

TW-T506S用户手册

TW-T506S MANUAL

目录

目录	1
文档修订目录	2
文档版本	2
Version3.0	2
安全警示及使用注意事项	4
简介	5
产品规格	6
处理器及核心模块	6
编解码	6
接口	7
供电	8
结构	8
环境	8
尺寸	9
服务与支持	10
技术支持	10
保修	10
功能介绍	11
系统介绍	11
系统重刷	11
开关机	11
工作模式切换	12
接口说明	13
正面接口	13
串口说明	14
CAN 功能测试	16
安装盒子支架	22
Xshell 的使用	23
4G 模块安装	26
4G 拨号联网	29
固态硬盘(SSD)安装	32
WiFi 热点设定及连接	33
采用 M.2 Key M SSD 为系统盘	36
系统安装	42
Jtop 安装	44
VNC Viewer 安装(远程图形界面工具)	46
常用框架安装	48

文档修订目录

文档版本

Version3.0

文档版本号	修订日期	修订内容
V1.0	2020/07/23	初始发布
V2.0	2020/10/29	全面重制, 去除错误信息, 完善资料
V3.0	2021/12/31	去除错误, 增加新的内容

前言

在使用本手册之前，请您认真阅读以下使用许可协议，只有在同意以下使用许可协议的情况下方能使用本手册中介绍的产品。

版权声明

保留对本文档及本声明的最终解释权和修改权。本文档中出现的任何文字叙述、文档格式、插图、照片、方法、过程等内容，除另有特别注明外，其著作权或其他相关权利均属于本公司。未经本公司书面同意，任何人不得以任何方式或形式对本手册内的任何部分进行复制、摘录、备份、修改、传播、翻译成其它语言、将其全部或部分用于商业用途。

免责条款

本文档依据现有信息制作，其内容如有更改，恕不另行通知。在编写该文档的时候已尽最大努力保证其内容准确可靠，但对本文档中的遗漏、不准确、或错误导致的损失和损害承担责任。

技术支持与信息反馈

如果您在使用我们的产品时遇到问题,或者您认为我们的产品有某些功能缺陷,请访问我们的官网<https://www.52solution.com> 联系我们的客服,我们将为您解决问题和反馈;或者需要技术支持指导以及有任何宝贵意见,也请您通过官网或者电话联系我们:

安全警示及使用注意事项

● 安全说明

在使用本产品之前，必须先查阅本文档，对该产品有初步的认识与了解，且须遵守本产品使用手册中的安全说明以保证您的个人安全并避免损坏设备，若盲目操作造成损失或伤害，制造商对其错误操作造成的设备及个人生命财产安全的任何问题均不负责。

● 电源电压

TW-T506S 边缘计算平台输入端电源稳定可靠，功率 30W。

电源范围：12- 36 V DC；（如果要使用在移动装置上，使用蓄电池供电，应该保证电源电压满足 12V 的要求，可以在蓄电池上安装升压模块，保证电压能达到正常工作需求，避免因电源供电不足导致设备无法启动，如果使用车载电源，建议等车内其他设备均已启动完毕再插电开启设备）

电流：5A (MAX)

● 环境要求：

工作温度：-20℃ - 65℃

通风要求：计算平台安装的周边必须有良好通风的条件。

● 接地要求

电源适配器的供电源必须有良好的接地，特殊情况需安装计算平台上接地螺丝接地。

● 静电防护

电子元件和电路对静电放电很敏感，虽然本公司在设计电路板卡产品时会板卡上的主要接口做防静电保护设计，但很难对所有元件及电路做到防静电安全防护。因此在处理任何电路板组件时，建议遵守防静电安全保护措施。防静电安全保护措施包括，但不限于以下几点：

- ◆ 运输、存储过程中应将盒子放在防静电袋中，直至安装部署时再拿出板卡；
- ◆ 在身体接触盒子之前应将身体内寄存的静电释放掉：佩戴放电接地腕带；
- ◆ 仅在静电放点安全区域内操作盒子；
- ◆ 避免在铺有地毯的区域搬移盒子。

● 操作与维护

操作或维护人员需先经培训合格，方可参与操作或维护。

简介

TW-T506S 为一款基于 NVIDIA® Jetson Xavier NX 系列模块设计的计算平台，内置集成 Xavier NX 模块，预装 Ubuntu 18.04 操作系统，具备 21TOPS 的 AI 处理能力，支持电源适配器和 PoE 网口两种电源输入方式，超强固轻型铝合金材料设计，无风扇结构传导被动散热，尺寸轻巧外观新颖，丰富 IO 接口类型，预留便于现场安装的底部支架，具备超长 MTBF 稳定运行能力，可应用于机器人、无人配送车、低空防御、智能巡检、智慧楼宇等自主化机器，是边缘端部署 AI 算力进行深度学习的理想载体。

TW-T506S 边缘计算平台概述

- 内嵌 NVIDIA® Jetson Xavier NX
- M.2 KEY M (PCIex4 NVMe 2280)
- 支持多种接口(如 CAN/USB/POE 以太网/串口/I2C/GPIO 等)
- 支持双频 WiFi/5G 模组
- 支持 4xPoEPSE 供电端 1000BASE-T RJ45 端口和 1xPoEPD 受电端 1000BASE-T RJ45 端口
- 无风扇被动散热设计
- 内置 Ubuntu 18.04 系统和 Jetpack SDKs
- 宽压 12-36V 凤凰端子输入，LNA1 端口支持 PoE 输入供电。



产品规格

处理器及核心模块

Processor	NVIDIA Jetson NX
AI	21TOPS
CPU	6-core NVIDIA Carmel ARM®v8.2 64-bit CPU 6MB L2 + 4MB L3
GPU	384-core NVIDIA Volta™ GPU with 48 Tensor Cores
Memory	8 GB 128-bit LPDDR4x 51.2GB/s
Storage	1x 16GB eMMC 5.1 1x M.2 Key M NVMe 2280
DL Accelerator	2x NVDLA Engines

编解码

Video Encode	2x 4Kp30 6x 1080p60 14x 1080p30 (H.265 & H.264)
Video Decode	2x 4Kp60 4x 4Kp30 12x 1080p60 32x 1080p30 (H.265) 2x 4Kp30 6x 1080p60 16x 1080p30 (H.264)

接口

	Interface	Quantity	Note
Network	Ethernet	5×RJ45 Gigabit Network port	Alternative RJ45 and waterproof port
	WIFI	1	2.4G/5.8G 300Mbps
Video output	HDMI	1x HDMI 2.0 @ 4Kp60	
USB	USB	4×USB 3.0TYPE A	USB 5V, 1A
	USB-OTG	1xUSB2.0 Micro	
I/O	GPIO	4xGPIO	独立 3.3V TTL 电平 GPIO
	CAN	1xCAN 2.0b	With CAN chip
	UART	1xRS232	DB9 Terminal
	I2C	1	3.3V
User Expansion	TF	1xTF Slot	MicroSD card supported
	M.2	1×M.2 M Key	PCIE NVME 2280 SSD
Function Key	Reset KEY	1	Button
	Recovery KEY	1	Button

供电

Power Supply	Spec
Input Type	1xDC
Input Voltage	Wide input 12-36V DC
Typical consumption	30W

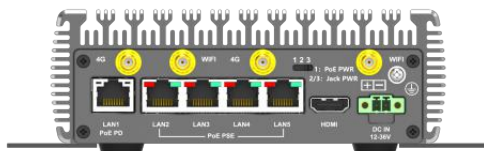
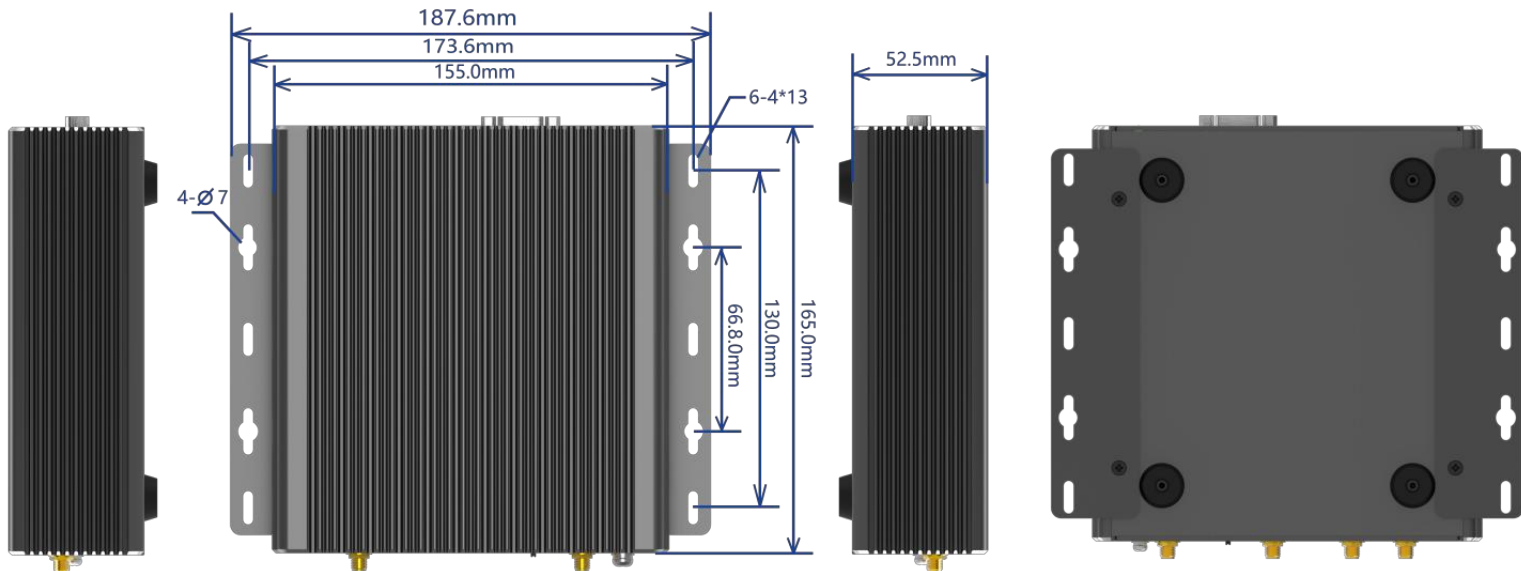
结构

Mechanical	Spec
Dimensions (W×D×H)	155mm×165mm ×52.5mm
Weight	1.5Kg

环境

Environmental	Spec
Operating Temperature	-20°C-65°C
Storage Humidity	10%-90% non-condensing

尺寸



WIGHT:1.5KG
 CHASSIS DIMENSIONS:187.6mm(W)*
 165.0mm(D)*52.5mm(H)

服务与支持

技术支持

如果您遇到问题，或者您认为您的产品有缺陷，请带着您的问题访问我们官网，浏览我们常见问题一栏以查找常见问题的解决方案，也可电话或微信联系我们，我们会及时根据您的需求做出相应的工作安排，为您排忧解难。

保修

保修期：图文设备保修期为自购买之日起一年。 保修条例：保修期内产品，若出现非人为损坏的故障将进行免费保修。请通过购买平台客服对话以及电话联系获取保修协助。

功能介绍

系统介绍

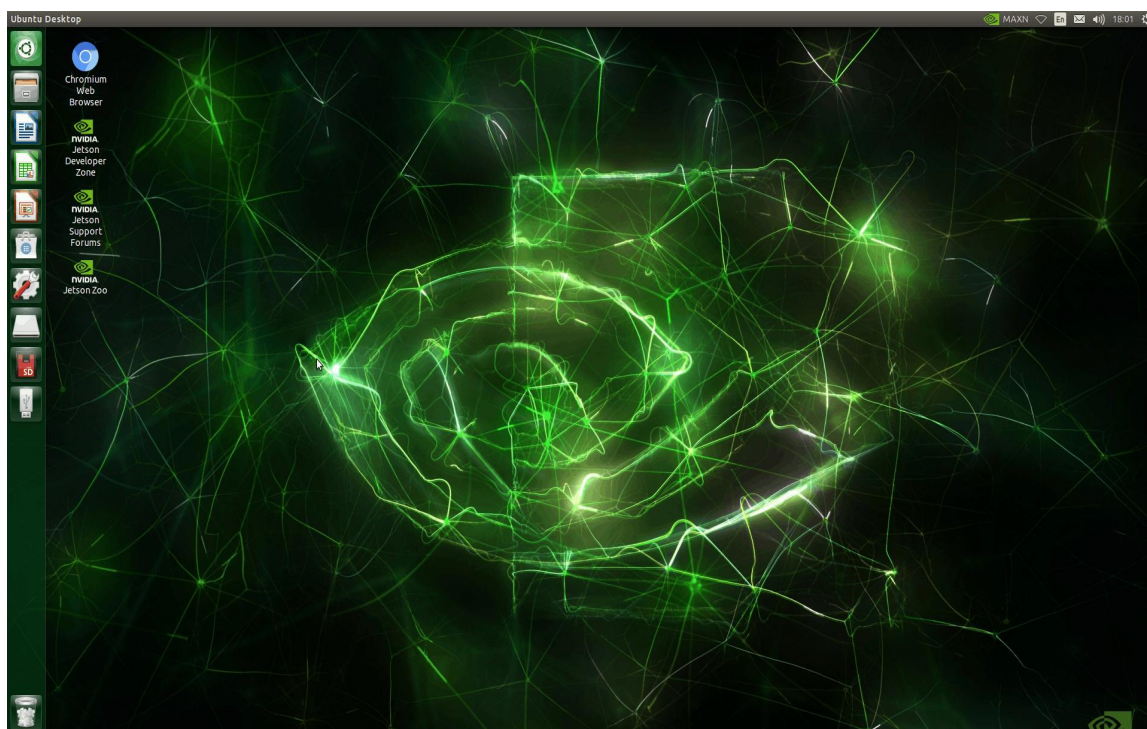
T506S 设备采用 Ubuntu 18.04 系统。默认用户名：nvidia 密码：nvidia 我司未设置 root 用户名和密码,如需进入 root 用户,请执行如下命令进行操作:sudo-s
输入密码:nvidia

系统重刷

请参考后续系统安装章节,里面详细介绍刷机方式及方法.

开关机

开机：TW-T506S 设备默认开机模式为上电自启动。插入电源，并将显示器通过 HDMI 接口与设备相连，开机画面如图所示：



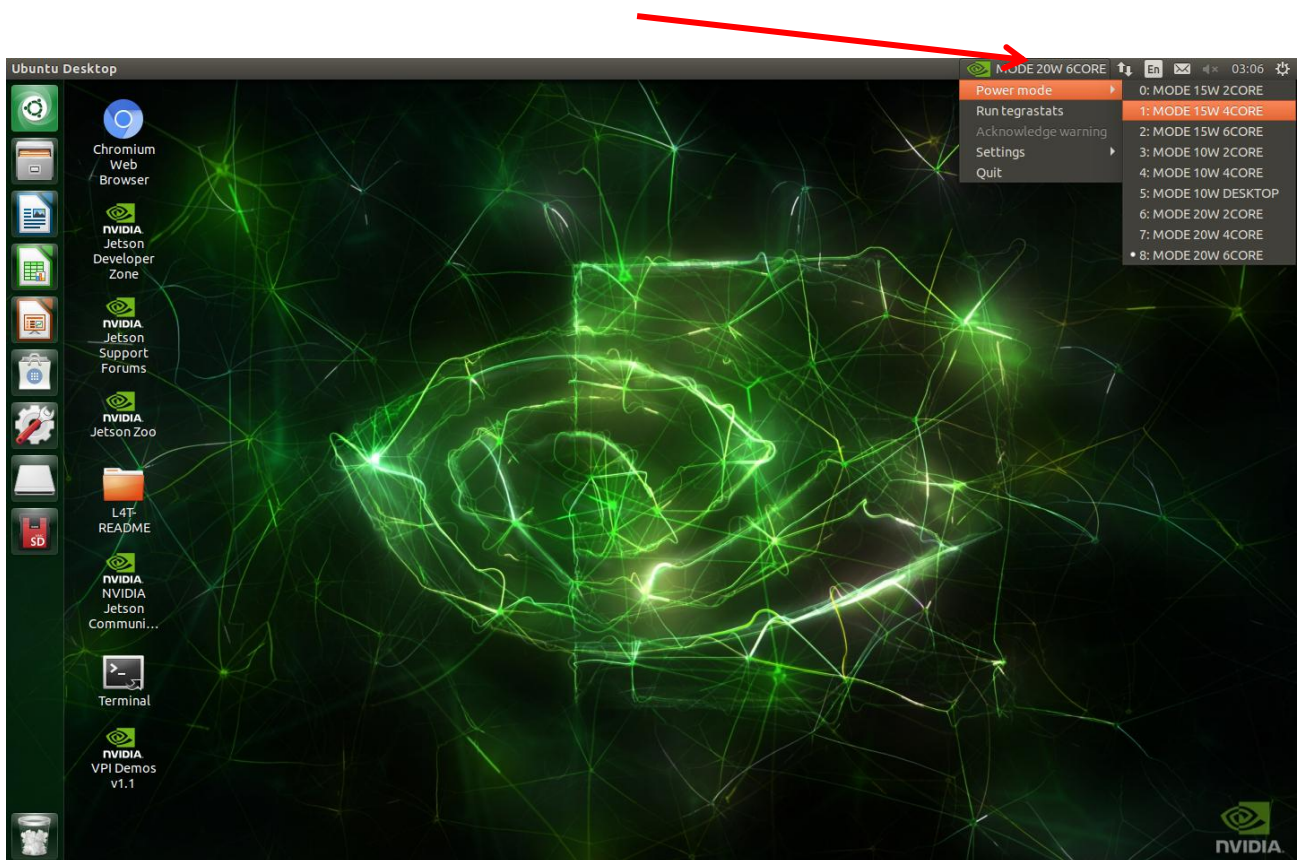
开机画面

关机：长按 POWER 键关机,或在命令行中执行 `sudo poweroff`,

重启：在命令行中执行 `$ sudo reboot`，完成重启;也可以通过按键 reset 键重启(按一下即可)

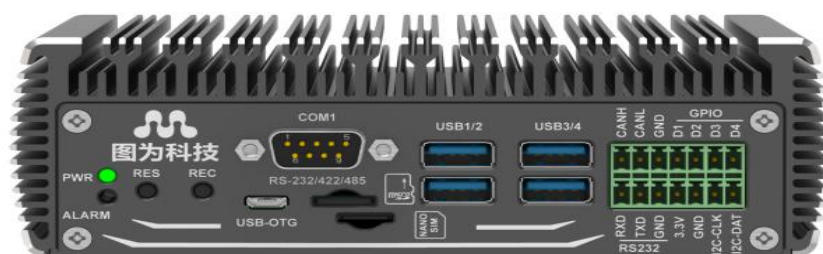
工作模式切换

不同工作模式,使用 cpu 核心和功率不同,可按照自身需求进行选择,点击箭头所指选项,进行模式选择(下图举例为 jetson nx 模块的工作模式)



接口说明

正面接口



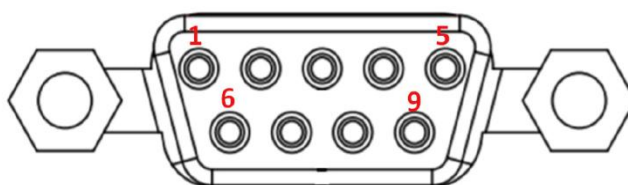
TW-506S 正面接口示意图

接口	接口名称	接口说明
RES	恢复按钮	单独使用为复位重启,搭配 REC 进入 recover 模式
REC	复位按钮	单独使用无作用,配合复位键进入 recover 模式
SIM	SIM 卡槽	如图所示方向安装 SIM 卡
TF	TF 卡槽	如图所示方向安装 TF 卡
COM1	RS232	标准 RS232 串口
USB-OTG	Micro usb 接口	可通过连接普通 usb 数据线设备连接进行数据交换
POWER_LED	电源状态指示灯	接通电源后: 指示灯为绿色常亮
ALARM_LED	故障状态指示灯	无实际作用
USB	4xUSB	USB3.0 接口, 向下兼容 USB2.0 接口 支持超速、高速及低速模式
GPIO	多功能接口	CAN. GPIO, RS232, I2C

注:recover 模式:即下载模式,主要用于重新安装系统以及使用 SDK manager 安装部分 sdk,本设备进入 recover 模式的方法为:先按住 REC 按钮,不松手再按住 RES 按钮,2 秒后松开 RES 按钮,最后松开 REC 按钮,主机终端输入 lsusb 查看是否有 nvidia corp,有则表示进入成功,无则检查 type-c 数据线是否连接好,以及主机端 usb 是否连接好,按钮顺序及时间长短是否正确,主机端 usb 接口建议使用 usb3.0 接口.

串口说明

RS232

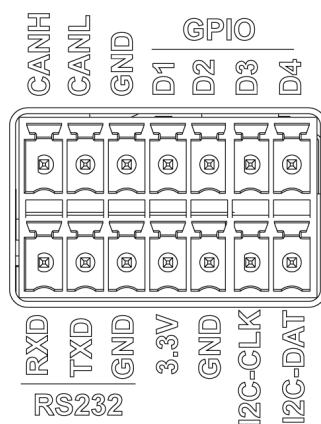


D-SUB 连接器 PIN 顺序

RS-232 连接器 PIN 定义			
PIN#	定义	PIN#	定义
1	DCD	2	RXD
3	TXD	4	DTR
5	GND	6	DSR
7	RTS	8	CTS
9	RI		

后面板 IO

多功能连接器：后面板包括双路 CAN、GPIO，I2C 接口



多功能连接器 PIN 定义

多功能连接器		GPIO-PIN 定义	
PIN#	定义	PIN#	定义
D1	GPIO-TTL3.3V-输入 (GPIO 429)	D2	GPIO-TTL3.3V-输出 (GPIO 424)
D3	GPIO-TTL3.3V-输入 (GPIO 428)	D4	GPIO-TTL3.3V-输出 (GPIO 484)

GPIO 测试:

1. 安装 gpio python library

```
git clone https://github.com/vitiral/gpio.git
```

```
cd gpio/
```

```
sudo python3 setup.py install
```

2. 运行附录 nx_gpio.py 参考示例代码

3. 通过如下查看设置 HI 和 LO

```
sudo cat /sys/kernel/debug/gpio | grep '429'
```

附录: nx_gpio.py 参考示例代码

```
vi nx_gpio.py
```

```
import time
```

```
import gpio
```

```
tst_gpio_pin = 429    #429 对应 T506S 扩展的 D1
```

```
gpio.setup(tst_gpio_pin, gpio.OUT)
```

```
gpio.set(tst_gpio_pin, 0)
```

```
print("Starting now! Press CTRL+C to exit")
```

```
try:
```

```
    while True:
```

```
        gpio.set(tst_gpio_pin, 1)
```

```
        print("TEST High.")
```

```
        time.sleep(5)
```

```
        gpio.set(tst_gpio_pin, 0)
```

```
        print("TEST Low")
```

```
        time.sleep(5)
```

```
finally:
```

```
    gpio.cleanup()
```


CAN 功能测试

如需接入外部 can 设备测试,请将设备 CAN-H 与被测试设备 CAN-H 连接,CAN-L 与被测试设备 CAN-L 连接,连接无误后,请打开文件 home/twork 下的 can 测试脚本,目前脚本中设置的波特率为 80000,如需配置不同波特率,可自行修改脚本内容,一定要设置测试与被测试设备波特率一致:

运行接收脚本:sudo bash can_recieve.sh

```
#!/bin/sh
clear
设置参数
declare -i i=1
str_can0='can0'
str_can1='can1'
str='
#加载 can 模块
sudo modprobe can
sudo modprobe can_raw
sudo modprobe mttcan
#启用 can0
echo 'Enable can0'
#设置 can0 波特率为 80k bps
sudo ip link set can0 type can bitrate 80000
#开启 can0
sudo ip link set up can0
#在 can0 接收端执行 candump,阻塞等待报文,并生成 can 收发信息文档记录
candump can0 2>&1 | tee can0.dump.txt
#等待 60s
sleep 60
#关闭 can0
sudo ip link set can0 down
#退出
exit
```

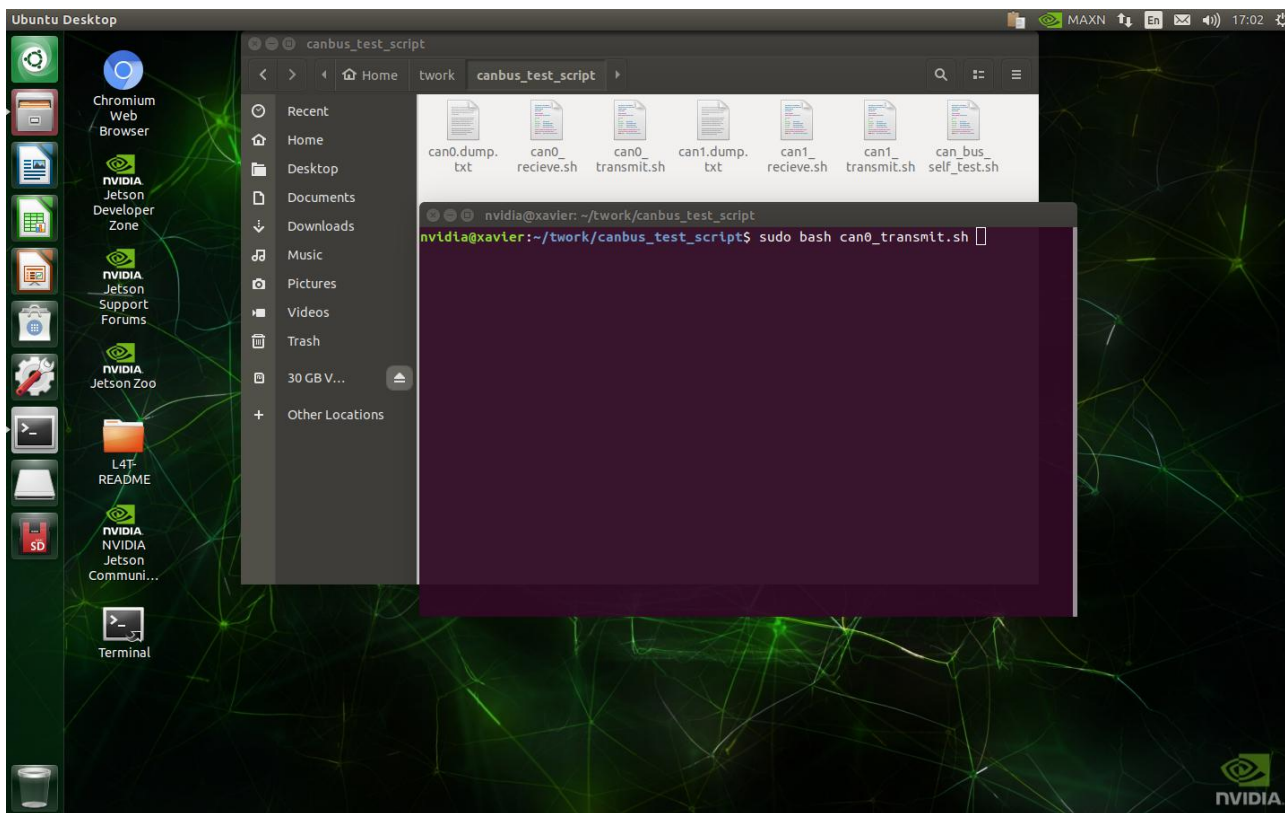
运行发送脚本:sudo bash can_transmit.sh

```
#!/bin/sh
clear
#设置参数
declare -i i=1
str_can0='can0'
str_can1='can1'
str="
#加载 can 相关模块
sudo modprobe can
sudo modprobe can_raw
```

```
sudo modprobe mttcan
#启用 can0
echo 'Enable can0'
#设置 can 波特率为 80k bps
sudo ip link set can0 type can bitrate 80000
#开启 can0
sudo ip link set up can0
#设置 for 循环,设置发送数据次数为 30 次
for i in {1..30}
do
#设置发送报文内容
    cansend can0 1F334455#1122334455667788
    sleep 1

    echo 'can0 已发送, 等待对方接收'
done
#关闭 can0
sudo ip link set can0 down
#退出
exit
```

实际测试如下
终端运行发送脚本



另一端接入下图分析仪,秉承 H-H-L-L 的接线方式



电脑上打开 can 分析仪的测试工具,点击设备操作里的启动设备,设置波特率为 80k bps,接收信息如下

USB-CAN Tool V9.11 - USBCAN-II - SN:序列号: 31500015BA9, 固件版本号: V3.37 - 创芯科技

设备型号(D) 设备操作(O) 参数设定(S) 信息(I) 显示(V) 帮助(H) 语言(L)

CAN发送

帧格式: 标准帧 帧类型: 数据帧 帧ID(HEX): 00 00 00 01 CAN通道: 2 发送总帧数: 1 ID递增

数据(HEX): 00 00 00 00 00 00 00 00 发送消息 发送周期: 10 ms 数据递增

CAN中继状态 **智能过滤** 保存总帧数: 0 停止发送 发送文件

Unused CAN1设置 CAN2设置 打开CAN接收 清空 实时存储

统计数据: 通道1 帧率R: 1 帧率T: 0 **统计数据: 通道2** 帧率R: 0 帧率T: 0

序号	系统时间	时间标识	CAN通道	传输方向	ID号	帧类型	帧格式	长度	数据
00008	17:02:28.579	0xCDC676	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88
00009	17:02:29.570	0xCDEDD3	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88
00010	17:02:30.589	0xCE1521	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88
00011	17:02:31.579	0xCE3C61	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88
00012	17:02:32.599	0xCE63A3	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88
00013	17:02:33.619	0xCE8AF8	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88
00014	17:02:34.609	0xCEB23B	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88
00015	17:02:35.630	0xCED984	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88
00016	17:02:36.619	0xCF00D8	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88
00017	17:02:37.638	0xCF282C	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88
00018	17:02:38.659	0xCF4F7D	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88
00019	17:02:39.649	0xCF76DC	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88
00020	17:02:40.670	0xCF9E34	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88
00021	17:02:41.659	0xCFC587	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88
00022	17:02:42.679	0xCFECCB	ch1	接收	0x1F334455	数据帧	扩展帧	0x08	x 11 22 33 44 55 66 77 88

串口测试方法:

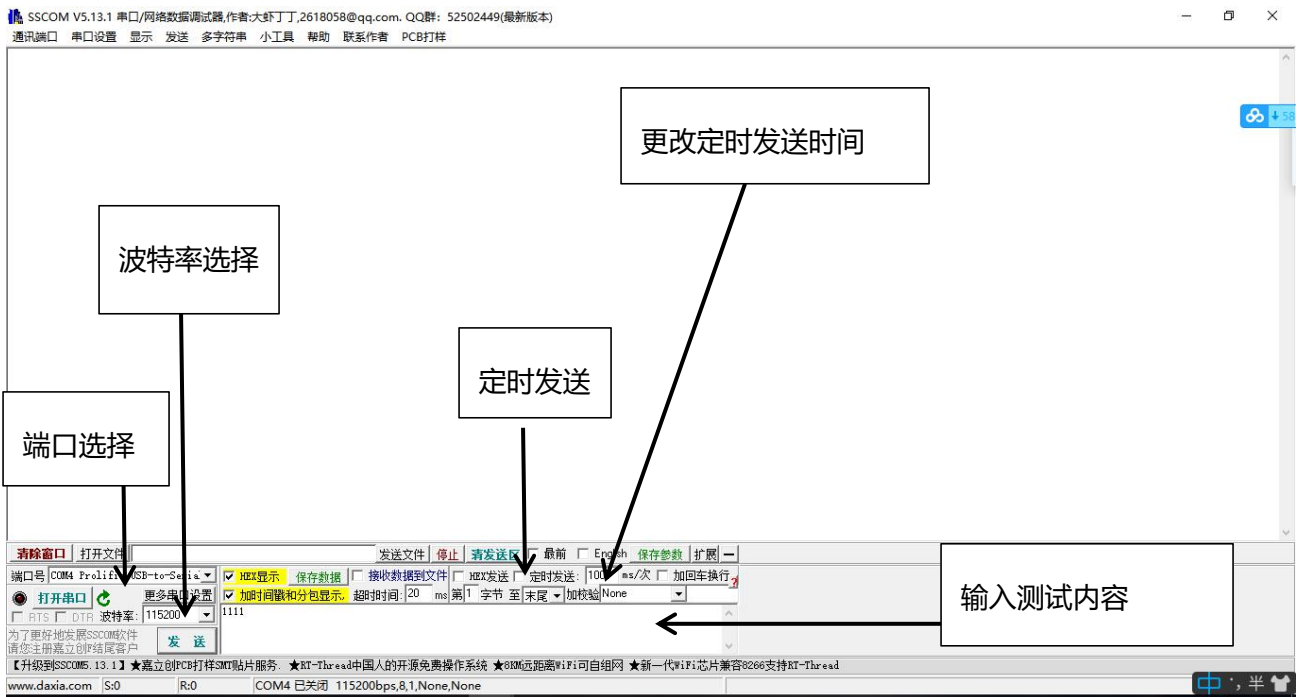
我司使用如图所示 usb 转串口线对 TW-506S 设备进行相应测试, 串口均为标准串口, 测试需自行购买标准 usb 转串口线, 我司出厂产品不附赠此线,



USB=RS232

RS485 测试:

步骤一:使用 usb-RS232 转接线接头连接 TW-506S 设备 COM1, usb 端连接电脑, 电脑端开启串口测试工具 sscom(下面有下载链接)



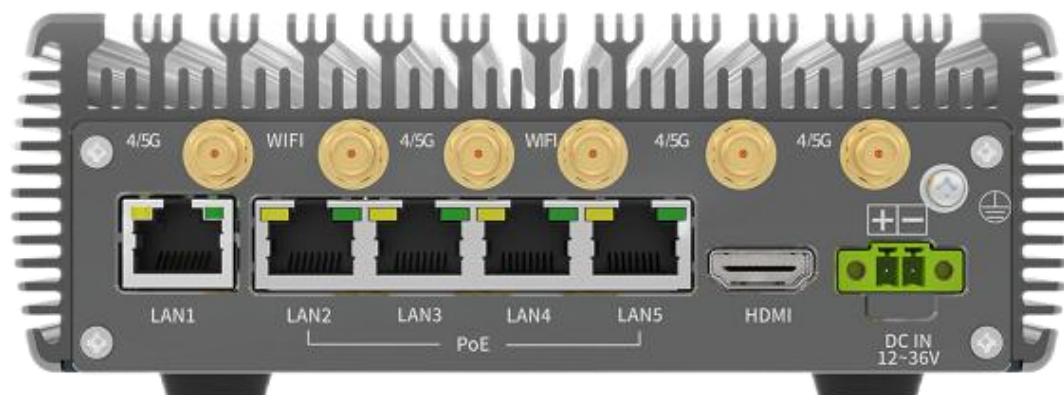
端口号进行选择, 波特率选择 115200 , 输入框输入测试内容, 先修改定时发送时间, 最后点击定时发送

步骤二:TW506S 设备终端输入 sudo cutecom , 打开 cutecom(默认设置即可), 串口号选择 ttyTHS0, 打开串口, 查看对话框是否有收到正确信息(除中文乱码外, 数字出现乱码或传输数据不正确则表明不正常, 传输数据一致则表明该串口通信正常)

测试串口工具下载链接:链接: https://pan.baidu.com/s/1RdwcRGH1_28_X6L20YJqXw

提取码: pf3d

背面接口

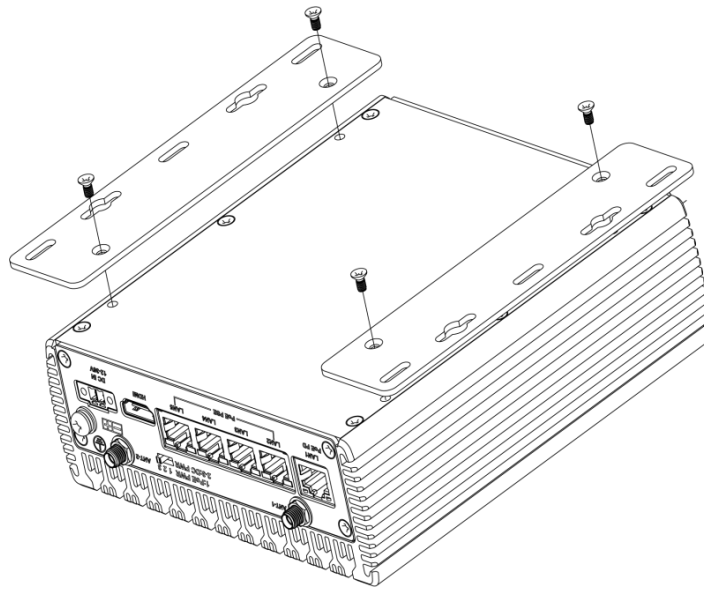


TW-T506S 背面接口示意图

接口	接口名称	接口说明
DC IN	凤凰端电源接口	凤凰端 DC 电源输入 12-36V
WIFI-1	WIFI 天线接口	外接 WIFI 天线
5G-1,5G-2,5G-3, 5G-4	5G 天线接口	外接 5G 天线
多功能接口	多功能接口	包含 CAN0\CAN1\I2C
LAN(RJ45)x5	以太网千兆口	独立千兆网口, 向下兼容百兆网口 (LAN1 支持 POE PD 供电)
HDMI	HDMI 接口	标准 HDMI2.0 接口
DC IN	电源接口	输入电源 12-36V
螺母	接地螺母	使设备接地

扩展设备的安装方式

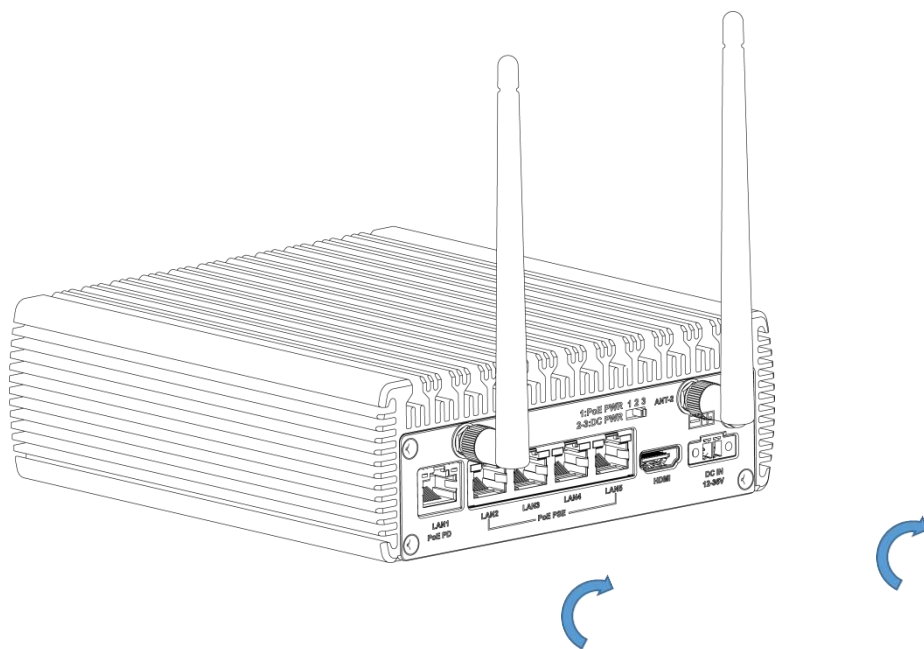
安装盒子支架



盒子支架安装

盒子允许它安装在任何方便的空间在工作环境中，并保持四周通风的环境。

1. 安装支架，用六颗螺丝穿过每个支架上的孔固定支架到盒子底板。
2. 将盒子固定在您想要挂载盒子的地方。

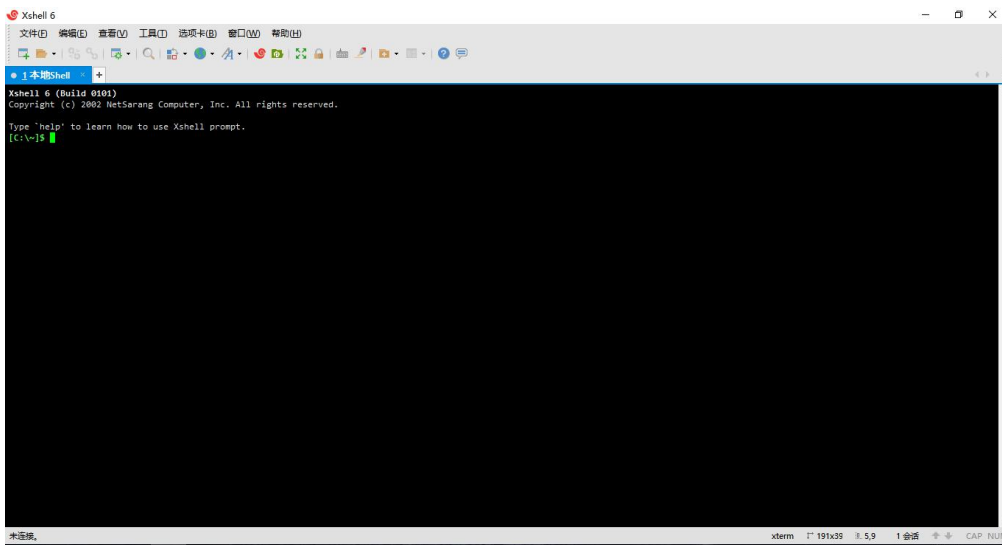


wifi/4G 天线安装

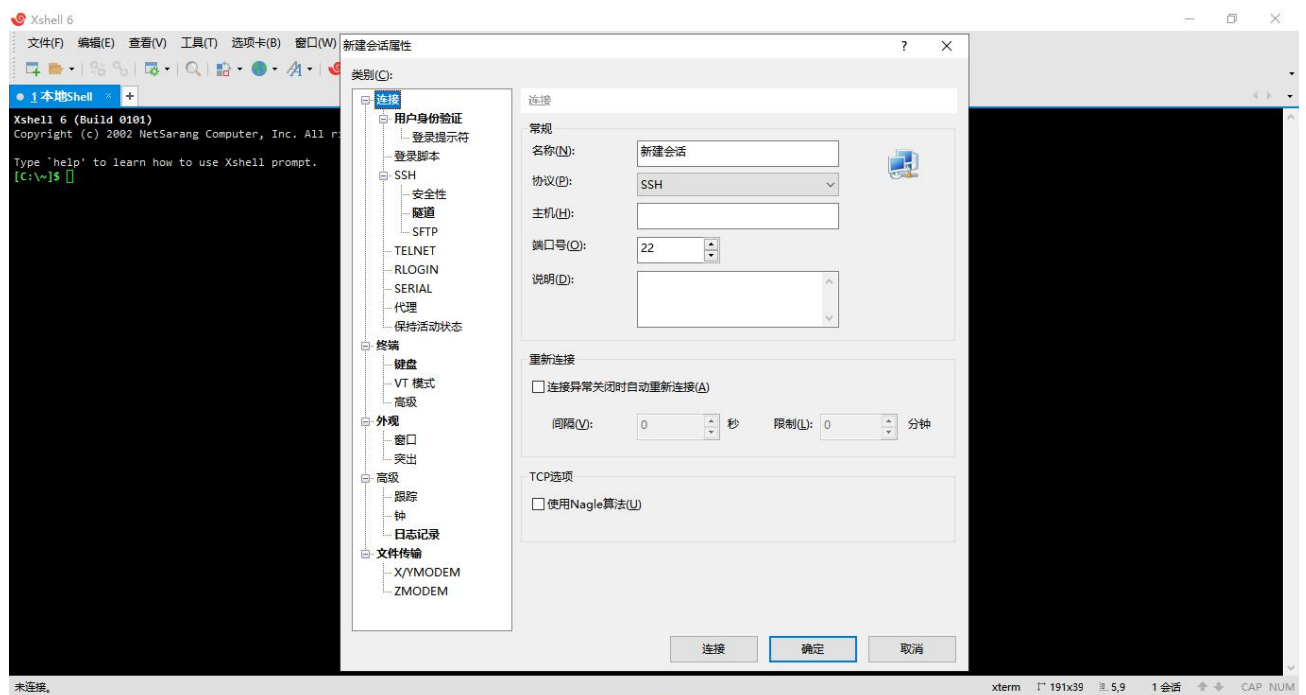
Xshell 的使用

Xshell 是一个强大的安全终端模拟软件，它支持 SSH1, SSH2, 以及 Microsoft Windows 平台的 TELNET 协议。Xshell 通过互联网到远程主机的安全连接以及它创新性的设计和特色帮助用户在复杂的网络环境中享受他们的工作。Xshell 可以在 Windows 界面下用来访问远端不同系统下的服务器，从而比较好的达到远程控制终端的目的。xshell 不是必需品，但是它能更好的辅助我们使用设备。他可以将你的 Windows 系统和你的 Ubuntu 系统链接起来，让你在 Windows 系统下，去操作你的 Linux 系统。安装 xshell 可以在网上百度搜索下载安装即可。(当智盒无法进入桌面系统的情况时，也可以通过 xshell 来进行远程控制，修改配置上的错误)

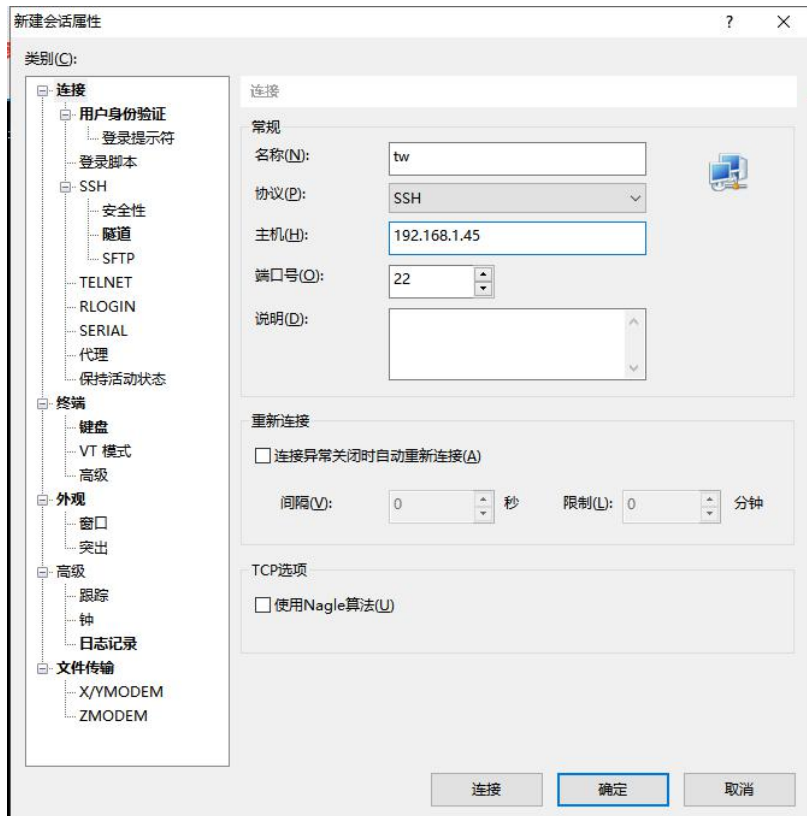
1. 打开 xshell



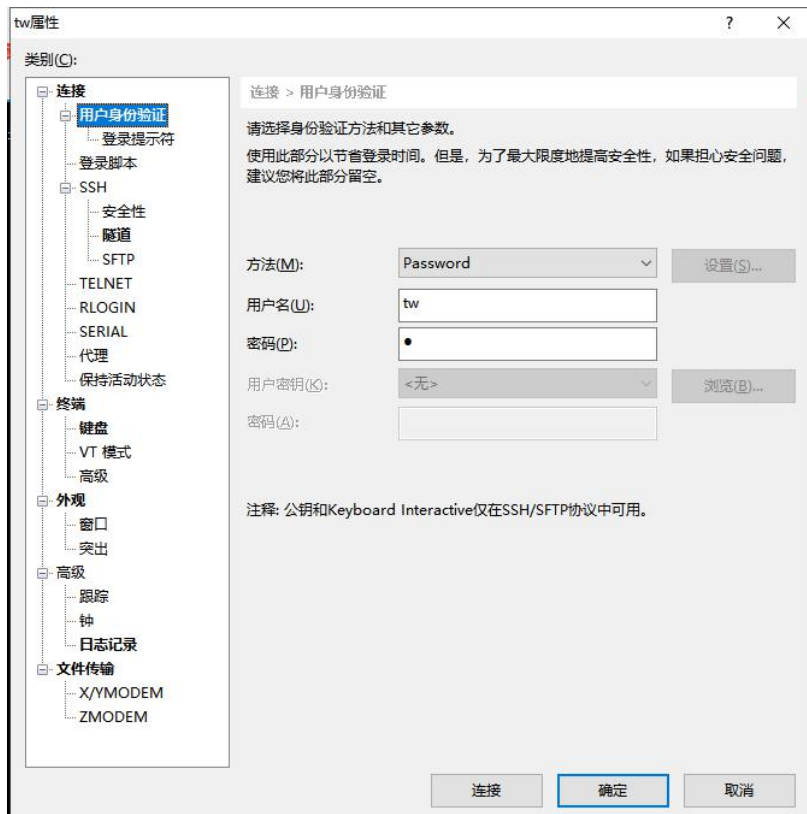
2. 点击新建



3. 填写名称以及主机 ip(正常情况下可以通过网络 ip 来连接,若不知道 ip 的情况下,可以通过 usb 数据线连接电脑和设备 OTG 口, ip 填固定 ip192.168.55.1 来进行连接)



4. 输入用户和密码



5. 点击连接进入命令行界面

```
Xshell 6 (Build 0101)
Copyright (c) 2002 NetSarang Computer, Inc. All rights reserved.

Type `help` to learn how to use Xshell prompt.
[~:~]$

Connecting to 192.168.1.116:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+J'.

Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-72-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

40 packages can be updated.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Wed Apr 28 14:11:59 2021 from 192.168.1.232
tw@tw:~$
```

6. 通过 xshell 远程操作 jetson 设备.

```
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-72-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

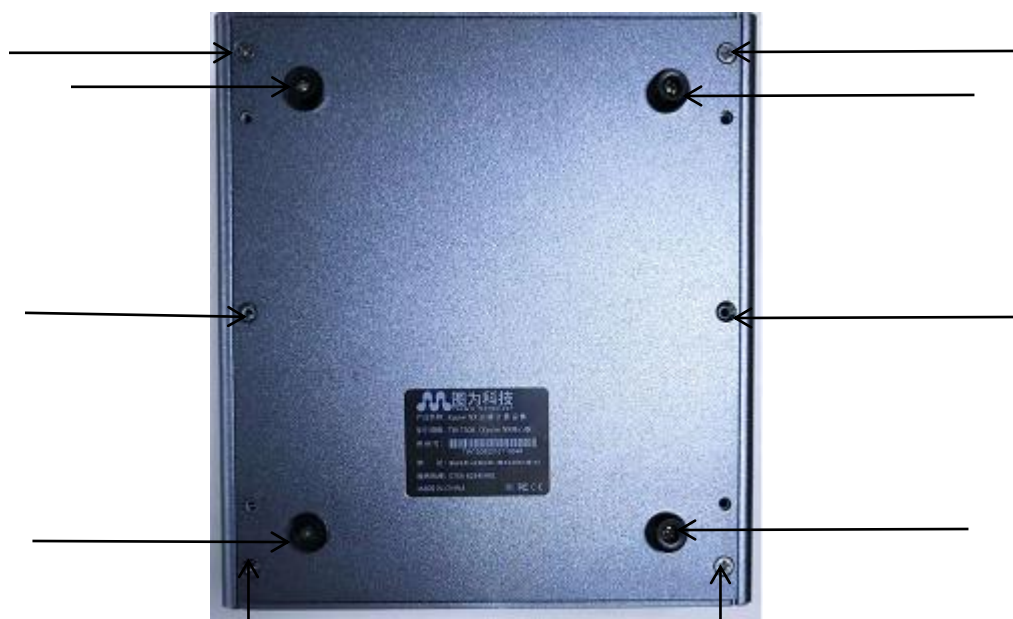
40 packages can be updated.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Wed Apr 28 14:11:59 2021 from 192.168.1.232
tw@tw:~$ ll
total 7609912
drwxr-xr-x 23 tw tw      4096 4月 29 11:49 ./
drwxr-xr-x  4 root root    4096 1月 26 11:40 ../
-rw-r--r--  1 tw tw      3850 2月 23 17:35 .apollo_agreement.txt
-rw-r--r--  1 root root      89 4月 28 09:44 a.txt
-rw-----  1 tw tw     44130 4月 29 09:25 .bash_history
-rw-r--r--  1 tw tw      220 1月 21 11:27 .bash_logout
-rw-r--r--  1 tw tw     3771 1月 21 11:27 .bashrc
drwx----- 19 tw tw     4096 3月 25 16:21 .cache/
-rw-rw-r--  1 tw tw        0 3月 25 17:10 clear
drwx----- 21 tw tw     4096 4月 27 11:17 .config/
-rw-rw-r--  1 tw tw        0 3月 25 17:10 create
drwx-----  3 root root    4096 3月 25 16:21 .dbus/
drwxr-xr-x  2 tw tw     4096 2月 24 13:57 Desktop/
drwxr-xr-x  2 tw tw     4096 1月 21 11:51 Documents/
drwxr-xr-x  7 tw tw     4096 4月 28 14:20 Downloads/
-rw-r--r--  1 tw tw     8980 1月 21 11:27 examples.desktop
drwx-----  2 tw tw     4096 1月 28 12:01 .gconf/
drwx-----  3 tw tw     4096 3月  3 10:16 .gnupg/
-rw-----  1 tw tw     11174 4月 25 08:57 .ICEauthority
-rw-r--r--  1 tw tw    7792314278 3月 25 17:02 jetson-xavier-rx.img.gz
drwx-----  3 tw tw     4096 1月 21 11:51 .local/
```

4G 模块安装

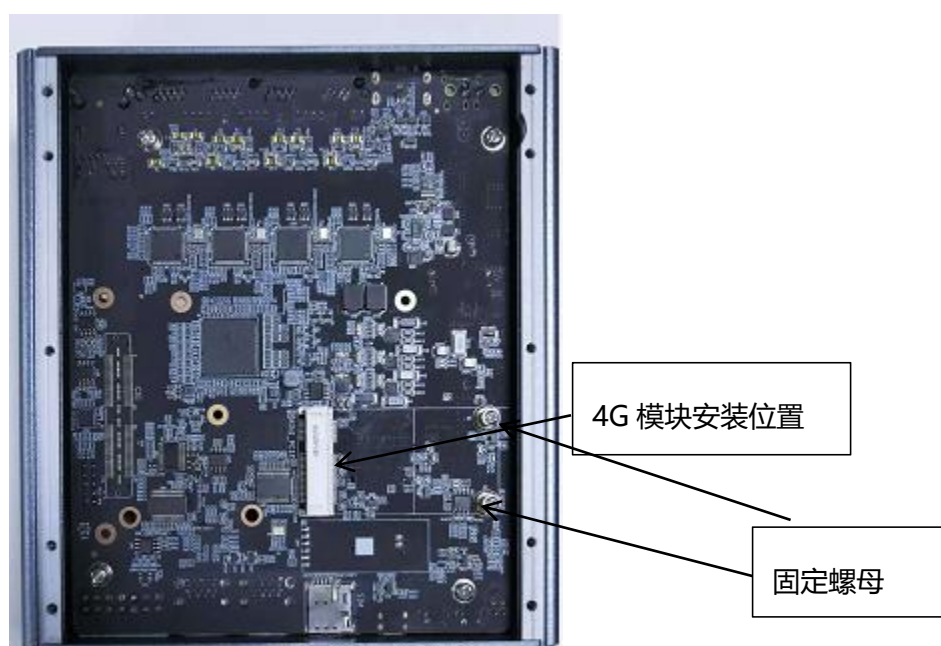
本产品标准件不搭载 4G 模块, 该模块为选配安装, 如需自己独立安装请参考本节内容. (建议拆卸前将原装螺母的位置拍照, 进行相关记录, 螺母螺柱使用盒子或者其他物品收集好避免丢失, 过程需小心谨慎, 切勿暴力拆卸, 如因拆卸不当造成的损失, 我司不承担相应责任)

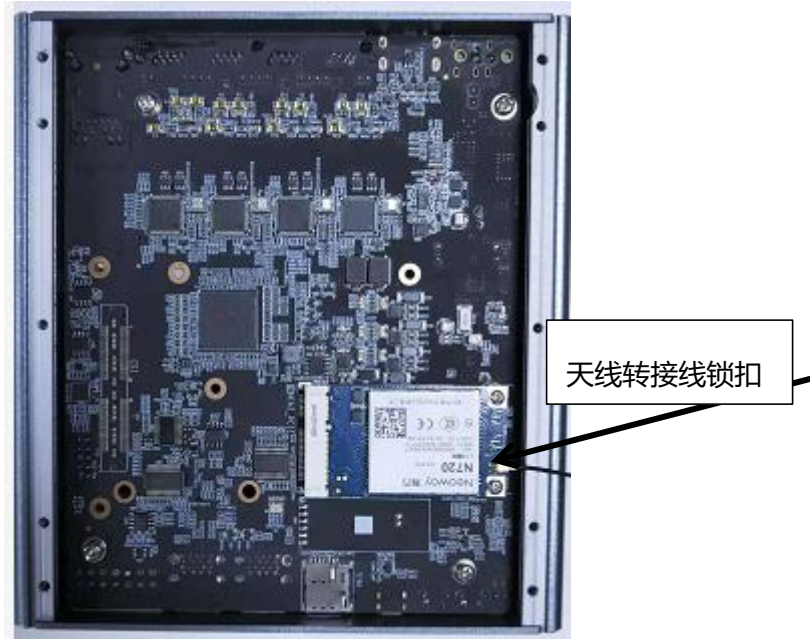
步骤一: 如图箭头所示, 准备好十字螺丝刀, 依次拆下脚垫螺母, 4 角固定螺母, 使用镊子或者螺丝刀将底层盖板取下;



底板拆卸示意图

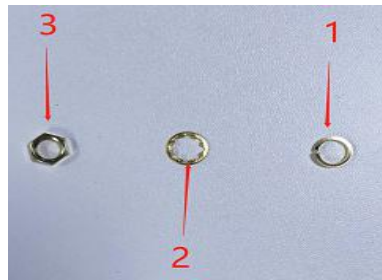
步骤二: 取下 4G 模块底座的固定螺母, 将 4G 模块装入如图所示的位置, 找准模块上锁扣天线的位置, 将天线转接线安装在模块上;





模块安装示意图

步骤三: 如图所示依次将天线转接头固定好, 外接天线棒;



零件安装顺序图

步骤四: 将 SIM 卡插入设备箭头标注的地方 (SIM 卡一定要在不连接电源的情况插入, 避免造成不必要的故障);

步骤五: 设备开机. 输入 lsusb, 查看是否识别 4G 模块, 确认识别拨打手机号测试 4G 模块通信, 如需配置 4G 网络参考 4G 模块配置方法一节进行相关设置.



SIM 卡槽

```
nvidia@Xavier-4:~$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 093a:2510 Pixart Imaging, Inc. Optical M
Bus 001 Device 005: ID 0461:0010 Primax Electronics, Ltd HP PR1
Bus 001 Device 007: ID 2949:8247
Bus 001 Device 006: ID 0bda:c811 Realtek Semiconductor Corp.
```

4G 拨号联网

步骤一:

将提供的压缩包文件解压后复制到智盒设备中, 使用命令将其复制到/etc/ppp/peers 下, 命令为压缩包

下载路径:

链接: https://pan.baidu.com/s/1iF0iKqKTGYuNTDZ5w_I9Mg

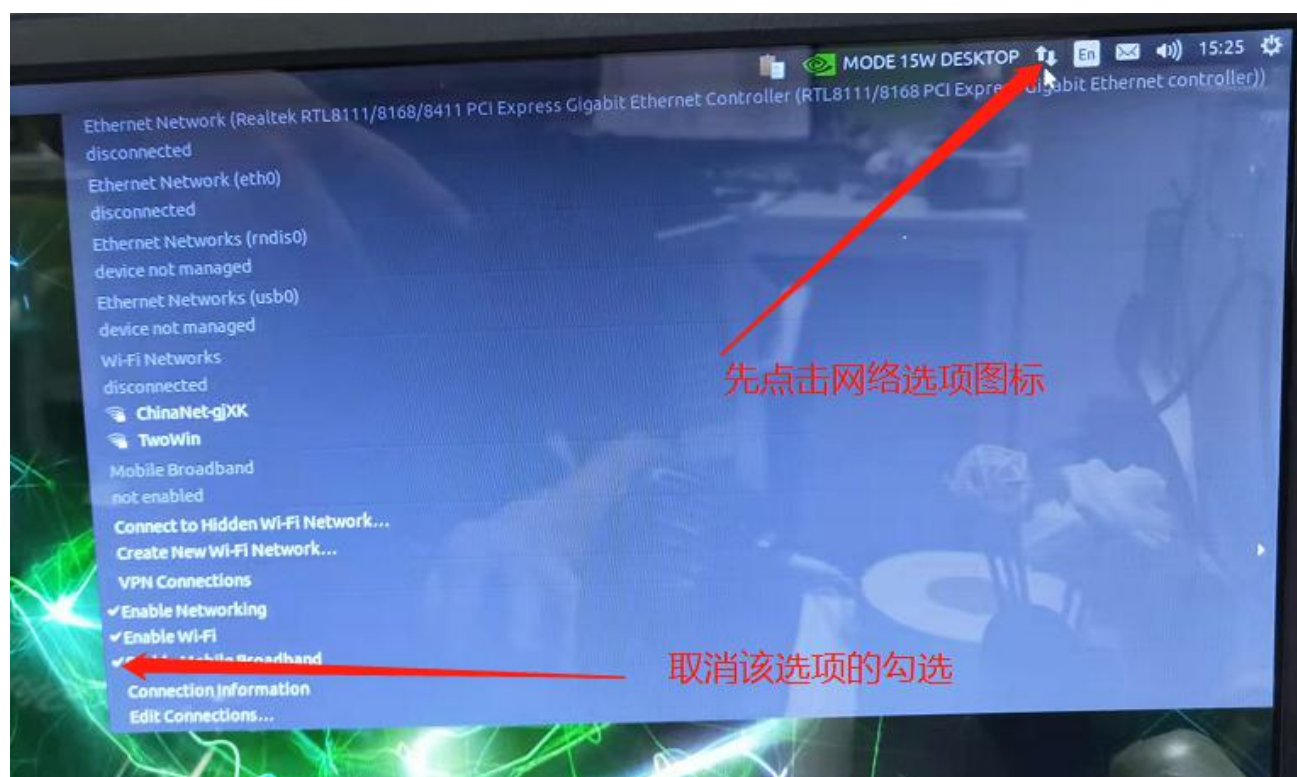
提取码: 33sm

执行命令为:

```
sudo cp -f 4g_dail/* /etc/ppp/peers
```

步骤二:

关闭网络连接里面的 Enable Mobile Broadband 选项, 具体操作可参考下图:



步骤三:

进入/etc/ppp/peers 目录, 找到 n720-ppp-dial.sh 脚本, 第一次执行需要赋予执行权限, 具体操作为:

```
cd /etc/ppp/peers
```

```
sudo chmod +x n720-ppp-dial.sh
```

```
sudo ./n720-ppp-dial.sh
```

如若需要关闭拨号,可通过运行 ppp-kill. sh 脚本结束拨号

如需实现开机 4G 自动连网功能,需要先设置 rc.local 启动脚本,以增加系统自启动时的应用

步骤一:

添加 rc-local.service

```
sudo ln -fs /lib/systemd/system/rc-local.service /etc/systemd/system/rc-local.service
```

```
sudo vi /etc/systemd/system/rc-local.service
```

添加:

```
[Install]
```

```
WantedBy=multi-user.target
```

```
Alias=rc-local.service
```

步骤二:

编写 rc.local 脚本

```
sudo touch /etc/rc.local
```

```
sudo chmod 755 /etc/rc.local
```

```
sudo gedit /etc/rc.local
```

添加内容可将我们提供的 rc.local 文件文本内容粘贴复制即可,以下为文本内容

```
#!/bin/bash
```

```
LOG_DIR=/var/log/twlog
```

```
mkdir -p $LOG_DIR
```

```
#4g auto dial if register on network. sleep 30s wait for 4g module prepared
```

```
TIME=`date +%Y%m%d%H%M`
```

```
echo $TIME >> $LOG_DIR/ppp-dial.log
```

```
echo "Auto dial" >> $LOG_DIR/ppp-dial.log
```

```
nohup /etc/ppp/peers/n720-ppp-dial.sh >> $LOG_DIR/ppp-dial.log &
```

```
sleep 10
```

```
#Set default gateway
```

```
def_gw=`/sbin/ifconfig ppp0|grep destination|grep -v 127.0.0.1|grep -v inet6 | awk '{print $6}' | tr -d "addr:"`
```

```
#`route -n | grep ppp0 | grep UG | awk '{print $2}'`
```

```
echo $def_gw >> $LOG_DIR/ppp-dial.log
```

```
if [ -n "$def_gw" ]; then
```

```
    #Set default gateway using ppp0/4G
```

```
    sudo route add default gw $def_gw
```

```
else
```

```
    sleep 10
```

```
    def_gw=`/sbin/ifconfig ppp0|grep destination|grep -v 127.0.0.1|grep -v inet6 | awk '{print $6}' | tr -d "addr:"`
```

```
    if [ -n "$def_gw" ]; then
```

```
        sudo route add default gw $def_gw
```

```
    fi
```

```

fi
#Get ppp0 IP
fourg_ip=`/sbin/ifconfig ppp0|grep inet|grep -v 127.0.0.1|grep -v inet6 | awk '{print $2}' | tr -d "addr:"`
if [ -n "$fourg_ip" ]; then
    echo $fourg_ip >> $LOG_DIR/ppp-dial.log
else
    sleep 15
    if [ -n "$fourg_ip" ]; then
        echo $fourg_ip >> $LOG_DIR/ppp-dial.log
    else
        echo "4G no ip $fourg_ip" >> $LOG_DIR/ppp-dial.log
        echo $TIME"-Kill pppd and redial" >> $LOG_DIR/ppp-kill.log
        nohup /etc/ppp/peers/ppp-kill.sh >> $LOG_DIR/ppp-kill.log &
    fi
fi
#end 4g auto dial

#Running maxn mode
/usr/bin/jetson_clocks

exit 0

```

文件默认设置为移动网络联网,如若使用其他运营商的 4g 卡,需修改 n720-ppp-dial.sh 文件

```

QL_DEVNAME=/dev/ttyUSB0
QL_APN=cmnet
QL_USER=guest
QL_PASSWORD=guest

```

修改运营商接入点名称

移动为:cmnet
 联通为:3gnet
 电信为:ctnet

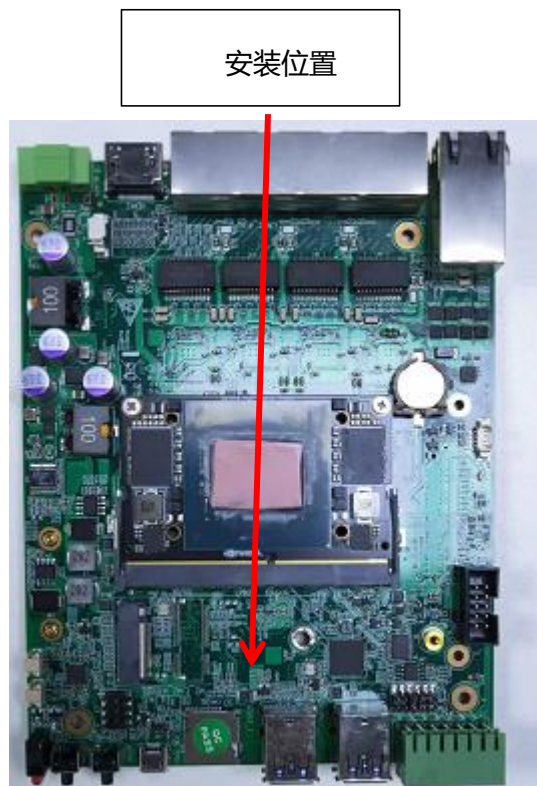
测试 4G 网络是否有成功连接,可通过打开网页或 ping 百度来测试即可.

固态硬盘(SSD)安装

本产品标准件不搭载固态硬盘, 该模块为选配安装, 如需自己独立安装请参考本节内容. (建议拆卸前将原装螺母的位置拍照, 进行相关记录, 螺母螺柱使用盒子或者其他物品收集好避免丢失, 过程需小心谨慎, 切勿暴力拆卸, 如因拆卸不当造成的损失, 我司不承担相应责任)

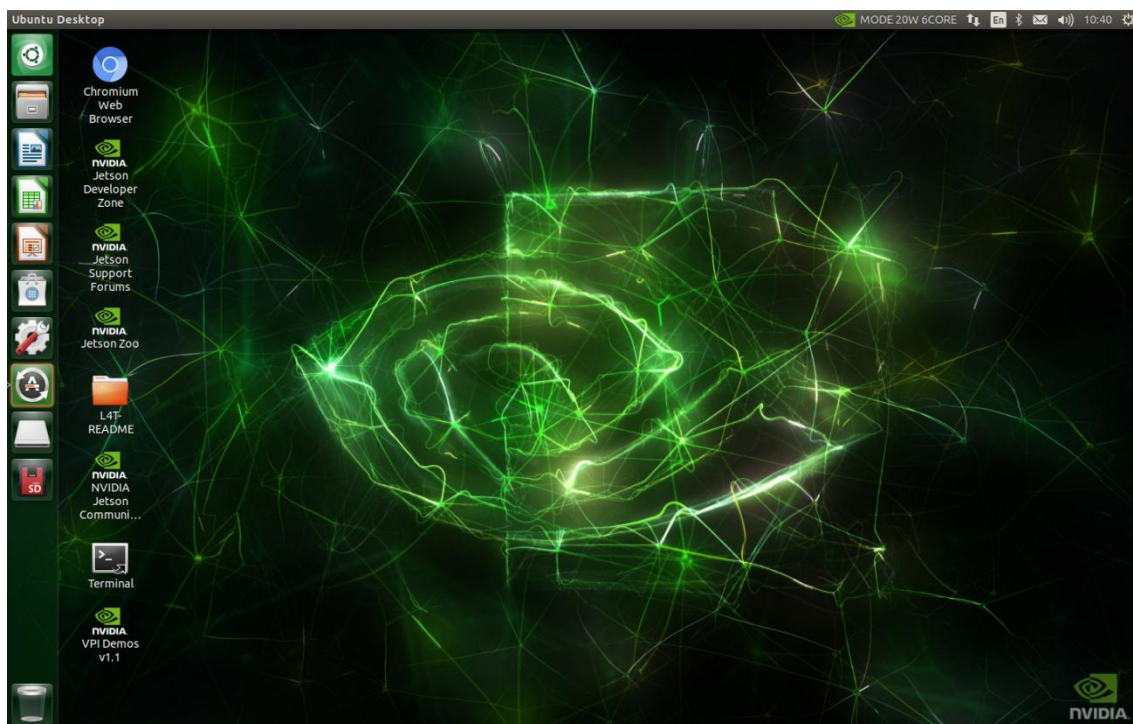
步骤一: 参考 4G 模块安装步骤, 拧下固定主板的 4 颗螺丝后取下主板;

步骤二: 找到如图所示的 SSD 安装位置, 取下固定螺母, 装上 SSD 后锁好螺母, 确认安装无误后, 将设备恢复原状, 安装完毕后参考采用 M.2 Key M SSD 为系统盘一节内容进行相应设置.

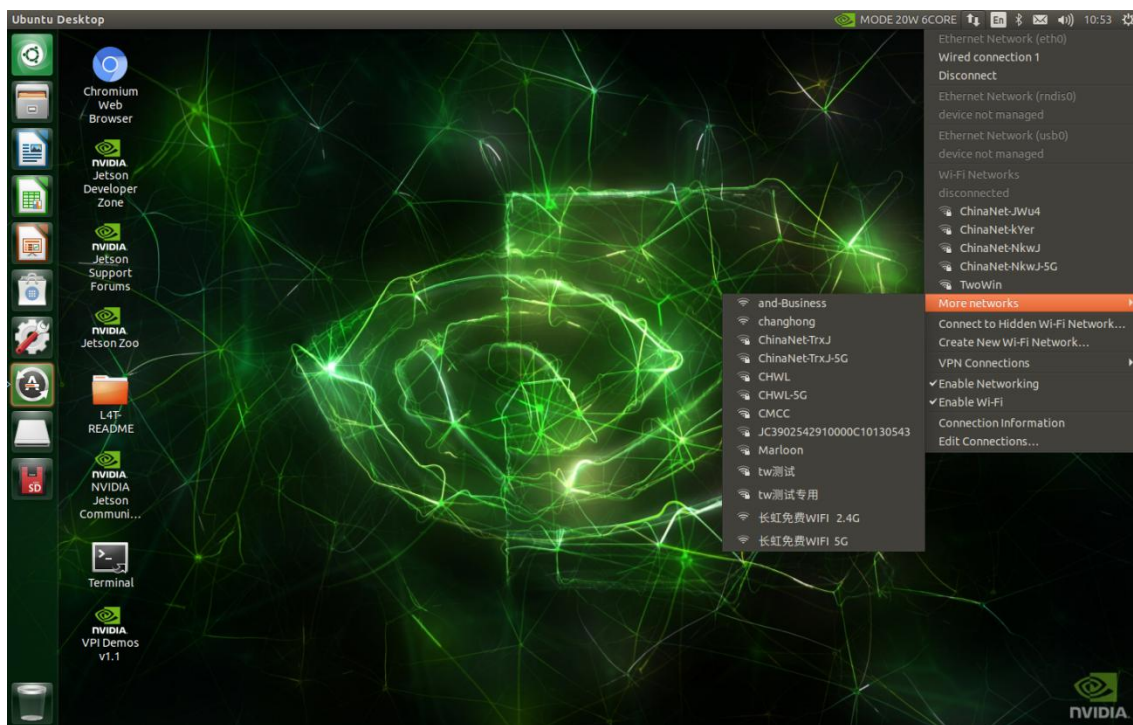


WiFi 热点设定及连接

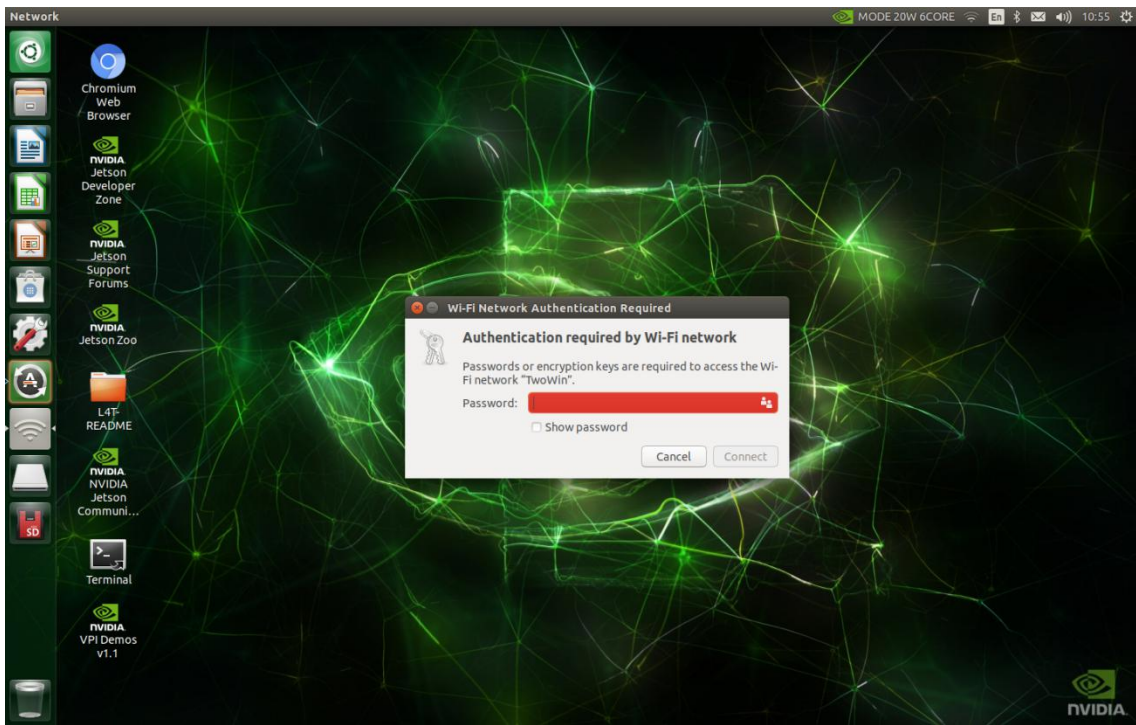
1. 点击箭头所示网络图标



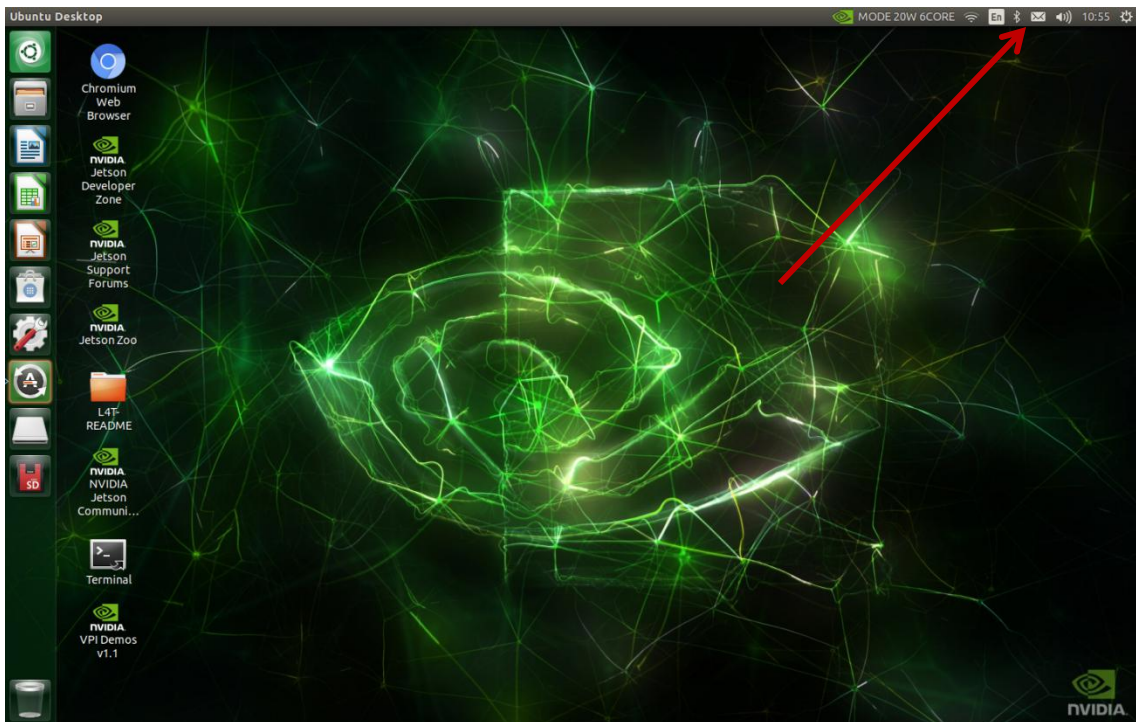
2. 搜寻可连接的 wifi 选项



3. 输入密码, 可勾选下方显示密码选项, 从而检查密码是否输入正确



4. 网络图标变成下图所示, 即表示联网成功



5. 打开网页确认是否网络正常

百度一下, 你就知道 - Chromium

百度一下, 你就知道 x +

百度一下, 你就知道

Chromium isn't your default browser Set as default

新闻 hao123 地图 贴吧 视频 图片 网盘 更多 设置 登录

Baidu 百度

百度一下

百度热搜 > 换一换

- 1 数读十九届六中全会精神
- 2 31省份增本土确诊166例 陕西165例
- 3 稳住农业基本盘做好三农工作
- 4 2021年最后一天
- 5 元旦假期全国大部地区天气晴好
- 6 江歌母亲诉刘鑫案31日宣判取消

关于百度 About Baidu 使用百度前必读 帮助中心 京公网安备11000002000001号 京ICP证030173号 ©2021 Baidu 互联网药品信息服务资格证书(京)-经营性-2017-0020 信息网络传播视听节目许可证 0110516

采用 M.2 Key M SSD 为系统盘

SSD 作用

NVMe SSD 硬盘仅作为系统盘（rootfs 和用户区），系统的启动引导依然是通过 SD 卡或 EMMC，比如升级设备树 dtb 还是在 SD 卡或 EMMC 中。

步骤一、准备 SSD 并格式化为 GPT

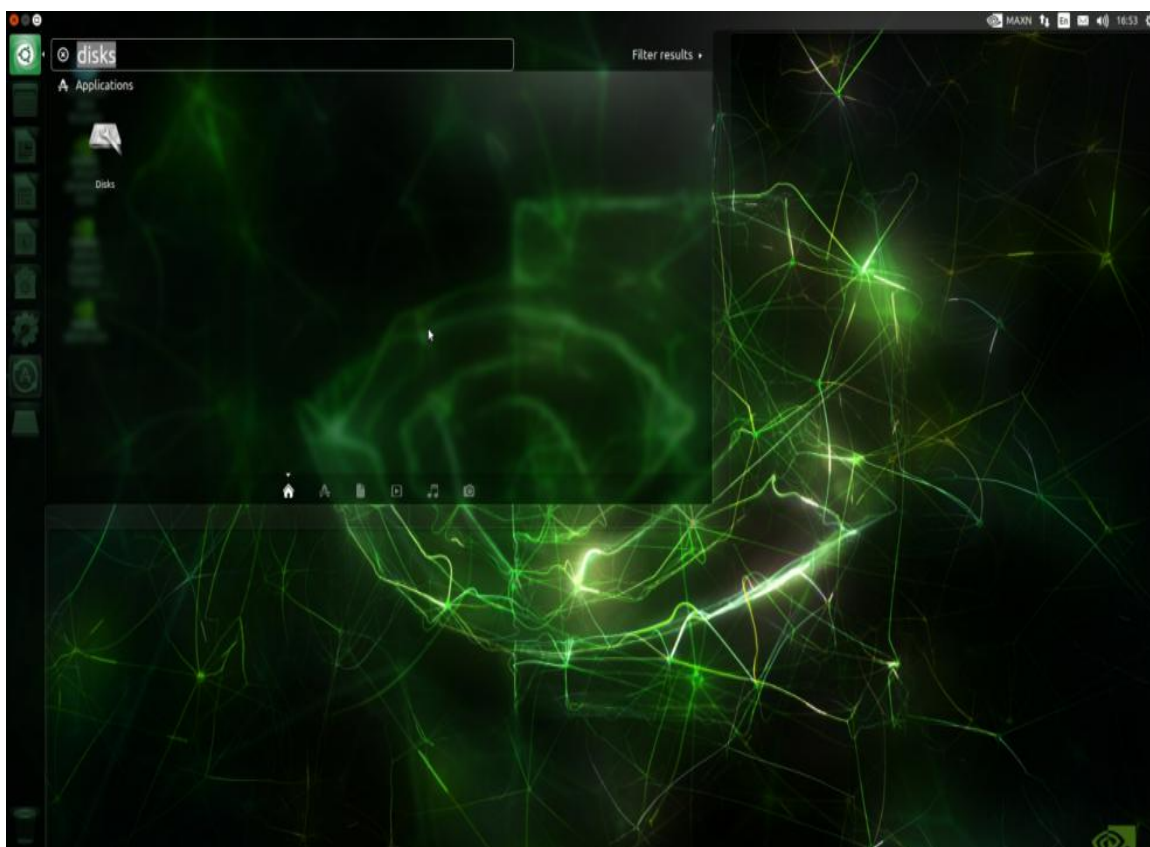
1. 准备 M.2 Key M SSD

《本例采用的是铠侠 CL1-8D128 SSD》

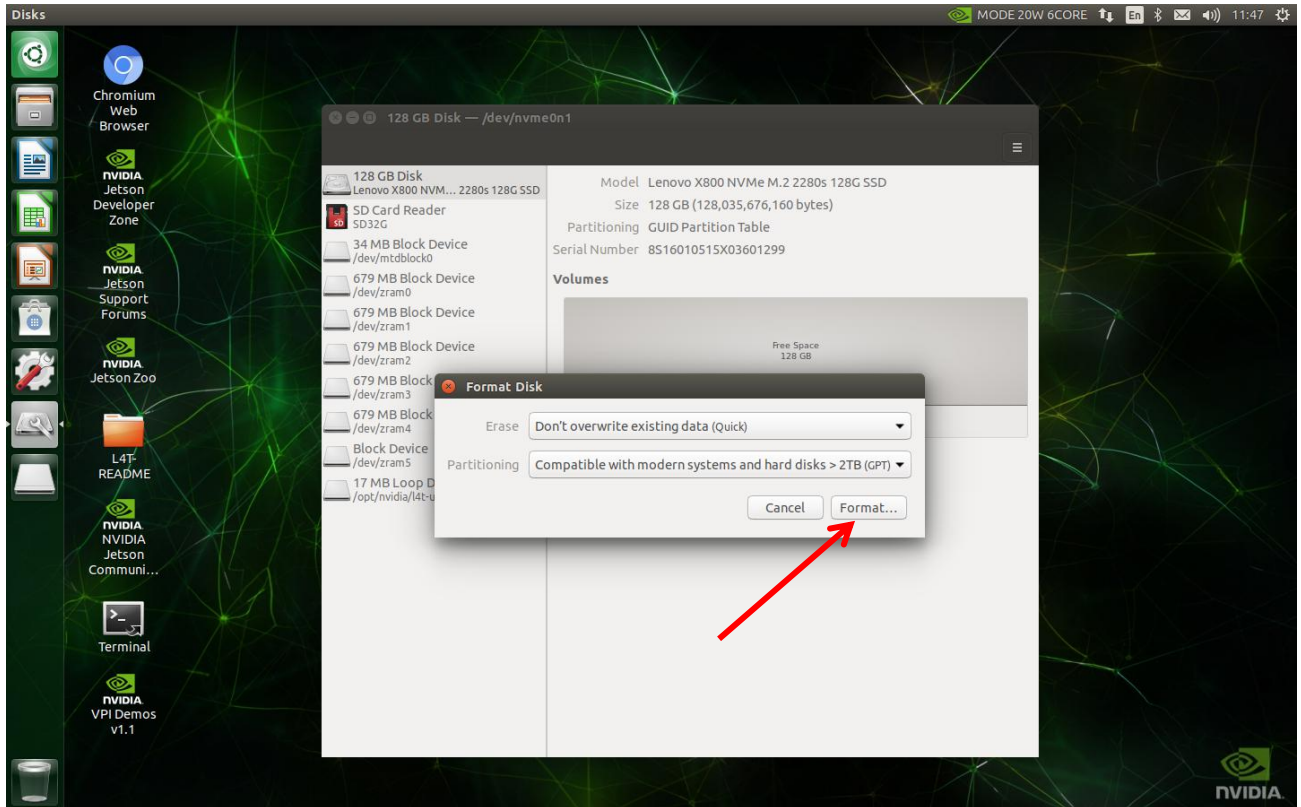
2. 打开 Ubuntu18.04 自带 Disks 工具(), 找到安装的 ssd 硬盘, 首先按键” Ctrl+F” 将其快速格式化为.

3. 具体操作参考下图所示(需严格按照下面方式进行相关操作, 避免造成失误导致设备无法进入桌面系统):

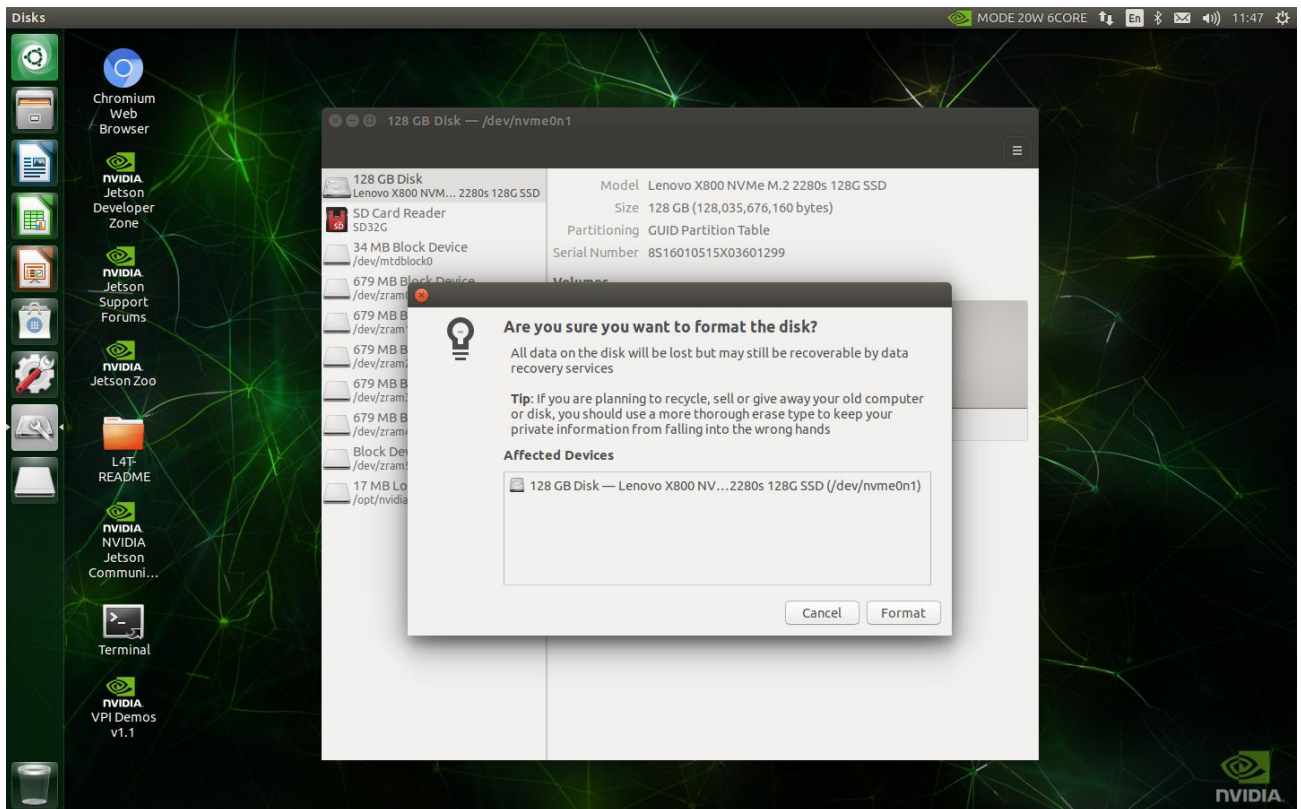
1) 首先打开方式为按 ctrl 键出现搜索框, 搜索” disks” 工具;



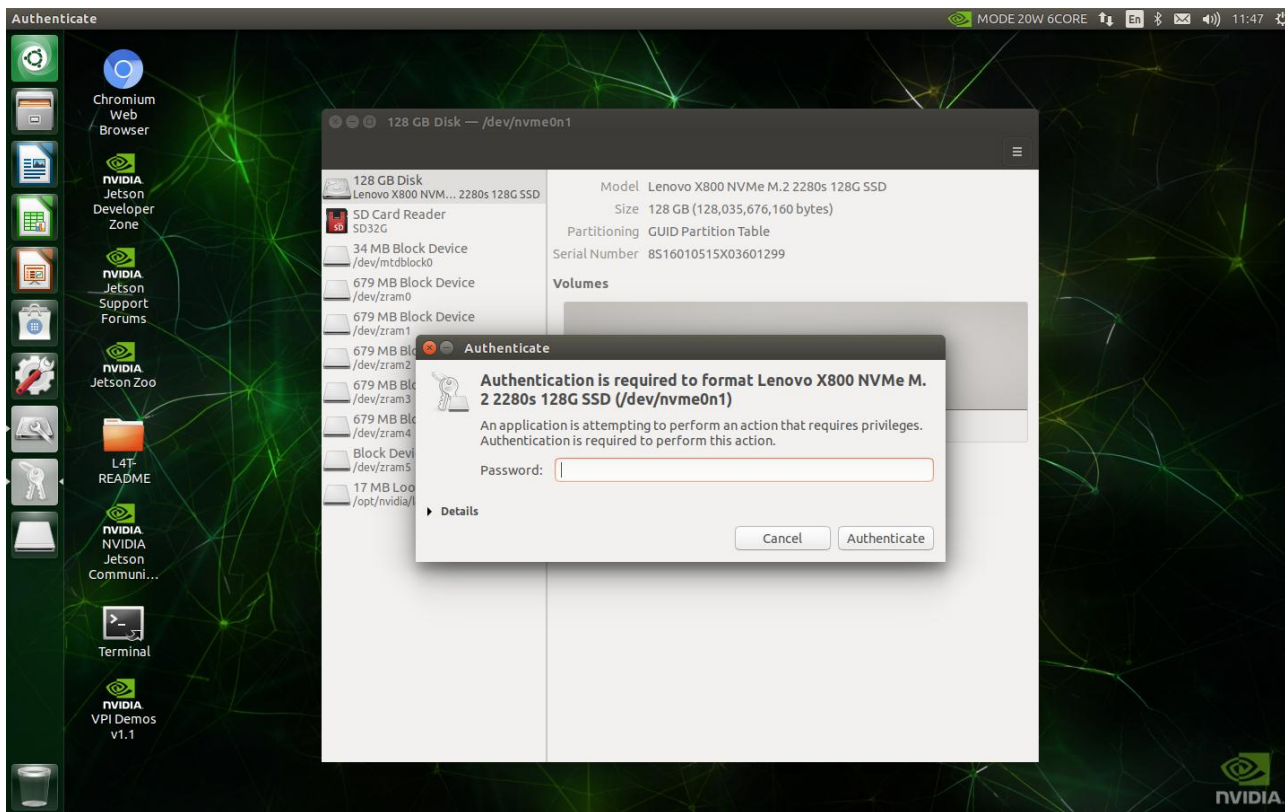
2) 进入 disks, 找到安装的 ssd, 然后组合按键” ctrl+F” , 对硬盘进行快速格式化, 点击 Formaat;



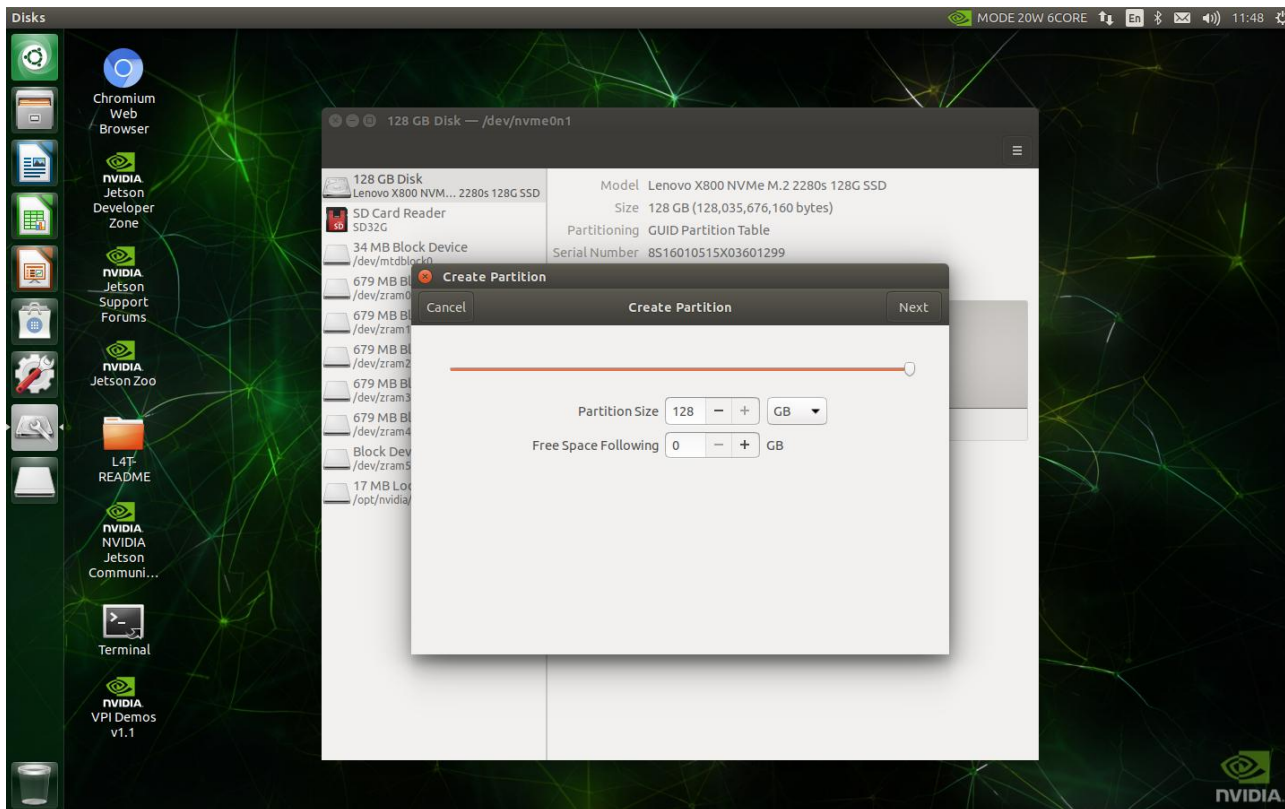
3) 继续点击 format



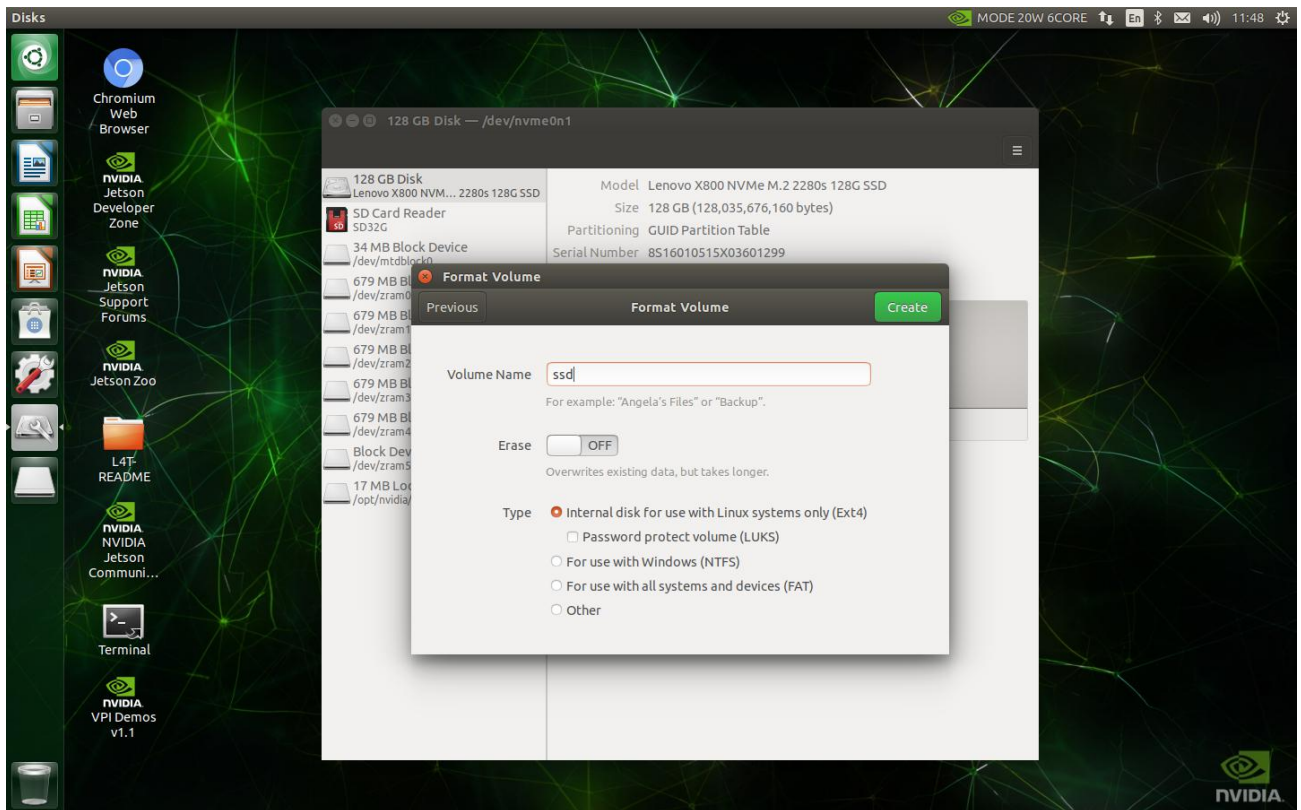
4) 输入密码, 若未自己修改密码, 则输入默认密码” nvidia” (注意大小写)



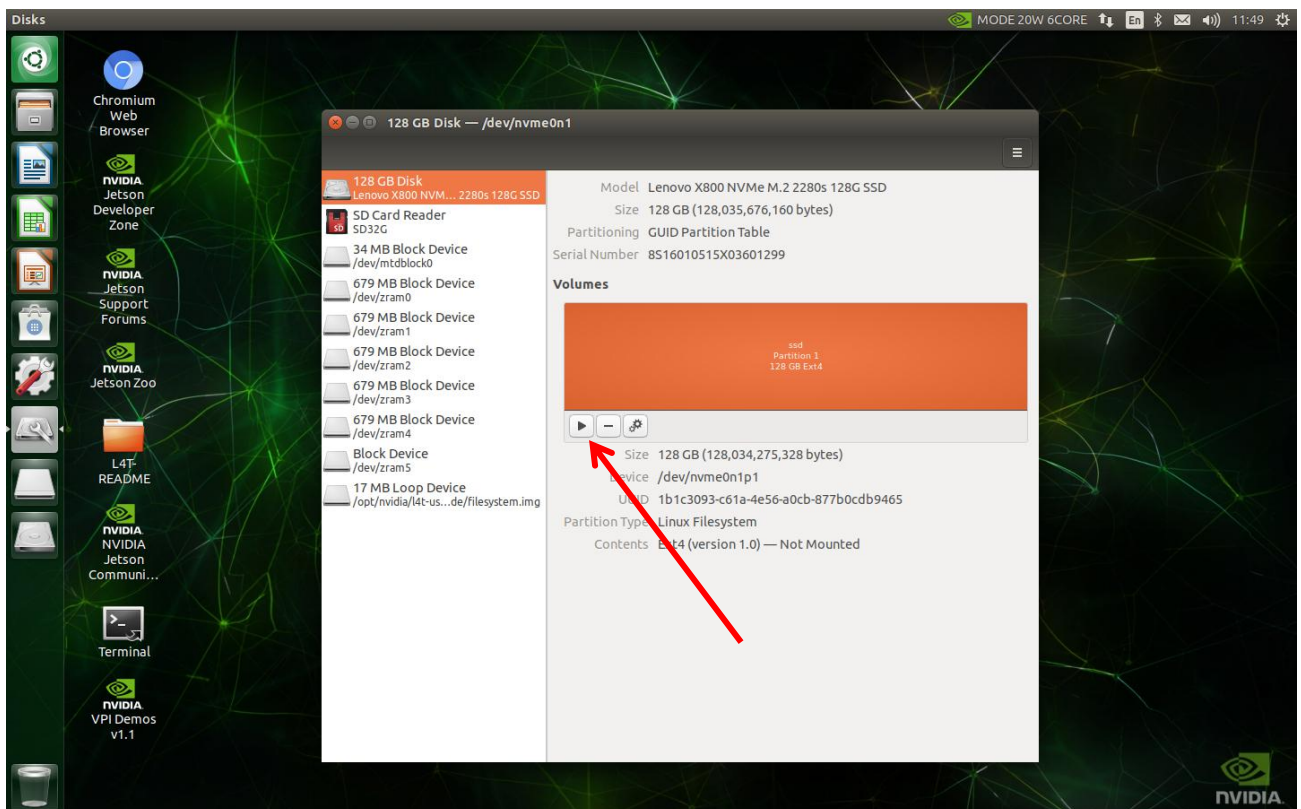
5) 可以自己划分大小占比, 下图是默认最大分区, 点击 next



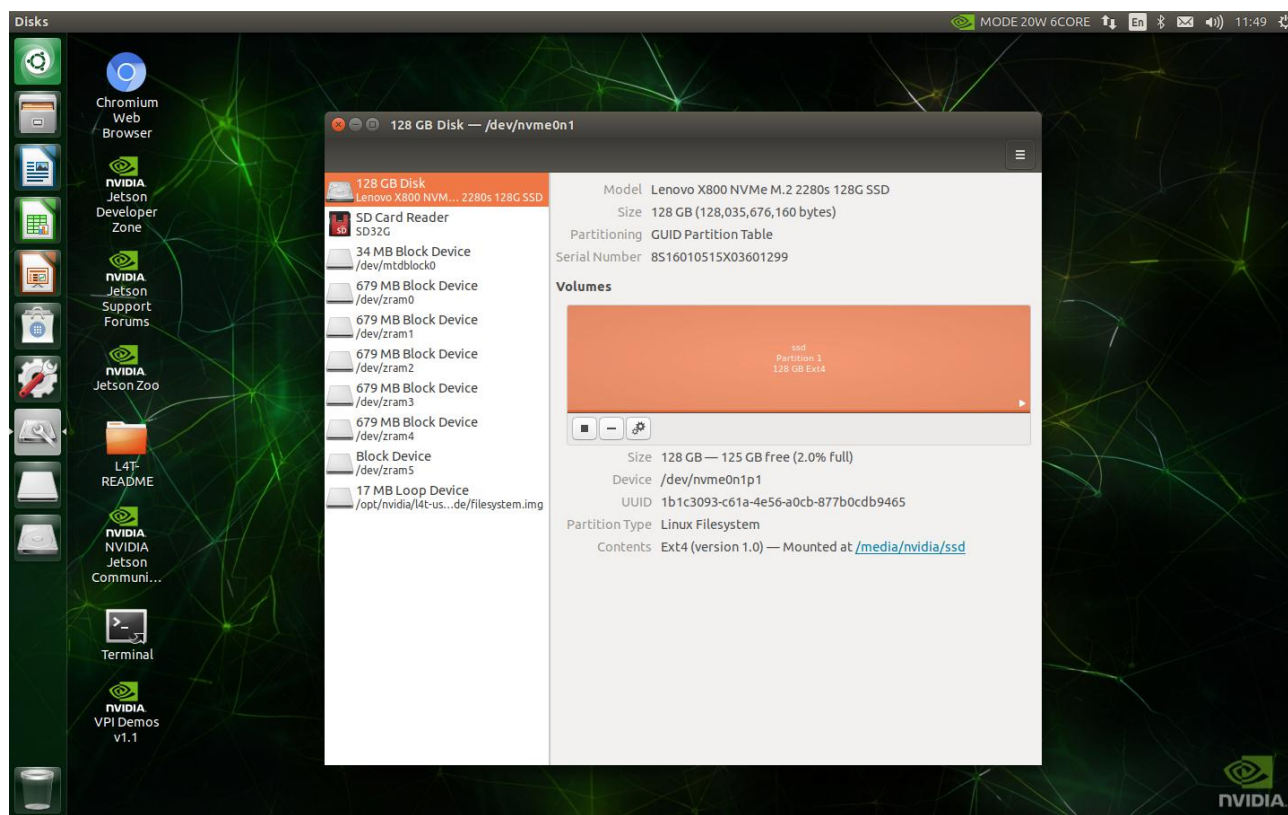
6) 给分区起名字, 其他均为默认选项无需修改, 然后点击 create;



7) 出现下图所示画面, 点击三角符号挂载



8) 如下图显示则为挂载成功



步骤二: (系统盘转换)

`git clone https://github.com/jetsonhacks/rootOnNVMe.git` (如若 `git` 出现 `fail`, 请根据网站链接另行下载压缩包, 然后 `copy` 到设备上解压)

`cd rootOnNVMe`

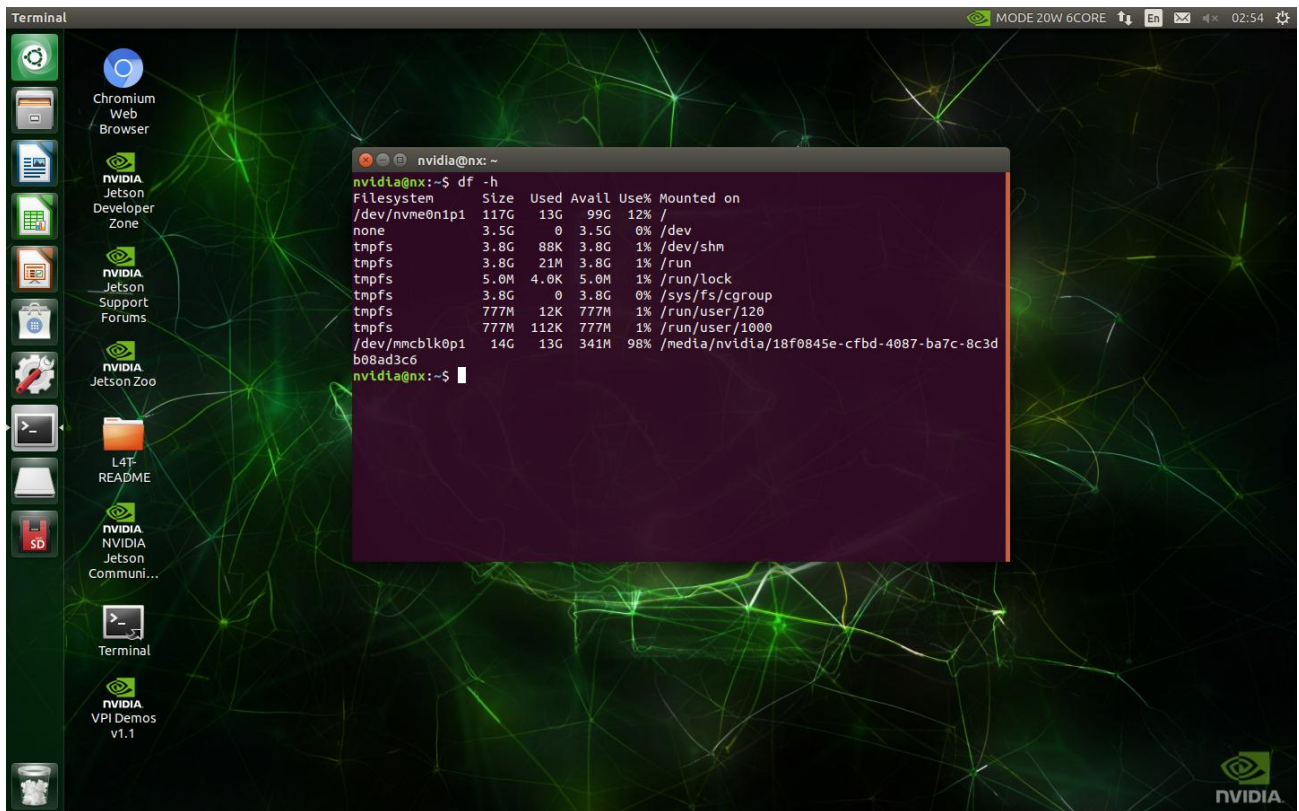
`./copy-rootfs-ssd.sh` (这个时间比较长, 请耐心等待执行完毕再执行下一步)

`./setup-service.sh`

`sudo reboot`

步骤三: 检验

打开终端输入 `df -h`, 查看硬盘设备 `/dev/nvme0n1p1` 是否挂载在 `/` 根目录上, 如若不是, 请从头开始执行



```
Terminal
MODE 20W 6CORE
En 02:54

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

系统安装

我司 T506S 产品系公司自行研发主板搭配 NVIDIA XAVIER NX 核心板, 配套软件驱动也 由我司人员开发设计. 若在使用过程中, 由于需要重新刷机或更改其他配置导致 usb 以及其他接口无法正常使用, 需要下载我司提供的驱动包, 安装我方驱动, 方可使外置接口正常使用.

下载前准备

Ubuntu 18.04 系统的电脑 1 台

Micro USB 数据线 1 条

设置下载模式

NVIDIA Jetson 是通过 Micro USB 接口升级系统, 更新前需让 T506S 进入 Recovery 模式. Recovery 模式下可以进行文件系统更新包含: 内核 Kernel, 启动 Bootloader, 文件系统 Rootfs 等.

T506S 进入 Recovery 模式的步骤:

- 1) 连接 T506S 系统电源;
- 2) 使用 USB TYPE-C 的数据线连接 Jetson 和 Ubuntu Host 主机(一端插在 T506S 的 OTG 口, 一端插在 Ubuntu Host 主机的 usb 3.0 插口);
- 3) 使用 T506S 配置的电源为系统进行上电开机;
- 4) 先按下 RECOVERY 同时再按住 RESET 复位键后;
- 5) 保持按下 RECOVERY 按键 2 秒后释放 RESET 复位按键, 最后释放 RECOVERY 按键, 此时 T506S 进入 Recovery 刷机模式 (可通过在 Ubuntu Host 主机上运行命令: `lsusb` 查看是否有 Nvidia Corporation 设备 (不同 Jetson 模块 USB VID/PID 不同) 来确认是否进入正常)

```
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0
Bus 003 Device 005: ID 0955:7c18 NVidia Corp.
```

注意: 在进入 USB Recovery 模式下, 系统不会启动, 串口不会有调试信息输出

软件升级步骤

第一步、将百度网盘镜像包下载到 Ubuntu 电脑(注意:不要下载到中文目录下, 后续操作也不要再在中文目录下操作, 会造成刷机失败), 并使用该电脑作为主机, 后续命令运行都在该电脑上进行

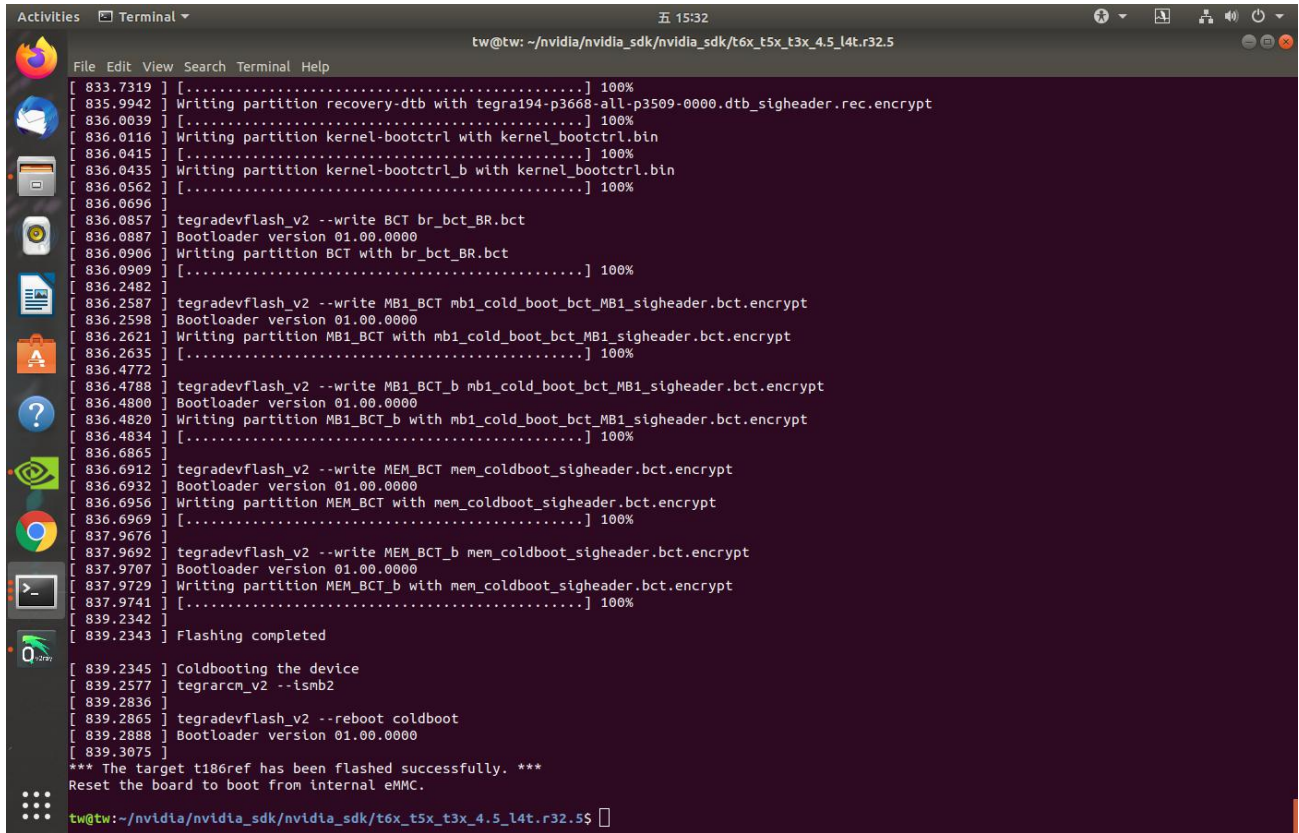
链接: <https://pan.baidu.com/s/191TmusenDQrYq4CIUSWNyw>

提取码: 4mrd



第二步、按照镜像包内说明进行相关操作

第二步、等待刷机完成



```
tw@tw: ~/nvidia/nvidia_sdk/nvidia_sdk/t6x_t5x_t3x_4.5_l4t.r32.5
[ 833.7319 ] [.....] 100%
[ 835.9942 ] Writing partition recovery-dtb with tegra194-p3668-all-p3509-0000.dtb_sigheader.rec.encrypt
[ 836.0039 ] [.....] 100%
[ 836.0116 ] Writing partition kernel-bootctrl with kernel_bootctrl.bin
[ 836.0415 ] [.....] 100%
[ 836.0435 ] Writing partition kernel-bootctrl_b with kernel_bootctrl.bin
[ 836.0562 ] [.....] 100%
[ 836.0696 ]
[ 836.0857 ] tegradevflash_v2 --write BCT br_bct_BR.bct
[ 836.0887 ] Bootloader version 01.00.0000
[ 836.0906 ] Writing partition BCT with br_bct_BR.bct
[ 836.0909 ] [.....] 100%
[ 836.2482 ]
[ 836.2587 ] tegradevflash_v2 --write MB1_BCT mb1_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 836.2598 ] Bootloader version 01.00.0000
[ 836.2621 ] Writing partition MB1_BCT with mb1_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 836.2635 ] [.....] 100%
[ 836.4772 ]
[ 836.4788 ] tegradevflash_v2 --write MB1_BCT_b mb1_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 836.4800 ] Bootloader version 01.00.0000
[ 836.4820 ] Writing partition MB1_BCT_b with mb1_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 836.4834 ] [.....] 100%
[ 836.6865 ]
[ 836.6912 ] tegradevflash_v2 --write MEM_BCT mem_coldboot_sigheader.bct.encrypt
[ 836.6932 ] Bootloader version 01.00.0000
[ 836.6956 ] Writing partition MEM_BCT with mem_coldboot_sigheader.bct.encrypt
[ 836.6969 ] [.....] 100%
[ 837.9676 ]
[ 837.9692 ] tegradevflash_v2 --write MEM_BCT_b mem_coldboot_sigheader.bct.encrypt
[ 837.9707 ] Bootloader version 01.00.0000
[ 837.9729 ] Writing partition MEM_BCT_b with mem_coldboot_sigheader.bct.encrypt
[ 837.9741 ] [.....] 100%
[ 839.2342 ]
[ 839.2343 ] Flashing completed
[ 839.2345 ] Coldbooting the device
[ 839.2577 ] tegrarcv2 --lsmb2
[ 839.2836 ]
[ 839.2865 ] tegradevflash_v2 --reboot coldboot
[ 839.2888 ] Bootloader version 01.00.0000
[ 839.3075 ]
*** The target t186ref has been flashed successfully. ***
Reset the board to boot from internal eMMC.
tw@tw:~/nvidia/nvidia_sdk/nvidia_sdk/t6x_t5x_t3x_4.5_l4t.r32.5$
```

Jtop 安装

Jtop (一个系统监视实用程序, 可在终端上运行, 并实时查看和控制 NVIDIA Jetson 的状态), 安装也非常方便, 如果 Jetson 产品上已安装了 Jetpack SDK, 可以按照如下步骤安装运行:

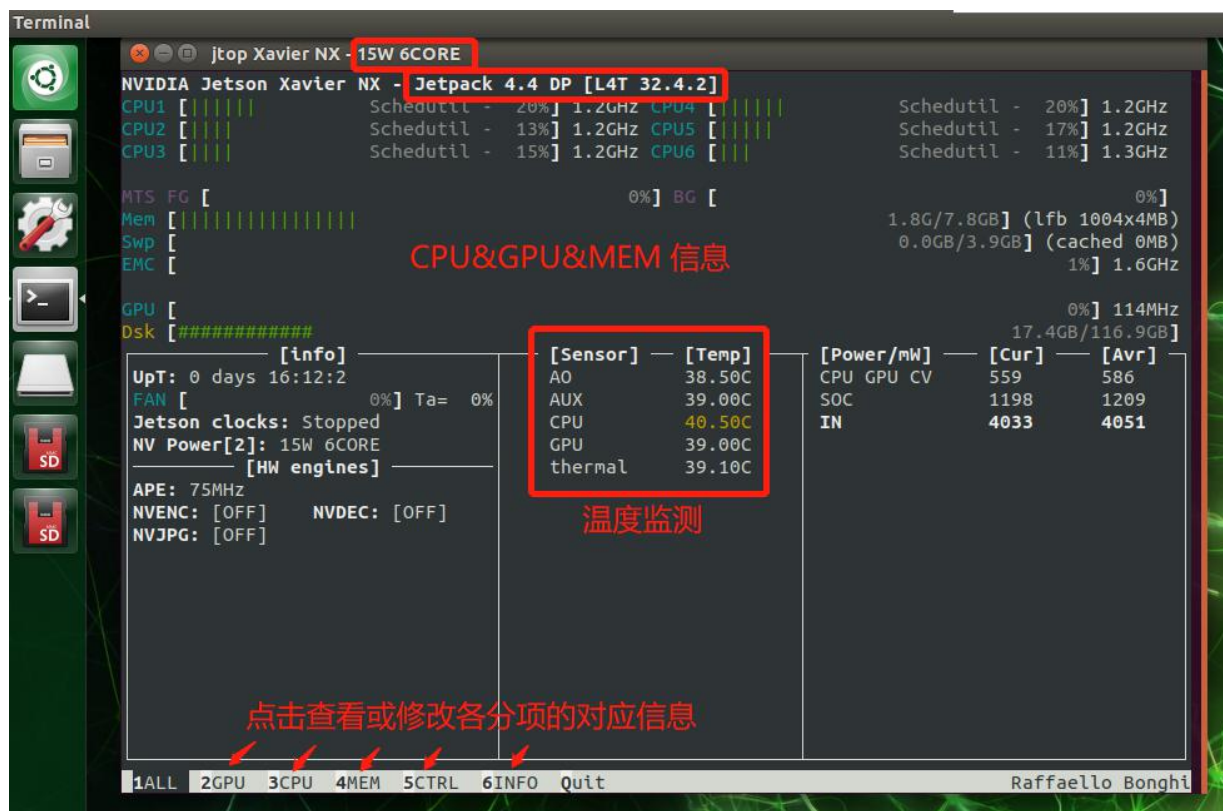
安装及运行

```
sudo apt-get install python3-pip
```

```
sudo pip3 install jetson-stats (包含 jtop)
```

```
sudo jtop
```

开启后的界面如下:



非常方便地看到当前 Jetson 机器上的各种完整信息, 一般在首页就可以读取到很丰富的数据信息:

ALL

包含模块运行信息包括: CPU、内存、GPU、磁盘、风扇、jetson_clock 状态、NVModel 等等

GPU

实时 GPU 状态

CPU

实时 CPU 状态

CTRL

可以控制的状态

INFO

Lib 库、CUDA、Serial Number、Interface 等信息

可以使用以下键盘命令, 控制 NVIDIA Jetson 的相关配置:

在第 3 页 MEM 中：

c：清除缓存

s：启用/禁用额外交换

+/-：增加和减少交换大小

在第 4 页中 CTRL：

启动 / 停止 jetson_clocks 服务（注：仅 jetson_clocks 从时间 60 秒后开始）

e：在启动时启用/禁用 jetson_clocks

+/-：增加和减少 NVP 模型

f：风扇的手动/ jetson_clocks 模式

p/m：增加和降低风扇速度

1、jetson_release 命令显示 NVIDIA Jetson 的状态和所有信息

```
nvdi@nx:~$ jetson_release
```

```
- NVIDIA Jetson Xavier NX
```

```
* Jetpack 4.4 DP [L4T 32.4.2]
```

```
* NV Power Mode: MODE_15W_6CORE - Type: 2
```

```
* jetson_clocks service: inactive
```

```
- Libraries:
```

```
* CUDA: 10.2.89
```

```
* cuDNN: 8.0.0.145
```

```
* TensorRT: 7.1.0.16
```

```
* Visionworks: 1.6.0.501
```

```
* OpenCV: 4.3.0 compiled CUDA: YES
```

```
* VPI: 0.2.0
```

```
* Vulkan: 1.2.70
```

2、tegrastats 命令行查看各资源信息

```
$ tegrastatsRAM 1778/7763MB (1fb 986x4MB) SWAP 0/3882MB (cached 0MB) CPU
```

```
[20%@1190,5%@1190,5%@1190,5%@1190,4%@1190,4%@1190] EMC_FREQ 0% GR3D_FREQ 0% AO@37.5C GPU@37.5C PMIC@100C AUX@37.5C
```

```
CPU@38.5C thermal@38C VDD_IN 3913/3877 VDD_CPU_GPU_CV 440/429 VDD_SOC 1198/1198
```

RAM：内存占用 CPU：CPU 各核心的占用率 EMC：external memory controller，外存控制器，单位 bus%MHzGR3D：GPU 占用了%Thermal：各模块温度数据

VNC Viewer 安装(远程图形界面工具)

在 Jetson 上搭建一个 VNC 服务器，从而允许其他设备通过网络访问 Jetson 的 Linux 图形界面进行远程工作，并避免了连接 HDMI 显示器、USB 键盘或鼠标的需要。

如下时搭建 VNC 服务器的过程，搭建完成后即可通过 ViewVNC 进行远程访问：

步骤一、安装 vino

```
sudo apt update
sudo apt install vino
```

步骤二、Enable VNC 服务（此时手动可打开 vnc server）

每次登陆时使能 VNC Server:

```
sudo ln -s ../vino-server.service /usr/lib/systemd/user/graphical-session.target.wants
```

配置 VNC server:

```
gsettings set org.gnome.Vino prompt-enabled false
gsettings set org.gnome.Vino require-encryption false
```

额外添加并手动启动 VNC Service: -1) Edit the org.gnome.Vino schema to restore the missing "enabled" parameter

```
sudo vi /usr/share/glib-2.0/schemas/org.gnome.Vino.gschema.xml Add this key: (最后)
```

```
<key name='enabled' type='b'>
  <summary>Enable remote access to the desktop</summary>
  <description>
    If true, allows remote access to the desktop via the RFB
    protocol. Users on remote machines may then connect to the
    desktop using a VNC viewer.
  </description>
  <default>>false</default>
</key>
```

-2) Compile the schemas for Gnome:

```
sudo glib-compile-schemas /usr/share/glib-2.0/schemas
```

-3) Now the screen sharing panel in unity-control-center works... but this is not enough to get vino running! So you need to add in the programs at session start: Vino-server with the following command line:

```
/usr/lib/vino/vino-server
```

步骤三、设置 VNC 登陆密码('12345678' 修改为自己的密码)

```
gsettings set org.gnome.Vino authentication-methods "[vnc]"
gsettings set org.gnome.Vino vnc-password $(echo -n '12345678' |base64)
```

步骤四、重启机器，验证是否设置 vnc 成功

```
sudo reboot
```

步骤五、设置开机自启动 VNC Server

The VNC server is only available after you have logged in to Jetson locally. If you wish VNC to be available automatically, use the system settings application to enable automatic login.

```
gsettings set org.gnome.Vino enabled true
```

```
mkdir -p ~/.config/autostart
```

```
vi ~/.config/autostart/vino-server.desktop
```

```
[Desktop Entry]
Type=Application
Name=Vino VNC server
Exec=/usr/lib/vino/vino-server
NoDisplay=true
```

连接 VNC Server

Use any standard VNC client application to connect to the VNC server that is running on Linux for Tegra. Popular examples for Linux are gview and remmina. Use your own favorite client for Windows or MacOS.

To connect, you will need to know the IP address of the Linux for Tegra system. Execute the following command to determine the IP address:

```
ifconfig
```

Search the output for the text "inet addr:" followed by a sequence of four numbers, for the relevant network interface (e.g. eth0 for wired Ethernet, wlan0 for WiFi, or l4tbr0 for the USB device mode Ethernet connection).

设置显示桌面分辨率 Setting the Desktop Resolution

如果没有显示器连接，默认 VNC 连接后的分辨率为 640x480，通过修改添加如下内容，将其默认 VNC 分辨率设置

```
sudo vi /etc/X11/xorg.conf
```

Section "Screen"

```
Identifier "Default Screen"
```

```
Monitor "Configured Monitor"
```

```
Device "Tegra0"
```

SubSection "Display"

```
Depth 24
```

```
Virtual 1920 1080 # Modify the resolution by editing these values
```

```
EndSubSection
```

```
EndSection
```


常用框架安装

pytorch 介绍

PyTorch 是一个开源的 Python 机器学习库，基于 Torch，用于自然语言处理等应用程序。

2017 年 1 月，由 Facebook 人工智能研究院（FAIR）基于 Torch 推出了 PyTorch。它是一个基于 Python 的可续计算包，提供两个高级功能：

- 具有强大的 GPU 加速的张量计算（如 NumPy）；

- 包含自动求导系统的深度神经网络。

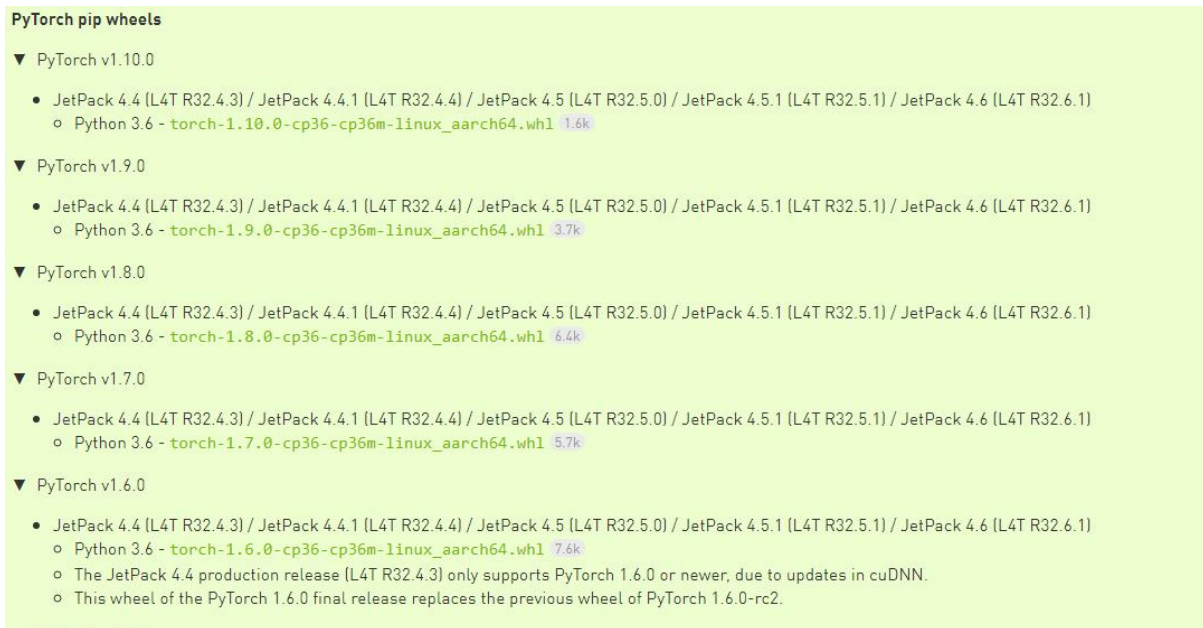
发展:PyTorch 的前身是 Torch，其底层和 Torch 框架一样，但是使用 Python 重新写了很多内容，不仅更加灵活，支持动态图，而且提供了 Python 接口。它是由 Torch7 团队开发，是一个以 Python 优先的深度学习框架，不仅能够实现强大的 GPU 加速，同时还支持动态神经网络。PyTorch 既可以看作加入了 GPU 支持的 numpy，同时也可以看成一个拥有自动求导功能的强大的深度神经网络。除了 Facebook 外，它已经被 Twitter、CMU 和 Salesforce 等机构采用。

优点:PyTorch 是相当简洁且高效快速的框架,设计追求最少的封装,设计符合人类思维,它让用户尽可能地专注于实现自己的想法,与 google 的 Tensorflow 类似,FAIR 的支持足以确保 PyTorch 获得持续的开发更新,PyTorch 作者亲自维护的论坛 供用户交流和求教问题,入门简单.



安装

Pytorch 安装跟系统版本紧密联系, 安装 pythoch 不同版本之前, 先确定设备的 jetpack 版本号, 根据下面图片来确定需要安装什么版本的 pythrch 才能正常使用.



基于 python3.6 安装方式(以下方式为 1.8.0 为例):

```
wget https://nvidia.box.com/shared/static/p57jwntv436lfrd78inwl7iml6p13fzh.whl -O
torch-1.8.0-cp36-cp36m-linux_aarch64.whl
sudo apt-get install python3-pip libopenblas-base libopenmpi-dev
pip3 install Cython
pip3 install numpy torch-1.8.0-cp36-cp36m-linux_aarch64.whl
```

基于 python2.7 安装方式(以下方式为 1.4.0 为例):

```
wget https://nvidia.box.com/shared/static/1v2cc4ro6zvsbu0p8h6qcuacqolqcsif.whl -O
torch-1.4.0-cp27-cp27mu-linux_aarch64.whl
sudo apt-get install libopenblas-base libopenmpi-dev
pip install future torch-1.4.0-cp27-cp27mu-linux_aarch64.whl
```

注意:PyTorch v1.4.0 for L4T R32.4.2 是最后一个支持 Python 2.7 的版本, 所已建议在 python3.6 下安装.

torchvision 安装

```
sudo apt-get install libjpeg-dev zlibg-dev libpython3-dev libavcodec-dev libavformat-dev libswscale-dev
git clone --branch v0.x.0 https://github.com/pytorch/vision torchvision
cd torchvision
export BUILD_VERSION=0.x.0 # (填写 torchvision 安装的版本号)
python3 setup.py install
```

验证:

```
python3
>>> import torch
>>> print(torch.__version__)
>>> print('CUDA available: ' + str(torch.cuda.is_available()))
>>> print('cuDNN version: ' + str(torch.backends.cudnn.version()))
>>> a = torch.cuda.FloatTensor(2).zero_()
>>> print('Tensor a = ' + str(a))
>>> b = torch.randn(2).cuda()
>>> print('Tensor b = ' + str(b))
>>> c = a + b>>> print('Tensor c = ' + str(c))
>>> import torchvision
>>> print(torchvision.__version__)
```

Pytorch whl 包下载地址:

链接: <https://pan.baidu.com/s/1J2DeI9Hmt0MkH-ZgGfsFnA>

提取码: wlf1

TensorFlow 介绍

TensorFlow™是一个基于数据流编程 (dataflow programming) 的符号数学系统, 被广泛应用于各类机器学习 (machine learning) 算法的编程实现, 其前身是谷歌的神经网络算法库 DistBelief。

Tensorflow 拥有多层次结构, 可部署于各类服务器、PC 终端和网页并支持 GPU 和 TPU 高性能数值计算, 被广泛应用于谷歌内部的产品开发和各领域的科学研究。

TensorFlow 由谷歌人工智能团队谷歌大脑 (Google Brain) 开发和维护, 拥有包括 TensorFlow Hub、TensorFlow Lite、TensorFlow Research Cloud 在内的多个项目以及各类应用程序接口 (Application Programming Interface, API)。自 2015 年 11 月 9 日起, TensorFlow 依据阿帕奇授权协议 (Apache 2.0 open source license) 开放源代码。

TensorFlow 是世界上最受欢迎的开源机器学习框架, 它具有快速、灵活并适合产品级大规模应用等特点, 让每个开发者和研究者都能方便地使用人工智能来解决多样化的挑战。



TensorFlow

安装

以下是 XavierNX 的官方 TensorFlow 版本信息。

Python 3.6+JetPack4.5

```
sudo apt-get install libhdf5-serial-dev hdf5-tools libhdf5-dev zlib1g-dev zip libjpeg8-dev liblapack-dev libblas-dev gfortran
```

```
sudo apt-get install python3-pip
```

```
sudo pip3 install -U pip testresources setuptools==49.6.0
```

```
sudo pip3 install -U numpy==1.16.1 future==0.18.2 mock==3.0.5 h5py==2.10.0 keras_preprocessing==1.1.1
```

```
keras_applications==1.0.8 gast==0.2.2 futures protobuf pybind11
```

```
# TF-2.x
```

```
sudo pip3 install --pre --extra-index-url https://developer.download.nvidia.com/compute/redis/jp/v45 tensorflow# #
```

```
# TF-1.15
```

```
sudo pip3 install --pre --extra-index-url https://developer.download.nvidia.com/compute/redis/jp/v45
```

```
‘tensorflow<2’
```

Python 3.6+JetPack4.4

```
sudo apt-get update
```

```
sudo apt-get install libhdf5-serial-dev hdf5-tools libhdf5-dev zlib1g-dev zip libjpeg8-dev liblapack-dev libblas-dev gfortran
```

```
sudo apt-get install python3-pip
```

```
sudo pip3 install -U pip testresources setuptools
```

```
sudo pip3 install -U numpy==1.16.1 future==0.17.1 mock==3.0.5 h5py==2.9.0 keras_preprocessing==1.0.5
```

```
keras_applications==1.0.8 gast==0.2.2 futures protobuf pybind11
```

```
# TF-2.x
```

```
sudo pip3 install --pre --extra-index-url https://developer.download.nvidia.com/compute/redis/jp/v44
```

```
tensorflow==2.3.1+nv20.12
```

```
# TF-1.15
```

```
sudo pip3 install --pre --extra-index-url https://developer.download.nvidia.com/compute/redis/jp/v44
```

```
‘tensorflow<2’
```

TensorFlow 安装包下载地址:

链接: <https://pan.baidu.com/s/1gLC-1Z5pC40ASezrpfY2Rw>

提取码: 6iyv

保修条例

重要提示

保证提供的每个嵌入式产品，就其所知在材料与工艺上均无任何缺陷，完全符合原厂正式发货之规格。

保修范围包括全部原厂产品，由经销商配置的配件出现故障时请与经销商协商解决。提供的所有产品的保修期限均为一年（超出保修期限的提供终身维修服务），保修期限的起始时间自出厂之日起开始计算，对于保修期内维修好的产品，维修部分延长质保12个月。除非另行通知，否则您的原厂发货单日期即为出厂日期。

如何获得保修服务

如果您在保修期内产品不能正常运行，请与经销商联系以获得保修服务，产品保修时请出示购货发票证明（这是您获得保修服务的权利证明）。

保修解决措施

当您要求保修服务时，您需要遵循规定的问题确定和解决程序。您需要接受技术人员通过电话或以电子邮件方式与您进行首次诊断，届时需要您配合详细填写我们所提供的报修单上所有问题，以确保我们准确判断故障原因及造成损毁位置（过保产品我们还会提供收费单，需要您确认）。对所报修产品进行“维修”或“更换”，如果产品被“更换”或“维修”，被更换的“故障”产品或修理后更换后的“故障”零件将被返回。因部分维修产品需发往原厂，为避免意外损失，请您购买运输保险，如果用户放弃保险，那么所寄物品在运输途中损坏或遗失，不承担责任。对于保修期限内的产品，用户承担维修产品返回厂家时的运费，承担维修后的产品返还用户的运费。

以下情况不在保修之列

- 1、 产品的不适当安装、使用不当、误用、滥用（如超出工作负荷等）
- 2、 不当的维护保管（如火灾、爆炸等）或自然灾害（如雷电、地震、台风等）所致产品故障或损坏。
- 3、 对产品的改动（如电路特性、机械特性、软件特性、三防处理等）。
- 4、 其它显然是由于使用不当造成的故障（如电压过高、电压过低、浮地电压过高、极性接反、针脚弯曲或折断、接错总线、器件脱落、静电击穿、外力挤压、坠落受损、温度过高、湿度过大、运输不良等）。
- 5、 产品上的标志和部件号曾被删改或去除。
- 6、 产品超过保修期。

特别说明：

如多个产品出现同一故障或多次在同一设备出现相同故障或损坏时，为查找原因以确认责任。我司有权要求使用者提供周边设备实物或技术资料，例如：监视器，I/O 设备，电缆，电源，连接示意图，系统结构图等。否则，我们有权拒绝履行保修，维修时将按照市场价格收取费用，并收取维修保证金。