



图为科技  
TWOWIN TECHNOLOGY

让世界更智慧，让生活更美好！

[www.twowininit.com](http://www.twowininit.com)

# 图为 TW-T906G 用户手册

**TWOWIN TW-T906G USER MANUAL**



让世界更智慧 让生活更美好

图为信息科技(深圳)有限公司

TWOWIN TECHNOLOGY CO. LTD

图为信息科技（深圳）有限公司

## 目录

安全警示及使用注意事项 .....	1
简介 .....	2
产品规格 .....	3
处理器及核心模块 .....	3
接口 .....	4
编解码 .....	5
供电 .....	5
结构 .....	5
环境 .....	6
尺寸 .....	7
服务与支持 .....	8
技术支持 .....	8
保修 .....	8
接口说明 .....	9
正面接口 .....	9
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
串口说明 .....	13
背面接口 .....	15
多功能接口 .....	16
接口定义 .....	16
CAN 功能测试 .....	17
GPIO 测试 .....	19
GPS 使用 .....	20
GMSL 摄像头使用 .....	25
WiFi 连接 .....	27
4G 拨号联网 .....	30
采用 SSD 为系统盘 .....	32
系统安装 .....	43
PaddlePaddle FastDeploy .....	45
Jtop 安装 .....	49
Vnc viewer 安装(远程图形界面工具) .....	53

## 文档修订目录

### 文档版本

Version1.0

文档版本号	修订日期	修订内容
V1.0	2020/07/23	初始发布
V2.0	2022/01/03	内容增加, 错误去除



让世界更智慧，让生活更美好！

[www.twowininit.com](http://www.twowininit.com)

## 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

图为信息科技(深圳)有限公司版权所有，并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图为信息科技(深圳)有限公司。未经图为信息科技(深圳)有限公司书面同意，任何人不得以任何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。图为信息科技(深圳)有限公司在编写该文档的时候已尽最大努力保证其内容准确可靠，但图为信息科技(深圳)有限公司不对本文档中的遗漏、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 [www:Https://twowin.com](https://twowin.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：

联系人：图为技术支持

手机：15920093612

电话：0755-82840481

网址：[www.twowininit.com](http://www.twowininit.com)

地址：深圳市南山区科技南十二路长虹科技大厦 2512

## 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-906G 边缘计算平台输入端电源稳定可靠，典型功率 45W 最大功率 75W；

电源范围： 9 – 36 v

### ● 环境要求：

工作温度： -20°C – 55°C

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对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-906G 为一款基于 NVIDIA® JETSON AGX ORIN™系列模块面向无人驾驶车载系统的计算平台，内置集成 AGX ORIN 模块，预装 Ubuntu 20.04 操作系统，具备 200 TOPS 浮点运算的 AI 处理能力，采用超强固轻型铝合金材料设计，传导被动散热，具备优秀的散热能力，尺寸轻巧外观新颖，支持 USB、CAN、RS232、GPIO、同步信号等丰富 IO 接口类型，内置 4G 通信模块和 WIFI 模块，支持同步信号输入输出。预留便于坚固安装支架，具有计算能力强、可靠性高、集成度高、功耗低特点，可应用于无人清洁车、无人配送车、智能巡检、AGV 等无人驾驶领域。

### TW-906G 边缘计算平台概述

- 内嵌 NVIDIA® JETSON AGX ORIN™
- 支持 M. 2 KEY M (PCIE X4 NVME 2280)
- 支持 M. 2 KEY E (PCIE X1 2230)
- 支持 8 路 GMSL2 摄像头
- 支持多种接口(如 CAN/USB/以太网/同步信号/串口/GPIO 等)
- 支持双频 WIFI/蓝牙/4G/5G 模组
- 日本 JAE 车规级 IO 插头 (1xPOWER, 3xCAN, 2xGPIO)
- 风扇和被动散热设计
- 内置 ubuntu 20.04 系统和 JETPACK SDKS



## 产品规格

### 处理器及核心模块

Processor	NVIDIA AGX ORIN
CPU	8-core NVIDIA Arm® Cortex A78AE v8.2 64-bit CPU 2MB L2 + 4MB L3
CPU Max Frequency	2.2 GHz
GPU	1792-core NVIDIA Ampere GPU with 56 Tensor Cores
GPU Max Frequency	939 MHz
Memory	32 GB 256-bit LPDDR5
DL accelerator	2 x NVDLA x 2.0
DLA Max Frequency	1.4 GHz
Storage	64GB eMMC 5.1 1 x m.2 key m nvme 2280(选配 安装)

## 接口

	Interface	Quantity	Note
<b>Network</b>	4G	1	有方 N720/移远 EC20/广和通 NL668(选配安装)
	5G	1	广和通 FM150-AE/芯讯通 SIM7600CE(选配安装)
	ETHERNET	2×RJ45 GIGABIT NETWORK PORT	ALTERNATIVE RJ45 AND WATERPROOF PORT(1000BASE-T)
	WIFI	1	2.4G/5.8G 300MBPS
<b>Video output</b>	HDMI	1×HDMI 2.0 TYPE A	不支持转接信号(例如 vga 转 hdmi 形式)
<b>I/O</b>	USB	4×USB 3.0TYPE A	USB 5V, 1A
	TYPE-C	1xOTG	USB2.0
	GPIO	2xGPIO(独立输入输出)	3.3VTTL 电平
	CAN	2xCAN	WITH CAN CHIP
	UART	1x RS-232 D-SUB9	TYPE-C USB FORM UART DEBUG (BUILT IN USB-TO-UART CONVERTER)
	TIME SYNC	1x 输入输入同步信号 D-SUB9	RS232 DB9 TERMINAL
	M.2	1×M.2 M KEY 1X M.2 KEY E (2230)	1xPCIE NVME 2280 SSD(选配安装) 1xUSB 2230 WIFI(选配安装)
	Power key	1	Button
<b>Function key</b>	Reset key	1	Button
	Recovery key	1	Button

## 编解码

	2x 4K60 (H.265) 4x 4K30 (H.265) 8x 1080p60 (H.265) 16x 1080p30 (H.265)
<b>Video Encode</b>	1x 8K30 (H.265) 3x 4K60 (H.265) 7x 4K30 (H.265) 11x 1080p60 (H.265) 22x 1080p30 (H.265)
<b>Video Decode</b>	

## 供电

Power supply	Spec
Input type	日本 JAE 连接器 MX23A12SF1
Input voltage	Wide input 9-36v
Typical consumption	典型功率 45W 最大 75W

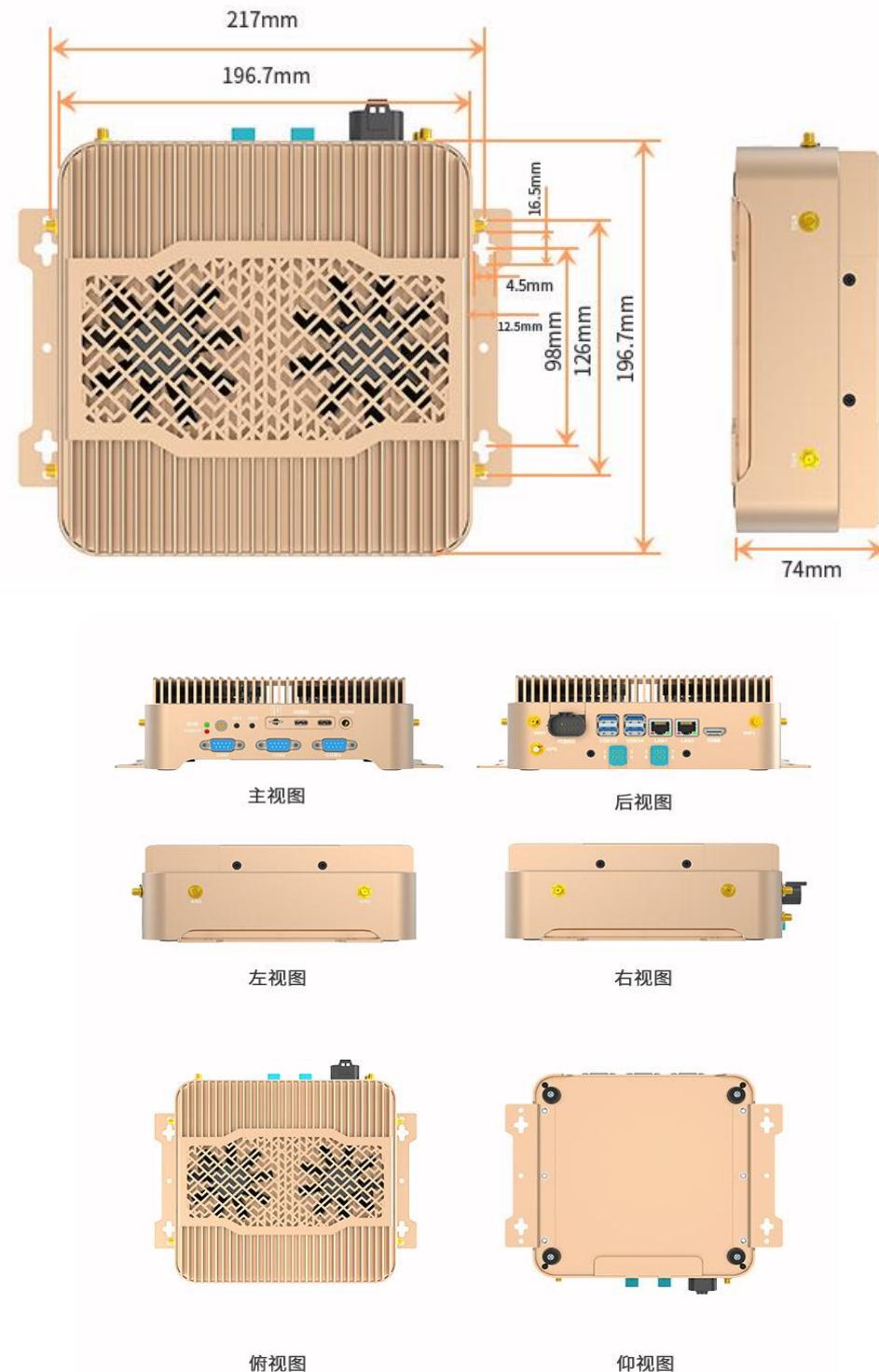
## 结构

Mechanical	Spec
Dimensions (w×h×d)	196.7(w)mm×196.7(d)mm ×74.0(h)mm
Weight	2.5kg

## 环境

Environmental	Spec
Operating temperature	-20°C-55°C
Storage humidity	10%-90% non-condensing

## 尺寸



## 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: 图为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图为将进行免费保修。请通过购买平台客服对话以及电话联系获取保修协助。(详情请参考图为信息科技(深圳)有限公司保修条例)。

## 接口说明

### 正面接口



TW-906G 正面接口示意图

接口	接口名称	接口说明
POWER	电源按键	长按关机,按一下屏幕显示四个相关选项
RES	恢复按键	单独使用为复位重启,搭配 REC 进入 RECOVER 模式
REC	复位按键	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
DEBUG	信息调试串口	可通过连接 TYPE-C 线输出调试信息
OTG	TYPE-C 接口	可通过连接 TYPE-C 线设备连接进行数据交换
AUDIO	音频接口	外接声音播放设备
POWER_LED	电源状态指示灯	接通电源后：指示灯为绿色常亮
RUN_LED	系统状态指示灯	接通电源后：指示灯为红色常亮
COM1	RS232	标准 RS232 串口 ( <b>对应设备号为 ttyTHS0</b> )
COM2	RS232	标准 RS232 串口 ( <b>对应设备号为 ttyTHS1, 如果内置 GPS 则该接口被占用</b> )
TIME SYNC	RS232	标准 RS232 串口 ( <b>对应设备号为 ttyTHS4</b> )



让世界更智慧, 让生活更美好!

[www.twowinit.com](http://www.twowinit.com)

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 按键,不松手再按住 RES 按键,2 秒后松开 RES 按键,最后松开 REC 按键,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,按键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.

## 功能介绍

### 系统介绍

图为 T906G 设备采用 Ubuntu20.04 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

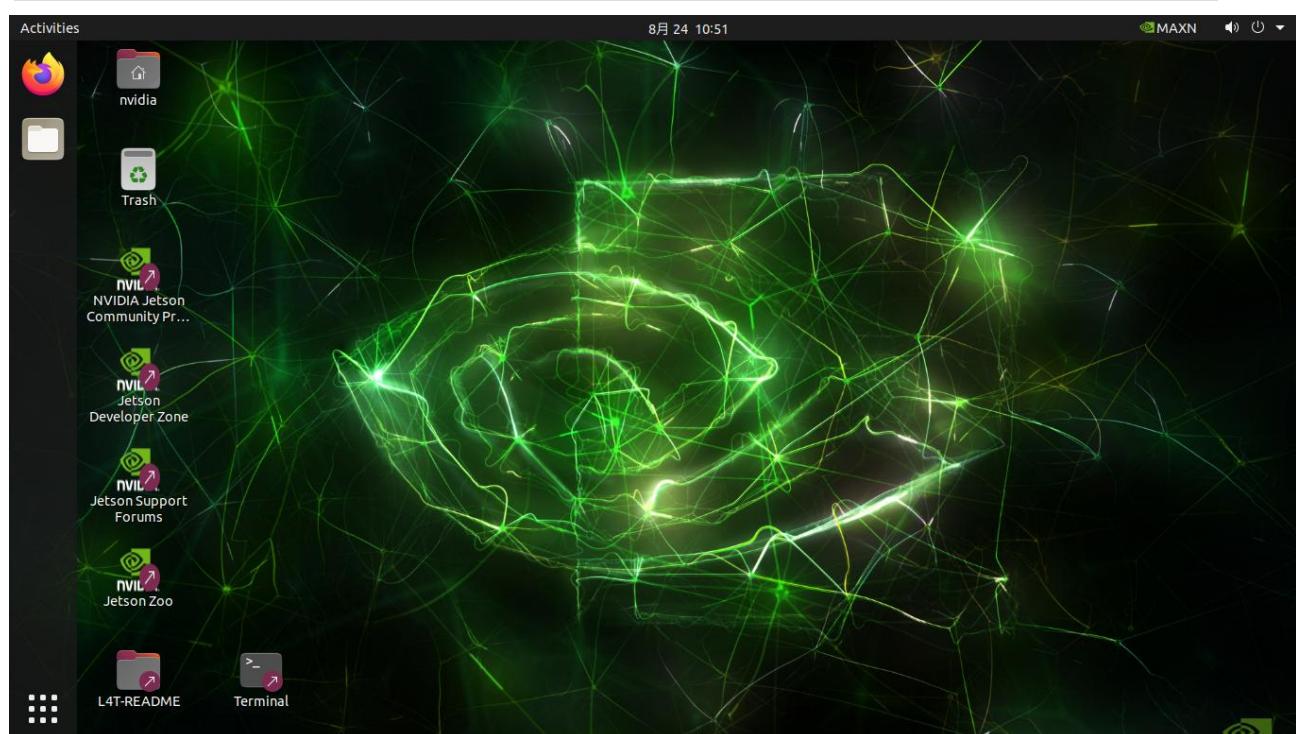
输入密码:nvidia

### 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

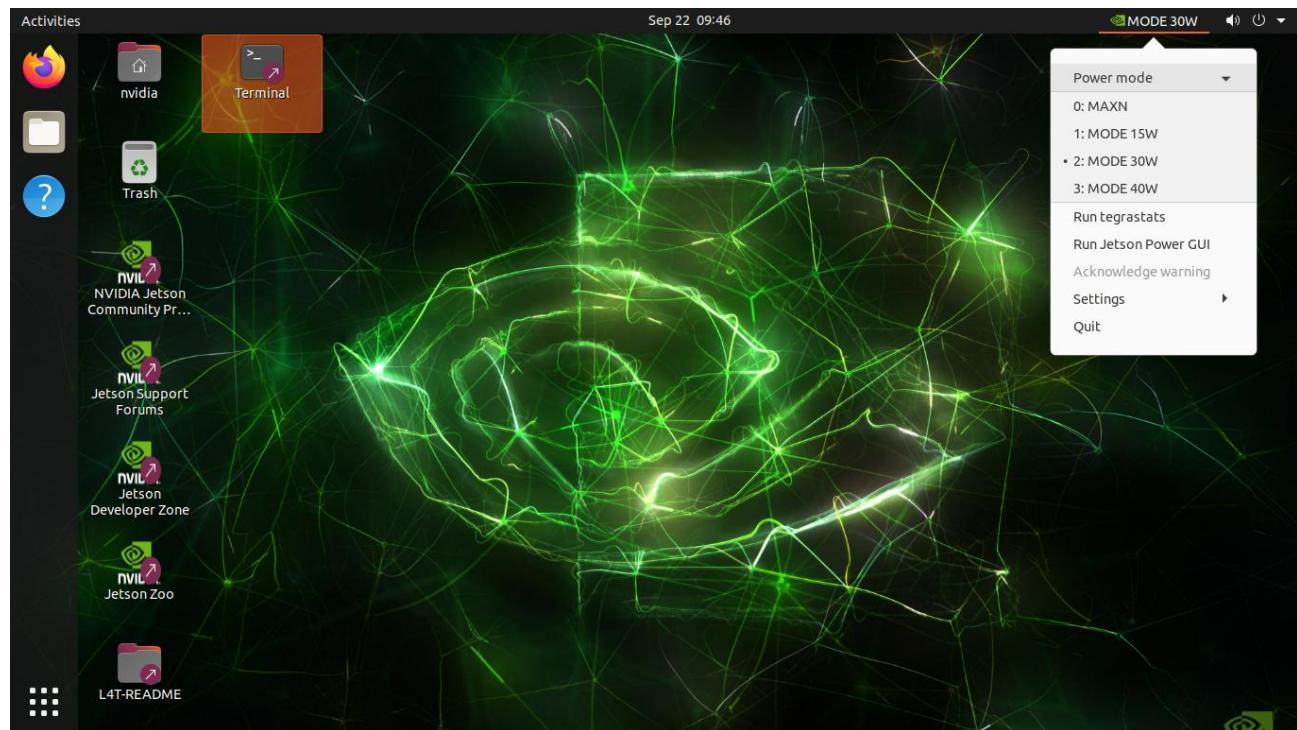
开机: TW-T906G 设备默认开机模式为上电自启动。插入电源, 并将显示器通过 HDMI 接口与设备相连, 开机画面如图所示:



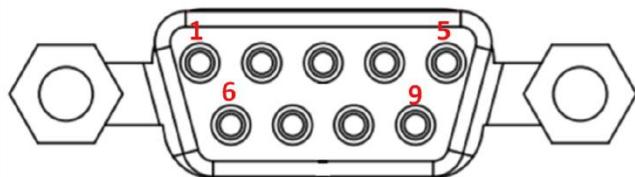
开机画面

## 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;



## 串口说明



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		

串口测试方法:

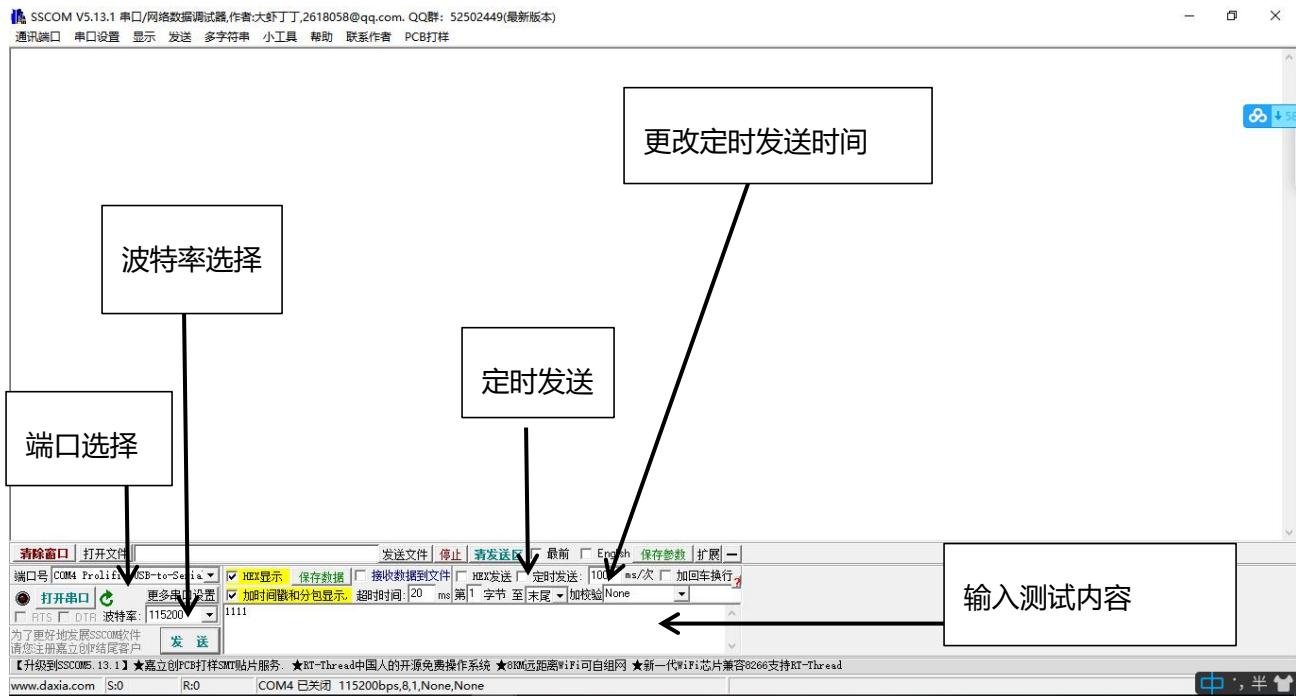
我司使用如图所示 usb 转串口线对 TW-906G 设备进行相应测试, 串口均为标准串口, 测试需自行购买标准 usb 转串口线, 我司出厂产品不附赠此线,



USB-RS232

RS232 测试:

步骤一: 使用 usb-RS232 转接线母头连接 TW-906G 设备 COM1 或者 COM2, 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

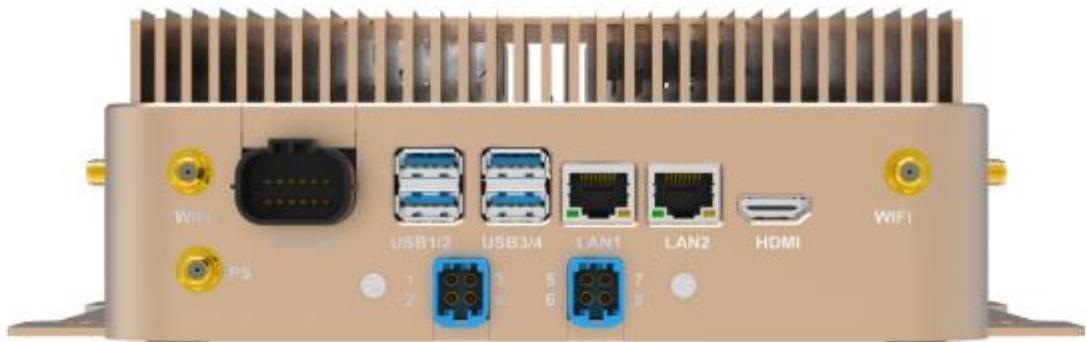
步骤二: TW-906G 设备终端输入 sudo cutecom , 打开 cutecom(默认设置即可), 串口号选择 ttyTHS1(COM1), 打开串口, 查看对话框是否有收到正确信息(除中文乱码外, 数字出现乱码或传输数据不正确则表明不正常, 传输数据一致则表明该串口通信正常)

步骤三: 参考步骤二, 测试 COM2. COM3.

测试串口工具下载链接: 链接: [https://pan.baidu.com/s/1rdwcrgh1\\_28\\_x6l20yjqxw](https://pan.baidu.com/s/1rdwcrgh1_28_x6l20yjqxw)

提取码: pf3d

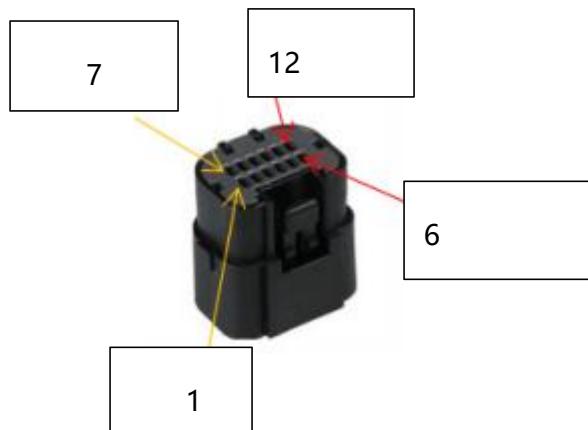
## 背面接口



TW-906G 背面接口示意图

接口	接口名称	接口说明
WIFI	WIFI 天线接口	外接 WIFI 天线
GPS	GPS 天线接口	外接 GPS 天线
多功能接口	多功能 IO 接口	包含 CAN0/CAN1,4xGPIO 以及供电输入
USB×4	USB 3.0 接口	USB3.1 接口，向下兼容 USB2.0 接口 支持超速、高速及低速模式
1 2 3 4 5 6 7 8	GMSL 摄像头接口	4 合 1GMSL2 摄像头接口
LAN1	以太网万兆口	万兆网口
LAN2	以太网千兆口	千兆网口，向下兼容百兆网口
HDMI	HDMI 接口	标准 HDMI2.0 接口

## 多功能接口



1	2	3	4	5	6
SPI-CAN-H0	CAN-H0	CAN-H1	GND	GPIO1	VCC-IN
7	8	9	10	11	12
SPI-CAN-L0	CAN-L0	CAN-L1	GND	GPIO2	VCC-IN

## 接口定义

PIN 定义			
PIN#	定义	PIN#	定义
1	SPI-CAN-H0	7	SPI-CAN-L0
2	CAN_H0	8	CAN_L0
3	CAN-H1	9	CAN_L1
4	GND	10	GND
5	GPIO1 (GPIO-350 PA.02 0-1.8V) 默认为输出, 低电平	11	GPIO2 (GPIO-390 PG.07 0-3.3V) 默认为输出, 低电平
6	VDDIN(12-24V)	12	VDDIN(12-24V)

## CAN 功能测试

### 1、 can\_test 目录

can\_test 目录下包含如下文件：

canSendTest : CAN 通信发送测试可执行文件

canRecvTest : CAN 通信接收测试可执行文件

canUp. sh : CAN 通信驱动加载脚本

ko 文件: SPI-CAN 驱动模块

### 2、 驱动加载, 设置并设能 CAN 接口

在当前目录下运行命令

```
sudo ./canUp. sh
```

在运行结束后, 打印 “set can0 up success!!!” , 表示 can0 接口已设置并启用。

如下图, 运行后, 有 can0 、 can1 、 can2 可以使用。

注意 bitrate 波特率。

```
nvidia@tegra-ubuntu:~/nfs$ sudo ./canUp.sh
[sudo] password for nvidia:
-----print arguments
bitrate = 500000
dbitrate = 2000000
spiCan = mcp251xfd
-----print end

-----set can down
set can0 down !!!
set can1 down !!!
set can2 down !!!

-----rmmodding CAN driver ...
rmmod: mcp251xfd
rmmod: mtcan
rmmod: can_raw
rmmod: can
-----rmmod CAN driver end

-----Loading CAN driver ...
modprobe: can
modprobe: can_raw
modprobe: mtcan
insmod: mcp251xfd
-----Loaded CAN driver end

-----set can up
set can0 up success!!!
set can1 up success!!!
set can2 up success!!!
bitrate=500000
dbitrate=2000000
spiCan=mcp251xfd
nvidia@tegra-ubuntu:~/nfs$
```

### 3、 can 接收测试（以 can2 测试为例）

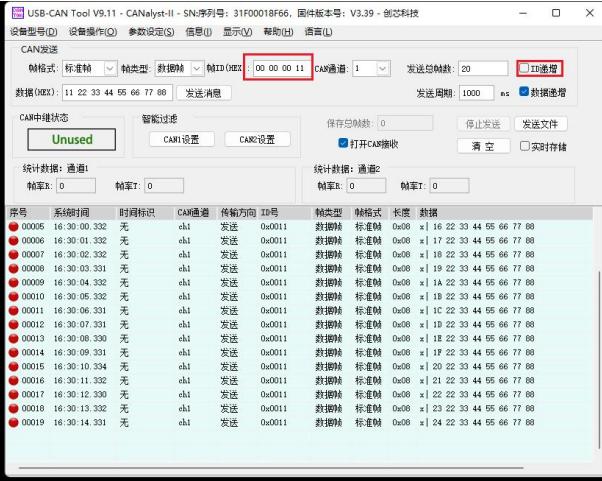
运行命令 `./canRecvTest can2`。此测试 demo 默认接收帧 ID 为 0x11。

运行命令后等待 can2 接收数据。

在 USB-CAN 工具设置帧 ID 为 “00 00 00 11”，取消勾选 ID 递增。设置波特率为上一步获取的 bitrate。点击“发送消息”按钮，发送数据到 CAN 总线。

在终端窗口即可接收到从 USB-CAN 发送的数据。

```
nvidia@tegra-ubuntu:~/nfs$ ./canRecvTest can2
start can2 send data test
recv count = 1 ID=0x11 DLC=8 data = 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 2 ID=0x11 DLC=8 data = 0x12 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 3 ID=0x11 DLC=8 data = 0x13 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 4 ID=0x11 DLC=8 data = 0x14 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 5 ID=0x11 DLC=8 data = 0x15 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 6 ID=0x11 DLC=8 data = 0x16 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 7 ID=0x11 DLC=8 data = 0x17 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 8 ID=0x11 DLC=8 data = 0x18 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 9 ID=0x11 DLC=8 data = 0x19 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 10 ID=0x11 DLC=8 data = 0x1A 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 11 ID=0x11 DLC=8 data = 0x1B 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 12 ID=0x11 DLC=8 data = 0x1C 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 13 ID=0x11 DLC=8 data = 0x1D 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 14 ID=0x11 DLC=8 data = 0x1E 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 15 ID=0x11 DLC=8 data = 0x1F 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 16 ID=0x11 DLC=8 data = 0x20 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 17 ID=0x11 DLC=8 data = 0x21 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 18 ID=0x11 DLC=8 data = 0x22 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 19 ID=0x11 DLC=8 data = 0x23 0x22 0x33 0x44 0x55 0x66 0x77 0x88
recv count = 20 ID=0x11 DLC=8 data = 0x24 0x22 0x33 0x44 0x55 0x66 0x77 0x88
```

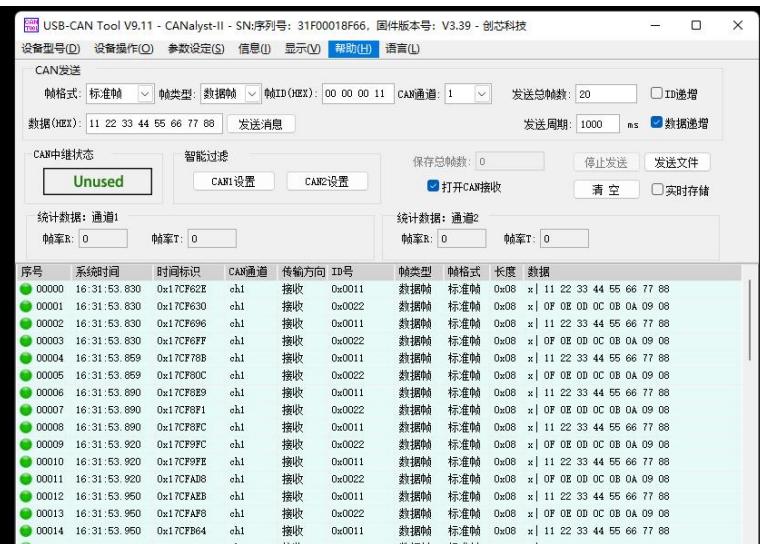


### 4、 can 发送测试（以 can2 测试为例）

运行命令 `./canSendTest can2`。此测试 demo 默认交替发送帧 ID 围殴 0x11、0x22 的数据。

运行命令后 USB-CAN 接收到 can2 发送的数据。

```
nvidia@tegra-ubuntu:~/nfs$ ./canSendTest can2
start can2 send data test
send count = 1
send count = 2
send count = 3
send count = 4
send count = 5
send count = 6
send count = 7
send count = 8
send count = 9
send count = 10
send count = 11
send count = 12
send count = 13
send count = 14
send count = 15
send count = 16
send count = 17
send count = 18
send count = 19
send count = 20
send count = 21
send count = 22
send count = 23
send count = 24
send count = 25
send count = 26
send count = 27
send count = 28
send count = 29
send count = 30
send count = 31
send count = 32
send count = 33
send count = 34
send count = 35
```



## GPIO 测试(D1 和 D2 均可设置为输入或者输出)

将 GPIO-D1 设置为输出:

```
sudo chmod 777 /sys/class/gpio/export
sudo echo 350 > /sys/class/gpio/export
sudo chmod 777 /sys/class/gpio/PA.02/value
sudo chmod 777 /sys/class/gpio/PA.02/direction
sudo echo "out" > /sys/class/gpio/PA.02/direction
sudo echo 1 > /sys/class/gpio/PA.02/value
sudo echo 0 > /sys/class/gpio/PA.02/value
```

将 GPIO-D2 设置为输入:

```
sudo chmod 777 /sys/class/gpio/export
sudo echo 390 > /sys/class/gpio/export
sudo chmod 777 /sys/class/gpio/PG.07/value
sudo chmod 777 /sys/class/gpio/PG.07/direction
sudo echo "in" > /sys/class/gpio/PG.07/direction
```

## GPS 使用

GPS 模块为选配模块, 非标准品自带, 我司采用 GPS 模块为 HX-26-GN 系列模块, 是 12X16 尺寸的高性能 BDS/GNSS 全星座定位航模块系列的总称。该系列模块产品都是基于中科微第四代低功耗 GNSS SOC 单芯片—AT6558, 支持多种卫星导航系统, 包括中国的 BDS (北斗卫星导航系统), 美国的 GPS, 俄罗斯的 GLONASS, 欧盟的 GALILEO, 日本的 QZSS 以及卫星增强系统 SBAS (WAAS, EGNOS, GAGAN, MSAS)。AT6558 是一款真正意义的六合一多模卫星导航定位芯片, 包含 32 个跟踪通道, 可以同时接收六个卫星导航系统的 GNSS 信号, 并且实现联合定位、导航与授时, HX-26-GN 系列模块具有高灵敏度、低功耗、低成本等优势, 适用于车载导航、手持定位、可穿戴设备, 可以直接替换 Ublox NEO 系列模块。

### 性能指标

出色的定位导航功能, 支持 BDS/GPS/GLONASS 卫星导航系统的单系统定位, 以及任意组合的多系统联合定位, 并支持 QZSS 和 SBAS 系统;

支持 A-GNSS;

冷启动捕获灵敏度: -148dBm;

跟踪灵敏度: -162dBm;

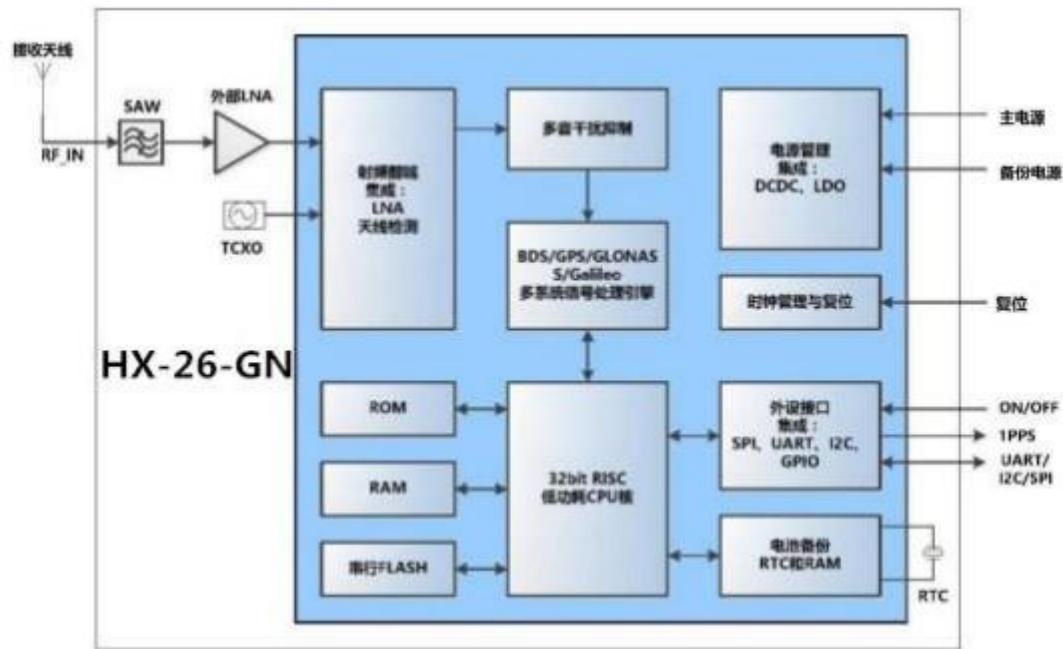
定位精度: 2.5 米 (CEP50);

首次定位时间: 32 秒;

低功耗: 连续运行<25mA (@3.3V);

内置天线检测及天线短路保护功能 .

功能框图:



## 技术规范

指标	技术参数
信号接收	BDS/GPS/GLONASS/GALILEO/QZSS/SBAS
射频通道数目	三通道射频，支持全星座BDS、GPS和GLONASS同时接收
冷启动TTFF	≤35s
热启动TTFF	≤1s
重捕获TTFF	≤1s
冷启动捕获灵敏度	-148dBm
热启动捕获灵敏度	-156dBm
重捕获灵敏度	-160dBm
跟踪灵敏度	-162dBm
定位精度	<2m (1σ)
测速精度	<0.1m/s (1σ)
授时精度	<30ns (1σ)
定位更新率	1Hz (默认)，最大10Hz
串口特性	波特率范围：4800 bps ~115200 bps, 默认 9600bps, 8个数据位, 无校验, 1个停止位
协议	NMEA0183
最大高度	18000m
最大速度	515m/s
最大加速度	4g
后备电池	1.5V ~ 3.6V
电源供电	2.7V ~ 3.6V
GPS&BD典型功耗	<25mA @3.3V
工作温度	-40到+85摄氏度
存储温度	-45到+125摄氏度
尺寸	16.0mm×12.2mm×2.4mm
重量	1.6g

## 使用方法

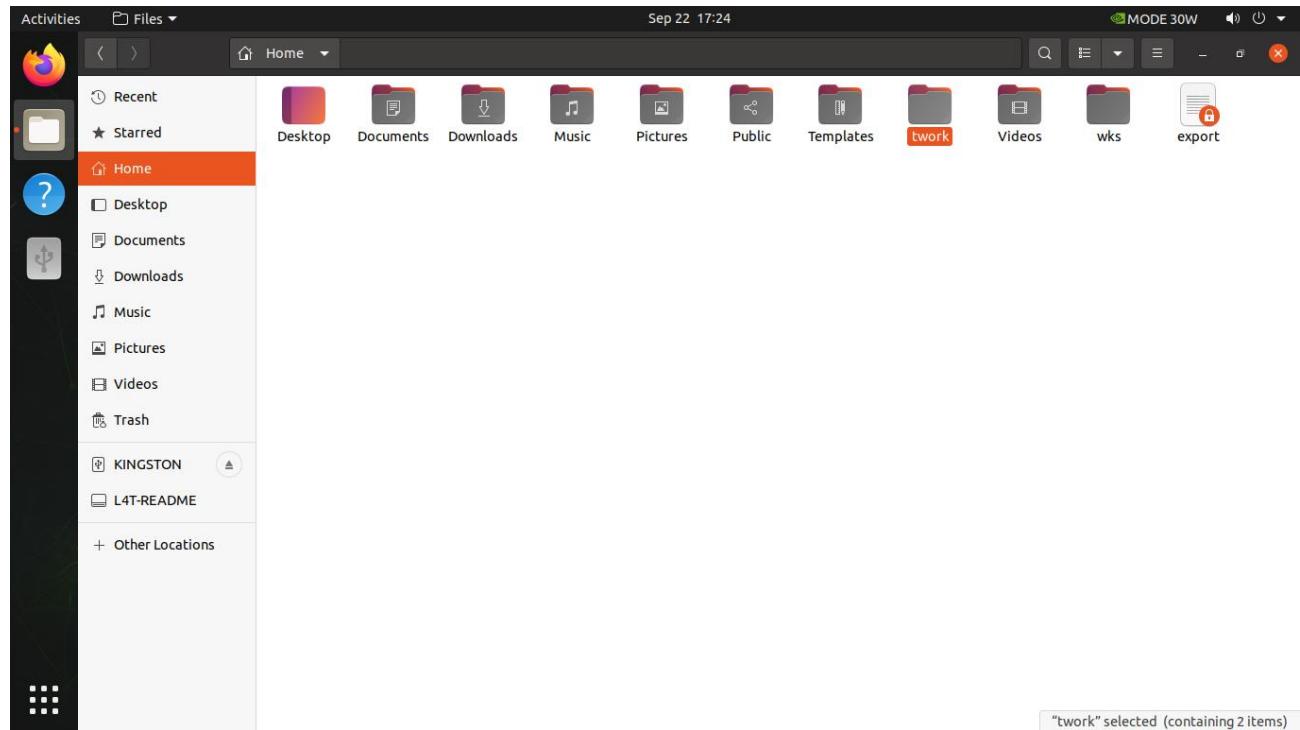
在使用 GPS 之前, 需要将 GPS 天线(图一)接在外围天线孔(图二), 确认连接无误后, 打开系统内自带 twork 文件, 执行预留程序, 读取 GPS 信息即可. 具体操作步骤如下图所示;



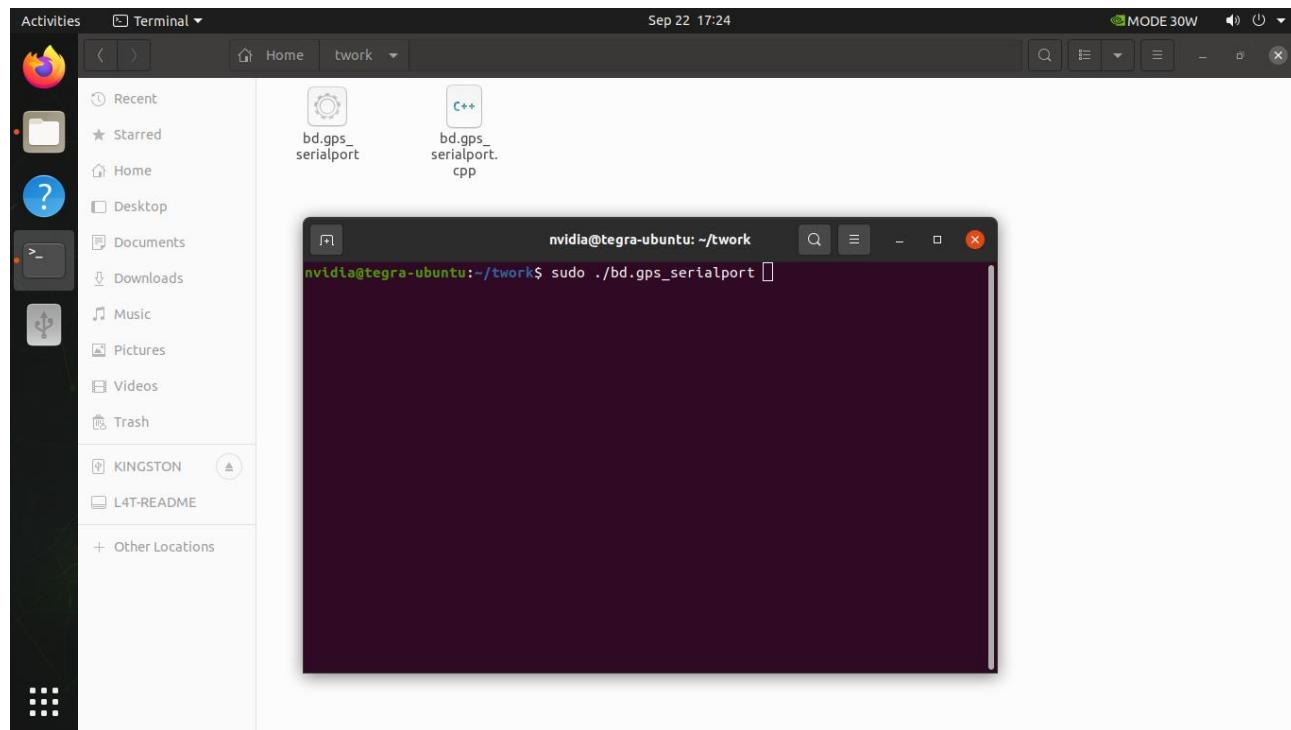
图一

图二

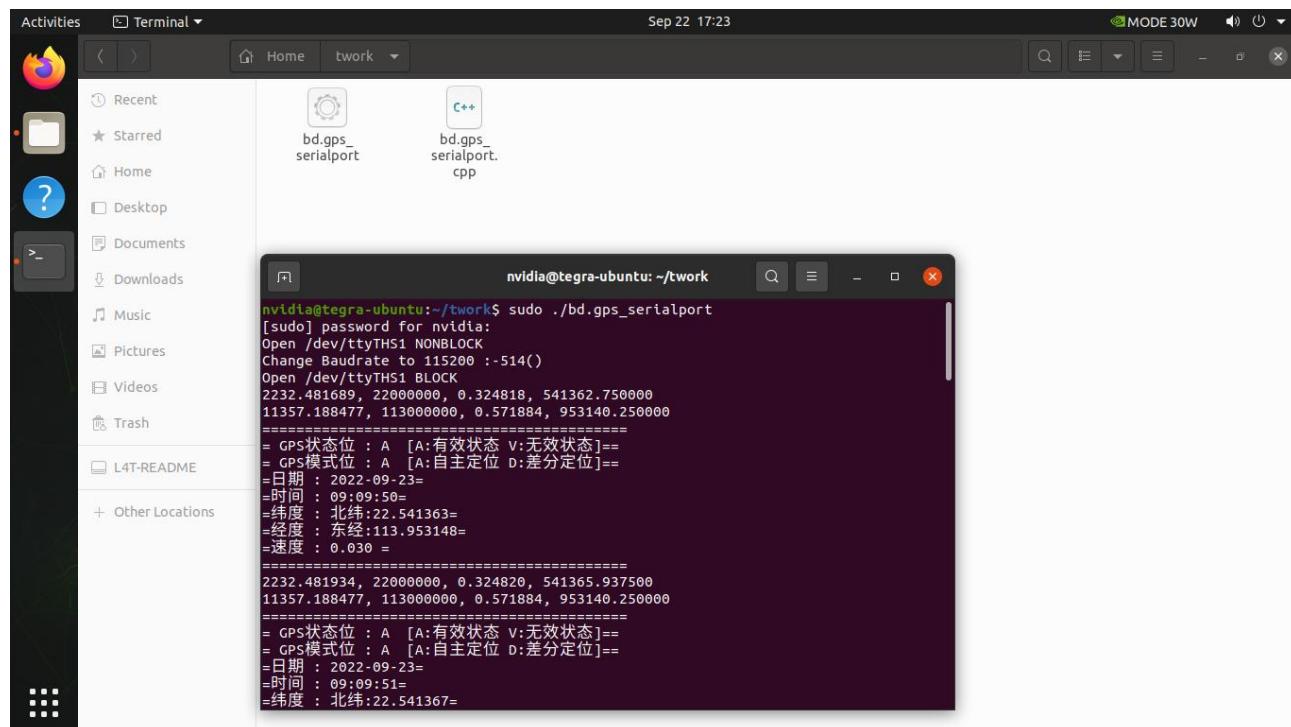
找到 home 下自带的 twork 文件, 进入该文件夹



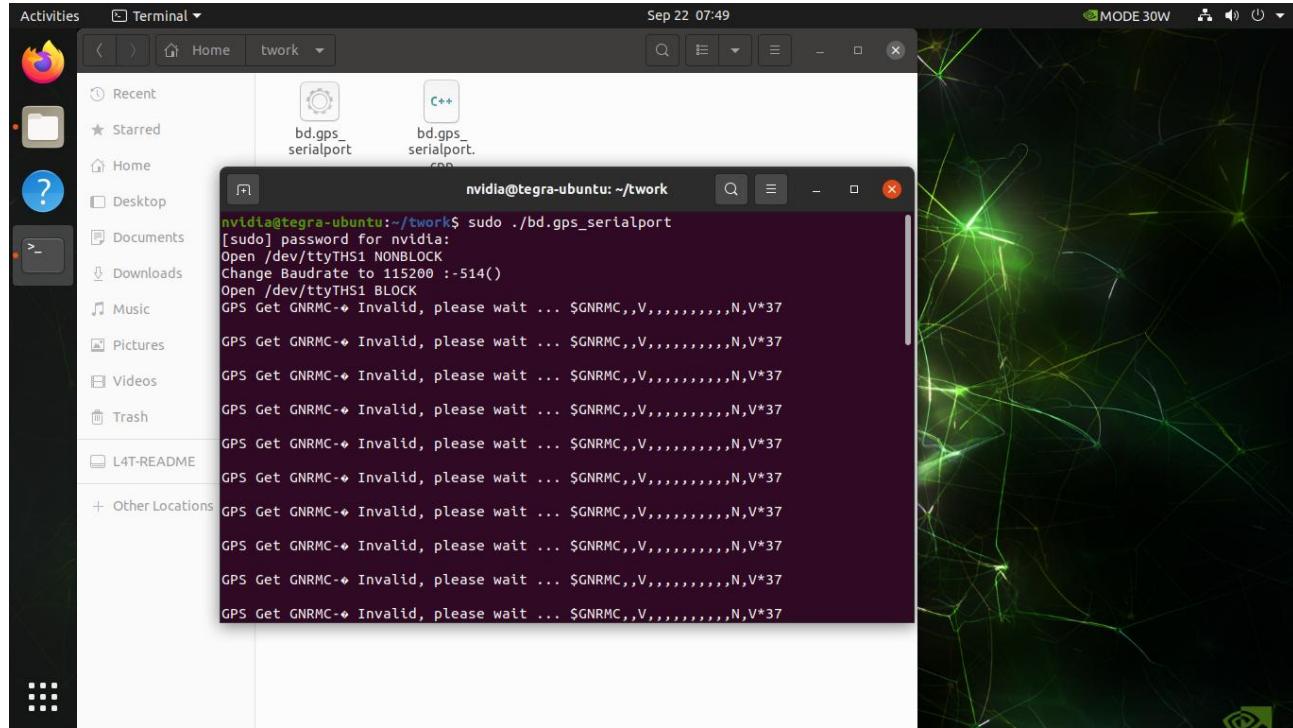
选择空白处点击鼠标右键选择”Open in Terminal”，执行命令”sudo ./bd.gps\_serialport”



执行后结果如图所示，则表示获取到定位信息：



执行后结果如下图所示则表示没有成功, 说明此地信号差, 建议换到开阔地获取信号比较定位信息;

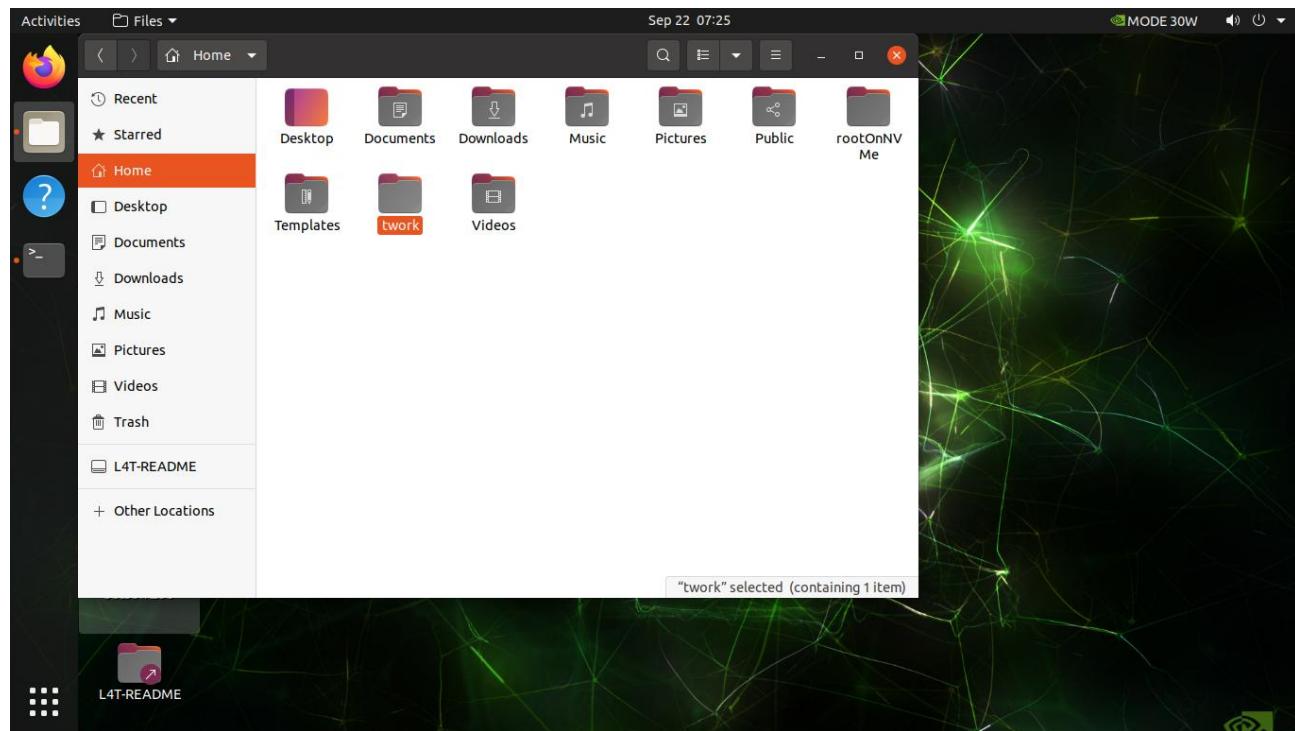


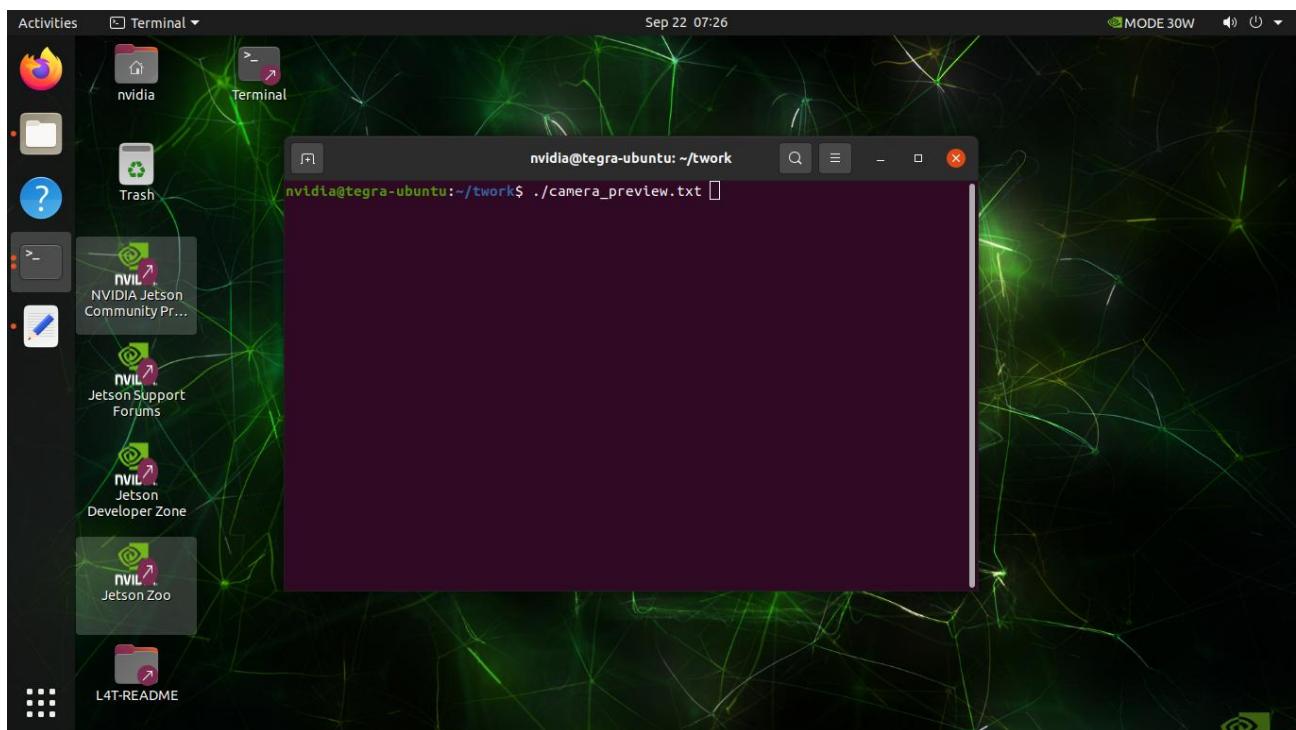
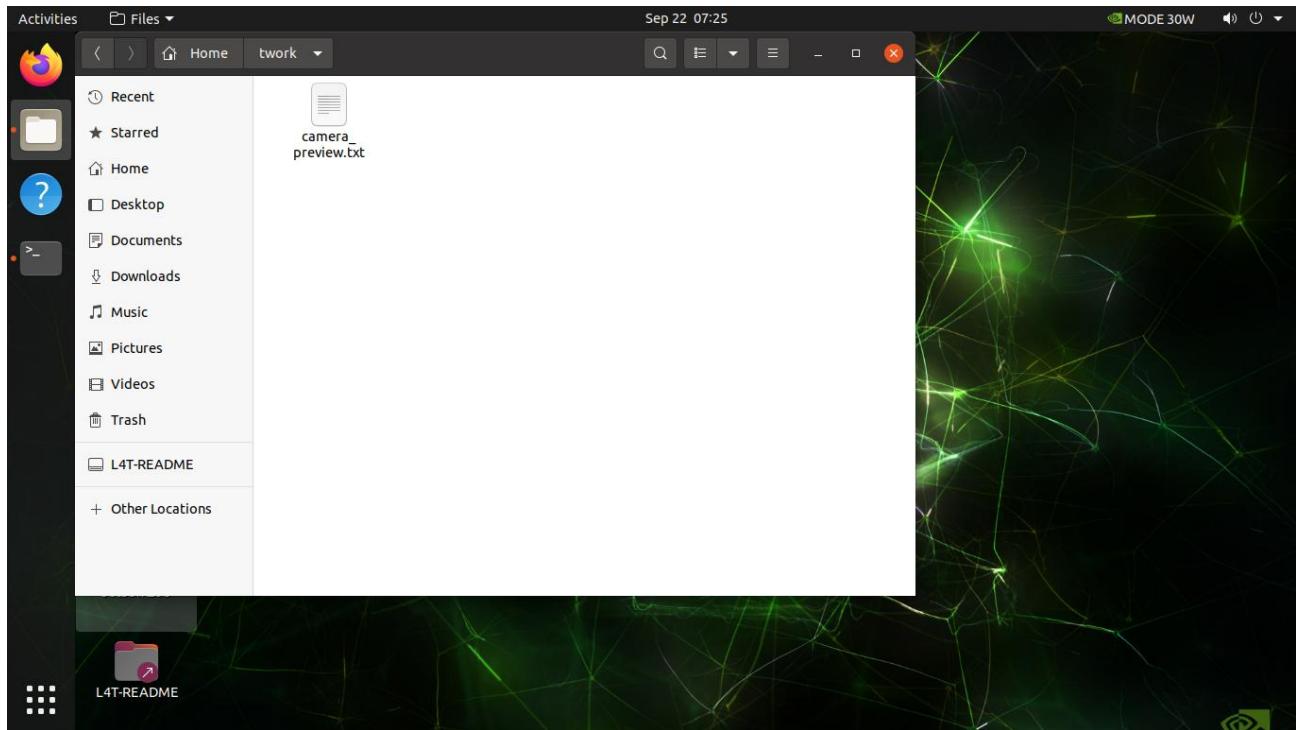
## GMSL 摄像头使用

目前根据我司研发人员的不断努力,实现在 jetson-agx-orin 设备上的 GMSL 摄像头的接入,目前我们这款 T906G 仅支持部分摄像头,后续我们会根据客户的不同需求,在出厂时给设备搭载不同的摄像头驱动,目前我们的摄像头驱动不是惟一的兼容性驱动,而是只针对一款摄像头一种驱动,每个设备也只搭载一款驱动.

摄像头预览命令:

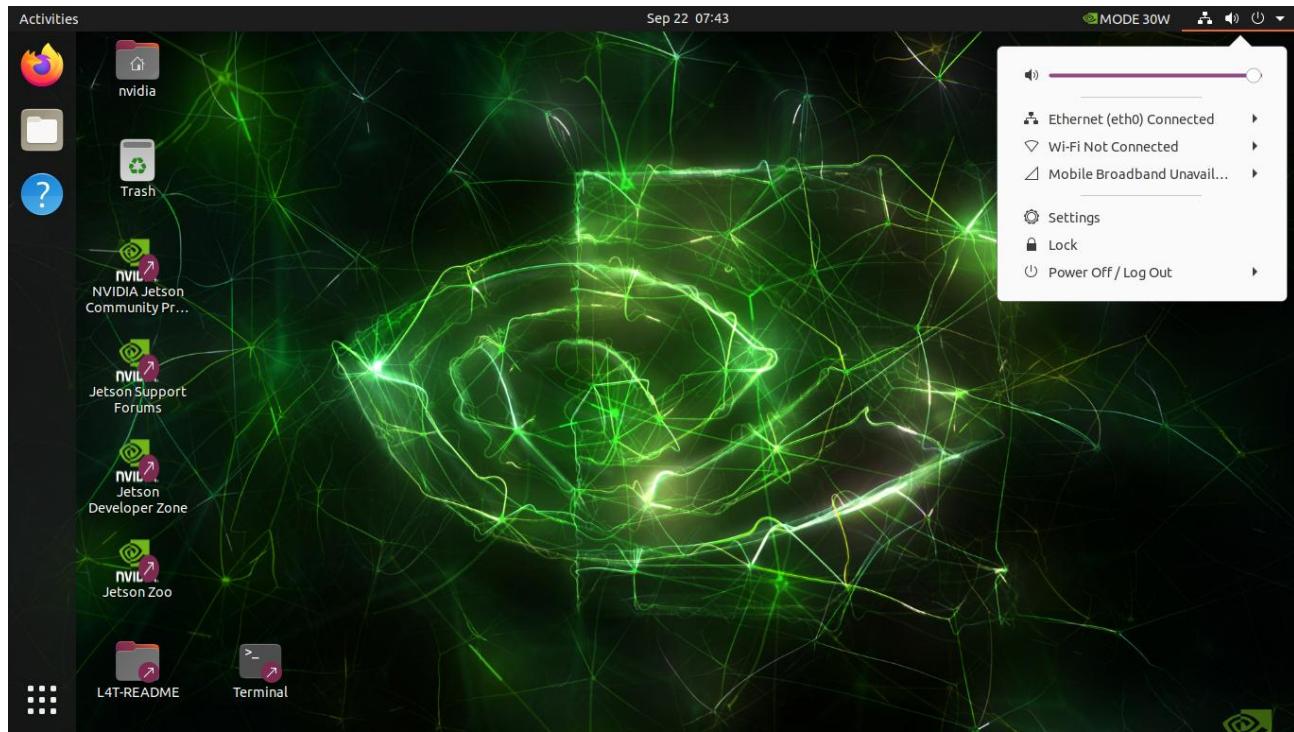
```
gst-launch-1.0 v4l2src device=/dev/video0 ! 'video/x-raw,format=UYVY,width=1920,height=1080' ! videoconvert ! fpsdisplaysink\nvideo-sink=xvimagesink sync=false &\n\ngst-launch-1.0 v4l2src device=/dev/video1 ! 'video/x-raw,format=UYVY,width=1920,height=1080' ! videoconvert ! ffpsdisplaysink\nvideo-sink=xvimagesink sync=false
```



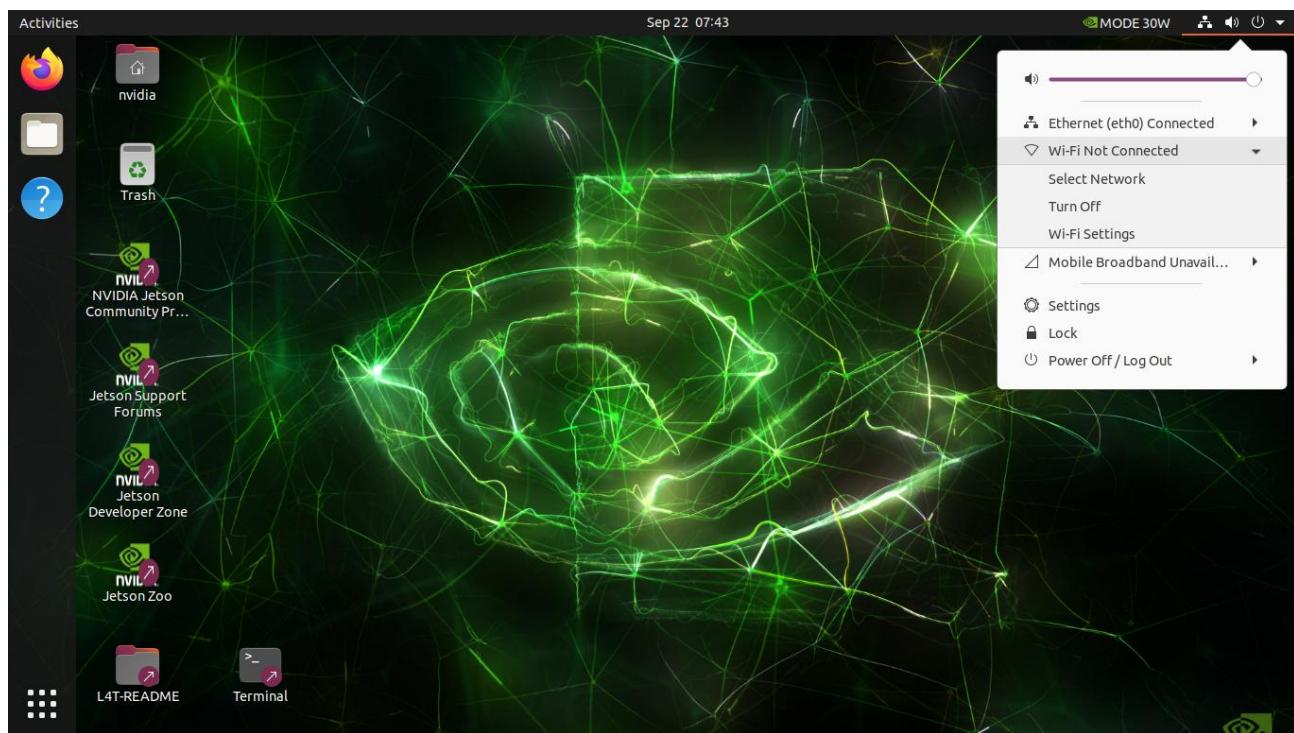


## WiFi 连接

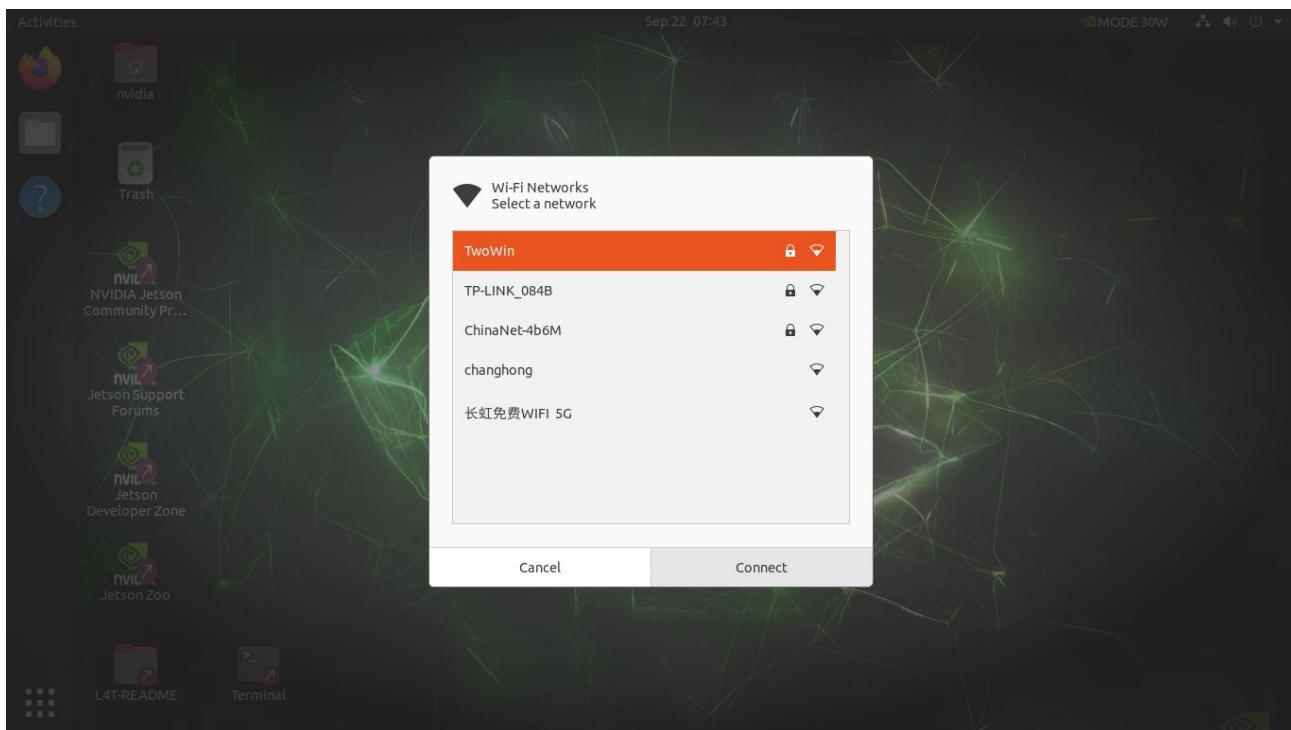
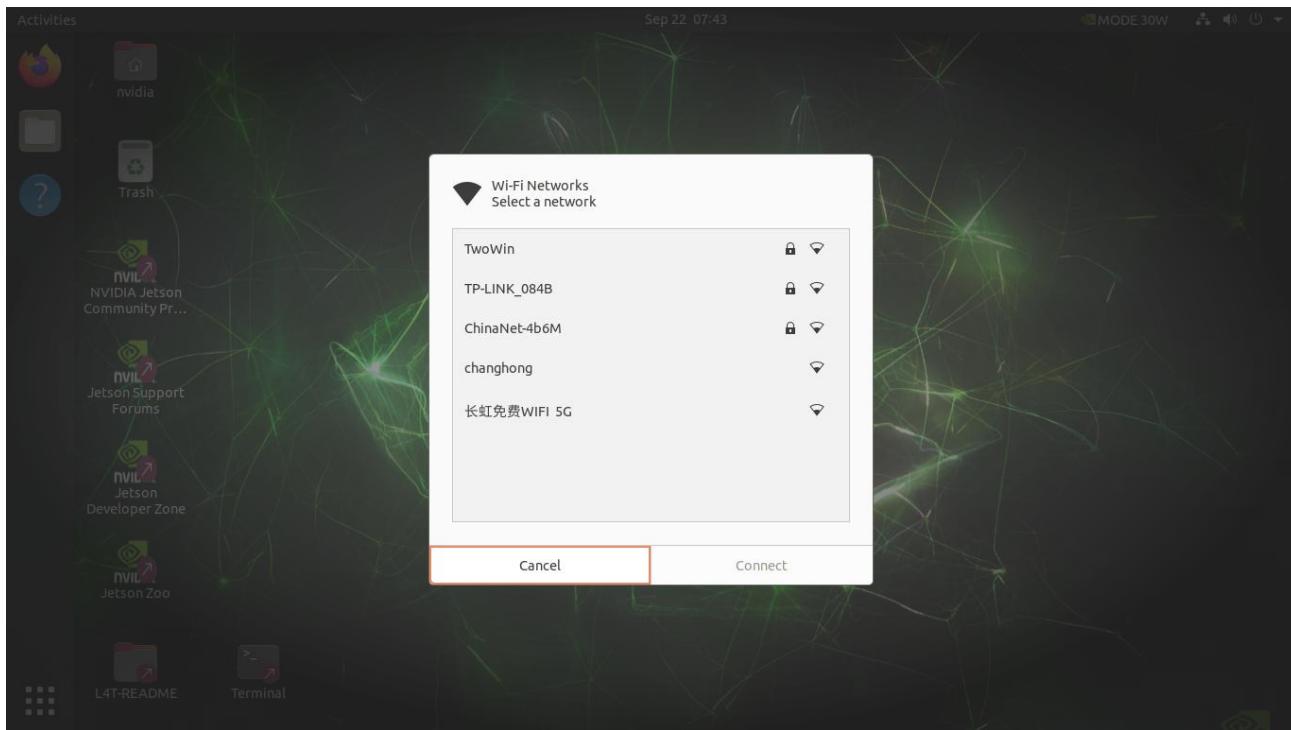
1. 点击箭头所示网络图标, 出现网络设置



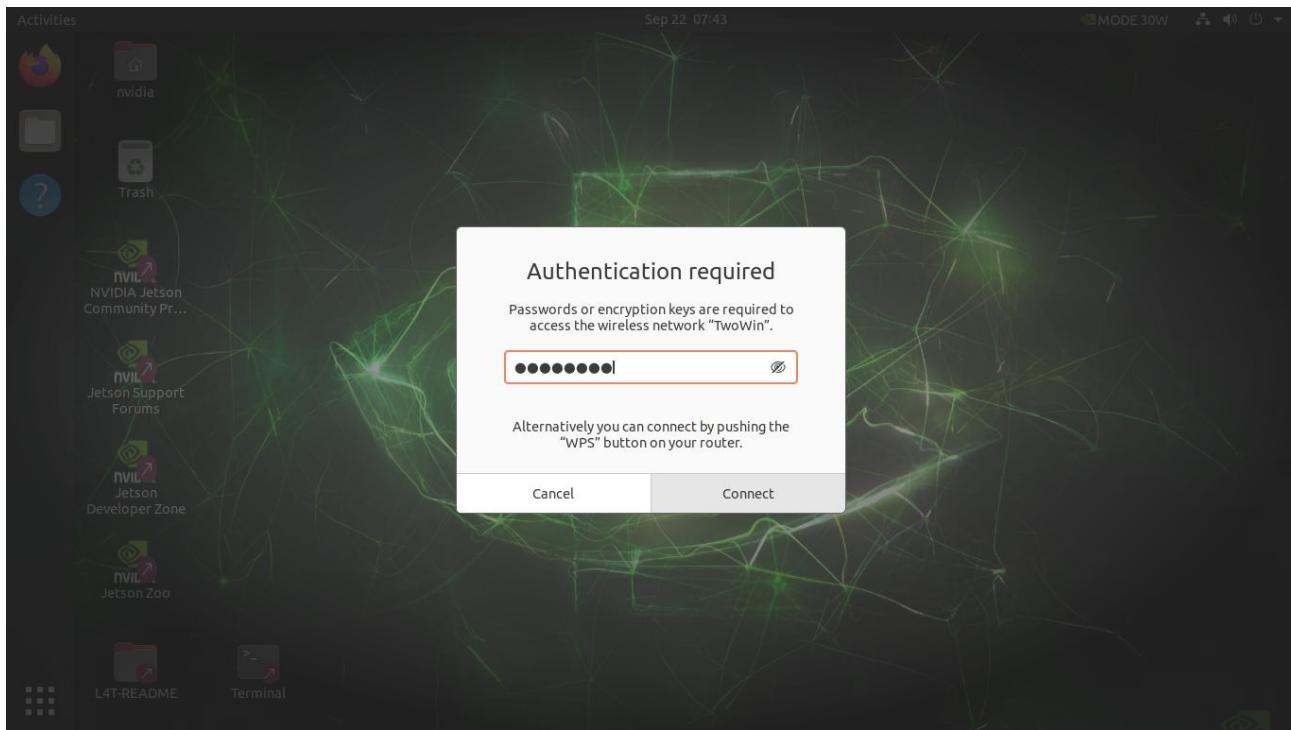
2. 选择第三项, 点击网络选择”Select Network”



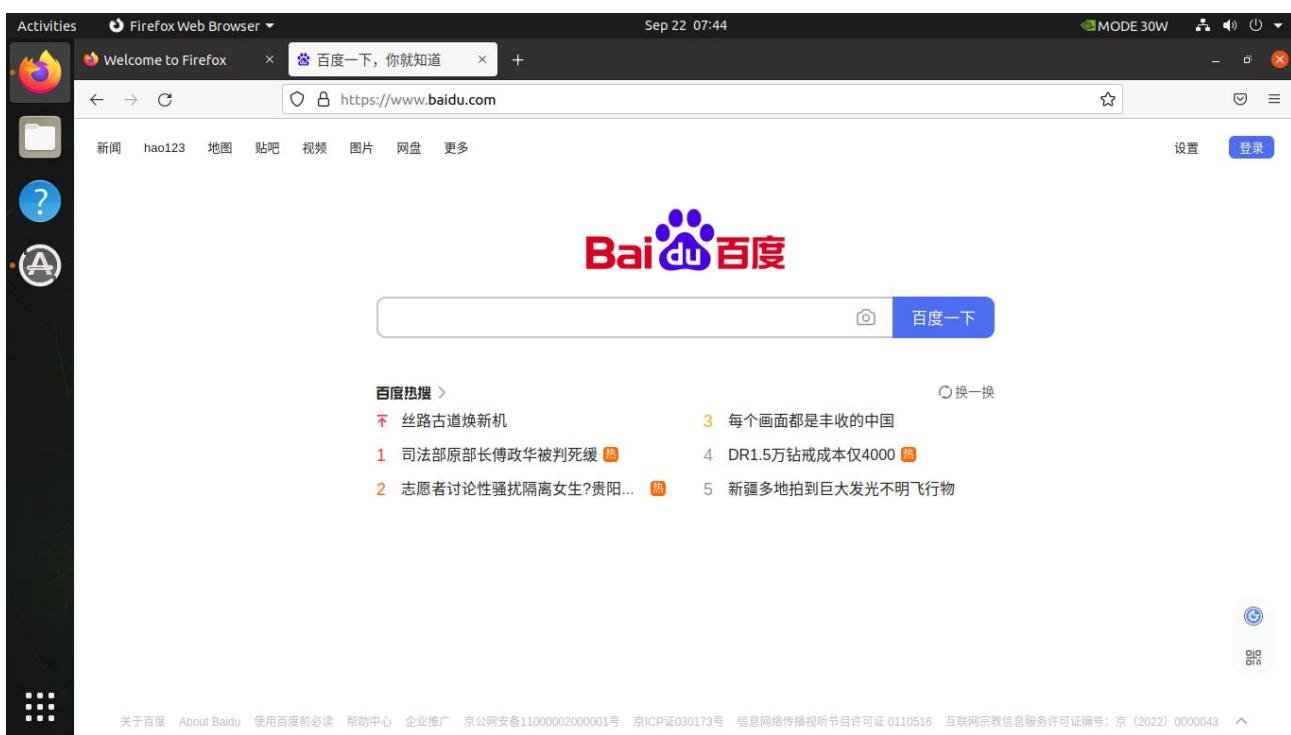
3. 选择当前搜寻到的可用网络, 找到适合的 wifi, 选中后点击” Connect ”



## 4. 输入 wifi 密码



## 5. 联网后打开网页确认是否网络正常



## 4G 拨号联网

步骤一: (以下内容以有方 n720 模块为例)

将图为提供的压缩包文件解压后复制到图为智盒设备中, 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](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:

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
#endif 4g auto dial

#Running maxn mode
/usr/bin/jetson_clocks

exit 0
```

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.

## 采用 SSD 为系统盘

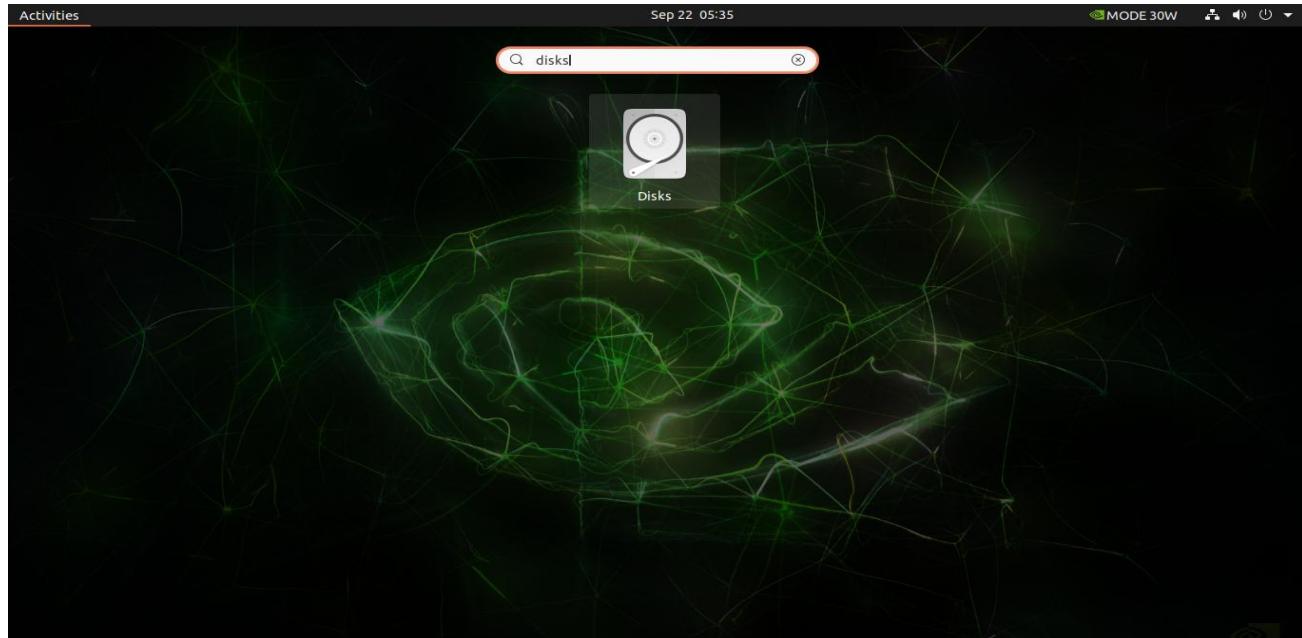
SSD 作用

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。

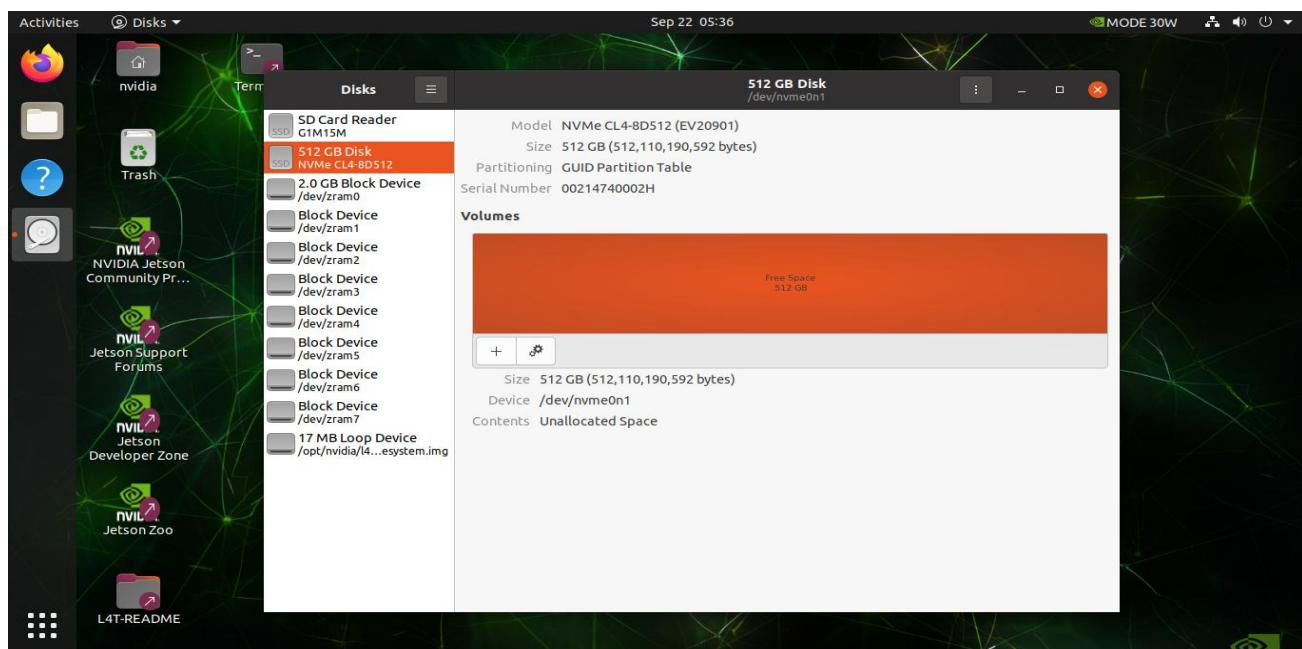
步骤一、准备 SSD 并格式化为 GPT

1. 准备 M.2 Key M SSD
2. 打开 Ubuntu20.04 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）：

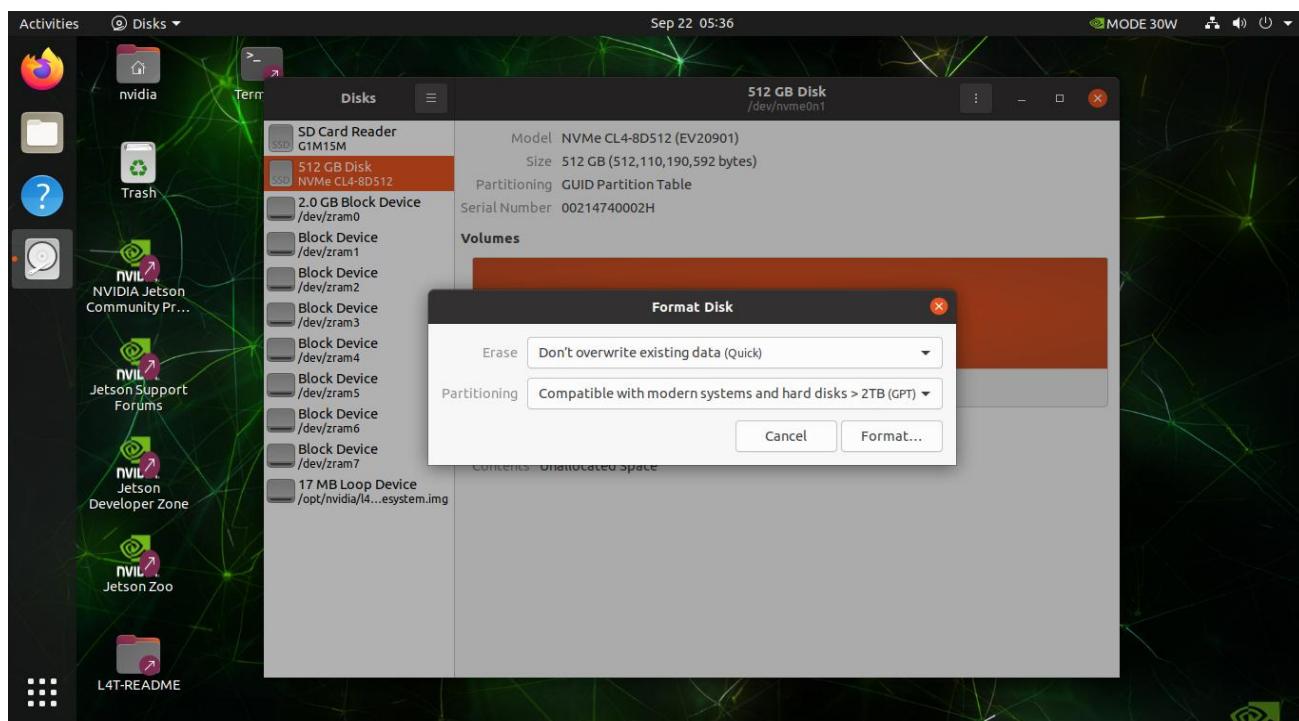
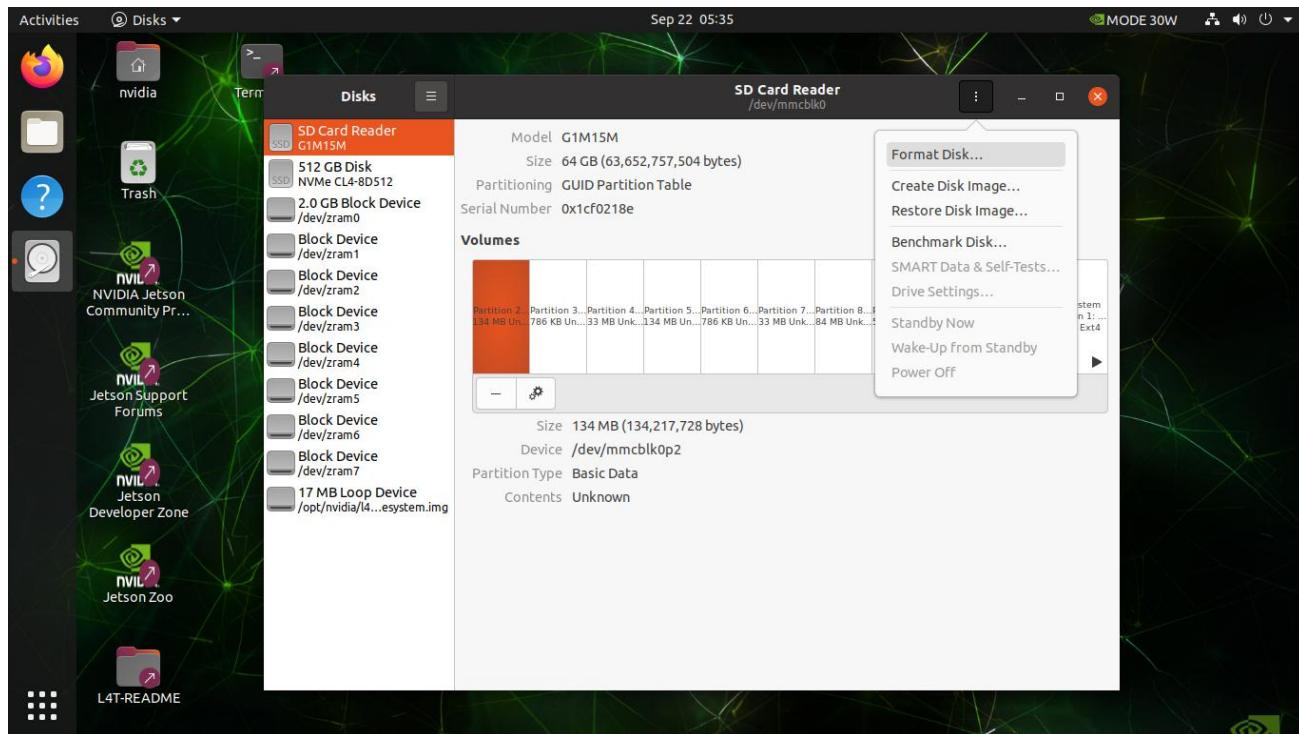
1) 首先打开方式为按 win 键出现搜索框，搜索“disks”工具；

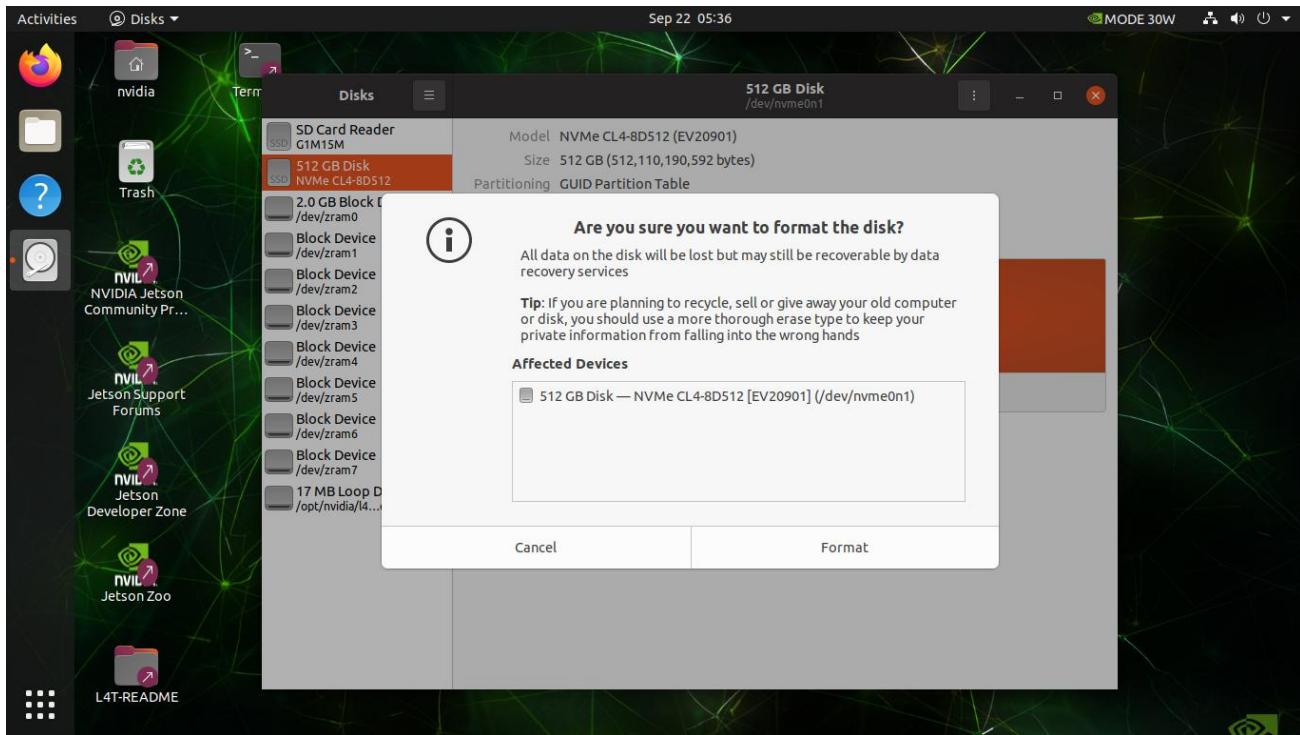


2) 进入 disks，找到安装的 ssd；

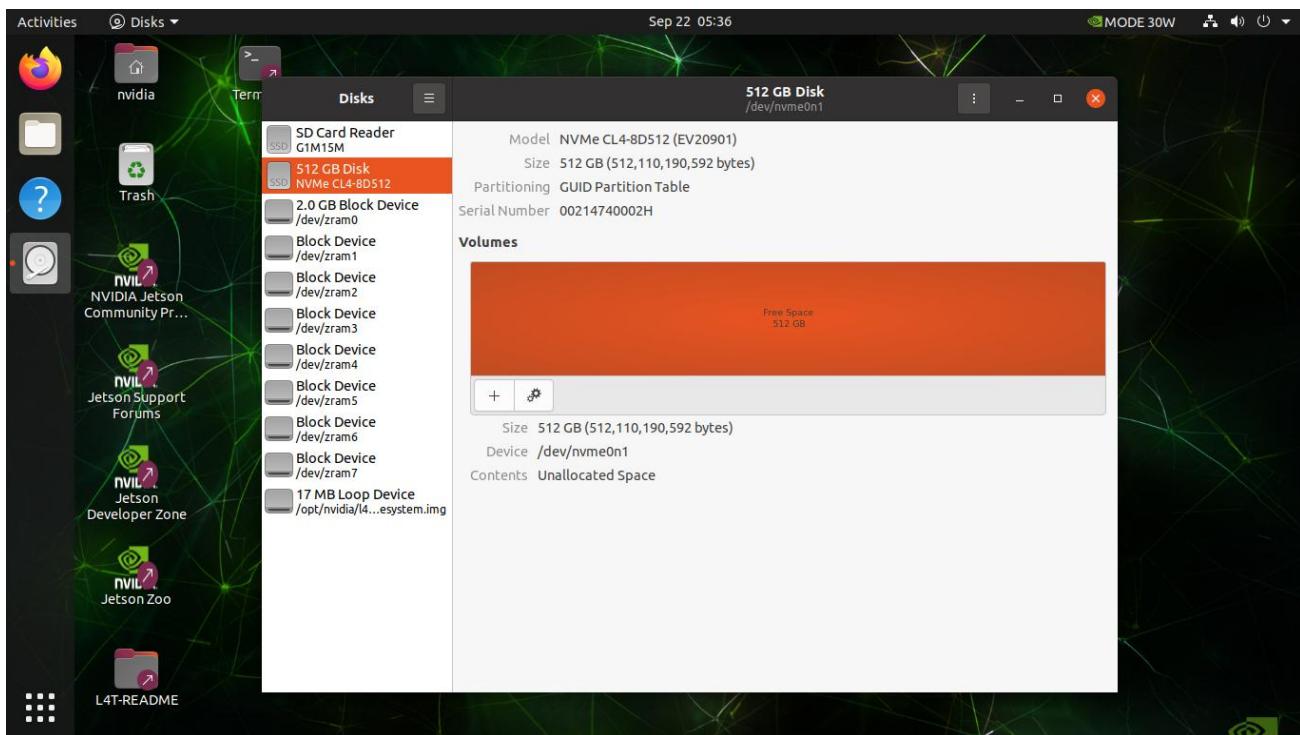


3)点击右上角：“继续点击第一个选项”Format Disk”，然后出现弹窗点击”Format”

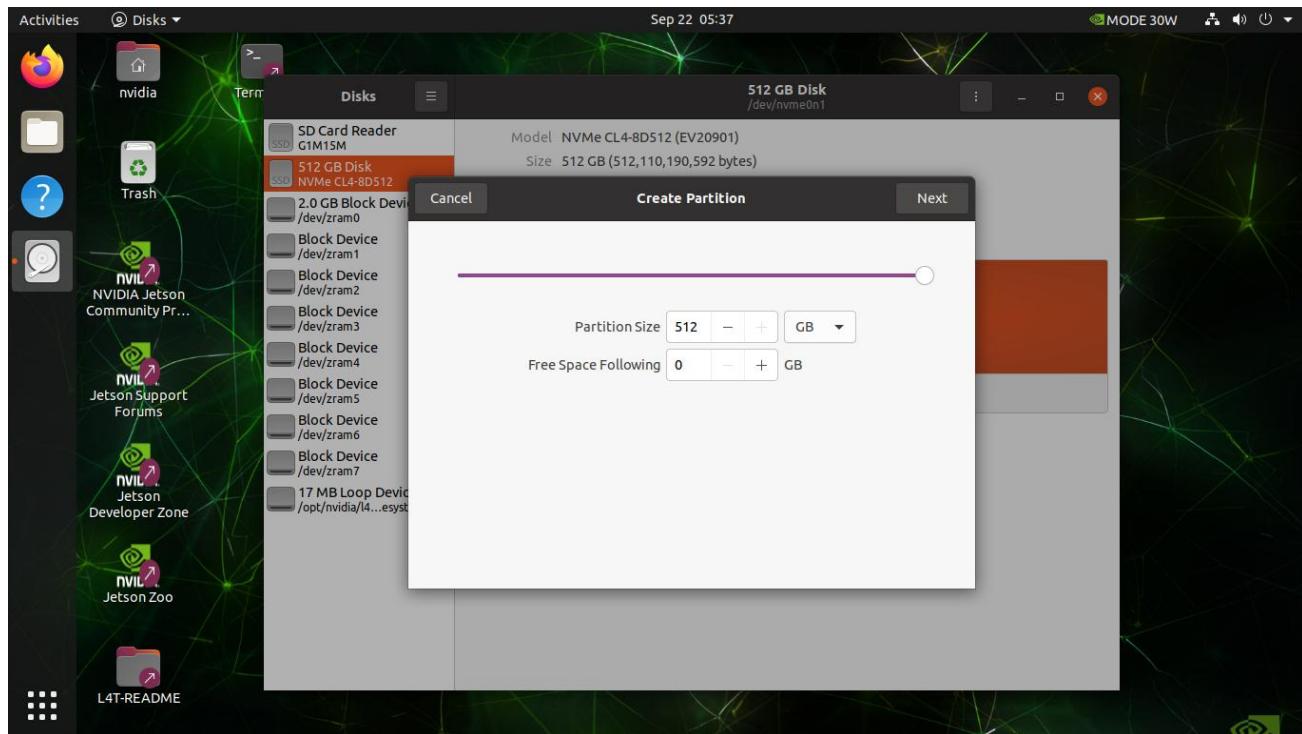




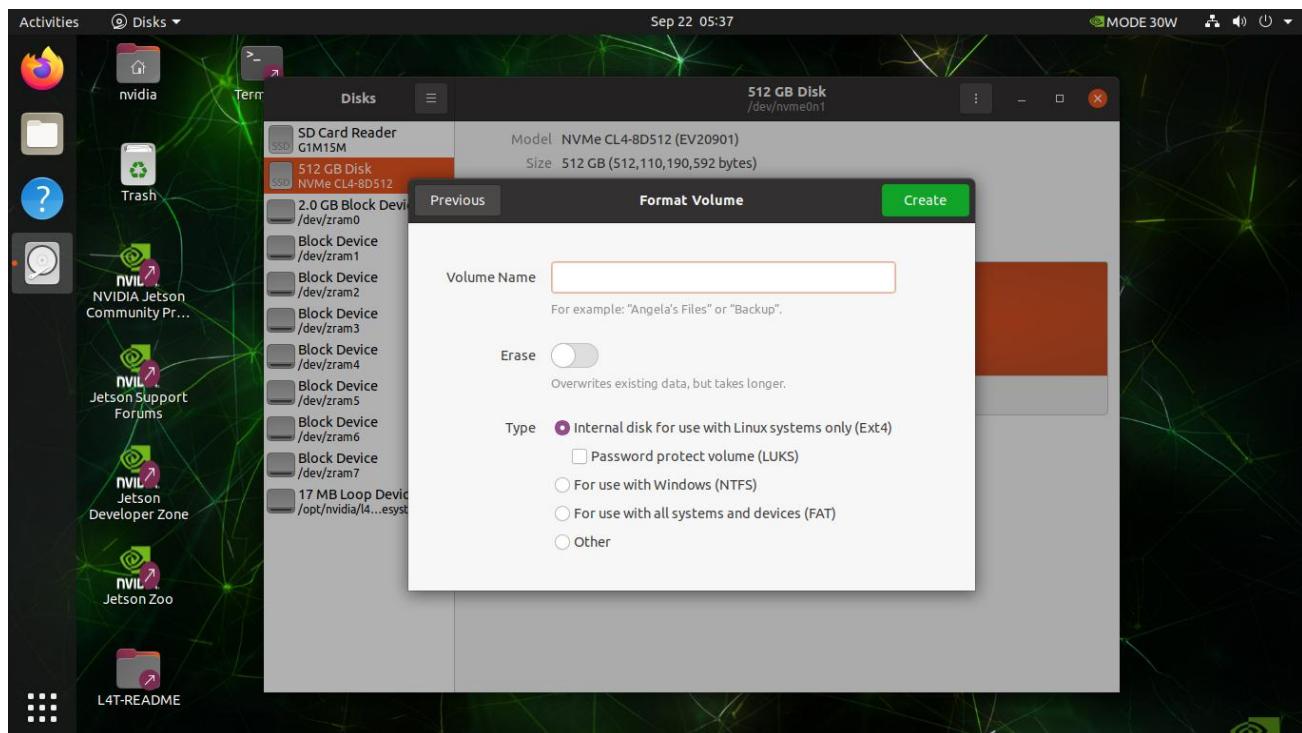
3) 格式化完毕后, 点击”+”新建分区,



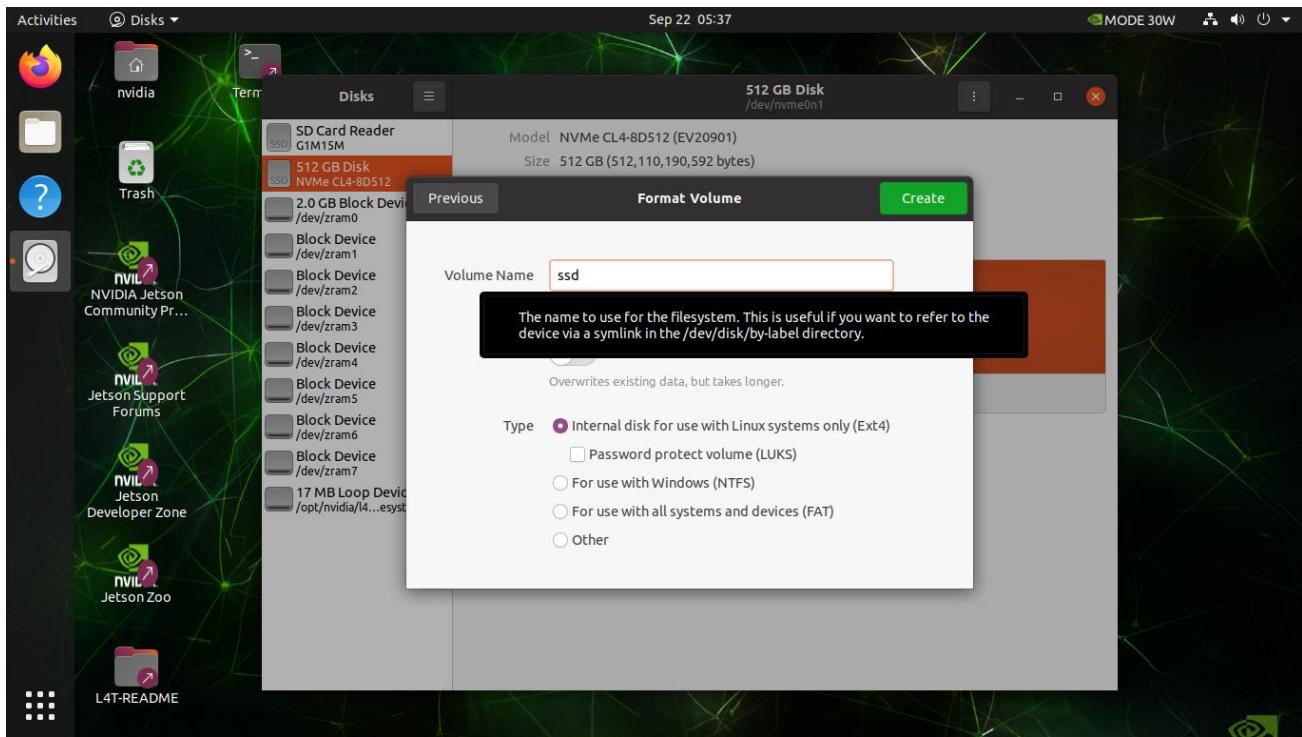
4)划分新建分区大小，默认全部分配，这个建议默认就好，如果需要更改，就根据自身需求进行合理分配，然后点击“Next”



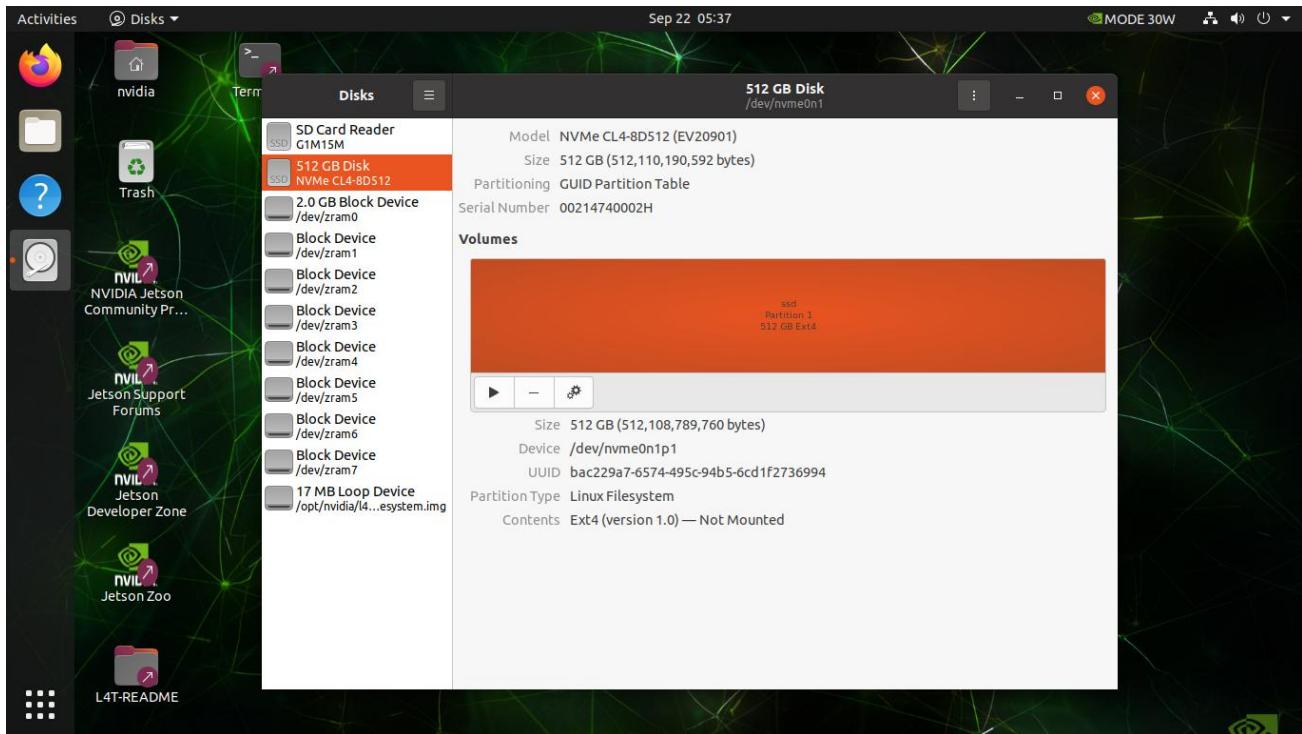
5)选择文件格式，默认为“ext4”，这个无需更改：



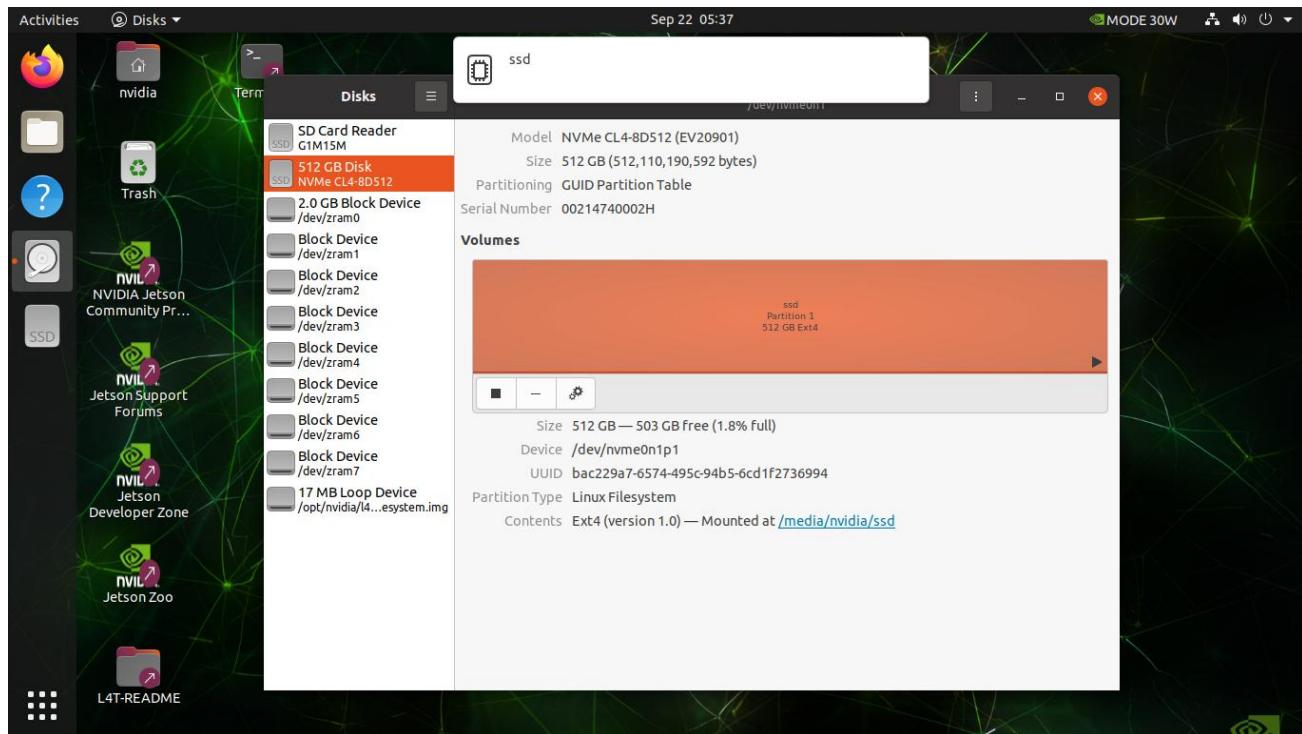
6)给新分区命名,这个按照自身意愿命名即可;



7)创建分区完毕后出现下图所示画面,点击三角符号挂载

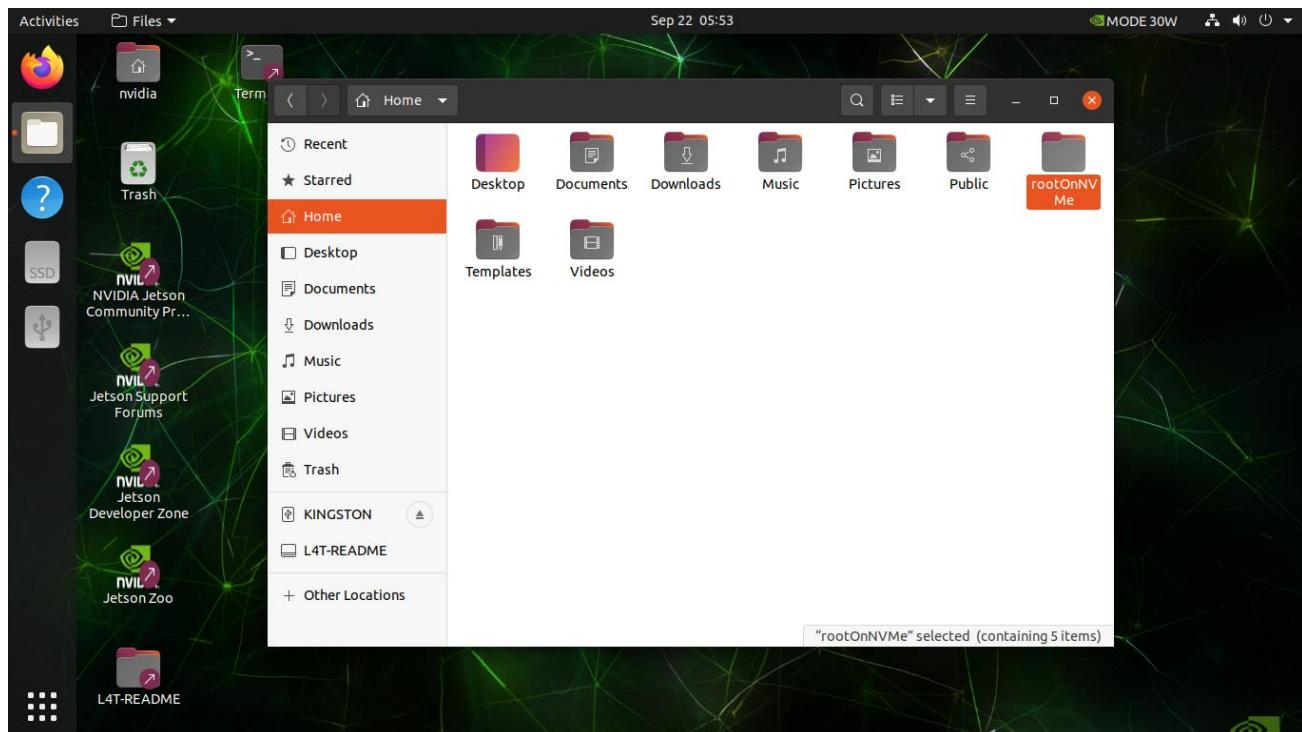


8) 如下图显示则为挂载成功

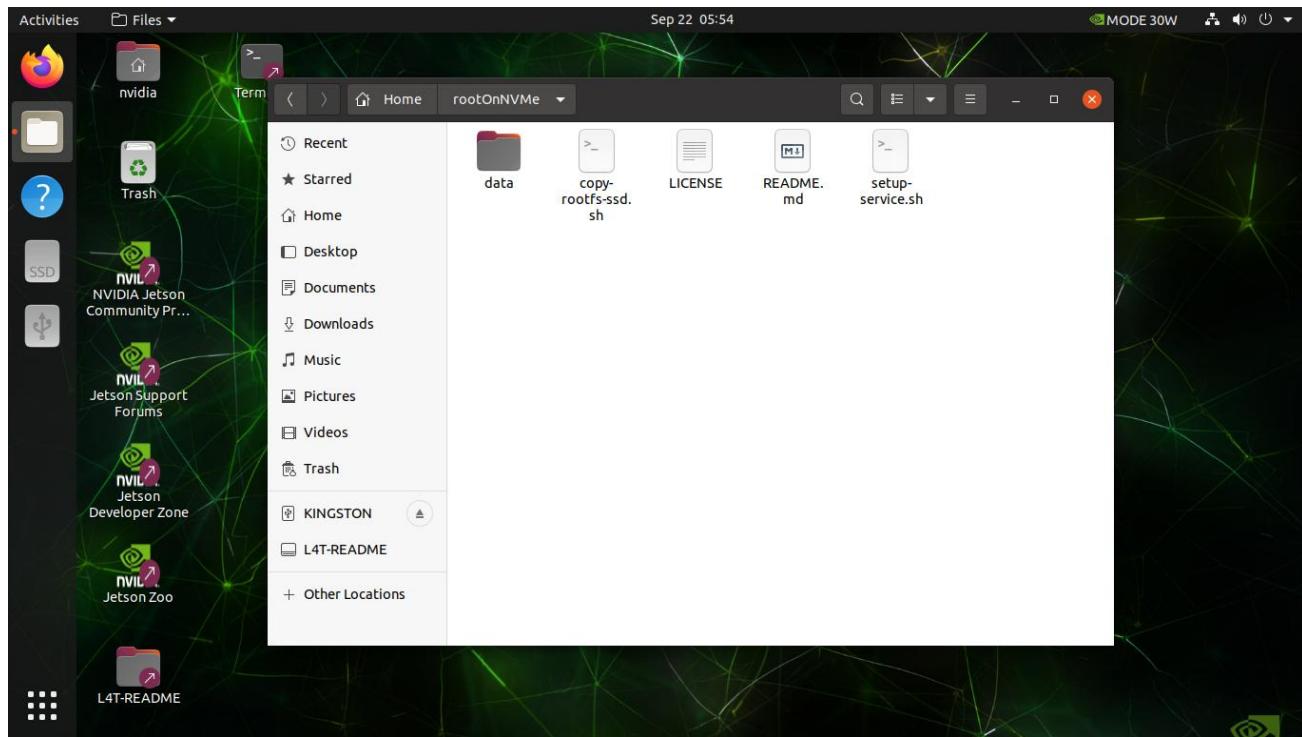


## 步骤二：(系统盘转换)

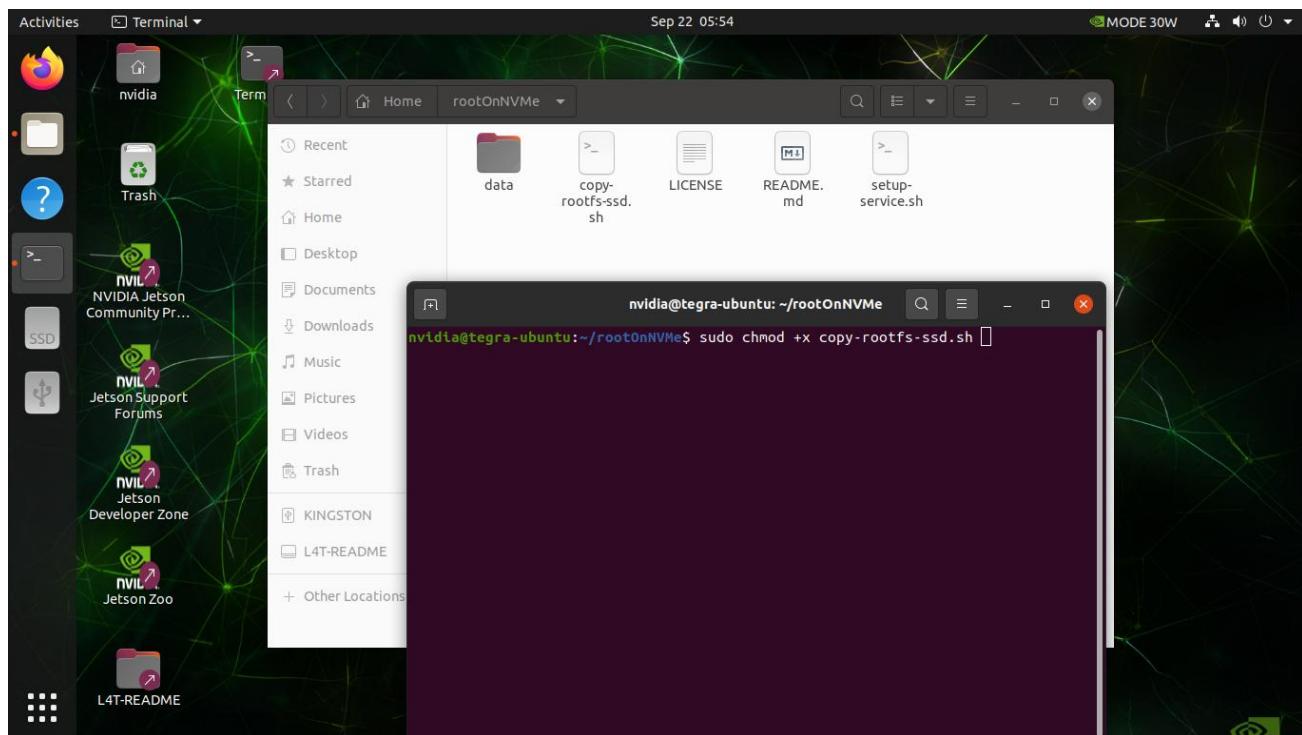
1) 进入/home 目录, 找到 rootOnNVMe 文件



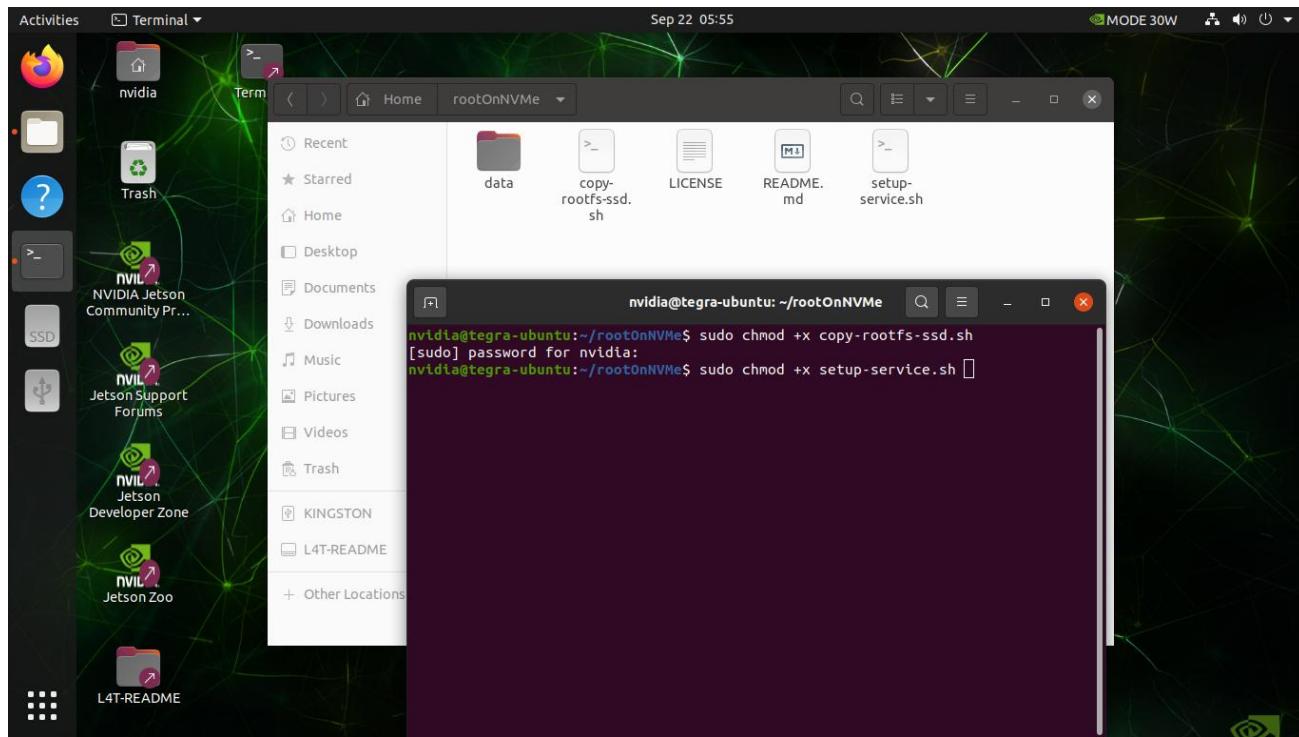
2)进入该文件夹；



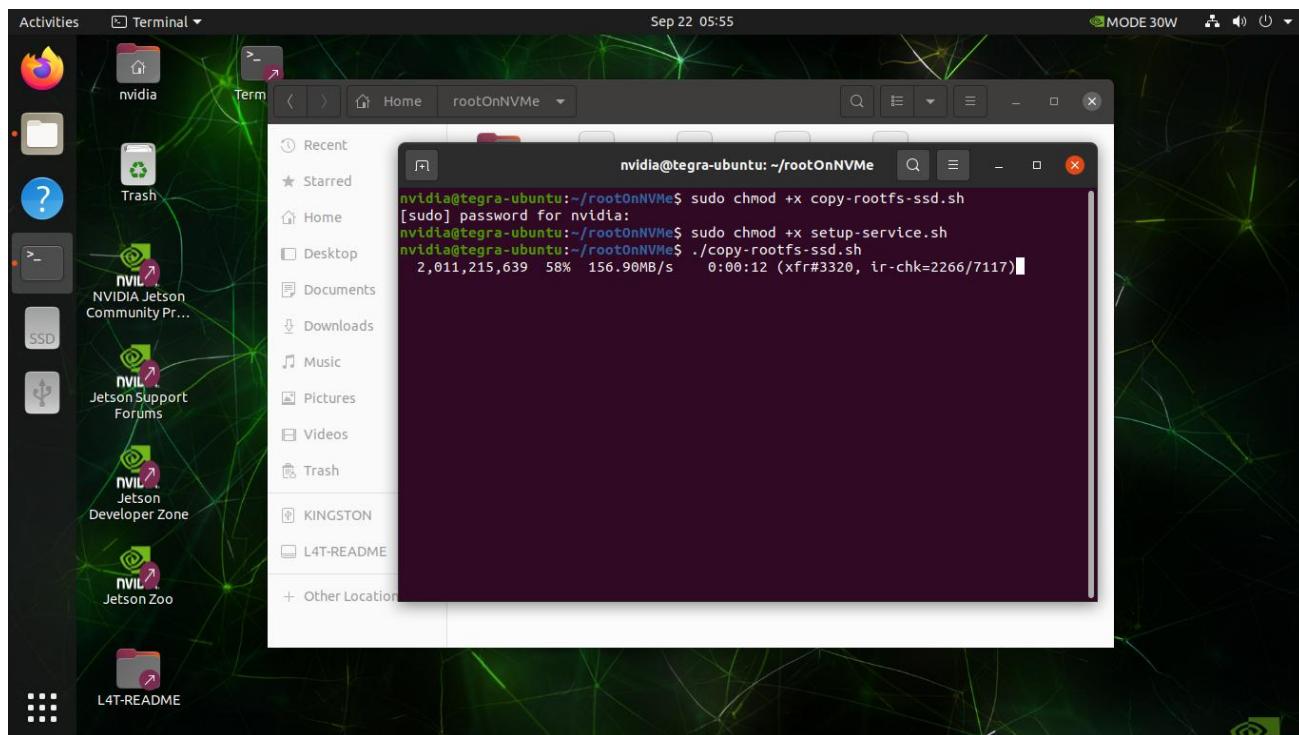
3)在文件夹空白处右键选择”Open in Terminal”，执行命令：“sudo chmod +x copy-rootfs-ssd.sh”

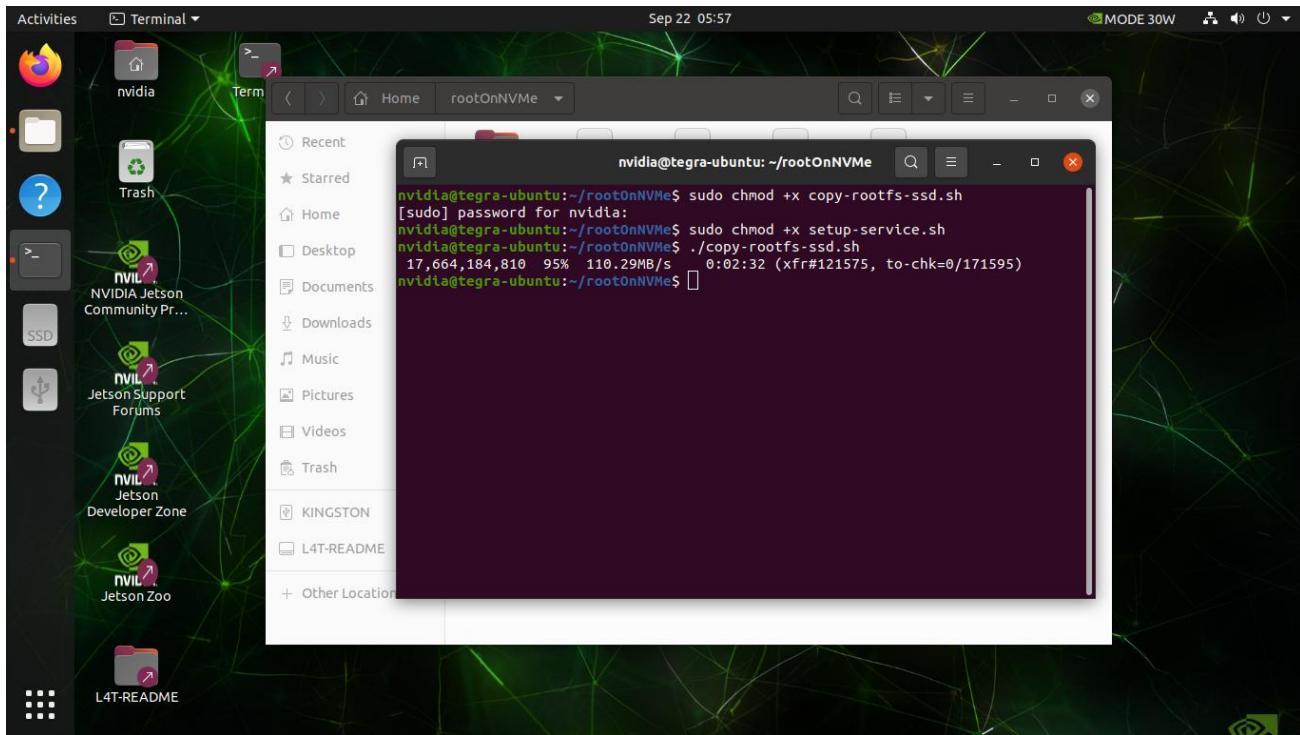


4) 执行命令：“sudo chmod +x setup-service.sh”

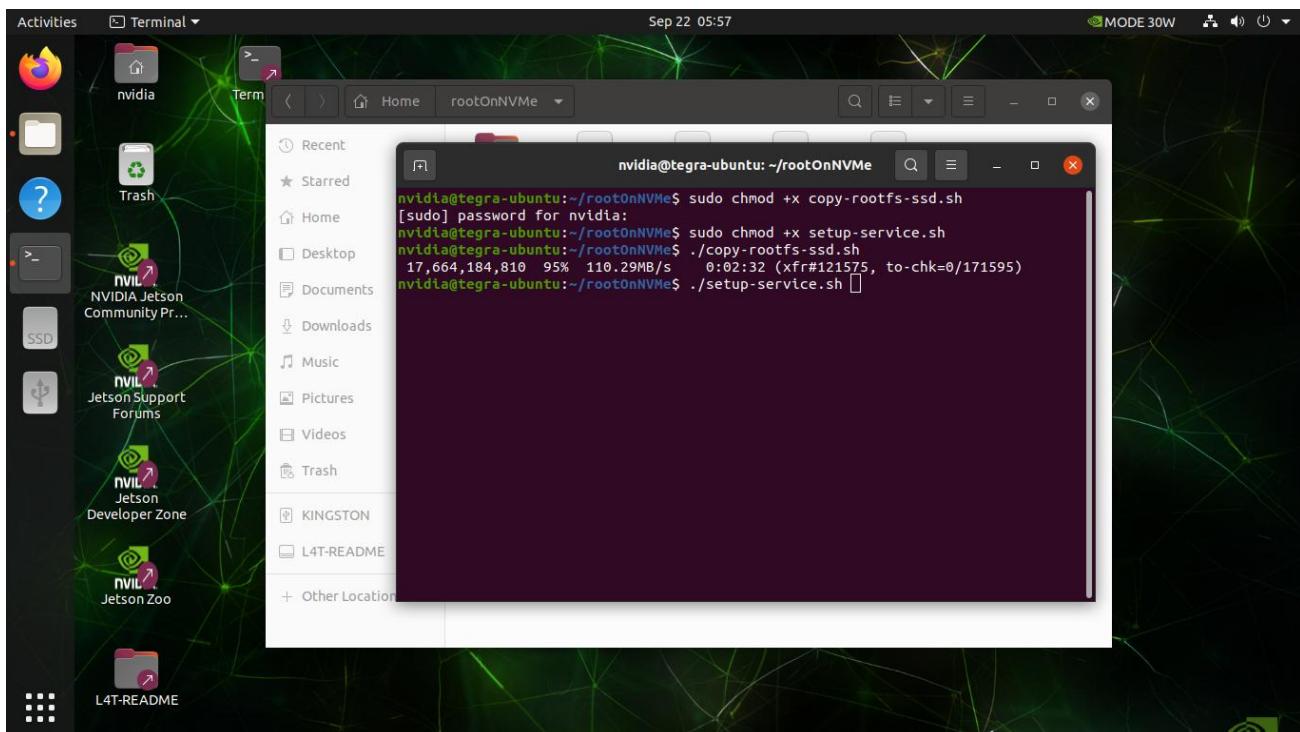


5) 执行脚本：“copy-rootfs-ssd.sh”

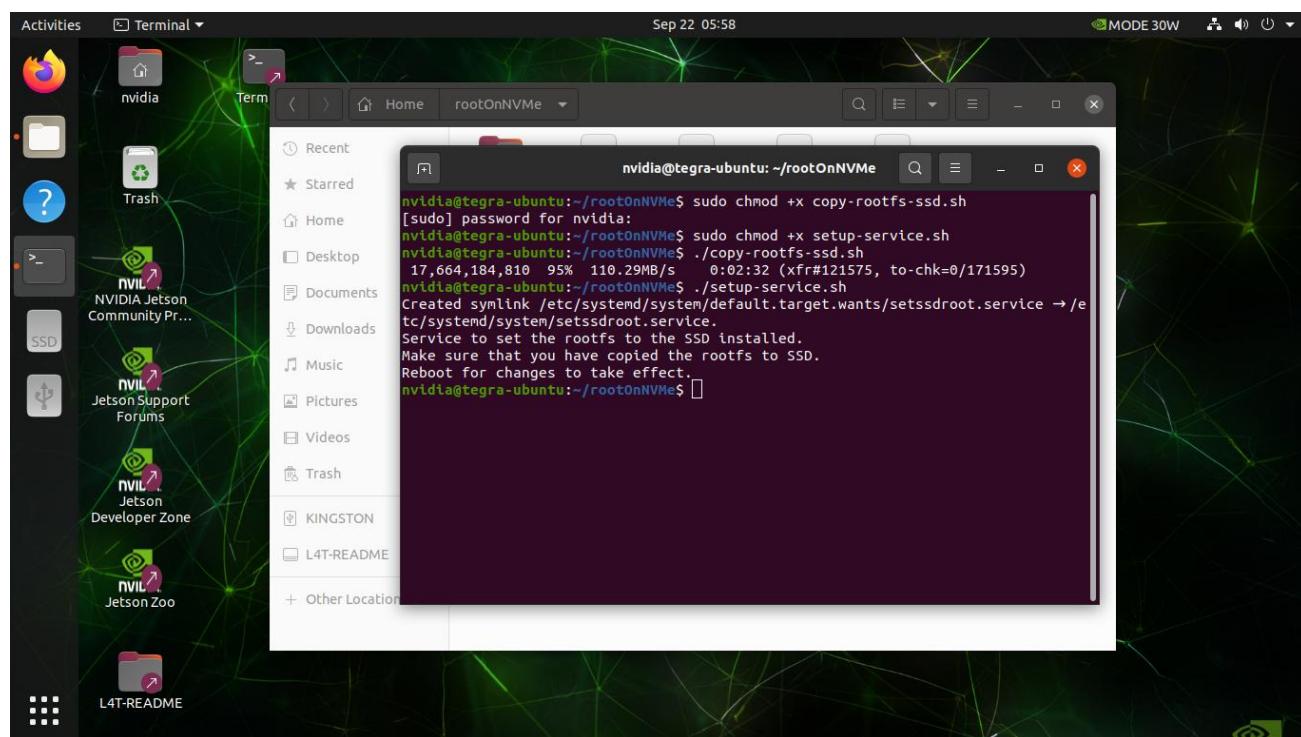
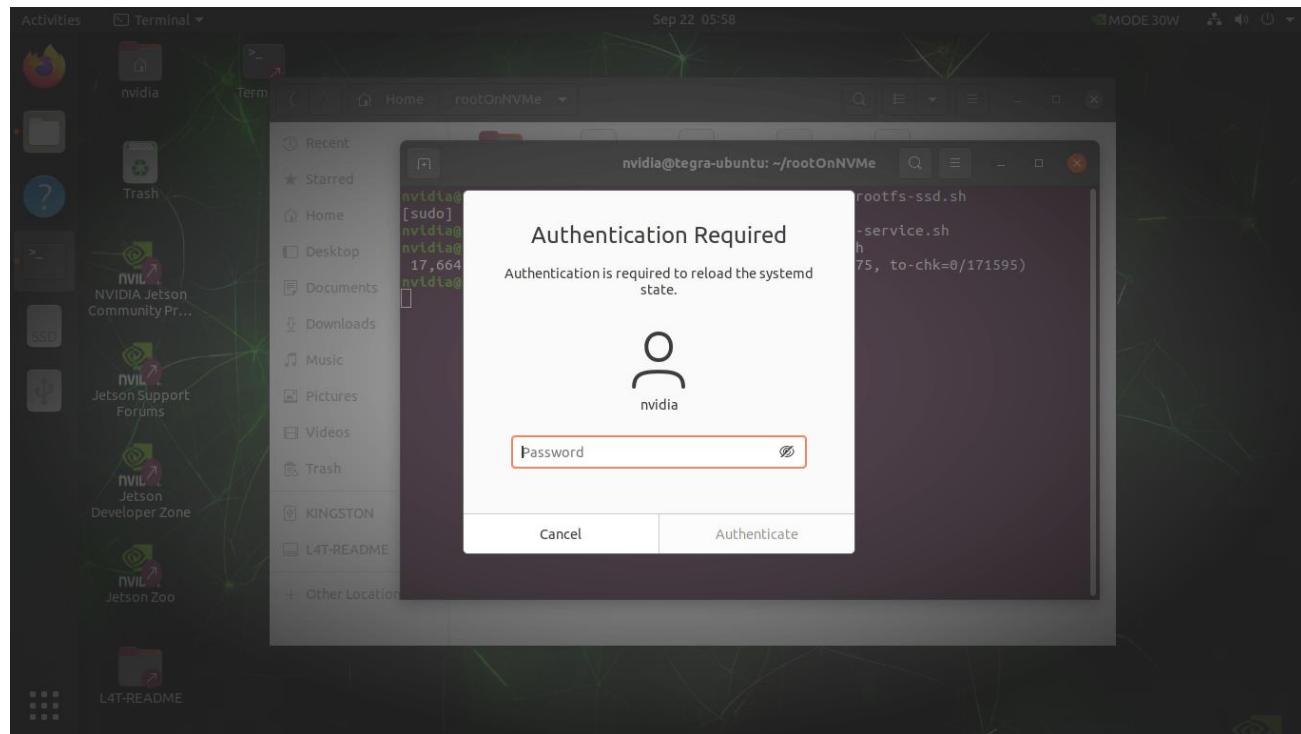




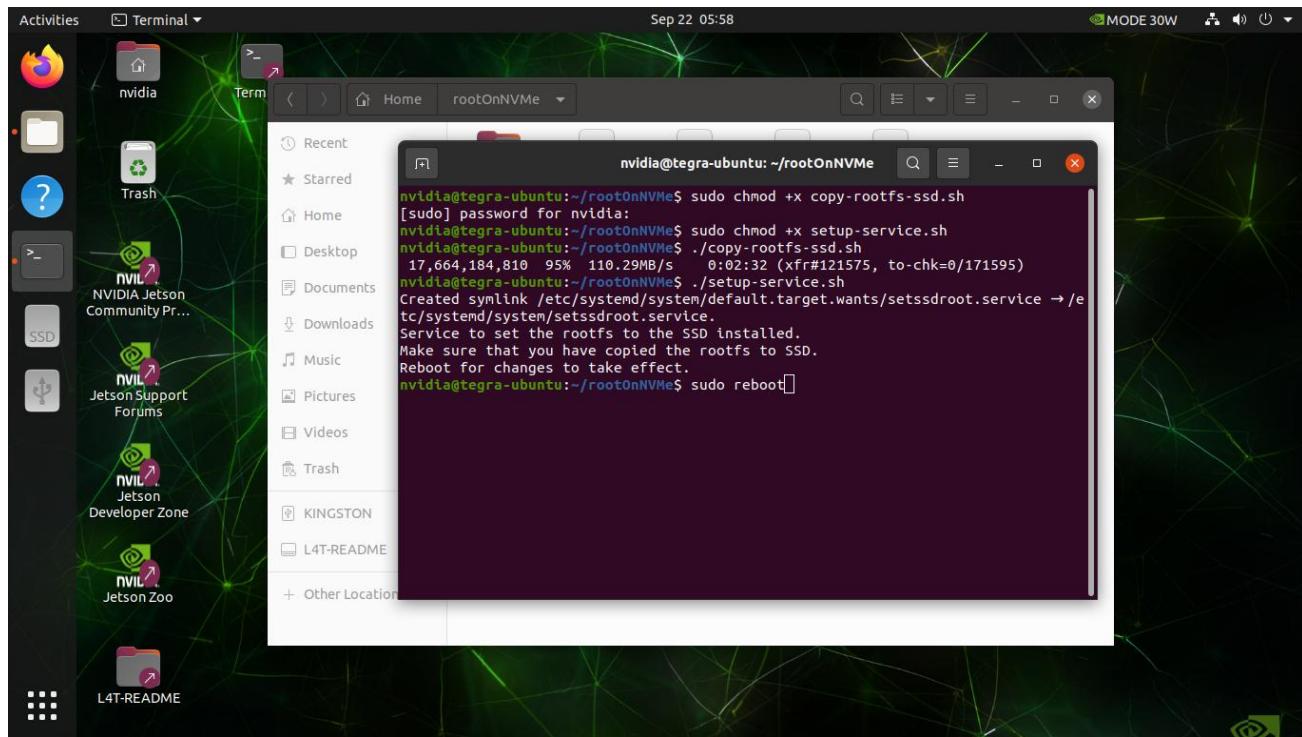
6) 执行脚本：“setup-service.sh”



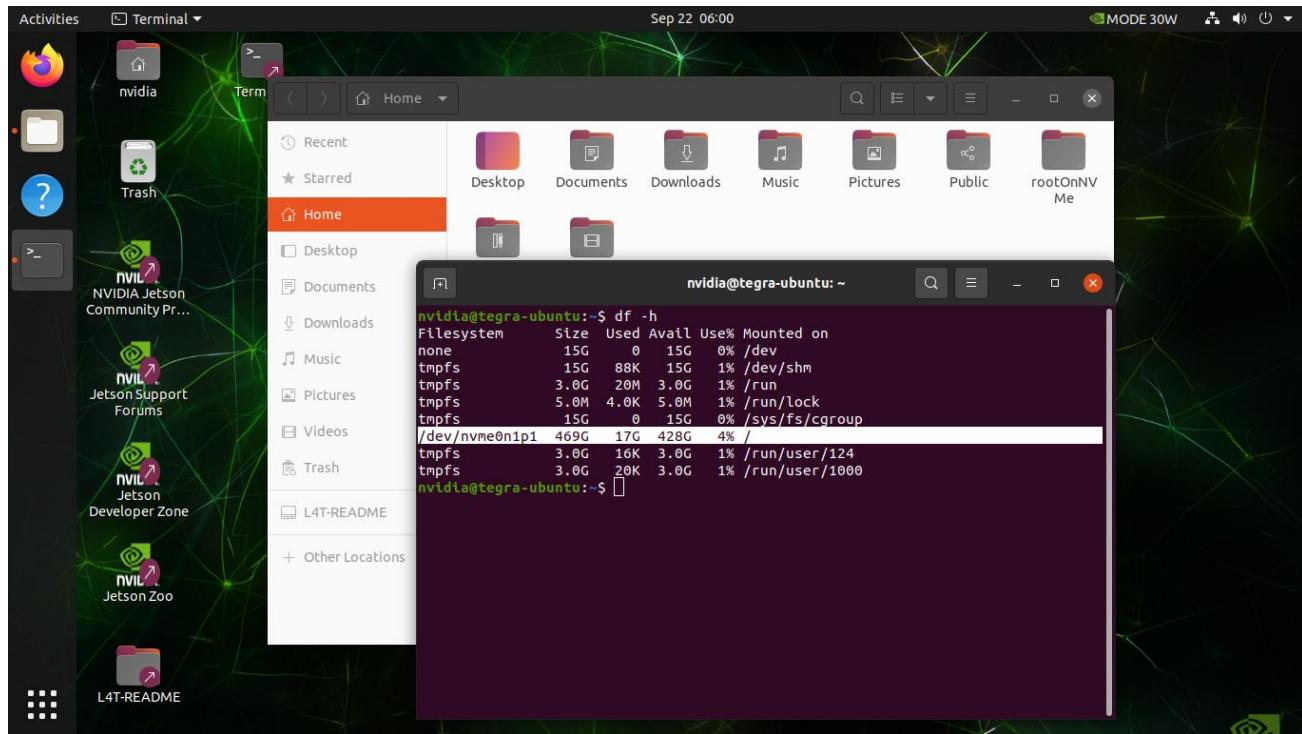
7) 输入密码，默认为”nvidia”



8) 输入命令 "sudo reboot"



9) 等待重启完成后, 打开终端输入命令 "df -h" 查看存储分布情况, 如下图显示则为成功



## 系统安装

我司 T906G 产品系公司自行研发主板搭配 nvidia agx orin 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 或 20.04 系统的电脑 1 台(虚拟机也可以)

Type-c usb 数据线 1 条

### 设置下载模式

Nvidia jetson 是通过 usb type-c 接口升级系统, 更新前需让 T906G 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。

T906G 进入 recovery 模式的步骤:

- 1) 连接 T906G 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 T906G 的 OTG 口, 一端插在 ubuntu host 主机的 usb 3.0 插口);
- 3) 使用 T906G 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) 2 秒后释放 reset 复位按键, 最后释放 recovery 按键, 此时 T906G 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 corp 设备 (不同 jetson 模块 usb vid/pid 不同) 来确认是否进入正常)

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:

下载百度网盘系统镜像, 将此镜像文件 copy 至 ubuntu 电脑;

链接: <https://pan.baidu.com/s/10kWqcxHHCGMT15Pbsnmpg>

提取码: ht5h

链接打开后如图所示, 需全部下载.



第二步:

在 ubuntu 电脑上执行以下命令操作:

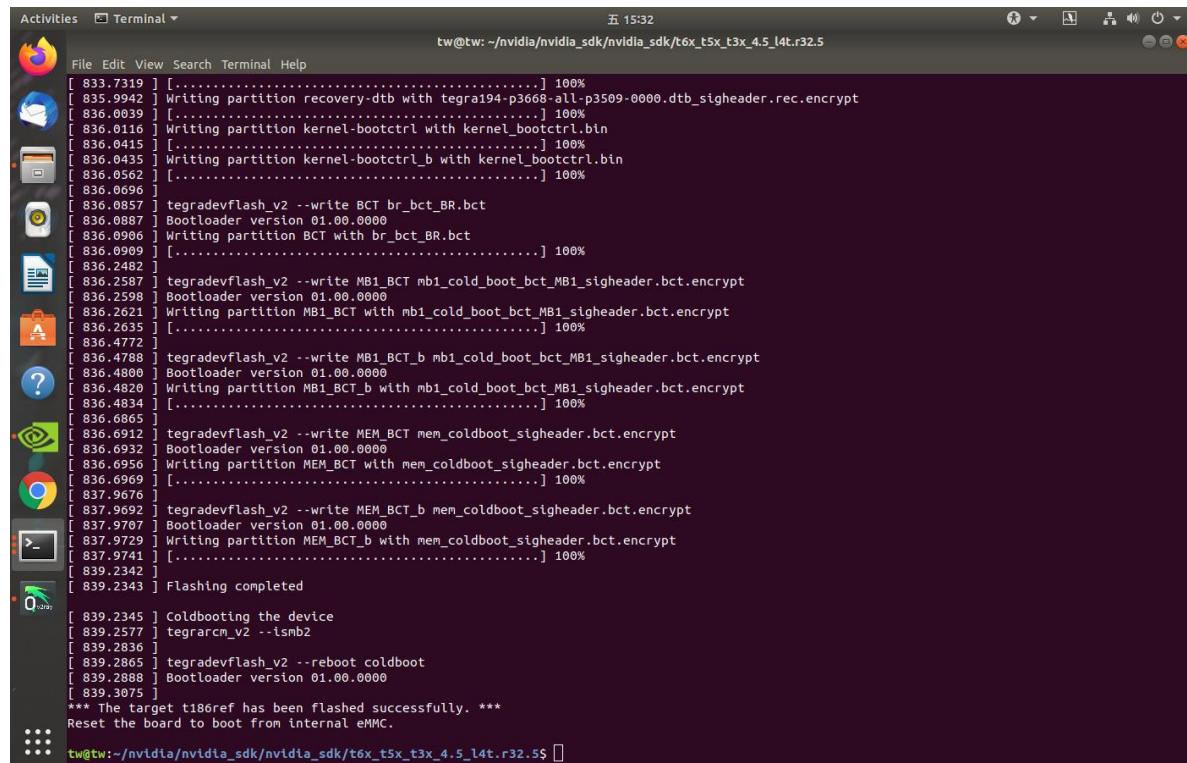
合并压缩包: cat T906G\_JP5.0.2.tar.gz\* > T906G\_JP5.0.2.tar.gz

解压: sudo tar -zxvf T906G\_JP5.0.2.tar.gz

cd Linux\_for\_Tegra

执行刷机脚本 sudo ./flash.sh jetson-agx-orin-devkit mmcblk0p1

刷机完成后如下图所示, 等待刷机完成后重启, 接入显示器判断刷机是否完全成功.



```
五 15:32
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 ] [.....] 100%
[ 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 ] [.....] 100%
[ 836.2587 ] tegradevflash_v2 --write MB1_BCT mbi_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 836.2598 ] Bootloader version 01.00.0000
[ 836.2621 ] Writing partition MB1_BCT with mbi_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 836.2635 ] [.....] 100%
[ 836.4772 ] [.....] 100%
[ 836.4788 ] tegradevflash_v2 --write MB1_BCT_b mbi_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 836.4800 ] Bootloader version 01.00.0000
[ 836.4820 ] Writing partition MB1_BCT_b with mbi_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 836.4834 ] [.....] 100%
[ 836.6865 ] [.....] 100%
[ 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 ] [.....] 100%
[ 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 ] [.....] 100%
[ 839.2343 ] Flashing completed

[ 839.2345 ] Coldbooting the device
[ 839.2577 ] tegrarcm_v2 --ismb2
[ 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$
```

## PaddlePaddle FastDeploy

FastDeploy 是一款全场景、易用灵活、极致高效的 AI 推理部署工具。提供开箱即用的云边端部署体验，支持超过 150+ Text, Vision, Speech 和跨模态模型，并实现端到端的推理性能优化。包括图像分类、物体检测、图像分割、人脸检测、人脸识别、关键点检测、抠图、OCR、NLP、TTS 等任务，满足开发者多场景、多硬件、多平台的产业部署需求。

Image Classification	Object Detection	Semantic Segmentation	Portrait Segmentation
			
Image Matting	Real-Time Matting	OCR	Face Alignment
			
Pose Estimation	Behavior Recognition	NLP	Speech
		<pre> from paddlenlp import Taskflow seg = Taskflow("word_segmentation") seg("PaddleNLP的Taskflow是一个全新的NLP任务，包含词性标注、命名实体识别、关系抽取等。") ner = Taskflow("ner") ner("早上好")     </pre>	<p>input : 早上好今天是2020/10/29, 最低温度是-3°C。</p> <p>output:</p> 

PaddlePaddle FastDeploy 官网开源网址: <https://github.com/PaddlePaddle/FastDeploy>, 所有更新以官方公布为准。

### Jetson 部署库编译

FastDeploy 当前在 Jetson 仅支持 ONNX Runtime CPU 和 TensorRT GPU/Paddle Inference 三种后端推理。

### C++ SDK 编译安装

编译需满足

gcc/g++ >= 5.4(推荐 8.2)

cmake >= 3.10.0

jetpack >= 4.6.1

如果需要集成 Paddle Inference 后端，在 Paddle Inference 预编译库页面根据开发环境选择对应的 Jetpack C++包下载，并解压。

```

git clone https://github.com/PaddlePaddle/FastDeploy.git
cd FastDeploy
mkdir build && cd build
cmake .. -DBUILD_ON_JETSON=ON \
-DENABLE_VISION=ON \
-DENABLE_PADDLE_BACKEND=ON \ # 可选项，如若不需要 Paddle Inference 后端，可关闭
-DPADDLEINFERENC DIRECTORY=/Download/paddle_inference_jetson \
-DCMAKE_INSTALL_PREFIX=${PWD}/installed_fastdeploy
    
```

```
make -j8
make install
编译完成后，即在 CMAKE_INSTALL_PREFIX 指定的目录下生成 C++ 推理库。
```

### C++ 推理

```
#include "fastdeploy/runtime.h"
namespace fd = fastdeploy;

int main(int argc, char* argv[]) {
    std::string model_file = "mobilenetv2/inference.pdmodel";
    std::string params_file = "mobilenetv2/inference.pdiparams";
    // setup option
    fd::RuntimeOption runtime_option;
    runtime_option.SetModelPath(model_file, params_file, fd::ModelFormat::PADDLE);
    runtime_option.UseOrtBackend();
    runtime_option.SetCpuThreadNum(12);
    // init runtime
    std::unique_ptr<fd::Runtime> runtime =
        std::unique_ptr<fd::Runtime>(new fd::Runtime());
    if (!runtime->Init(runtime_option)) {
        std::cerr << "--- Init FastDeploy Runitme Failed! "
              << "\n--- Model: " << model_file << std::endl;
        return -1;
    } else {
        std::cout << "--- Init FastDeploy Runitme Done! "
              << "\n--- Model: " << model_file << std::endl;
    }
    // init input tensor shape
    fd::TensorInfo info = runtime->GetInputInfo(0);
    info.shape = {1, 3, 224, 224};

    std::vector<fd::FDTensor> input_tensors(1);
    std::vector<fd::FDTensor> output_tensors(1);
    std::vector<float> inputs_data;
    inputs_data.resize(1 * 3 * 224 * 224);
    for (size_t i = 0; i < inputs_data.size(); ++i) {
        inputs_data[i] = std::rand() % 1000 / 1000.0f;
    }
    input_tensors[0].SetExternalData({1, 3, 224, 224}, fd::FDDataType::FP32, inputs_data.data());
    // get input name
    input_tensors[0].name = info.name;
    runtime->Infer(input_tensors, &output_tensors);
    output_tensors[0].PrintInfo();
    return 0;
}
```

## Python 编译安装

编译过程同样需要满足

gcc/g++ >= 5.4(推荐 8.2)

cmake >= 3.10.0

jetpack >= 4.6.1

python >= 3.6

Python 打包依赖 wheel，编译前请先执行 pip install wheel

如果需要集成 Paddle Inference 后端，在 Paddle Inference 预编译库页面根据开发环境选择对应的 Jetpack C++包下载，并解压。

所有编译选项通过环境变量导入

```
git clone https://github.com/PaddlePaddle/FastDeploy.git
```

```
cd FastDeploy/python
```

```
export BUILD_ON_JETSON=ON
```

```
export ENABLE_VISION=ON
```

```
# ENABLE_PADDLE_BACKEND & PADDLEINFERENC DIRECTORY 为可选项
```

```
export ENABLE_PADDLE_BACKEND=ON
```

```
export PADDLEINFERENC DIRECTORY=/Download/paddle_inference_jetson
```

```
python setup.py build
```

```
python setup.py bdist_wheel
```

编译完成即会在 FastDeploy/python/dist 目录下生成编译后的 wheel 包，直接 pip install 即可。

编译过程中，如若修改编译参数，为避免带来缓存影响，可删除 FastDeploy/python 目录下的 build 和 setuptools-cmake-build 两个子目录后再重新编译。

## Python 推理

```
import fastdeploy as fd
model_url = "https://bj.bcebos.com/fastdeploy/models/mobilenetv2.tgz"
fd.download_and_decompress(model_url, path=".")

option = fd.RuntimeOption()
option.set_model_path("mobilenetv2/inference.pdmodel",
                      "mobilenetv2/inference.pdiparams")

# **** CPU 配置 ****
option.use_cpu()
option.use_ort_backend()
option.set_cpu_thread_num(12)

# 初始化构造 runtime
runtime = fd.Runtime(option)

# 获取模型输入名
```

```
input_name = runtime.get_input_info(0).name

# 构造随机数据进行推理
results = runtime.infer({
    input_name: np.random.rand(1, 3, 224, 224).astype("float32")
})
print(results[0].shape)
```

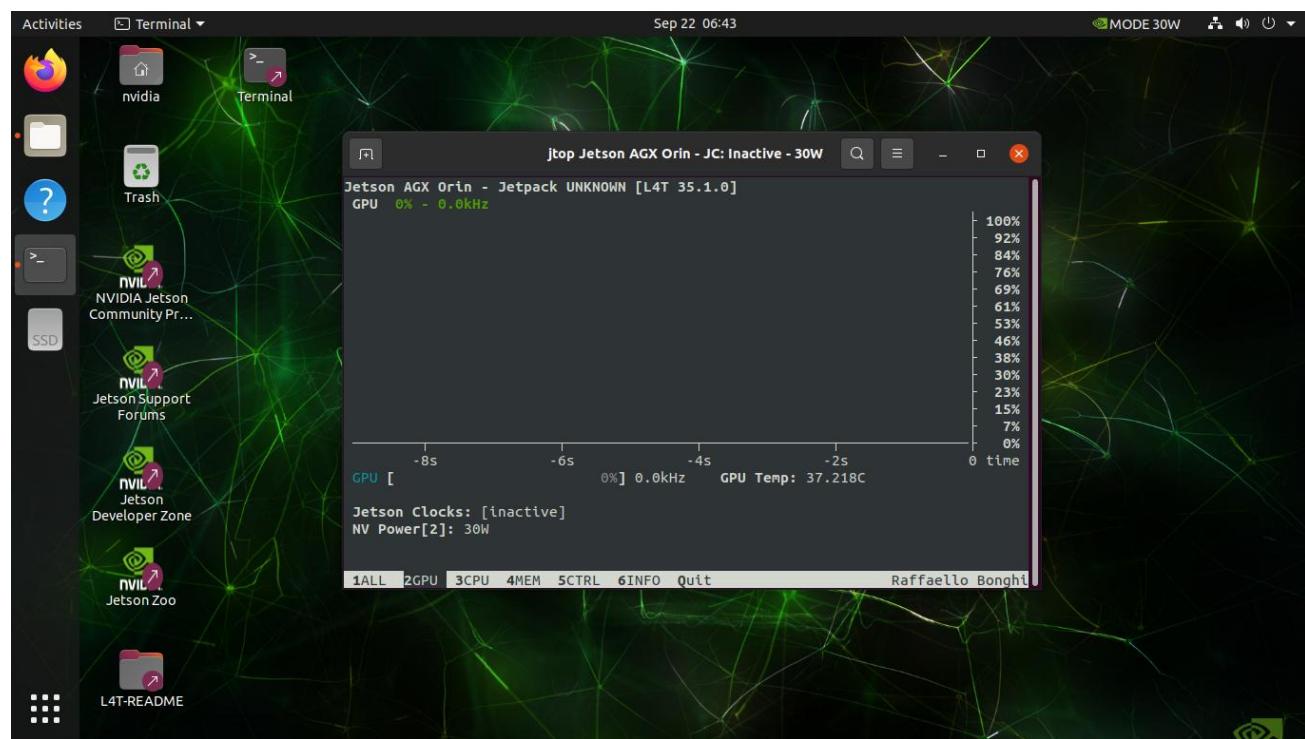
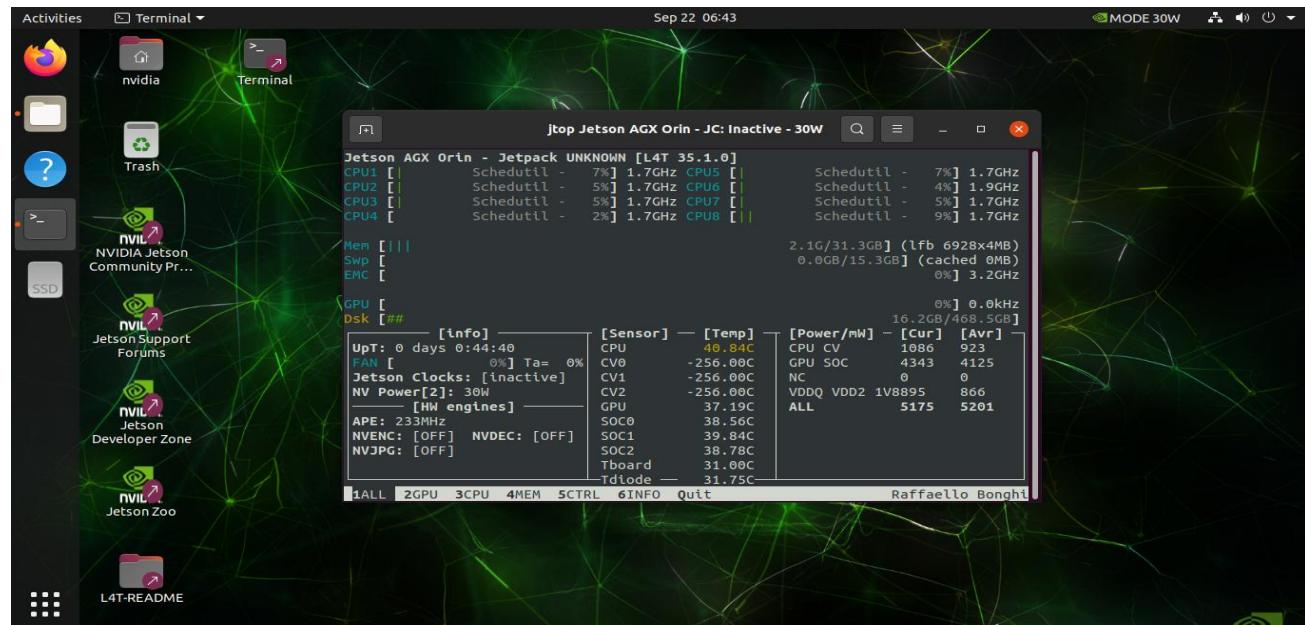
## Jtop 安装

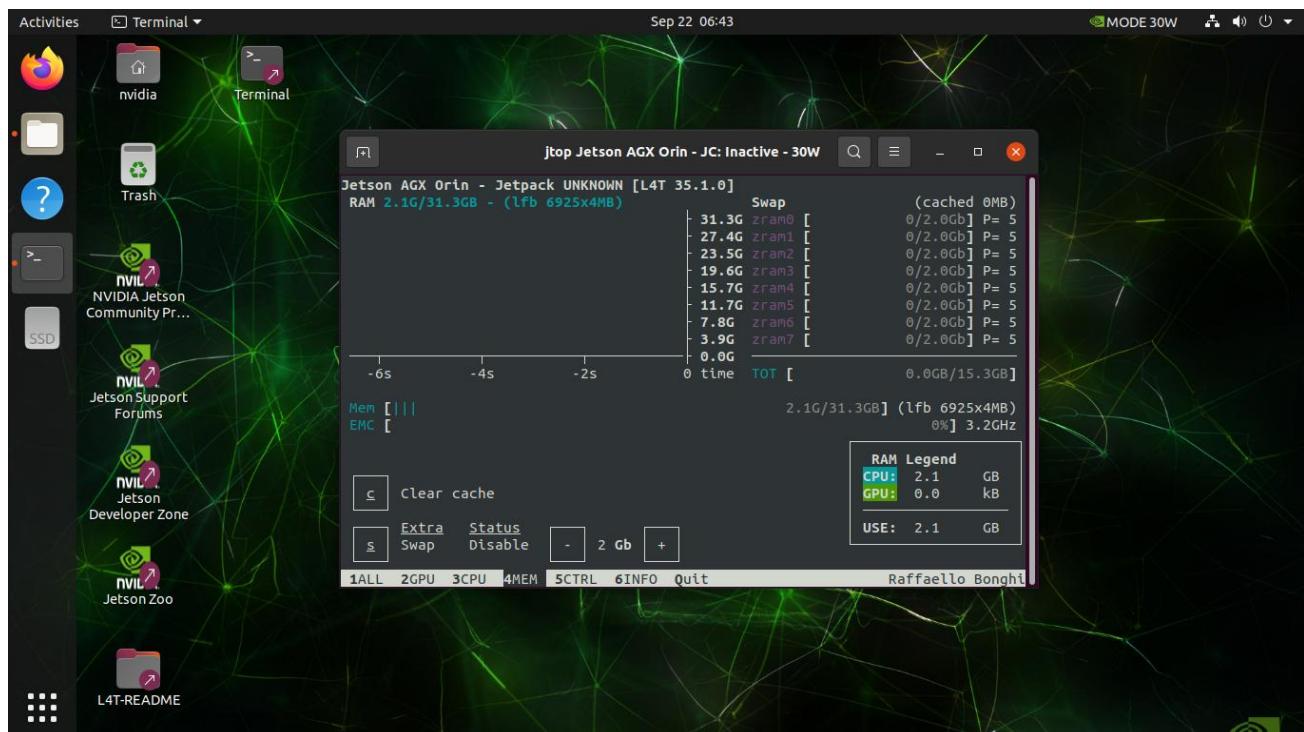
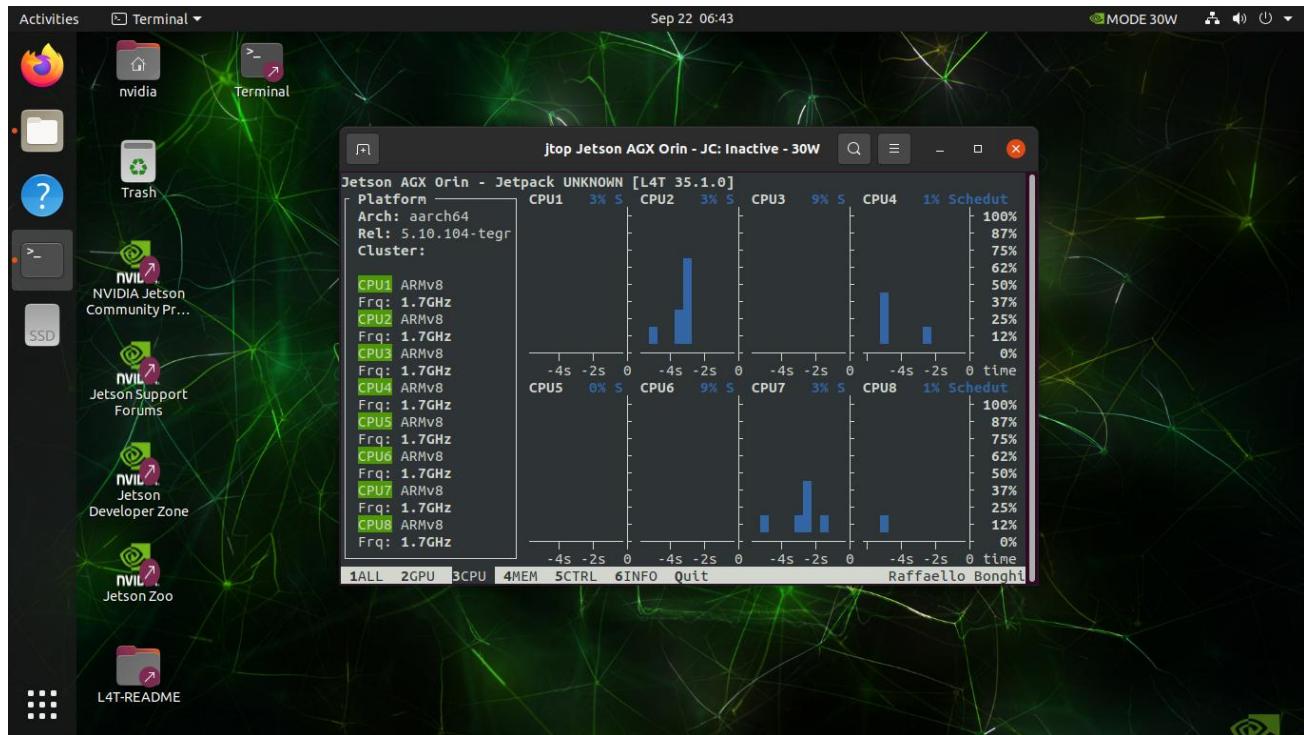
jtop（一个系统监视实用程序，可在终端上运行，并实时查看和控制 nvidia jetson 的状态），安装也非常方便，如果 jetson 产品上已安装了 jetpack sdk，可以按照如下步骤安装运行：

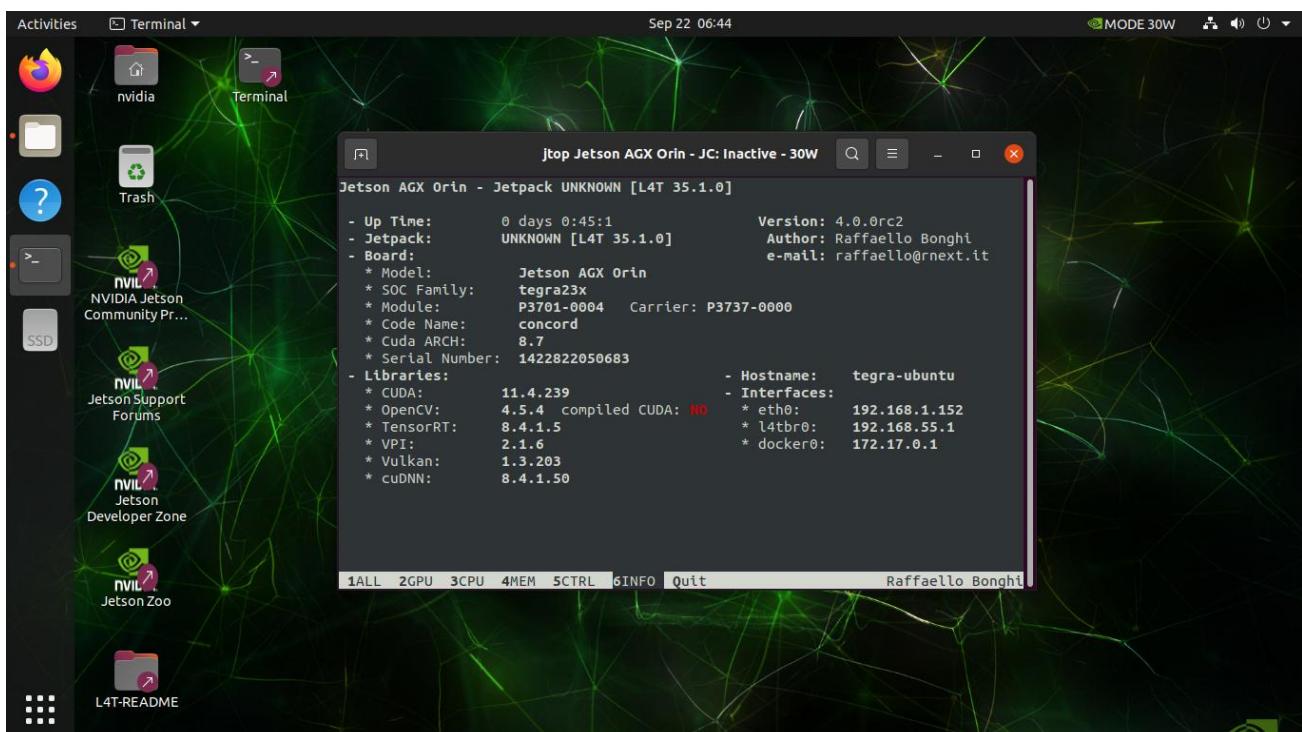
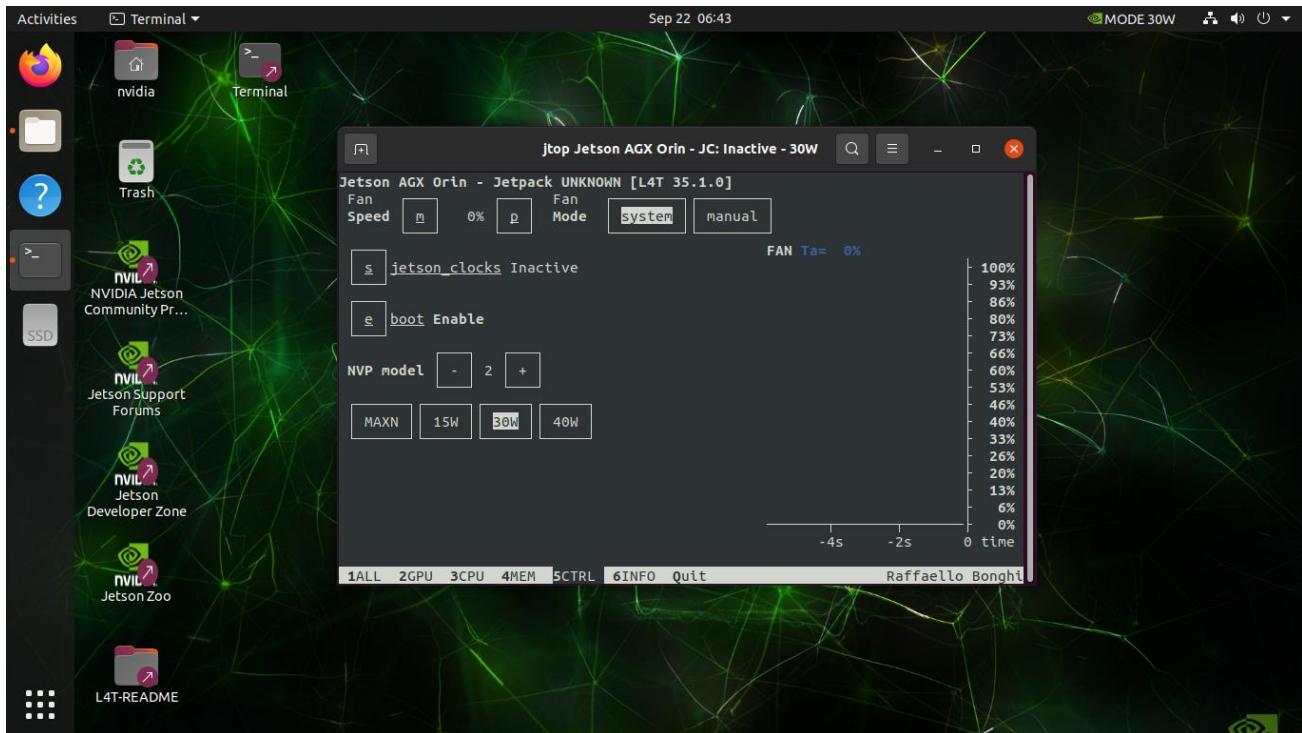
安装及运行

```
sudo apt install python3-pip
sudo -H pip3 install -U pip
sudo -H pip install jetson-stats==4.0.0rc2
```

开启后的界面如下，分别按数字键 1 2 3 4 5 6 进行页面切换，按 Q 退出







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：

1) all

包含模块运行信息包括：cpu、内存、gpu、磁盘、风扇、jetson\_clock 状态、nvpmode等

2) gpu

实时 gpu 状态

3) cpu

实时 cpu 状态

4) mem

内存状态

5) ctrl

可以控制的状态, 包括风扇转速以及功率模式的选择;

6) info

lib 库、cuda、serial number、interface 等信息, **自带 opencv 不支持 cuda, 若需要支持 cuda, 需要卸载自带 opencv 然后手动编译 opencv;**

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

2、tegrastats 命令行查看各资源信息

```
$ tegrastatsram 1778/7763mb (lfb 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%@mhz gr3d :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
```

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
```

## 图为信息科技（深圳）有限公司保修条例

### 重要提示

图为信息科技（深圳）有限公司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。

图为信息科技（深圳）有限公司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。图为科技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。除非图为科技另行通知，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，请与图为科技或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，您需要遵循图为科技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）。图为科技有权对所报修产品进行“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”零件将被返回图为科技。因部分维修产品需发往原厂，为避免意外损失，图为科技提请您购买运输保险，如果用户放弃保险，那么所寄物品在运输途中损坏或遗失，图为科技不承担责任。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，图为科技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 颇坏。
- 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、坠落受陨、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。