recording function of the onboard MIC.

3. 10. 1. Headphone jack audio playback test

1) First, plug the earphone into the 3.5mm headphone jack of the development board.



2) Then go to the directory where the audio test program is located.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo -i
(base) root@orangepiaipro-20t:~# cd /opt/opi_test/audio
(base) root@orangepiaipro-20t:/opt/opi_test/audio# ls
```



```
capture.sh play.sh qzgy_48k_16_mono_30s.pcm sample_audio
```

3) Then use the following command to play the test audio to the headphones.

```
(base) root@orangepiaipro-20t:/opt/opi_test/audio# ./sample_audio play 2 qzgy_48k_16_mono_30s.pcm
```

3. 10. 2. HDMI audio playback test

Please refer to the section "[How to use the HDMI interface to display images and play audio](#)" for instructions.

3. 10. 3. Headphone MIC recording test

1) First insert the headset with MIC function into the 3.5mm headset interface of the development board.



2) Then go to the directory where the audio test program is located.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo -i
(base) root@orangepiaipro-20t:~# cd /opt/opi_test/audio
(base) root@orangepiaipro-20t:/opt/opi_test/audio# ls
capture.sh play.sh qzgy_48k_16_mono_30s.pcm sample_audio
```

3) Then you can use the following command to record a 5-second audio.

```
(base) root@orangepiaipro-20t:/opt/opi_test/audio# ./sample_audio capture test.pcm
```

4) After the recording is completed, a recording file named **test.pcm** will be generated in the current directory.

```
(base) root@orangepiaipro-20t:/opt/opi_test/audio# ls test.pcm
test.pcm
```

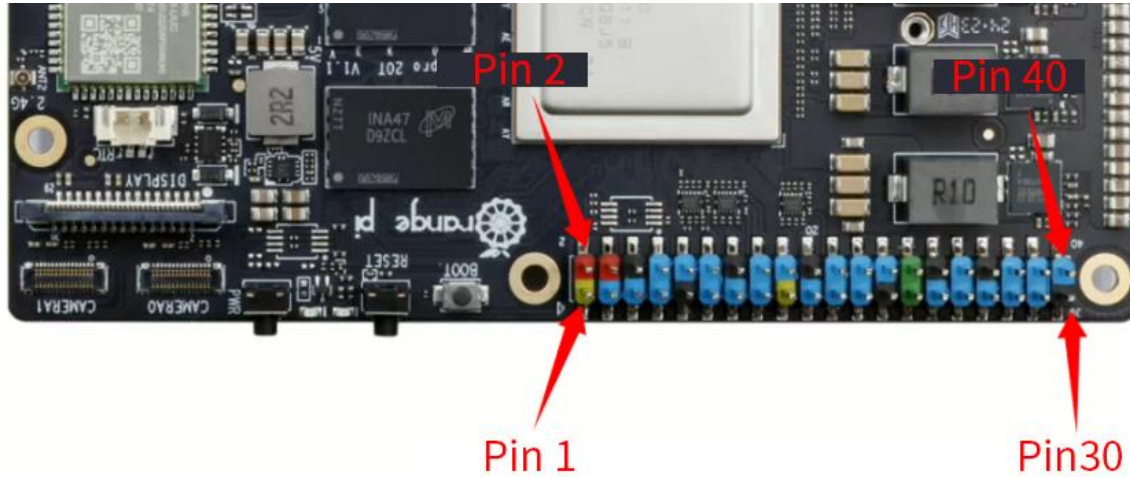
5) Then use the following command to play the recorded audio file to the headphones.

```
(base) root@orangepiaipro-20t:/opt/opi_test/audio# ./sample_audio play 2 test.pcm
```



3. 11. 40 Pin interface pin function description

The order of the 40-pin interface pins of the development board is shown in the figure below:



The functions of the 40 pin interface pins of the development board are shown in the following table:

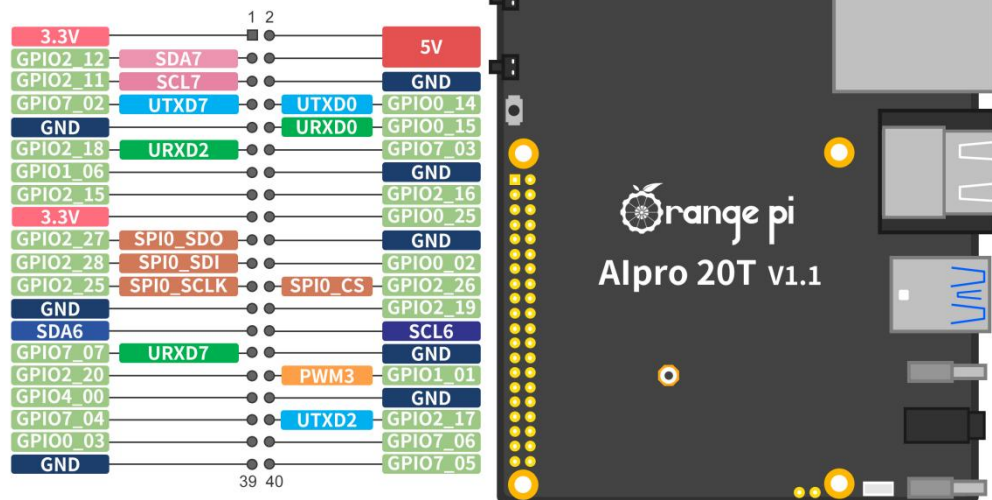
GPIO number	GPIO	Function	Pin
		3.3V	1
76	GPIO2_12	SDA7	3
75	GPIO2_11	SCL7	5
226	GPIO7_02	UTXD7	7
		GND	9
82	GPIO2_18	URXD2	11
38	GPIO1_06		13
79	GPIO2_15		15
		3.3V	17
91	GPIO2_27	SPI0_SDO	19
92	GPIO2_28	SPI0_SDI	21
89	GPIO2_25	SPI0_SCLK	23
		GND	25
		SDA6	27
231	GPIO7_07	URXD7	29

Pin	Function	GPIO	GPIO number
2	5V		
4	5V		
6	GND		
8	UTXD0	GPIO0_14	14
10	URXD0	GPIO0_15	15
12		GPIO7_03	227
14	GND		
16		GPIO2_16	80
18		GPIO0_25	25
20	GND		
22		GPIO0_02	2
24	SPI0_CS	GPIO2_26	90
26		GPIO2_19	83
28	SCL6		
30	GND		



84	GPI02_20		31
128	GPI04_00		33
228	GPI07_04		35
3	GPI00_03		37
		GND	39

32	PWM3	GPI01_01	33
34	GND		
36	UTXD2	GPI02_17	81
38		GPI07_06	230
40		GPI07_05	229



The precautions for using the 40-pin interface are as follows:

- 1) There are **26** GPIO ports in the 40-pin interface, but pins 8 and 10 are used for debug serial port functions by default, and these two pins are connected to the Type-C USB debug serial port, so please do not set these two pins to GPIO functions.
- 2) The voltage of all GPIO ports is **3.3v**.
- 3) In the 40-pin interface, pins 27 and 28 only have I2C functions and do not have other multiplexing functions such as GPIO. In addition, the default voltage of these two pins is **1.8v**.



3. 12. 40 pin interface GPIO, I2C, UART, SPI, PWM and CAN test

3. 12. 1. 40 pin GPIO port test method

The functions of the 40-pin interface pins of the development board are shown in the following table. The pins marked in red are configured as GPIO functions by default and can be used directly. Other pins with GPIO multiplexing functions need to modify the DTS configuration to use the GPIO function normally.

GPIO number	GPIO	Function	Pin	Pin	Function	GPIO	GPIO number
		3.3V	1	2	5V		
76	GPIO2_12	SDA7	3	4	5V		
75	GPIO2_11	SCL7	5	6	GND		
226	GPIO7_02	UTXD7	7	8	UTXD0	GPIO0_14	14
		GND	9	10	URXD0	GPIO0_15	15
82	GPIO2_18	URXD2	11	12		GPIO7_03	227
38	GPIO1_06		13	14	GND		
79	GPIO2_15		15	16		GPIO2_16	80
		3.3V	17	18		GPIO0_25	25
91	GPIO2_27	SPI0_SDO	19	20	GND		
92	GPIO2_28	SPI0_SDI	21	22		GPIO0_02	2
89	GPIO2_25	SPI0_SCLK	23	24	SPI0_CS	GPIO2_26	90
		GND	25	26		GPIO2_19	83
		SDA6	27	28	SCL6		
231	GPIO7_07	URXD7	29	30	GND		
84	GPIO2_20		31	32	PWM3	GPIO1_01	33
128	GPIO4_00		33	34	GND		
228	GPIO7_04		35	36	UTXD2	GPIO2_17	81
3	GPIO0_03		37	38		GPIO7_06	230
		GND	39	40		GPIO7_05	229

The **gpio_operate** tool is pre-installed in the Linux image to set the input and output direction of the GPIO pins. Each GPIO pin can also be independently set to 0 or 1. The detailed usage of the **gpio_operate** tool is as follows:

- 1) The **gpio_operate** tool must be executed using the **root** account.



2) The **gpio_operate -h** command can obtain help information for the **gpio_operate** tool:

```
(base) root@orangepiaipro-20t:~# gpio_operate -h
Usage: gpio_operate <Command|-h> [Options...]
gpio_operate Command:
    -h                : This command's help information.
    set_value         : Set gpio pin value.
    get_value         : Get gpio pin value.
    set_direction     : Set gpio pin direction value.
    get_direction     : Get gpio pin direction value.
```

3) **gpio_operate get_direction gpio_group gpio_pin** is used to query the direction of the GPIO pin.

a. The **gpio_group** and **gpio_pin** parameter descriptions are as follows:

type	describe
<i>gpio_group</i>	GPIO group number, value is [0, 8]
<i>gpio_pin</i>	GPIO pin number, value is [0, 31]

b. For example, the GPIO corresponding to pin 31 in the 40 pins is GPIO2_20, so its GPIO group number is 2, and the GPIO pin number is 20. The command to get its direction is:

```
(base) root@orangepiaipro-20t:~# gpio_operate get_direction 2 20
Get gpio pin direction value succeeded, value is 0.
```

c. Output print information description

Fields	illustrate
direction	GPIO pin direction, value is [0, 1] • 0: Input Direction • 1: Output Direction

4) **gpio_operate set_direction gpio_group gpio_pin direction** is used to set the GPIO pin direction.

a. The **gpio_group**, **gpio_pin** and **direction** parameters are described as follows:



type	describe
<i>gpio_group</i>	GPIO group number, value is [0, 8]
<i>gpio_pin</i>	GPIO pin number, value is [0, 31]
<i>direction</i>	GPIO pin direction, value is [0, 1] • 0: Input Direction • 1: Output Direction

b. For example, if the GPIO corresponding to pin 31 in the 40 pins is GPIO2_20, then its GPIO group number is 2, and the GPIO pin number is 20. The command to set its direction to output is:

```
(base) root@orangepiaipro-20t:~# gpio_operate set_direction 2 20 1
Set gpio pin direction value succeeded.
```

5) The **gpio_operate get_value gpio_group gpio_pin** command is used to query the value of the GPIO pin.

a. *gpio_group* and *gpio_pin* parameter descriptions are as follows:

type	describe
<i>gpio_group</i>	GPIO group number, value is [0, 8]
<i>gpio_pin</i>	GPIO pin number, value is [0, 31]

b. For example, the GPIO corresponding to pin 31 in the 40 pins is GPIO2_20, so its GPIO group number is 2, and the GPIO pin number is 20. The command to query its pin value is as follows:

```
(base) root@orangepiaipro-20t:~# gpio_operate get_value 2 20
Get gpio pin value succeeded, value is 0.    #The value queried here is 0, which is a low level
```

6) The **gpio_operate set_value gpio_group gpio_pin value** command is used to set the GPIO pin value to a high level or a low level. **Note that before setting the pin value, make sure that the direction of the GPIO pin has been set to output.**

a. *gpio_group*, *gpio_pin* and *value* parameters are described as follows:

type	describe
------	----------



type	describe
<i>gpio_group</i>	GPIO group number, value is [0, 8]
<i>gpio_pin</i>	GPIO pin number, value is [0, 31]
<i>value</i>	GPIO pin value, the value is [0, 1] When the GPIO pin direction is input, you are not allowed to set the GPIO pin value.

b. For example, the GPIO corresponding to pin 31 in the 40 pins is GPIO2_20, so its GPIO group number is 2, and the GPIO pin number is 20. The command to set its output to a high level is:

```
(base) root@orangepiaipro-20t:~# gpio_operate set_value 2 20 1
```

3. 12. 2. 40 pin SPI loopback test

The functions of the 40-pin interface pins of the development board are shown in the following table. The pins marked in red have SPI functions, and the Linux system is configured for SPI functions by default and can be used directly.

GPIO number	GPIO	Function	Pin	Pin	Function	GPIO	GPIO number
		3.3V	1	2	5V		
76	GPIO2_12	SDA7	3	4	5V		
75	GPIO2_11	SCL7	5	6	GND		
226	GPIO7_02	UTXD7	7	8	UTXD0	GPIO0_14	14
		GND	9	10	URXD0	GPIO0_15	15
82	GPIO2_18	URXD2	11	12		GPIO7_03	227
38	GPIO1_06		13	14	GND		
79	GPIO2_15		15	16		GPIO2_16	80
		3.3V	17	18		GPIO0_25	25
91	GPIO2_27	SPI0_SDO	19	20	GND		
92	GPIO2_28	SPI0_SDI	21	22		GPIO0_02	2
89	GPIO2_25	SPI0_SCLK	23	24	SPI0_CS	GPIO2_26	90
		GND	25	26		GPIO2_19	83
		SDA6	27	28	SCL6		
231	GPIO7_07	URXD7	29	30	GND		
84	GPIO2_20		31	32	PWM3	GPIO1_01	33



128	GPI04_00		33
228	GPI07_04		35
3	GPI00_03		37
		GND	39

34	GND		
36	UTXD2	GPI02_17	81
38		GPI07_06	230
40		GPI07_05	229

The SPI bus in the 40-pin interface is SPI0. Before testing, please ensure that the **spidev0.0** device node exists in **/dev**.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ ls /dev/spidev0.0
/dev/spidev0.0
```

Then do not short the mosi and miso pins of SPI0. The output result of running **spidev_test** is as shown below. It can be seen that the data of TX and RX are inconsistent.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo spidev_test -v -D /dev/spidev0.0
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D | .....@.....|
RX | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 | .....|
```

Then use Dupont wire to short the mosi (pin 19 in the 40-pin interface) and miso (pin 21 in the 40-pin interface) pins of SPI1 and run **spidev_test** again. The output is as follows. You can see that the sent and received data are the same, indicating that the loopback test is successful.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo spidev_test -v -D /dev/spidev0.0
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D | .....@.....|
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D | .....@.....|
```

3. 12. 3. 40 Pin I2C Test

The functions of the 40-pin interface pins of the development board are shown in the



following table. The pins marked in red have I2C functions, and the Linux system is configured for I2C functions by default and can be used directly.

GPIO number	GPIO	Function	Pin	Pin	Function	GPIO	GPIO number
		3.3V	1	2	5V		
76	GPIO2_12	SDA7	3	4	5V		
75	GPIO2_11	SCL7	5	6	GND		
226	GPIO7_02	UTXD7	7	8	UTXD0	GPIO0_14	14
		GND	9	10	URXD0	GPIO0_15	15
82	GPIO2_18	URXD2	11	12		GPIO7_03	227
38	GPIO1_06		13	14	GND		
79	GPIO2_15		15	16		GPIO2_16	80
		3.3V	17	18		GPIO0_25	25
91	GPIO2_27	SPI0_SDO	19	20	GND		
92	GPIO2_28	SPI0_SDI	21	22		GPIO0_02	2
89	GPIO2_25	SPI0_SCLK	23	24	SPI0_CS	GPIO2_26	90
		GND	25	26		GPIO2_19	83
		SDA6	27	28	SCL6		
231	GPIO7_07	URXD7	29	30	GND		
84	GPIO2_20		31	32	PWM3	GPIO1_01	33
128	GPIO4_00		33	34	GND		
228	GPIO7_04		35	36	UTXD2	GPIO2_17	81
3	GPIO0_03		37	38		GPIO7_06	230
		GND	39	40		GPIO7_05	229

After starting the Linux system, first confirm that the device nodes of i2c6 and i2c7 exist under **/dev**.

```
(base) HwHiAiUser@orangeipiaipro-20t:~$ ls /dev/i2c-6
/dev/i2c-6
(base) HwHiAiUser@orangeipiaipro-20t:~$ ls /dev/i2c-7
/dev/i2c-7
```

Then connect an i2c device to the i2c6 or i2c7 pin of the 40-pin interface.

	i2c6	i2c7
sda pin	Corresponding to pin 27	Corresponding to pin 3 in



	in 40 pins	40 pins
scl pin	Corresponding to pin 28 of 40 pins	Corresponding to pin 5 in 40 pins
3.3v pin	Corresponding to pin 1 in 40 pins	Corresponding to pin 1 in 40 pins
gnd pin	Corresponding to pin 6 in 40 pins	Corresponding to pin 6 in 40 pins

Then use the **i2cdetect** command. If the address of the connected i2c device can be detected, it means that i2c can be used normally.

1) The commands used by i2c6 are as follows:

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo i2cdetect -y -r 6
```

2) The commands used by i2c7 are as follows:

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo i2cdetect -y -r 7
```

Different i2c devices have different addresses. The address 0x38 in the figure below is just an example. Please refer to the actual address.

```
(base) HwHiAiUser@orangepiaipro:~$ sudo i2cdetect -y -r 7
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:    -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:    -- -- -- -- -- -- -- -- -- -- -- -- -- --
20:    -- -- -- -- -- -- -- -- -- -- -- -- -- --
30:    -- -- -- -- -- -- -- 38 -- -- -- -- -- --
40:    -- -- -- -- -- -- -- -- -- -- -- -- -- --
50:    -- -- -- -- -- -- -- -- -- -- -- -- -- --
60:    -- -- -- -- -- -- -- -- -- -- -- -- -- --
70:    -- -- -- -- -- -- -- -- -- -- -- -- -- --
(base) HwHiAiUser@orangepiaipro:~$
```

3. 12. 4. 40 pin UART test

The functions of the 40 pin interface pins of the development board are shown in the following table. The pins marked in red have UART functions, and the Linux system is configured as UART functions by default, so they can be used directly. In addition, please note that UART0 is set to debug serial port function by default, so please do not use it as a normal serial port.

GPIO	GPIO	Function	Pin	Pin	Function	GPIO	GPIO
------	------	----------	-----	-----	----------	------	------



number			
		3.3V	1
76	GPI02_12	SDA7	3
75	GPI02_11	SCL7	5
226	GPI07_02	UTXD7	7
		GND	9
82	GPI02_18	URXD2	11
38	GPI01_06		13
79	GPI02_15		15
		3.3V	17
91	GPI02_27	SPI0_SDO	19
92	GPI02_28	SPI0_SDI	21
89	GPI02_25	SPI0_SCLK	23
		GND	25
		SDA6	27
231	GPI07_07	URXD7	29
84	GPI02_20		31
128	GPI04_00		33
228	GPI07_04		35
3	GPI00_03		37
		GND	39

			number
2	5V		
4	5V		
6	GND		
8	UTXD0	GPI00_14	14
10	URXD0	GPI00_15	15
12		GPI07_03	227
14	GND		
16		GPI02_16	80
18		GPI00_25	25
20	GND		
22		GPI00_02	2
24	SPI0_CS	GPI02_26	90
26		GPI02_19	83
28	SCL6		
30	GND		
32	PWM3	GPI01_01	33
34	GND		
36	UTXD2	GPI02_17	81
38		GPI07_06	230
40		GPI07_05	229

After starting the Linux system, first confirm that the uart device node exists under **/dev**.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ ls /dev/ttyAMA*
/dev/ttyAMA0 /dev/ttyAMA1 /dev/ttyAMA2
```

The corresponding relationship between uart device node and uart is as follows:

uart device node	uart interface
/dev/ttyAMA1	uart2
/dev/ttyAMA2	uart7

Then start testing the UART interface. First, use a Dupont line to short-circuit the RX and TX pins of the UART interface to be tested. The pins in the 40-pin interface corresponding to the RX and TX pins of different UARTs are as follows:



uart interface	rx pin	tx pin
uart2	pin 11 in pin 40	pin 36 in pin 40
uart7	pin 29 in pin 40	pin 7 in pin 40

Then enter the path of the serial port test program.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo -i
(base) root@orangepiaipro-20t:~# cd /opt/opi_test/uart
(base) root@orangepiaipro-20t:/opt/opi_test/uart# ls
serial serial.c
```

The usage of the serial port test program serial is as follows:

```
(base) root@orangepiaipro-20t:/opt/opi_test/uart# ./serial
Usage: ./serial <serialport>
```

You can use the serial test program to test the serial port's self-transmission and self-reception. The serial program will open the corresponding serial port and loop continuously: send a string - **Hello, Serial Port!**, and then print the received string. If the self-transmission and self-reception strings are the same, it means the test is successful.

1) The uart2 test command is as follows:

```
(base) root@orangepiaipro-20t:/opt/opi_test/uart# ./serial /dev/ttyAMA1
W: Hello, Serial Port!
R: Hello, Serial Port!
.....
```

2) The uart7 test command is as follows:

```
(base) root@orangepiaipro-20t:/opt/opi_test/uart# ./serial /dev/ttyAMA2
W: Hello, Serial Port!
R: Hello, Serial Port!
.....
```

3. 12. 5. 40 pin PWM test

The functions of the 40-pin interface pins of the development board are shown in the following table, where the pins marked in red have PWM functions.

GPIO number	GPIO	Function	Pin	Pin	Function	GPIO	GPIO number
		3.3V	1	2	5V		
76	GPIO2_12	SDA7	3	4	5V		



75	GPI02_11	SCL7	5	6	GND		
226	GPI07_02	UTXD7	7	8	UTXD0	GPI00_14	14
		GND	9	10	URXD0	GPI00_15	15
82	GPI02_18	URXD2	11	12		GPI07_03	227
38	GPI01_06		13	14	GND		
79	GPI02_15		15	16		GPI02_16	80
		3.3V	17	18		GPI00_25	25
91	GPI02_27	SPI0_SDO	19	20	GND		
92	GPI02_28	SPI0_SDI	21	22		GPI00_02	2
89	GPI02_25	SPI0_SCLK	23	24	SPI0_CS	GPI02_26	90
		GND	25	26		GPI02_19	83
		SDA6	27	28	SCL6		
231	GPI07_07	URXD7	29	30	GND		
84	GPI02_20		31	32	PWM3	GPI01_01	33
128	GPI04_00		33	34	GND		
228	GPI07_04		35	36	UTXD2	GPI02_17	81
3	GPI00_03		37	38		GPI07_06	230
		GND	39	40		GPI07_05	229

Currently, the only way to test the PWM3 pin outputting a waveform is by operating the register. The test steps are as follows:

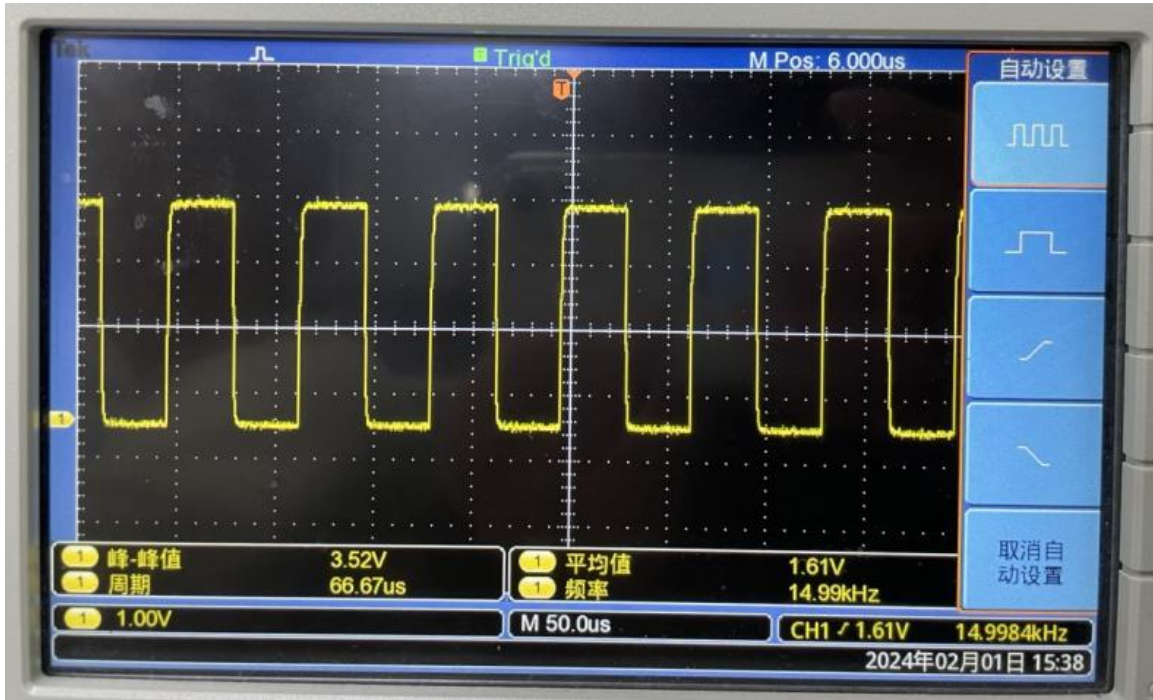
1) First enter the path of the pwm test code.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo -i
[sudo] password for HwHiAiUser:
(base) root@orangepiaipro-20t:~# cd /opt/opi_test/pwm
(base) root@orangepiaipro-20t:/opt/opi_test/pwm# ls
test.sh
```

2) Then run the test.sh script to output a square wave with a 50% duty cycle.

```
(base) root@orangepiaipro-20t:/opt/opi_test/pwm# ./test.sh
```

3) Then use an oscilloscope to measure the 32nd pin in the 40 pin to view the PWM output waveform, as shown below:



3. 12. 6. Test method for 40 pin CAN

3. 12. 6. 1. Method of opening CAN

1) The functions of the 40 pin interface pins on the development board are shown in the table below, with the pins highlighted in red having CAN function.

GPIO serial number	GPIO	Function	Pin	Pin	Function	GPIO	GPIO serial number
		3.3V	1	2	5V		
76	GPIO2_12	SDA7	3	4	5V		
75	GPIO2_11	SCL7	5	6	GND		
226	GPIO7_02	UTXD7	7	8	UTXD0	GPIO0_14	14
		GND	9	10	URXD0	GPIO0_15	15
82	GPIO2_18	CAN_RX3	11	12		GPIO7_03	227
38	GPIO1_06		13	14	GND		
79	GPIO2_15		15	16		GPIO2_16	80
		3.3V	17	18		GPIO0_25	25
91	GPIO2_27	SPI0_SDO	19	20	GND		



92	GPI02_28	SPI0_SDI	21	22		GPI00_02	2
89	GPI02_25	SPI0_SCLK	23	24	SPI0_CS	GPI02_26	90
		GND	25	26		GPI02_19	83
		SDA6	27	28	SCL6		
231	GPI07_07	URXD7	29	30	GND		
84	GPI02_20		31	32	PWM3	GPI01_01	33
128	GPI04_00		33	34	GND		
228	GPI07_04		35	36	CAN_TX3	GPI02_17	81
3	GPI00_03		37	38		GPI07_06	230
		GND	39	40		GPI07_05	229

2) The pins corresponding to CAN3 are:

	CAN3
RX Pin	Corresponding to pin 11 of 40 pin
TX Pin	Corresponding to pin 36 of 40 pin

3) The 11th and 36th pins of the 40pin in the Linux system are set to the UART2 function by default in the device tree. We need to update the dt.img of the Linux system to enable the CAN3 function (**UART2 cannot be used after opening CAN3, choose either of these two functions**). The specific steps are as follows:

- Firstly, download and open the dt.img file with CAN3 functionality from the **official tools** on the development board.



CAN3总线测试相关的文件

- Then upload the dt.img file to the Linux system of the development board. For the method of uploading files to the development board Linux system, please refer to the instructions in the section on **uploading files to the development board Linux system**.
- Then use the following command to update the dt.img file.

- TF card startup:

```
sudo dd if=dt.img of=/dev/mmcblk1 count=4096 seek=114688 bs=512
sudo dd if=dt.img of=/dev/mmcblk1 count=4096 seek=376832 bs=512
```

- NVMe SSD startup:



```
sudo dd if=dt.img of=/dev/nvme0n1 count=4096 seek=114688 bs=512
sudo dd if=dt.img of=/dev/nvme0n1 count=4096 seek=376832 bs=512
```

c) eMMC startup:

```
sudo dd if=dt.img of=/dev/mmcblk0 count=4096 seek=114688 bs=512
sudo dd if=dt.img of=/dev/mmcblk0 count=4096 seek=376832 bs=512
```

d. Then restart the Linux system.

4) After restarting the Linux system and using the **sudo ifconfig -a** command, if you can see the device nodes of CAN3, it indicates that CAN3 has been opened correctly.

```
(base) HwHiAiUser@orangepiaipro:~$ sudo ifconfig -a
can3: flags=128<NOARP> mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

3. 12. 6. 2. Use CANalyst-II analyzer to test CAN bus for sending and receiving messages

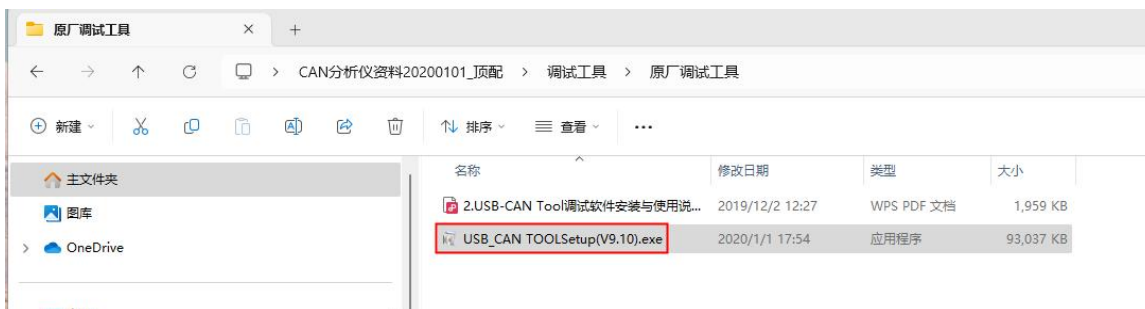
1) The CANalyst-II analyzer used for testing is shown in the following figure:



2) CANalyst-II analyzer data download link.

<https://www.zhcxgd.com/3.html>

3) Firstly, you need to install the USL_CAN ToolSetup software.



4) The shortcut for installing USB-CAN ToolSetup is:



5) Additionally, it is necessary to install the USB driver.



6) The USB interface of the CANalyst-II analyzer needs to be connected to the USB interface of the computer.



7) To test the CAN function, it is necessary to prepare a CAN transceiver as shown in the figure below. The main function of the CAN transceiver is to convert the TTL signal of the CAN controller into a differential signal of the CAN bus.

- The 3.3V pin of the CAN transceiver needs to be connected to the 3.3V pin in the 40 pin of the development board.
- The GND pin of the CAN transceiver needs to be connected to the GND pin in



- pin 40 of the development board.
- The CAN TX pin of the CAN transceiver needs to be connected to the TX pin of the CAN bus in pin 40 of the development board.
 - The CAN RX pin of the CAN transceiver needs to be connected to the RX pin of the CAN bus in pin 40 of the development board.
 - The CANH pin of the CAN transceiver needs to be connected to the H interface of the analyzer.
 - The CANL pin of the CAN transceiver needs to be connected to the L interface of the analyzer.



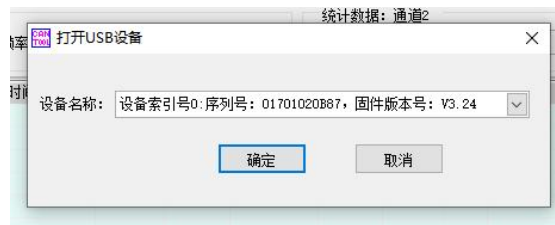
8) Then you can open the USB-CAN software.



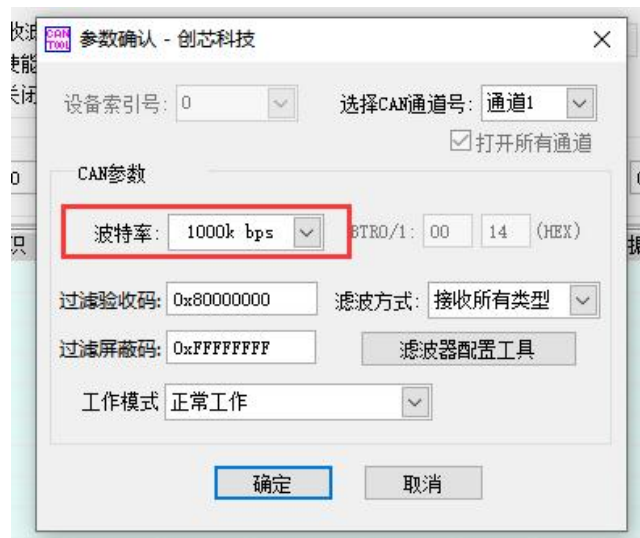
9) Then click to start the device.



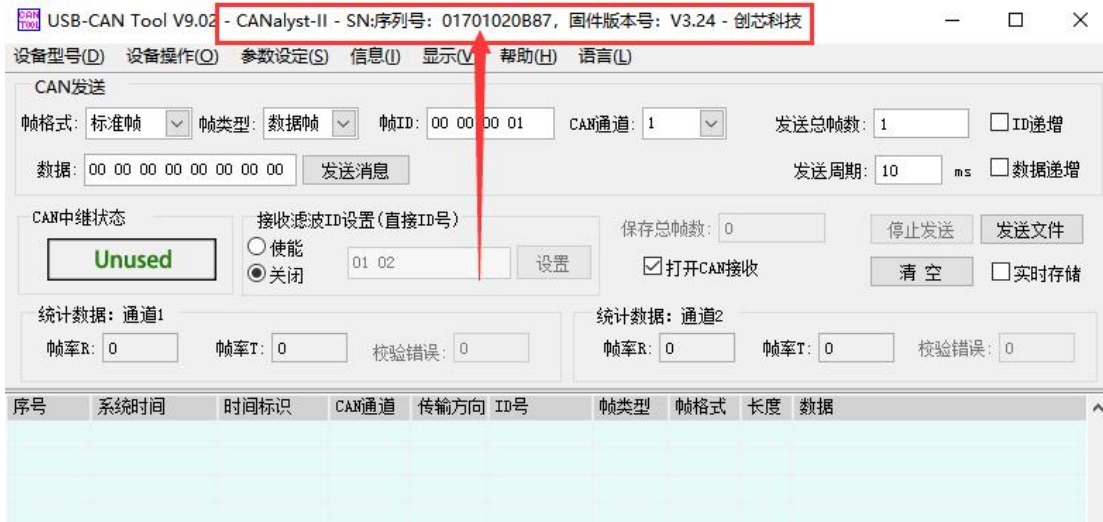
10) Then click OK.



11) Set the baud rate to 1000k bps.



12) After successfully opening, the USB-CAN software will display serial number and other information.



13) Then install the can utils package.

```
(base) HwHiAiUser@orangepi:~$ sudo apt-get install -y can-utils
```

If the package manager cannot be used to install can utils in a Linux system, then can only be installed through source code. The source code link for can utils is as follows:

<https://github.com/linux-can/can-utils>

After downloading the source code, use the make&&make install command to compile and install can utils.

14) Development board receives CAN message testing.

- a. Firstly, set the baud rate of the CAN bus to **1000kbps** in the Linux system of the development board.

```
(base) HwHiAiUser@orangepi:~$ sudo ip link set can3 down
```

```
(base) HwHiAiUser@orangepi:~$ sudo ip link set can3 type can bitrate 1000000
```

```
(base) HwHiAiUser@orangepi:~$ sudo ip link set can3 up
```

- b. Then run the **candump can3** command to prepare to receive messages

```
(base) HwHiAiUser@orangepi:~$ sudo candump can3
```

- c. Then send a message to the development board in the USB-CAN software.



- d. If the development board can receive messages sent by the analyzer, it indicates that the CAN bus can be used normally.

```
(base) HwHiAiUser@orangePi:~$ sudo candump can3
can3 001 [8] 01 02 03 04 05 06 07 08
```

15) Development board sends CAN message test.

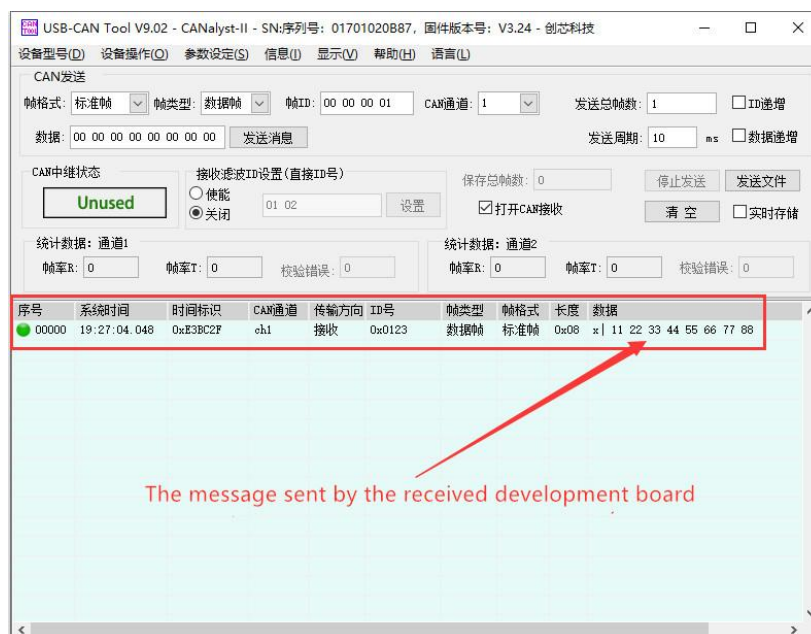
- a. Firstly, set the baud rate of CAN to **1000kbps** in the Linux system.

```
(base) HwHiAiUser@orangePi:~$ sudo ip link set can3 down
(base) HwHiAiUser@orangePi:~$ sudo ip link set can3 type can bitrate 1000000
(base) HwHiAiUser@orangePi:~$ sudo ip link set can3 up
```

- b. Execute the **cansend** command in the development board and send a message.

```
(base) HwHiAiUser@orangePi:~$ sudo cansend can3 123#1122334455667788
```

- c. If the USB-CAN software can receive messages from the development board, it indicates successful communication.





3. 13. Installation and usage methods of wiringOP

3. 13. 1. Method for installing wiringOP

1) Before installing wiringOP, please ensure that the `/etc/orangepi-release` configuration file exists in the Linux system, which contains the following content:
BOARD=orangepiaipro.

```
(base) HwHiAiUser@orangepi:~$ cat /etc/orangepi-release  
BOARD=orangepiaipro
```

2) If the `/etc/orangepi-release` configuration file is not available in Linux, you can create one using the following command.

```
(base) HwHiAiUser@orangepi:~$ echo "BOARD=orangepiaipro" | sudo tee /etc/orangepi-release
```

3) Download the code for wiringOP.

```
(base) HwHiAiUser@orangepi:~$ sudo apt-get update  
(base) HwHiAiUser@orangepi:~$ sudo apt-get install -y git  
(base) HwHiAiUser@orangepi:~$ git clone https://github.com/orangepi-xunlong/wiringOP.git -b next
```

Note that the source code needs to download the code for the wiringOP next branch. Please do not miss the - b next parameter.

4) Then compile and install wiringOP.

```
(base) HwHiAiUser@orangepi:~$ sudo apt-get install -y gcc make build-essential  
(base) HwHiAiUser@orangepi:~$ cd wiringOP  
(base) HwHiAiUser@orangepi:~/wiringOP$ sudo ./build clean  
(base) HwHiAiUser@orangepi:~/wiringOP$ sudo ./build
```

5) The output of the test gpio readall command is as follows:



```
(base) HwHiAiUser@orangepi:~$ gpio readall
```

GPIO	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	GPIO
		3.3V			1	2		5V		
76	0	SDA7	OFF	0	3	4		5V		
75	1	SCL7	OFF	0	5	6		GND		
226	2	GPIO7_02	OFF	0	7	8	0	UTXD0	3	14
		GND			9	10	0	URXD0	4	15
82	5	GPIO2_18	OFF	0	11	12	0	GPIO7_03	6	227
38	7	GPIO1_06	IN	1	13	14		GND		
79	8	GPIO2_15	IN	1	15	16	1	GPIO2_16	9	80
		3.3V			17	18	1	GPIO0_25	10	25
91	11	SPI0_SD0	OFF	0	19	20		GND		
92	12	SPI0_SDI	OFF	0	21	22	1	GPIO0_02	13	2
89	14	SPI0_CLK	OFF	0	23	24	0	SPI0_CS	15	90
		GND			25	26	1	GPIO2_19	16	83
		SDA6			27	28		SCL6		
231	17	URXD7	OFF	0	29	30		GND		
84	18	GPIO2_20	IN	1	31	32	1	PWM3	19	33
128	20	GPIO4_00	IN	1	33	34		GND		
228	21	GPIO7_04	OFF	0	35	36	0	GPIO2_17	22	81
3	23	GPIO0_03	IN	1	37	38	1	GPIO7_06	24	230
		GND			39	40	0	GPIO7_05	25	229
GPIO	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	GPIO
					AI	PRO				

3. 13. 2. Method of using wiringOP to control 40pin GPIO

1) Taking pin 7- corresponding to GPIO7:02- corresponding to wPi number 2- as an example, demonstrate how to set the direction and high and low levels of the GPIO port.

```
(base) HwHiAiUser@orangePi:~/wiringOP$ gpio readall
```

GPIO	wPi	Name	Mode	V	AI	PRO	V	Mode	Name	wPi	GPIO
		3.3V			1	2			5V		
76	0	SDA7	OFF	0	3	4			5V		
75	1	SCL7	OFF	0	5	6			GND		
226	2	GPIO7_02	OFF	0	7	8	0	OFF	UTXD0	3	14
		GND			9	10	0	OFF	URXD0	4	15
82	5	GPIO2_18	OFF	0	11	12	0	OFF	GPIO7_03	6	227
38	7	GPIO1_06	IN	1	13	14			GND		
79	8	GPIO2_15	IN	1	15	16	1	IN	GPIO2_16	9	80

2) Firstly, set the GPIO port to output mode, where the third parameter requires the serial number of the wPi corresponding to the input pin.

```
(base) HwHiAiUser@orangePi:~$ gpio mode 2 out
```

3) Then use the following command to view the current mode of GPIO, and you can see that the current mode is output -- **OUT**. The second parameter in the command needs to



input the serial number of the wPi corresponding to the pin.

```
(base) HwHiAiUser@orangePi:~$ gpio qmode 2  
OUT
```

4) Then you can start setting the pin output high and low levels. For example, the following command can be used to set the GPIO port output low level. After setting, a multimeter can be used to measure the voltage value of the pins. If it is 0v, it indicates that the low level setting is successful.

```
(base) HwHiAiUser@orangePi:~$ gpio write 2 0
```

5) In addition to using a multimeter to measure the voltage of the pins, you can also use the **gpio read** command to check the high and low level status of the pins. The current command output is 0, indicating that the pin is currently at a low level.

```
(base) HwHiAiUser@orangePi:~$ gpio read 2  
0
```

6) 然 Then, the GPIO port can be set to output a high level. After setting, a multimeter can be used to measure the voltage value of the pins. If it is 3.3v, it indicates that the high level has been successfully set.

```
(base) HwHiAiUser@orangePi:~$ gpio write 2 1
```

7) In addition to using a multimeter to measure the voltage of the pins, you can also use the **gpio read** command to check the high and low level status of the pins. The current command output is 1, indicating that the pin is currently at a high level.

```
(base) HwHiAiUser@orangePi:~$ gpio read 2  
1
```

8) Additionally, the **gpio readall** command can also be used to view the current settings of GPIO.



```
(base) HwHiAiUser@orangePi:~$ gpio readall
```

GPIO	wPi	Name	Mode	V	AI	PRO	V	Mode	Name	wPi	GPIO
		3.3V			1	2			5V		
76	0	SDA7	OFF	0	3	4			5V		
75	1	SCL7	OFF	0	5	6			GND		
226	2	GPIO7_02	OUT	1	7	8	0	OFF	UTXD0	3	14
		GND			9	10	0	OFF	URXD0	4	15
82	5	GPIO2_18	OFF	0	11	12	0	OFF	GPIO7_03	6	227
38	7	GPIO1_06	IN	1	13	14			GND		
79	8	GPIO2_15	IN	1	15	16	1	IN	GPIO2_16	9	80

9) The following command can be used to set the GPIO port to input mode, where the third parameter requires the serial number of the wPi corresponding to the input pin.

```
(base) HwHiAiUser@orangePi:~$ gpio mode 2 in
```

10) After setting the GPIO port to input mode, when the GPIO port is connected to the GND pin with a DuPont cable, using the **gpio read** command can see that the input value of the GPIO port will be 0.

```
(base) HwHiAiUser@orangePi:~$ gpio read 2
0
```

11) After setting the GPIO port to input mode, when connecting the GPIO port to the 3.3V pin with a DuPont cable, using the **gpio read** command can see that the input value of the GPIO port will be 1.

```
(base) HwHiAiUser@orangePi:~$ gpio read 2
1
```

12) The setting method for other pins is similar, just modify the serial number of wPi to the corresponding serial number of the pin.

3. 14. WiringOP hardware PWM usage method

Before using wiringOP to operate PWM, please ensure that wiringOP is installed on the Linux system. If the **gpio readall** command can be used normally, it means wiringOP has been installed. If prompted that the command cannot be found, please refer to the [instructions in the installation and usage section of wiringOP](#) to install wiringOP first.



The PWM3 PWM can be used in the 40 pin interface of the development board, and the pin positions are shown in the following figure:



3. 14. 1. Method of setting PWM using the gpio command of WiringOP

3. 14. 1. 1. Method for setting the corresponding pin to PWM mode

1) The functions of the 40 pin interface pins on the development board are shown in the table below, with the pins highlighted in red having pwm functionality

GPIO Serial number	GPIO	function	pin	pin	function	GPIO	GPIO Serial number
		3.3V	1	2	5V		
76	GPIO2_12	SDA7	3	4	5V		
75	GPIO2_11	SCL7	5	6	GND		
226	GPIO7_02	UTXD7	7	8	UTXD0	GPIO0_14	14
		GND	9	10	URXD0	GPIO0_15	15
82	GPIO2_18	URXD2	11	12		GPIO7_03	227
38	GPIO1_06		13	14	GND		
79	GPIO2_15		15	16		GPIO2_16	80
		3.3V	17	18		GPIO0_25	25
91	GPIO2_27	SPI0_SDO	19	20	GND		
92	GPIO2_28	SPI0_SDI	21	22		GPIO0_02	2
89	GPIO2_25	SPI0_SCLK	23	24	SPI0_CS	GPIO2_26	90
		GND	25	26		GPIO2_19	83
		SDA6	27	28	SCL6		
231	GPIO7_07	URXD7	29	30	GND		



84	GPI02_20		31
128	GPI04_00		33
228	GPI07_04		35
3	GPI00_03		37
		GND	39

32	PWM3	GPI01_01	33
34	GND		
36	UTXD2	GPI02_17	81
38		GPI07_06	230
40		GPI07_05	229

2) The corresponding relationship between PWM pins and wPi serial numbers in the 40Pin development board is shown in the following table:

PWM Pin	wPi Serial number
PWM3	19

3) The command to set the pin to PWM mode is as follows, and the third parameter requires inputting the serial number of wPi corresponding to the PWM pin.

```
(base) HwHiAiUser@orangePi:~$ gpio mode 19 pwm
```

4) After setting the pin to PWM mode, a square wave with a frequency of 1kHz, a period of 1ms, and a duty cycle of 50% will be output by default. At this time, we can use an oscilloscope to measure the PWM3 pin and see the waveform below.



5) Then, using the `gpio qmode 19` command, you can see that the mode of the PWM3 pin is set to PWM.

```
(base) HwHiAiUser@orangePi:~$ gpio qmode 19
```



PWM

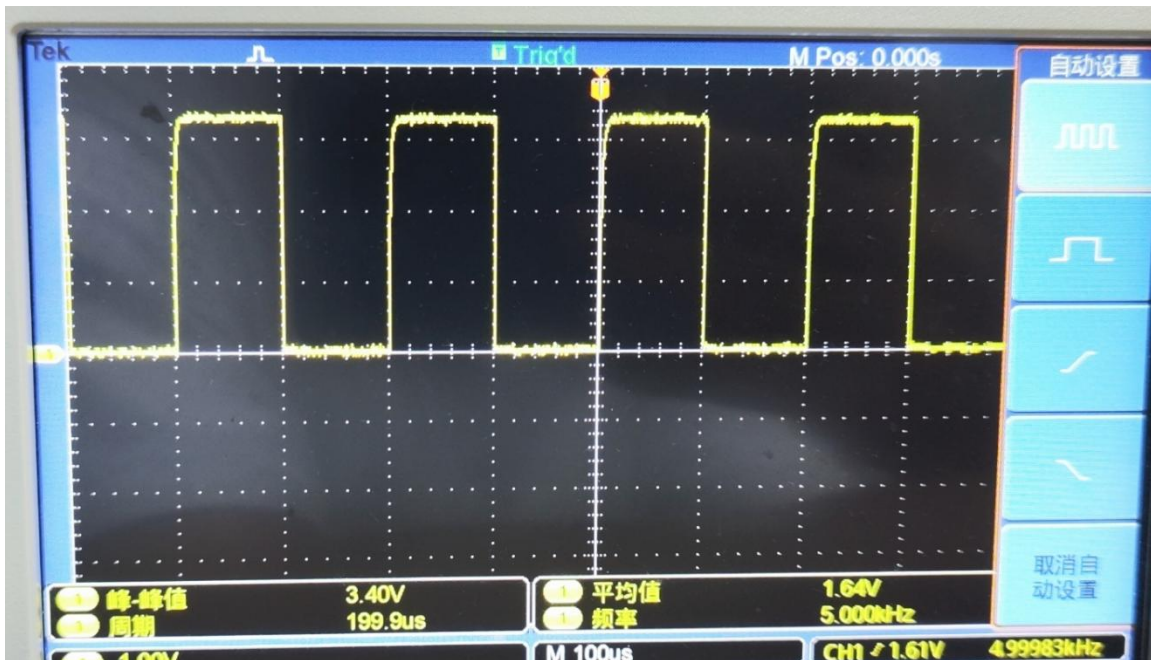
3. 14. 1. 2. Method of directly setting PWM frequency

Before setting the frequency of PWM, please ensure that the corresponding pin has been set to PWM mode.

1) We can use the `gpio pwmTone` command to set the frequency of the PWM pin, for example, the following command can be used to set the PWM frequency to 5000Hz. At the same time as setting the frequency, the `gpio pwmTone` command also sets the duty cycle to 50%. Additionally, it should be noted that the PWM frequency can be set within the range of 1 ~ 32000hz. Frequency values outside of this range will result in an error.。

```
(base) HwHiAiUser@orangePi:~$ gpio pwmTone 19 5000
```

2) Then, through an oscilloscope, it can be observed that the PWM frequency has changed to 5000Hz and the duty cycle is 50%.





3. 14. 1. 3. Method of setting PWM period through pulse period register

1) The description of the PWM3 pulse period register - PWM_PRD3 is as follows:

PWM_PRD3	3通道脉冲周期寄存器	0x2C	32	0x0FFFFFFF	reserved	31:28	-	0x0	保留
					pwm_prd3	27:0	RW	0xFFFFFFFF	通道3输出PWM信号的周期

2) The APB bus clock used by PWM is 150MHz, and this clock cannot be divided (**so it is not supported to modify the clock frequency through the pwmc command**). The calculation formula for the PWM output waveform period is as follows:

PWM cycle = value of PWM_PRD3 register / 150 (PWM cycle unit is us)

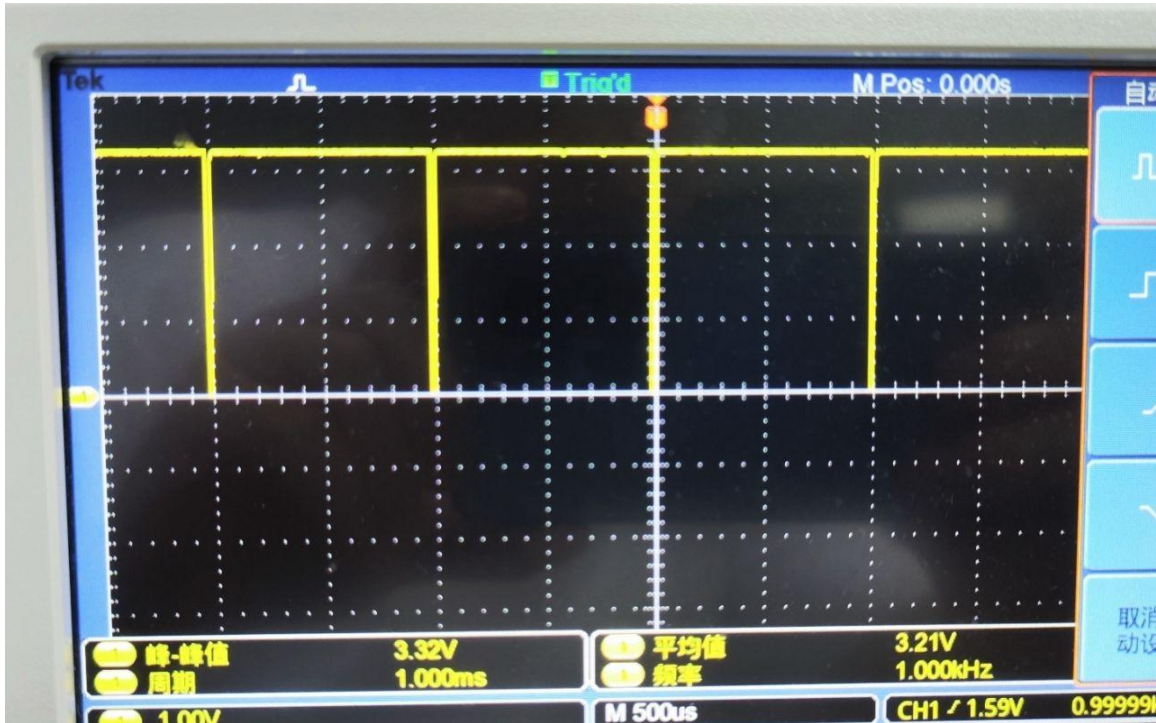
value of PWM_PRD3 register = PWM cycle x 150 (PWM cycle unit is us)

3) The **gpio pwmr** command can be used to set the value of the pulse period register. For example, to set the period of the PWM output waveform to 1000us, the formula above shows that the value of the PWM_PRD3 register needs to be set to 150000.

```
(base) HwHiAiUser@orangePi:~$ gpio pwmr 19 150000
```

Due to the recommended PWM frequency value between 1 ~ 32000hz, the range of values for the PWM_PRD3 register is 4688 ~ 150000000.

4) After setting up, the waveform of PWM can be seen through the oscilloscope as shown below, with a display period of 1ms, which is 1000us.



5) Given the period, the frequency can be calculated, so this command can also be used to set the frequency of the PWM output waveform. But this command only modifies the pulse period register and does not set the duty cycle to 50%.

3. 14. 1. 4. Methods for Adjusting PWM Duty Cycle

1) **gpio pwm** 命 The **gpio pwm** command can set the duty cycle of PWM. The usage of the **gpio pwm** command is as follows:

- a. The **pin** needs to be filled with the wPi serial number corresponding to the pwm pin.
- b. **value** 需 The value needs to be filled in with the period **value** occupied by the high level. **Duty cycle=value/value of pulse period register**. For example, if the pulse period register is set to 150000 and the duty cycle needs to be set to 50%, the value needs to be set to 75000. Also, please note that the maximum value is: **the value of the pulse period register -1**.

Usage: **gpio pwm <pin> <value>**

2) Example of setting PWM duty cycle.



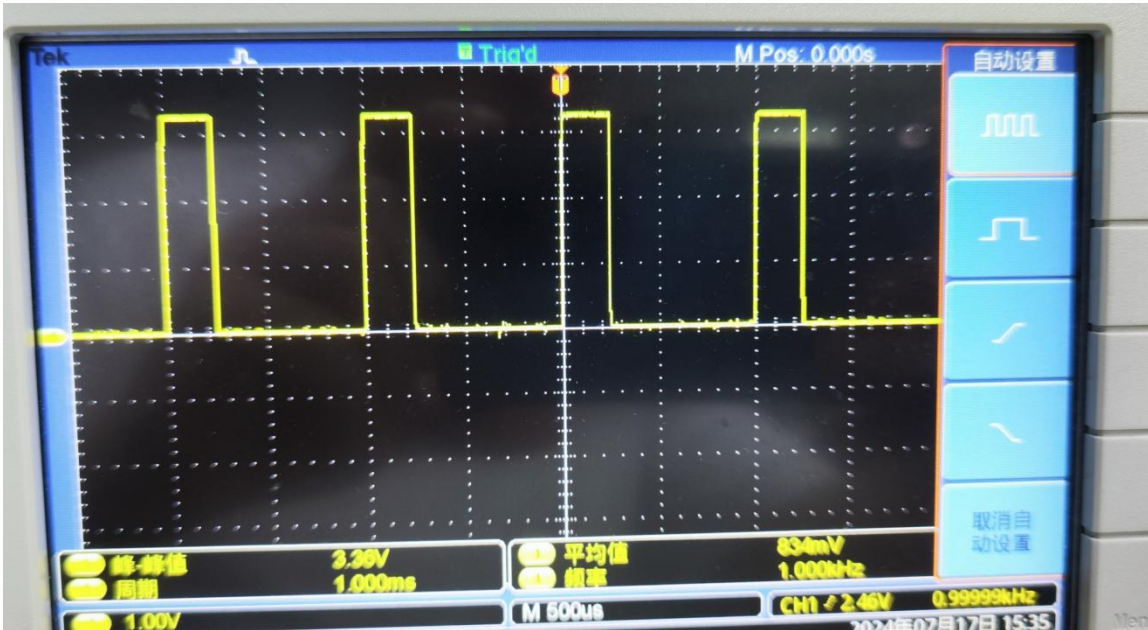
- a. First, set the pulse period register, for example, use the following command to set it to 150000.

```
(base) HwHiAiUser@orangepi:~$ gpio pwmr 19 150000
```

- b. Then use the following command to set the duty cycle to 25%.

```
(base) HwHiAiUser@orangepi:~$ gpio pwm 19 37500
```

3) After running the above command, it can be observed through the oscilloscope that the PWM duty cycle will change to 25%.



3. 14. 2. Usage of PWM Test Program

1) In addition to using gpio command to control PWM, you can also call PWM related API in C language program to control PWM waveform. In the example directory of wiringOP, there is a program called pwm.c that demonstrates the use of PWM related API in wiringOP to operate PWM.

```
(base) HwHiAiUser@orangepi:~$ cd wiringOP
(base) HwHiAiUser@orangepi:~/wiringOP$ cd examples
(base) HwHiAiUser@orangepi:~/wiringOP/examples$ ls pwm.c
pwm.c
```

2) The command to compile **pwm.c** into an executable program is as follows:

```
(base) HwHiAiUser@orangepi:~/wiringOP/examples$ gcc -o pwm pwm.c -lwiringPi
```



3) Then you can execute the PWM test program. When executing the PWM test program, you need to specify the wPi number corresponding to the PWM pin. For example, you can use the following command to test the PWM3 pin:

```
sorangepi@orangepi:/usr/src/wiringOP/examples$ sudo ./pwm 19
```

4) After the pwm program is executed, the following contents will be tested sequentially:

- a. By setting ARR, which is the pulse period register, the PWM duty cycle can be adjusted.
- b. By setting the CCR, which is the number of cycles occupied by high levels, the PWM duty cycle can be adjusted.
- c. Directly set the PWM frequency.

5) After completing each test, the pwm test program will maintain the last pwm waveform for 5 seconds. After completing all test contents, a new round of testing will be restarted.

6) The detailed execution process of the PWM test program is as follows:

- a. By setting ARR to adjust PWM duty cycle: Through the oscilloscope, it can be observed that the PWM waveform changes every 0.5 seconds. After 8 changes, the PWM duty cycle changes from 50% to 75% and remains for 5 seconds. Then, the PWM waveform changes every 0.5 seconds. After 8 changes, the PWM duty cycle changes from 75% to 50% and remains for 5 seconds.
- b. By setting the CCR to adjust the PWM duty cycle: The oscilloscope can observe that the PWM waveform changes every 0.5 seconds. After 8 changes, the PWM duty cycle changes from 50% to 100% and remains for 5 seconds. Then, the PWM waveform changes every 0.5 seconds. After 8 changes, the PWM duty cycle changes from 100% to 50% and remains for 5 seconds.
- c. Directly setting the PWM frequency: Through the oscilloscope, it can be observed that the PWM frequency first changes to 2000Hz, and then increases by 2000Hz every two seconds. After 9 changes, the PWM frequency changes to 20000Hz and remains for 5 seconds.

3. 15. wiringOP-Python installation and usage instructions

wiringOP-Python is a library for the Python language version of wiringOP, used



to operate hardware resources such as GPIO, I2C, SPI, and UART on development boards in Python programs.

Also, please note that the following commands are operated under the **root** user.

Before installing wiringOP Python, please ensure that the Linux system has installed wiringOP. If the gpio readall command can be used normally, it means wiringOP has been installed. If prompted that the command cannot be found, please refer to the instructions [in the installation and usage section of wiringOP to install wiringOP first](#).

3. 15. 1. wiringOP-Python installation method

1) First, install the dependency package.

```
(base) root@orangepi:~# apt-get update
```

```
(base) root@orangepi:~# apt-get -y install git swig python3-dev python3-setuptools
```

Note that if using the openEuler system, please use the following command to install the dependency package.

```
-sh-5.1# dnf update
```

```
-sh-5.1# dnf install -y git swig python3-devel python3-setuptools
```

2) Then use the following command to download the source code of wiringOP-Python.

Note that the `git clone--recursive` command below will automatically download the source code of wiringOP, as wiringOP-Python relies on wiringOP. Please ensure that there are no errors during the download process due to network issues.

```
(base) root@orangepi:~# git clone --recursive https://github.com/orangepi-xunlong/wiringOP-Python -b next
```

```
(base) root@orangepi:~# cd wiringOP-Python
```

```
(base) root@orangepi:~/wiringOP-Python# git submodule update --init --remote
```

3) Then use the following command to compile wiringOP-Python and install it into the Linux system of the development board.

```
(base) root@orangepi:~/wiringOP-Python# python3 generate-bindings.py > bindings.i
```

```
(base) root@orangepi:~/wiringOP-Python# python3 setup.py install
```

4) Then enter the following command. If there is a help message output, it indicates that wiringOP-Python installation is successful. Press the **q** key to exit the help message



interface.

```
(base) root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; help(wiringpi)"
```

Help on module wiringpi:

NAME

wiringpi

DESCRIPTION

This file was automatically generated by SWIG (<http://www.swig.org>).

Version 4.0.2

#

Do not make changes to this file unless you know what you are doing--modify

the SWIG interface file instead.

5) The steps to test whether wiringOP-Python is successfully installed under the python command line are as follows:

- a. First, use the python3 command to enter the command-line mode of python3.

```
(base) root@orangepi:~# python3
```

- b. Then import the python module of wiringpi.

```
>>> import wiringpi;
```

- c. Finally, enter the following command to view the help information for wiringOP-Python. Press the **q** key to exit the help information interface.

```
>>> help(wiringpi)
```

Help on module wiringpi:

NAME

wiringpi

DESCRIPTION

This file was automatically generated by SWIG (<http://www.swig.org>).

Version 4.0.2

#

Do not make changes to this file unless you know what you are doing--modify

the SWIG interface file instead.



CLASSES

builtins.object

GPIO

I2C

Serial

nes

class GPIO(builtins.object)

| GPIO(pinmode=0)

|

>>>

3. 15. 2. 40pin GPIO Port Test

wiringOP-Python, like wiringOP, can also determine which GPIO pin to operate on by specifying the wPi number. Since there is no command to view the wPi number in wiringOP Python, the correspondence between the wPi number of the board and the physical pin can only be viewed through the gpio command in wiringOP.

```
(base) HwHiAiUser@orangePi:~$ gpio readall
```

GPIO	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	GPIO
		3.3V			1	2		5V		
76	0	SDA7	OFF	0	3	4		5V		
75	1	SCL7	OFF	0	5	6		GND		
226	2	GPIO7_02	OFF	0	7	8	0	UTXD0	3	14
		GND			9	10	0	URXD0	4	15
82	5	GPIO2_18	OFF	0	11	12	0	GPIO7_03	6	227
38	7	GPIO1_06	IN	1	13	14		GND		
79	8	GPIO2_15	IN	1	15	16	1	GPIO2_16	9	80
		3.3V			17	18	1	GPIO0_25	10	25
91	11	SPI0_SD0	OFF	0	19	20		GND		
92	12	SPI0_SDI	OFF	0	21	22	1	GPIO0_02	13	2
89	14	SPI0_CLK	OFF	0	23	24	0	SPI0_CS	15	90
		GND			25	26	1	GPIO2_19	16	83
		SDA6			27	28		SCL6		
231	17	URXD7	OFF	0	29	30		GND		
84	18	GPIO2_20	IN	1	31	32	1	PWM3	19	33
128	20	GPIO4_00	IN	1	33	34		GND		
228	21	GPIO7_04	OFF	0	35	36	0	GPIO2_17	22	81
3	23	GPIO0_03	IN	1	37	38	1	GPIO7_06	24	230
		GND			39	40	0	GPIO7_05	25	229
GPIO	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	GPIO
					AI PRO					



1) Taking pin 7- corresponding to GPIO as GPO7_02- corresponding to wPi serial number 2- as an example to demonstrate how to set the high and low levels of GPIO port.

```
(base) HwHiAiUser@orangePi:~/wiringOP$ gpio readall
```

AI PRO											
GPIO	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	GPIO	
		3.3V			1	2		5V			
76	0	SDA7	OFF	0	3	4		5V			
75	1	SCL7	OFF	0	5	6		GND			
226	2	GPIO7_02	OFF	0	7	8	0	UTXD0	3	14	
		GND			9	10	0	URXD0	4	15	
82	5	GPIO2_18	OFF	0	11	12	0	GPIO7_03	6	227	
38	7	GPIO1_06	IN	1	13	14		GND			
79	8	GPIO2_15	IN	1	15	16	1	GPIO2_16	9	80	

2) The steps for directly testing with commands are as follows:

- Firstly, set the GPIO port to output mode, where the first parameter of the **pinMode** function is the serial number of the wPi corresponding to the pin, and the second parameter is the GPIO mode.

```
(base) root@orangePi:~/wiringOP-Python# python3 -c "import wiringpi; \
from wiringpi import GPIO; wiringpi.wiringPiSetup() ; \
wiringpi.pinMode(2, GPIO.OUTPUT) ; "
```

- Then set the GPIO port to output a low level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 0v, it indicates that the low level has been successfully set.

```
(base) root@orangePi:~/wiringOP-Python# python3 -c "import wiringpi; \
from wiringpi import GPIO; wiringpi.wiringPiSetup() ; \
wiringpi.digitalWrite(2, GPIO.LOW)"
```

- Then set the GPIO port to output a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 3.3v, it indicates that the high level has been successfully set.

```
(base) root@orangePi:~/wiringOP-Python# python3 -c "import wiringpi; \
from wiringpi import GPIO; wiringpi.wiringPiSetup() ; \
wiringpi.digitalWrite(2, GPIO.HIGH)"
```

3) The steps for testing on the command line in python3 are as follows:

- First, use the python3 command to enter the command-line mode of python3.

```
(base) root@orangePi:~# python3
```



b. Then import the python module of wiringpi.

```
>>> import wiringpi
>>> from wiringpi import GPIO
```

c. Then set the GPIO port to output mode, where the first parameter of the **pinMode** function is the serial number of the wPi corresponding to the pin, and the second parameter is the GPIO mode.

```
>>> wiringpi.wiringPiSetup()
0
>>> wiringpi.pinMode(2, GPIO.OUTPUT)
```

d. Then set the GPIO port to output a low level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 0v, it indicates that the low level has been successfully set.

```
>>> wiringpi.digitalWrite(2, GPIO.LOW)
```

e. Then set the GPIO port to output a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 3.3v, it indicates that the high level has been successfully set.

```
>>> wiringpi.digitalWrite(2, GPIO.HIGH)
```

4) The method of setting GPIO high and low levels in python code using wiringOP-Python can refer to the **blink.py** test program in the examples. The **blink.py** test program will set the voltage of all GPIO ports in the 40 pin development board to continuously change high and low.

```
(base) root@orangepi:~/wiringOP-Python# cd examples
(base) root@orangepi:~/wiringOP-Python/examples# ls blink.py
blink.py
(base) root@orangepi:~/wiringOP-Python/examples# python3 blink.py
```

3. 15. 3. 40pin SPI test

The functions of the 40 pin interface pins on the development board are shown in the table below. The pins highlighted in red have SPI function, and the Linux system is configured with SPI function by default, which can be used directly.

GPIO serial number	GPIO	function	pin
		3.3V	1
76	GPIO2_12	SDA7	3

pin	function	GPIO	GPIO serial number
2	5V		
4	5V		



75	GPI02_11	SCL7	5
226	GPI07_02	UTXD7	7
		GND	9
82	GPI02_18	URXD2	11
38	GPI01_06		13
79	GPI02_15		15
		3.3V	17
91	GPI02_27	SPI0_SDO	19
92	GPI02_28	SPI0_SDI	21
89	GPI02_25	SPI0_SCLK	23
		GND	25
		SDA6	27
231	GPI07_07	URXD7	29
84	GPI02_20		31
128	GPI04_00		33
228	GPI07_04		35
3	GPI00_03		37
		GND	39

6	GND		
8	UTXD0	GPI00_14	14
10	URXD0	GPI00_15	15
12		GPI07_03	227
14	GND		
16		GPI02_16	80
18		GPI00_25	25
20	GND		
22		GPI00_02	2
24	SPI0_CS	GPI02_26	90
26		GPI02_19	83
28	SCL6		
30	GND		
32	PWM3	GPI01_01	33
34	GND		
36	UTXD2	GPI02_17	81
38		GPI07_06	230
40		GPI07_05	229

The SPI bus in the 40 pin interface is SPI0. Before testing, please ensure that there is a **spidev0.0** device node under **/dev**.

```
(base) root@orangepi:~# ls /dev/spidev0.0
/dev/spidev0.0
```

Then you can use the **spidev_test.py** program in the examples to test the loopback function of SPI. The **spidev_test.py** program needs to specify the following two parameters:

- 1) **--channel**: Specify the channel number for SPI.
- 2) **--port**: Specify the port number for SPI.

First, do not short-circuit the mosi and miso pins of SPI. The output result of running **spidev_test.py** is shown below, and it can be seen that the data of TX and RX are inconsistent.

```
(base) root@orangepi:~/wiringOP-Python# cd examples
(base) root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py --channel 0 --port 0
```



```

spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev0.0
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF F0 0D |.....@.....|
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF |.....|

```

Then use a DuPont wire to short-circuit the txd and rxd pins of SPI and run the output of spidev_test.py as follows. It can be seen that the data sent and received is the same, indicating that the SPI loopback test is normal.

```

(base) root@orangepi:~/wiringOP-Python# cd examples
(base) root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py --channel 0 --port 0
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev0.0
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF F0 0D |.....@.....|
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF F0 0D |.....@.....|

```

3. 15. 4. 40pin I2C Test

The functions of the 40 pin interface pins on the development board are shown in the table below. The pins highlighted in red have I2C function, and the Linux system is configured with I2C function by default, which can be used directly.

GPIO serial number	GPIO	function	pin
		3.3V	1
76	GPIO2_12	SDA7	3
75	GPIO2_11	SCL7	5
226	GPIO7_02	UTXD7	7
		GND	9
82	GPIO2_18	URXD2	11
38	GPIO1_06		13
79	GPIO2_15		15

pin	function	GPIO	GPIO serial number
2	5V		
4	5V		
6	GND		
8	UTXD0	GPIO0_14	14
10	URXD0	GPIO0_15	15
12		GPIO7_03	227
14	GND		
16		GPIO2_16	80



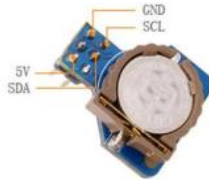
		3.3V	17	18		GPI00_25	25
91	GPI02_27	SPI0_SDO	19	20	GND		
92	GPI02_28	SPI0_SDI	21	22		GPI00_02	2
89	GPI02_25	SPI0_SCLK	23	24	SPI0_CS	GPI02_26	90
		GND	25	26		GPI02_19	83
		SDA6	27	28	SCL6		
231	GPI07_07	URXD7	29	30	GND		
84	GPI02_20		31	32	PWM3	GPI01_01	33
128	GPI04_00		33	34	GND		
228	GPI07_04		35	36	UTXD2	GPI02_17	81
3	GPI00_03		37	38		GPI07_06	230
		GND	39	40		GPI07_05	229

After starting the Linux system, first confirm the existence of i2c6 and i2c7 device nodes in **/dev**.

```
(base) root@orangepi:~# ls /dev/i2c-6
/dev/i2c-6
(base) root@orangepi:~# ls /dev/i2c-7
/dev/i2c-7
```

Then connect an I2C device to the i2c6 or i2c7 pin of the 40 pin interface, using the DS1307 RTC module as an example.

	i2c6	i2c7
Sda pin	Corresponding to pin 27 out of 40	Corresponding to pin 3 of the 40 pins
Scl pin	Corresponding to pin 28 out of 40	Corresponding to pin 5 out of 40
3.3v pin	Corresponding to pin 1 of the 40 pin	Corresponding to pin 1 of the 40 pin
Gnd pin	Corresponding to pin 6 of the 40 pin	Corresponding to pin 6 of the 40 pin



Then use the **i2cdetect** command. If the address of the connected i2c device can be detected, it means that i2c can be used normally.

1) The commands used by i2c6 are as follows:

```
(base) root@orangepi:~# i2cdetect -y -r 6
```

2) The commands used by i2c7 are as follows:

```
(base) root@orangepi:~# i2cdetect -y -r 7
```

```
(base) root@orangepi:~# i2cdetect -y -r 7
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- 68 -- -- -- -- -- --
70: -- -- -- -- -- -- --
(base) root@orangepi:~#
```

Then you can run the **ds1307.py** test program in the examples to read the RTC time.

1) The i2c6 test command is as follows:

```
(base) root@orangepi:~/wiringOP-Python# cd examples
(base) root@orangepi:~/wiringOP-Python/examples# python3 ds1307.py --device /dev/i2c-6
Wed 2025-02-19 10:31:33
Wed 2025-02-19 10:31:34
Wed 2025-02-19 10:31:35
^C
exit
```

2) The i2c7 test command is as follows:

```
(base) root@orangepi:~/wiringOP-Python# cd examples
(base) root@orangepi:~/wiringOP-Python/examples# python3 ds1307.py --device /dev/i2c-7
Wed 2025-02-19 11:56:00
```



Wed 2025-02-19 11:56:01

Wed 2025-02-19 11:56:02

^C

exit

3. 15. 5. 40pin UART Test

The functions of the 40 pin interface pins on the development board are shown in the table below. The pins highlighted in red have uart function, and the Linux system is configured with uart function by default, which can be used directly. Also, please note that uart0 is set to debug serial port functionality by default, and do not use it as a regular serial port.

GPI0 serial number	GPI0	function	pin
		3.3V	1
76	GPI02_12	SDA7	3
75	GPI02_11	SCL7	5
226	GPI07_02	UTXD7	7
		GND	9
82	GPI02_18	URXD2	11
38	GPI01_06		13
79	GPI02_15		15
		3.3V	17
91	GPI02_27	SPI0_SDO	19
92	GPI02_28	SPI0_SDI	21
89	GPI02_25	SPI0_SCLK	23
		GND	25
		SDA6	27
231	GPI07_07	URXD7	29
84	GPI02_20		31
128	GPI04_00		33
228	GPI07_04		35
3	GPI00_03		37
		GND	39

pin	function	GPI0	GPI0 serial number
2	5V		
4	5V		
6	GND		
8	UTXD0	GPI00_14	14
10	URXD0	GPI00_15	15
12		GPI07_03	227
14	GND		
16		GPI02_16	80
18		GPI00_25	25
20	GND		
22		GPI00_02	2
24	SPI0_CS	GPI02_26	90
26		GPI02_19	83
28	SCL6		
30	GND		
32	PWM3	GPI01_01	33
34	GND		
36	UTXD2	GPI02_17	81
38		GPI07_06	230
40		GPI07_05	229



After starting the Linux system, first confirm the existence of uart device nodes in **/dev**.

```
(base) root@orangepi:~# ls /dev/ttyAMA*
/dev/ttyAMA0 /dev/ttyAMA1 /dev/ttyAMA2
```

The corresponding relationship between UART device nodes and UART is as follows:

uart device node	uart interface
/dev/ttyAMA1	uart2
/dev/ttyAMA2	uart7

Then start testing the UART interface by short circuiting the RX and TX pins of the UART interface to be tested using DuPont wires. The pins in the 40 pin interface corresponding to the rx and tx pins of different UART are as follows:

uart interface	rx pin	tx pin
uart2	Pin 11 of 40pin	Pin 36 of 40pin
uart7	Pin 29 of 40pin	Pin 7 of 40pin

The serial port loopback function was tested using the **serialTest.py** program in the examples as shown below. If you can see the print below, it indicates that the serial communication is normal.

1) The uart2 test command is as follows:

```
(base) root@orangepi:~/wiringOP-Python# cd examples
(base) root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device /dev/ttyAMA1
Out: 0: -> 0
Out: 1: -> 1
Out: 2: ^C
exit
```

2) The uart7 test command is as follows:

```
(base) root@orangepi:~/wiringOP-Python# cd examples
(base) root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device /dev/ttyAMA2
Out: 0: -> 0
Out: 1: -> 1
Out: 2: ^C
```



```
exit
```

3. 16. How to upload files to the Linux system of the development board

3. 16. 1. How to upload files from Ubuntu PC to the Linux system of the development board

3. 16. 1. 1. How to upload files using the scp command

1) Use the scp command to upload files from the Ubuntu PC to the Linux system of the development board. The specific commands are as follows:

- a. **file_path**: Need to be replaced with the path to the file to be uploaded.
- b. **root**: It is the user name of the development board Linux system, which can also be replaced by other names, such as **HwHiAiUser**
- c. **192.168.xx.xx**: It is the IP address of the development board. Please modify it according to the actual situation.
- d. **/root**: The path in the Linux system of the development board can also be modified to other paths.

```
test@test:~$ scp file_path root@192.168.xx.xx:/root/
```

2) If you want to upload a folder, you need to add the **-r** parameter

```
test@test:~$ scp -r dir_path root@192.168.xx.xx:/root/
```

3) There are more uses for scp. Please use the following command to view the man manual.

```
test@test:~$ man scp
```

3. 16. 1. 2. How to upload files using FileZilla

1) First install filezilla in your Ubuntu PC.

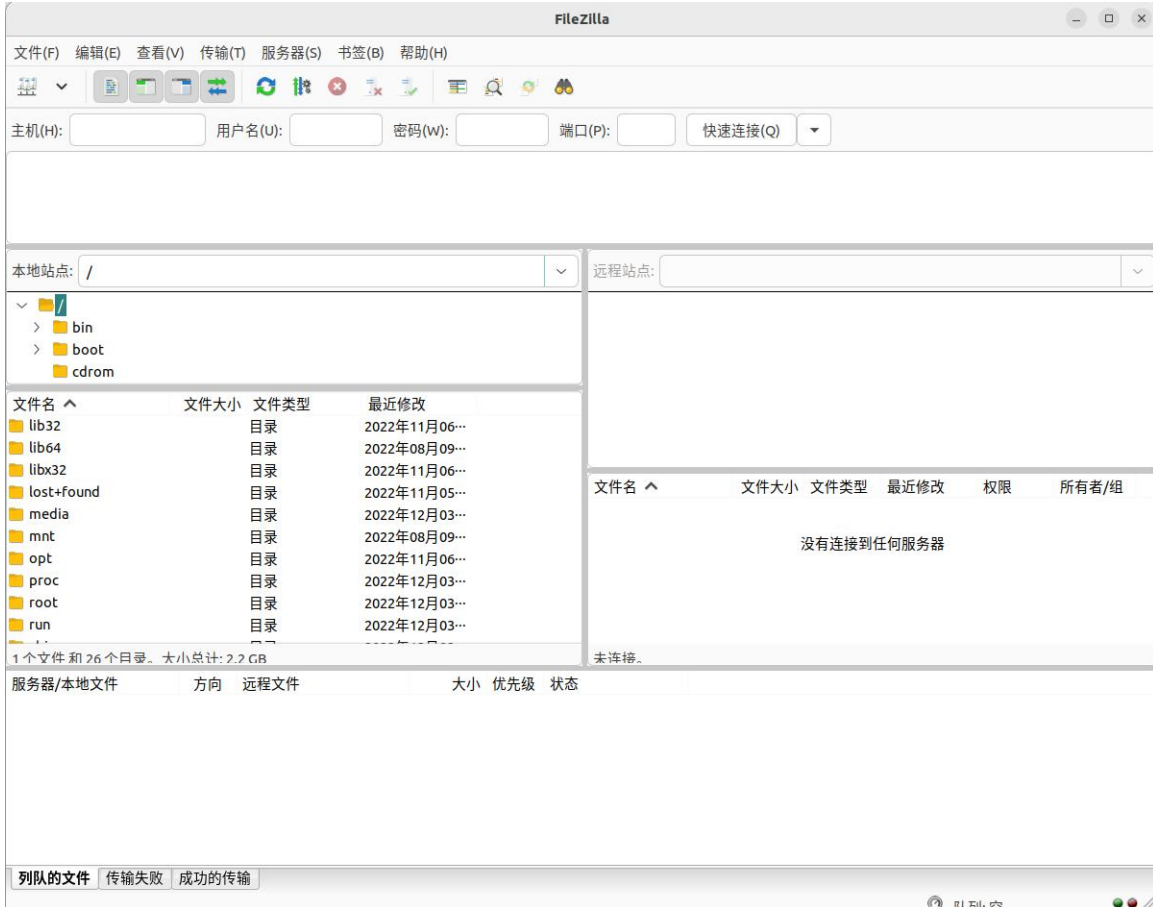
```
test@test:~$ sudo apt-get install -y filezilla
```

2) Then open filezilla using the command below.



```
test@test:~$ filezilla
```

3) The interface after opening filezilla is as shown below. At this time, the remote site on the right is empty.



4) The method of connecting the development board is shown in the figure below:



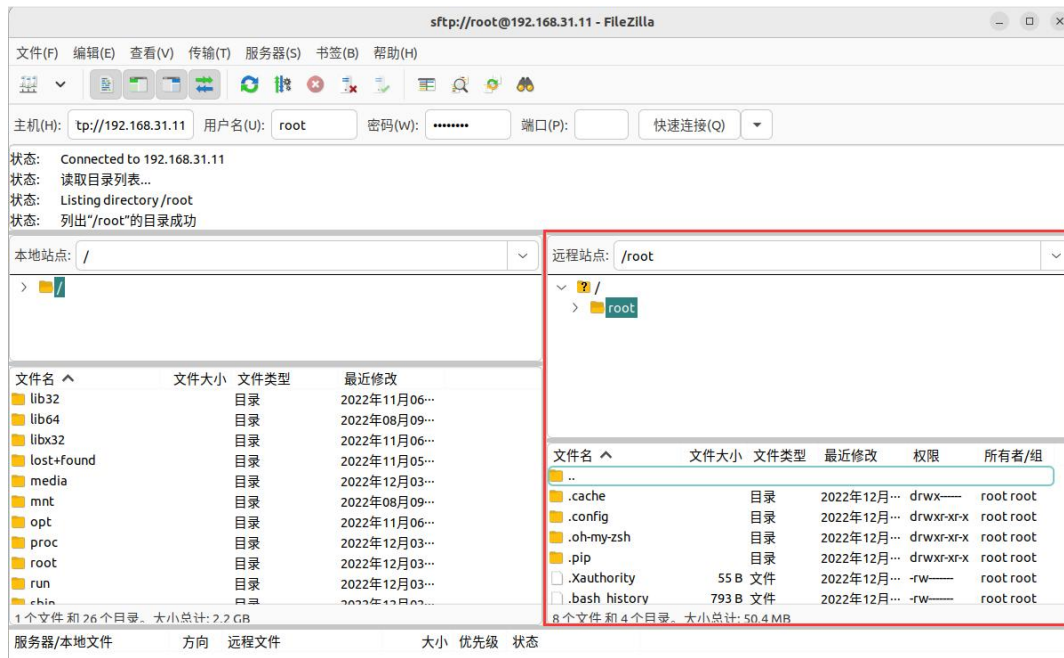
5) Then select **Save Password** and click **OK**.



6) Then select **Always trust this host** and click **OK**.



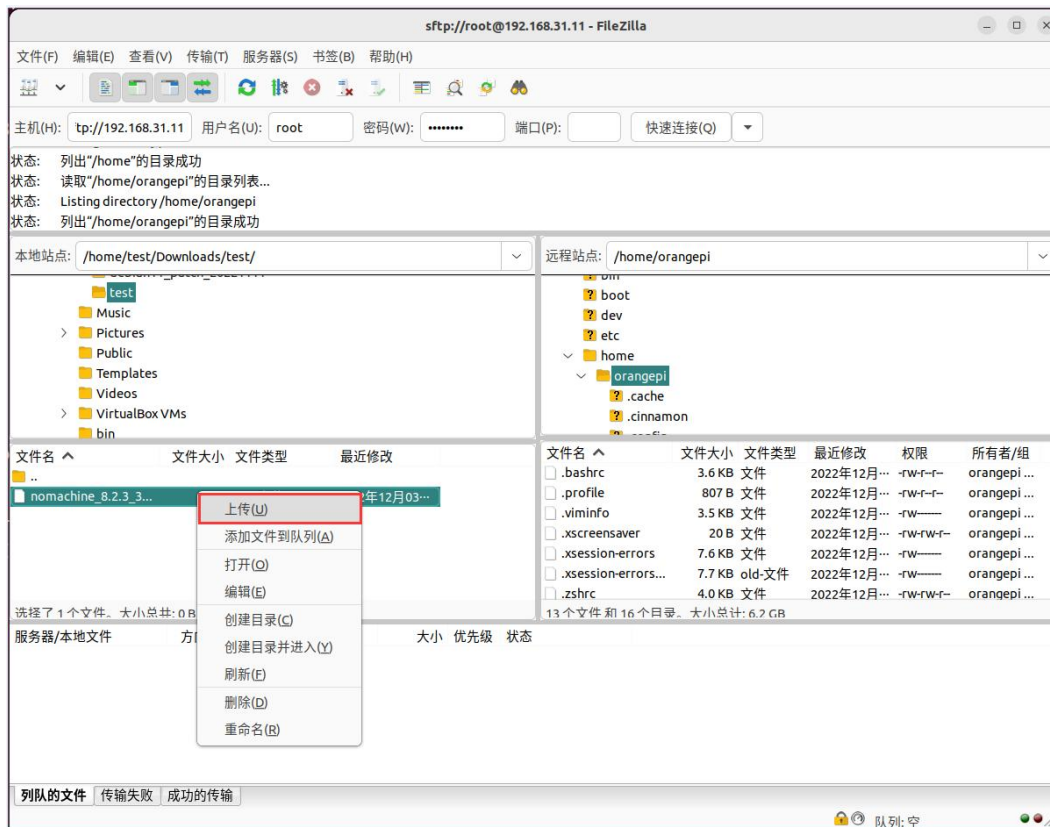
7) After the connection is successful, you can see the directory structure of the development board's Linux file system on the right side of the filezilla software.



8) Then select the path to be uploaded to the development board on the right side of the



filezilla software, then select the file to be uploaded in the Ubuntu PC on the left side of the filezilla software, right-click the mouse, and then click the upload option to start uploading the file to the development board.



9) After uploading is complete, you can go to the corresponding path in the Linux system of the development board to view the uploaded files.

10) The method for uploading a folder is the same as that for uploading a file, so I will not go into details here.

3. 16. 2. How to upload files from Windows PC to the Linux system of the development board

3. 16. 2. 1. How to upload files using FileZilla

1) First download the installation file of the Windows version of the filezilla software. The download link is as follows:



<https://filezilla-project.org/download.php?type=client>

FileZilla The free FTP solution

Home
FileZilla
Features
Screenshots
Download
Documentation
FileZilla Pro
FileZilla Server
Download
Community
Forum
Wiki
General
FAQ
Support
Contact
License
Privacy Policy
Trademark Policy
Development
Source code
Nightly builds
Translations
Version history
Changelog
Issue tracker
Other projects

Download FileZilla Client for Windows (64bit x86)
The latest stable version of FileZilla Client is 3.62.2
Please select the file appropriate for your platform below.

Windows (64bit x86)

Download FileZilla Client

Click here to download

This installer may include bundled offers. Check below for more options.
The 64bit versions of Windows 8.1, 10 and 11 are supported.

More download options

Other platforms:

Not what you are looking for?

Show additional download options

Please select your edition of FileZilla Client

	FileZilla	FileZilla with manual	FileZilla Pro	FileZilla Pro + CLI
Standard FTP	Yes	Yes	Yes	Yes
FTP over TLS	Yes	Yes	Yes	Yes
SFTP	Yes	Yes	Yes	Yes
Comprehensive PDF manual	-	Yes	Yes	Yes
Amazon S3	-	-	Yes	Yes
Backblaze B2	-	-	Yes	Yes
Dropbox	-	-	Yes	Yes
Microsoft OneDrive	-	-	Yes	Yes
Google Drive	-	-	Yes	Yes
Google Cloud Storage	-	-	Yes	Yes
Microsoft Azure Blob + File Storage	-	-	Yes	Yes
WebDAV	-	-	Yes	Yes
OpenStack Swift	-	-	Yes	Yes
Box	-	-	Yes	Yes
Site Manager synchronization	-	-	Yes	Yes
Command-line interface	-	-	-	Yes
Batch transfers	-	-	-	Yes

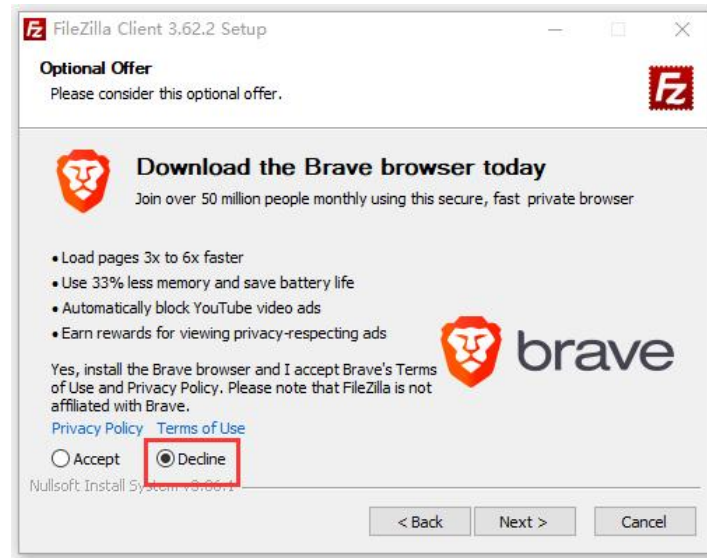
Then select here to download

Download **Select** **Select** **Select**

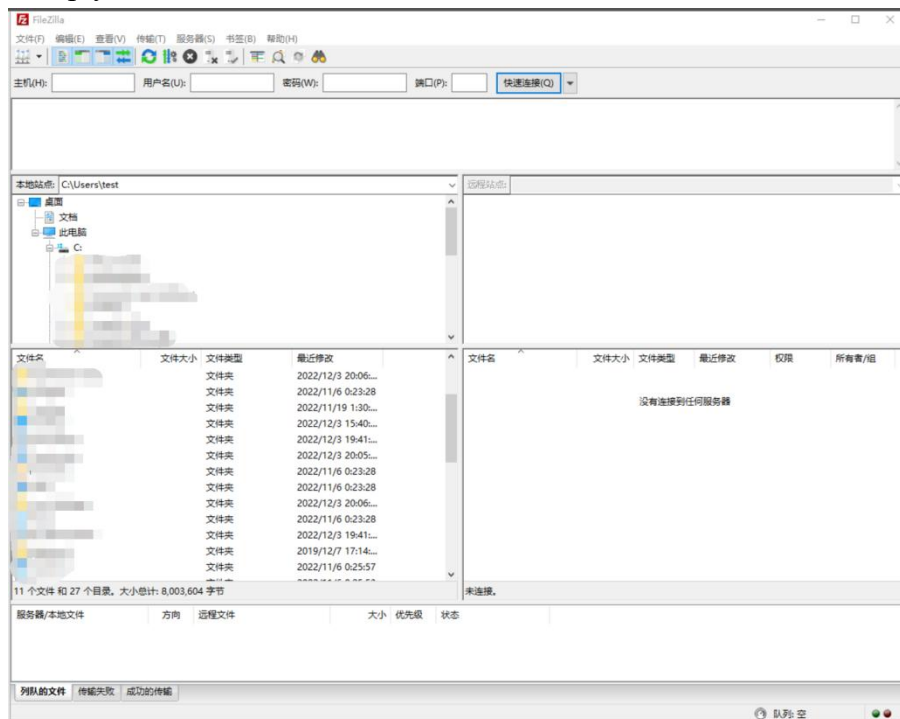
2) The downloaded installation package is as shown below, then double-click to install directly.

FileZilla_Server_x.x.x_win64-setup.exe

During the installation process, select **Decline** on the following installation interface, and then select **Next>**.



3) The interface after opening filezilla is as shown below. At this time, the remote site on the right is empty.



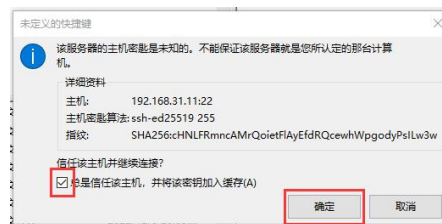
4) The method of connecting the development board is shown in the figure below:



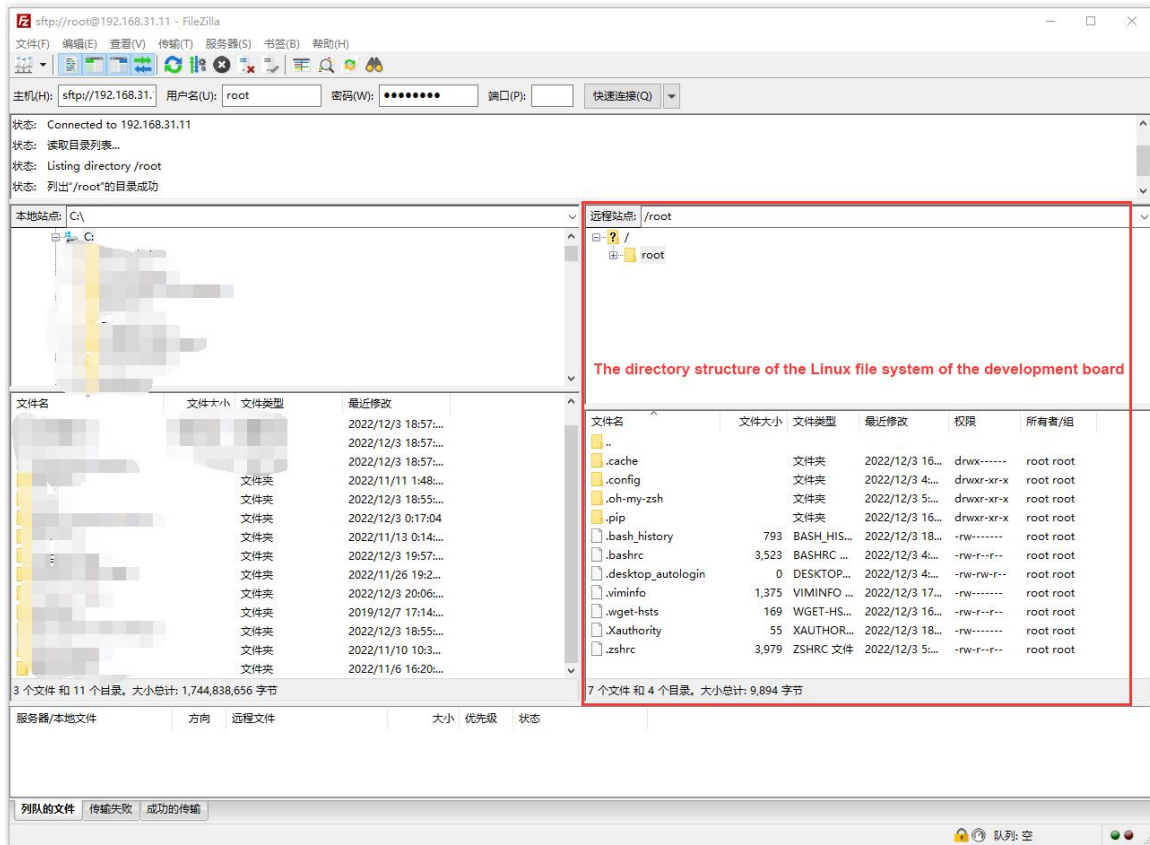
5) Then select **Save Password** and click **OK**.



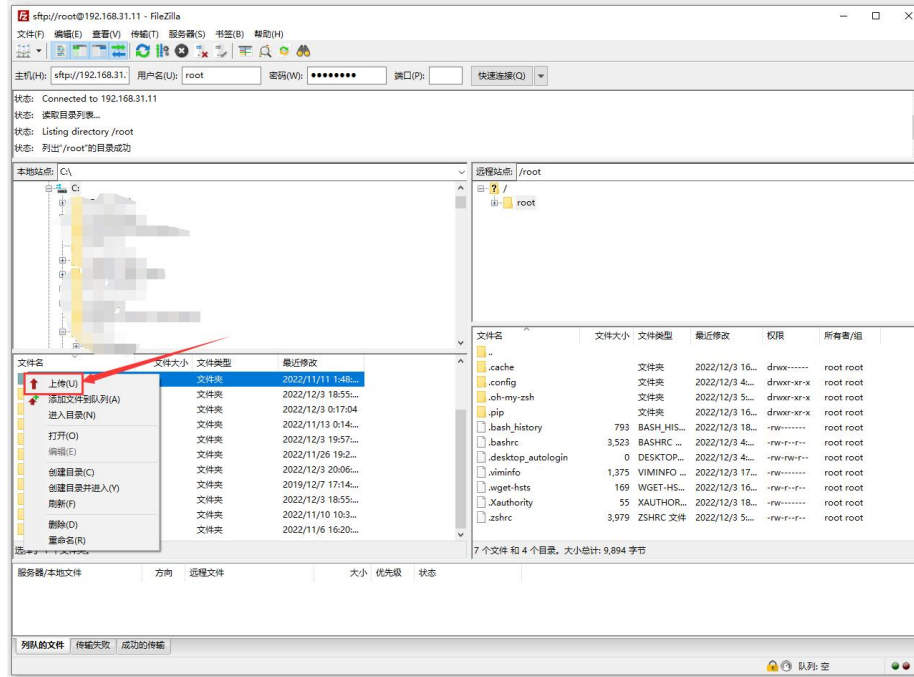
6) Then select **Always trust this host** and click **OK**.



7) After the connection is successful, you can see the directory structure of the development board's Linux file system on the right side of the filezilla software.



8) Then select the path to be uploaded to the development board on the right side of the filezilla software, then select the file to be uploaded in the Windows PC on the left side of the filezilla software, right-click the mouse, and then click the upload option to start uploading the file to the development board.



9) After uploading is complete, you can go to the corresponding path in the Linux system of the development board to view the uploaded files.

10) The method for uploading a folder is the same as that for uploading a file, so I will not go into details here.

3. 17. How to use the cooling fan

The location of the development board cooling fan interface is as follows:



The cooling fan used by the development board is 12V, with a 4-pin interface and a 1.0mm pitch. The fan speed can be controlled by PWM.



Use the npu-smi command to query and control the PWM fan. The detailed usage is as follows:

1) The command to query the fan mode is as follows:

(base) HwHiAiUser@orangepiaipro-20t:~\$ sudo npu-smi info -t pwm-mode	
pwm-mode : auto	
Fields	illustrate
pwm-mode	Fan Mode There are two modes:: • manual: Manual Mode • auto: Automatic mode The default mode is automatic.

2) The command to query the fan speed ratio is as follows:

(base) HwHiAiUser@orangepiaipro-20t:~\$ sudo npu-smi info -t pwm-duty-ratio	
pwm-duty-ratio(%) : 15	

3) The command to set the fan mode to manual mode is as follows:

(base) HwHiAiUser@orangepiaipro-20t:~\$ sudo npu-smi set -t pwm-mode -d 0	
describe	
Fan enabling mode. It is divided into manual mode and automatic mode. The default is automatic mode. • 0: Manual Mode • 1: Automatic mode	

4) After setting the fan to manual mode, you can use the following command to set the fan speed ratio. For example, the following command will set the fan speed ratio to 100, and the fan will run at the maximum speed after setting.

(base) HwHiAiUser@orangepiaipro-20t:~\$ sudo npu-smi set -t pwm-duty-ratio -d 100	
describe	



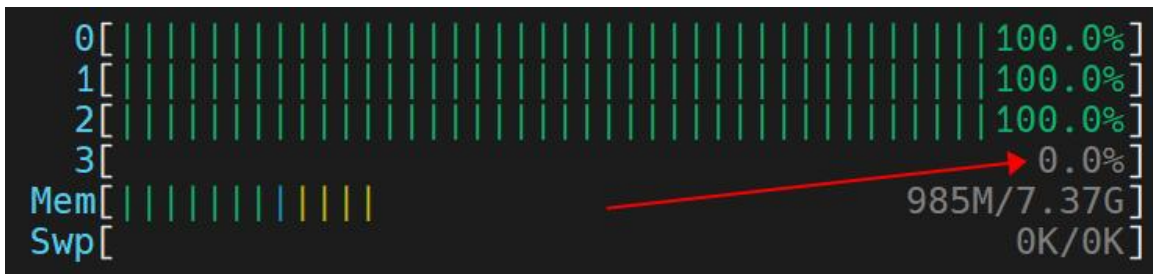
describe
Fan speed ratio
Ranges: [0-100]

3. 18. Setting method of AI CPU and control CPU

The Shengteng SOC used by the development board has a total of 4 CPUs. These 4 CPUs can be set as control CPUs or AI CPUs. By default, the allocation number of control CPU and AI CPU is 3:1. Use the **npu-smi info** command to check the allocated numbers of control CPU and AI CPU.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ npu-smi info -t cpu-num-cfg -i 0 -c 0
Current AI CPU number      : 1
Current control CPU number : 3
Current data CPU number    : 0
(base) HwHiAiUser@orangepiaipro-20t:~$
```

When the Linux system is fully utilized, you will see that the usage of one CPU is always close to 0 using the **htop** command. Please note that this is normal because this CPU is used by default for AI CPU.



If there is no AI CPU operator in the current environment model and the AI CPU usage is continuously 0 when running services, you can configure the number of AI CPUs to 0. The command to query the AI CPU usage is as follows:

```
(base) HwHiAiUser@orangepiaipro-20t:~$ npu-smi info -t usages -i 0 -c 0
Memory Capacity(MB)      : 7545
Memory Usage Rate(%)     : 20
Hugepages Total(page)    : 15
Hugepages Usage Rate(%)  : 100
```




```

Aicore Usage Rate(%)      : 0
Aicpu Usage Rate(%)      : 0
Ctrlcpu Usage Rate(%)    : 1
Memory Bandwidth Usage Rate(%) : 1

```

If you do not need to use the AI CPU, you can use the following command to set all four CPUs as control CPUs. After setting, you need to restart the system for the configuration to take effect.

```

(base) HwHiAiUser@orangepiaipro-20t:~$ sudo npu-smi set -t cpu-num-cfg -i 0 -c 0 -v 0:4:0
Status      : OK
Message : The cpu-num-cfg of the chip is set successfully. Reset system for the configuration to take effect.

```

After all four CPUs are set as control CPUs, run tasks to fully utilize all CPUs. Using the htop command, you can see that the usage of all four CPUs has reached 100%.

```

0[|||||||||||||||||||||||||||||||||||||100.0%]
1[|||||||||||||||||||||||||||||||||||||100.0%]
2[|||||||||||||||||||||||||||||||||||||100.0%]
3[|||||||||||||||||||||||||||||||||||||100.0%]
Mem[|||||||||||||||||913M/7.37G]
Swp[|||||0K/0K]

```

3. 19. How to set swap memory

Although the development board has 12GB or 24GB of memory, some applications require more memory than 12GB or 24GB. In this case, we can use swap memory to expand the maximum memory capacity that the system can use. The method is as follows:

1) First, create a swap file. The following command will create a 16GB swap file. Please modify the capacity according to your needs.

```

(base) HwHiAiUser@orangepiaipro-20t:~$ sudo fallocate -l 16G /swapfile

```

If the Swap partition is enabled and you run the fallocate command, the following error will be reported:

fallocate: fallocate failed: Text file busy

You need to run the `swapoff /swapfile` command to close the swap partition on the system first.



Note: Before adding Swap memory, make sure that the remaining space on the TF card, eMMC or SSD is larger than the capacity of the Swap memory to be added.

2) Then modify the file permissions to ensure that only the root user can read and write.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo chmod 600 /swapfile
```

3) Then set this file as swap space.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo mkswap /swapfile
```

4) Then enable swap.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo swapon /swapfile
```

5) After completing the above steps, you can use the following command to check whether the swap memory has been added successfully.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ free -h
```

	total	used	free	shared	buff/cache	available
Mem:	11Gi	802Mi	9.8Gi	48Mi	723Mi	10Gi
Swap:	15Gi	0B	15Gi			

6) If you need the swap settings to remain valid after reboot, run the following command to add the corresponding configuration to the `/etc/fstab` file.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab
```

3. 20. Methods for testing MindSpore

MindSpore is pre-installed in the Ubuntu system of the development board. You can use the following method to test whether MindSpore can be used normally.

1) First, please add an additional 16GB of swap memory to the system according to the instructions in the section [How to set up swap memory](#).

2) Then execute the following command in the **HwHiAiUser** user:

```
(base) HwHiAiUser@orangepiaipro-20t:~$ python -c \
"import mindspore;mindspore.set_context(device_target='Ascend');mindspore.run_check()"
```

3) After waiting for a while, if you can see the following output, it means MindSpore can be used normally.

```
MindSpore version: 2.x.xx.2024xxxx
```



The result of multiplication calculation is correct, MindSpore has been installed on platform [Ascend] successfully!

4) The command to update the MindSpore version is as follows:

```
(base) HwHiAiUser@orangepiaipro-20t:~$ pip install --upgrade mindspore
```

3. 21. Using ascend hardware accelerated ffmpeg

FFmpeg is an open-source project that provides a set of libraries and programs for processing audio and video content, enabling functions such as recording, converting, and streaming audio and video. FFmpeg supports multiple media formats and can handle almost all common audio and video data.

We can significantly improve the speed of video encoding and decoding by using the FFmpeg hardware codec implemented through the DVPP interface. By offloading video processing tasks to dedicated hardware on the Ascend chip, the CPU load can be significantly reduced, accelerating the processing of video and image data.

3. 21. 1. Using compiled deb software packages

1) The deb format software package that supports Ascend hardware codec can be downloaded from the data download page of the development board. The steps are:

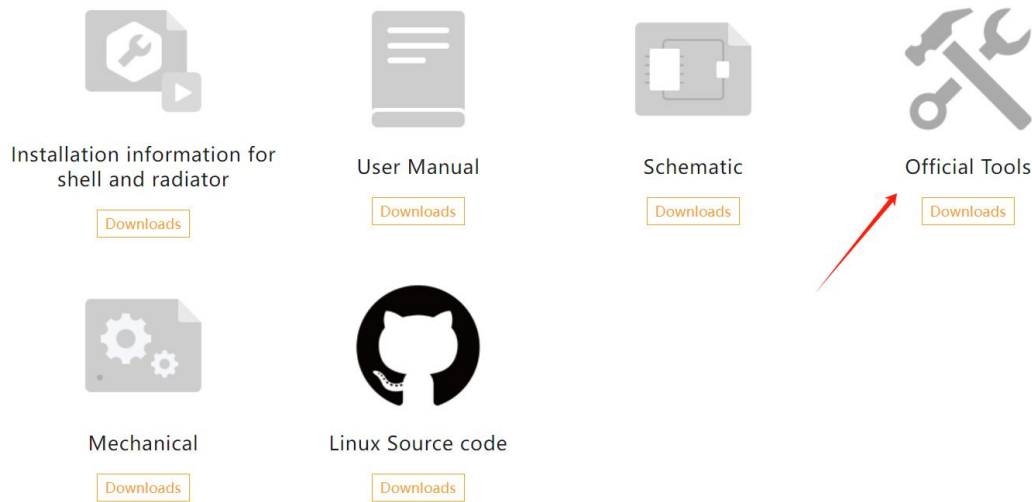
a. Open the following link:

[http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro\(20t\).html](http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro(20t).html)

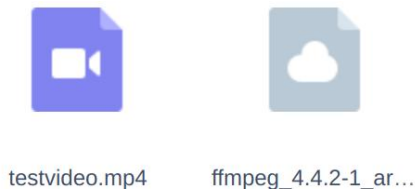
b. Then choose the official tool.



Official Resources



c. Then download the **ffmpeg_4.4.2-1_arm64.deb** package and test video from the FFmpeg_ascend folder.



2) Then refer to the manual to **upload the file to the Linux system on the development board** and upload it to the following directory on the development board.

```
/home/HwHiAiUser/
```

3) First, add the following two environment variables at the end of the "/home/HwHiAiUser/.bashrc" file and make them effective.

```
HwHiAiUser@orangepi:~$ vim .bashrc
source /usr/local/Ascend/ascend-toolkit/set_env.sh
export LD_LIBRARY_PATH=/usr/lib:/usr/local/Ascend/ascend-toolkit/latest/acclib/lib64:
$LD_LIBRARY_PATH
HwHiAiUser@orangepi:~$ source .bashrc
```

4) Execute the following command on the development board to install the software package.



```
HwHiAiUser@orangepi:~$ ls
ffmpeg_4.4.2-1_arm64.deb
HwHiAiUser@orangepi:~$ sudo dpkg -i ffmpeg_4.4.2-1_arm64.deb
(Reading database ... 129897 files and directories currently installed.)
Preparing to unpack ffmpeg_4.4.2-1_arm64.deb ...
Unpacking ffmpeg (4.4.2-1) over (4.4.2-1) ...
Setting up ffmpeg (4.4.2-1) ...
Processing triggers for man-db (2.10.2-1) ...
```

5) Verify installation. Enter the following 3 commands in sequence. If the contents listed below exist at the end of the output, it indicates that the codec installation is successful.

```
HwHiAiUser@orangepi:~$ ffmpeg -hwaccels | grep ascend
ascend
HwHiAiUser@orangepi:~$ ffmpeg -encoders | grep ascend
V..... h264_ascend      Ascend HiMpi H264 encoder (codec h264)
V..... h265_ascend      Ascend HiMpi H265 encoder (codec hevc)
HwHiAiUser@orangepi:~$ ffmpeg -decoders | grep ascend
V..... h264_ascend      Ascend HiMpi H264 decoder (codec h264)
V..... h265_ascend      Ascend HiMpi H265 decoder (codec hevc)
```

3. 21. 2. Building from Source Code

1) The codec patch package can be downloaded from the data download page of the development board. The steps are:

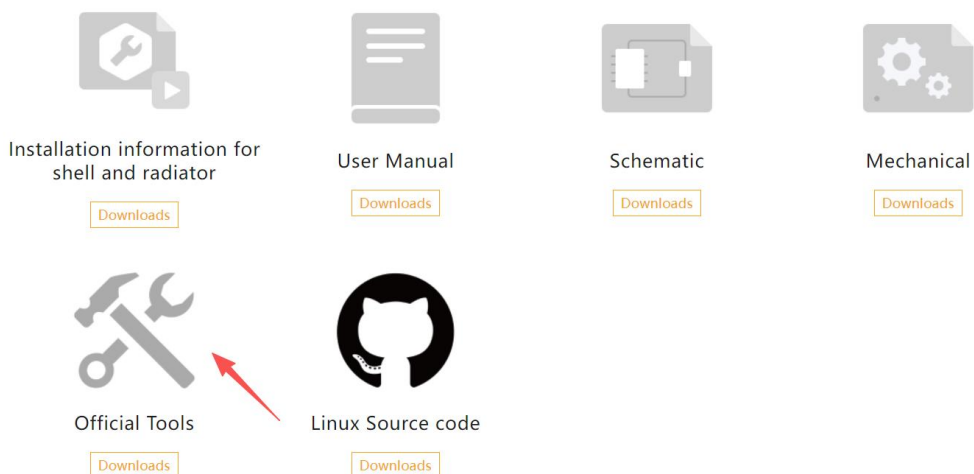
a. Open the following link:

[http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro\(20t\).html](http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro(20t).html)

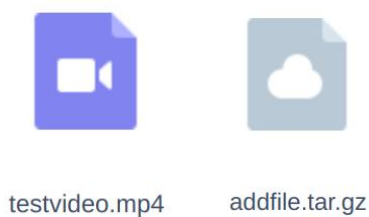
b. Then choose the official tool.



Official Resources

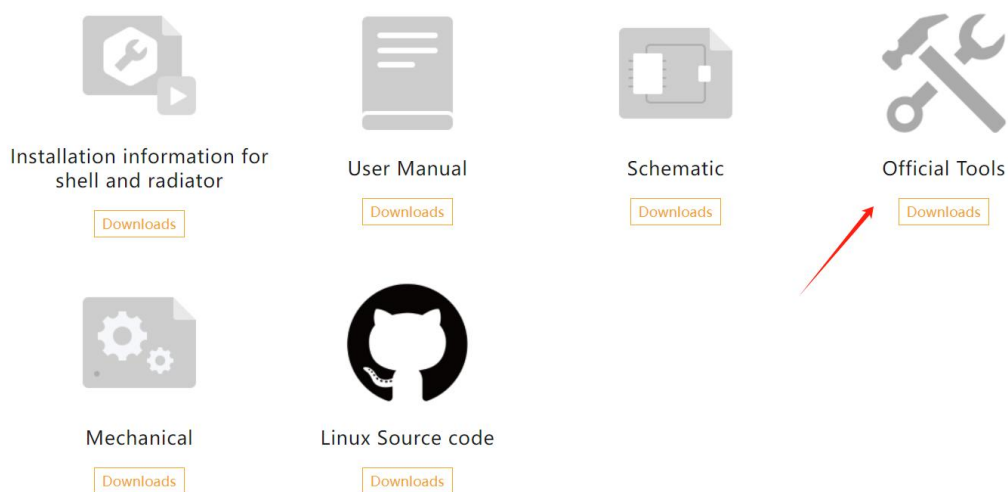


- c. Then download **addfile.tar.gz** and the test video from the FFmpeg_ascend folder.



- d. Then choose the official tool

Official Resources



- e. Then download **addfile.tar.gz** and the test video from the FFmpeg_ascend folder.



testvideo.mp4



addfile.tar.gz

2) Then refer to the manual to **upload the file to the Linux system on the development board** and upload it to the following directory on the development board.

```
/home/HwHiAiUser/
```

3) Next, download the ffmpeg source code. First, open the **/etc/apt/sources.list** configuration file, then remove the comments before **deb-src**, use the **apt update** command to update the cache of the package list, and then use the **apt source ffmpeg** command to download the ffmpeg source code.

```
HwHiAiUser@orangepi:~$ sudo vim /etc/apt/sources.list
deb http://repo.huaweicloud.com/ubuntu-ports/ jammy main restricted universe multivers
e
deb http://repo.huaweicloud.com/ubuntu-ports/ jammy-security main restricted universe
multiverse
deb http://repo.huaweicloud.com/ubuntu-ports/ jammy-updates main restricted universe
multiverse
deb http://repo.huaweicloud.com/ubuntu-ports/ jammy-proposed main restricted universe
multiverse
deb-src http://repo.huaweicloud.com/ubuntu-ports/ jammy main restricted universe multi
verse
deb-src http://repo.huaweicloud.com/ubuntu-ports/ jammy-security main restricted univer
se multiverse
deb-src http://repo.huaweicloud.com/ubuntu-ports/ jammy-updates main restricted univer
se multiverse
deb-src http://repo.huaweicloud.com/ubuntu-ports/ jammy-proposed main restricted univ
erse multiverse
HwHiAiUser@orangepi:~$ sudo apt update
HwHiAiUser@orangepi:~$ sudo apt-get install dpkg-dev
HwHiAiUser@orangepi:~$ apt source ffmpeg
```

4) After running the **apt source ffmpeg** command, the current directory will have



additional files and folders as shown below, where the **ffmpeg-4.4.2** folder is the directory of the ffmpeg source code.

```
HwHiAiUser@orangepi:~$ ls | grep ffmpeg
ffmpeg-4.4.2
ffmpeg_4.4.2-0ubuntu0.22.04.1.debian.tar.xz
ffmpeg_4.4.2-0ubuntu0.22.04.1.dsc
ffmpeg_4.4.2.ascend-1_arm64.deb
ffmpeg_4.4.2.orig.tar.xz
ffmpeg_4.4.2.orig.tar.xz.asc
```

5) Then extract the patch file of ffmpeg dvpp and copy it to the folder of ffmpeg source code.

```
HwHiAiUser@orangepi:~$ tar xzf addfile.tar.gz
HwHiAiUser@orangepi:~$ cp -r addfile/* ./ffmpeg-4.4.2
```

6) Then add the following two environment variables at the end of the "/home/HwHiAiUser/.bashrc" file and make them effective.

```
HwHiAiUser@orangepi:~$ vim .bashrc
source /usr/local/Ascend/ascend-toolkit/set_env.sh
export LD_LIBRARY_PATH=/usr/lib:/usr/local/Ascend/ascend-toolkit/latest/acclib/lib64:
$LD_LIBRARY_PATH
HwHiAiUser@orangepi:~$ source .bashrc
```

7) Then you can start compiling and installing ffmpeg packages with dvpp functionality. The specific commands are as follows:

```
HwHiAiUser@orangepi:~$ cd ffmpeg-4.4.2
HwHiAiUser@orangepi:~/ffmpeg-4.4.2$ ./configure \
    --prefix=/usr \
    --enable-shared \
    --extra-cflags="-I${ASCEND_HOME_PATH}/acclib/include" \
    --extra-ldflags="-L${ASCEND_HOME_PATH}/aarch64-linux/lib64" \
    --extra-libs="-lacl_dvpp_mpi -lascendcl" \
    --enable-ascend
HwHiAiUser@orangepi:~/ffmpeg-4.4.2$ make -j4
HwHiAiUser@orangepi:~/ffmpeg-4.4.2$ sudo make install
```



Note that the configuration listed after `./configure` is the simplest necessary configuration. If you need to add other configurations, add them yourself in the format, such as `--enable-libx264`.

8) Verify installation. Enter the following 3 commands in sequence. If the contents listed below exist in the last few lines of the output, it indicates that the codec installation is successful.

```
HwHiAiUser@orangePi:~$ ffmpeg -hwaccels | grep ascend
ascend
HwHiAiUser@orangePi:~$ ffmpeg -encoders | grep ascend
V.... h264_ascend      Ascend HiMpi H264 encoder (codec h264)
V.... h265_ascend      Ascend HiMpi H265 encoder (codec hevc)
HwHiAiUser@orangePi:~$ ffmpeg -decoders | grep ascend
V.... h264_ascend      Ascend HiMpi H264 decoder (codec h264)
V.... h265_ascend      Ascend HiMpi H265 decoder (codec hevc)
```

3. 21. 3. Application Scenarios

3. 21. 3. 1. Video Encoding and Decoding

Here we take the H264 format test video provided as an example. Running the following command can use ffmpeg to call the h264_ascend decoder to decode the test video (the usage of h265_ascend is the same as h264_ascend, just change the two h264 in the following command to h265):

```
HwHiAiUser@orangePi:~$ ffmpeg -i testvideo.mp4 -vcodec h264_ascend test.264
```

The output of ffmpeg runtime is shown below, and running the entire transcoding process takes approximately 205s



```

HwHiAiUser@orangepiaipro-20: ~/Desktop/testvideo
frame= 9339 fps=181 q=-0.0 size= 758528kB time=00:02:35.53 bitrate=39952.0kbits/s dup=2 drop=0 speed=3.02x
frame= 9432 fps=181 q=-0.0 size= 768000kB time=00:02:37.08 bitrate=40051.7kbits/s dup=2 drop=0 speed=3.02x
frame= 9523 fps=181 q=-0.0 size= 776960kB time=00:02:38.60 bitrate=40131.5kbits/s dup=2 drop=0 speed=3.02x
frame= 9620 fps=182 q=-0.0 size= 785152kB time=00:02:40.21 bitrate=40145.4kbits/s dup=2 drop=0 speed=3.02x
frame= 9712 fps=182 q=-0.0 size= 790272kB time=00:02:41.75 bitrate=40024.2kbits/s dup=2 drop=0 speed=3.02x
frame= 9815 fps=182 q=-0.0 size= 795136kB time=00:02:43.46 bitrate=39847.6kbits/s dup=2 drop=0 speed=3.03x
frame= 9908 fps=182 q=-0.0 size= 802304kB time=00:02:45.01 bitrate=39829.2kbits/s dup=2 drop=0 speed=3.03x
frame= 9995 fps=182 q=-0.0 size= 810496kB time=00:02:46.46 bitrate=39885.4kbits/s dup=2 drop=0 speed=3.03x
frame=10086 fps=182 q=-0.0 size= 818176kB time=00:02:47.98 bitrate=39899.8kbits/s dup=2 drop=0 speed=3.03x
frame=10175 fps=182 q=-0.0 size= 826880kB time=00:02:49.46 bitrate=39971.3kbits/s dup=2 drop=0 speed=3.03x
frame=10272 fps=182 q=-0.0 size= 835072kB time=00:02:51.08 bitrate=39985.8kbits/s dup=2 drop=0 speed=3.03x
frame=10367 fps=182 q=-0.0 size= 840704kB time=00:02:52.66 bitrate=39886.4kbits/s dup=2 drop=0 speed=3.03x
frame=10461 fps=182 q=-0.0 size= 849920kB time=00:02:54.23 bitrate=39961.0kbits/s dup=2 drop=0 speed=3.03x
frame=10567 fps=182 q=-0.0 size= 853760kB time=00:02:56.00 bitrate=39738.6kbits/s dup=2 drop=0 speed=3.03x
frame=10673 fps=182 q=-0.0 size= 857856kB time=00:02:57.76 bitrate=39532.5kbits/s dup=2 drop=0 speed=3.04x
frame=10776 fps=183 q=-0.0 size= 864256kB time=00:02:59.48 bitrate=39446.5kbits/s dup=2 drop=0 speed=3.04x
frame=10867 fps=183 q=-0.0 size= 872448kB time=00:03:01.00 bitrate=39486.7kbits/s dup=2 drop=0 speed=3.04x
frame=10951 fps=182 q=-0.0 size= 880128kB time=00:03:02.40 bitrate=39528.6kbits/s dup=2 drop=0 speed=3.04x
frame=11051 fps=183 q=-0.0 size= 887808kB time=00:03:04.06 bitrate=39512.4kbits/s dup=2 drop=0 speed=3.04x
frame=11139 fps=183 q=-0.0 size= 896512kB time=00:03:05.53 bitrate=39584.4kbits/s dup=2 drop=0 speed=3.04x
frame=11231 fps=183 q=-0.0 size= 905728kB time=00:03:07.06 bitrate=39663.5kbits/s dup=2 drop=0 speed=3.04x
frame=11323 fps=183 q=-0.0 size= 914688kB time=00:03:08.60 bitrate=39730.2kbits/s dup=2 drop=0 speed=3.04x
frame=11419 fps=183 q=-0.0 size= 922624kB time=00:03:10.20 bitrate=39737.8kbits/s dup=2 drop=0 speed=3.04x

```

From the above figure, we can see that during the transcoding process, the average fps is over 180 and the speed is around 3 times. As a comparison, the following are the results of libx264 transcoding, with an average fps of only over 20 and a time consumption of over 1700 seconds.

```
HwHiAiUser@orangepi:~$ ffmpeg -i testvideo.mp4 -vcodec libx264 test.264
```

Note that if it is compiled and installed, the configuration given above is the simplest and necessary configuration, and the libx264 encoder is not enabled. Running it will prompt "Unknown encoder 'libx264'". Please first execute the 'sudo apt install libx264-dev' command to install the libx264 package, and then follow the instructions above to add configurations' --enable-libx264'and' --enable-gpl 'before recompiling and installing.

```

HwHiAiUser@orangepiaipro-20: ~/Desktop/testvideo
frame= 1910 fps= 24 q=31.0 size= 13824kB time=00:00:30.96 bitrate=3657.0kbits/s dup=2 drop=0 speed=0.389x
frame= 1919 fps= 24 q=31.0 size= 13824kB time=00:00:31.11 bitrate=3639.4kbits/s dup=2 drop=0 speed=0.388x
frame= 1927 fps= 24 q=31.0 size= 14080kB time=00:00:31.25 bitrate=3691.0kbits/s dup=2 drop=0 speed=0.387x
frame= 1935 fps= 24 q=31.0 size= 14080kB time=00:00:31.38 bitrate=3675.3kbits/s dup=2 drop=0 speed=0.387x
frame= 1941 fps= 24 q=31.0 size= 14080kB time=00:00:31.48 bitrate=3663.6kbits/s dup=2 drop=0 speed=0.385x
frame= 1948 fps= 24 q=31.0 size= 14592kB time=00:00:31.60 bitrate=3782.8kbits/s dup=2 drop=0 speed=0.384x
frame= 1958 fps= 24 q=31.0 size= 14848kB time=00:00:31.76 bitrate=3829.0kbits/s dup=2 drop=0 speed=0.384x
frame= 1969 fps= 24 q=31.0 size= 14848kB time=00:00:31.95 bitrate=3807.0kbits/s dup=2 drop=0 speed=0.383x
frame= 1978 fps= 24 q=31.0 size= 14848kB time=00:00:32.10 bitrate=3789.2kbits/s dup=2 drop=0 speed=0.382x
frame= 1987 fps= 24 q=31.0 size= 15104kB time=00:00:32.25 bitrate=3836.7kbits/s dup=2 drop=0 speed=0.382x
frame= 1995 fps= 23 q=31.0 size= 15104kB time=00:00:32.38 bitrate=3820.9kbits/s dup=2 drop=0 speed=0.381x
frame= 2003 fps= 23 q=31.0 size= 15104kB time=00:00:32.51 bitrate=3805.2kbits/s dup=2 drop=0 speed=0.38x
frame= 2013 fps= 23 q=31.0 size= 15360kB time=00:00:32.68 bitrate=3849.9kbits/s dup=2 drop=0 speed=0.379x
frame= 2022 fps= 23 q=31.0 size= 15360kB time=00:00:32.83 bitrate=3832.4kbits/s dup=2 drop=0 speed=0.379x
frame= 2030 fps= 23 q=31.0 size= 15616kB time=00:00:32.96 bitrate=3880.5kbits/s dup=2 drop=0 speed=0.378x
frame= 2037 fps= 23 q=31.0 size= 15616kB time=00:00:33.08 bitrate=3866.8kbits/s dup=2 drop=0 speed=0.377x
frame= 2046 fps= 23 q=31.0 size= 15616kB time=00:00:33.23 bitrate=3849.3kbits/s dup=2 drop=0 speed=0.376x
frame= 2054 fps= 23 q=31.0 size= 15872kB time=00:00:33.36 bitrate=3896.8kbits/s dup=2 drop=0 speed=0.376x
frame= 2063 fps= 23 q=31.0 size= 15872kB time=00:00:33.51 bitrate=3879.4kbits/s dup=2 drop=0 speed=0.375x
frame= 2071 fps= 23 q=31.0 size= 15872kB time=00:00:33.65 bitrate=3864.0kbits/s dup=2 drop=0 speed=0.375x
frame= 2078 fps= 23 q=31.0 size= 16128kB time=00:00:33.76 bitrate=3912.8kbits/s dup=2 drop=0 speed=0.374x
frame= 2086 fps= 23 q=31.0 size= 16128kB time=00:00:33.90 bitrate=3897.4kbits/s dup=2 drop=0 speed=0.373x
frame= 2093 fps= 23 q=31.0 size= 16384kB time=00:00:34.01 bitrate=3945.6kbits/s dup=2 drop=0 speed=0.372x

```

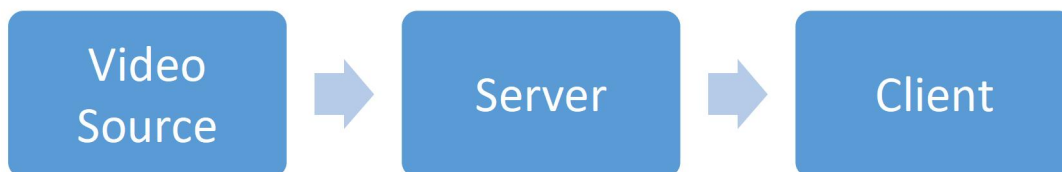


Codec	Average fps	Time Consuming
libx264	22	1702s
h264_ascend	186	205s

From this, we can see that calling the ascend codec can improve the performance by about 9 times compared to libx264, effectively enhancing the encoding and decoding performance. For video streams that require real-time processing, hardware decoding can ensure that video data is processed quickly, thereby maintaining or improving real-time performance in scenarios such as object detection. Hardware decoding enables the model to receive large amounts of data faster, thereby increasing the scale of batch processing, which helps improve the throughput of model inference.

3. 21. 3. 2. Live streaming promotion

The entire process can be roughly divided into three parts. Firstly, use a camera or video source to collect video data, and push the collected data to the server through streaming software. Users pull video streams from the server through client software and present them after decoding and rendering.



3. 21. 3. 2. 1. Setting up the server

Here we use an open-source project called mediamtx (project address: <https://github.com/bluenvoron/mediamtx>) As a server. We can directly download the compiled standalone binary file from the download page of the development board. Of course, you can also refer to the project documentation to install it on other platforms or use other servers.

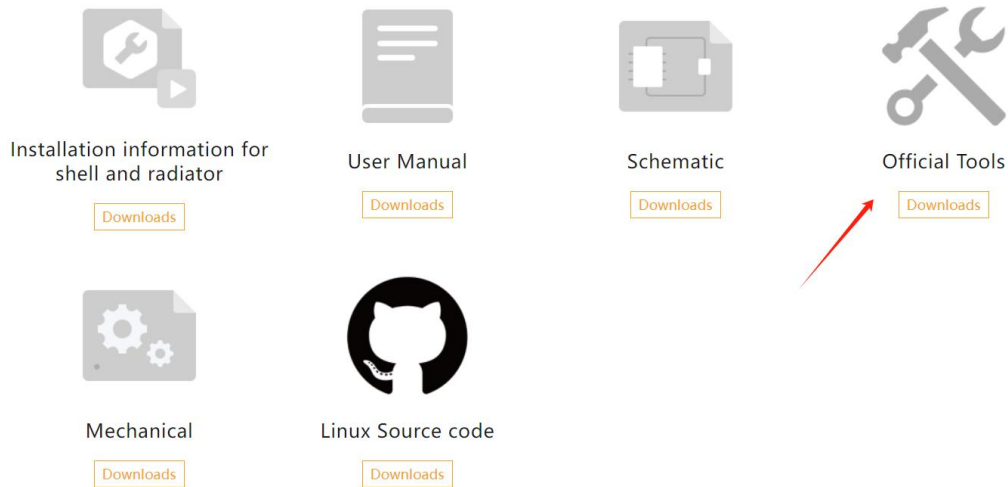
- a. Open the following link:

[http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro\(20t\).html](http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro(20t).html)

- b. Then choose the official tool.



Official Resources



- c. Then download the **mediamtx_v1.8.4_linux_arm64v8.tar.gz** compressed file from the FFmpeg_ascend folder.



mediamtx_v1.8.4_...

- d. Then refer to the manual to **upload the file to the Linux system on the development board** and upload it to the following directory on the development board.

```
/home/HwHiAiUser/
```

- e. Then use the following command to unzip the software to the '~/mediamtx' directory.

```
HwHiAiUser@orangePi:~$ mkdir mediamtx
HwHiAiUser@orangePi:~$ tar -zxvf mediamtx_v1.8.4_linux_arm64v8.tar.gz -C mediamtx
HwHiAiUser@orangePi:~$ cp mediamtx/mediamtx.yml ./
```

- f. Query the server IP, for example, 10.31.3.120 below is the server IP. Please do not copy the IP address, and refer to what you actually see.

```
HwHiAiUser@orangePi:~$ ifconfig
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.31.3.120 netmask 255.255.0.0 broadcast 10.31.255.255
```




```
inet6 fe80::6532:3661:21ab:ebf1 prefixlen 64 scopeid 0x20<link>
ether c0:74:2b:fd:f0:f7 txqueuelen 1000 (Ethernet)
```

g. Start the service. If there are special requirements, please modify the configuration file first.

```
HwHiAiUser@orangePi:~$ vim ./mediamtx.yml          #Modify configuration
HwHiAiUser@orangePi:~$ sudo mediamtx/mediamtx      #Start service
2024/07/18 20:21:40 INF MediaMTX v1.8.4
2024/07/18 20:21:40 INF configuration loaded from /home/HwHiAiUser/mediamtx.yml
2024/07/18 20:21:40 INF [RTSP] listener opened on :8554 (TCP), :8000 (UDP/RTP), :8001 (UDP/RTCP)
2024/07/18 20:21:40 INF [RTMP] listener opened on :1935
2024/07/18 20:21:40 INF [HLS] listener opened on :8888
2024/07/18 20:21:40 INF [WebRTC] listener opened on :8889 (HTTP), :8189 (ICE/UDP)
2024/07/18 20:21:40 INF [SRT] listener opened on :8890 (UDP)
```

3. 21. 3. 2. 2. Pushing Flow

Here are two video input methods for streaming: camera image input and offline video input.

1) The method of streaming the camera image input is as follows:

- a) Firstly, connect the USB camera to the USB interface of the development board. After connecting the camera, you can first refer to the instructions in the [USB camera testing](#) section to test the USB camera and ensure that it can be used normally.

Note that if you need to use two USB cameras simultaneously in the current version, please connect one to the development board using the Type-C interface and the other to the development board using the USB3.0 HOST interface. If both USB cameras are connected to the USB3.0 HOST interface, there will be a problem of insufficient bandwidth.

Currently, up to 2 USB2.0 cameras can be connected.

- b) Then execute the following command to start streaming the video captured by the USB camera.

```
root@orangePi:~# export LD_LIBRARY_PATH=/usr/lib:/usr/local/Ascend/ascend-toolkits/latest/acllib/lib64:$LD_LIBRARY_PATH
```



```
root@orangepi:~# ffmpeg -f v4l2 -input_format mjpeg -video_size 1920x1080 -framerate 30 -i /dev/video0 -c:v h264_ascend -b:v 1000k -maxrate 5000k -bufsize 256k -g 5 -keyint_min 5 -rtsp_transport udp -f rtsp rtsp://10.31.3.120:8554/mystream1
```

Please fill in /dev/video0 in the command according to the actual situation, and replace the ip address of the rtsp video stream with the IP address of the server above. If the streaming device and server are the same device, please use rtsp://localhost:8554/mystream1 Otherwise, the streaming will fail.

- c) The parameters related to the ffmpeg streaming command are described as follows:

-f v4l2	Specify the use of V4L2 (Video4Linux2) driver interface to read video devices
-input_format mjpeg	Specify the input video format as MJPEG (yuv format will be limited to 5fps)
-video_size 1920x1080	Set the video resolution to 1920x1080 (1080p)
-framerate 30	Set the frame rate of the video to 30 frames per second
-i /dev/video0	Specify input device as video0
-c:v h264_ascend	Specify video encoder as h264_ascend
-b:v 1000k	Set the bit rate of the video to 1000 kbit/s
-maxrate 5000k	Set the maximum bit rate for the video
-bufsize 256k	Set the buffer size of the encoder
-g 5	Set GOP size to 5 frames (reducing GOP size helps reduce latency)
-keyint_min 5	Set the minimum keyframe interval to 5 frames (corresponding to the -g parameter)
-rtsp_transport udp	Use UDP protocol to transmit RTSP streams (tcp is recommended for those with image quality requirements)
-f rtsp	Specify the output format as RTSP
rtsp://10.31.3.120:8554/mystream1	Specify the address and stream name of the RTSP server, port 8554 on 10.31.3.120, and stream name mystream1

- d) The log of the streaming end is as follows, printing out the relevant configuration information and status during streaming:



```
[swscaler @ 0xaaaaed198130] deprecated pixel format used, make sure you did set
range correctly
[h264_ascend_enc @ 0xaaaaed16ebc0] Device id is: 0.
[AVHWDeviceContext @ 0xaaaaed1f62f0] device id is: 0.
[h264_ascend_enc @ 0xaaaaed16ebc0] Create venc channels success. Channel id is 0
.
Encode thread start.
Output #0, rtsp, to 'rtsp://10.31.3.120:8554/mystream1':
  Metadata:
    encoder      : Lavf58.76.100
  Stream #0:0: Video: h264, nv12(tv, bt470bg/unknown/unknown, progressive), 1920
x1080, q=2-31, 1000 kb/s, 30 fps, 90k tbn
  Metadata:
    encoder      : Lavc58.134.100 h264_ascend
[rtsp @ 0xaaaaed169a50] Timestamps are unset in a packet for stream 0. This is d
eprecated and will stop working in the future. Fix your code to set the timestam
ps properly
[rtsp @ 0xaaaaed169a50] Encoder did not produce proper pts, making some up.
frame= 15 fps= 13 q=-0.0 size=N/A time=00:00:00.23 bitrate=N/A dup=13 drop=0 s
frame= 29 fps= 17 q=-0.0 size=N/A time=00:00:00.70 bitrate=N/A dup=18 drop=0 s
frame= 45 fps= 21 q=-0.0 size=N/A time=00:00:01.23 bitrate=N/A dup=24 drop=0 s
frame= 62 fps= 23 q=-0.0 size=N/A time=00:00:01.80 bitrate=N/A dup=30 drop=0 s
frame= 78 fps= 24 q=-0.0 size=N/A time=00:00:02.33 bitrate=N/A dup=36 drop=0 s
frame= 94 fps= 25 q=-0.0 size=N/A time=00:00:02.86 bitrate=N/A dup=42 drop=0 s
```

- e) There will be the following log on the server, indicating that the server has received the RTSP format video stream "mystream1" pushed from IP "10.31.1.109":

```
2024/07/18 14:09:34 INF [RTSP] [conn 10.31.1.109:45706] opened
2024/07/18 14:09:34 INF [RTSP] [session 5116116d] created by 10.31.1.109:45706
2024/07/18 14:09:34 INF [RTSP] [session 5116116d] is publishing to path 'mystream1', 1
track (H264)
```

- 2) The method for streaming offline videos is as follows:

- a) The push stream command is as follows:

```
HwHiAiUser@orangePi:~$ ffmpeg -re -i testvideo.mp4 -c:a aac -b:a 128k -ac 2 -ar 44
100 -c:v h264_ascend -b:v 2000k -maxrate 5000k -bufsize 256k -rtsp_transport udp
-f rtsp rtsp://10.31.3.120:8554/mystream2
```

- b) The difference in parameters compared to the camera above is the use of the "-re" parameter, which allows for a loop to push the current video.

Please note that this parameter "-c:a aac -b:a 128k -ac 2 -ar 44100" must be added when streaming. Otherwise, the following error will be reported:



```

Encode thread start.
[aac @ 0xaaaadb75bfa0] Using a PCE to encode channel layout "5.1(side)"
Could not write header for output file #0 (incorrect codec parameters ?): Server
returned 400 Bad Request
Error initializing output stream 0:1 --
[h264_ascend_enc @ 0xaaaadb638840] Enc sem_timewait = -1, time out, semvalue = 0
...
[h264_ascend_enc @ 0xaaaadb638840] Call hi_mpi_venc_query_status failed, ret is
-1610055675.
[h264_ascend_enc @ 0xaaaadb638840] Call get_stream_loop failed, ret is -1.
[aac @ 0xaaaadb75bfa0] Qavg: nan
Conversion failed!
(base) HwHiAiUser@orangepiaipro-20t:~/Desktop/testvideo$

```

If the streaming device and server are the same device, please use `rtsp://localhost:8554/mystream2` Otherwise, the streaming will fail.

3. 21. 3. 2. 3. Flow pulling

Here we can use the `ffplay` tool to directly play the video stream we push to the server. Use the following command to play, remember to replace the ip with the server ip:

```
HwHiAiUser@orangepi:~$ ffplay rtsp://10.31.3.120:8554/mystream1
```

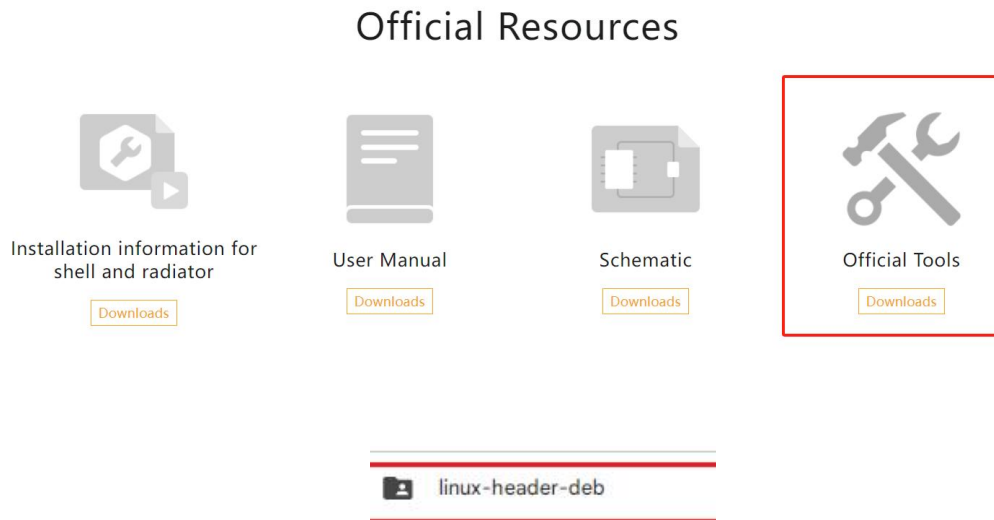
If the streaming device and server are the same device, please also use the server's ip address. Using localhost instead of ip will result in an error. Before running, please verify if the stream address is correct, especially the last stream name 'mystream1'.





3. 22. Method for installing kernel header files

1) The Linux system of the development board cannot directly use the apt command to install kernel header files, and requires the use of specially made deb packages. The pre made deb package can be downloaded from the **official tool** on the development board information download page.



2) After downloading, please upload the **linux-header-deb** folder to the Linux system of the development board. For the method of uploading files to the Linux system of the development board, please refer to the instructions in the section on **uploading files to the Linux system of the development board**. Then use the following command to install the deb package of the kernel header file.

```
root@orangepi:~# cd linux-header-deb
root@orangepi:~/linux-header-deb# sudo apt update
root@orangepi:~/linux-header-deb# sudo apt install -y ./linux-headers-linux-5.10.0_1.0_arm64.deb
```

3) After installation, you can see the folder where the kernel header files are located in **/usr/src**.

```
root@orangepi:~/linux-header-deb# ls /usr/src
linux-headers-5.10.0+
```

4) Then you can test whether the helle_test kernel module in the **linux-header-deb** folder



can compile and run normally:

- a. Firstly, use the **make** command to compile the **hello** kernel module. The output of the compilation process is as follows:

```
root@orangepi:~/linux-header-deb# cd helle_test
root@orangepi:~/linux-header-deb/helle_test# make
make -C /lib/modules/5.10.0+/build M=/root/linux-header-deb/helle_test modules
make[1]: Entering directory '/usr/src/linux-headers-5.10.0+'
CC [M] /root/linux-header-deb/helle_test/hello.o
MODPOST /root/linux-header-deb/helle_test/Module.symvers
CC [M] /root/linux-header-deb/helle_test/hello.mod.o
LD [M] /root/linux-header-deb/helle_test/hello.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.10.0+'

```

- b. After compilation, the **hello.ko** kernel module will be generated.

```
root@orangepi:~/linux-header-deb# ls *.ko
hello.ko

```

- c. The **insmod** command can be used to insert the **hello.ko** kernel module into the kernel

```
root@orangepi:~/linux-header-deb# sudo insmod hello.ko

```

- d. Then use the **dmesg** command to view the output of the **hello.ko** kernel module. If you can see the following output, it indicates that the **hello.ko** kernel module is loaded correctly.

```
root@orangepi:~/linux-header-deb# dmesg | grep "Hello"
[ 2871.893988] Hello Orange Pi -- init

```

- e. The **rmmod** command can be used to uninstall the **hello.ko** kernel module.

```
root@orangepi:~/linux-header-deb# sudo rmmod hello
root@orangepi:~/linux-header-deb# dmesg | grep "Hello"
[ 2871.893988] Hello Orange Pi -- init
[ 3173.800892] Hello Orange Pi -- exit

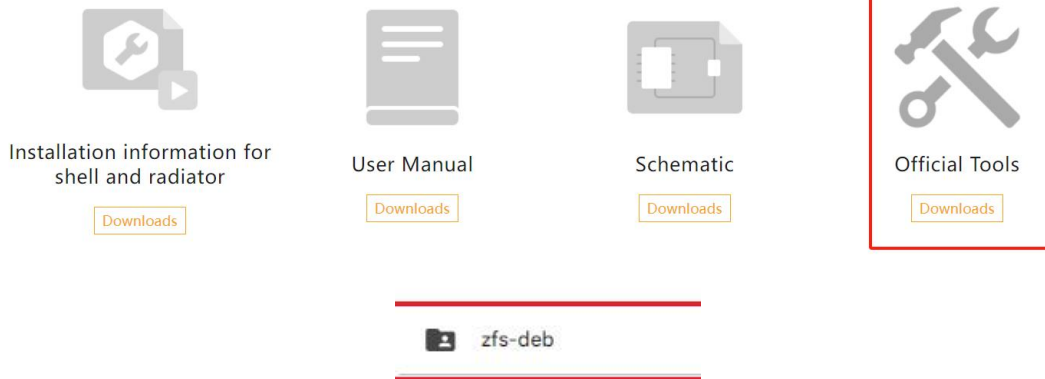
```

3. 23. Method of installing ZFS

1) In Ubuntu 22.04 system, zfs cannot be installed directly through apt. To solve this problem, we provide deb packages of zfs that can be installed normally, which can be downloaded from the [official tools](#) of the development board. Open the **official tool** and download the **zfs-deb** folder.



Official Resources



2) After downloading zfs-deb, please upload it to the Linux system of the development board. Please refer to the instructions in the section on [uploading files to the Linux system on the development board](#) for the upload method.

3) After the transfer is complete, use the following command to install the deb package of zfs. **Before running the following command, ensure that the network is unblocked.**

```
root@orangepi:~# cd zfs-deb
root@orangepi:~/zfs-deb# sudo apt update
root@orangepi:~/zfs-deb# sudo apt -y install ./*.deb
```

4) After installation, use the following command to see the zfs related kernel modules:

```
root@orangepi:~/zfs-deb# ls /lib/modules/5.10.0+/extra/zcommon
spl.ko zfs.ko
```

5) Then restart the linux system and you will see that the zfs kernel module will automatically load:

```
root@orangepi:~# lsmod | grep zfs
zfs          4665344  0
spl          131072  1 zfs
```

6) Use the following command to confirm if zfs has been installed and loaded correctly.

```
root@orangepi:~# zfs version
```



```
zfs-2.2.99-687_gb3b749161
zfs-kmod-2.2.99-687_gb3b749161
```

7) You can now use zfs in the usual way. For example, creating a new zfs pool. If there are no extra disks to create the zfs pool, a virtual disk can be created.

```
root@orangePi:~# truncate -s 1G /tmp/zfs-disk.img
root@orangePi:~# sudo losetup /dev/loop0 /tmp/zfs-disk.img
root@orangePi:~# sudo zpool create myzpool /dev/loop0
```

8) After creation, you can use the following command to view the status of the pool:

```
root@orangePi:~# zpool status
pool: myzpool
state: ONLINE
config:

    NAME      STATE    READ WRITE CKSUM
    myzpool    ONLINE   0      0      0
        loop0  ONLINE   0      0      0

errors: No known data errors
```

9) Method for creating zfs file system.

- a) The following command can be used to create a zfs file system on the created zfs pool.

```
root@orangePi:~# sudo zfs create myzpool/mydataset
```

- b) You can check if the file system has been successfully created using the following command:

```
root@orangePi:~# zfs list
NAME                USED  AVAIL REFER  MOUNTPOINT
myzpool              138K  832M    24K    /myzpool
myzpool/mydataset    24K   832M    24K    /myzpool/mydataset
```

- c) Then you can write some files to the newly created zfs file system and verify if it works properly. It can be seen that the strings written and read are the same, indicating that the zfs file system is working properly.

```
root@orangePi:~# sudo touch /myzpool/mydataset/testfile.txt
```



```
root@orangepi:~# echo "Hello, ZFS!" | sudo tee /myzpool/mydataset/testfile.txt
Hello, ZFS!
root@orangepi:~# cat /myzpool/mydataset/testfile.txt
Hello, ZFS!
```

3. 24. How to shut down and restart the development board

1) If you unplug the power supply directly while the Linux system is running, some data may be lost in the file system. It is recommended that you use the **poweroff** command to shut down the Linux system of the development board before unplugging the power supply.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo poweroff
```

2) In addition to the **poweroff** command to shut down the Linux system, you can also use the power button on the development board to shut down the Linux system of the development board, and then unplug the power.



3) After shutting down, short press the power button on the development board to turn it on.

4) Use the **reboot** command to restart the Linux system in the development board.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ sudo reboot
```




4. Experience AI application examples (openEuler)

We have pre installed Jupyter Lab software in the image. Jupyter Lab software is a web-based interactive development environment that integrates features such as a code editor, terminal, and file manager, allowing developers to complete various tasks in one interface. And we have also pre-set some AI application samples in the image that can run in Jupyter Lab software. These examples are all written in Python and call the Python version of the AscendCL programming interface. This chapter introduces how to log in to Jupyter Lab and run these pre installed AI application samples in Jupyter Lab.

4. 1. Login to Jupyter Lab

1) Firstly, log in to the Linux system desktop, then open the terminal and switch to the directory where the MindSpore AI application sample is saved.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ cd samples
```

2) There are 8 folders and 1 shell file in the current directory, corresponding to 8 AI application samples and Jupyter Lab startup script **start_notebook.sh**, respectively.

```
(base) HwHiAiUser@orangepiaipro-20t:~/samples$ ls
01-SSD 02-CNNCTC 03-ResNet50 04-HDR 05-CycleGAN 06-Shufflenet 07-FCN 08-Pix2Pix start_notebook.sh
```



3) Then execute the start_notebook.sh script to start Jupyter Lab.

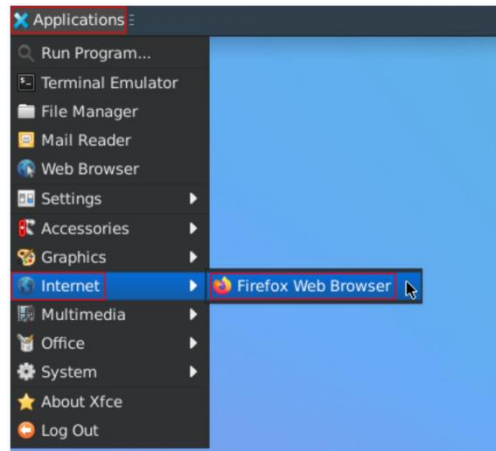
```
(base) HwHiAiUser@orangepiaipro-20t:~/samples$ ./start_notebook.sh
```

4) After executing the script, the terminal will display the following print message, which will include a link to log in to Jupyter Lab.

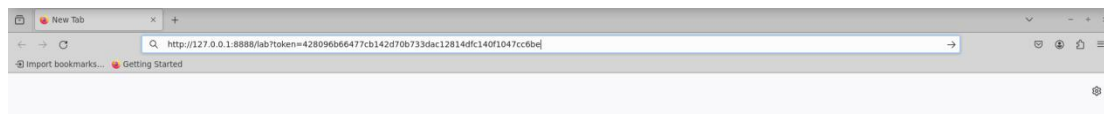
```
(base) HwHiAiUser@orangepiaipro:~/samples$ ./start_notebook.sh
[W 2024-06-03 17:35:23.674 ServerApp] A '_jupyter_server_extension_points' function was not found in notebook shim. Instead, a '_jupyter_server_extension_path'
s' function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2024-06-03 17:35:23.675 ServerApp] jupyter lsp | extension was successfully linked.
[I 2024-06-03 17:35:23.687 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2024-06-03 17:35:23.702 ServerApp] jupyterlab | extension was successfully linked.
[I 2024-06-03 17:35:23.705 ServerApp] Writing Jupyter server cookie secret to /home/HwHiAiUser/.local/share/jupyter/runtime/jupyter_cookie_secret
[I 2024-06-03 17:35:24.300 ServerApp] notebook_shim | extension was successfully linked.
[I 2024-06-03 17:35:24.558 ServerApp] notebook_shim | extension was successfully loaded.
[I 2024-06-03 17:35:24.564 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-06-03 17:35:24.567 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2024-06-03 17:35:24.573 LabApp] JupyterLab extension loaded from /usr/local/miniconda3/lib/python3.9/site-packages/jupyterlab
[I 2024-06-03 17:35:24.573 LabApp] JupyterLab application directory is /usr/local/miniconda3/share/jupyter/lab
[I 2024-06-03 17:35:24.575 LabApp] Extension Manager is 'pypi'.
[I 2024-06-03 17:35:24.582 ServerApp] jupyterlab | extension was successfully loaded.
[I 2024-06-03 17:35:24.583 ServerApp] Serving notebooks from local directory: /home/HwHiAiUser/samples
[I 2024-06-03 17:35:24.583 ServerApp] Jupyter Server 2.12.5 is running at:
[I 2024-06-03 17:35:24.584 ServerApp] http://127.0.0.1:8888/lab?token=7c876b0dc78ff8853bd93bdf1e42a6454cf685e2c0c6d141
[I 2024-06-03 17:35:24.584 ServerApp] http://127.0.0.1:8888/lab?token=7c876b0dc78ff8853bd93bdf1e42a6454cf685e2c0c6d141
[I 2024-06-03 17:35:24.584 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

To access the server, open this file in a browser:
file:///home/HwHiAiUser/.local/share/jupyter/runtime/jpserver-4434-open.html
Or copy and paste one of these URLs:
http://127.0.0.1:8888/lab?token=7c876b0dc78ff8853bd93bdf1e42a6454cf685e2c0c6d141
http://127.0.0.1:8888/lab?token=7c876b0dc78ff8853bd93bdf1e42a6454cf685e2c0c6d141
[I 2024-06-03 17:35:24.659 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-server-nodejs, javascript-typescript-langserv
er, jedi-language-server, julia-language-server, pyright, python-language-server, python-lsp-server, r-language-server, sql-language-server, texlab, typescript
-language-server, unified-language-server, vscode-css-language-server-bin, vscode-html-language-server-bin, vscode-json-language-server-bin, yamll-language-server
```

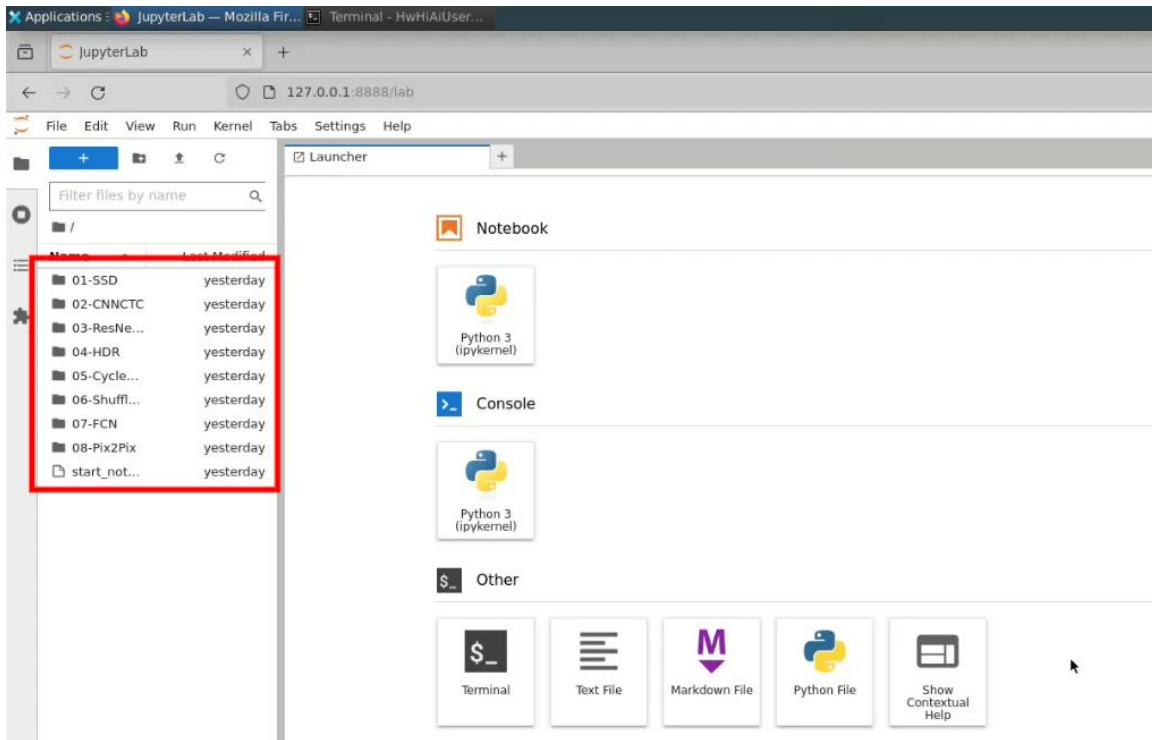
5) Then open Firefox browser.



6) Then enter the URL link you see above in the browser to log in to the Jupyter Lab software.



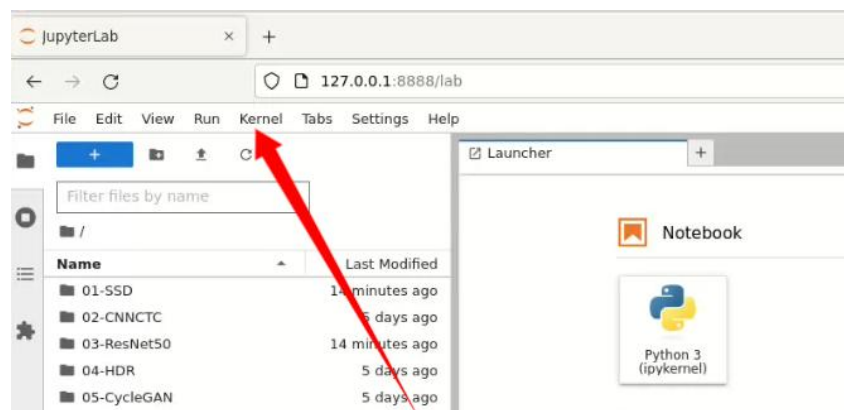
7) The interface after logging into Jupyter Lab is shown below. On the left side of the file manager are 8 AI application samples and Jupyter Lab startup scripts.



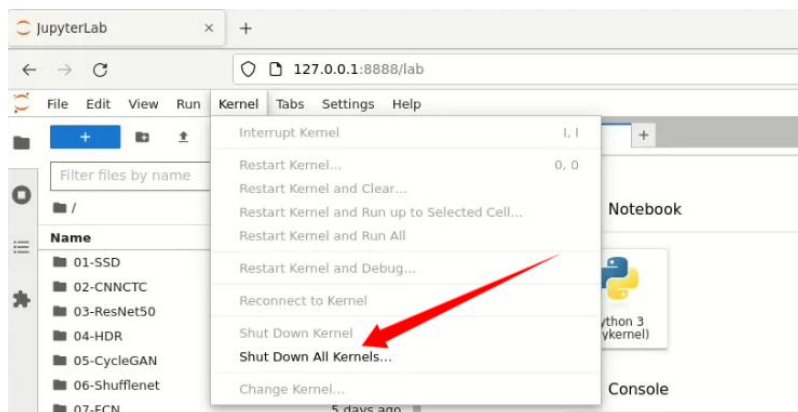
4. 2. Methods for freeing memory

After running a sample in Jupyter Lab, closing the sample's tab will not release the memory occupied by the corresponding sample. We can reclaim memory by closing the Kernel, as follows:

1) First, choose **Kernel**.



2) Then select **Shut Down All Kernels...**.



3) Then select **Shut Down All** to free up the memory occupied by the sample.



After testing each sample, it is recommended to free up memory before running the next sample to avoid sample failure due to insufficient memory.

4. 3. Running object detection samples

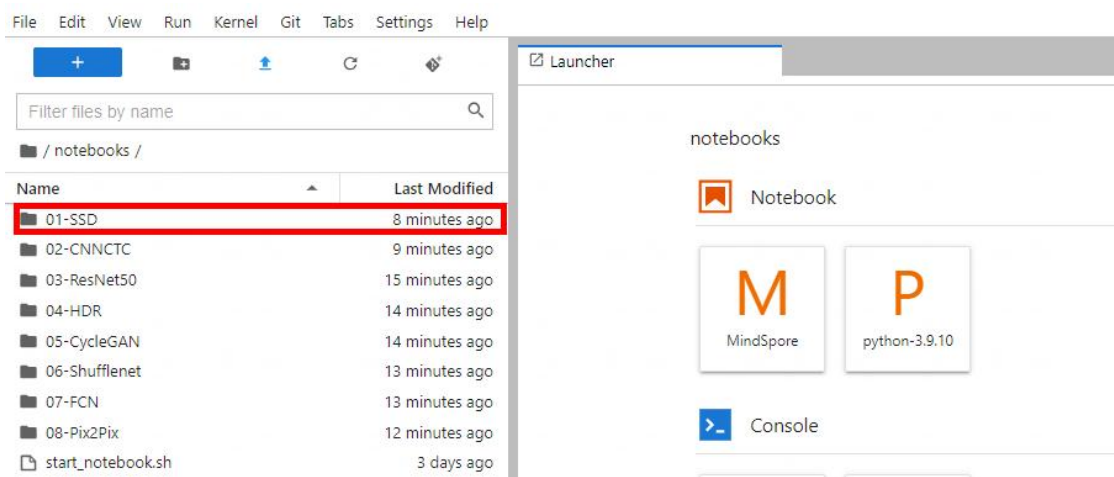
SSD is a single-stage object detection algorithm that extracts features through convolutional neural networks. SSD extracts different feature layers for detection output, so SSD is a multi-scale detection method. The dataset used in this example is COCO 2017. In order to save and load data more conveniently, the COCO dataset is first converted to MindRecord format before data reading in this example.

To run this sample smoothly, at least 12GB of Swap memory needs to be set on the development board. Please refer to the instructions in the section on [how to set Swap memory](#).

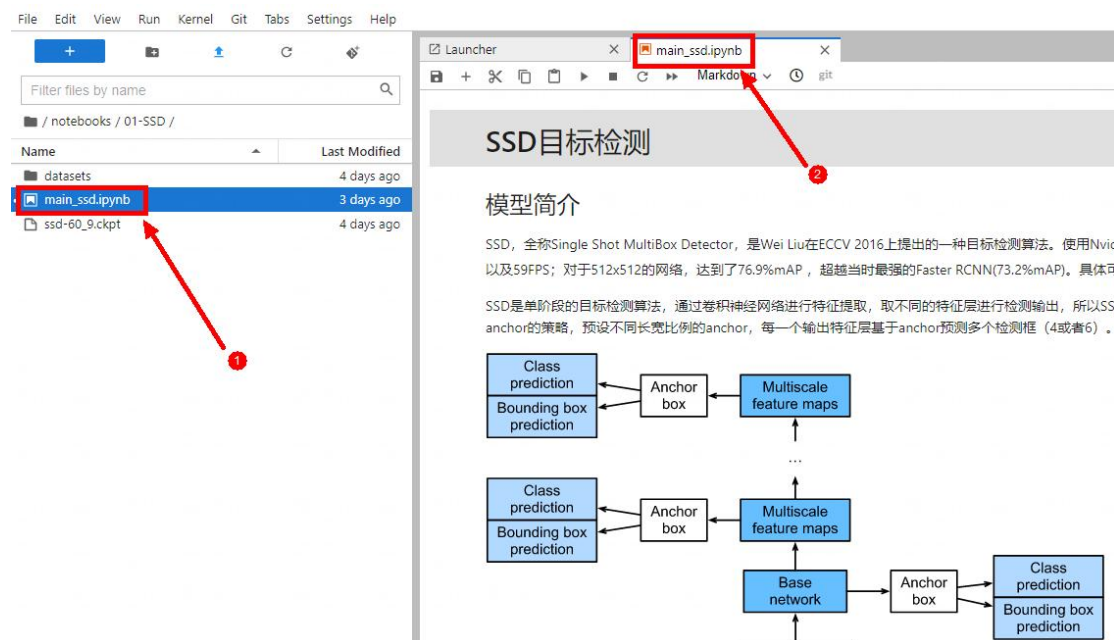
1) Firstly, double-click on the **01-SSD** shown in the following image on the Jupyter Lab



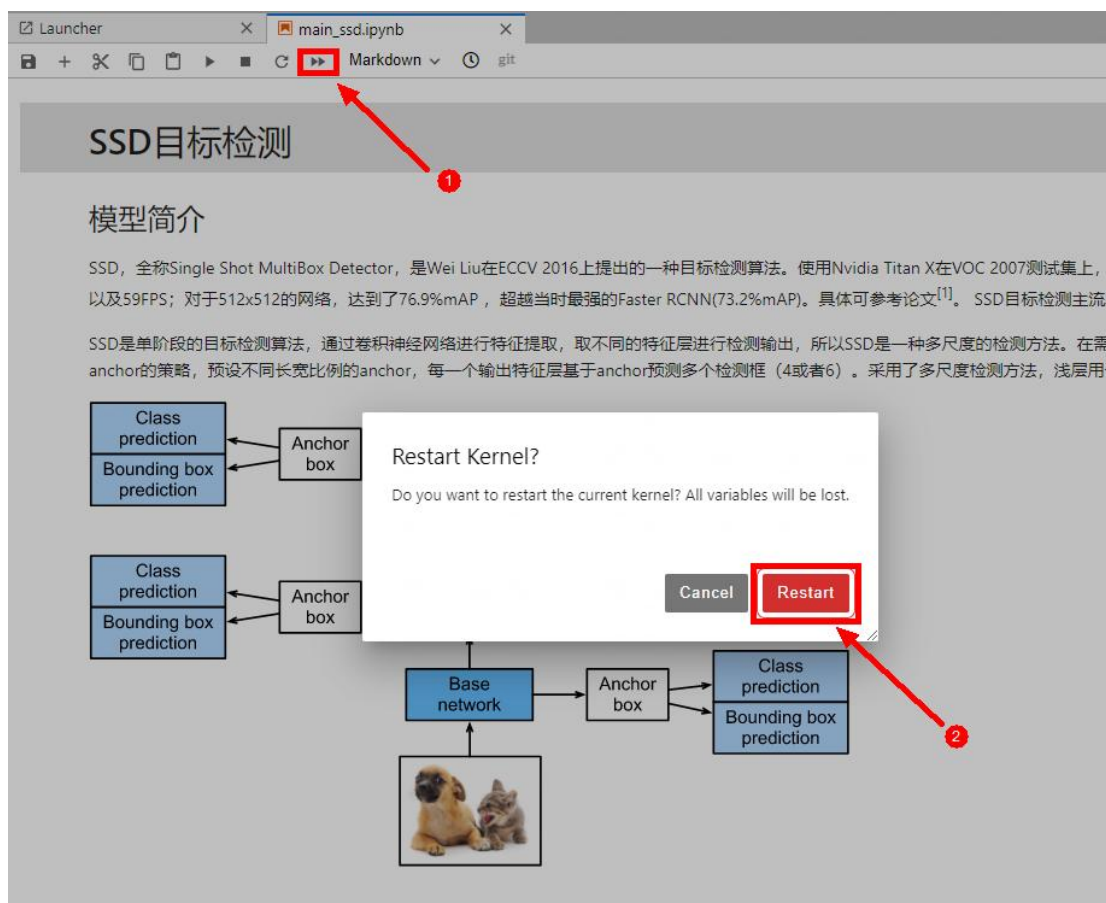
interface to enter the directory of the sample.



2) There are all the resources to run the example in this directory, where **main_ssd.ipynb** is the file to run the example in Jupyter Lab. Double click to open **main_ssd.ipynb**, and the contents of the file will be displayed in the window on the right, as shown in the following figure:



3) Click the **▶▶** button to run the sample, and then click the **Restart** button in the pop-up dialog box.



4) After the program is executed, the results of object detection will be displayed in the notebook document. As shown in the following figure:

```
def eval_net():
    print("Start Eval!")
    ssd_eval(mindrecord_file, "./ssd-60_9.ckpt", anno_json)

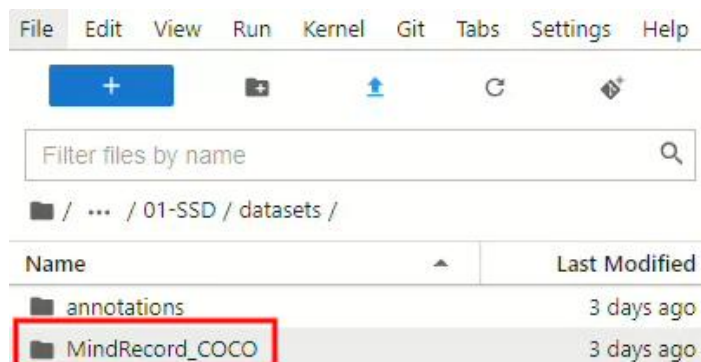
eval_net()
```

```
=====
Accumulating evaluation results...
DONE (t=1.24s).
Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.001
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=100 ] = 0.002
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=100 ] = 0.000
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.039
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.004
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.001
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.020
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.054
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.057
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.283
=====
mAP: 0.0006232523571113707
```

5) The path for saving test data is as follows:



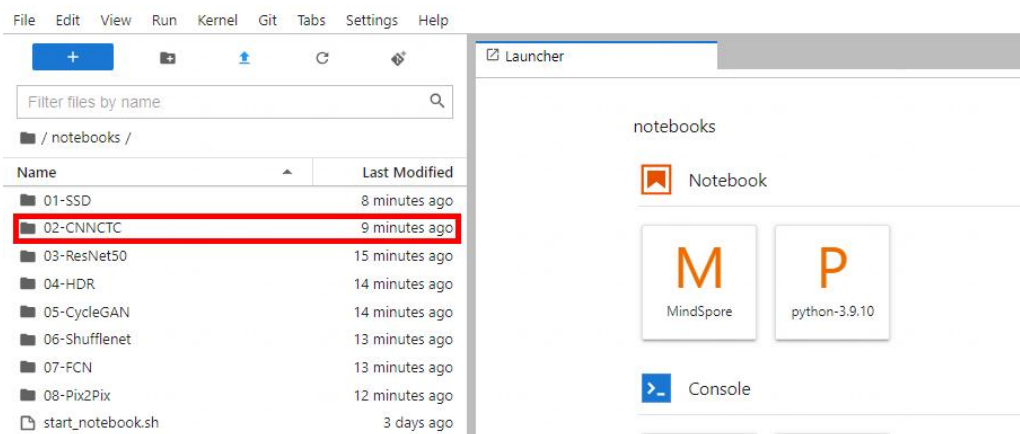
/home/HwHiAiUser/samples/01-SSD /datasets/



4. 4. Running Text Recognition Sample

Text recognition refers to identifying text from an image, converting the text regions into character information, typically using CNN networks to extract rich feature information from the image, and then recognizing based on the extracted feature information. ResNet is used as the feature extraction network here, and CTC method is employed for recognition. The converted om model and test images are already included in the sample. You can run the sample in Jupyter Lab according to the following process.

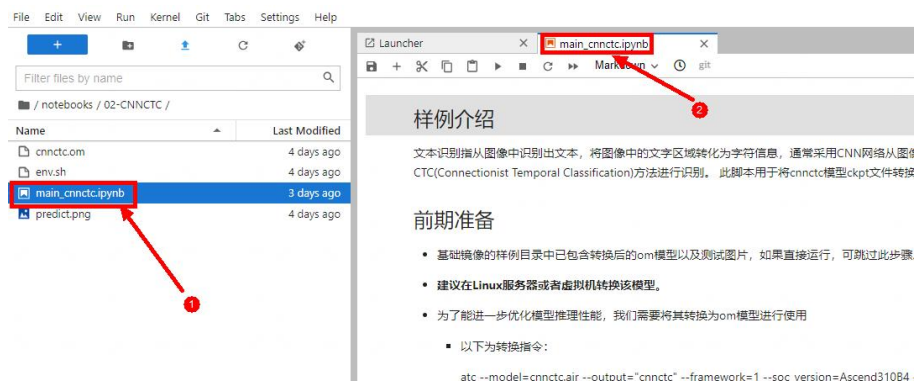
1) Firstly, double-click on **02-CNNCTC** as shown in the following image on the Jupyter Lab interface to enter the directory of the sample.。



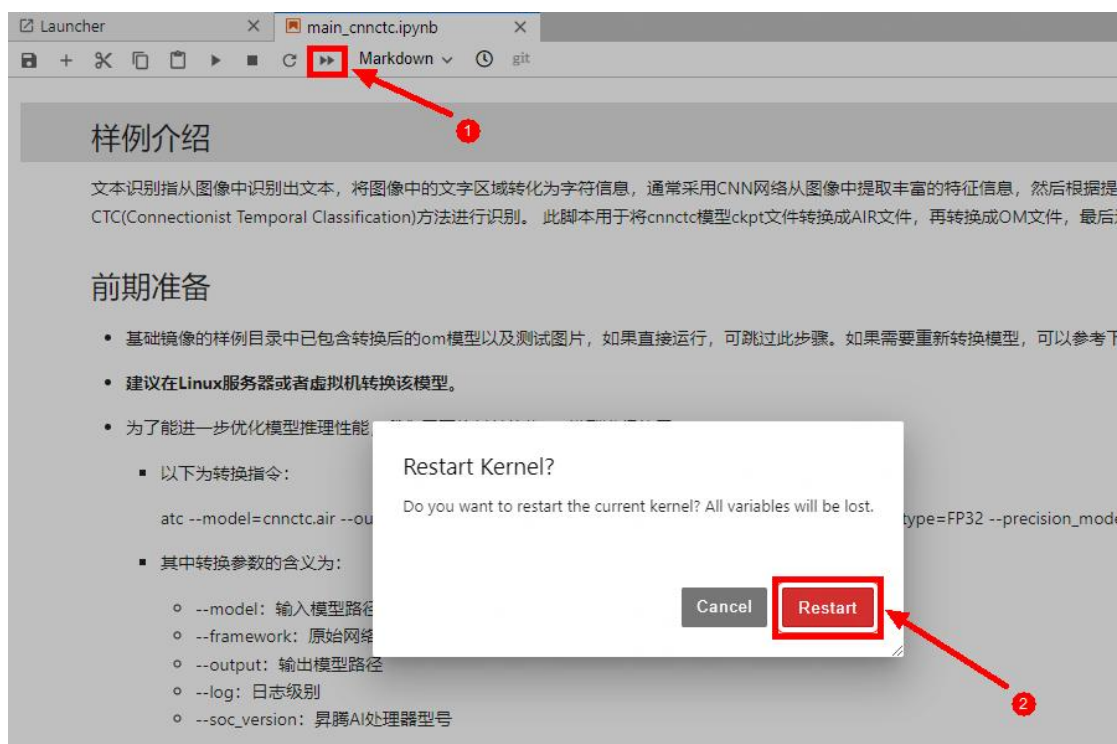
2) Then there are all the resources to run the example in that directory, where **main_cnnctc.ipynb** is the file to run the example in Jupyter Lab. Double click to open **main_cnnctc.ipynb**, and the contents of this file will be displayed in the window on the



right, as shown in the following figure:



3) Click the  button to run this sample, and then click the **Restart** button in the pop-up dialog box.



4) After the program is executed, the result of text recognition in the image can be seen in the notebook document. As shown in the following figure:



```

img = np.array(img, dtype=np.float32)
img = np.expand_dims(img, axis=0)
img = np.transpose(img, [0, 3, 1, 2])

# 定义推理的时间
start = time.time()
model_predict = model.execute([img])[0]
end = time.time()
print(f'infer use time:{(end-start)*1000}ms')

# 初始化文本编码函数
character = '0123456789abcdefghijklmnopqrstuvwxyz'
converter = CTCLabelConverter(character)

# 推理过程
preds_size = np.array([model_predict.shape[1]])
preds_index = np.argmax(model_predict, 2)
preds_index = np.reshape(preds_index, [-1])
preds_str = converter.decode(preds_index, preds_size)
print('Predict: ', preds_str)

infer use time:8.622884750366211ms
Predict: ['parking']

```

5) The save path for test images is as follows:

/home/HwHiAiUser/samples/02-CNNCTC/predict.png

/ notebooks / 02-CNNCTC /

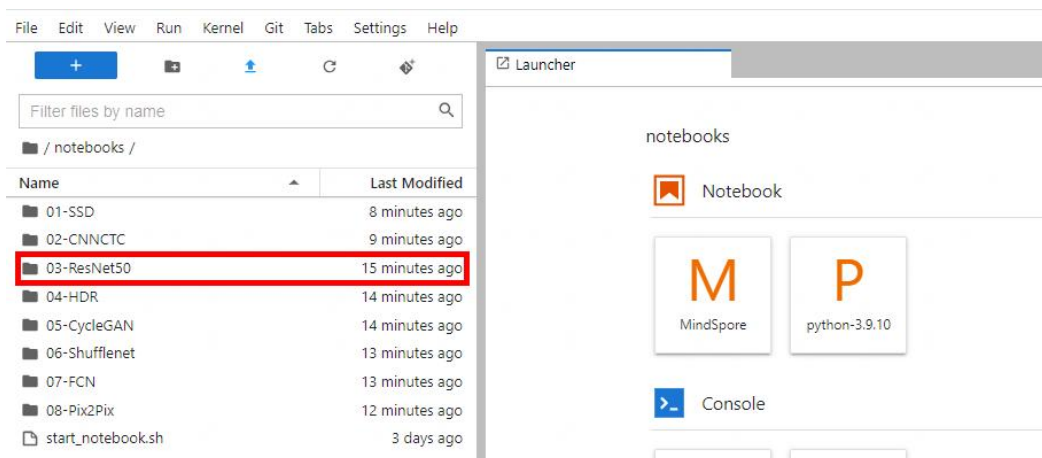
Name	Last Modified
cnnectc.om	4 days ago
env.sh	4 days ago
main_cnnctc.ipynb	3 days ago
predict.png	4 days ago

4. 5. Running Target Classification Example

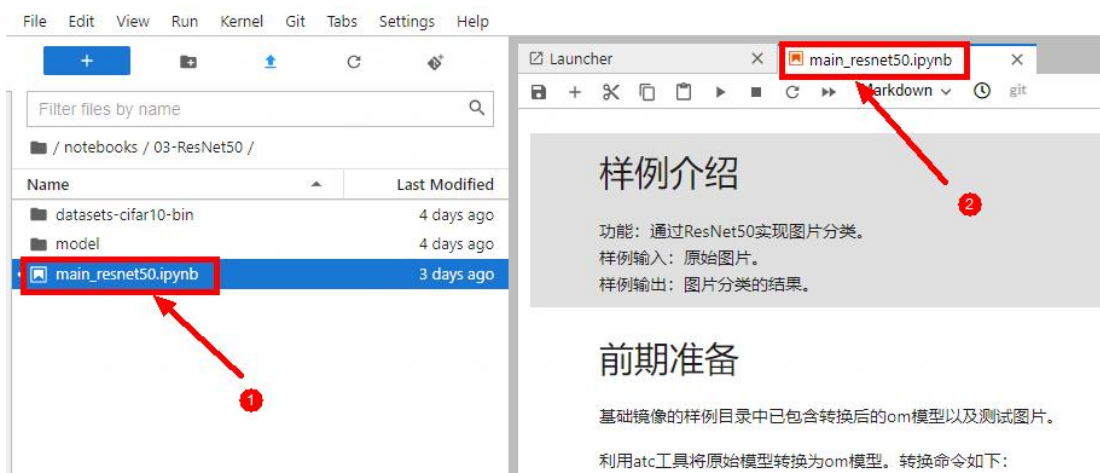
Image classification is the most fundamental computer vision application, belonging to the category of supervised learning, such as determining the category of an image given (cat, dog, airplane, car, etc.). ResNet proposed the Residual Network structure to alleviate the degradation problem, and using ResNet network can achieve the construction of deeper network structures (breaking through 1000 layers). This example uses ResNet50 network to perform offline inference classification on the CIFAR-10 dataset; The converted OM file and notebook document for inference have been provided in the sample. The running of the sample can be completed through the following steps:



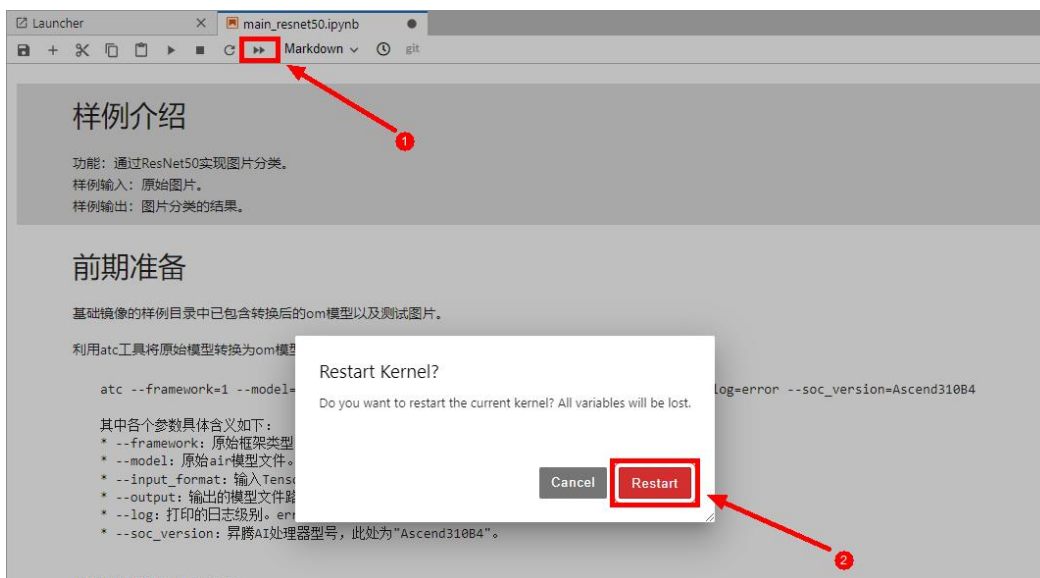
1) Firstly, double-click on **03-ResNet50** as shown in the following image in the Jupyter Lab interface to enter the directory of the sample.



2) There are all the resources in this directory for running the example, where **main_resnet50.ipynb** is the file for running the example in Jupyter Lab. Double click to open **main_resnet50.ipynb**, and the contents of this file will be displayed in the window on the right, as shown in the following figure:



3) Click the **▶▶** button to run the sample, and then click the **Restart** button in the pop-up dialog box.



4) After the program is executed, the image classification results can be successfully obtained in the notebook document. As shown in the following figure:

```

Init model resource success...
[AcLiteModel] create model output dataset:
malloc output 0, size 160
Create model output dataset success
Init model resource success
=====
crop_and_paste_image: <mindspore.dataset.engine.datasets.BatchDataset object at 0xe7ff3e2cd9d0>
resized_image['label']: [3 8 8 0]
pred: [3 8 8 0]

```

predict:cat



predict:ship



predict:ship



predict:airplane



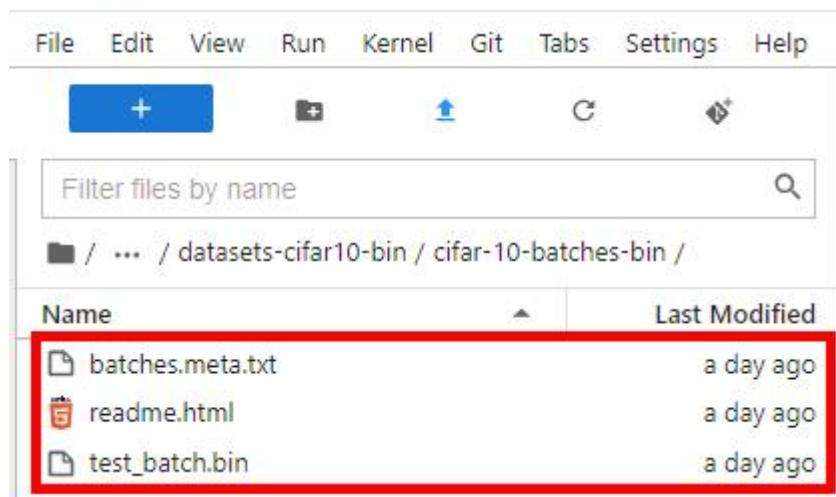
```

acl resource release all resource
dvpp resource release success
AcLiteModel release source success

```

5) The path for saving test data is as follows:

/home/HwHiAiUser/samples/03-ResNet50/datasets-cifar10-bin/cifar-10-batches-bin/

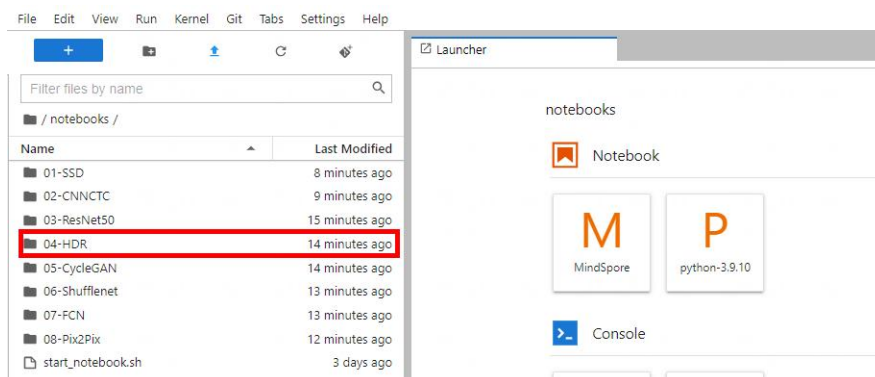


4. 6. Running Image Exposure Enhancement Sample

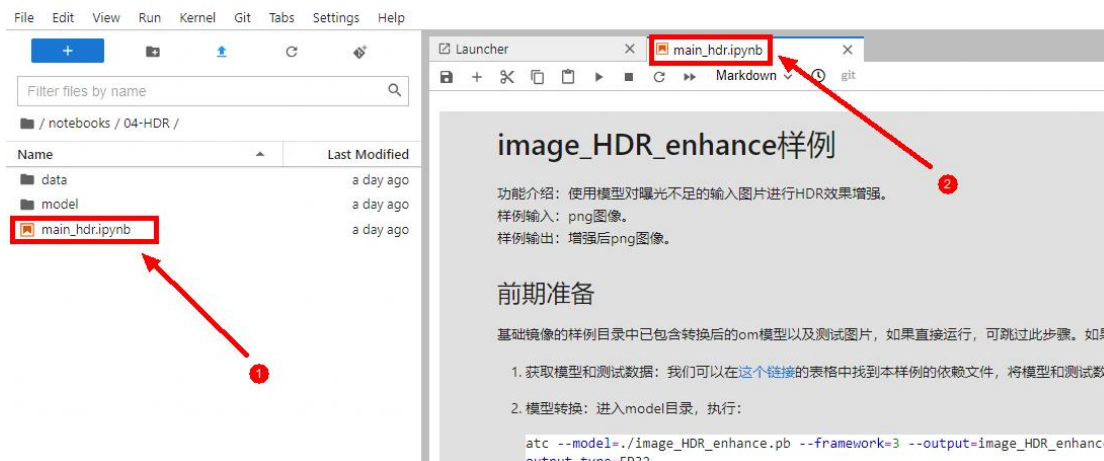
This example can enhance HDR effects on input images with insufficient exposure. The converted om model and test images are already included in the sample. You can run the sample in Jupyter Lab according to the following process.

Note that this example was not implemented using MindSpore. In addition, all other examples in this chapter are implemented using MindSpore.

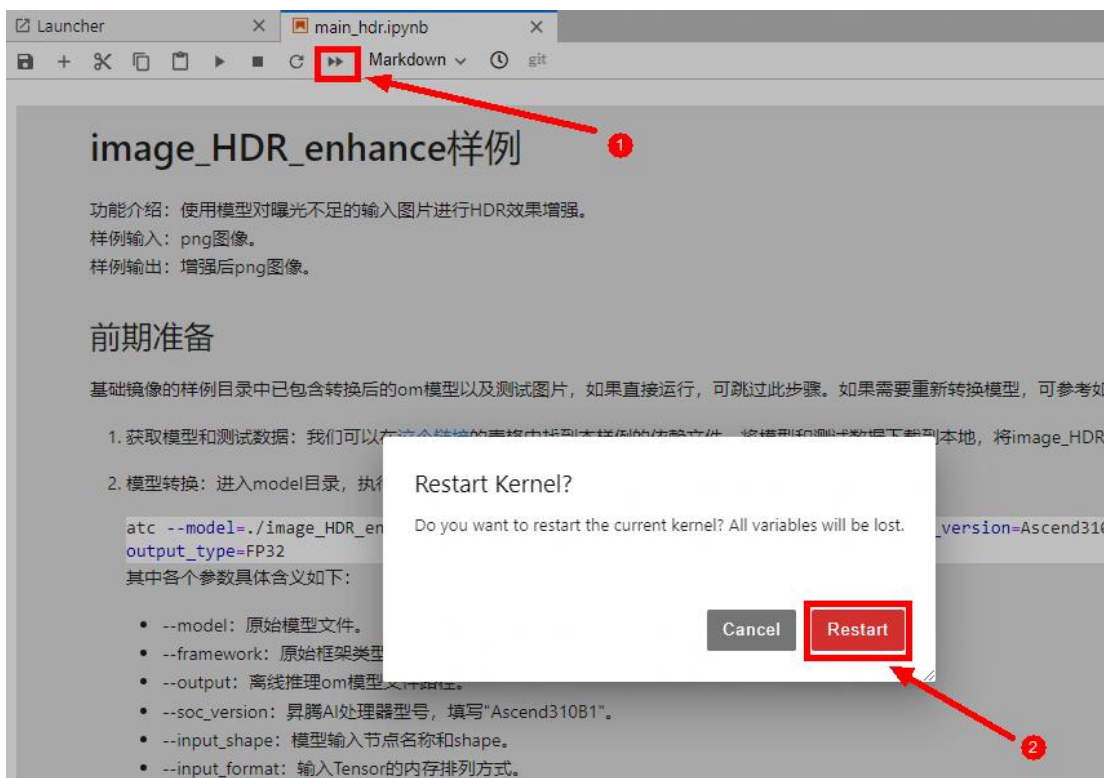
1) Firstly, double-click on **04-HDR** as shown in the following image on the Jupyter Lab interface to enter the directory of the sample.



2) In this directory, there are all the resources to run the example, where **main_hdr.ipynb** is the file to run the example in Jupyter Lab. Double click to open **main_hdr.ipynb**, and the contents of this file will be displayed in the window on the right.



3) Click the **Run** button to run this sample, and then click the **Restart** button in the pop-up dialog box.



4) After a few seconds, two images appeared in the window. The left side was the original appearance of the image, and the right side was the image processed by the model. We can see that the model has enhanced the exposure of the image, improving its visual appeal.



5) The save path for test images is as follows:

`/home/HwHiAiUser/samples/04-HDR/data/data1.png`



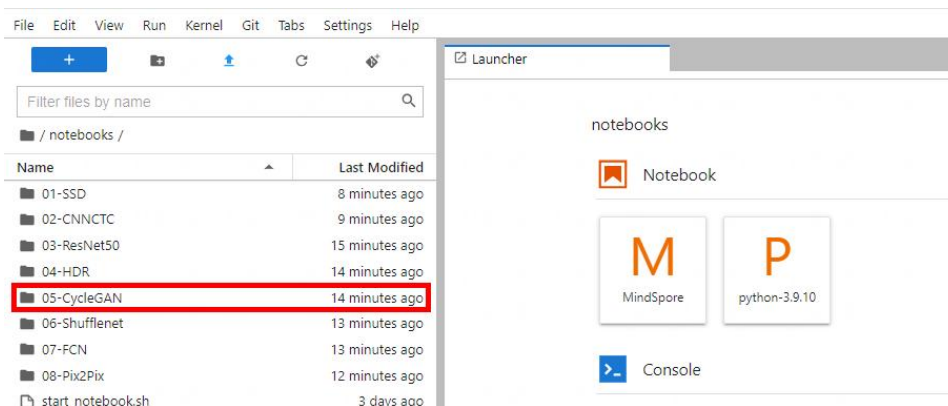
4. 7. Running Image Style Migration Sample

CycleGAN (Cycle Generative Adversarial Network) refers to a cyclic adversarial generative network. One important application area of this model is domain adaptation, which can be easily understood as image style transfer. And it only requires data from two domains without strict correspondence between them, making it a new unsupervised image transfer network.

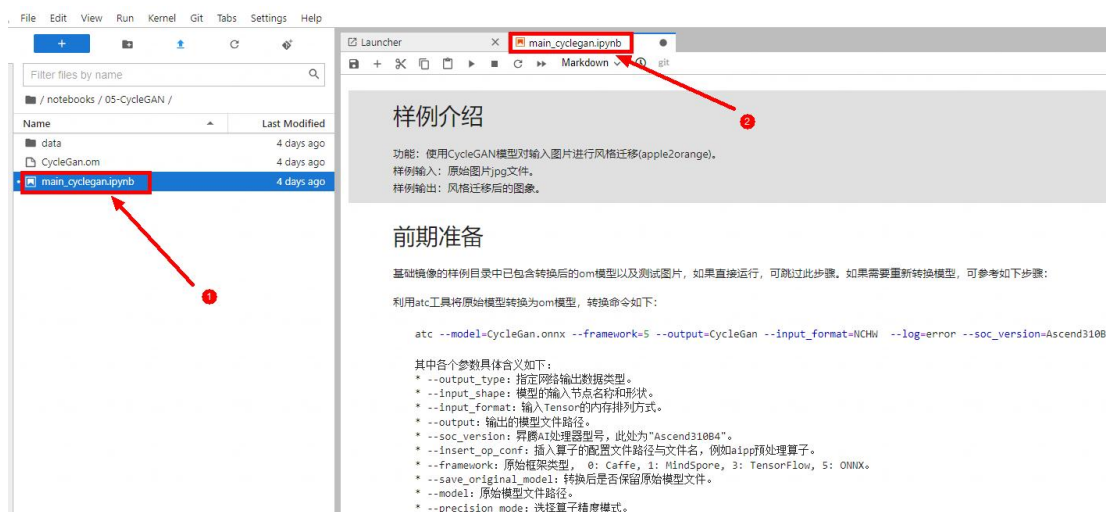
This example uses the CycleGAN model to transfer apples to orange style. The converted om model and test images are already included in the sample. You can run the sample in Jupyter Lab according to the following process.

To run this sample smoothly, the development board also needs to set up 12GB of Swap memory. Please refer to the instructions in the section on [how to set up Swap memory](#).

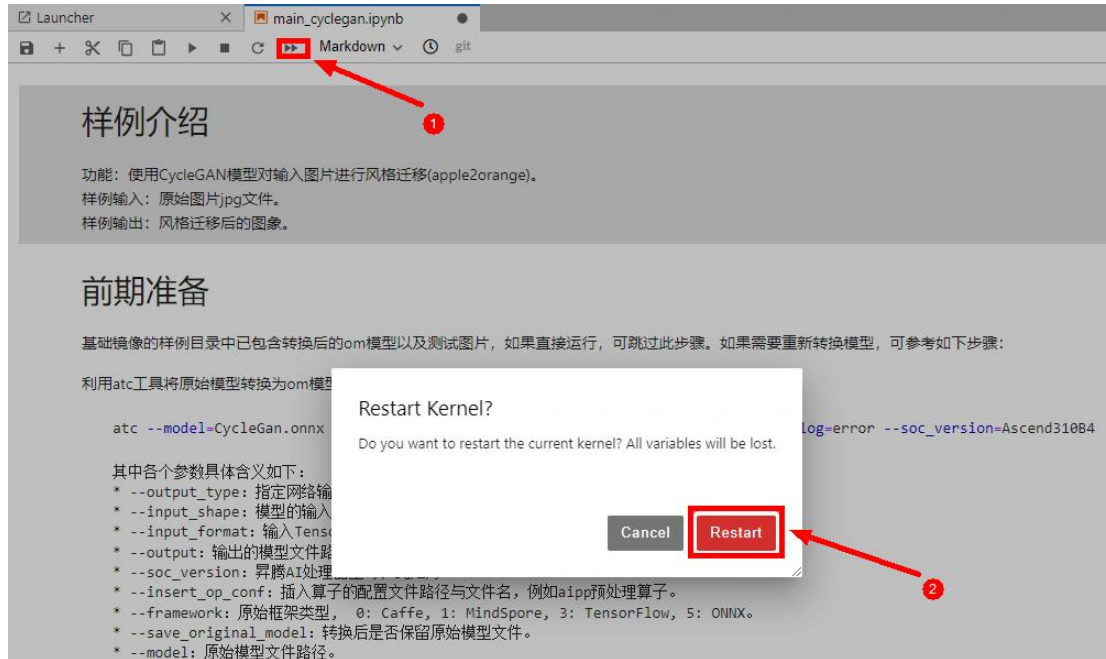
1) Firstly, double-click on the **05-CycleGAN** shown in the following image on the Jupyter Lab interface to enter the directory of the sample.



2) There are all resources in this directory for running the example, where **main_cyclegan.ipynb** is the file for running the example in Jupyter Lab. Double click to open **main_cyclegan.ipynb**, and the contents of this file will be displayed in the window on the right.



3) Click the  button to run this sample, and then click the **Restart** button in the pop-up dialog box.



4) After a few seconds, a picture of an apple and its image in orange style will appear in the window.



5) The save path for test images is as follows:

/home/HwHiAiUser/samples/05-CycleGAN /data/img.jpg

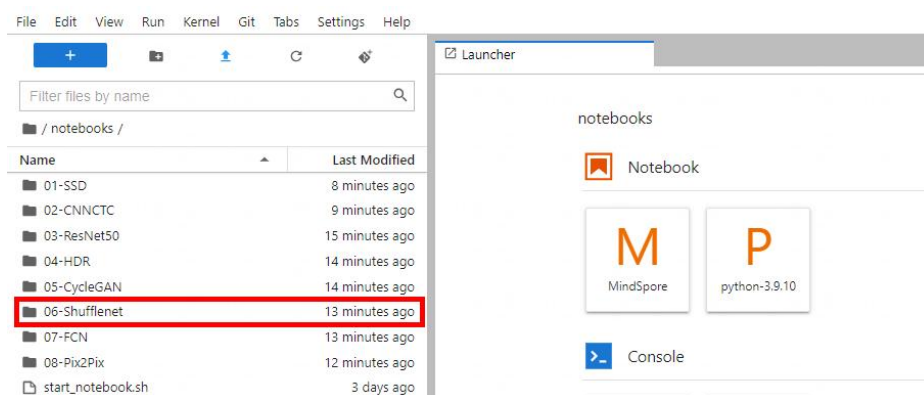




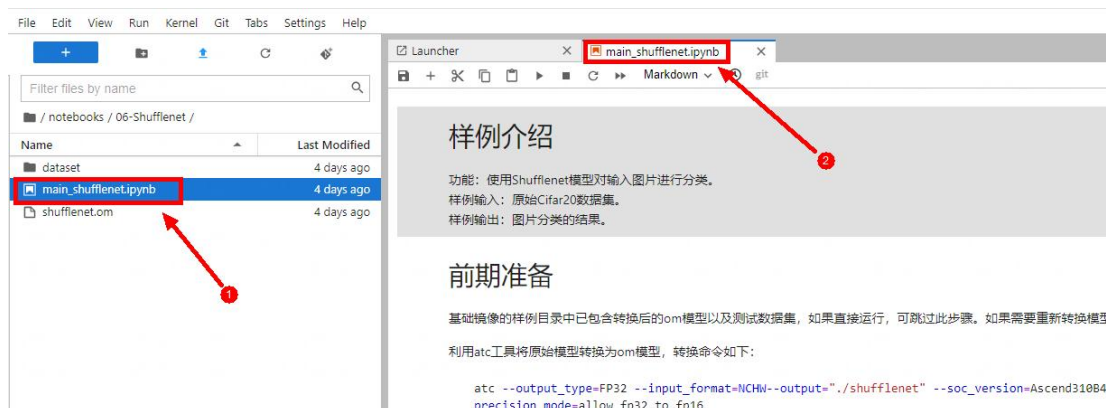
4. 8. Running Image Classification Sample

ShuffleNetV1 is a computationally efficient CNN model proposed by Megvii Technology. Like MobileNet, SqueezeNet, and other models, it is mainly used on mobile devices. Therefore, the design goal of the model is to use limited computing resources to achieve the best model accuracy. The design core of ShuffleNetV1 is the introduction of two operations: Pointwise Group Convolution and Channel Shuffle, which greatly reduces the computational complexity of the model while maintaining accuracy. Therefore, ShuffleNetV1, similar to MobileNet, achieves model compression and acceleration by designing more efficient network structures.

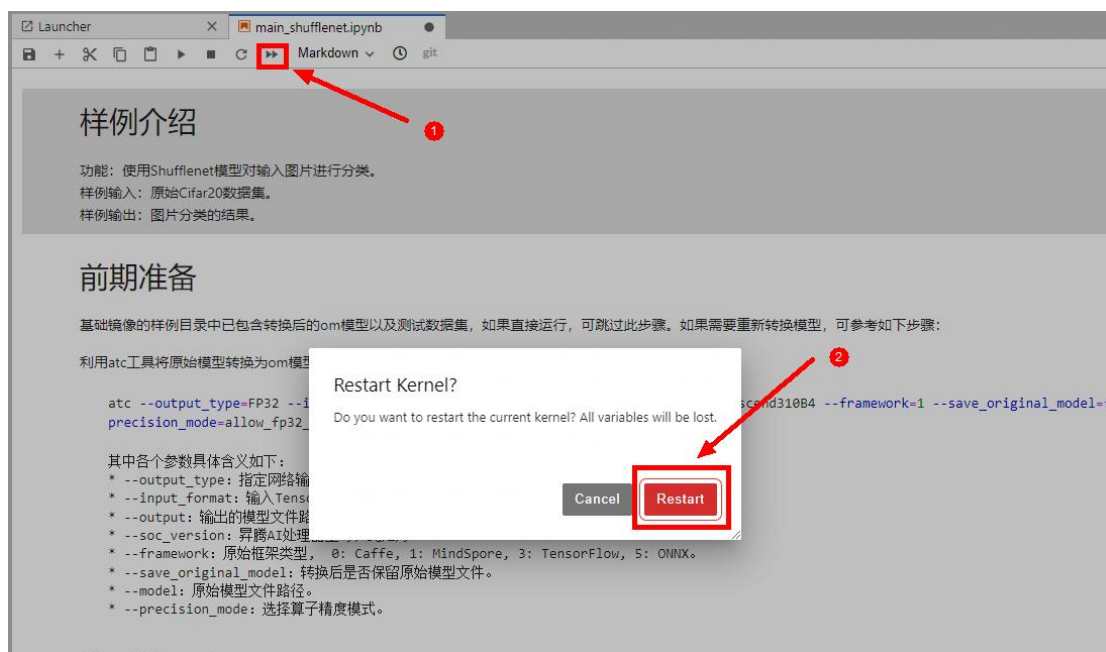
1) Firstly, double-click on the **06-Shufflenet** shown in the following image on the Jupyter Lab interface to enter the directory of the sample.



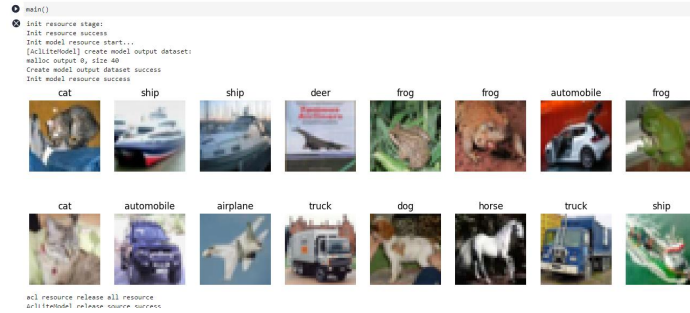
2) In this directory, there are all the resources to run the example, where **main_shufflenet.ipynb** is the file to run the example in Jupyter Lab. Double click to open **main_shufflenet.ipynb**, and the contents of this file will be displayed in the window on the right, as shown in the following figure:



3) Click the  button to run this sample, and then click the **Restart** button in the pop-up dialog box.

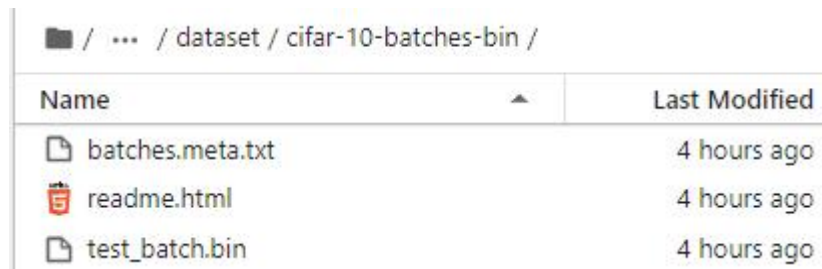


4) After the program is executed, the image classification results can be successfully displayed in the notebook document. As shown in the following figure:



5) The path for saving test data is as follows:

`/home/HwHiAiUser/samples/06-Shufflenet /dataset/cifar-10-batches-bin/`



4. 9. Running FCN Image Semantic Segmentation Sample

Semantic segmentation of images is an important part of image processing and machine vision technology related to image understanding. It is an important branch in the field of AI and is often applied in areas such as facial recognition, object detection, medical imaging, satellite image analysis, and autonomous driving perception. The purpose of semantic segmentation is to classify each pixel in an image. Unlike ordinary classification tasks that only output a certain category, semantic segmentation tasks output images of the same size as the input, with each pixel in the output image corresponding to the category of each pixel in the input image. Semantics in the field of images refers to the content of an image and the understanding of its meaning.

FCN (Fully Convolutional Networks) is the first fully convolutional network to perform pixel level prediction end-to-end. Directly obtain a label map of the same size as the original image through pixel level prediction. Because FCN discards fully connected layers and replaces them with fully convolutional layers, all layers in the network are convolutional layers, hence it is called a fully convolutional network.

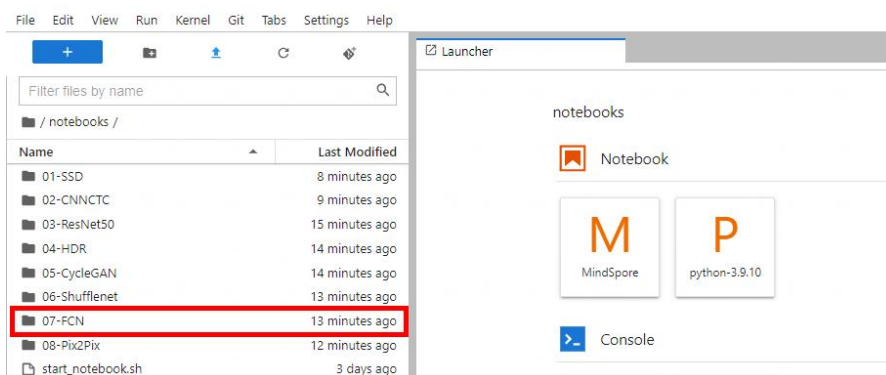
FCN network features:



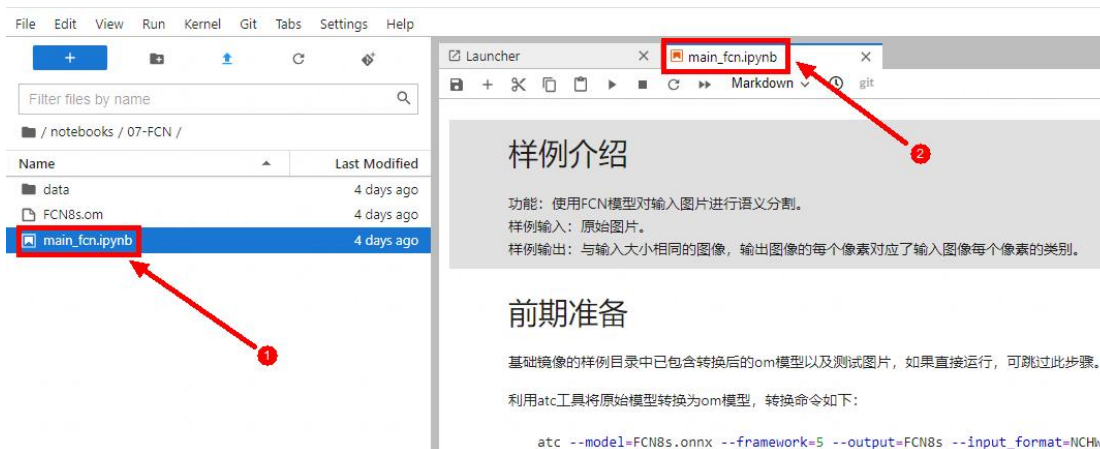
1. A fully convolutional (fully conv) network without a fully connected (fc) layer can adapt to inputs of any size.
2. The deconvolution (deconv) layer that increases the data size can output fine results.
3. A skip-level structure that integrates results from different depth layers, ensuring both robustness and precision.

This example can segment people and horses in images. The converted om model and test data are already included in the sample. You can run the sample in Jupyter Lab according to the following process.

1) Firstly, double-click on **07-FCN** as shown in the following image on the Jupyter Lab interface to enter the directory of the sample.

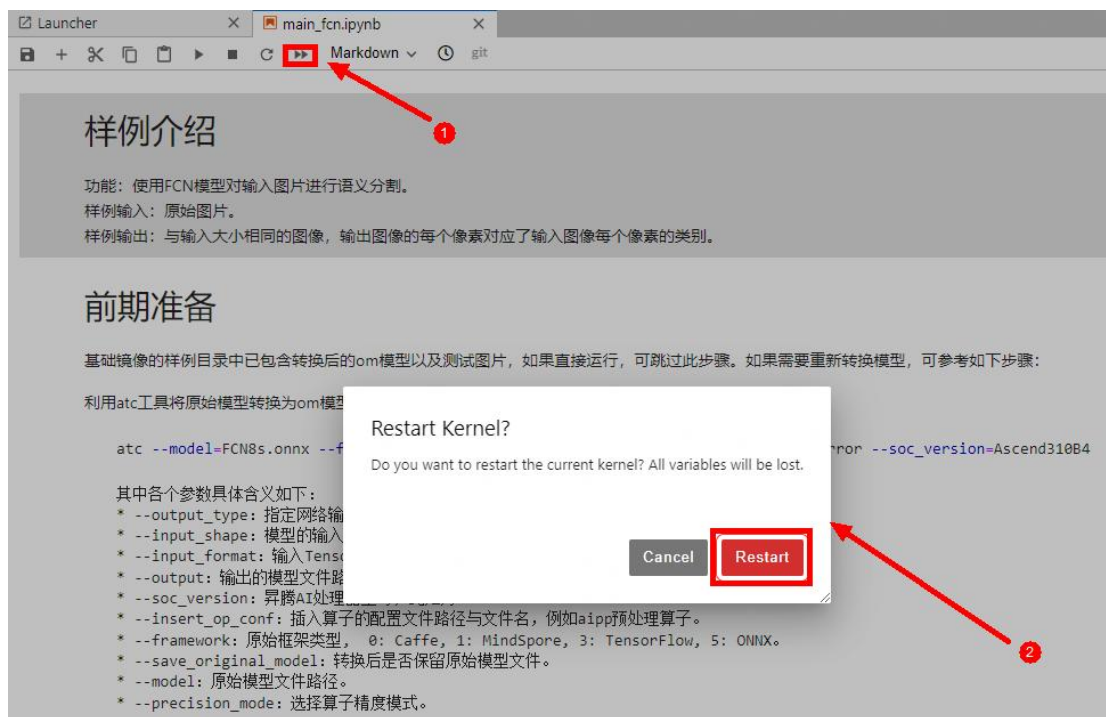


2) There are all resources in this directory for running the example, where **main_fcn.ipynb** is the file for running the example in Jupyter Lab. Double click to open **main_fcn.ipynb**, and the contents of this file will be displayed in the window on the right, as shown in the following figure:





3) Click the button to run this sample, and then click the **Restart** button in the pop-up dialog box.



4) After the program is executed, the results of image semantic segmentation can be successfully displayed in the notebook document. As shown in the following figure:





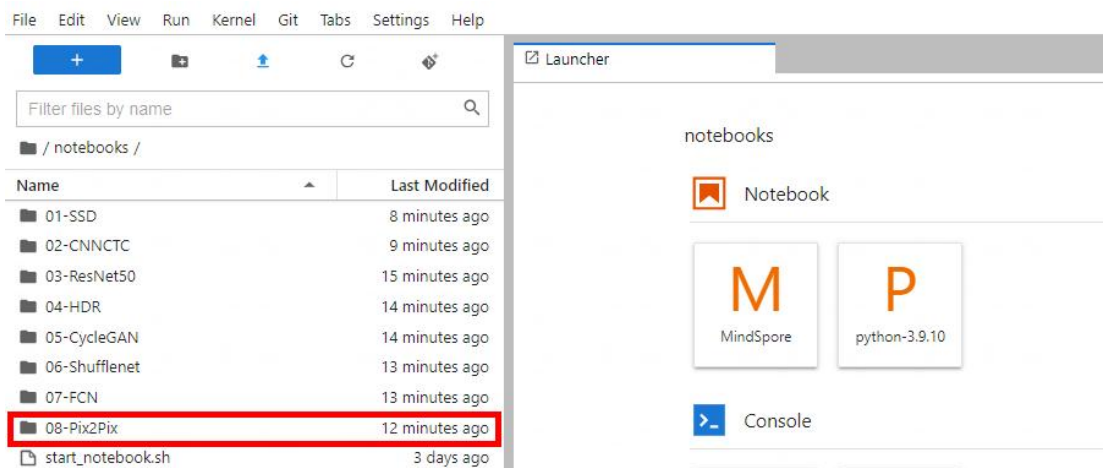
4. 10. Running Implementation Image Conversion Sample

Pix2Pix is a deep learning image transformation model based on Conditional Generative Adversarial Networks (cGAN). The model was proposed by Phillip Isola et al. at the 2017 CVPR, and can achieve the transformation of semantics/labels to real images, grayscale images to color images, aerial images to maps, day to night, and line draft images to physical images. Pix2Pix is a classic work that applies cGAN to supervised image to image translation, consisting of two models: a generator and a discriminator.

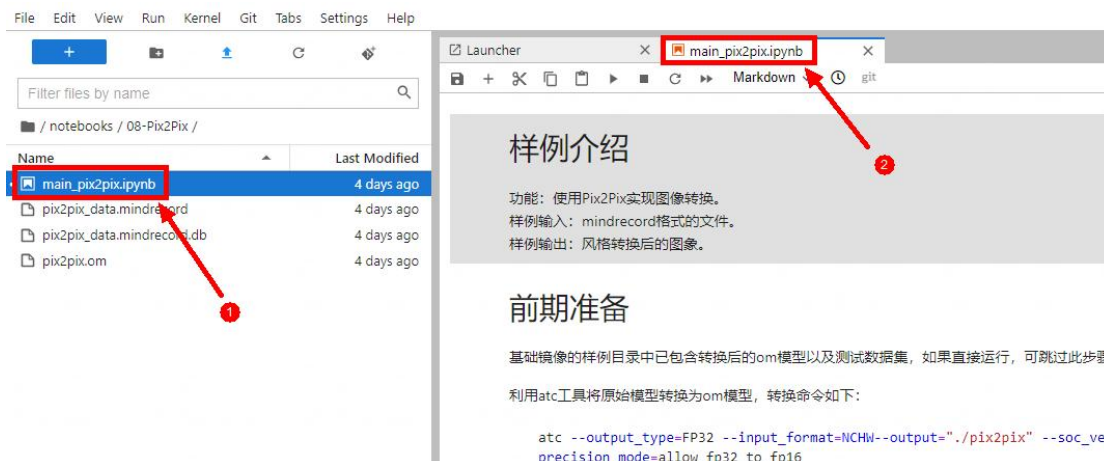
Traditionally, although the goal of such tasks is to predict pixels from pixels, each item is processed by a separate dedicated machine. The network used by Pix2Pix serves as a universal framework, utilizing the same architecture and objectives, and only training on different data to achieve satisfactory results. Given this, many people have already used this network to publish their own art works.

We can run the sample in Jupyter Lab according to the following process.

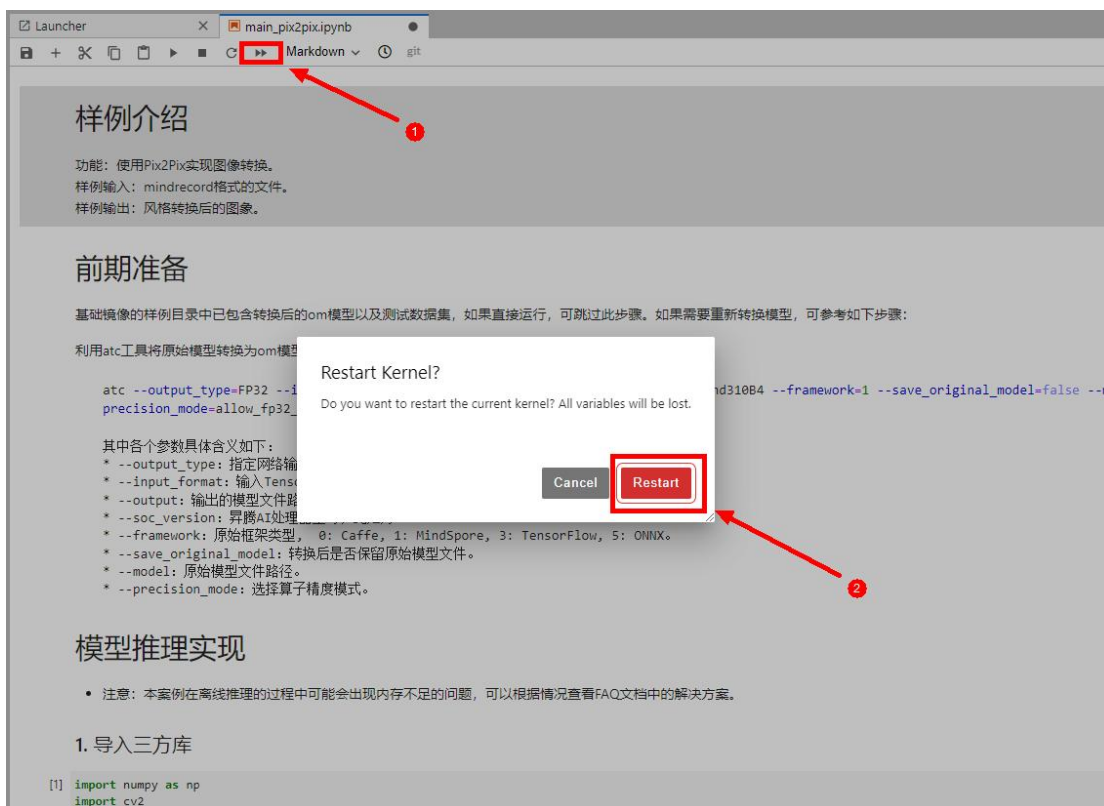
1) Firstly, double-click on **08-Pix2Pix** in the Jupyter Lab interface as shown in the figure below to enter the directory of the sample.



2) There are all the resources in this directory for running the example, where **main_pix2pix.ipynb** is the file for running the example in Jupyter Lab. Double click to open **main_pix2pix.ipynb**, and the contents of this file will be displayed in the window on the right, as shown in the following figure:



3) Click the button to run this sample, and then click the **Restart** button in the pop-up dialog box.



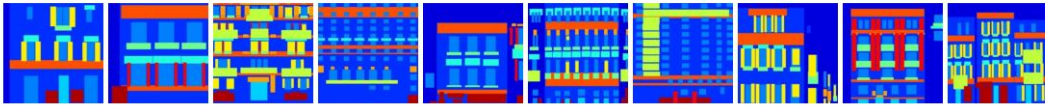
4) After the program is executed, the result of image conversion can be successfully displayed in the notebook document. As shown in the following figure:



5. 运行

运行完成后，会显示推理后的图片，如下所示。

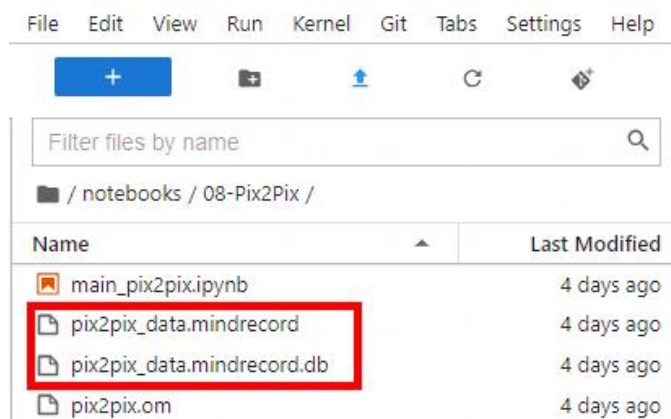
```
main()
init resource stage:
Init resource success
Init model resource start...
[AclliteModel] create model output dataset:
malloc output 0, size 12582912
Create model output dataset success
Init model resource success
```



```
acl resource release all resource
AclliteModel release source success
acl resource release stream
acl resource release context
Reset acl device 0
Release acl resource success
```

5) The path for saving test data is as follows:

/home/HwHiAiUser/samples/08-Pix2Pix/





5. Experience AI application examples (Ubuntu)

We have pre installed Jupyter Lab software and version 2.4 of the MindSpore framework in the image. Jupyter Lab software is a web-based interactive development environment that integrates features such as a code editor, terminal, and file manager, allowing developers to complete various tasks in one interface. And we have also pre-set some AI application samples based on the MindSpore framework developed by Shengsi that can run in Jupyter Lab software in the image. These examples are all written in Python and call the Python version of the AscendCL programming interface. This chapter introduces how to log in to Jupyter Lab and run these pre installed AI application samples in Jupyter Lab.

For guidance on running online cases on the Orange Pie development board, you can also refer to the [Orange Pi development chapter](#) on the MindSpore official website.

5.1. Login to jupyter Lab

1) Firstly, log in to the Linux system desktop, then open the terminal and switch to the directory where the MindSpore AI application sample is saved.

```
(base) HwHiAiUser@orangepiaipro-20t:~$ cd orange-pi-mindspore
```

2) There are 2 folders, 1 shell file, 1 documentation, and 1 license file in the current directory. Among them, 7 offline model cases are stored in the "**Offline**" folder, and 15 online inference cases are stored in the "**Online**" folder. There is a Jupyter Lab startup script **start_notebook.sh** in the directory.

```
(base) HwHiAiUser@orangepiaipro-20t:~/orange-pi-mindspore$ tree -L 2
|__ LICENSE
|__ Offline
|   |__ 01-CNNCTC
|   |__ 02-ResNet50
|   |__ 03-HDR
|   |__ 04-CycleGAN
|   |__ 05-Shufflenet
|   |__ 06-FCN
|   |__ 07-Pix2Pix
|   |__ README.md
```



```
| |— __init__.py
| |— 基于昇思 MindSpore+Orangepi AIpro 的训推全流程指导书(离线推理)
|— Online
| |— 01-quick start
| |— 02-ResNet50
| |— 03-ViT
| |— 04-FCN
| |— 05-ShuffleNet
| |— 06-SSD
| |— 07-RNN
| |— 08-LSTM+CRF
| |— 09-GAN
| |— 10-DCGAN
| |— 11-Pix2Pix
| |— 12-Diffusion
| |— 13-ResNet50_transfer
| |— 14-qwen1.5-0.5b
| |— 15-tinyllama
| |— README.md
| |— __init__.py
| |— images
|— README.md
|— start_notebook.sh
```

3) Then execute the start_notebook.sh script to start Jupyter Lab. If you need to use a computer browser for the experience, please add the ip address of the development board after the script, such as "**./start_notebook.sh 192.168.1.2**".

```
(base) HwHiAiUser@orangepiaipro-20t:~/~/orange-pi-mindspore$ ./start_notebook.sh
```

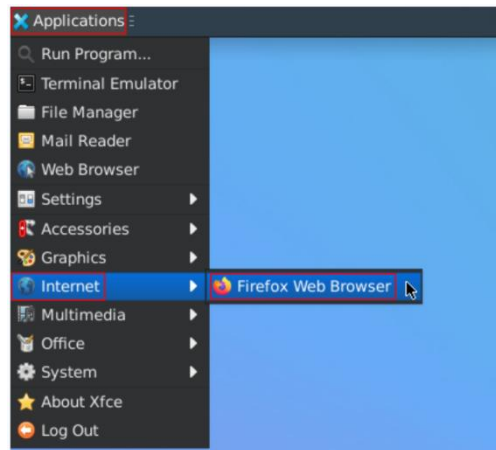
4) After executing the script, the terminal will display the following print message, which will include a link to log in to Jupyter Lab.



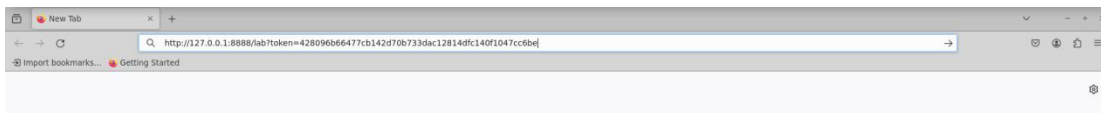
```
(base) HwHiAiUser@orangepiaipr:~/samples$ ./start_notebook.sh
[W 2024-06-03 17:35:23.674 ServerApp] A `_jupyter_server_extension_points' function was not found in notebook shim. Instead, a `_jupyter_server_extension_path' function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2024-06-03 17:35:23.675 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2024-06-03 17:35:23.687 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2024-06-03 17:35:23.702 ServerApp] jupyterlab | extension was successfully linked.
[I 2024-06-03 17:35:23.705 ServerApp] Writing Jupyter server cookie secret to /home/HwHiAiUser/.local/share/jupyter/runtime/jupyter_cookie_secret
[I 2024-06-03 17:35:24.390 ServerApp] notebook_shim | extension was successfully linked.
[I 2024-06-03 17:35:24.558 ServerApp] notebook_shim | extension was successfully loaded.
[I 2024-06-03 17:35:24.564 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-06-03 17:35:24.567 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2024-06-03 17:35:24.573 LabApp] JupyterLab extension loaded from /usr/local/miniconda3/lib/python3.9/site-packages/jupyterlab
[I 2024-06-03 17:35:24.575 LabApp] JupyterLab application directory is /usr/local/miniconda3/share/jupyter/lab
[I 2024-06-03 17:35:24.582 ServerApp] jupyterlab | extension was successfully loaded.
[I 2024-06-03 17:35:24.583 ServerApp] Serving notebooks from local directory: /home/HwHiAiUser/samples
[I 2024-06-03 17:35:24.583 ServerApp] Jupyter Server 2.12.5 is running at:
[I 2024-06-03 17:35:24.584 ServerApp] http://127.0.0.1:8888/lab?token=7c876b0dc78ff8853bd93bdf1e42a6454cf685e2c0c6d141
[I 2024-06-03 17:35:24.584 ServerApp] http://127.0.0.1:8888/lab?token=7c876b0dc78ff8853bd93bdf1e42a6454cf685e2c0c6d141
[I 2024-06-03 17:35:24.584 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

To access the server, open this file in a browser:
file:///home/HwHiAiUser/.local/share/jupyter/runtime/jpserver-4434-open.html
Or copy and paste one of these URLs:
http://127.0.0.1:8888/lab?token=7c876b0dc78ff8853bd93bdf1e42a6454cf685e2c0c6d141
http://127.0.0.1:8888/lab?token=7c876b0dc78ff8853bd93bdf1e42a6454cf685e2c0c6d141
[I 2024-06-03 17:35:24.659 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-server-nodejs, javascript-typescript-langserv
er, jedi-language-server, julia-language-server, pyright, python-language-server, python-lsp-server, r-language-server, sql-language-server, texlab, typescript
-language-server, unified-language-server, vscode-css-languageserver-bin, vscode-html-languageserver-bin, vscode-json-languageserver-bin, yamll-language-server
```

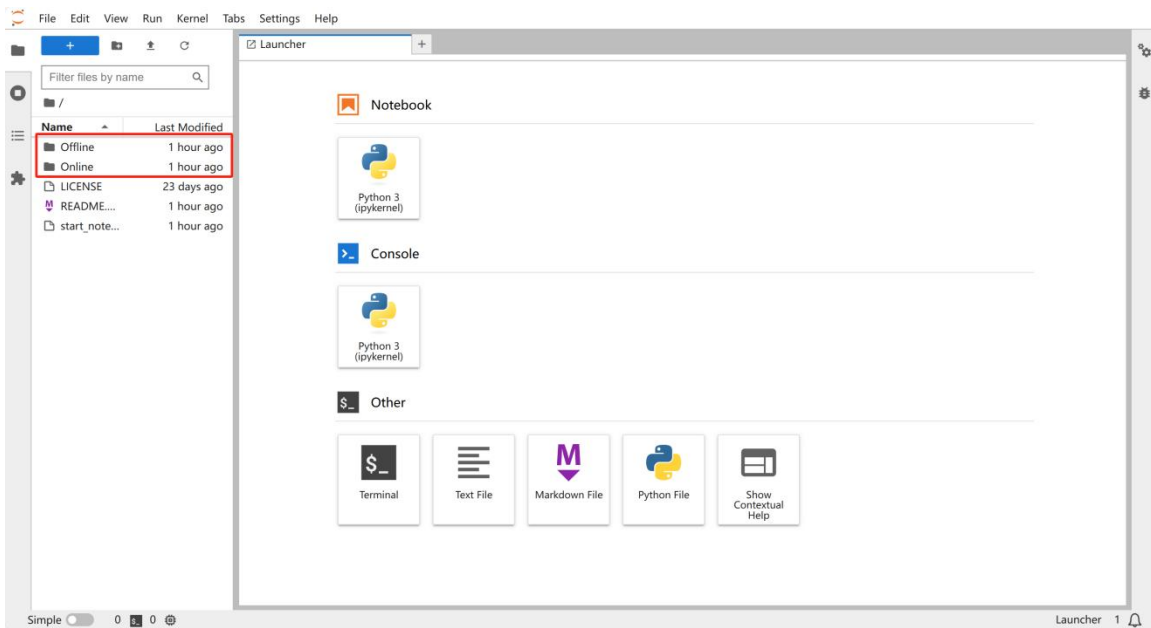
5) Then open Firefox browser.



6) Then enter the URL link you see above in the browser to log in to the Jupyter Lab software.



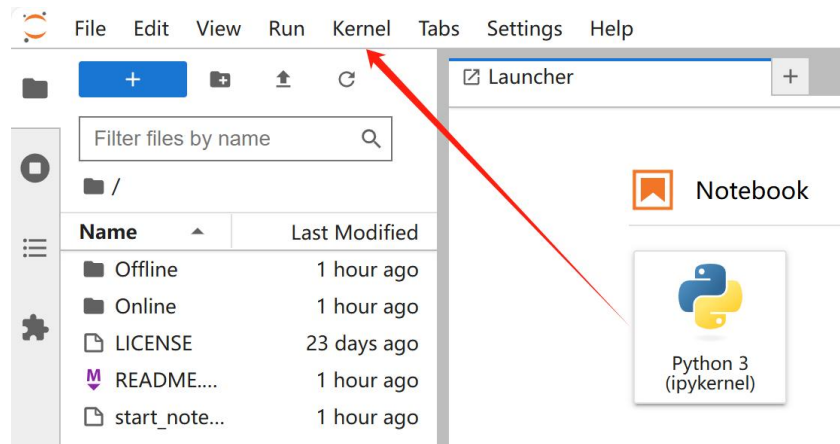
7) The interface after logging into Jupyter Lab is shown below. On the left side of the file manager are the AI application sample folders for two running modes, Jupyter Lab startup scripts, and some instruction files.



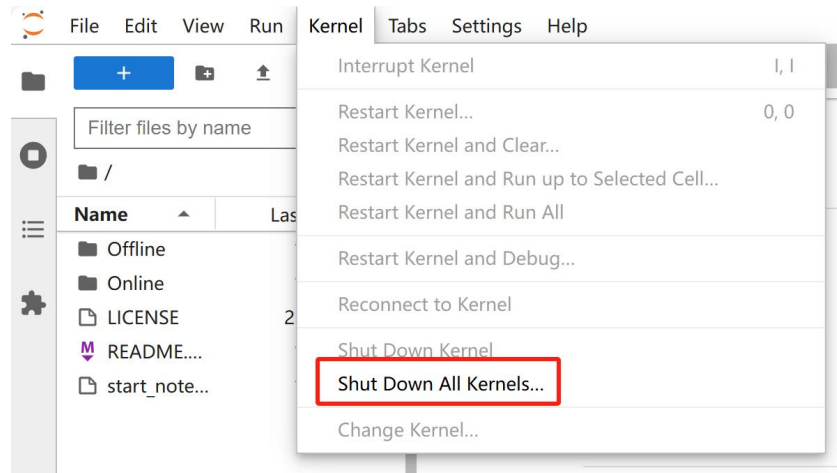
5.2. Methods for freeing memory

After running a sample in Jupyter Lab, closing the sample's tab will not release the memory occupied by the corresponding sample. We can reclaim memory by closing the kernel, as follows:

1) First, choose **Kernel**.



2) Then select **Shut Down All Kernels...**.



3) Then select **Shut Down All** to free up the memory occupied by the sample.



After testing each sample, it is recommended to free up memory before running the next sample to avoid sample failure due to insufficient memory.

5. 3. Methods for Running Online Reasoning Cases

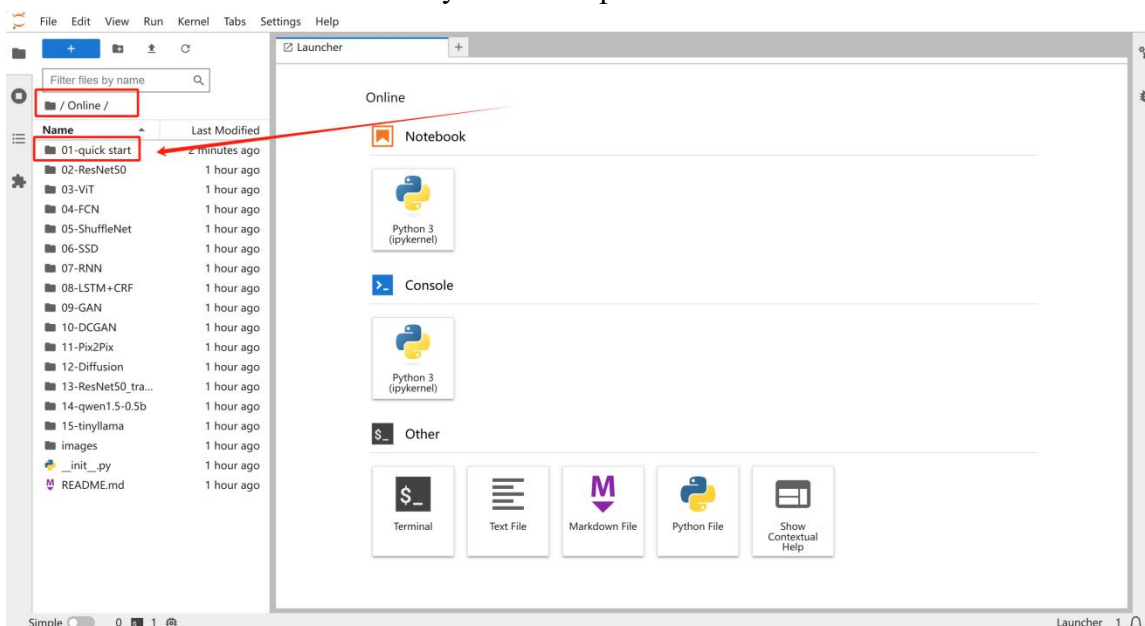
Note that if you need to run these samples, please try to run them on a development board with 24GB of memory. To run this sample smoothly on a development board with 12GB of memory, it is necessary to set at least 12GB of Swap memory. Otherwise, the system may automatically kill processes due to insufficient memory. You can refer to the instructions in the section on [how to set Swap memory](#).

5. 3. 1. Running a Simple Deep Learning Model

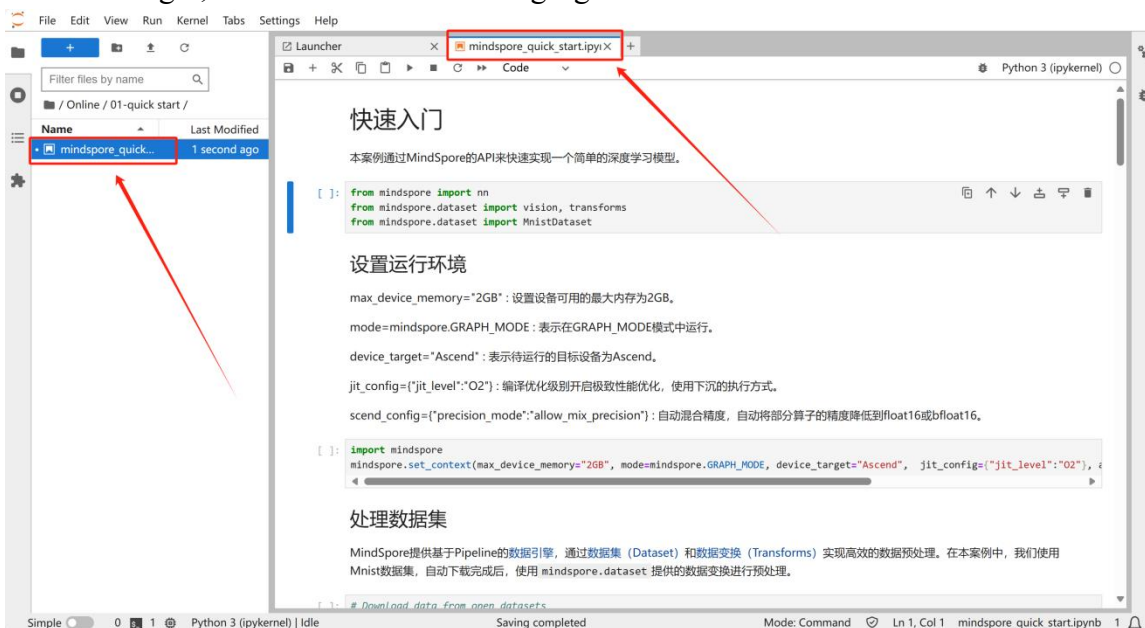
This case uses MindSpore's API to quickly implement a simple deep learning model. You can run the sample in Jupyter Lab according to the following process.



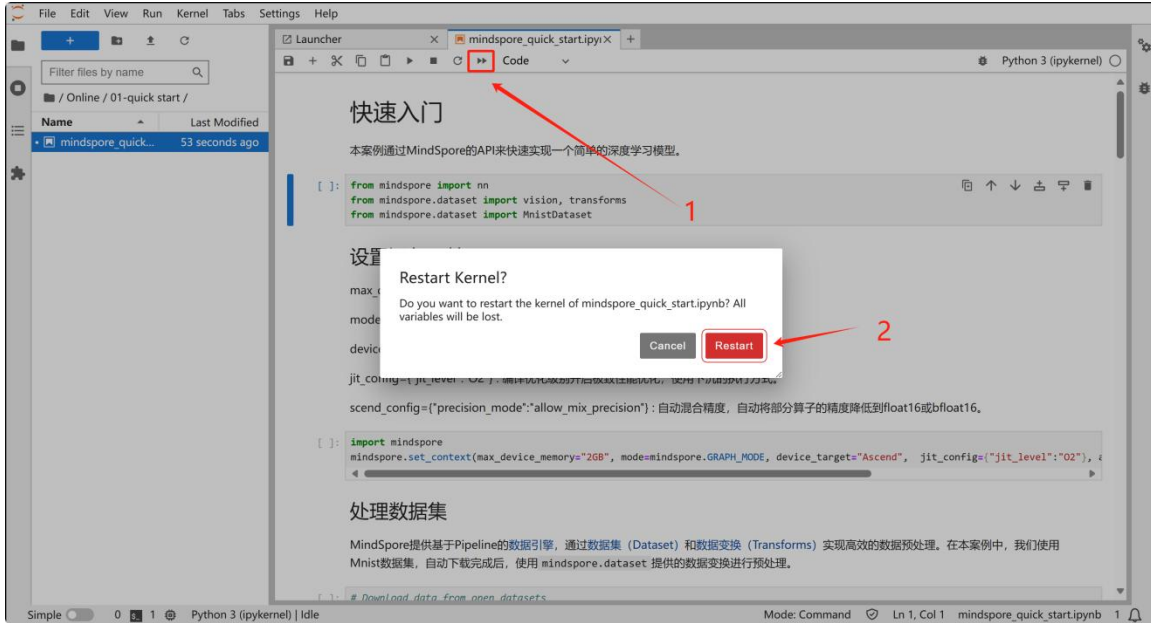
1) Firstly, double-click on **01-quick start** as shown in the following image on the Jupyter Lab interface to enter the directory of the sample.



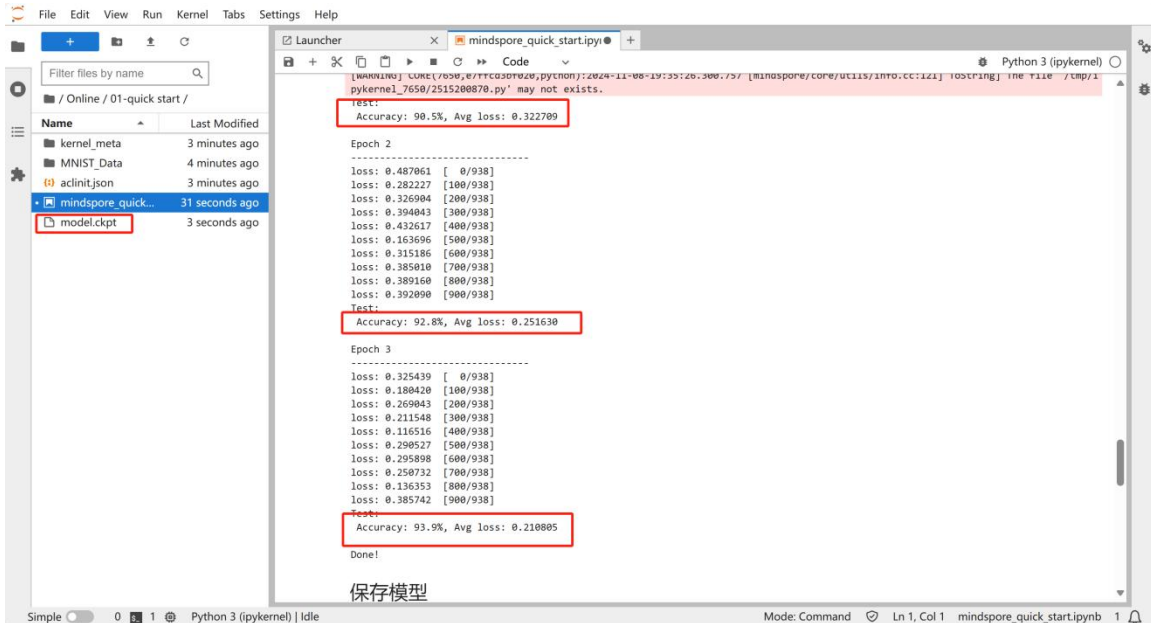
2) In this directory, there are all the resources to run the example, among which **mindspore_quick_start.ipynb** is the file to run the example in Jupyter Lab. Double click to open **mindspore_quick_start.ipynb**, and the contents of the file will be displayed in the window on the right, as shown in the following figure:



3) Click the button to run the sample, and then click the Restart button in the pop-up dialog box.



4) During model training, the loss value and prediction Accuracy for each round will be printed, and it can be seen that the loss is continuously decreasing while the Accuracy is continuously improving. After the training is completed, we can save the model as shown in the following figure:



5) Then we can load the newly trained model and make predictions, and the prediction results are as follows:



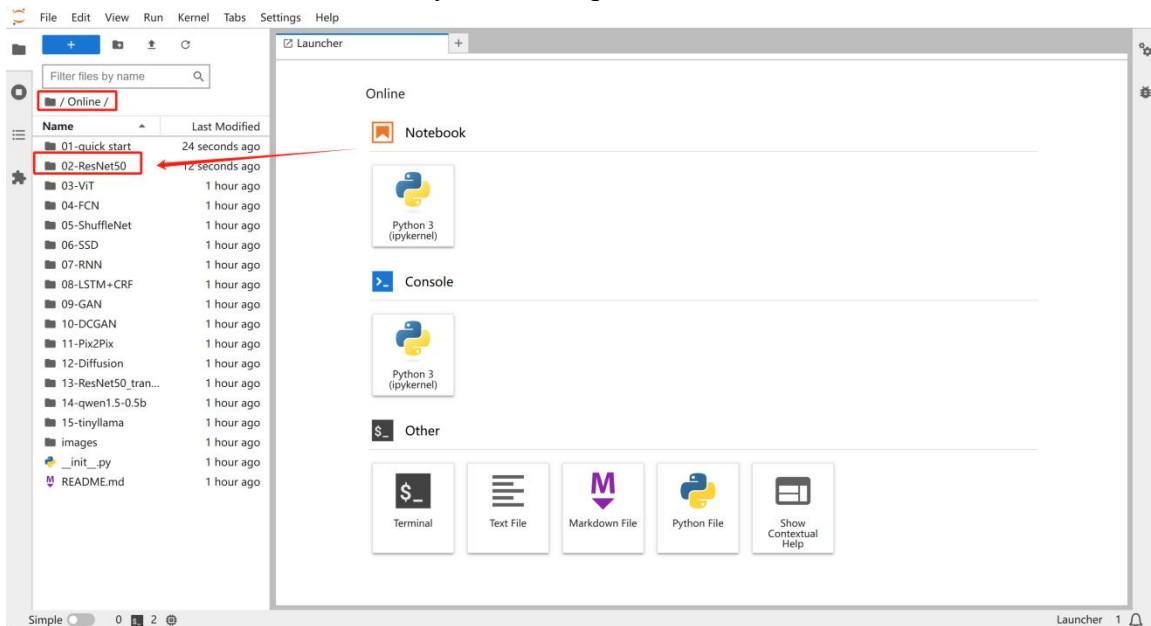
```
[16]: model.set_train(False)
      for data, label in test_dataset:
          pred = model(data)
          predicted = pred.argmax(1)
          print(f'Predicted: "{predicted[:10]}"', Actual: "{label[:10]}")
          break
```

Predicted: "[4 1 3 3 2 1 2 1 3 3]", Actual: "[4 1 3 3 2 1 2 1 3 3]"

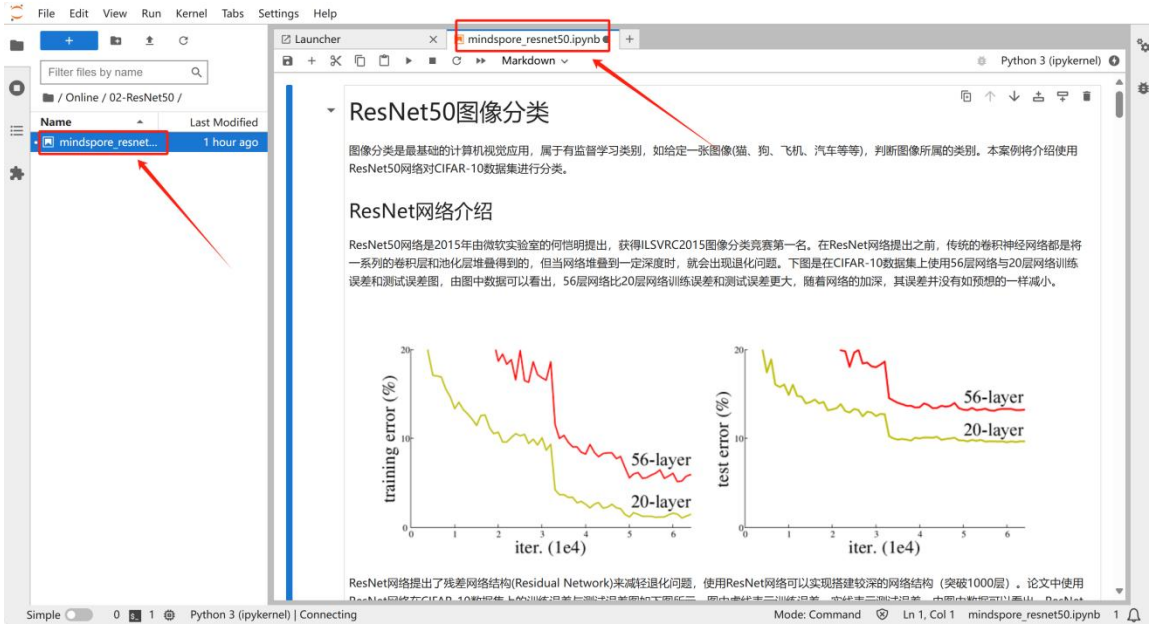
5.3.2. Running ResNet50 Image Classification Sample

Image classification is the most fundamental computer vision application, belonging to the category of supervised learning, such as determining the category of an image given (cat, dog, airplane, car, etc.). ResNet proposed the Residual Network structure to alleviate the degradation problem, and using ResNet network can achieve the construction of deeper network structures (breaking through 1000 layers). This example uses ResNet50 network to perform offline inference classification on the CIFAR-10 dataset. The running of the sample can be completed through the following steps:

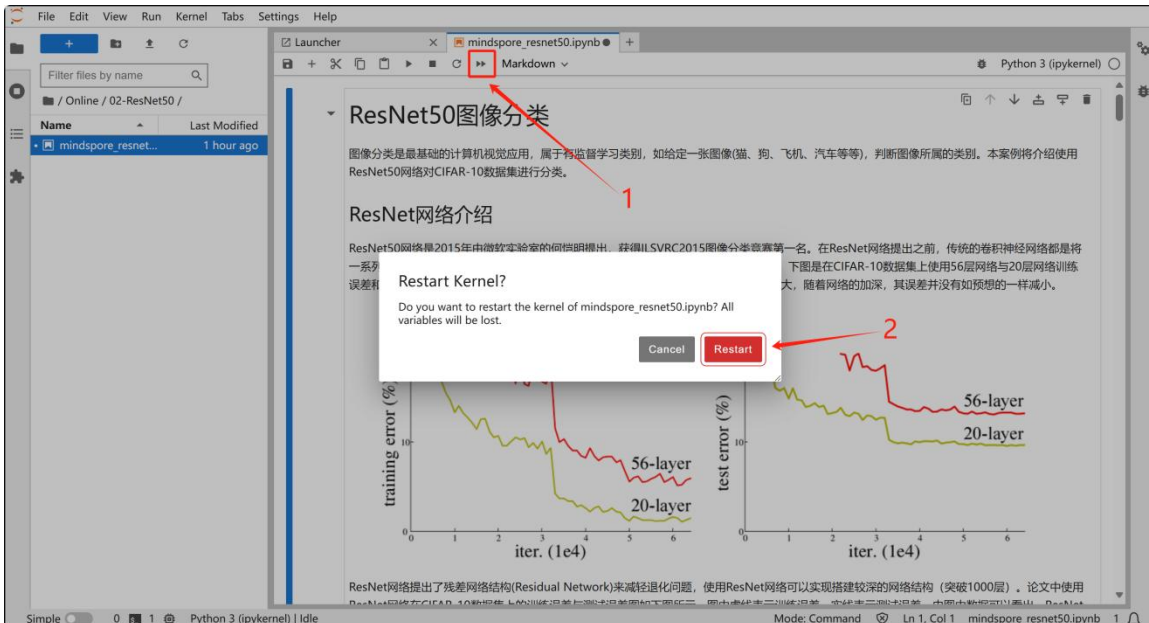
1) Firstly, double-click on **02-ResNet50** as shown in the following image on the Jupyter Lab interface to enter the directory of the sample.



2) In this directory, there are all the resources to run the example, among which **mindspore_resnet50.ipynb** is the file to run the example in Jupyter Lab. Double click to open **mindspore_resnet50.ipynb**, and the contents of this file will be displayed in the window on the right, as shown in the following figure:



3) Click the  button to run the sample, and then click the **Restart** button in the pop-up dialog box.



4) After the program is executed, the image classification results can be successfully obtained in the notebook document. If the predicted font color is **blue**, it indicates a correct prediction; if the predicted font color is **red**, it indicates a prediction error. As shown in the following figure:



5) The path for saving test data is as follows:

/home/HwHiAiUser/orange-pi-mindspore/Online/02-ResNet50/datasets-cifar10-bin/cifar-10-batches-bin/

+ + ↑ ↺ Filter files by name 📁 / ... / datasets-cifar10-bin / cifar-10-batches-bin /	
Name	Last Modified
batches.meta.txt	15 years ago
data_batch_1.bin	15 years ago
data_batch_2.bin	15 years ago
data_batch_3.bin	15 years ago
data_batch_4.bin	15 years ago
data_batch_5.bin	15 years ago
readme.html	15 years ago
test_batch.bin	15 years ago

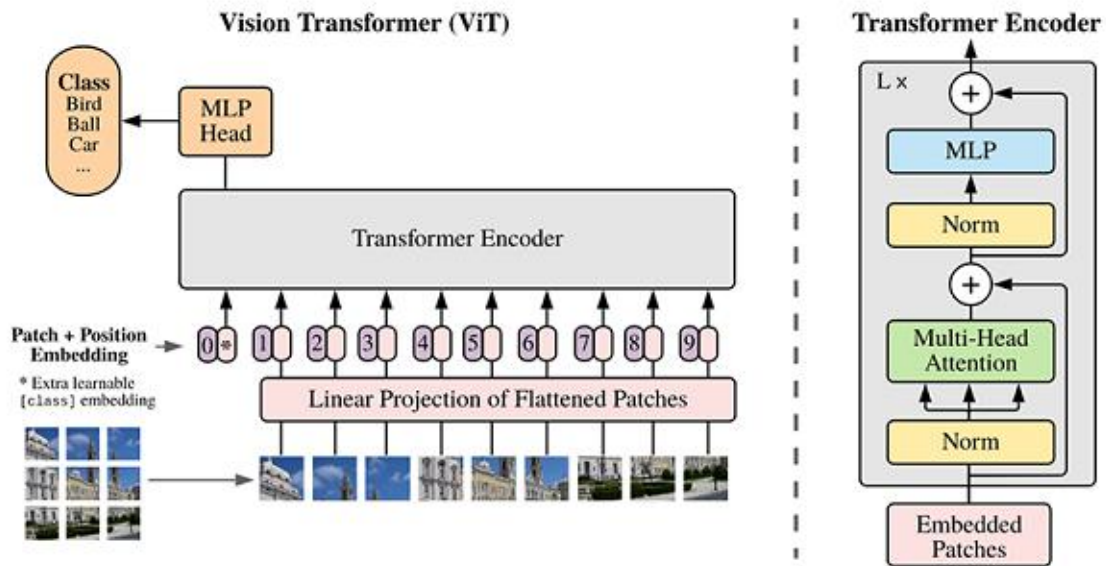
5. 3. 3. Running Vision Transformer Image Classification Sample

In recent years, with the development of models based on self attention structures, especially the introduction of Transformer models, it has greatly promoted the development of natural language processing models. ViT is the fusion of natural language processing a



nd computer vision. It can still achieve good results in image classification tasks without relying on convolution operations.

The main structure of the ViT model is based on the Encoder part of the Transformer model (some structures have been adjusted in order, such as the normalization position being different from the standard Transformer), and its structural diagram is as follows:



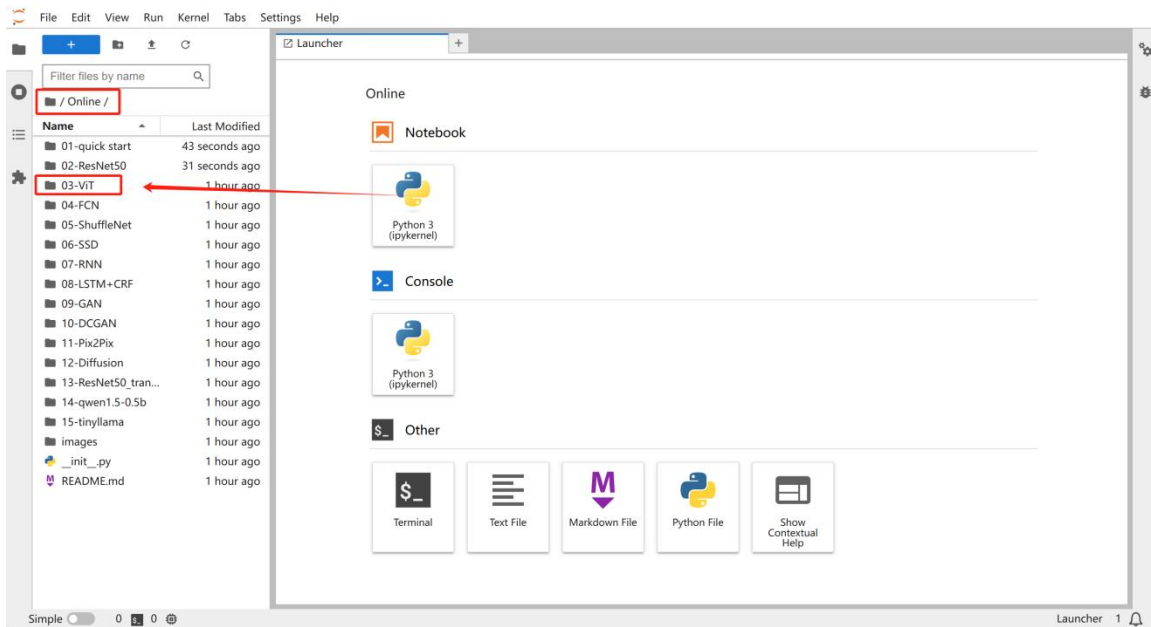
The ViT model is mainly applied in the field of image classification. Therefore, its model structure has the following characteristics compared to traditional Transformers:

1. After the original images in the dataset are divided into multiple patches (image blocks), the two-dimensional patches (without considering channels) are converted into one-dimensional vectors, and then class vectors and position vectors are added as model inputs.
2. The Block structure of the model body is based on the Encoder structure of Transformer, but the position of Normalization has been adjusted. Among them, the most important structure is still the Multi head Attention structure.
3. After stacking the Blocks, the model is connected to a fully connected layer and receives the output of the class vector as input for classification. Normally, we refer to the final fully connected layer as the Head and the Transformer Encoder part as the backbone.

The following will explain in detail the implementation of ImageNet classification task based on ViT through code examples.



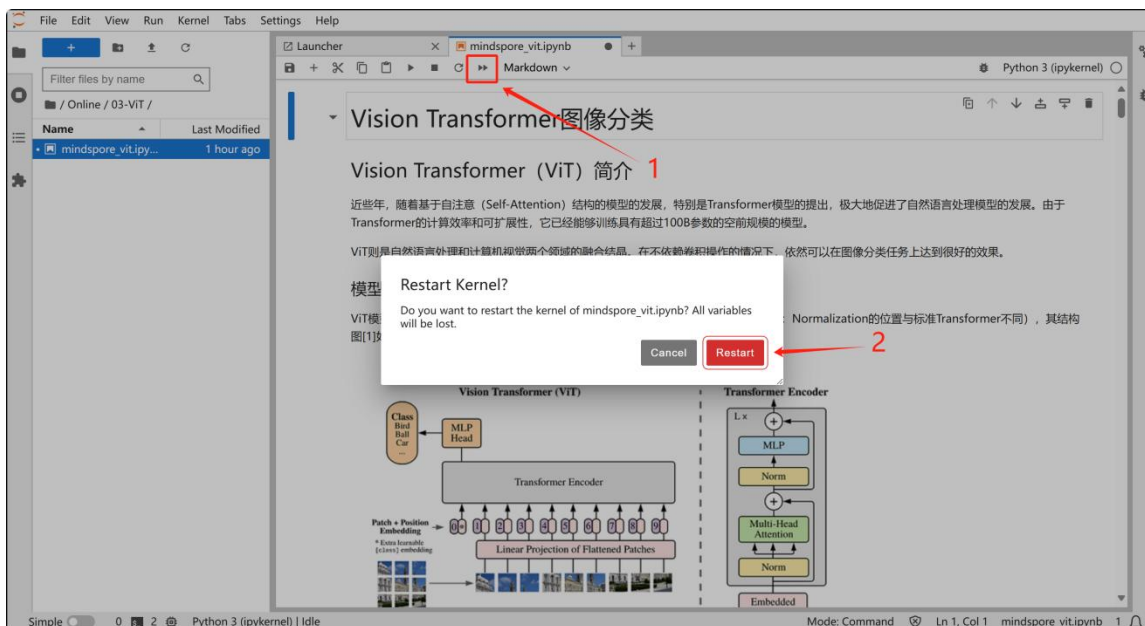
1) Firstly, double-click on **03-ViT** as shown in the following image on the Jupyter Lab interface to enter the directory of the sample.



2) There are all the resources to run the example in this directory, where **mindspore_vit.ipynb** is the file to run the example in Jupyter Lab. Double click to open **mindspore_vit.ipynb**, and the contents of this file will be displayed in the window on the right, as shown in the following figure:



3) Click the **Run** button to run this sample, and then click the **Restart** button in the pop-up dialog box.



4) After the program is executed, the inference result of the image (ILSVRC2012_test_0000279. JPEG) can be found in the inference folder (dataset/info). It can be seen that the predicted result is Doberman, which is the same as the expected result, verifying the accuracy of the model. As shown in the following figure:

```
Downloading data from https://mindspore-courses.obs.cn-north-4.myhuaweicloud.com/orange-pi-online-infer/03-ViT/vit\_b\_16\_224.ckpt (330.2 MB)
```

```
file_sizes: 100%|██████████| 346M/346M [01:53<00:00, 3.05MB/s]
```

```
Successfully downloaded file to ./vit_b_16_224.ckpt
```

```
[ERROR] CORE(4893,e7f5ee5f7020,python):2024-09-10-17:48:21.896.210 [mindspore/core/utils/file_utils.cc:253] GetRealPath] GetRealPath failed, path[/tmp/ipykernel_4893/3144920803.py]
[WARNING] CORE(4893,e7f5ee5f7020,python):2024-09-10-17:48:21.896.333 [mindspore/core/utils/info.cc:120] ToString] The file '/tmp/ipykernel_4893/3144920803.py' may not exists.
[ERROR] CORE(4893,e7f5ee5f7020,python):2024-09-10-17:48:21.904.338 [mindspore/core/utils/file_utils.cc:253] GetRealPath] GetRealPath failed, path[/tmp/ipykernel_4893/3144920803.py]
[WARNING] CORE(4893,e7f5ee5f7020,python):2024-09-10-17:48:21.904.377 [mindspore/core/utils/info.cc:120] ToString] The file '/tmp/ipykernel_4893/3144920803.py' may not exists.
{236: 'Doberman'}
```

推理过程完成后，在推理文件夹下可以找到图片的推理结果，可以看出预测结果是Doberman，与期望结果相同，验证了模型的准确性。





5. 3. 4. Running FCN Image Semantic Segmentation Sample

Semantic segmentation of images is an important part of image processing and machine vision technology related to image understanding. It is an important branch in the field of AI and is often applied in areas such as facial recognition, object detection, medical imaging, satellite image analysis, and autonomous driving perception. The purpose of semantic segmentation is to classify each pixel in an image. Unlike ordinary classification tasks that only output a certain category, semantic segmentation tasks output images of the same size as the input, with each pixel in the output image corresponding to the category of each pixel in the input image. Semantics in the field of images refers to the content of an image and the understanding of its meaning.

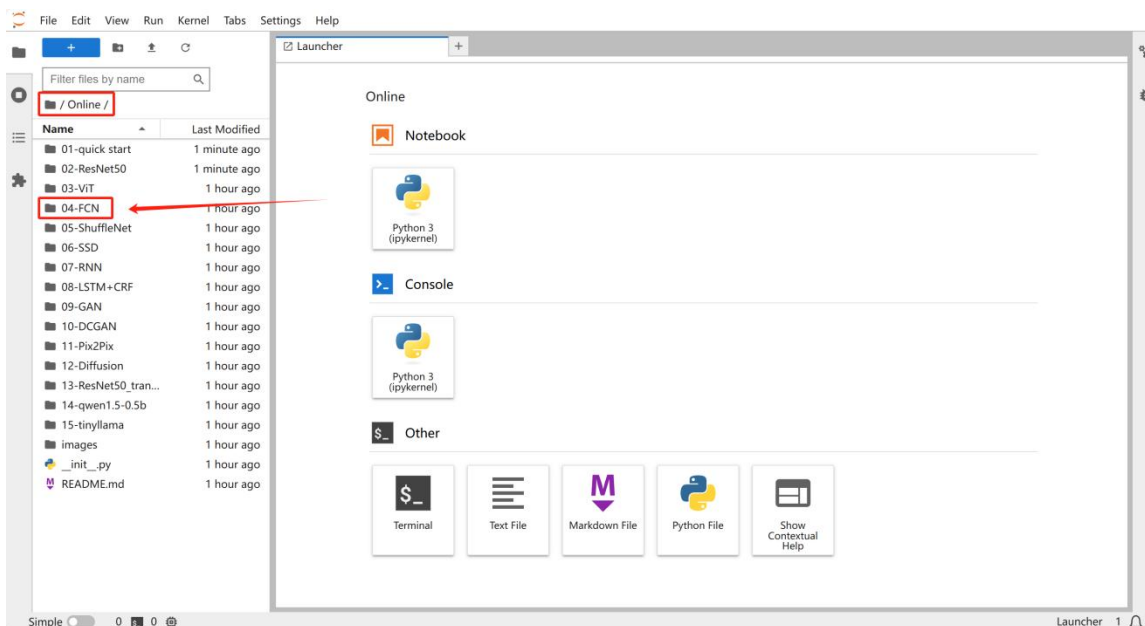
FCN (Fully Convolutional Networks) is the first fully convolutional network to perform pixel level prediction end-to-end. Directly obtain a label map of the same size as the original image through pixel level prediction. Because FCN discards fully connected layers and replaces them with fully convolutional layers, all layers in the network are convolutional layers, hence it is called a fully convolutional network.

FCN network features:

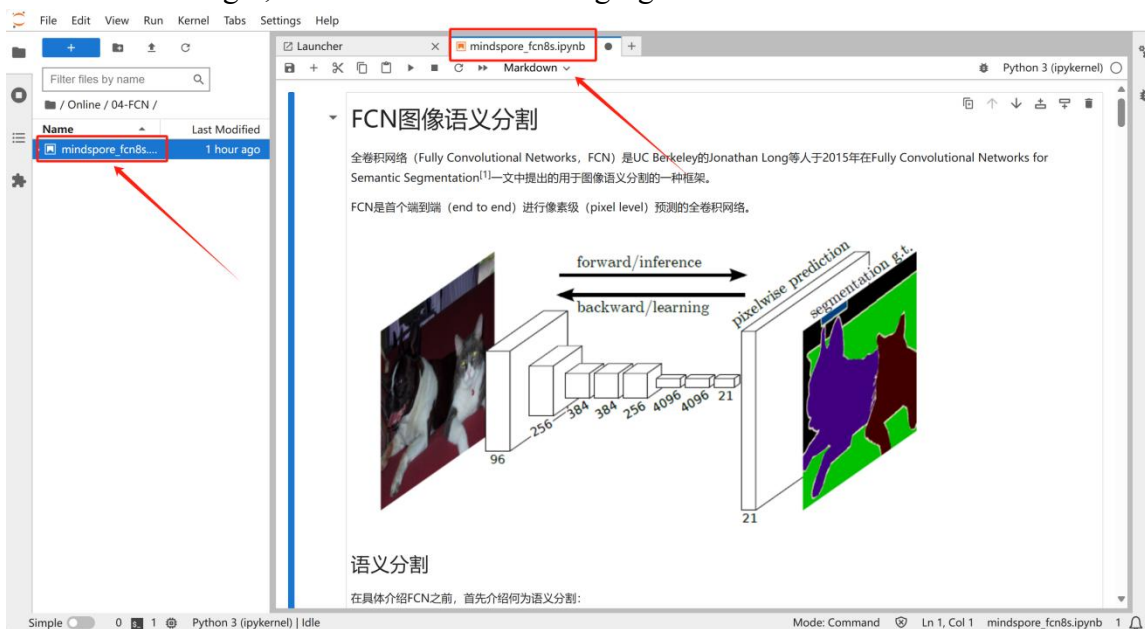
1. A fully convolutional network without fully connected layers (fc) that can adapt to inputs of any size.
2. The deconvolution layer with increased data size can output fine results.
3. Combining skip structures with results from different depth layers while ensuring robustness and accuracy.

This example can segment people and horses in images. You can run the sample in Jupyter Lab according to the following process.

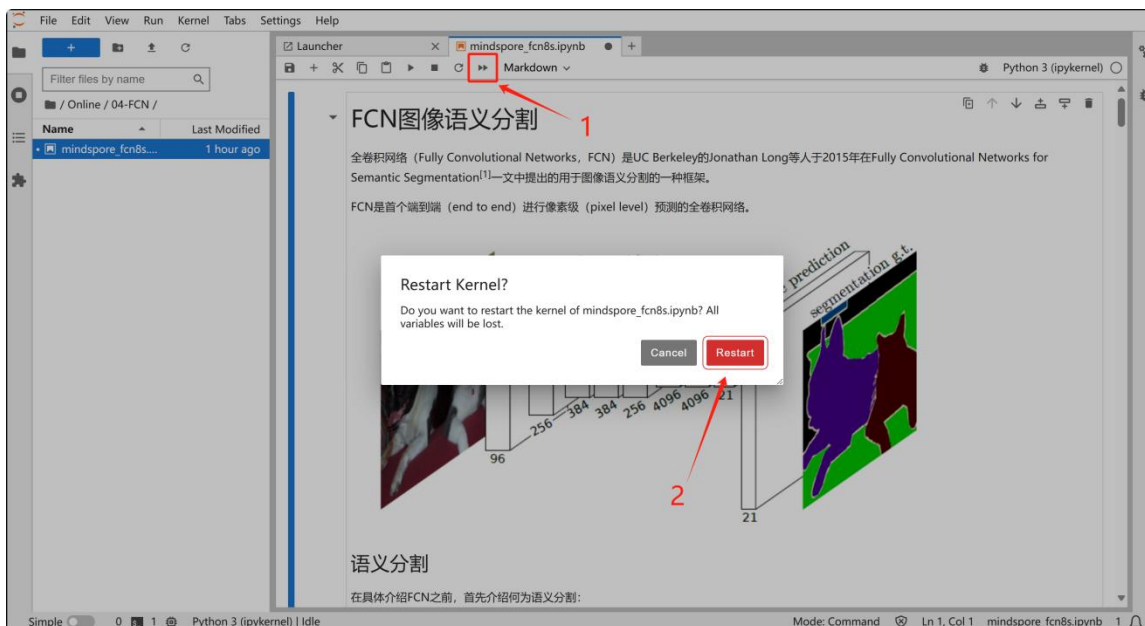
- 1) Firstly, double-click on the **04-FCN** shown in the following image on the Jupyter Lab interface to enter the directory of the sample.



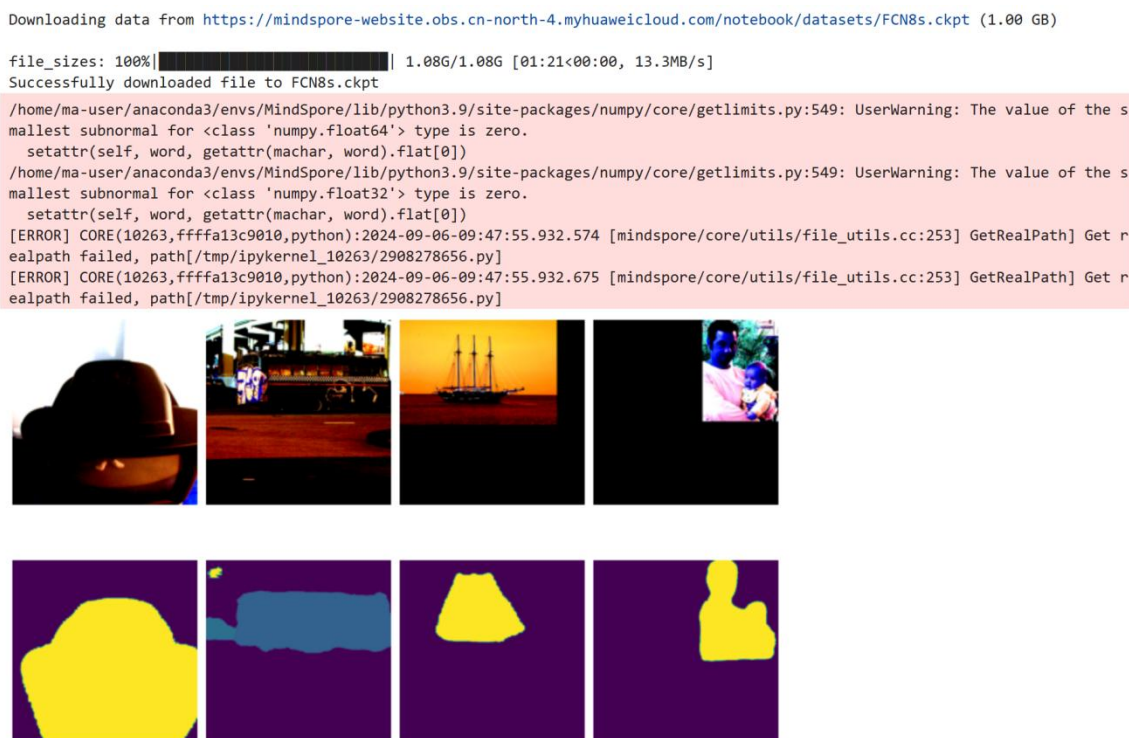
2) There are all the resources in this directory for running the example, among which **mindspore_fcn8s.ipynb** is the file for running the example in Jupyter Lab. Double click to open **mindspore_fcn8s.ipynb**, and the contents of this file will be displayed in the window on the right, as shown in the following figure:



3) Click the **▶▶** button to run this sample, and then click the **Restart** button in the pop-up dialog box.



4) After the program is executed, the result of image conversion can be successfully displayed in the notebook document. As shown in the following figure:



5. 3. 5. Running ShuffleNet Image Classification Sample

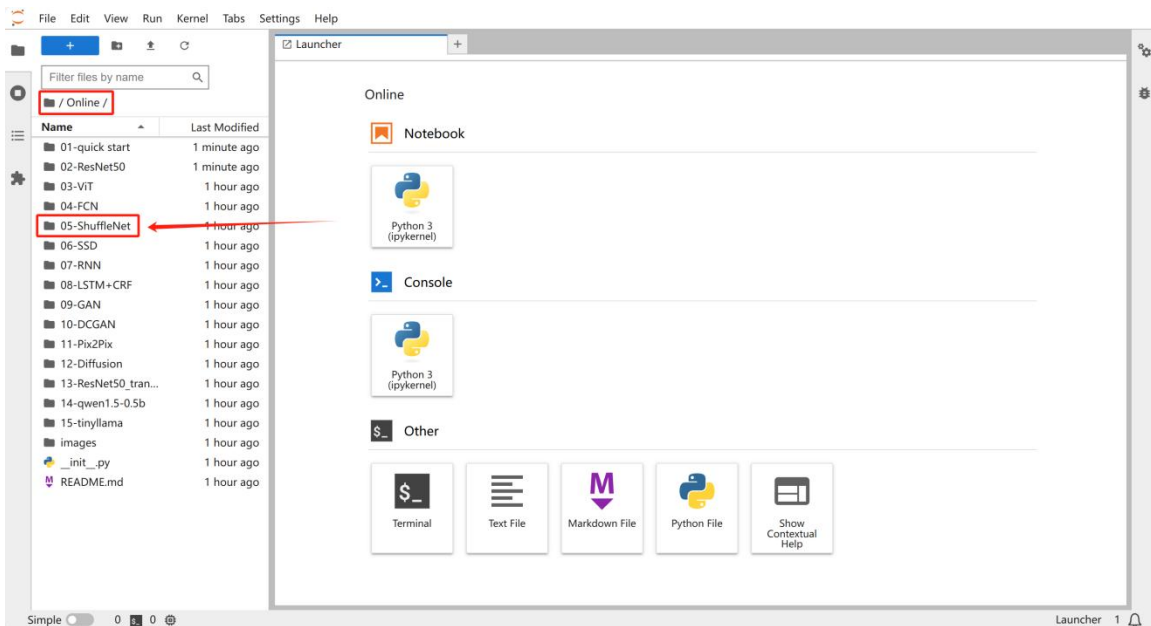
ShuffleNetV1 is a computationally efficient CNN model proposed by Megvii Technology, along with MobileNet, SqueezeNet and others are mainly used on mobile devices,



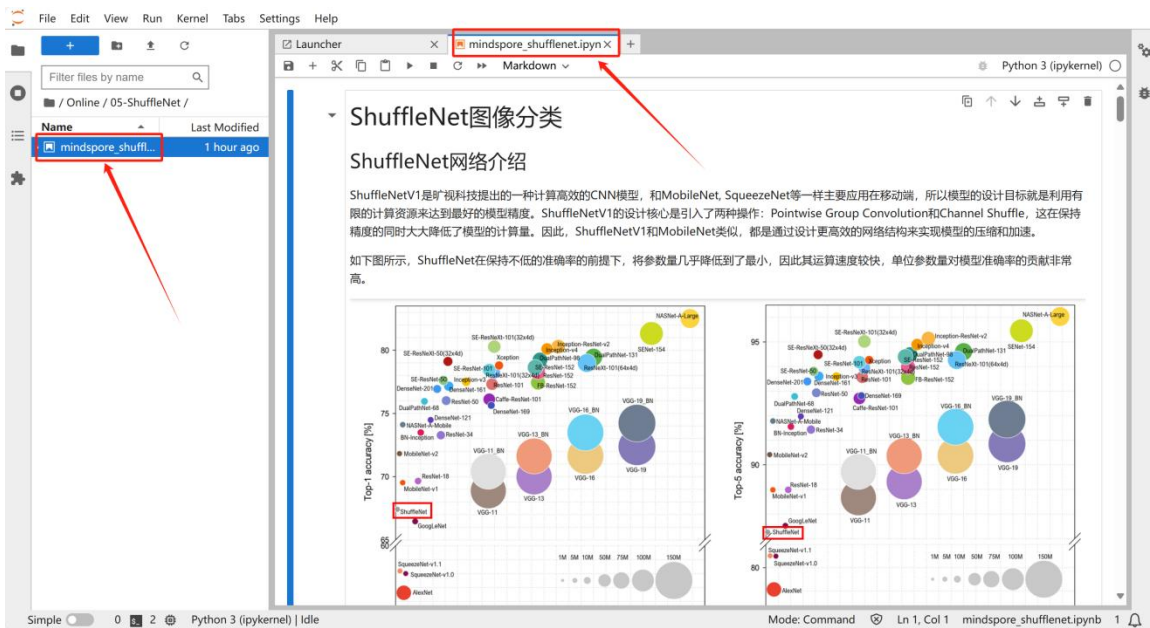
so the design goal of the model is to achieve the best model accuracy using limited computing resources. The design core of ShuffleNetV1 is the introduction of two operations: Pointwise Group Convolution and Channel Shuffle, which greatly reduces the computational complexity of the model while maintaining accuracy. Therefore, ShuffleNetV1, similar to MobileNet, achieves model compression and acceleration by designing more efficient network structures.

You can run the sample in Jupyter Lab according to the following process.

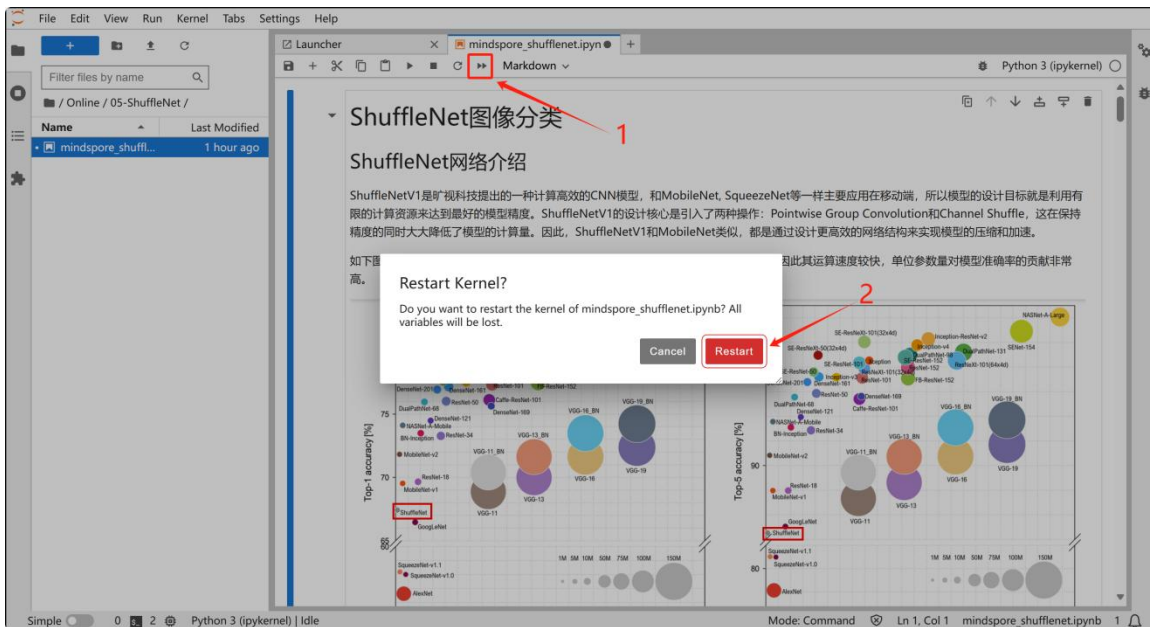
1) Firstly, double-click on **05-ShuffleNet** as shown in the following image on the Jupyter Lab interface to enter the directory of the sample.



2) There are all the resources to run the example in this directory, where **mindspore_shuffleNet.ipynb** is the file to run the example in Jupyter Lab. Double click to open **mindspore_shuffleNet.ipynb**, and the contents of this file will be displayed in the window on the right, as shown in the following figure:



3) Click the  button to run this sample, and then click the **Restart** button in the pop-up dialog box.



4) After the program is executed, the image classification results can be successfully displayed in the notebook document. As shown in the following figure:

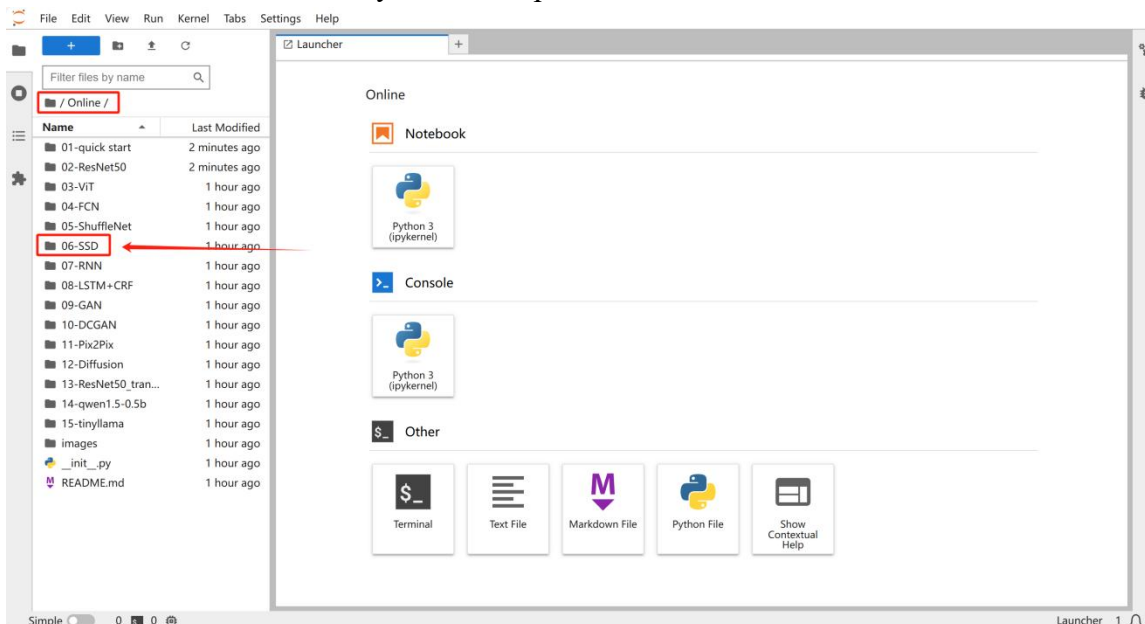


5. 3. 6. Running SSD Object Detection Sample

SSD is a single-stage object detection algorithm that extracts features through convolutional neural networks. SSD extracts different feature layers for detection output, so SSD is a multi-scale detection method. The dataset used in this example is COCO 2017. In order to save and load data more conveniently, the COCO dataset is first converted to MindRecord format before data reading in this example.

We can run the sample in Jupyter Lab according to the following process.

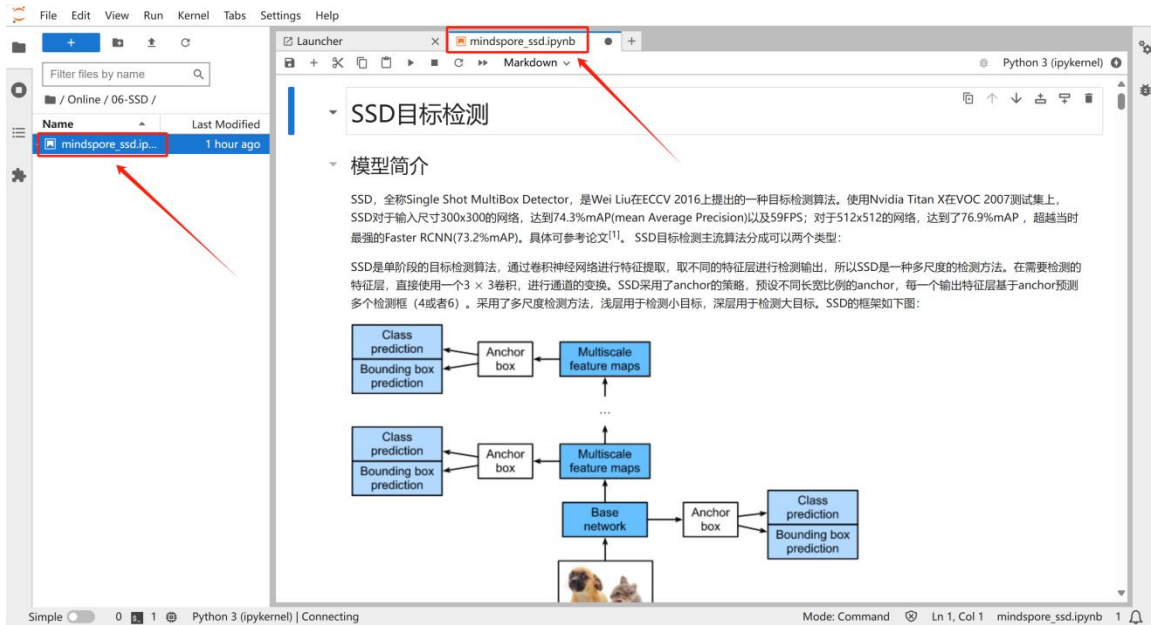
1) Firstly, double-click on the **06-SSD** shown in the following image on the Jupyter Lab interface to enter the directory of the sample.



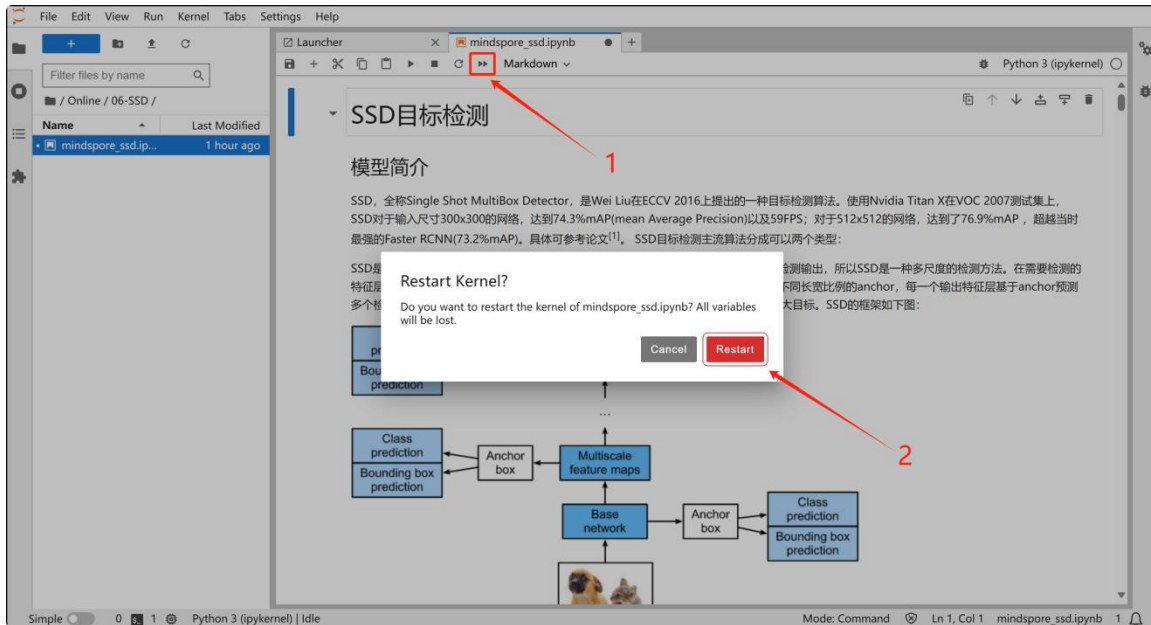
2) There are all the resources to run the example in this directory, where **mindspore_ssd.ipynb** is the file to run the example in Jupyter Lab. Double click to open **mindspore_ssd.ipynb**, and the contents of the file will be displayed in the window on the



right, as shown in the following figure:



3) Click the  button to run the sample, and then click the **Restart** button in the pop-up dialog box.



4) After the program is executed, the results of object detection will be displayed in the notebook document. As shown in the following figure:



```

creating index...
index created!
Running per image evaluation...
Evaluate annotation type *bbox*
DONE (t=4.92s).
Accumulating evaluation results...
DONE (t=1.10s).
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.001
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.002
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.000
Average Precision  (AP) @[ IoU=0.50:0.95 | area=  small | maxDets=100 ] = 0.000
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.039
Average Precision  (AP) @[ IoU=0.50:0.95 | area=  large | maxDets=100 ] = 0.004
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.001
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.020
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.054
Average Recall     (AR) @[ IoU=0.50:0.95 | area=  small | maxDets=100 ] = 0.000
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.057
Average Recall     (AR) @[ IoU=0.50:0.95 | area=  large | maxDets=100 ] = 0.283

=====
mAP: 0.0006232523571113707

```

5.3.7. Running RNN to Implement Emotion Classification Example

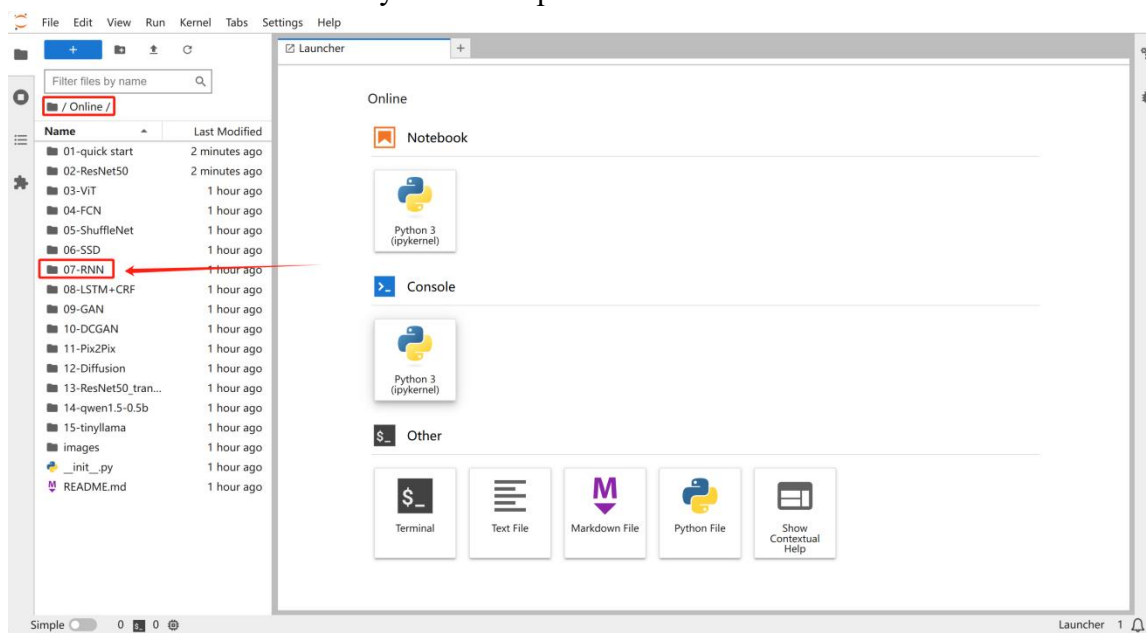
Emotion classification is a classic task in natural language processing and a typical classification problem. This case uses MindSpore to implement an emotion classification model based on RNN network, achieving the following effects:

input: This film is terrible
Correct label: Negative
Prediction label: Negative

input: This film is great
Correct label: Positive
Prediction label: Positive

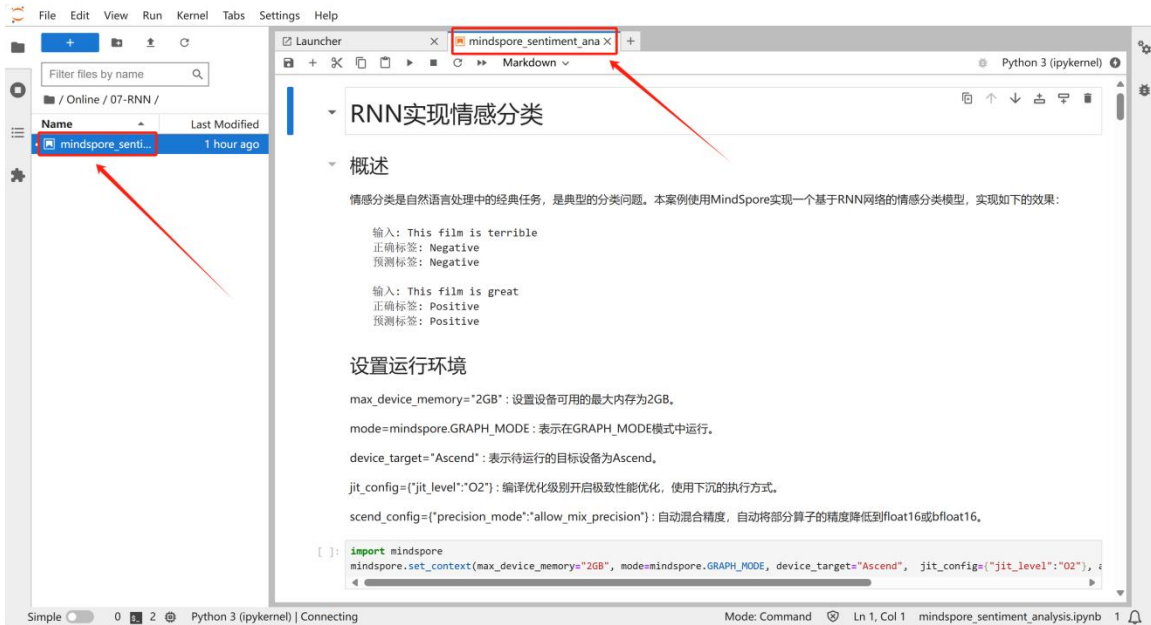
We can run the sample in Jupyter Lab according to the following process.

1) Firstly, double-click on the **07-RNN** shown in the following image on the Jupyter Lab interface to enter the directory of the sample.

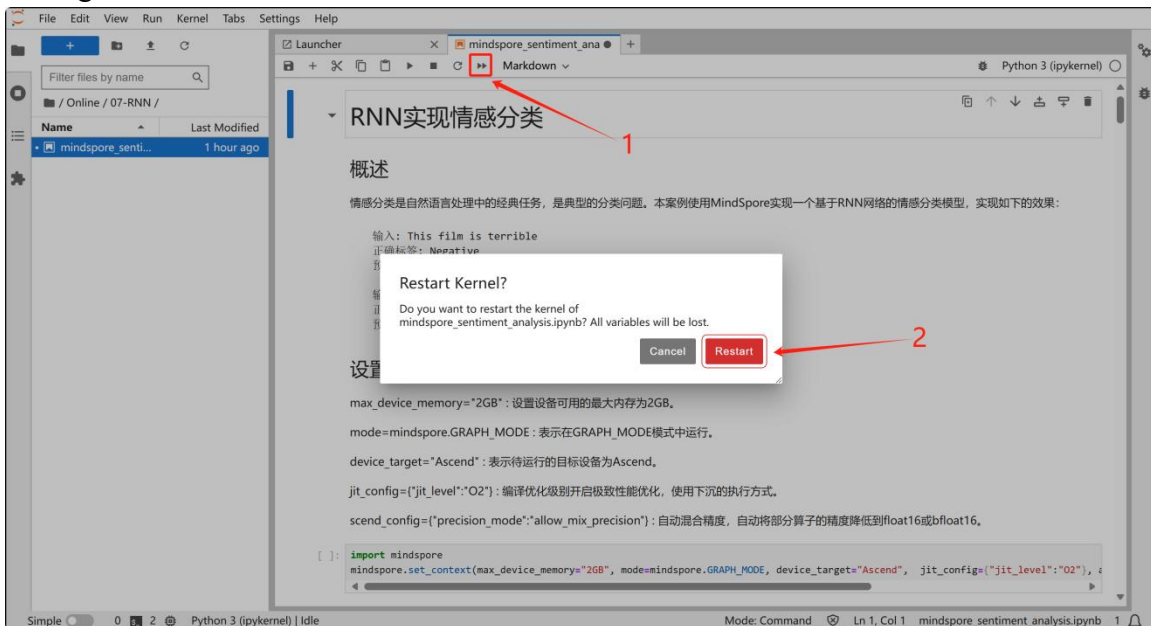




2) In this directory, there are all the resources to run the example, among which **mindspore_sentiment_analysis.ipynb** is the file to run the example in Jupyter Lab. Double click to open **mindspore_sentiment_analysis.ipynb**, and the contents of the file will be displayed in the window on the right, as shown in the following figure:



3) Click the **Run** button to run the sample, and then click the **Restart** button in the pop-up dialog box



4) After the program is executed, the results of sentiment classification will be displayed in the notebook document. As shown in the following figure:



```
[28]: 'Negative'
```

```
[29]: predict_sentiment(model, vocab, "This film is great")
```

```
[29]: 'Positive'
```

5. 3. 8. Running LSTM+CRF Sequence Annotation Sample

Sequence labeling refers to the process of labeling each Token in a given input sequence. The sequence tagging problem is commonly used for information extraction from text, including word segmentation, position tagging, Named Entity Recognition (NER), and so on. Taking named entity recognition as an example:

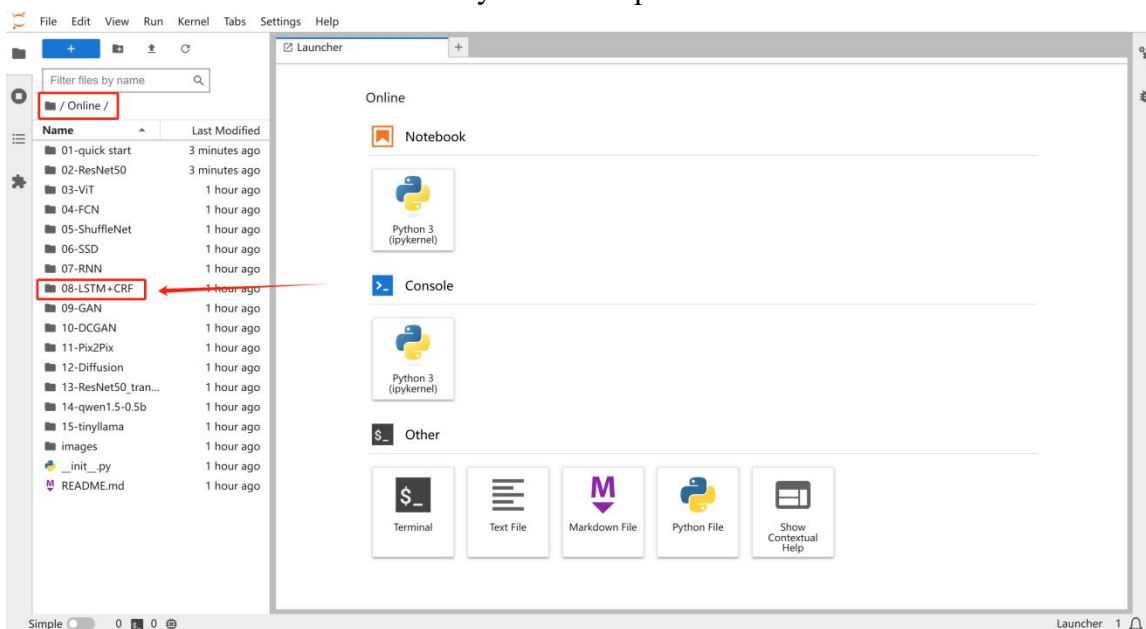
输入序列	清	华	大	学	座	落	于	首	都	北	京
输出标注	B	I	I	I	O	O	O	O	O	B	I

As shown in the table above, **Tsinghua University** and **Beijing** are place names that need to be recognized. We predict the labels of each input word and finally identify entities based on the labels.

Here, a common annotation method for named entity recognition is used - "BIOES" annotation, which marks the beginning of an entity as B, the rest as I, and non entities as O.

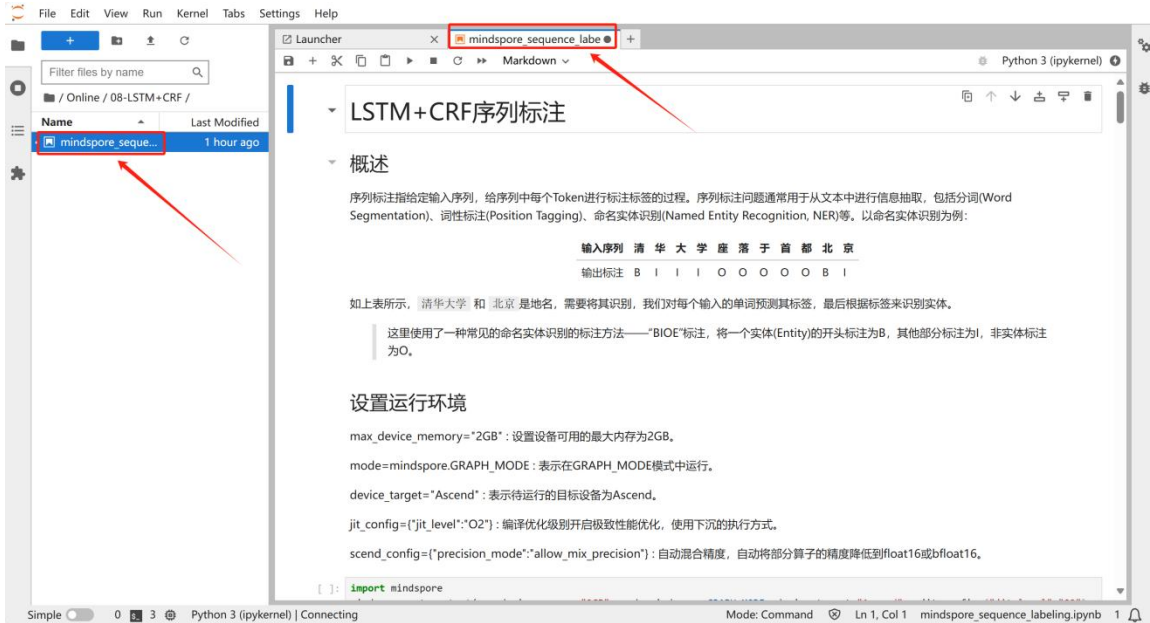
We can run the sample in Jupyter Lab according to the following process.

1) Firstly, double-click on **08-LSTM+CRF** as shown in the following image on the Jupyter Lab interface to enter the directory of the sample.

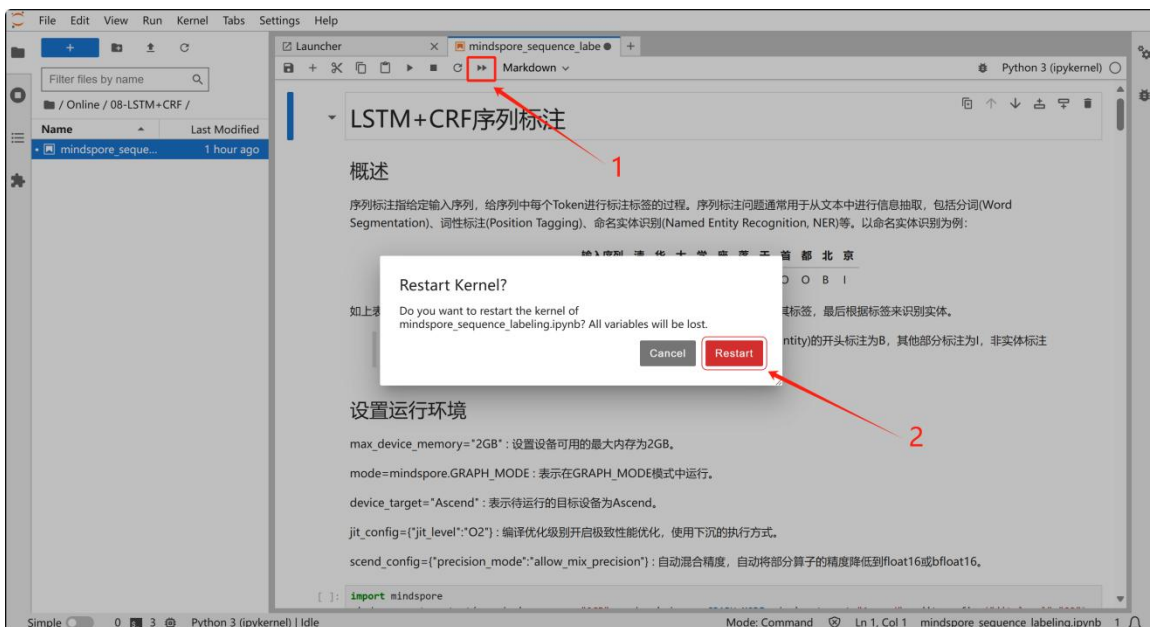




2) In this directory, there are all the resources to run the example, among which **mindspore_sequence_labeling.ipynb** is the file to run the example in Jupyter Lab. Double click to open **mindspore_sequence_labeling.ipynb**, and the contents of the file will be displayed in the window on the right, as shown in the following figure:



3) Click the **▶▶** button to run this sample, and then click the **Restart** button in the pop-up dialog box.



4) After the program is executed, the results of sequence annotation will be displayed in



the notebook document. As shown in the following figure:

```
[21]: idx_to_tag = {idx: tag for tag, idx in tag_to_idx.items()}

def sequence_to_tag(sequences, idx_to_tag):
    outputs = []
    for seq in sequences:
        outputs.append([idx_to_tag[i] for i in seq])
    return outputs

[22]: sequence_to_tag(predict, idx_to_tag)

[22]: [['B', 'I', 'I', 'I', 'O', 'O', 'O', 'O', 'O', 'B', 'I'],
       ['B', 'I', 'O', 'O', 'O', 'O', 'O', 'O', 'O']]
```

5.3.9. Running GAN Image Generation Sample

Generative Adversarial Networks (GANs) are a type of generative machine learning model and one of the most promising methods for unsupervised learning on complex distributions in recent years. It is mainly composed of two different models - the Generative Model and the Discriminative Model.

- The task of the generator is to generate "fake" images that appear to be training images;
- The discriminator needs to determine whether the image output from the generator is a real training image or a fake image.

GAN has achieved quite good output by designing two modules, the generative model and the discriminative model, to learn from each other through game theory. The core of the GAN model lies in proposing a novel framework for estimating generative models through adversarial processes. In this framework, two models will be trained simultaneously - the generative model G that captures the data distribution and the discriminative model D that estimates whether the samples come from the training data.

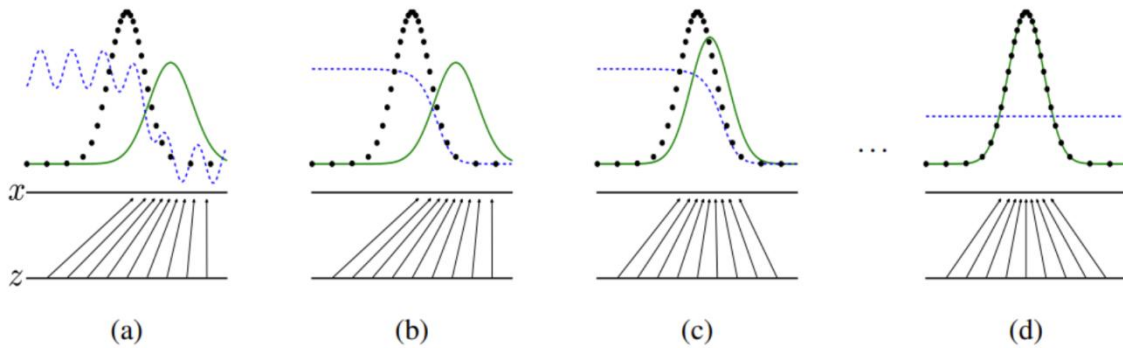
During the training process, the generator will constantly try to deceive the discriminator by generating better fake images, and the discriminator will gradually improve its discriminative ability in this process. The equilibrium point of this game is that when the distribution of the fake images generated by the generator is exactly the same as that of the training data images, the discriminator has a confidence level of 50% for true and false judgments.

Below we will briefly explain the game process between the generator and discriminator:

1. At the beginning of training, the quality of both the generator and discriminator is



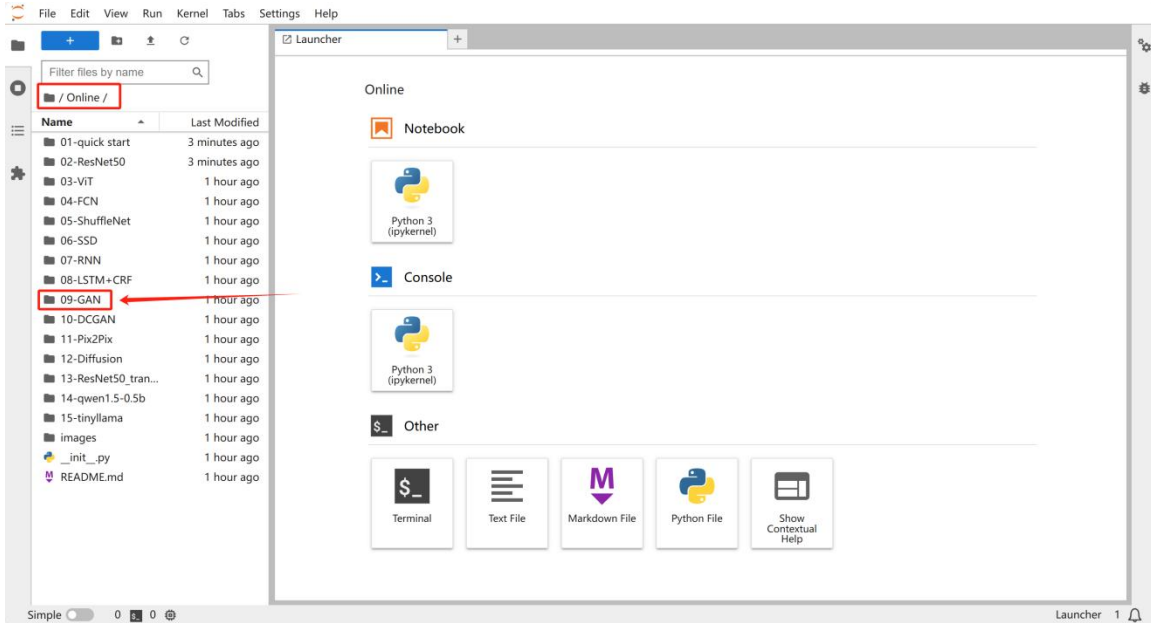
- relatively poor, and the generator will randomly generate a data distribution.
2. The discriminator optimizes the network by calculating the gradient and loss function, determining data close to the true data distribution as 1 and data close to the data distribution generated by the generator as 0.
 3. The generator generates data that is closer to the distribution of real data through optimization.
 4. When the data generated by the generator has the same distribution as the real data, the output of the discriminator is 1/2.



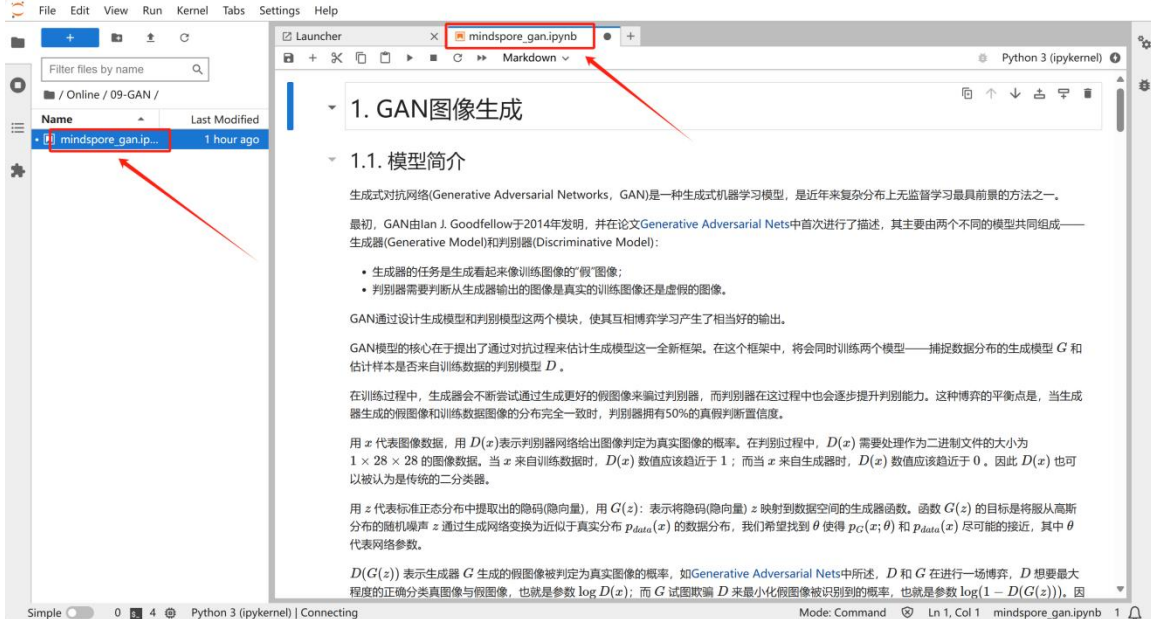
In the above figure, the blue dashed line represents the discriminator, the black dashed line represents the true data distribution, the green solid line represents the false data distribution generated by the generator, z represents the hidden code, and x represents the generated false image $G(z)$.

We can run the sample in Jupyter Lab according to the following process.

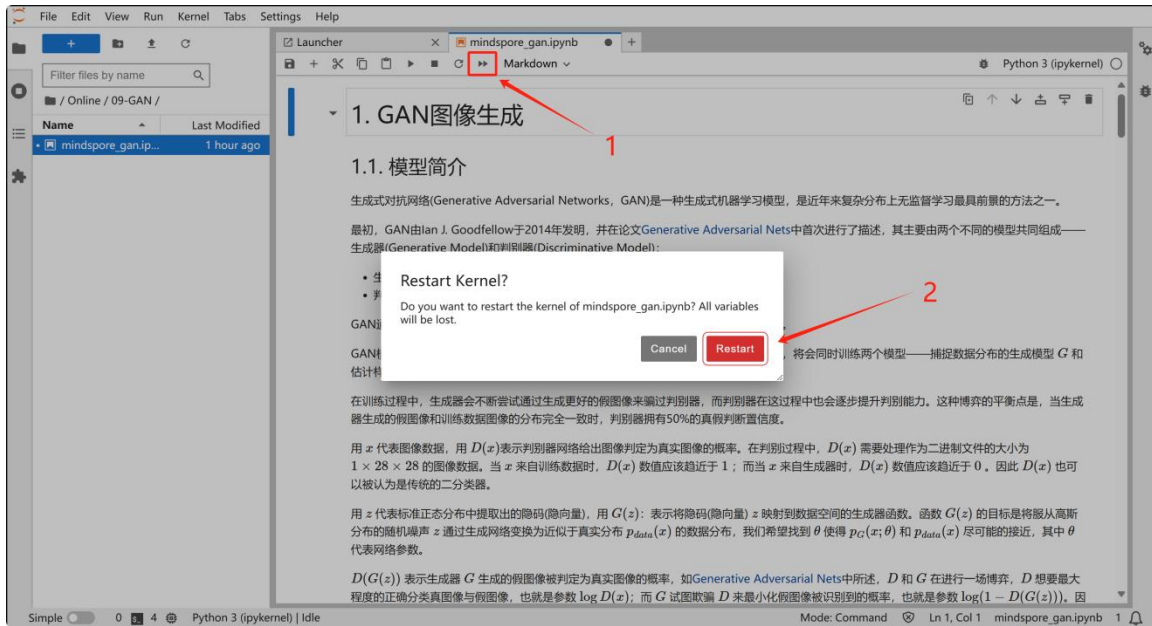
- 1) Firstly, double-click on the **09-GAN** shown in the following image on the Jupyter Lab interface to enter the directory of the sample.



2) There are all the resources to run the example in this directory, where **mindspore_gan.ipynb** is the file to run the example in Jupyter Lab. Double click to open **mindspore_gan.ipynb**, and the contents of this file will be displayed in the window on the right, as shown in the following figure:



3) Click the **▶▶** button to run the sample, and then click the **Restart** button in the pop-up dialog box.



4) As shown in the following figure, after the program is executed, the image generated by loading the generator network model parameter file can be successfully displayed in the notebook document:

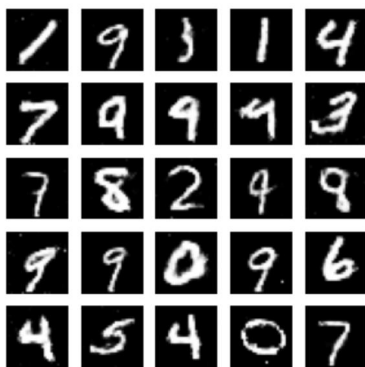
Downloading data from <https://cloudrive-kwe.huawei.com/kwe7/api/7606233104CC74CA2FA4D9FD2E46A535214755A9ABD9DF77F8BD3E79/476dac3cb5e0711ce51fead5004be1d4/Generator199.ckpt> (5.8 MB)

file_sizes: 100% | 6.06M/6.06M [00:00:00:00, 32.7MB/s]

[WARNING] ME(17688:281473359740944,MainProcess):2024-09-04-15:30:44.963.035 [mindspore/train/serialization.py:1469] For 'load_param_into_net', remove parameter prefix name: optim_g, continue to load.
Successfully downloaded file to ./Generator199.ckpt

[ERROR] CORE(17688,ffff9f9f0010,python):2024-09-04-15:30:45.207.436 [mindspore/core/utils/file_utils.cc:253] GetRealPath
Get realpath failed, path[/tmp/ipykernel_17688/3799052259.py]

[ERROR] CORE(17688,ffff9f9f0010,python):2024-09-04-15:30:45.207.521 [mindspore/core/utils/file_utils.cc:253] GetRealPath
Get realpath failed, path[/tmp/ipykernel_17688/3799052259.py]



5. 3. 10. Running DCGAN to Generate Manga Avatar Sample

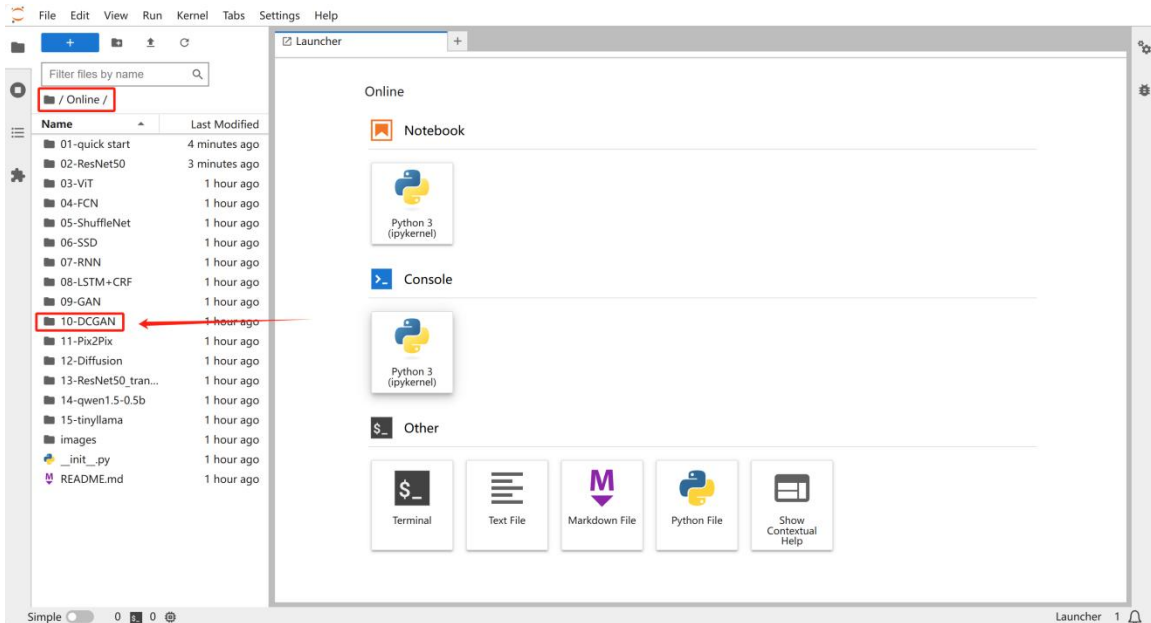
DCGAN (Deep Convolutional Generative Adversarial Networks) is a direct extension of GAN. The difference is that DCGAN uses convolutional and transposed convolutional layers separately in the discriminator and generator.



This case will use an anime avatar dataset to train a generative adversarial network, and then use the network to generate anime avatar images.

We can run the sample in Jupyter Lab according to the following process.

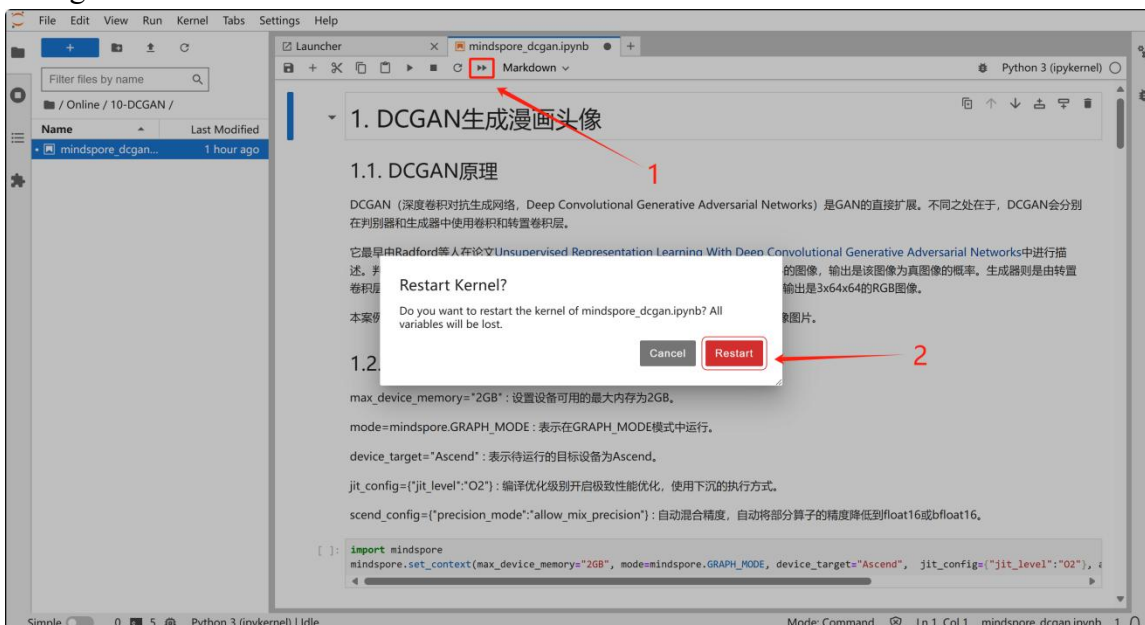
1) Firstly, double-click on the **10-DCGAN** shown in the following image on the Jupyter Lab interface to enter the directory of the sample.



2) There are all resources in this directory for running the example, where **mindspore_dcgan.ipynb** is the file for running the example in Jupyter Lab. Double click to open **mindspore_dcgan.ipynb**, and the contents of this file will be displayed in the window on the right, as shown in the following figure:



3) Click the  button to run this sample, and then click the **Restart** button in the pop-up dialog box.



4) As shown in the following figure, after the program is executed, the generated anime avatar image can be successfully displayed in the notebook document:



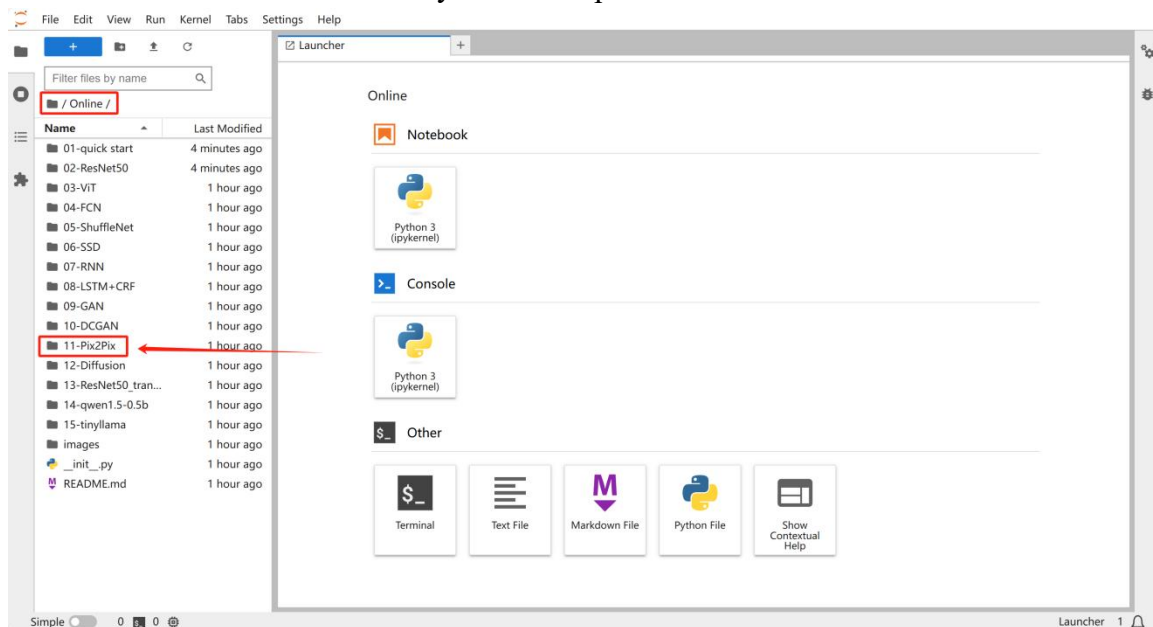
5. 3. 11. Running Pix2Pix to Implement Image Conversion Sample

Pix2Pix is a deep learning image transformation model based on Conditional Generative Adversarial Networks (cGAN). The model was proposed by Phillip Isola et al. at the 2017 CVPR, and can achieve the transformation of semantics/labels to real images, grayscale images to color images, aerial images to maps, day to night, and line draft images to physical images. Pix2Pix is a classic work that applies cGAN to supervised image to image translation, consisting of two models: a generator and a discriminator.

Traditionally, although the goal of such tasks is to predict pixels from pixels, each item is processed by a separate dedicated machine. The network used by Pix2Pix serves as a universal framework, utilizing the same architecture and objectives, and only training on different data to achieve satisfactory results. Given this, many people have already used this network to publish their own art works.

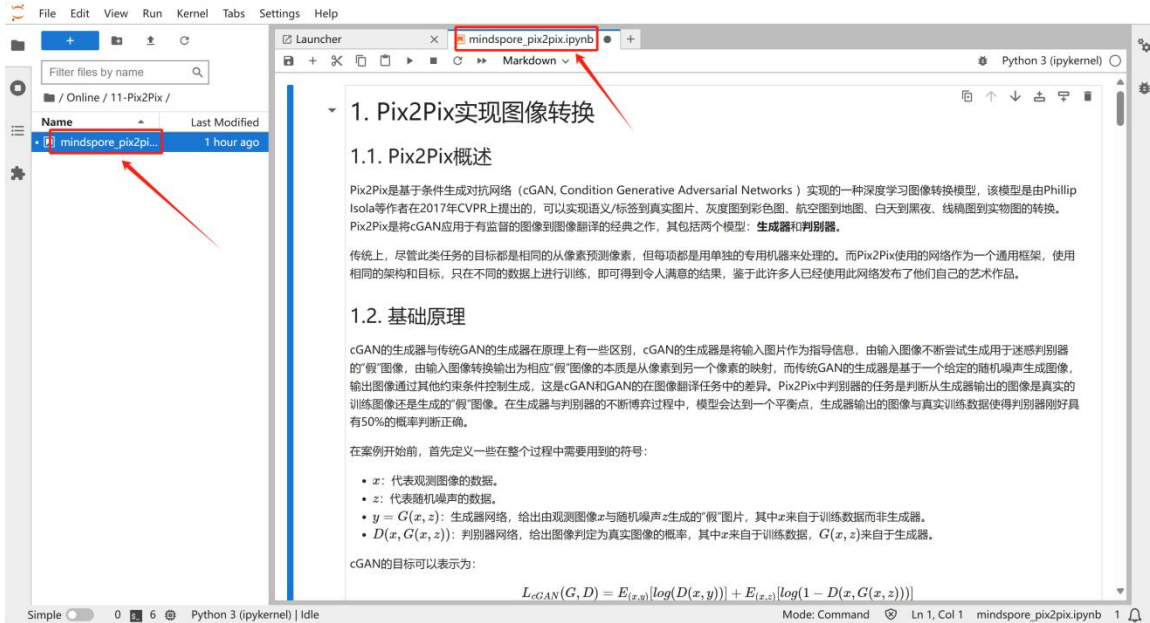
We can run the sample in Jupyter Lab according to the following process.

1) Firstly, double-click on **11-Pix2Pix** as shown in the following image on the Jupyter Lab interface to enter the directory of the sample.

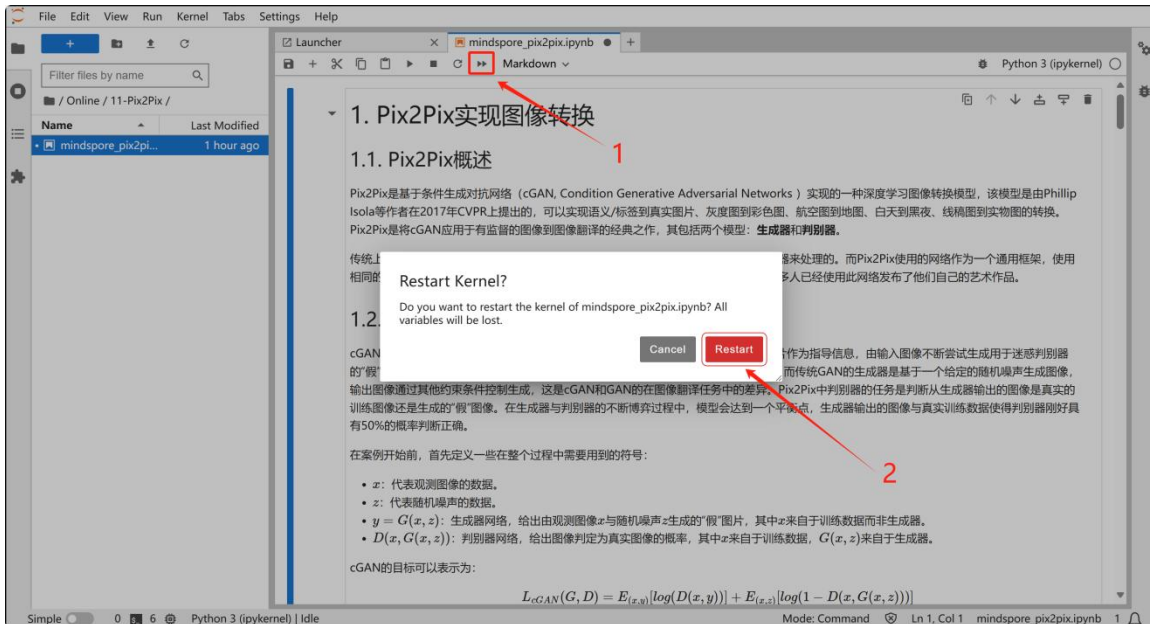




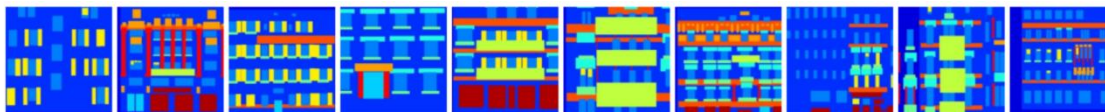
2) In this directory, there are all the resources to run the example, among which **mindspore_pix2pix.ipynb** is the file to run the example in Jupyter Lab. Double click to open **mindspore_pix2pix.ipynb**, and the contents of this file will be displayed in the window on the right, as shown in the following figure:



3) Click the **▶▶** button to run the sample, and then click the **Restart** button in the pop-up dialog box.

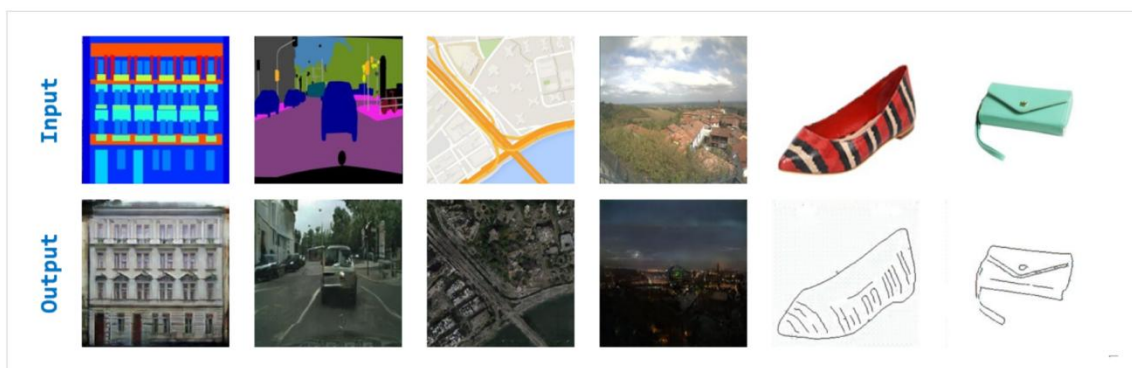


4) After the program is executed, the result of image conversion can be successfully displayed in the notebook document. As shown in the following figure:



time: 280.9498896598816

各数据集分别推理的效果如下

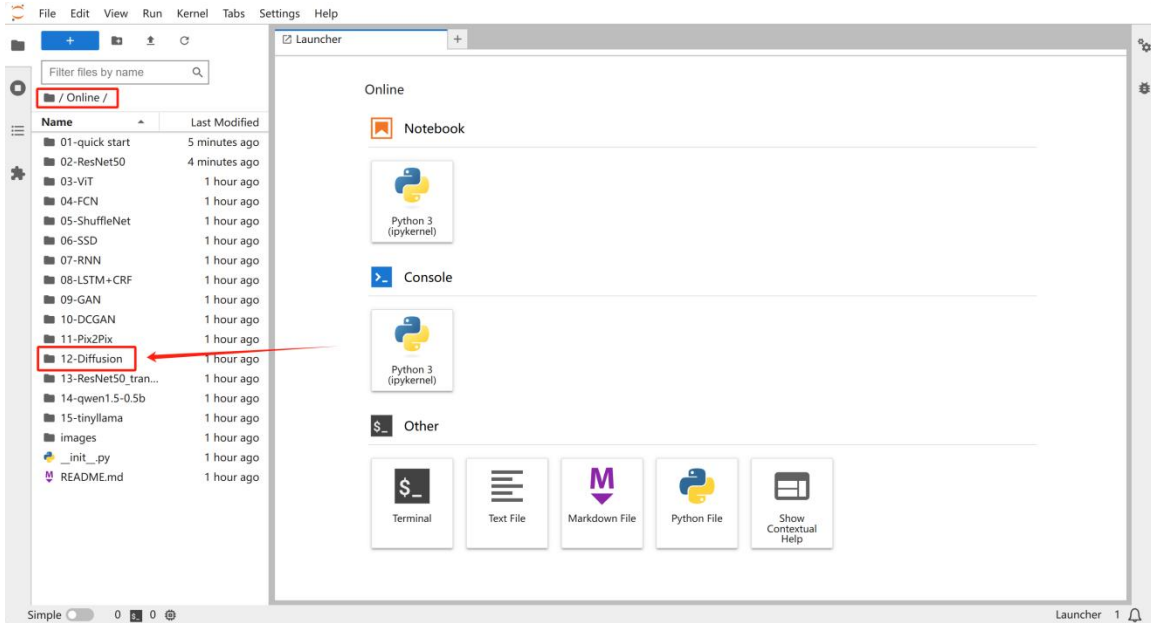


5. 3. 12. Example of Running Diffusion Model

The introduction of this case is based on the Denoising Diffusion Probabilistic Model (DDPM), which has achieved significant results in the field of (unconditional) image/audio/video generation. This case is based on Phil Wang's replication of the PyTorch framework (which is itself implemented based on TensorFlow) and migrated to the MindSpore AI framework for implementation.

We can run the sample in Jupyter Lab according to the following process.

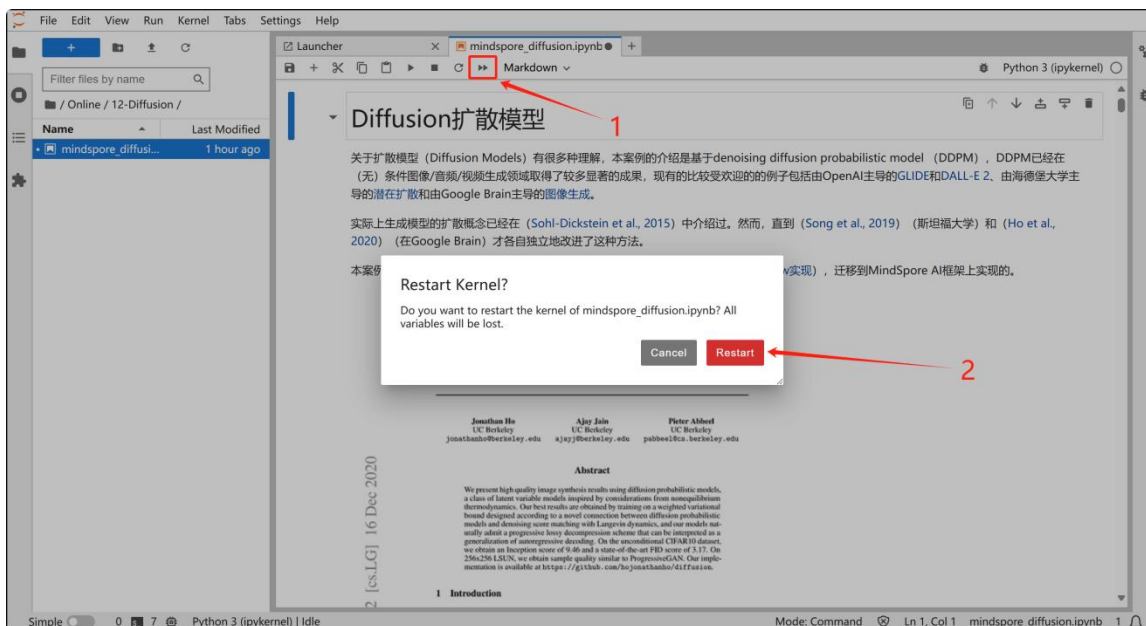
1) Firstly, double-click on the **12-Diffusion** shown in the following image on the Jupyter Lab interface to enter the directory of the sample.



2) In this directory, there are all the resources to run the example, among which **mindspore_diffusion.ipynb** is the file to run the example in Jupyter Lab. Double click to open **mindspore_diffusion.ipynb**, and the contents of this file will be displayed in the window on the right, as shown in the following figure:

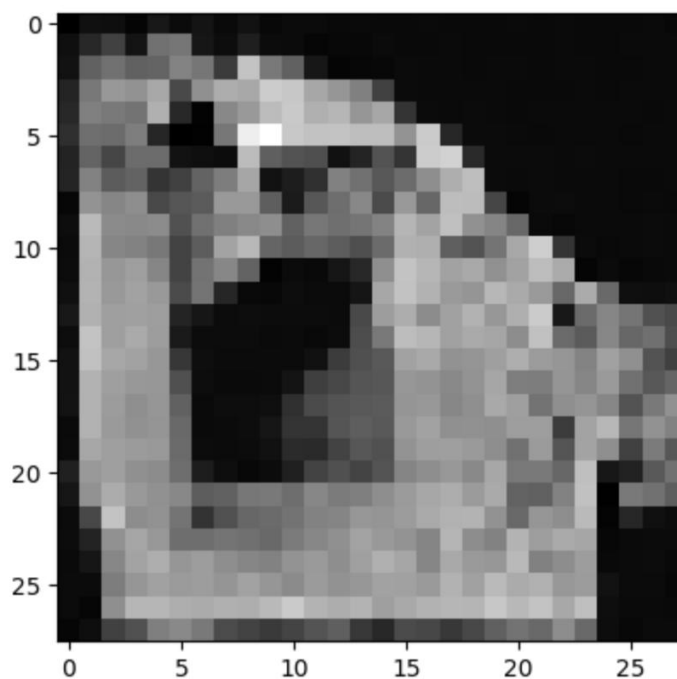


3) Click the **▶▶** button to run the sample, and then click the **Restart** button in the pop-up dialog box.



4) As shown in the following figure, after the execution of the program is completed, you can see in the notebook document that this model can produce a dress! :

```
<matplotlib.image.AxesImage at 0xe7fed58fa220>
```



5. 3. 13. Running ResNet50 Transfer Learning Sample

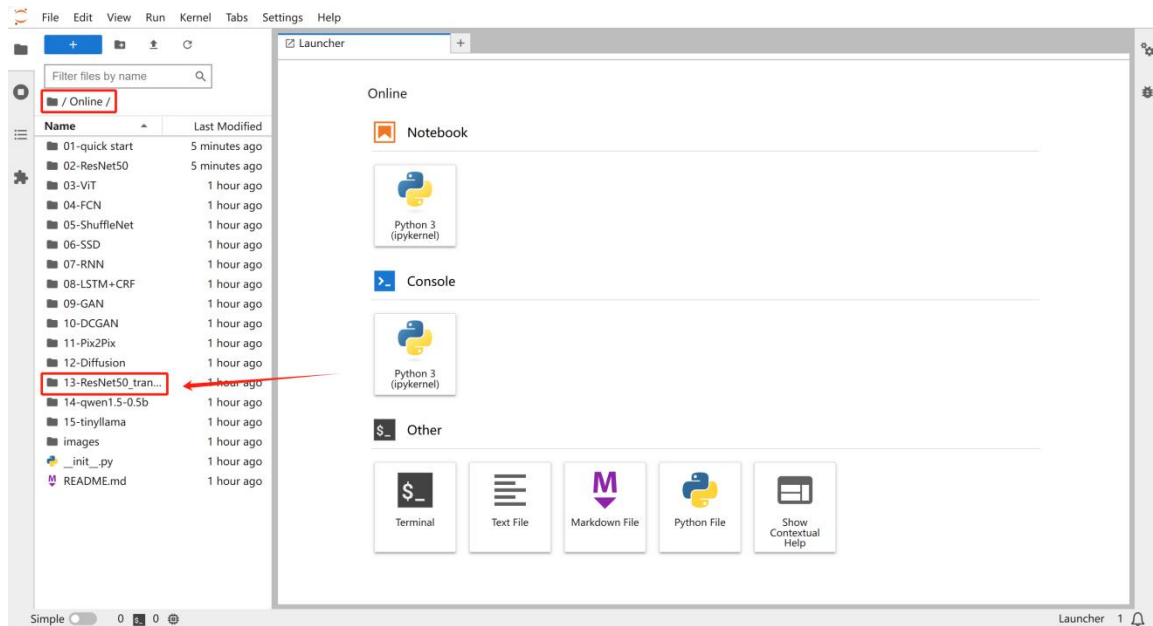
In practical application scenarios, due to insufficient training datasets, few people will start training the entire network from scratch. The common practice is to train a pre-trai



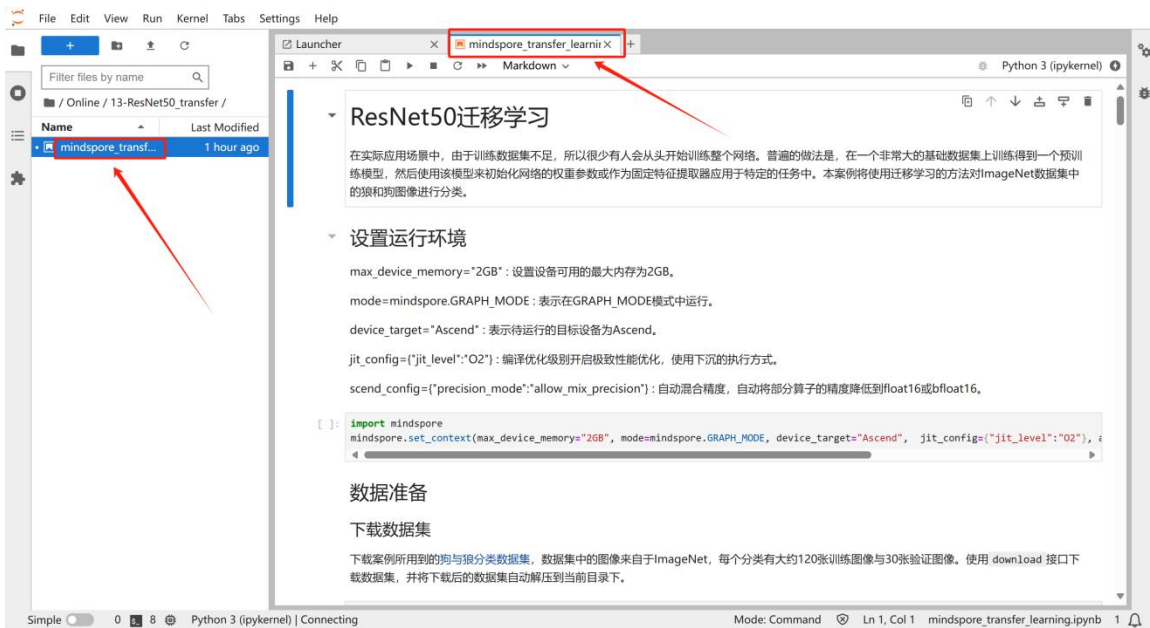
ned model on a very large base dataset, and then use this model to initialize the network's weight parameters or as a fixed feature extractor for specific tasks. This case study will use transfer learning methods to classify wolf and dog images in the ImageNet dataset.

We can run the sample in Jupyter Lab according to the following process.

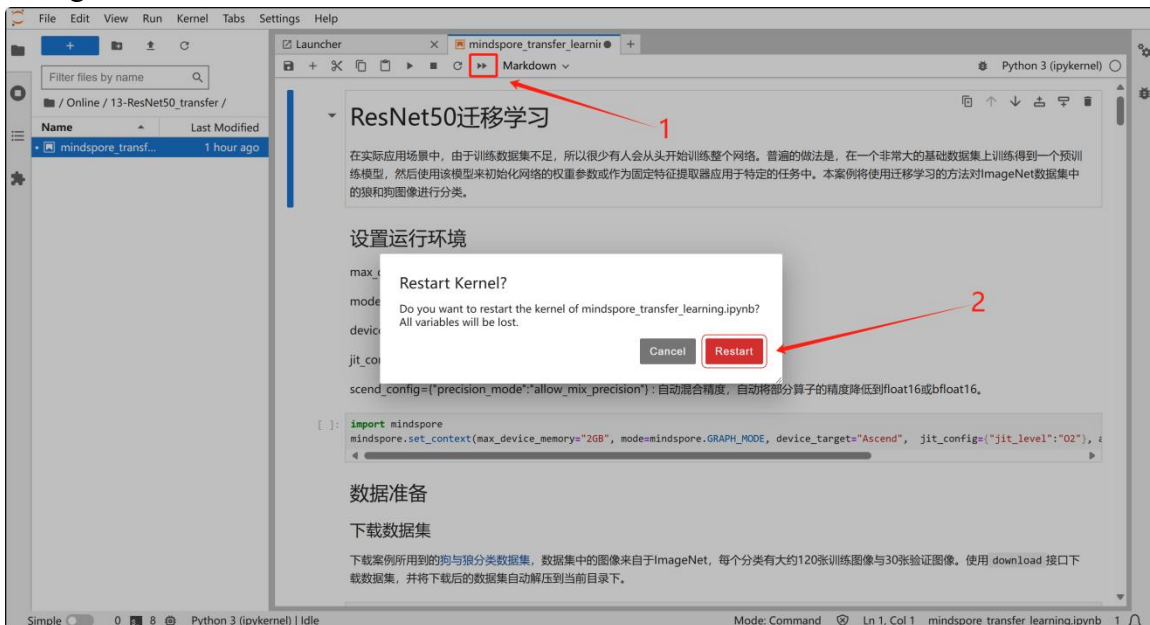
1) Firstly, double-click on **13-ResNet50_transfer** as shown in the following image in the Jupyter Lab interface to enter the directory of the sample.



2) In this directory, there are all the resources to run the example, among which **mindspore_transfer_learning.ipynb** is the file to run the example in Jupyter Lab. Double click to open **mindspore_transfer_learning.ipynb**, and the contents of this file will be displayed in the window on the right, as shown in the following figure:



3) Click  the button to run this sample, and then click the **Restart** button in the pop-up dialog box.



4) After the program is executed, the results of ResNet50 transfer learning can be successfully displayed in the notebook document. As shown in the following figure:



5. 4. Method for Running llm Large Language Model

Attention: Our current testing has found that the following dependency conditions must be met simultaneously in order for the inference program of the large model to run properly. We recommend using the latest released images directly, as we have already configured them. If you do not want to refresh the image, you can configure the environment according to the method written in the "orange pi mindspore/Online/14-qwen1.5-0.5b/Shengsi MindSpore Orange Pi qwen Chatbot Guidebook. md".

1. Ascend-cann-toolkit_8.0.RC3.alpha002
2. Ascend-cann-kernels-310b_8.0.RC3.alpha002
3. mindspore 2.4.0
4. mindnlp 0.4.0
5. gradio 4.44.0

The two major model cases are Case 14 and Case 15 under the "orange-pi-mindspore/Online" directory.



Note that the models of the current two LLM cases can only run on development boards with 24GB of memory. A development board with 12GB of memory may encounter errors due to memory limitations.

When running the model, please make sure to turn off the swap, otherwise there may be issues with thread synchronization failure leading to inability to run. Run the `sudo swapoff /swapfile` command to **close the swap partition on the system.**

5.4.1. qwen1.5-0.5b

- 1) Execute the following command to initiate inference.

```
(base) HwHiAiUser@orangepiaipro-20t:~/orange-pi-mindspore/Online/14-qwen1.5-0.5b
$ python qwen1.5-0.5b.py
```

- 2) The first startup will automatically download the model, and the specific time depends on the network environment. The model will be downloaded to the "`~/orange-pi-mindspore/Online/14-qwen1.5-0.5b/.mindnlp/model/Qwen/Qwen1.5-0.5B-Chat/`" folder.
- 3) By default, the inference code will check for relevant dependencies at startup. If the network environment is not good at this time, it may cause the code to fail to start. If it is not the first time to start and the model has already been downloaded, you can modify the startup code according to the following instructions to change the path to a local absolute path, so that you can start offline.

```
(base) HwHiAiUser@orangepiaipro-20t:~/orange-pi-mindspore/Online/14-qwen1.5-0.5b
$ vim qwen1.5-0.5b.py
```

Modify lines 8 and 9 in the code to look like the following: "`Qwen/Qwen1.5-0.5B-Chat`" is changed to "`/home/HwHiAiUser/orange-pi-mindspore-master/Online/14-qwen1.5-0.5b/.mindnlp/model/Qwen/Qwen1.5-0.5B-Chat`":

```
import gradio as gr
import mindspore
from mindnlp.transformers import AutoModelForCausalLM, AutoTokenizer
from mindnlp.transformers import TextIteratorStreamer
from threading import Thread

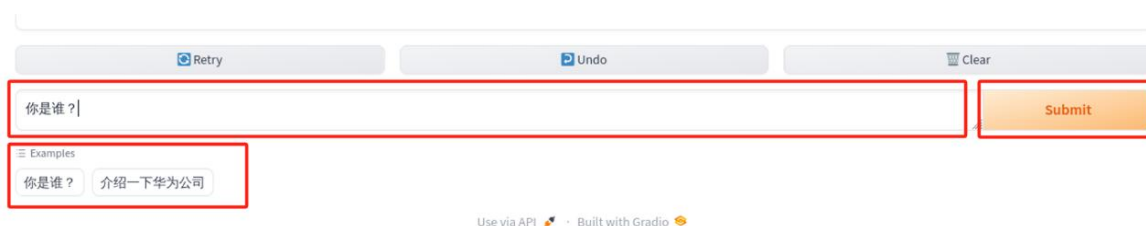
# Loading the tokenizer and model from Hugging Face's model hub.
tokenizer = AutoTokenizer.from_pretrained("/home/HwHiAiUser/orange-pi-mindspore-master/Online/14-qwen1.5-0.5b/.mindnlp/model/Qwen/Qwen1.5-0.5B-Chat", ms_dtype=mindspore.float16)
model = AutoModelForCausalLM.from_pretrained("/home/HwHiAiUser/orange-pi-mindspore-master/Online/14-qwen1.5-0.5b/.mindnlp/model/Qwen/Qwen1.5-0.5B-Chat", ms_dtype=mindspore.float16)
```

- 4) Wait for a while, an IP address will appear, copied to the address bar of the browser on the development board for access.



```
(base) HwHiAiUser@orangepiaipr:~/orange-pi-mindspore/llm/qwen1.5-0.5b$ python qwen1.5-0.5b.py
/usr/local/miniconda3/lib/python3.9/site-packages/numpy/core/getlimits.py:499: UserWarning: The value of the smallest subnormal for <class
'numpy.float64'> type is zero.
  setattr(self, word, getattr(machar, word).flat[0])
/usr/local/miniconda3/lib/python3.9/site-packages/numpy/core/getlimits.py:89: UserWarning: The value of the smallest subnormal for <class
'numpy.float64'> type is zero.
  return self._float_to_str(self.smallest_subnormal)
/usr/local/miniconda3/lib/python3.9/site-packages/numpy/core/getlimits.py:499: UserWarning: The value of the smallest subnormal for <class
'numpy.float32'> type is zero.
  setattr(self, word, getattr(machar, word).flat[0])
/usr/local/miniconda3/lib/python3.9/site-packages/numpy/core/getlimits.py:89: UserWarning: The value of the smallest subnormal for <class
'numpy.float32'> type is zero.
  return self._float_to_str(self.smallest_subnormal)
Building prefix dict from the default dictionary ...
Loading model from cache /tmp/jieba.cache
Loading model cost 3.457 seconds.
Prefix dict has been built successfully.
Qwen2ForCausalLM has generative capabilities, as 'prepare_inputs_for_generation' is explicitly overwritten. However, it doesn't directly in
herit from 'GenerationMixin'. 'PreTrainedModel' will NOT inherit from 'GenerationMixin', and this model will lose the ability to call 'gener
ate' and other related functions.
- If you are the owner of the model architecture code, please modify your model class such that it inherits from 'GenerationMixin' (after
'PreTrainedModel', otherwise you'll get an exception).
- If you are not the owner of the model architecture class, please contact the model code owner to update it.
Sliding Window Attention is enabled but not implemented for 'eager'; unexpected results may be encountered.
[WARNING] DEVICE(44662,e7ffec626020,python):2024-10-15-21:16:39.465.639 [mindspore/ccsrc/plugin/device/ascend/hal/device/ascend_memory_adap
ter.cc:117] Initialize] Free memory size is less than half of total memory size.Device 0 Device HBM total size:16367894528 Device HBM free
size:6314205184 may be other processes occupying this card, check as: ps -ef|grep python
/home/HwHiAiUser/.local/lib/python3.9/site-packages/gradio/analytcs.py:106: UserWarning: IMPORTANT: You are using gradio version 4.44.0, h
owever version 5.0.1 is available, please upgrade.
-----
warnings.warn(
Running on local URL: http://127.0.0.1:7860
To create a public link, set 'share=True' in 'launch()'.
```

- 5) After startup, you can enter any question in the message input box "Type a message..." at the bottom of the page, or click on the question set in the Examples below, and then click the "Submit" button on the right. The Qwen model will answer this.



- 6) The first answer will take a long time to load, approximately 1 minute. Please be patient and wait. The answer will be displayed in the chat box above.
- 7) If an error occurs, you can click the "retry" button to resend the previous message and have the model answer again; Click the 'undo' button to recall the previous message; Clicking the 'clear' button will clear the conversation in the chat box.





8) The output result is shown in the following figure:



5.4.2. Tinyllama

Note that currently, development boards with 12GB of memory are unable to run this case.

When running the model, please make sure to turn off the swap, otherwise there may be issues with thread synchronization failure leading to inability to run. Run the `sudo swapoff /swapfile` command to **close the swap partition on the system.**

1) Execute the following command to initiate inference.

```
(base) HwHiAiUser@orangepiaipro-20t:~/orange-pi-mindspore/Online/15-tinyllama$ python app.py
```

2) The first startup will automatically download the model, and the specific time depends on the network environment. The model will be downloaded to the folder "`~/orange-pi-mindspore/Online/15-tinyllama/.mindnlp/model/TinyLlama/TinyLlama-1.1B-Chat-v1.0/`".

3) By default, the inference code will check for relevant dependencies at startup. If the



network environment is not good at this time, it may cause the code to fail to start. If it is not the first time to start and the model has already been downloaded, you can modify the startup code according to the following instructions to change the path to a local absolute path, so that you can start offline.

```
(base) HwHiAiUser@orangepiaipro-20t:~/orange-pi-mindspore/Online/15-tinyllama$ vim ap.py
```

Modify lines 8 and 9 in the code to look like the following, which is to change "TinyLlama/TinyLlama-1.1B-Chat-v1.0" to "/home/HwHiAiUser/orange-pi-mindspore-master/Online/15-tinyllama/.mindnlp/model/TinyLlama/TinyLlama-1.1B-Chat-v1.0":

```
import gradio as gr
import mindspore
from mindnlp.transformers import AutoModelForCausalLM, AutoTokenizer
from mindnlp.transformers import StoppingCriteria, StoppingCriteriaList, TextIteratorStreamer
from threading import Thread

# Loading the tokenizer and model from Hugging Face's model hub.
tokenizer = AutoTokenizer.from_pretrained("/home/HwHiAiUser/orange-pi-mindspore-master/Online/15-tinyllama/.mindnlp/model/TinyLlama/TinyLlama-1.1B-Chat-v1.0")
model = AutoModelForCausalLM.from_pretrained("/home/HwHiAiUser/orange-pi-mindspore-master/Online/15-tinyllama/.mindnlp/model/TinyLlama/TinyLlama-1.1B-Chat-v1.0", ms_dtype=mindspore.float16)
print(model.dtype)
```

- 4) Wait for a while, an ip address will appear, copied to the address bar of the browser on the development board for access.

```
(base) HwHiAiUser@orangepiaipro:~/orange-pi-mindspore/llm/tinyllama$ python app.py
/usr/local/miniconda3/lib/python3.9/site-packages/numpy/core/getlimits.py:499: UserWarning: The value of the smallest subnormal for <class 'numpy.float64'> type is zero.
  setattr(self, word, getattr(machar, word).flat[0])
/usr/local/miniconda3/lib/python3.9/site-packages/numpy/core/getlimits.py:89: UserWarning: The value of the smallest subnormal for <class 'numpy.float64'> type is zero.
  return self._float_to_str(self.smallest_subnormal)
/usr/local/miniconda3/lib/python3.9/site-packages/numpy/core/getlimits.py:499: UserWarning: The value of the smallest subnormal for <class 'numpy.float32'> type is zero.
  setattr(self, word, getattr(machar, word).flat[0])
/usr/local/miniconda3/lib/python3.9/site-packages/numpy/core/getlimits.py:89: UserWarning: The value of the smallest subnormal for <class 'numpy.float32'> type is zero.
  return self._float_to_str(self.smallest_subnormal)
Building prefix dict from the default dictionary ...
Loading model from cache /tmp/jieba.cache
Loading model cost 3.461 seconds.
Prefix dict has been built successfully.
LlamaForCausalLM has generative capabilities, as 'prepare_inputs_for_generation' is explicitly overwritten. However, it doesn't directly inherit from 'GenerationMixin'. 'PreTrainedModel' will NOT inherit from 'GenerationMixin', and this model will lose the ability to call 'generate' and other related functions.
- If you are the owner of the model architecture code, please modify your model class such that it inherits from 'GenerationMixin' (after 'PreTrainedModel', otherwise you'll get an exception).
- If you are not the owner of the model architecture class, please contact the model code owner to update it.
[WARNING] DEVICE(S) [59752, e7ffe4046020, python]:2024-10-15-21:39:15.726.028 [mindspore/ccsrc/plugin/device/ascend/hal/device/ascend/memory_adapter.cc:117] Initialize] Free memory size is less than half of total memory size. Device 0 Device HBM total size:16367894528 Device HBM free size:5305196544 may be other processes occupying this card, check as: ps -ef|grep python
Float16
Running on local URL: http://127.0.0.1:7860
To create a public link, set 'share=True' in 'launch()'.
/home/HwHiAiUser/.local/lib/python3.9/site-packages/gradio/analytics.py:106: UserWarning: IMPORTANT: You are using gradio version 4.44.0, however version 5.0.1 is available, please upgrade.
  warnings.warn(
```

- 5) After startup, you can enter any question in the message input box "Type a message..." at the bottom of the page, or click on the question set in the Examples below, and then click the "Submit" button on the right. The Tinyllama model will answer it.



5. 4. 3. DeepSeek-R1-Distill-Qwen-1.5B

Note that currently, development boards with 12GB of memory are unable to run this case.

When running the model, please make sure to turn off the swap, otherwise there may be issues with thread synchronization failure leading to inability to run. Run the `sudo swapoff /swapfile` command to **close the swap partition on the system.**

- 6) Execute the following command to initiate inference.

```
(base) HwHiAiUser@orangepiaipro-20t:~/orange-pi-mindspore/Online/17-DeepSeek-R1-Distill-Qwen-1.5B$ python deepseek-r1-distill-qwen-1.5b.py
```

- 7) The first startup will automatically download the model, and the specific time depends on the network environment. The model will be downloaded to the "`~/orange-pi-mindspore/Online/17-DeepSeek-R1-Distill-Qwen-1.5B/.mindnlp/model/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B`" folder.
- 8) By default, the inference code will check for relevant dependencies at startup. If the network environment is not good at this time, it may cause the code to fail to start. If it is not the first time to start and the model has already been downloaded, you can modify the startup code according to the following instructions to change the path to a local absolute path, so that you can start offline.

```
(base) HwHiAiUser@orangepiaipro-20t:~/orange-pi-mindspore/Online/17-DeepSeek-R1-Distill-Qwen-1.5B$ vim deepseek-r1-distill-qwen-1.5b.py
```

Modify lines 8 and 9 in the code to look like the following, which is to change "`deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B`" to "`/home/HwHiAiUser/orange-pi-mindspore-master/Online/17-DeepSeek-R1-Distill-Qwen-1.5B/.mindnlp/model/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B`":



```
import gradio as gr
import mindspore
from mindnlp.transformers import AutoModelForCausalLM, AutoTokenizer
from mindnlp.transformers import TextIteratorStreamer
from threading import Thread

# Loading the tokenizer and model from Hugging Face's model hub.
tokenizer = AutoTokenizer.from_pretrained("/home/HwHiAiUser/orange-pi-mindspore-master/Online/17-DeepSeek-R1-Distill-Qwen-1.5B/.mindnlp/model/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B", ms_dtype=mindspore.float16)
model = AutoModelForCausalLM.from_pretrained("/home/HwHiAiUser/orange-pi-mindspore-master/Online/17-DeepSeek-R1-Distill-Qwen-1.5B/.mindnlp/model/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B", ms_dtype=mindspore.float16)
```

- 9) Wait for a while, an ip address will appear, copied to the address bar of the browser on the development board for access.

```
- If you are the owner of the model architecture code, please modify your model class such that it inherits from `GenerationMixin` (after `PreTrainedModel`, otherwise you'll get an exception).
- If you are not the owner of the model architecture class, please contact the model code owner to update it.
Sliding Window Attention is enabled but not implemented for `eager`; unexpected results may be encountered.
[WARNING] DEVICE(17752,e7ffde089020,python3):2025-02-10-20:13:29.684.892 [mindspore/ccsrc/plugin/device/ascend/hal/device/ascend_memory_adapter.cc:117] Initialize] Free memory size is less than half of total memory size. Device 0 Device HBM total size:24823529472 Device HBM free size:12277534720 may be other processes occupying this card, check as: ps -ef|grep python
/home/HwHiAiUser/.local/lib/python3.9/site-packages/gradio/analytics.py:106: UserWarning: IMPORTANT: You are using gradio version 4.44.0, however version 4.44.1 is available, please upgrade.
-----
warnings.warn(
Running on local URL: http://127.0.0.1:7860
To create a public link, set `share=True` in `launch()`.
```

- 10) After startup, you can enter any question in the message input box "Type a message..." at the bottom of the page, or click on the question set in the Examples below, and then click the "Submit" button on the right. The DeepSeek-R1-Distill-Qwen-1.5B model will answer this question.后，可在页面下方消息输入框“Type a message...”

- 11) The output result is as follows:

DeepSeek-R1-Distill-Qwen-1.5B

问几个问题

Chatbot

你是谁?

您好！我是由中国的深度求索（DeepSeek）公司开发的智能助手DeepSeek-R1。如您有任何任何问题或需要帮助，我会尽我所能为您提供帮助。



5. 5. Methods for Running Offline Reasoning Cases

Please refer to the following link in the infer folder or Github of the built-in cases on the development board for details:

<https://github.com/mindspore-courses/orange-pi-mindspore/tree/master/Offline>



6. Installation of AI Application Environment (OpenHarmony)

6.1. Installation of inference environment

- 1) 从[开发板的资料下载页面](#)下载 Download the Ascend310B-OpenHarmony-CANN. zip compressed file from [the download page of the development board](#)
- 2) Connect the TF card with OpenHarmony system to a Linux PC through the card reader

- 3) By using lsblk, you can see the partitions and their sizes

```
(base) root@orangepiaipro-20t:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sdx          8:0  1 59.5G  0 disk
├─sdx1       8:1  1 100M  0 part
├─sdx2       8:2  1 39.7G  0 part
├─sdx3       8:3  1 100M  0 part
├─sdx4       8:4  1   3G   0 part
├─sdx5       8:5  1  9.8G  0 part
├─sdx6       8:6  1 512M  0 part
└─sdx7       8:7  1 512M  0 part
```

- 4) Mount directory. If Linux has automatically mounted sdx2, this step can be skipped

```
mount -o rw /dev/sdx2 /home/test
```

- 5) Extract the file, and the size of the data1 folder after decompression is about 12GB. Wait for synchronization to complete before removing the mount.

```
unzip Ascend310B-OpenHarmony-CANN.zip -d /home/test/.
sync
```

- 6) Unmount

```
umount /dev/sdx2
```




- 7) Start the AI PRO 20T OpenHarmony system using a TF card, refer to the [startup step of the OpenHarmony system development board](#), wait for init.sh to complete execution and restart before proceeding with the following steps.
- 8) After starting OpenHarmony, enter the following command to initialize the environment

```
cd /data1/
mount -o rw,remount /
chmod 777 *.sh
bash initmodel.sh
```

If initmodel.sh fails to execute, mask the first line devmem 0xc42e0000 8 0x0 and retry 果 initmodel.sh

- 9) Install runtime

```
cd /data1/nnrt/run_package/runtime
./runtime/scripts/install.sh -- --full
```

```
# vi ./run_runtime_upgrade.sh
# ./data1/nnrt/run_package/runtime/runtime/scripts/install.sh -- --full
/system/bin/sh: ./data1/nnrt/run_package/runtime/runtime/scripts/install.sh: inaccessible or not found
age/runtime/runtime/scripts/install.sh -- --full
[Runtime] [1970-01-01 08:22:50] [INFO]: Start time:1970-01-01 08:22:50
[Runtime] [1970-01-01 08:22:50] [INFO]: LogFile:/var/log/ascend_seclog/ascend_install.log
[Runtime] [1970-01-01 08:22:50] [INFO]: InputParams:--full
[Runtime] [1970-01-01 08:22:50] [INFO]: OperationLogFile:/var/log/ascend_seclog/operation.log
[Runtime] [1970-01-01 08:22:50] [INFO]: Check version matched!
[Runtime] [1970-01-01 08:22:50] [INFO]: base version is none.
[Runtime] [1970-01-01 08:22:50] [INFO]: parent_dir value is [/usr/local] and children_dir value is [Ascend]
[Runtime] [1970-01-01 08:22:50] [INFO]: [Ascend] belongs to root.
[Runtime] [1970-01-01 08:22:50] [INFO]: [Ascend] permission is ok.
[Runtime] [1970-01-01 08:22:51] [INFO]: parent_dir value is [/usr] and children_dir value is [local]
[Runtime] [1970-01-01 08:22:51] [INFO]: [local] belongs to root.
[Runtime] [1970-01-01 08:22:51] [INFO]: [local] permission is ok.
[Runtime] [1970-01-01 08:22:51] [INFO]: parent_dir value is [/] and children_dir value is [usr]
[Runtime] [1970-01-01 08:22:51] [INFO]: [usr] belongs to root.
[Runtime] [1970-01-01 08:22:51] [INFO]: [usr] permission is ok.
[Runtime] [1970-01-01 08:22:51] [INFO]: parent_dir value is [/] and children_dir value is [/]
[Runtime] [1970-01-01 08:22:51] [INFO]: parent_dir permission check succeeded
[Runtime] [1970-01-01 08:22:51] [INFO]: Runtime package has been installed on the path /usr/local/Ascend, the version is none, and the version of this package is 7.
1.0.3.220, do you want to continue? [y/n]
y
[Runtime] [1970-01-01 08:22:54] [WARNING]: run runtime uninstall.sh not found.
[Runtime] [1970-01-01 08:22:54] [INFO]: Some files generated by user are not cleared, if necessary, manually clear them, get details in /var/log/ascend_seclog/ascend_install.log
[Runtime] [1970-01-01 08:22:55] [INFO]: install /usr/local/Ascend full
[Runtime] [1970-01-01 08:22:55] [INFO]: step into run runtime install.sh .....
[Runtime] [1970-01-01 08:22:55] [INFO]: install target dir /usr/local/Ascend/7.0.0, type full.
[Runtime] [1970-01-01 08:22:55] [INFO]: no need to install runtime files.
[Runtime] [1970-01-01 08:22:55] [INFO]: runtime install upgradePercentage:100%
[Runtime] [1970-01-01 08:22:55] [INFO]: Base version set successfully!
[Runtime] [1970-01-01 08:22:55] [INFO]: Runtime package installed successfully! The new version takes effect immediately.
Please make sure that
- LD_LIBRARY_PATH includes /usr/local/Ascend/7.0.0/runtime/lib64
[Runtime] [1970-01-01 08:22:55] [INFO]: End time:1970-01-01 08:22:55
```

- 10) Install pyacl;

```
cd /data1/nnrt/run_package/pyacl
./script/install.sh -- --full
```

```
# cd pyacl/
# ls
python script
# ./
python/ script/
# ./script/install.sh -- --full
[payACL] [1970-01-01 08:04:13] [INFO] install start
[payACL] [1970-01-01 08:04:13] [INFO] delete file /usr/local/Ascend/latest/pyACL successfully
[payACL] [1970-01-01 08:04:13] [INFO] delete file /usr/local/Ascend/latest/python/site-packages/acl.so successfully
[payACL] [1970-01-01 08:04:13] [INFO] delete file /usr/local/Ascend/latest/python/site-packages/acl/acl.so successfully
Please make sure that
- PYTHONPATH includes /usr/local/Ascend/7.0.0/python
[payACL] [1970-01-01 08:04:13] [INFO] Ascend-pyACL-7.0.0 install success
```



11) Install test;

```
cd /data1/nnrt/run_package/test
./script/install.sh -- --full --install-path=/usr/local/Ascend/
```

```
# cd /data1/nnrt/run_package/test
# ./script/install.sh -- --full --install-path=/usr/local/Ascend/
--install-path=/usr/local/Ascend/
[test-ops] [1970-01-01 08:02:52] [INFO]: new_version info 7.0.0
[test-ops] [1970-01-01 08:02:53] [INFO]: your install path is /usr/local/Ascend//7.0.0/opp/test-ops
[test-ops] [1970-01-01 08:02:53] [INFO]: install is success
[test-ops] [1970-01-01 08:02:53] [INFO]: re-install successfully
#
```

12) Install aicpu

```
cd /data1/nnrt/run_package/aicpu
./aicpu/script/install.sh -- --full
```

```
# ./aicpu/script/install.sh -- --full
[Aicpu] [1970-01-01 10:21:38] [INFO]: Start time: 1970-01-01 10:21:38
[Aicpu] [1970-01-01 10:21:38] [INFO]: LogFile: /var/log/ascend_seclog/ascend_install.log
[Aicpu] [1970-01-01 10:21:38] [INFO]: OperationLogFile: /var/log/ascend_seclog/operation.log
[Aicpu] [1970-01-01 10:21:38] [WARNING]: ascend_install.info not exist
[Aicpu] [1970-01-01 10:21:38] [INFO]: do you want to continue installing? [y/n]
y
[Aicpu] [1970-01-01 10:21:46] [INFO]: Some files generated by user are not cleared, if necessary, manually clear them, get details in /var/log/ascend_seclog/ascend_install.log
[Aicpu] [1970-01-01 10:21:46] [INFO]: upgradePercentage: 20%
[Aicpu] [1970-01-01 10:21:54] [INFO]: upgradePercentage: 50%
[Aicpu] [1970-01-01 10:21:55] [INFO]: upgradePercentage: 100%
[Aicpu] [1970-01-01 10:21:55] [INFO]: Aicpu_kernels package installed successfully! The new version takes effect immediately.
[Aicpu] [1970-01-01 10:21:55] [INFO]: Using requirements: when aicpu module install finished or before you run the aicpu module, execute the command
[ export ASCEND_AICPU_PATH=/usr/local/Ascend/7.0.0 ] to set the environment path.
[ export ASCEND_OPP_PATH=/usr/local/Ascend/7.0.0/opp ] to set the environment path.
[Aicpu] [1970-01-01 10:21:55] [INFO]: End time: 1970-01-01 10:21:55
```

13) Install aicpu310

```
cd /data1/nnrt/run_package/aicpu310
./aicpu/script/install.sh -- --full
```

```
# cd aicpu310
# ./aicpu/script/install.sh -- --full
[Aicpu] [1970-01-01 09:40:20] [INFO]: Start time: 1970-01-01 09:40:20
[Aicpu] [1970-01-01 09:40:20] [INFO]: LogFile: /var/log/ascend_seclog/ascend_install.log
[Aicpu] [1970-01-01 09:40:20] [INFO]: OperationLogFile: /var/log/ascend_seclog/operation.log
[Aicpu] [1970-01-01 09:40:21] [WARNING]: driver version.info not exist
[Aicpu] [1970-01-01 09:40:21] [INFO]: do you want to continue installing? [y/n]
y
[Aicpu] [1970-01-01 09:40:22] [INFO]: Some files generated by user are not cleared, if necessary, manually clear them, get details in /var/log/ascend_seclog/ascend_install.log
[Aicpu] [1970-01-01 09:40:23] [INFO]: upgradePercentage: 20%
[Aicpu] [1970-01-01 09:40:29] [INFO]: upgradePercentage: 50%
[Aicpu] [1970-01-01 09:40:31] [INFO]: upgradePercentage: 100%
[Aicpu] [1970-01-01 09:40:31] [INFO]: Aicpu_kernels package installed successfully! The new version takes effect immediately.
[Aicpu] [1970-01-01 09:40:31] [INFO]: Using requirements: when aicpu module install finished or before you run the aicpu module, execute the command
[ export ASCEND_AICPU_PATH=/usr/local/Ascend/7.0.0 ] to set the environment path.
[ export ASCEND_OPP_PATH=/usr/local/Ascend/7.0.0/opp ] to set the environment path.
[Aicpu] [1970-01-01 09:40:31] [INFO]: End time: 1970-01-01 09:40:31
#
```

14) Install aicpu310mini

```
cd /data1/nnrt/run_package/aicpu310mini
./aicpu/script/install.sh -- --full
```



```
# cd aicpu310mini/
# ./aicpu/script/install.sh -- -- --full
[Aicpu] [1970-01-01 08:01:05] [INFO]: Start time: 1970-01-01 08:01:05
[Aicpu] [1970-01-01 08:01:05] [INFO]: LogFile: /var/log/ascend_seclog/ascend_install.log
[Aicpu] [1970-01-01 08:01:05] [INFO]: OperationLogFile: /var/log/ascend_seclog/operation.log
[Aicpu] [1970-01-01 08:01:06] [WARNING]: driver version.info not exist
[Aicpu] [1970-01-01 08:01:06] [INFO]: do you want to continue installing? [y/n]
y
[Aicpu] [1970-01-01 08:01:07] [INFO]: Some files generated by user are not cleared, if necessary, manually clear them, get details in /var/log/ascend_seclog/ascend_install.log
[Aicpu] [1970-01-01 08:01:08] [INFO]: upgradePercentage: 20%
[Aicpu] [1970-01-01 08:01:14] [INFO]: upgradePercentage: 50%
[Aicpu] [1970-01-01 08:01:16] [INFO]: upgradePercentage: 100%
[Aicpu] [1970-01-01 08:01:16] [INFO]: Aicpu_kernels package installed successfully! The new version takes effect immediately.
[Aicpu] [1970-01-01 08:01:16] [INFO]: Using requirements: when aicpu module install finished or before you run the aicpu module, execute the command
[ export ASCEND_AICPU_PATH=/usr/local/Ascend/7.0.0 ] to set the environment path.
[ export ASCEND_OPP_PATH=/usr/local/Ascend/7.0.0/opp ] to set the environment path.
[Aicpu] [1970-01-01 08:01:16] [INFO]: End time: 1970-01-01 08:01:16
```

15) Install aicpu310p

```
cd /data1/nnrt/run_package/aicpu310p
```

```
./aicpu/script/install.sh -- -- --full
```

```
# cd aicpu310p
# ./aicpu/script/install.sh -- -- --full
[Aicpu] [1970-01-01 09:41:14] [INFO]: Start time: 1970-01-01 09:41:14
[Aicpu] [1970-01-01 09:41:14] [INFO]: LogFile: /var/log/ascend_seclog/ascend_install.log
[Aicpu] [1970-01-01 09:41:14] [INFO]: OperationLogFile: /var/log/ascend_seclog/operation.log
[Aicpu] [1970-01-01 09:41:15] [WARNING]: driver version.info not exist
[Aicpu] [1970-01-01 09:41:15] [INFO]: do you want to continue installing? [y/n]
y
[Aicpu] [1970-01-01 09:41:17] [INFO]: Some files generated by user are not cleared, if necessary, manually clear them, get details in /var/log/ascend_seclog/ascend_install.log
[Aicpu] [1970-01-01 09:41:18] [INFO]: upgradePercentage: 20%
[Aicpu] [1970-01-01 09:41:23] [INFO]: upgradePercentage: 50%
[Aicpu] [1970-01-01 09:41:24] [INFO]: upgradePercentage: 100%
[Aicpu] [1970-01-01 09:41:24] [INFO]: Aicpu_kernels package installed successfully! The new version takes effect immediately.
[Aicpu] [1970-01-01 09:41:24] [INFO]: Using requirements: when aicpu module install finished or before you run the aicpu module, execute the command
[ export ASCEND_AICPU_PATH=/usr/local/Ascend/7.0.0 ] to set the environment path.
[ export ASCEND_OPP_PATH=/usr/local/Ascend/7.0.0/opp ] to set the environment path.
[Aicpu] [1970-01-01 09:41:25] [INFO]: End time: 1970-01-01 09:41:25
```

16) View NPU status

```
export LD_LIBRARY_PATH=/usr/local/Ascend/latest/aarch64-linux/lib64:/libaarch64/aarch64-linux-gnu:/usr/local/Ascend/driver/lib64:/usr/lib64/var/dmp_daemon -I -M -U 8087 &
```

```
/usr/local/sbin/npu-smi info
```

```
# /usr/local/sbin/npu-smi info
[ERROR] DRV(2485,npu-smi):1970-01-01-01:34:39.830.857 [dsmi_common.c:243][dmp] [dsmi_check_out_valid 243] recv msg data error code 65534, recv msg data opcode 0x696, opcode 0x696.
[ERROR] DRV(2485,npu-smi):1970-01-01-01:34:39.831.006 [dsmi_common.c:666][dmp] [dsmi_send_msg_rec_res 666] call dsmi_check_out_valid error:65534.
[ERROR] DRV(2485,npu-smi):1970-01-01-01:34:39.831.030 [dsmi_product_ext.c:750][dmp] [dsmi_cmd_get_memory_info 750] dev(0) dsmi_send_msg_rec_res failed, ret = 65534.
[ERROR] DRV(2485,npu-smi):1970-01-01-01:34:39.831.049 [dsmi_product_ext.c:766][dmp] [dsmi_get_memory_info_v2 766] Dsmi_cmd_get_memory_info call error.(dev_id=0; ret =65534)
[ERROR] DRV(2485,npu-smi):1970-01-01-01:34:39.897.810 [dsmi_common.c:243][dmp] [dsmi_check_out_valid 243] recv msg data error code 65534, recv msg data opcode 0x696, opcode 0x696.
[ERROR] DRV(2485,npu-smi):1970-01-01-01:34:39.897.878 [dsmi_common.c:666][dmp] [dsmi_send_msg_rec_res 666] call dsmi_check_out_valid error:65534.
[ERROR] DRV(2485,npu-smi):1970-01-01-01:34:39.897.901 [dsmi_product_ext.c:750][dmp] [dsmi_cmd_get_memory_info 750] dev(0) dsmi_send_msg_rec_res failed, ret = 65534.
[ERROR] DRV(2485,npu-smi):1970-01-01-01:34:39.897.921 [dsmi_product_ext.c:766][dmp] [dsmi_get_memory_info_v2 766] Dsmi_cmd_get_memory_info call error.(dev_id=0; ret =65534)
-----
npu-smi 23.0.0                               Version: 23.0.0
-----
| NPU   Name      | Health | Power(W) | Temp(C) | Memory-Usage(MB) | Hugepages-Usage(page) |
| Chpt  Device    | Bus-Id | AICore(%) |           |                   |                       |
-----
| 0     310B4     | 0       | 0.0      | 78      | 0 / 0             | 0 / 0                 |
| 0     0         | NA      | 0        | 0       | 0 / 0             | 0 / 0                 |
-----
```

17) Run reasoning demo

```
cd /data1/
```

```
bash runmodel.sh
```

The execution result is as follows:

It can be seen that the probability of classifying the inference as beagle is 0.905956.



```

[EVENT] CECPU(3505,main):1970-01-01-02:07:08.285.978 [ae_so_manager.cc:228][getApi][tid:3507][AICPU_PROCESSER] Get api DvppGetVersion from so libdvpp_kernels.so
success.
[EVENT] AICPU(3505,main):1970-01-01-02:07:08.289.013 [dvpp_get_version_kernels.cc:45][DoCompute][tid:3507][DVPP_KERNELS] Kernel:DvppGetVersion get version success,
version=300002.
[EVENT] CECPU(3505,main):1970-01-01-02:07:08.292.253 [ae_so_manager.cc:228][getApi][tid:3507][AICPU_PROCESSER] Get api DvppCreateChannel from so libdvpp_kernels.so
success.
[EVENT] GE(3505,main):1970-01-01-02:07:09.467.235 [davinci_model.cc:412]3505 InitFeatureMapAndP2PMem:[IMAS]InitFeatureMapAndP2PMem graph_0 MallocMemory type[F] mem
ddr[0x80004600200] mem_size[3696940]
[EVENT] RUNTIME(3505,main):1970-01-01-02:07:09.487.233 [logger.cc:1046] 3505 ModelBindStream: model_id=0, stream_id=3, flag=0.
[INFO] top 1: index[162] value[0.905956] class[beagle]
[INFO] top 2: index[161] value[0.092549] class[bassetbasset hound]
[INFO] top 3: index[169] value[0.000758] class[walker houndwalker foxhound]
[INFO] top 4: index[167] value[0.000592] class[English foxhound]
[INFO] top 5: index[163] value[0.000076] class[bloodhound sleuthhound]
[EVENT] CECPU(3505,main):1970-01-01-02:07:09.507.970 [ae_so_manager.cc:220][getApi][tid:3507][AICPU_PROCESSER] Get api DvppDestroyChannel from so libdvpp_kernels.
so success.

```

6.2. Python Environment Installation

1) Install python package

```

cp -r /data1/Python /usr/local/
source /usr/local/Python/set_python_env.sh
cd /usr/local/Python/bin
python3.10 -m ensurepip --upgrade
pip3 --version

```

2) Installing an online python library requires offline operation.

Taking mindspore as an example.

```

source /usr/local/Python/set_python_env.sh
pip3 install mindspore

```

```

# pip3 install mindspore
Collecting mindspore
  Using cached mindspore-2.5.0-cp310-none-any.whl (345.0 MB)
Requirement already satisfied: psutil<=5.6.1 in /usr/local/Python/lib/python3.10/site-packages (from mindspore) (6.1.0)
Requirement already satisfied: protobuf<=3.13.0 in /usr/local/Python/lib/python3.10/site-packages (from mindspore) (6.31.0)
Requirement already satisfied: numpy<2.0.0,>=1.20.0 in /usr/local/Python/lib/python3.10/site-packages (from mindspore) (1.26.4)
Requirement already satisfied: astunparse<=1.6.3 in /usr/local/Python/lib/python3.10/site-packages (from mindspore) (1.6.3)
Requirement already satisfied: dill<=0.3.7 in /usr/local/Python/lib/python3.10/site-packages (from mindspore) (0.4.0)
Requirement already satisfied: packaging<=20.0 in /usr/local/Python/lib/python3.10/site-packages (from mindspore) (25.0)
Requirement already satisfied: pillow<=6.2.0 in /usr/local/Python/lib/python3.10/site-packages (from mindspore) (11.2.1)
Requirement already satisfied: safetensors<=2.0.4 in /usr/local/Python/lib/python3.10/site-packages (from mindspore) (0.5.3)
Requirement already satisfied: asttokens<=2.0.4 in /usr/local/Python/lib/python3.10/site-packages (from mindspore) (3.0.0)
Requirement already satisfied: scipy<=1.5.4 in /usr/local/Python/lib/python3.10/site-packages (from mindspore) (1.15.3)
Requirement already satisfied: six<2.0,>=1.6.1 in /usr/local/Python/lib/python3.10/site-packages (from astunparse<=1.6.3->mindspore) (1.16.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/Python/lib/python3.10/site-packages (from astunparse<=1.6.3->mindspore) (0.45.1)
Installing collected packages: mindspore
Successfully installed mindspore-2.5.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended
that you use virtual environments: https://pip.pypa.io/warnings/venv

```

6.3. Installing the toolkit package

1) Install the toolkit

```

source /usr/local/Python/set_python_env.sh
cd /data1/toolkit/run_package/toolkit
./toolkit/scripts/install.sh -- --full

```




```

# cd toolkit/
# ls
toolkit  tools  version.info
# pwd
/data1/toolkit/run_package/toolkit
# /data1/toolkit/run_package/toolkit^c
# ls
toolkit  tools  version.info
# ./to
toolkit/  tools/
# ./toolkit/scripts/in
install.sh
install_common_parser.sh
install_msprof_fitter.sh
install_profiling_msprof.sh
./toolkit/scripts/install.sh -- -- --full
[Toolkit] [1970-01-01 13:40:53] [INFO]: Start Time: 1970-01-01 13:40:53
[Toolkit] [1970-01-01 13:40:53] [INFO]: LogFile: /var/log/ascend_seclog/ascend_install.log
[Toolkit] [1970-01-01 13:40:53] [INFO]: OperationLogFile: /var/log/ascend_seclog/operation.log
[Toolkit] [1970-01-01 13:40:53] [INFO]: InputParams: --full
[Toolkit] [1970-01-01 13:40:54] [INFO]: version compatibility check successfully!
[Toolkit] [1970-01-01 13:40:54] [INFO]: base version is 7.1.0.3.220.
[Toolkit] [1970-01-01 13:40:54] [INFO]: Toolkit package has been installed on the path /usr/local/Ascend/7.0.0, the version is 7.1.0.3.220, and the version of this package is 7.1.0.3.220, do you want to continue? [y/n]
y
[Toolkit] [1970-01-01 13:40:57] [INFO]: [Profiler]: Uninstall profiling successfully.
[Toolkit] [1970-01-01 13:40:55] [INFO]: Toolkit package uninstalled successfully! Uninstallation takes effect immediately.
[Toolkit] [1970-01-01 13:54:36] [INFO]: [Profiler]: Install profiling successfully.
[Toolkit] [1970-01-01 13:54:36] [INFO]: upgradePercentage: 100%
[Toolkit] [1970-01-01 13:54:36] [INFO]: InstallPath: /usr/local/Ascend/7.0.0
[Toolkit] [1970-01-01 13:54:36] [INFO]: Toolkit package installed successfully! The new version takes effect immediately.
Please make sure that
- TOOLCHAIN_HOME set with /usr/local/Ascend/7.0.0/toolkit
[Toolkit] [1970-01-01 13:54:36] [INFO]: End Time: 1970-01-01 13:54:36
#

```

2) Install runtime

```
cd /data1/toolkit/run_package/runtime
```

```
./runtime/scripts/install.sh -- -- --full
```

```

# ls
runtime  version.info
# ./runtime/scripts/install.sh -- -- --full
[Runtime] [1970-01-01 14:18:34] [INFO]: Start time:1970-01-01 14:18:34
[Runtime] [1970-01-01 14:18:34] [INFO]: LogFile:/var/log/ascend_seclog/ascend_install.log
[Runtime] [1970-01-01 14:18:34] [INFO]: InputParams:--full
[Runtime] [1970-01-01 14:18:34] [INFO]: OperationLogFile:/var/log/ascend_seclog/operation.log
[Runtime] [1970-01-01 14:18:34] [INFO]: Check version matched!
[Runtime] [1970-01-01 14:18:34] [INFO]: base version is 7.1.0.3.220.
[Runtime] [1970-01-01 14:18:34] [INFO]: parent_dir value is [/usr/local] and children_dir value is [Ascend]
[Runtime] [1970-01-01 14:18:34] [INFO]: [Ascend] belongs to root.
[Runtime] [1970-01-01 14:18:34] [INFO]: [Ascend] permission is ok.
[Runtime] [1970-01-01 14:18:34] [INFO]: parent_dir value is [/usr] and children_dir value is [local]
[Runtime] [1970-01-01 14:18:34] [INFO]: [local] belongs to root.
[Runtime] [1970-01-01 14:18:35] [INFO]: [local] permission is ok.
[Runtime] [1970-01-01 14:18:35] [INFO]: parent_dir value is [/] and children_dir value is [usr]
[Runtime] [1970-01-01 14:18:35] [INFO]: [usr] belongs to root.
[Runtime] [1970-01-01 14:18:35] [INFO]: [usr] permission is ok.
[Runtime] [1970-01-01 14:18:35] [INFO]: parent_dir value is [/] and children_dir value is [/]
[Runtime] [1970-01-01 14:18:35] [INFO]: parent_dirs.permission check succeeded
[Runtime] [1970-01-01 14:18:35] [INFO]: Runtime package has been installed on the path /usr/local/Ascend, the version is 7.1.0.3.220, and the version of this package is 7.1.0.3.220, do you want to continue? [y/n]
y
[Runtime] [1970-01-01 14:18:37] [INFO]: uninstall /usr/local/Ascend full
[Runtime] [1970-01-01 14:18:37] [INFO]: step into run_runtime_uninstall.sh .....
[Runtime] [1970-01-01 14:18:37] [INFO]: uninstall target dir /usr/local/Ascend/7.0.0, type full.
xargs: Unknown option 'no-run-if-empty' (see 'xargs --help')
ls: writer: Broken pipe
ln: cannot create symbolic link from 'linux/aarch64/libgraph.so' to 'libgraph.so': File exists
xargs: Unknown option 'no-run-if-empty' (see 'xargs --help')
xargs: Unknown option 'no-run-if-empty' (see 'xargs --help')
ln: cannot create symbolic link from 'linux/aarch64/libgraph.so' to 'libgraph.so': File exists
[Runtime] [1970-01-01 14:19:47] [INFO]: Runtime package uninstalled successfully! Uninstallation takes effect immediately.
[Runtime] [1970-01-01 14:19:47] [INFO]: Some files generated by user are not cleared, if necessary, manually clear them, get details in /var/log/ascend_seclog/ascend_install.log
[Runtime] [1970-01-01 14:19:47] [INFO]: install /usr/local/Ascend full
[Runtime] [1970-01-01 14:19:47] [INFO]: step into run_runtime_install.sh .....
[Runtime] [1970-01-01 14:19:47] [INFO]: install target dir /usr/local/Ascend/7.0.0, type full.
[Runtime] [1970-01-01 14:19:47] [INFO]: runtime install upgradePercentage:10%
[Runtime] [1970-01-01 14:21:35] [INFO]: runtime install upgradePercentage:100%
[Runtime] [1970-01-01 14:21:35] [INFO]: Upgrade base version successfully!
[Runtime] [1970-01-01 14:21:35] [INFO]: Runtime package installed successfully! The new version takes effect immediately.
Please make sure that
- LD_LIBRARY_PATH includes /usr/local/Ascend/7.0.0/runtime/lib64
[Runtime] [1970-01-01 14:21:35] [INFO]: End time:1970-01-01 14:21:35
#

```

3) Install opp

```
cd /data1/toolkit/run_package/opp
```

```
./opp/script/install.sh -- -- --full
```



```

install.sh install_common_parser.sh
# ./opp/script/install.sh -- --full
[Opp] [1970-01-01 14:30:09] [INFO]: Start Time: 1970-01-01 14:30:09
[Opp] [1970-01-01 14:30:09] [INFO]: Uninstall the version before multi version of opp successfully!
[Opp] [1970-01-01 14:30:09] [INFO]: Execute the opp run package.
[Opp] [1970-01-01 14:30:09] [INFO]: OperationLog file path: /var/log/ascend_seclog/ascend_install.log.
[Opp] [1970-01-01 14:30:09] [INFO]: Input params: --full
[Opp] [1970-01-01 14:30:10] [INFO]: Existed opp module version is 7.1.0.3.220, the new opp module version is 7.1.0.3.220.
[Opp] [1970-01-01 14:30:15] [WARNING]: file: /usr/local/Ascend/7.0.0/opp/built-in/opp_ump/ai_core/tbe/kernel/ascend310b/pack/Pack_1ef186ed955f153276d6de2e857a3f7_high_perform
ance.json
[Opp] [1970-01-01 14:30:15] [WARNING]: file: /usr/local/Ascend/7.0.0/opp/built-in/opp_ump/ai_core/tbe/kernel/ascend310b/pack/Pack_c4079280e2f300b0371463386596414_high_perform
ance.json
[Opp] [1970-01-01 14:30:15] [WARNING]: file: /usr/local/Ascend/7.0.0/opp/built-in/opp_ump/ai_core/tbe/kernel/ascend310b/mat_mul/MatMulV2_NO_NO_FP16_FP16_true_all.o
[Opp] [1970-01-01 14:30:15] [WARNING]: file: /usr/local/Ascend/7.0.0/opp/built-in/opp_ump/ai_core/tbe/kernel/ascend310b/mat_mul/MatMulV2_NO_NO_FP16_FP16_true_all.o
[Opp] [1970-01-01 14:30:15] [WARNING]: file: /usr/local/Ascend/7.0.0/opp/built-in/opp_ump/ai_core/tbe/kernel/ascend310b/strided_slice_v3/StridedSliceV3_e949b24ebf8bac953440707
314c7d3a_high_performance.o
[Opp] [1970-01-01 14:30:15] [WARNING]: file: /usr/local/Ascend/7.0.0/opp/built-in/opp_ump/ai_core/tbe/kernel/ascend310b/pack/Pack_4388d2ba5973b3a2490cf8dd0e925a_high_perform
ance.json
[Opp] [1970-01-01 14:30:15] [INFO]: Opp package has been installed on the path /usr/local/Ascend/7.0.0, the version is 7.1.0.3.220, and the version of this package is 7.1.0.3.
220, do you want to continue? [y/n]
[Opp] [1970-01-01 14:30:19] [INFO]: parent_dir value is [/usr/local] and children_dir value is [Ascend]
[Opp] [1970-01-01 14:30:19] [INFO]: [Ascend] belongs to root.
[Opp] [1970-01-01 14:30:19] [INFO]: [Ascend] permission is ok.
[Opp] [1970-01-01 14:30:19] [INFO]: parent_dir value is [/usr] and children_dir value is [local]
[Opp] [1970-01-01 14:30:20] [INFO]: [local] belongs to root.
[Opp] [1970-01-01 14:30:20] [INFO]: [local] permission is ok.
[Opp] [1970-01-01 14:30:20] [INFO]: parent_dir value is [/] and children_dir value is [usr]
[Opp] [1970-01-01 14:30:20] [INFO]: [usr] belongs to root.
[Opp] [1970-01-01 14:30:20] [INFO]: [usr] permission is ok.
[Opp] [1970-01-01 14:30:20] [INFO]: parent_dir value is [/] and children_dir value is [/]
[Opp] [1970-01-01 14:30:20] [INFO]: parent_dir permission checked successfully
[Opp] [1970-01-01 14:30:24] [INFO]: upgradePercentage:10%
[Opp] [1970-01-01 14:30:24] [INFO]: Begin upgrade opp module.
[Opp] [1970-01-01 14:30:25] [INFO]: Uninstall the existed opp module before upgrade.
[Opp] [1970-01-01 14:30:25] [INFO]: Begin uninstall the opp module.
[Opp] [1970-01-01 14:30:25] [INFO]: Unset the environment path [ export ASCEND_OPP_PATH=/usr/local/Ascend/7.0.0/opp ].
[Opp] [1970-01-01 14:30:26] [INFO]: Start time: 1970-01-01 14:30:26
[Opp] [1970-01-01 14:30:26] [INFO]: UninstallLog file: /var/log/ascend_seclog/ascend_install.log
[Opp] [1970-01-01 14:30:35] [INFO]: Some files generated by user are not cleared, if necessary, manually clear them, get details in /var/log/ascend_seclog/ascend_install
.log
[Opp] [1970-01-01 14:30:35] [INFO]: Opp_kernel package uninstalled successfully! Uninstallation takes effect immediately.
[Opp] [1970-01-01 14:30:35] [INFO]: End time: 1970-01-01 14:30:35
[Opp] [1970-01-01 14:30:36] [INFO]: Delete the installed opp source files in (/usr/local/Ascend/7.0.0).
[Opp] [1970-01-01 14:30:36] [INFO]: Delete /usr/local/Ascend/7.0.0/opp/built-in/opp_ump/ai_core/tbe/opp_tiling/lib/mtos old custom data path.
[Opp] [1970-01-01 14:30:36] [INFO]: Delete /usr/local/Ascend/7.0.0/opp/built-in/opp_ump/ai_core/tbe/opp_tiling/lib/mtos old custom data path.
[Opp] [1970-01-01 14:30:36] [INFO]: Delete the version info file (/usr/local/Ascend/7.0.0/opp/version.info).
[Opp] [1970-01-01 14:30:36] [INFO]: Delete /usr/local/Ascend/7.0.0/opp/opp_ump old custom data path.
[Opp] [1970-01-01 14:30:36] [INFO]: Delete /usr/local/Ascend/7.0.0/opp/opp_proto old custom data path.
[Opp] [1970-01-01 14:30:36] [INFO]: Delete /usr/local/Ascend/7.0.0/opp/framework old custom data path.
[Opp] [1970-01-01 14:30:36] [WARNING]: /usr/local/Ascend/7.0.0/opp/test-ops, has files changed by users, cannot be delete.
[Opp] [1970-01-01 14:30:36] [WARNING]: lib64, has files changed by users, cannot be delete.
[Opp] [1970-01-01 14:30:36] [WARNING]: script, has files changed by users, cannot be delete.
[Opp] [1970-01-01 14:30:36] [INFO]: Opp package uninstalled successfully! Uninstallation takes effect immediately.
[Opp] [1970-01-01 14:30:36] [INFO]: Update the opp install info.
[Opp] [1970-01-01 14:30:36] [INFO]: upgradePercentage:30%
[Opp] [1970-01-01 14:30:36] [INFO]: Install opp module path in the install folder.
[Opp] [1970-01-01 14:30:36] [INFO]: Set the environment path [ export ASCEND_OPP_PATH=/usr/local/Ascend/7.0.0/opp ].
[Opp] [1970-01-01 14:30:36] [INFO]: upgradePercentage:50%
[Opp] [1970-01-01 14:30:36] [INFO]: Copy opp mtos module successfully.
[Opp] [1970-01-01 14:30:36] [INFO]: Copying version.info
[Opp] [1970-01-01 14:30:36] [INFO]: Update the opp install info.
[Opp] [1970-01-01 14:30:36] [INFO]: /data1/toolkit/run_package/opp/opp/script/opp_upgrade.sh(728): arch: inaccessible or not found
[Opp] [1970-01-01 14:30:36] [INFO]: /data1/toolkit/run_package/opp/opp/script/opp_upgrade.sh(728): cd: no 0.DPMO
[Opp] [1970-01-01 14:30:36] [INFO]: upgradePercentage:100%
[Opp] [1970-01-01 14:30:37] [INFO]: Installation information listed below:
[Opp] [1970-01-01 14:30:37] [INFO]: Install path: (/usr/local/Ascend/7.0.0/opp)
[Opp] [1970-01-01 14:30:37] [INFO]: Install log file path: (/var/log/ascend_seclog/ascend_install.log)
[Opp] [1970-01-01 14:30:37] [INFO]: Operation log file path: (/var/log/ascend_seclog/operation.log)
[Opp] [1970-01-01 14:30:37] [INFO]: Using requirements: when opp module install finished or before you run the opp module, execute the command [ export ASCEND_OPP_PATH=/usr/lo
cal/Ascend/7.0.0 ] to set the opp path.
[Opp] [1970-01-01 14:30:37] [INFO]: Opp package upgraded successfully! The new version takes effect immediately.
[Opp] [1970-01-01 14:30:38] [INFO]: Set precheck info.
[Opp] [1970-01-01 14:30:38] [INFO]: Install MINOR root 1970-01-01 14:30:38 / 7.0.0 : OPP success install_type: cmdList=.
[Opp] [1970-01-01 14:30:38] [INFO]: End Time: 1970-01-01 14:30:38
#

```

4) Install ncs

```
cd /data1/toolkit/run_package/ncs
```

```
./ncs/script/install.sh -- --full
```

```

# ./ncs/sc script/
sceno.info script/
# ./ncs/script/ins
install.sh install_common_parser.sh
# ./ncs/script/install.sh -- --full
[Opp] [1970-01-01 15:19:06] [INFO]: Start time:1970-01-01 15:19:06
[Opp] [1970-01-01 15:19:06] [INFO]: LogFile:/var/log/ascend_seclog/ascend_install.log
[Opp] [1970-01-01 15:19:06] [INFO]: InputParams:--full
[Opp] [1970-01-01 15:19:06] [INFO]: OperationLogFile:/var/log/ascend_seclog/operation.log
[Opp] [1970-01-01 15:19:06] [WARNING]: base version was destroyed or not exist.
[Opp] [1970-01-01 15:19:06] [INFO]: parent_dir value is [/usr/local] and children_dir value is [Ascend]
[Opp] [1970-01-01 15:19:06] [INFO]: [Ascend] belongs to root.
[Opp] [1970-01-01 15:19:06] [INFO]: [Ascend] permission is ok.
[Opp] [1970-01-01 15:19:06] [INFO]: parent_dir value is [/usr] and children_dir value is [local]
[Opp] [1970-01-01 15:19:06] [INFO]: [local] belongs to root.
[Opp] [1970-01-01 15:19:06] [INFO]: [local] permission is ok.
[Opp] [1970-01-01 15:19:06] [INFO]: parent_dir value is [/] and children_dir value is [usr]
[Opp] [1970-01-01 15:19:07] [INFO]: [usr] belongs to root.
[Opp] [1970-01-01 15:19:07] [INFO]: [usr] permission is ok.
[Opp] [1970-01-01 15:19:07] [INFO]: parent_dir value is [/] and children_dir value is [/]
[Opp] [1970-01-01 15:19:07] [INFO]: parent_dir permission check success
[Opp] [1970-01-01 15:19:07] [INFO]: Ncs package has been installed on the path /usr/local/Ascend, the version is 7.1.0.3.220, and the version of this packa
ge is 7.1.0.3.220, do you want to continue? [y/n]
y
[Opp] [1970-01-01 15:19:09] [INFO]: uninstall /usr/local/Ascend/7.0.0/tools/ncs full
[Opp] [1970-01-01 15:19:09] [INFO]: step into run_ncs_uninstall.sh .....
[Opp] [1970-01-01 15:19:09] [INFO]: Install targetdir /usr/local/Ascend/7.0.0 , type full.
[Opp] [1970-01-01 15:19:09] [INFO]: /ncs/script/install.sh -- --full[Ncs] [1970-01-01 15:19:31] [INFO]: Ncs package uninstalled successfully! Uninstallation takes effect immediately.
[Opp] [1970-01-01 15:19:31] [INFO]: install /usr/local/Ascend/7.0.0/tools/ncs full
[Opp] [1970-01-01 15:19:31] [INFO]: step into run_ncs_install.sh .....
[Opp] [1970-01-01 15:19:31] [INFO]: install targetdir /usr/local/Ascend/7.0.0 , type full.
[Opp] [1970-01-01 15:19:31] [INFO]: ncs install upgradePercentage: 10%
[Opp] [1970-01-01 15:19:31] [INFO]: ncs install upgradePercentage: 100%
[Opp] [1970-01-01 15:19:50] [INFO]: Upgrade base version success
[Opp] [1970-01-01 15:19:50] [INFO]: Ncs package installed successfully! The new version takes effect immediately.
Please make sure that
- LD_LIBRARY_PATH includes /usr/local/Ascend/7.0.0/tools/ncs/lib64
[Opp] [1970-01-01 15:19:50] [INFO]: End time:1970-01-01 15:19:50

```

5) Install compiler

```
cd /data1/toolkit/run_package/compiler
```


**./compiler/scripts/install.sh -- --full**

```

[Compiler] [1970-01-01 15:26:04] [INFO]: /data1/toolkit/run_package/compiler/compiler/lib64/te-0.4.0-py3-none-any.whl installed successfully!
[Compiler] [1970-01-01 15:26:04] [INFO]: the compiler tool installed successfully!
[Compiler] [1970-01-01 15:26:04] [INFO]: install hccl extension module begin...
[Compiler] [1970-01-01 15:26:04] [INFO]: install python module package in /data1/toolkit/run_package/compiler/compiler/lib64/hccl-0.1.0-py3-none-any.whl
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use
a virtual environment instead: https://pip.pyppa.io/warnings/venv
[Compiler] [1970-01-01 15:26:09] [INFO]: the hccl extension module installed successfully!
[Compiler] [1970-01-01 15:26:09] [INFO]: start install python module package auto tune.
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use
a virtual environment instead: https://pip.pyppa.io/warnings/venv
[Compiler] [1970-01-01 15:26:17] [INFO]: auto tune installed successfully!
[Compiler] [1970-01-01 15:26:17] [INFO]: start install python module package auto deploy utils.
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use
a virtual environment instead: https://pip.pyppa.io/warnings/venv
[Compiler] [1970-01-01 15:26:22] [INFO]: auto deploy utils installed successfully!
[Compiler] [1970-01-01 15:26:22] [INFO]: start install python module package schedule_search.
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use
a virtual environment instead: https://pip.pyppa.io/warnings/venv
[Compiler] [1970-01-01 15:26:35] [INFO]: schedule_search installed successfully!
[Compiler] [1970-01-01 15:26:35] [INFO]: start install python module package opc tool.
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use
a virtual environment instead: https://pip.pyppa.io/warnings/venv
[Compiler] [1970-01-01 15:26:40] [INFO]: opc tool installed successfully!
[Compiler] [1970-01-01 15:26:40] [INFO]: start install python module package op_compile_tool.
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use
a virtual environment instead: https://pip.pyppa.io/warnings/venv
[Compiler] [1970-01-01 15:26:46] [INFO]: op_compile_tool installed successfully!
[Compiler] [1970-01-01 15:26:46] [INFO]: start install python module package dataflow.
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use
a virtual environment instead: https://pip.pyppa.io/warnings/venv
[Compiler] [1970-01-01 15:26:51] [INFO]: dataflow installed successfully!
[Compiler] [1970-01-01 15:26:52] [INFO]: The package te is already installed in python default path. It is recommended to install it using the '--pylocal'
parameter, install the package te in the /usr/local/Ascend/7.0.0/python/site-packages.
[Compiler] [1970-01-01 15:26:52] [INFO]: Compiler do migrate user assets.
[Compiler] [1970-01-01 15:28:02] [INFO]: Upgrade base version successfully!
[Compiler] [1970-01-01 15:28:02] [INFO]: Upgrade base version successfully!
[Compiler] [1970-01-01 15:28:04] [INFO]: Compiler package installed successfully! The new version takes effect immediately.
Please make sure that
- PATH includes /usr/local/Ascend/7.0.0/compiler/cccec_compiler/bin:/usr/local/Ascend/7.0.0/compiler/bin
- LD_LIBRARY_PATH includes /usr/local/Ascend/7.0.0/compiler/lib64:/usr/local/Ascend/7.0.0/compiler/lib64/plugin/opskernel:/usr/local/Ascend/7.0
.0/compiler/lib64/plugin/memline
- PYTHONPATH includes /usr/local/Ascend/7.0.0/compiler/python/site-packages
[Compiler] [1970-01-01 15:28:06] [INFO]: End time:1970-01-01 15:28:06
#

```

6) Install aoe

cd /data1/toolkit/run_package/aoe**./aoe/script/install.sh -- --full**

```

scene.info script/
# ./aoe/script/CC
# cd /data1/toolkit/run_package/aoe
# ./aoe/script/install.sh -- --full
[Aoe] [1970-01-01 15:31:56] [INFO]: Start time:1970-01-01 15:31:56
[Aoe] [1970-01-01 15:31:56] [INFO]: LogFile:/var/log/ascend_seclog/ascend_install.log
[Aoe] [1970-01-01 15:31:56] [INFO]: InputParams:--full
[Aoe] [1970-01-01 15:31:56] [INFO]: OperationLogFile:/var/log/ascend_seclog/operation.log
[Aoe] [1970-01-01 15:31:56] [INFO]: base version is 7.1.0.3.220.
[Aoe] [1970-01-01 15:31:56] [INFO]: [Ascend] permission is ok.
[Aoe] [1970-01-01 15:31:56] [INFO]: [local] permission is ok.
[Aoe] [1970-01-01 15:31:56] [INFO]: [usr] permission is ok.
[Aoe] [1970-01-01 15:31:56] [INFO]: parent_dirs_permission_check success
[Aoe] [1970-01-01 15:31:56] [INFO]: Aoe package has been installed on the path /usr/local/Ascend, the version is 7.1.0.3.
ge is 7.1.0.3.220, do you want to continue? [y/n]
y
[Aoe] [1970-01-01 15:31:58] [INFO]: uninstall /usr/local/Ascend/7.0.0/tools/aoe full
[Aoe] [1970-01-01 15:31:58] [INFO]: step into run aoe uninstall.sh .....
[Aoe] [1970-01-01 15:31:58] [INFO]: uninstall targetdir /usr/local/Ascend/7.0.0, type full.
[Common] [1970-01-01 15:32:15] [WARNING]: /usr/local/Ascend/7.0.0/aarch64-linux/conf/aoe.ini user configuration file has
[Aoe] [1970-01-01 15:32:20] [INFO]: Aoe package uninstalled successfully! Uninstallation takes effect immediately.
[Aoe] [1970-01-01 15:32:21] [INFO]: install /usr/local/Ascend/7.0.0/tools/aoe full
[Aoe] [1970-01-01 15:32:21] [INFO]: aoe install for all y
[Aoe] [1970-01-01 15:32:21] [INFO]: aoe enable install for all
[Aoe] [1970-01-01 15:32:21] [INFO]: step into run aoe install.sh .....
[Aoe] [1970-01-01 15:32:21] [INFO]: install target dir /usr/local/Ascend/7.0.0, type full.
[Aoe] [1970-01-01 15:32:21] [INFO]: aoe install upgradePercentage:10%
[Aoe] [1970-01-01 15:32:29] [INFO]: merge /usr/local/Ascend/7.0.0/tools/aoe/conf/aoe.ini, /data1/toolkit/run_package/aoe/
realpath: Unknown option 'q' (see "realpath --help")
realpath: Unknown option 'q' (see "realpath --help")
[Aoe] [1970-01-01 15:32:46] [INFO]: aoe install upgradePercentage:100%
[Aoe] [1970-01-01 15:32:46] [INFO]: Upgrade base version success.
[Aoe] [1970-01-01 15:32:46] [INFO]: Aoe package installed successfully! The new version takes effect immediately.
Please make sure that
- LD_LIBRARY_PATH includes /usr/local/Ascend/7.0.0/tools/aoe/lib64
[Aoe] [1970-01-01 15:32:46] [INFO]: End time:1970-01-01 15:32:46
#

```

7) Install pyacl

cd /data1/toolkit/run_package/pyacl



```
./script/install.sh -- -- --full
```

```
# ^C
# ^C
# cd /data1/toolkit/run_package/pyacl
# ./script/install.sh -- -- --full
./script/install.sh: line 5: arch: command not found
[pyACL] [1970-01-01 15:34:43] [INFO] install start
[pyACL] [1970-01-01 15:34:43] [INFO] delete file /usr/local/Ascend/latest/pyACL successfully
[pyACL] [1970-01-01 15:34:43] [INFO] delete file /usr/local/Ascend/latest/python/site-packages/acl.so successfully
[pyACL] [1970-01-01 15:34:43] [INFO] delete file /usr/local/Ascend/latest/python/site-packages/acl/acl.so successfully
Please make sure that
- PYTHONPATH includes /usr/local/Ascend/7.0.0/python
[pyACL] [1970-01-01 15:34:44] [INFO] Ascend-pyACL-7.0.0 install success
#
```

8) Install test

```
cd /data1/toolkit/run_package/test
```

```
./script/install.sh -- -- --full --install-path=/usr/local/Ascend/
```

```
PythonPATH includes /usr/local/Ascend/7.0.0/python
[pyACL] [1970-01-01 15:34:44] [INFO] Ascend-pyACL-7.0.0 install success
# cd /data1/toolkit/run_package/test
# ./script/install.sh -- -- --full --install-path=/usr/local/Ascend/
--install-path=/usr/local/Ascend/
./script/install.sh: line 7: arch: command not found
[test-ops] [1970-01-01 15:35:18] [INFO]: new_version_info 7.0.0
./script/install.sh: line 140: arch: command not found
[test-ops] [1970-01-01 15:35:19] [INFO]: your install path is /usr/local/Ascend/7.0.0/opp/test-ops
[test-ops] [1970-01-01 15:35:19] [INFO]: install is success
[test-ops] [1970-01-01 15:35:21] [INFO]: re-install successfully
#
```

6. 4. Kernels Environment Installation

```
source /usr/local/Python/set_python_env.sh
```

```
cd /data1/kernels/run_package/opp_kernel
```

```
./scripts/install.sh -- -- --full
```

```
# pwd
/data1/kernels/run_package/opp_kernel
# /data1/kernels/run_package/opp_kernel^C
# ./sc
scene.info scripts/
# ./scripts/in
install.sh install_common_parser.sh
# ./scripts/install.sh -- -- --full ./scripts/install.sh -- -- --full^C
# ./scripts/install.sh -- -- --full
[Opp_Kernel] [1970-01-01 15:37:32] [INFO]: Start time: 1970-01-01 15:37:32
[Opp_Kernel] [1970-01-01 15:37:32] [INFO]: LogFile: /var/log/ascend_seclog/ascend_install.log
[Opp_Kernel] [1970-01-01 15:37:32] [INFO]: OperationLogFile: /var/log/ascend_seclog/operation.log
[Opp_Kernel] [1970-01-01 15:37:34] [INFO]: Start to install opp kernel
[Opp_Kernel] [1970-01-01 15:38:54] [INFO]: installPercentage: 100%
[Opp_Kernel] [1970-01-01 15:38:54] [INFO]: Opp_kernels package installed successfully! The new version takes effect immediately.
[Opp_Kernel] [1970-01-01 15:38:54] [INFO]: Using requirements: when opp kernel module install finished or before you run the opp kernel module, execute the
command
[Opp_Kernel] [1970-01-01 15:38:54] [INFO]: End time: 1970-01-01 15:38:54
#
```

val edition here: <https://nubaxterm.mobatek.net>



7. MindSDK User Guide

7.1. Vision SDK Visual Analysis

The development of Computer Vision (CV) is a continuous process of exploration and development. CV was originally developed to enable simple processing of digital images by computers, and its research content mainly includes image processing, pattern recognition, machine learning, deep learning, and other aspects. In the Intelligent Video Analytics (IVA) industry, there are many application areas for traditional computing algorithms, such as object recognition, video structuring, and action behavior recognition.

With the continuous progress of hardware technology and algorithms, video and images have gradually become the main components of global Internet traffic. With the rapid growth of media services, video image processing based on AI image algorithms has gradually become a cost barrier and performance bottleneck in the computing process. In this context, Vision SDK is committed to accelerating video image processing algorithms, improving video image processing performance, reducing the development complexity of CV applications, and accelerating the deployment of CV application development.

7.1.1. Installation and Deployment

1) Execute the following commands in sequence to download and install Ascend-mindxsdk-mxvision.

a. Download the software package.

```
# wget https://ascend-repo.obs.cn-east-2.myhuaweicloud.com/MindX/MindX%206.0.0.SPC1/Ascend-mindxsdk-mxvision_6.0.0.SPC1_linux-aarch64.run
```

b. Install software packages.

```
# chmod +x Ascend-mindxsdk-mxvision_6.0.0.SPC1_linux-aarch64.run
# ./Ascend-mindxsdk-mxvision_6.0.0.SPC1_linux-aarch64.run --install --quiet --install-path=/usr/local/Ascend/
```



7. 1. 2. Usage

7. 1. 2. 1. API Interface Development Method (C++)

Below is a demonstration of developing an image object detection application using the Vision SDK C++ interface. The inference flowchart of the image object detection model is shown in the following figure. The sample is taken from the YoloV3 model of TensorFlow framework.



1) Execute the following command to download the case code to the "/root/mxvision-samples" directory:

```
# cd /root
# mkdir mxvision-samples && cd /root/mxvision-samples
# wget https://mindx.sdk.obs.cn-north-4.myhuaweicloud.com/mindx-sdk-referenceapps%20/mxV
ision/YoloV3Infer/YoloV3Infer.zip --no-check-certificate
```

2) Extract the compressed sample code and enter the extracted directory.

```
# unzip YoloV3Infer.zip
# cd YoloV3Infer
```

3) Download the model file.

```
# cd model
# wget https://obs-9be7.obs.cn-east-2.myhuaweicloud.com/003_Atc_Models/modelzoo/yolov3_tf.pb
```

4) Enable CANN environment variables using the following command.

```
# source /usr/local/Ascend/ascend-toolkit/set_env.sh
# source /usr/local/Ascend/mxVision-6.0.0.SPC1/set_env.sh
```

5) Operational case.

```
# cd /root/mxvision-samples/YoloV3Infer
# dos2unix run.sh
# bash run.sh
```

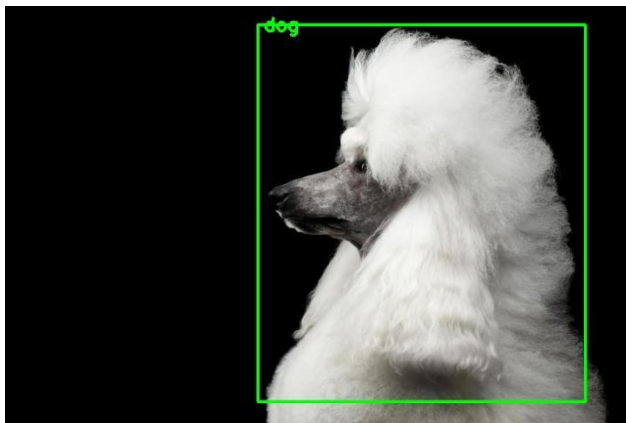


6) After successful operation, the following output will be generated:

```
yoloV3Outputs len=3
*****YoloV3PostProcess*****
Size of objectInfos is 1
objectInfo-0 ,Size:1

*****objectInfo-0:0
x0 is 412.861
y0 is 30.7561
x1 is 947.893
y1 is 644.556
confidence is 0.936385
classId is 16
className is dog
*****YoloV3PostProcess end*****
```

7) The running results are saved in the result.jpg image in the case directory, and when opened, the following image can be seen. Framed an object named dog.



7. 1. 2. 2. API Interface Development Method (Python)

Below is a demonstration of developing an image classification application using the Vision SDK Python interface. The inference process diagram of the image classification model is shown in the following figure. The sample uses the Caffe framework ResNet-50 model.



1) Execute the following command to download the case code to the "/root/mxvision-samples" directory:

```
# cd /root
# mkdir mxvision-samples && cd /root/mxvision-samples
# wget https://mindx.sdk.obs.cn-north-4.myhuaweicloud.com/mindxsdk-referenceapps%20/mxV
ision/resnet50_sdk_python/resnet50_sdk_python_sample.zip --no-check-certificate
```

2) Extract the compressed sample code and enter the extracted directory.

```
# unzip resnet50_sdk_python_sample.zip
# cd resnet50_sdk_python_sample
```

3) Enable CANN environment variables using the following command.

```
# source /usr/local/Ascend/ascend-toolkit/set_env.sh
# source /usr/local/Ascend/mxVision-6.0.0.SPC1/set_env.sh
```

4) Operational case.

```
# cd /root/mxvision-samples/resnet50_sdk_python_sample
# bash run.sh
```

5) After successful operation, the following output will be generated:

```
Standard Poodle: 0.98583984375
save infer result success
```

6) The running results are saved in the result.png image in the case directory, and when opened, the following image can be seen. The upper left corner is labeled with a Standard Poodle of 0.99.



7. 1. 2. 3. Development method for process orchestration

Below is a case study of image classification using the Vision SDK to introduce how to develop inference applications using the Vision SDK process orchestration method. The case uses the YoloV3 model to classify images and finally outputs the classification results. The sample is taken from the YoloV3 model of TensorFlow framework.

1) Execute the following command to download the case code to the "/root/mxvision-samples" directory:

```
# cd /root
# mkdir mxvision-samples && cd /root/mxvision-samples
# wget https://mindx.sdk.obs.cn-north-4.myhuaweicloud.com/mindxsdk-referenceapps%20/mxV
ision/pipelineSample/pipelineSample.zip --no-check-certificate
```

2) Extract the compressed sample code and enter the extracted directory.

```
# unzip pipelineSample.zip
# cd pipelineSample
```

3) Download the model file.

```
# cd models
# wget https://obs-9be7.obs.cn-east-2.myhuaweicloud.com/003_Atch_Models/modelzoo/yolov3_tf.
pb --no-check-certificate
```

4) Enable CANN environment variables using the following command.



```
# source /usr/local/Ascend/ascend-toolkit/set_env.sh
# source /usr/local/Ascend/mxVision-6.0.0.SPC1/set_env.sh
```

5) Operational case.

```
# cd /root/mxvision-samples/pipelineSample
# dos2unix run.sh
# bash run.sh
```

6) After successful operation, the following output will be generated:

- a. “classId” Indicate category number, this is the 16th category.
- b. “className” Represents the class name, where the class name is ' dog '.
- c. “confidence” The maximum confidence level for this classification is 99.5%.
- d. “headerVec” Represents the coordinates of four vertices.

```
Results:{"MxpiObject":[{"classVec":[{"classId":16,"className":"dog","confidence":0.995259583,"headerVec":[]}], "x0":113.482422,"x1":883.72522,"y0":127.949326,"y1":595.702576}]}
```



8. Instructions for using the Linux kernel source package

8.1. Requirements for the compilation host system

The current Linux kernel source code package has only been tested on X64 computers with **Ubuntu 22.04**, so first make sure that the Ubuntu version installed on your computer is Ubuntu 22.04. The command to check the Ubuntu version installed on your computer is as follows. If the Release field does not display **22.04**, it means that the current Ubuntu version does not meet the requirements. Please change the system before performing the following operations.

```
test@test:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:  Ubuntu 22.04 LTS
Release:      22.04
Codename:     jammy
```

If your computer is running Windows and does not have Ubuntu 22.04 installed, you can consider using **VirtualBox** or **VMware** to install an Ubuntu 22.04 virtual machine in Windows. However, please note that the Linux kernel source code package has not been tested in the WSL virtual machine, so it cannot be guaranteed to run properly in WSL. The installation image download address of Ubuntu 22.04 **amd64** version is:

<https://mirrors.tuna.tsinghua.edu.cn/ubuntu-releases/22.04/ubuntu-22.04-desktop-amd64.iso>

After installing Ubuntu 22.04 on your computer or in a virtual machine, please first set the software source of Ubuntu 22.04 to Tsinghua source (or other fast domestic sources), otherwise it is easy to make mistakes when installing the software later due to network reasons. The steps to replace Tsinghua source are as follows:

1) For the method of replacing Tsinghua source, please refer to the instructions on this webpage.

<https://mirrors.tuna.tsinghua.edu.cn/help/ubuntu/>

2) Note that the Ubuntu version needs to be switched to 22.04.



Ubuntu 镜像使用帮助

Ubuntu 的软件源配置文件是 `/etc/apt/sources.list`。将系统自带的该文件做个备份，将该文件替换为下面内容，即可使用 TUNA 的软件源镜像。

选择你的ubuntu版本:

22.04 LTS

```
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse

# 预发布软件源，不建议启用
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
```

3) The content of the `/etc/apt/sources.list` file that needs to be replaced is:

```
test@test:~$ sudo mv /etc/apt/sources.list cat /etc/apt/sources.list.bak
```

```
test@test:~$ sudo vim /etc/apt/sources.list
```

The source mirror is commented out by default to increase the speed of apt update. You can uncomment it if necessary.

```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
```

```
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
```

```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
```

```
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
```

```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
```

```
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
```

```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
```

```
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
```

```
# Pre-release software source, not recommended to enable
```

```
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
```

```
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
```

4) After the replacement, you need to update the package list and make sure there are no errors.

```
test@test:~$ sudo apt-get update
```



8.2. Install the cross-compilation toolchain and dependent packages

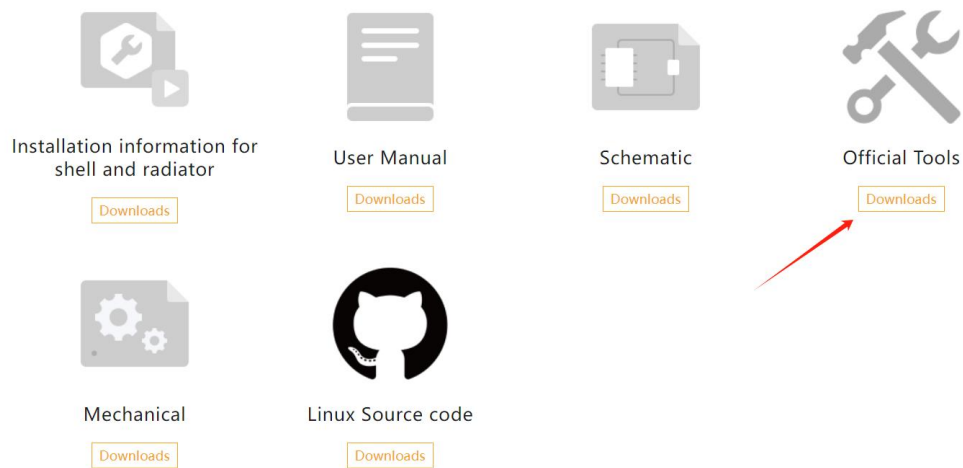
9) The compressed package of the cross-compilation tool chain can be downloaded from the development board's data download page. The steps are:

- a. Open the link below:

[http://www.orangepi.cn/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro\(20T\).html](http://www.orangepi.cn/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro(20T).html)

- b. Then select the official tool.

Official Resources



- c. Then download the **toolchain.tar.gz** compressed package in the cross-compilation toolchain folder.



10) Then execute the following command in the Ubuntu 22.04 computer to switch to the root user.

```
test@test:~$ sudo -i
```

11) Then install the following dependency packages

```
root@test:~# apt install -y python3 make gcc unzip pigz bison flex libncurses-dev cmake
root@test:~# apt install -y squashfs-tools bc device-tree-compiler libssl-dev rpm2cpio g++
```



12) Then execute the following command to create the **/opt/compiler** directory and enter the **/opt/compiler** directory.

```
root@test:~# mkdir /opt/compiler
root@test:~# cd /opt/compiler
```

13) Then copy the downloaded **toolchain.tar.gz** to **/opt/compiler**, and then use the following command to decompress **toolchain.tar.gz** to **/opt/compiler**.

```
root@test:/opt/compiler# tar -xvf toolchain.tar.gz
```

14) The decompressed cross-compilation tool chain is as follows:

```
root@test:/opt/compiler# ls toolchain
aarch64-target-linux-gnu bin include lib lib64 libexec sysroot
```

15) Then add the cross-compilation toolchain path in the configuration file.

```
root@test:~# echo "export PATH=/opt/compiler/toolchain/bin:$PATH: " >> /etc/profile
```

16) Then execute the following command to make the environment variables take effect.

```
root@test:~# source /etc/profile
```

17) Then you can execute the following command to check the cross-compilation toolchain version. If version information is displayed, it means that the toolchain is installed successfully.

```
root@test:~# aarch64-target-linux-gnu-gcc -v
Using built-in specs.
COLLECT_GCC=aarch64-target-linux-gnu-gcc
COLLECT_LTO_WRAPPER=/opt/compiler/toolchain/bin/../libexec/gcc/aarch64-target-linux-gnu/7.3.0/lto-wrapper
Target: aarch64-target-linux-gnu
.....
Thread model: posix
gcc version 7.3.0 (Do-Compiler V100R001C13B001)
```



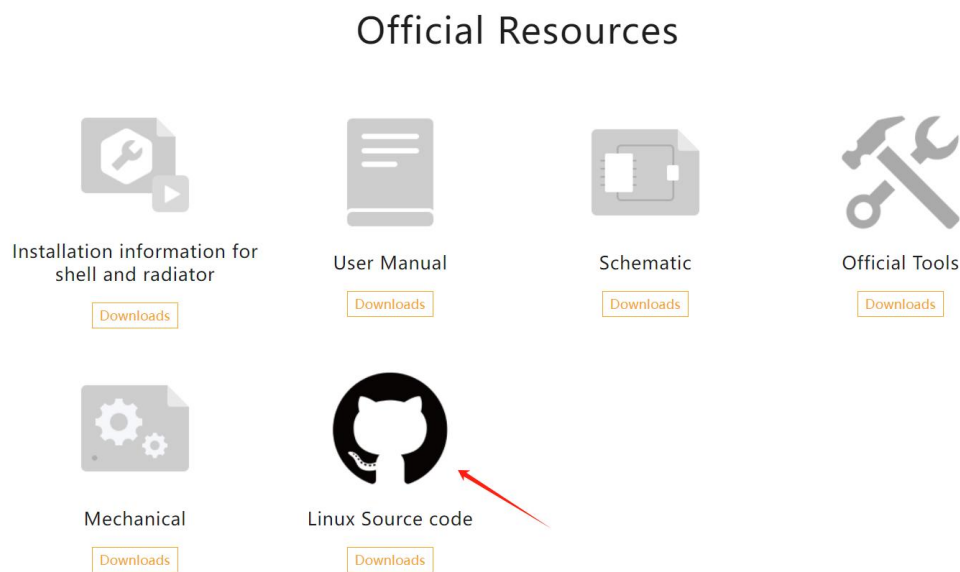
8.3. Download and decompress the Linux kernel source package

1) The Linux kernel source code compressed package can be downloaded from the development board's data download page. The steps are:

- a. Open the link below:

[http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro\(20t\).html](http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro(20t).html)

- b. Then select Linux source code.



- c. Then download the Linux kernel source code compressed package **Ascend310B-source-opi.tar.gz**.

Ascend310B-source-opi.tar.gz

2) Then in the Ubuntu 22.04 computer, execute the following command to switch to the root user.

```
test@test:~$ sudo -i
```

3) Then copy the downloaded Linux kernel source code compressed package **Ascend310B-source-opi.tar.gz** to the **/opt** directory of the Ubuntu 22.04 computer, and then use the following command to decompress the Linux kernel source code compressed package.



```
root@test:/opt # tar xzf Ascend310B-source-opi.tar.gz
```

4) The contents of the decompressed Linux kernel source package are as follows:

```
root@test:/opt # cd Ascend310B-source-opi  
root@test:/opt/Ascend310B-source-opi# ls  
abl build.sh config driver dtb kernel scripts tools
```

8. 4. How to compile and validate kernel image files

1) First enter the Ubuntu 22.04 computer, then execute the following command to switch to the root user.

```
test@test:~$ sudo -i
```

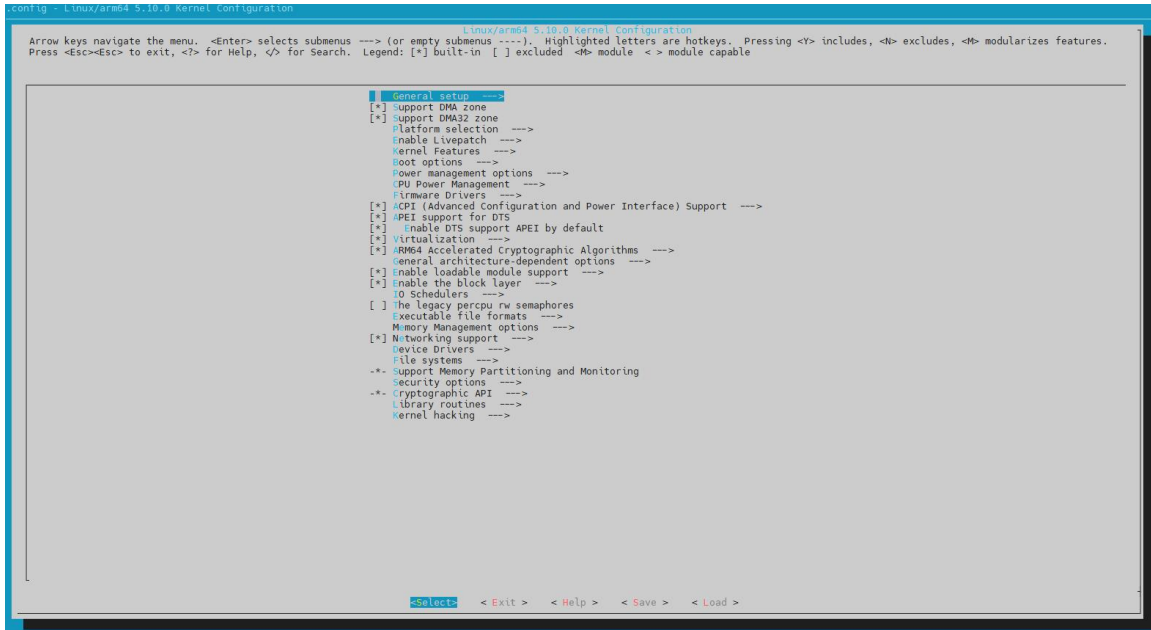
2) Then execute the following command to enter the “ **Ascend310B-source-opi** ” directory.

```
root@test:~# cd /opt/Ascend310B-source-opi
```

3) Then execute the following command to start compiling the kernel.

```
root@test:/opt/Ascend310B-source-opi# bash build.sh kernel
```

4) During the execution, a graphical interface for kernel configuration options will pop up. If no changes are required, simply select **Exit** to exit.



5) After the compilation is completed, the following information will be printed:

```
generate /opt/Ascend310B-source-opi/output/kernel_modules success!
generate /opt/Ascend310B-source-opi/output/Image success!
sign /opt/Ascend310B-source-opi/output/Image success!
```

6) The compiled Image file will be stored in the **Ascend310B-source/output** directory.

```
root@test:/opt/Ascend310B-source-opi# ls output/Image
output/Image
```

7) The method to update the kernel Image file after compilation is as follows:

- First log in to the Linux system of the development board.
- Then upload the compiled **Image** file to any directory of the development board Linux system, such as the **/root** directory.
- Then enter the **/root** directory.
- Then execute the following command to update the Image file.
 - NVMe SSD boot:

```
dd if=Image of=/dev/nvme0n1 count=61440 seek=32768 bs=512
```

- SATA SSD boot:

```
dd if=Image of=/dev/sda count=61440 seek=32768 bs=512
```

- eMMC boot:



```
dd if=Image of=/dev/mmcblk0 count=61440 seek=32768 bs=512
```

d) TF card startup:

```
dd if=Image of=/dev/mmcblk1 count=61440 seek=32768 bs=512
```

8. 5. How to compile and validate kernel DTB files

2) First enter the Ubuntu 22.04 computer, then execute the following command to switch to the root user.

```
test@test:~$ sudo -i
```

3) Then execute the following command to enter the **Ascend310B-source-opi** directory.

```
root@test:~# cd /opt/Ascend310B-source-opi
```

4) The DTS file used by the development board is as follows:

```
Ascend310B-source-opi/dtb/dts/hi1910b/hi1910BL/hi1910B-default.dts
```

5) Then execute the following command to start compiling DTB.

```
root@test:/opt/Ascend310B-source-opi# bash build.sh dtb
```

6) If the following information is printed at the end, it means that the kernel DTB file is compiled successfully.

```
generate /opt/Ascend310B-source-opi/output/dt.img success!  
sign /opt/Ascend310B-source-opi/output/dt.img success!
```

7) The generated DTB file is **dt.img** and is stored in the **output** directory:

```
root@orange-pi-M600:/opt/Ascend310B-source-opi# ls output/dt.img  
output/dt.img
```

8) The method to update the kernel DTB file after compilation is as follows:

- a. First log in to the Linux system of the development board.
- b. Then upload the compiled **dt.img** file to any directory of the development board Linux system, such as the **/root** directory.
- c. Then enter the **/root** directory.
- d. Then execute the following command to update the **dt.img** file. The DTB file has two copies, the primary and the backup. The first of the following two



commands updates the DTB file in the primary area, and the second one updates the DTB file in the backup area. When testing, generally only the DTB file in the primary area needs to be updated, and the DTB file in the backup area can be updated after the test is successful.

a) NVMe SSD boot:

```
dd if=dt.img of=/dev/nvme0n1 count=4096 seek=114688 bs=512
dd if=dt.img of=/dev/nvme0n1 count=4096 seek=376832 bs=512
```

b) SATA SSD boot:

```
dd if=dt.img of=/dev/sda count=4096 seek=114688 bs=512
dd if=dt.img of=/dev/sda count=4096 seek=376832 bs=512
```

c) eMMC boot:

```
dd if=dt.img of=/dev/mmcblk0 count=4096 seek=114688 bs=512
dd if=dt.img of=/dev/mmcblk0 count=4096 seek=376832 bs=512
```

d) TF card startup:

```
dd if=dt.img of=/dev/mmcblk1 count=4096 seek=114688 bs=512
dd if=dt.img of=/dev/mmcblk1 count=4096 seek=376832 bs=512
```

9. Instructions for using the Linux image compilation script

9. 1. Requirements for the compilation host system

The current Linux image compilation script has only been tested on X64 computers with **Ubuntu 22.04**, so first make sure that the Ubuntu version installed on your computer is Ubuntu 22.04. The command to check the Ubuntu version installed on the computer is as follows. If the Release field does not display **22.04**, it means that the current Ubuntu version does not meet the requirements. Please change the system before performing the following operations.

```
test@test:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
```



Description: Ubuntu 22.04 LTS

Release: **22.04**

Codename: **jammy**

If your computer is running Windows and does not have Ubuntu 22.04 installed, you can consider using **VirtualBox** or **VMware** to install an Ubuntu 22.04 virtual machine in Windows. However, please note that the Linux image compilation script has not been tested in the WSL virtual machine, so it cannot be guaranteed to run properly in WSL. The installation image download address for Ubuntu 22.04 **amd64** is:

<https://mirrors.tuna.tsinghua.edu.cn/ubuntu-releases/22.04/ubuntu-22.04-desktop-amd64.iso>

After installing Ubuntu 22.04 on your computer or in a virtual machine, please first set the software source of Ubuntu 22.04 to Tsinghua source (or other fast domestic sources), otherwise it is easy to make mistakes when installing the software later due to network reasons. The steps to replace Tsinghua source are as follows:

1) For the method of replacing Tsinghua source, please refer to the instructions on this webpage.

<https://mirrors.tuna.tsinghua.edu.cn/help/ubuntu/>

2) Note that the Ubuntu version needs to be switched to 22.04.

Ubuntu 镜像使用帮助

Ubuntu 的软件源配置文件是 `/etc/apt/sources.list`。将系统自带的该文件做个备份，将该文件替换为下面内容，即可使用 TUNA 的软件源镜像。

选择你的ubuntu版本:

22.04 LTS

```
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse

# 预发布软件源，不建议启用
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
```

3) The content of the `/etc/apt/sources.list` file that needs to be replaced is:

```
test@test:~$ sudo mv /etc/apt/sources.list cat /etc/apt/sources.list.bak
```

```
test@test:~$ sudo vim /etc/apt/sources.list
```

```
# The source mirror is commented out by default to increase the speed of apt update. You can uncomment it if necessary.
```

```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
```



```
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse

# Pre-release software source, not recommended to enable
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
```

4) After the replacement, you need to update the package list and make sure there are no errors.

```
test@test:~$ sudo apt-get update
```

9. 2. Things you need to prepare to make a Linux image

The hardware and software conditions required to create a Linux image are as follows:

- 1) An X64 computer with a USB port and Ubuntu 22.04.
- 2) A TF card reader.
- 3) A TF card with a capacity of at least 32GB.
- 4) Make sure the computer's network is accessible.

9. 3. Download the source code package of the Linux image compilation script

1) The scripts, driver packages, CANN packages, basic rootfs and other software needed to compile the Linux image are all packaged into a compressed package and placed on Baidu Netdisk. The download steps are as follows:

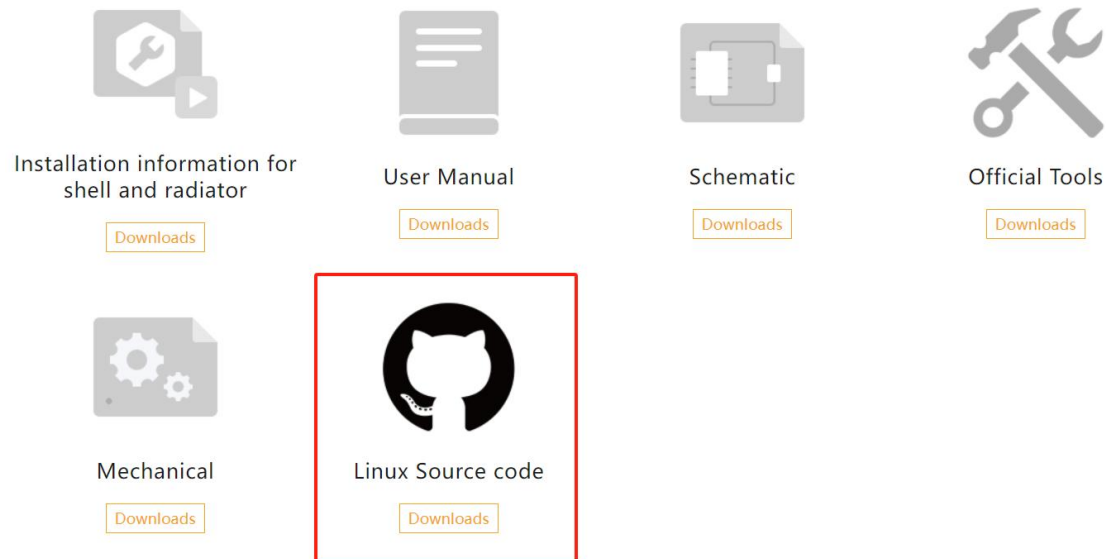
- a. First open the development board's download page:

```
http://www.orangepi.cn/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro\(20T\).html
```

- b. Then select Linux source.



Official Resources



- c. Then download the **image-builder.tar.gz** compressed package.

[image-builder.tar.gz](#)

- 2) Then execute the following command in the Ubuntu 22.04 computer to switch to the root user.

```
test@test:~$ sudo -i
```

- 3) Then copy the downloaded **image-builder.tar.gz** to the **/opt** directory and decompress it using the following command.

```
root@test:/opt# tar xzf image-builder.tar.gz
root@test:/opt# cd image-builder/src
root@test:/opt/image-builder/src# ls
complete compress minimal
```

- 4) The decompressed **image-builder/src** directory contains three modules: minimal image (minimal), complete image (complete), and compressed and expanded image (compress), which correspond to the three steps of making an image.

Module Name	Module Name	Function Introduction
Minimal	src/minimal	An image that can be started on the



image		development board but lacks some dependencies
Minimal image	src/minimal	Complete dependency image
Compressed and expanded image	src/compress	Complete dependency image with compression and expansion functions

9. 4. How to create a minimal image

1) First, make sure that the Ubuntu 22.04 system is not set to the Chinese environment. If so, please change it back to the English environment, otherwise the creation of the minimum image will fail.

2) Then insert the TF card into the card reader, and then insert the card reader into the USB port of the computer.

3) Then execute the following command in the Ubuntu 22.04 computer to switch to the root user.

```
test@test:~$ sudo -i
```

4) Then install the following dependency packages

```
root@test:~# apt-get install -y qemu qemu-user qemu-user-static binfmt-support
```

5) Then copy the two library files that the **emmc-head** command depends on to **/usr/lib64**. The steps are as follows:

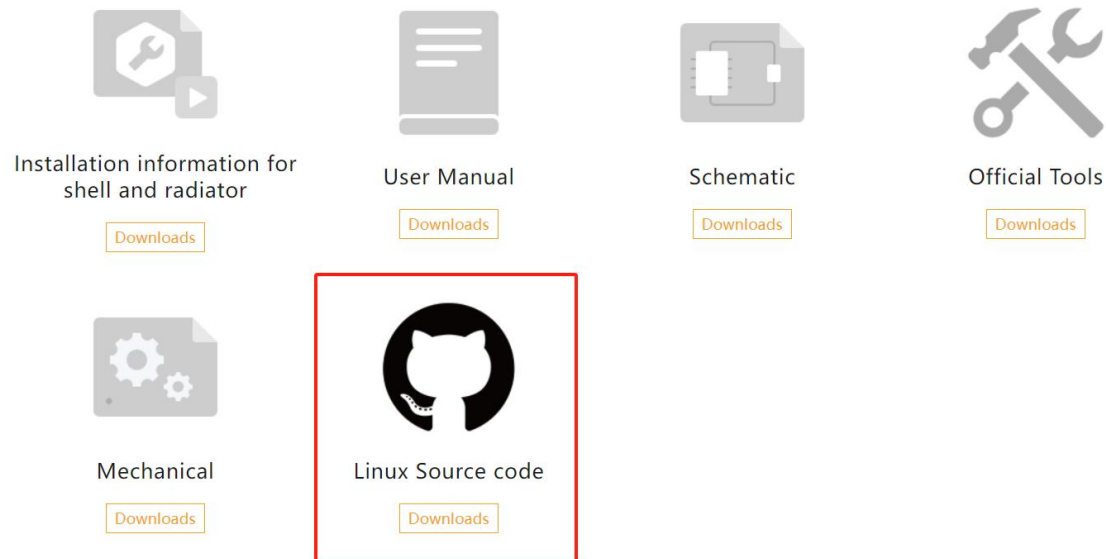
a. First open the development board's download page:

```
http://www.orangepi.cn/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro\(20T\).html
```

b. Then select Linux source code.



Official Resources



- c. Then download the dependent library files.



- d. Copy the downloaded library file to the `/usr/lib64` directory of the Ubuntu 22.04 system.

```
root@test:~# cp library/* /usr/lib64/
```

- e. Then run the **emmc-head** command. If the output is the same as below, it means the library file is installed correctly.

```
root@test:~# cd /opt/image-builder/src/minimal
root@test:/opt/image-builder/src/minimal# ./ubuntu/22.04/download/emmc-head --help
Usages: emmc-head firmware_path boot_a_devname boot_b_devname [force_recover]
The following files must be contained in firmware_path:
Image, itrustee.img, dt.img, initrd.
boot_a_devname: A Partition boot device name, for example, eMMC:mmcblk0p2, SD:mmcblk1p2
boot_b_devname: B Partition boot device name, for example, eMMC:mmcblk0p3, SD:mmcblk1p3
force_recover: force recover flag.
Example: /var/davinci/driver/emmc-head ./firmware /dev/mmcblk0p2 /dev/mmcblk0p3
```

- 6) Then go to the directory where the **image-builder** source code is located, and then go to the **src/minimal** directory.



```

root@test:~# cd /opt/image-builder
root@test:/opt/image-builder# cd src/minimal
root@test:/opt/image-builder/src/minimal# ls
base.sh openEuler ubuntu

```

7) The path where the pre-downloaded software needed for the image creation process is stored is:

- a. The path of the download folder corresponding to openEuler:

```
openEuler/22.03/download
```

- b. The path of the download folder corresponding to ubuntu:

```
ubuntu/22.04/download
```

8) Then use the **fdisk -l** command to check the disk number of the TF card, such as **/dev/sdb**.

```

root@test:~# fdisk -l
.....
Disk /dev/sdb: 29.72 GiB, 31914983424 bytes, 62333952 sectors
Disk model: MassStorageClass
.....

```

9) Then start making the minimum image to the TF card. The command is as follows:

- a. The command for Ubuntu 22.04 image is as follows:

```
root@test:/opt/image-builder/src/minimal# bash base.sh ubuntu/22.04/ /dev/sdX ubuntu/22.04/download/
```

- b. The command for the openEuler 22.03 image is as follows:

```
root@test:/opt/image-builder/src/minimal# bash base.sh openEuler/22.03/ /dev/sdX openEuler/22.03/download/
```

Note that **/dev/sdX** in the above command needs to be replaced with the disk number corresponding to the TF card, please do not copy it.

10) After normal operation, the following information will be printed, then you can eject the TF card, and then insert the TF card into the development board to start running (note that please do not delete the contents of the **ubuntu/22.04/download** or **openEuler/22.03/download** folders. Currently, the script cannot automatically download some software packages required to make the minimum image through the network, and can only use the cached ones).



[2024-02-03 15:49:41] [MINIMAL] Minimal image build successful!

11) Note that the created minimum image will automatically restart once when it is started for the first time. Please wait until the automatic restart is completed before logging into the system through the serial port to perform other operations.

12) Both Ubuntu and openEuler have a **cfg.json** file to control which steps the script runs. By default, all steps are run (as shown below, all are **y**). In the process of making a minimum image, each time a step is run, the y after the corresponding step will be changed to n. After all steps are run, all n will be changed back to y. If an error occurs in a step in the middle and the script is exited, when the script is executed again, it will continue from the previous interrupted step.

```
root@test:/opt/image-builder/src# cat minimal/ubuntu/22.04/cfg.json
{
.....

  "function": {
    "get_base_image": "y",
    "get_npu_driver": "y",
    "get_hdk": "y",
    "get_file_system": "y",
    "write_to_device": "y",
    "post_process": "y",
    "opi_func": "y"
  }
}
```

9. 5. How to create a complete image

1) First, please follow the instructions in the section "**How to make a minimum image**" to make a minimum image, then start the minimum image and log in to the serial port command line as the **root** user. If you do not have the environment to make a minimum image yourself, you can directly download the minimum image (**that is, the minimal version of the image**) file provided by Orange Pi, and then burn it to a TF card with a capacity of 32GB or more.



2) Then make sure the development board can access the Internet normally.

3) The minimum image (**i.e. the minimal version of the image**) already contains the scripts and some software packages required to make a complete image. The paths where they are stored are as follows:

```
/opt/complete/
```

4) Then execute the following command in the Ubuntu 22.04 computer to switch to the root user.

```
test@test:~$ sudo -i
```

5) Then enter **complete**.

```
root@orangepiaipro-20t:~# cd /opt/complete
root@orangepiaipro-20t:/opt/complete# ls
base.sh download openEuler ubuntu
```

6) Then use the following command to create a complete image.

a. The command for Ubuntu image is as follows:

```
root@orangepiaipro-20t:/opt/complete# bash base.sh -v ubuntu/22.04/ download/
```

b. The command of OpenEuler image is as follows:

```
root@orangepiaipro-20t:/opt/complete# bash base.sh -v openEuler/22.03/ download/
```

7) Both Ubuntu and openEuler have a **cfg.json** file to control which steps the script runs. By default, all steps are run (as shown below, all are **y**). In the process of making a complete image, each time a step is run, the y after the corresponding step will be changed to n. After all steps are run, all n will be changed back to y. If an error occurs in a step in the middle and the script is exited, when the script is executed again, it will continue from the previous interrupted step.

```
root@test:/opt/image-builder/src/complete# cat ubuntu/22.04/cfg.json
{
.....

  "function": {
    "pre_process": "y",
```




```
"apt_install": "y",
"install_miniconda": "y",
"python_pip_install": "y",
"install_cann": "y",
"install_mxvision": "y",
"install_acllite": "y",
"add_local_desktop": "y",
"add_remote_desktop": "y",
"opi_func": "y",
"post_process": "y"
},
.....
}
```

9. 6. Methods for making compression and expansion images

1) First, insert the prepared complete image TF card into the card reader, and then insert the card reader into the USB port of the computer.

2) Then execute the following command in the Ubuntu 22.04 computer to switch to the root user.

```
test@test:~$ sudo -i
```

3) Then copy the two library files that the **emmc-head** command depends on to **/usr/lib64** (if you have already done this step, you can skip it). The steps are as follows:

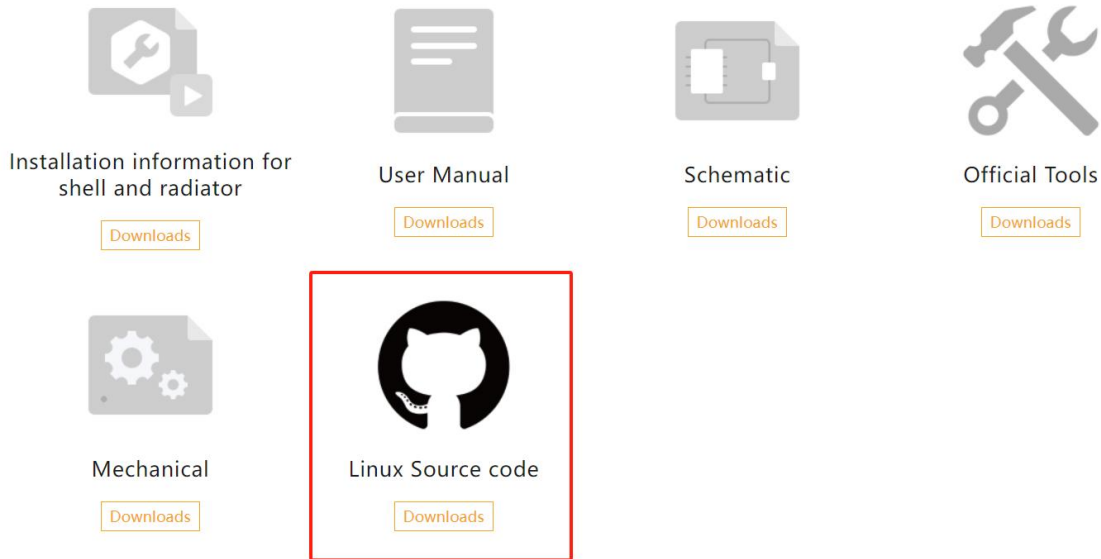
a. First open the development board's download page:

```
http://www.orange-pi.cn/html/hardware/computerAndMicrocontrollers/service-and-support/Orange-Pi-AIpro\(20T\).html
```

b. Then select Linux source code.



Official Resources



- c. Then download the dependent library files.



- d. Copy the downloaded library file to the `/usr/lib64` directory of the Ubuntu 22.04 system.

```
root@test:~# cp library/* /usr/lib64/
```

- e. Then run the **emmc-head** command. If the output is the same as below, it means that the library file is installed correctly.

```
root@test:~# cd /opt/image-builder/src/compress
root@test:/opt/image-builder/src/compress# ./download/emmc-head --help
Usages: emmc-head firmware_path boot_a_devname boot_b_devname [force_recover]
The following files must be contained in firmware_path:
Image, itrustee.img, dt.img, initrd.
boot_a_devname: A Partition boot device name, for example, eMMC:mmcblk0p2, SD:mmcblk1p2
boot_b_devname: B Partition boot device name, for example, eMMC:mmcblk0p3, SD:mmcblk1p3
force_recover: force recover flag.
Example: /var/davinci/driver/emmc-head ./firmware /dev/mmcblk0p2 /dev/mmcblk0p3
```

- 4) Then go to the **compress** directory of **image-builder**.

```
root@test:~# cd /opt/image-builder/src/compress/
```



```
root@test:/opt/image-builder/src/compress# ls
base.sh config.ini download E2E_samples_download_tool.sh openEuler ubuntu
```

5) Then use the **fdisk -l** command to check the disk number of the TF card, such as **/dev/sdb**.

```
root@test:/opt/image-builder/src/compress# fdisk -l
.....
Disk /dev/sdb: 29.72 GiB, 31914983424 bytes, 62333952 sectors
Disk model: MassStorageClass
.....
```

6) Then you can start exporting the image in the TF card. The command is as follows:

a. The command to export the Ubuntu 22.04 image is as follows:

```
root@test:/opt/image-builder/src/compress# bash base.sh -c ubuntu/22.04 /dev/sdX linux.img
```

b. The command to export the openEuler 22.03 image is as follows:

```
root@test:/opt/image-builder/src/compress# bash base.sh -c openEuler/22.03/ /dev/sdX linux.img
```

Note that **/dev/sdX** in the above command needs to be replaced with the disk number corresponding to the TF card, please do not copy it.

7) When you see the following output, the image has been exported and compressed.

```
[2024-02-03 16:28:57] [COMPRESS] sd card compress success!
```

8) The exported Linux image file is as follows:

a. **linux.img**: Linux Image File

b. **linux.img.xz**: Compressed Linux image file

```
root@test:/opt/image-builder/src/compress# ls linux.img*
linux.img linux.img.xz
```

9) If you do not need to compress the Linux image file, you can remove the **-c** option in the command.

a. The command to export the Ubuntu 22.04 image is as follows:

```
root@test:/opt/image-builder/src/compress# bash base.sh ubuntu/22.04 /dev/sdX linux.img
```

b. The command to export the openEuler 22.03 image is as follows:

```
root@test:/opt/image-builder/src/compress# bash base.sh openEuler/22.03/ /dev/sdX linux.img
```



10. Appendix

10.1. User Manual Update History

Version	Date	Release Notes
v0.1	2024-06-25	initial version
v0.2	2024-07-02	1. Method for installing wiringOP 2. Method of using wiringOP to control 40pin GPIO 3. Test method for 40 pin CAN
v0.3	2024-07-17	1. The usage method of wiringOP hardware PWM
v0.4	2024-07-19	1. ffmpeg with ascend hardware acceleration
v0.5	2024-09-04	1. Method for installing kernel header files



		2. Method of installing ZFS
v0.6	2025-02-18	1. Instructions for Updating Ubuntu Desktop Image Preset Cases
v0.7	2025-03-17	1. Method of burning Linux images using Win32Diskimager 2. wiringOP-Python installation and usage instructions
v0.8	2025-03-25	1. Add usage instructions for MindSDK Vision visual analysis suite
v0.9	2025-08-19	1. Add OpenHarmony installation methods and OpenHarmony AI application environment installation methods

10.2. Image Update History

Date	Release Notes
2024-06-25	opiaipro_20t_ubuntu22.04_minimal_aarch64_20240621.img.xz opiaipro_20t_ubuntu22.04_desktop_aarch64_20240618.img.xz opiaipro_20t_openEuler22.03_desktop_aarch64_20240620.img.xz *initial version
2024-09-24	opiaipro_20t_ubuntu22.04_minimal_aarch64_20240924.img.xz opiaipro_20t_ubuntu22.04_desktop_aarch64_20240924.img.xz opiaipro_20t_openEuler22.03_desktop_aarch64_20240924.img.xz * Optimize the automatic temperature control mechanism of PWM fan
2025-02-11	opiaipro_20t_ubuntu22.04_desktop_aarch64_20250211.img.xz * Update MindSpore AI application examples to support DeepSeek-R1-Distill-Qwen-1.5B model inference * Update CANN version to 8.0.0 * Update MindSpore version to v2.4.10 * Pre installed MindNLP v0.4.1
2025-08-19	Opiaipro-20t_ohos_v5.0.3_aarch64_20250730.zip Ascend310B-OpenHarmony-CANN.zip Ascend310B-boot-firmware-40M-for-Openharmony.run