TW-T609用户手册

TW-T609 MANUAL

目录

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	2021/10/29	全面重制, 去除错误信息, 完善资料
V3.0	2021/12/31	去除错误内容, 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

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。

注意:由于本机接口驱动是由我司研发,与 nvidia 开发板接口驱动不同,使用 upgrade 这个指令会升级内核,覆盖设备树,如果必须使用 upgrade 进行更新,在使用之前,请运行以下命令

```
sudo rm /etc/apt/sources.list.d/nvidia-l4t-apt-source.list
sudo apt-get update
sudo apt-get upgrade
```

- 电源电压

TW-609 边缘计算平台输入端电源稳定可靠，功率 60w。

电源范围： 13- 36 v

- 环境要求：

工作温度： -20℃ - 60℃

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，特俗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-609 为一款基于 NVIDIA® JETSON AGX XAVIER™系列模块面向无人驾驶车载系统的计算平台，内置集成 AGX XAVIER 模块，预装 UBUNTU 18.04 操作系统，具备 20/32TOPS 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、RS232/485、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，8 路 GMSL2 摄像头输入。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-609 边缘计算平台概述

- 内嵌 nvidia® jetson agx xavier™
- 支持 M.2 KEY M (PCIEX4 NVME 2280)
- 支持 M.2 KEY E (PCIEX1 2230)
- 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/4G 模组
- 日本 JAE 车规级 IO 插头（1xPOWER，2xCAN，4xGPIO）
- 无风扇被动散热设计
- 内置 ubuntu 18.04 系统和 JETPACK SDKS

产品规格

处理器及核心模块

Processor	**Nvidia jetson agx xavier**
CPU	8-core carmel arm v8.2 64-bit cpu, 8mb l2 + 4mb l3
GPU	512-core volta gpu with 64 tensor cores 11 tflops (fp16) | 22 tops (int8)
Memory	32gb 256-bit lpddr4
Dl accelerator	2×nv dla engines
Storage	1x 32gb emmc (xavier 模块内置) 2x m.2 key m nvme 2280(选配安装)

接口

	Interface	**Quantity**	**Note**
Network	4G	1	多频段/NANO SIM 卡
	ETHERNET	2×RJ45 GIGABIT NETWORK PORT	ALTERNATIVE RJ45 AND WATERPROOF PORT(1000BASE-T)
	WIFI	1	2.4G/5.8G 300MBPS
Video output	HDMI	1×HDMI 2.0 TYPE A	4Kp60
USB	USB	4×USB 3.0TYPE A	USB 5V, 1A

I/O	TYPE-C	1xOTG	USB2.0	
	GPIO	4xGPIO(独立输入输出)	3.3VTTL 电平	日本 JAE 连接器 MX23A12SF1
	CAN	2xCAN2.0B	WITH CAN CHIP	
	UART	2x RS-232/422/485 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	DB9 TERMINAL	
	M.2	2×M.2 M KEY 1X M.2 KEY E (2230)	2xPCIE NVME 2280 SSD 1xUSB 2230 WIFI	
Function key	Power key	1	Button	
	Reset key	1	Button	
	Recovery key	1	Button	
Camera interface	Gmsl 2	8x gsml 2	Input	

编解码

Video Encode	4x 4K60 (H.265) 8x 4K30 (H.265) 16x 1080p60 (H.265) 32x 1080p30 (H.265) 30x1080p30 (H.264)

Video Decode	2x 8K30 (H.265) 6x 4K60 (H.265) 12x 4K30 (H.265) 26x 1080p60 (H.265) 52x 1080p30 (H.265) 30x 1080p30 (H.264)

供电

Power supply	**Spec**
Input type	日本 JAE 连接器 MX23A12SF1
Input voltage	Wide input 13-36v
Typical consumption	60w

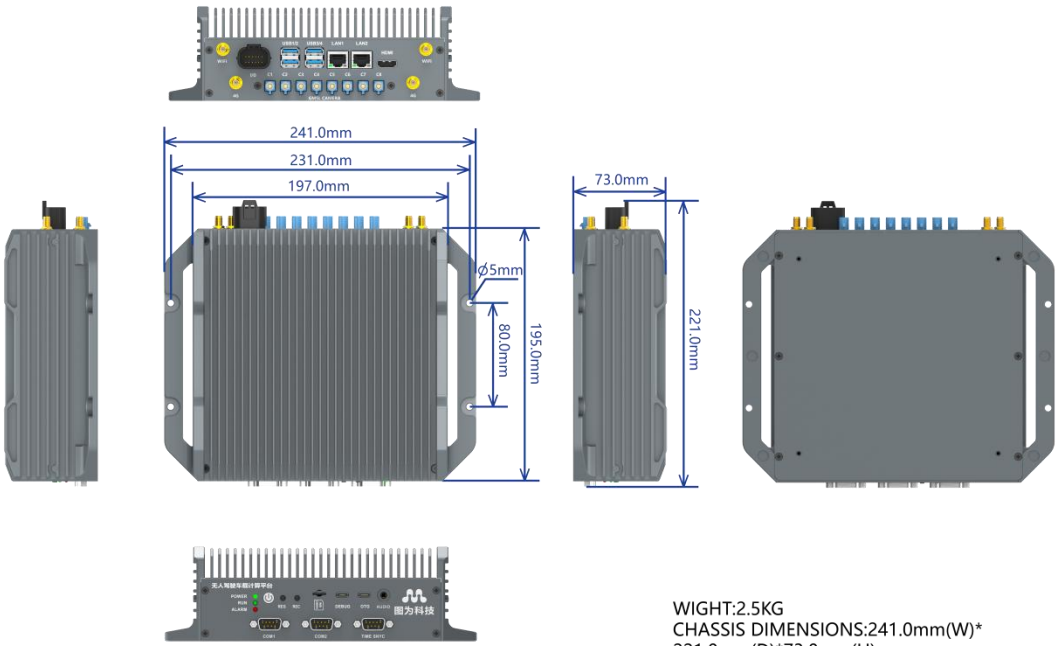
结构

Mechanical	**Spec**
Dimensions (w×h×d)	241(w)mm×195(d)mm ×73.0(h)mm
Weight	2.5kg

环境

Environmental	**Spec**
Operating temperature	-20℃-60℃
Storage humidity	10%-90% non-condensing

尺寸



WIGHT:2.5KG
CHASSIS DIMENSIONS:241.0mm(W)*
221.0mm(D)*73.0mm(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：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。。

接口说明

正面接口

Tw-609 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
ALARM_LED	故障状态指示灯	无作用
TIME SNYC	同步信号接口	输入输出同步信号 D-SUB9(对应设备号为 ttyTHS0)
COM1	RS232	标准 RS232 串口(对应设备号为 ttyTHS1)
COM2	RS232	标准 RS232 串口(对应设备号为 ttyTHS4)

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

T609设备采用Ubuntu18.04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-T609 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：

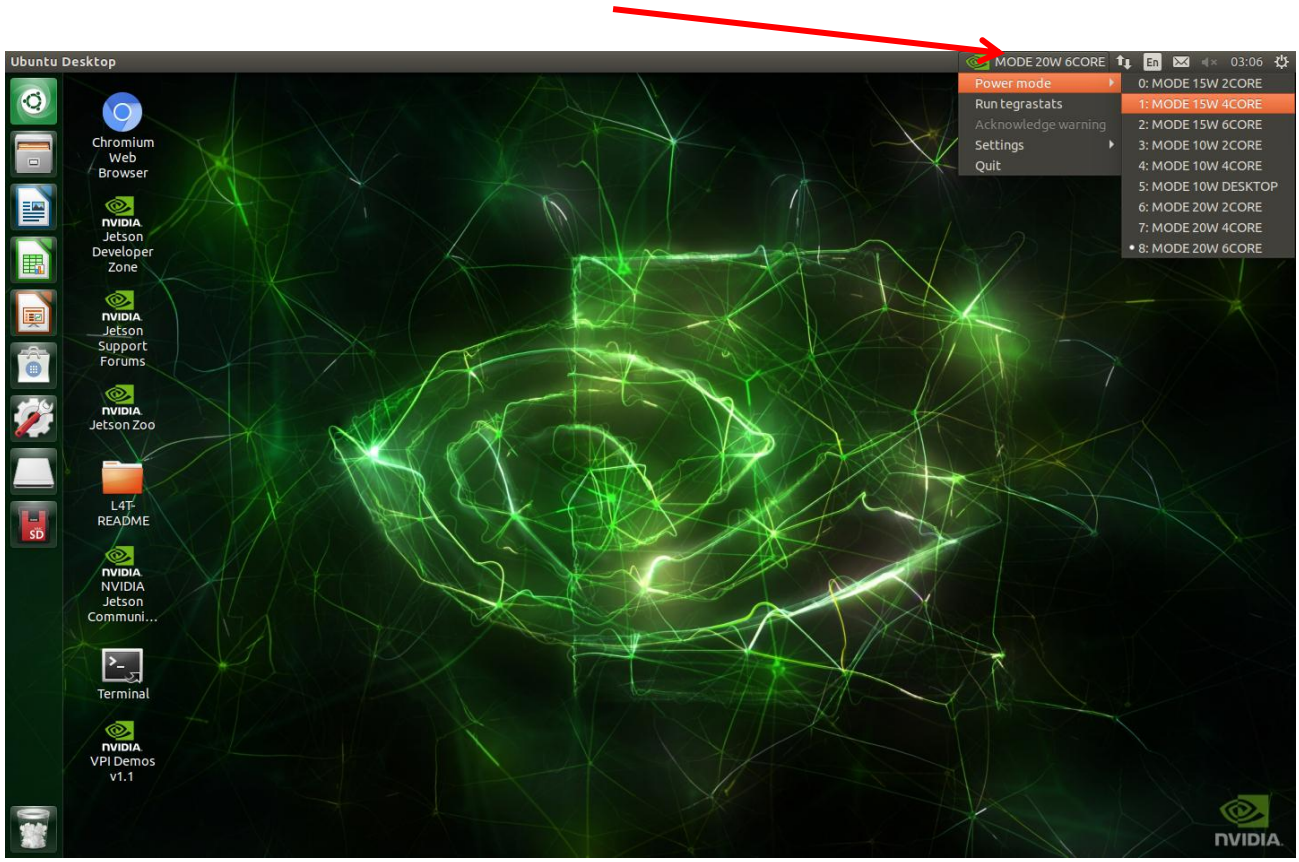
开机画面

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

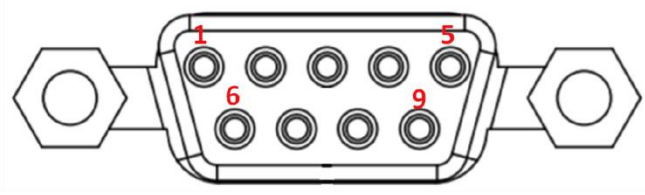
重启：在命行中执行$ sudo reboot，完成重启；也可以通过按键 reset 键重启（按一下即可）

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）



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		

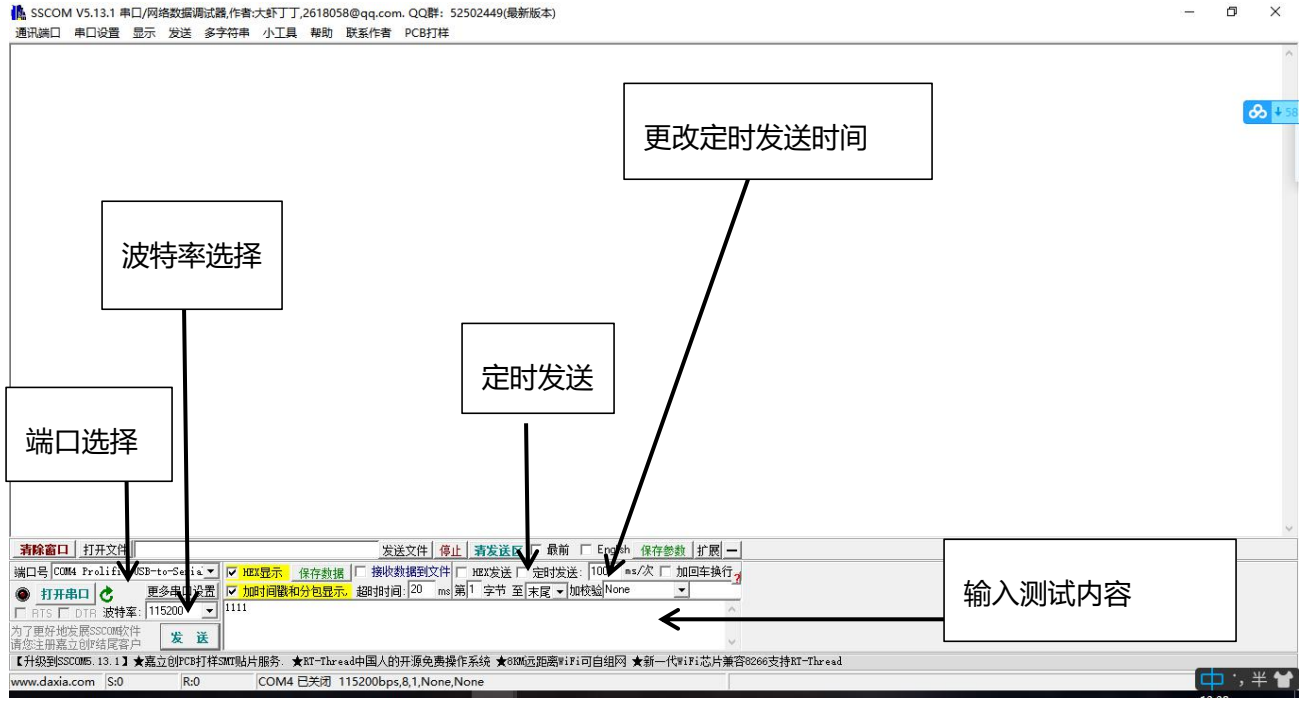
串口测试方法:

我司使用如图所示 usb 转串口线对 TW-609 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-609 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



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

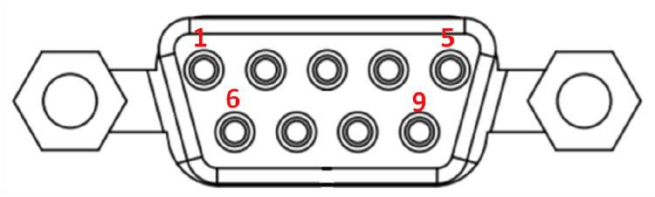
步骤二:TW-609 设备终端输入 sudo cutecom ,打开 cutecom(默认设置即可),串口号选择 ttyTHS1(COM1),打开串口,查看对话框是否有收到正确信息(除中文乱码外,数字出现乱码或传输数据不正确则表明不正常,传输数据一致则表明该串口通信正常)

步骤三:参考步骤二,测试 COM2,串口号选择 ttyTHS4.

测试串口工具下载链接:链接：https://pan.baidu.com/s/1rdwcrghl_28_x6120yjqxw

提取码：pf3d

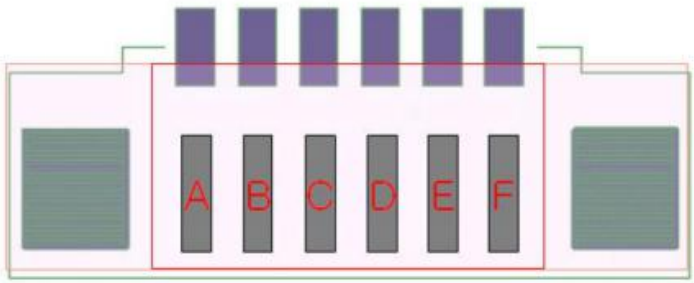
TIME SYNC 接口说明



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	PPS 信号脚
7	RTS	8	CTS
9	RI		

可通过外接 GPS 模块获取位置信息,具体接线为需看 GPS 模块的接线顺序,下图为我司选用 HX-28-GNF(**此模块的串口电平为 TTL 电平,TIME SYNC 接口电平为 232 电平,下文中的测试是在我司已修改 TIME SYNC 接口电平为 TTL 电平的情况下进行的测试,购买 GPS 模块时请核实电平情况**)的接线示意图,按照该模块的线序,连接我们 T609 设备时,需要将 3 号引脚 TX 接 RX(模块 D),2 号引脚 RX 接 TX(模块 C),PPS(模块 A)接 6 号引脚,另外该模块需要另外接 5v 供电,我们是通过设备 usb 接口给模块供电.(具体做法为改装 usb 数据线,然后接入将 usb 数据线正极接 VCC(模块 B),负极连接 GND(模块 E),也可以用其他外接电源给该模块供电)

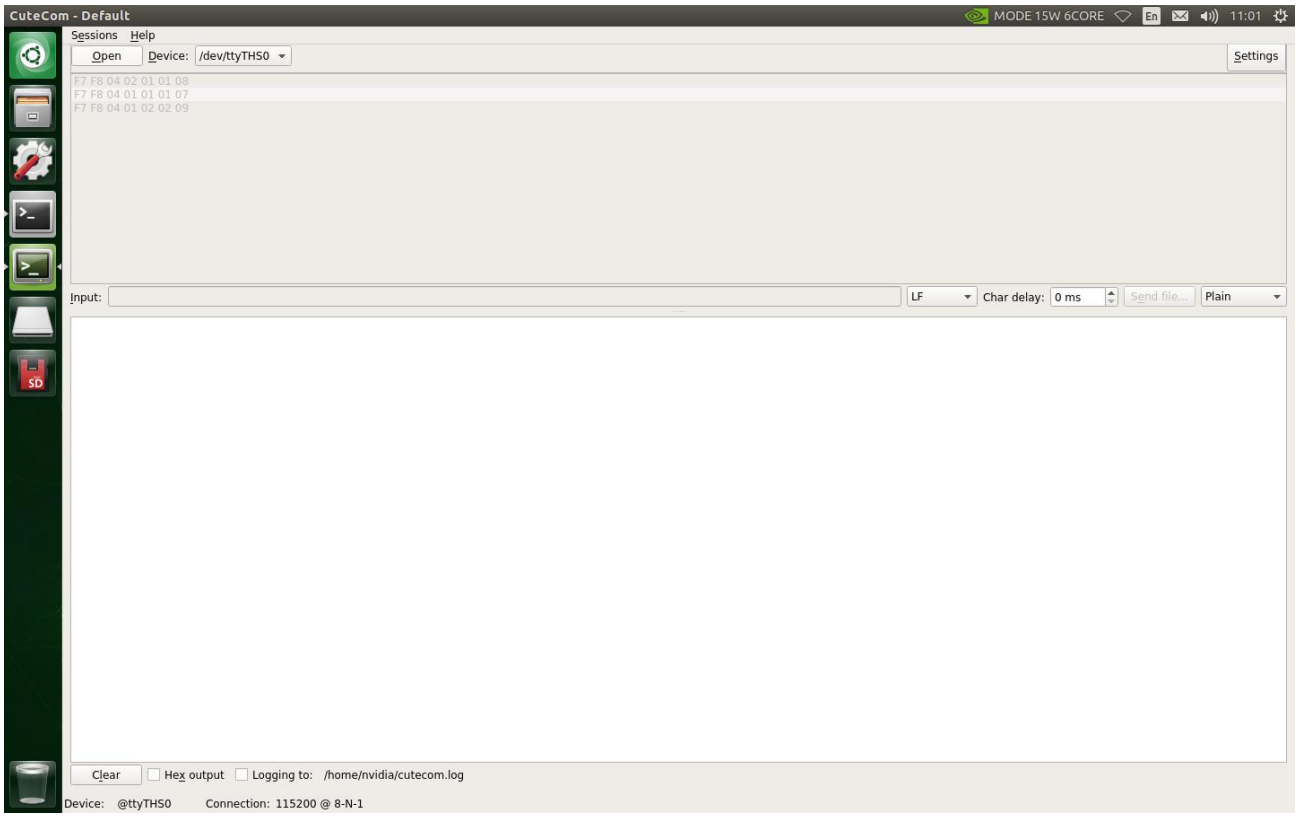


序号	名称	描述
A	PPS	时间标准脉冲输出
B	VCC	系统主电源,供电电压为+3.3V~+5V,工作时消耗电流约
C	TXDA	TTL界面,可选USB_DM及RS232_TXD
D	RXDA	TTL界面,可选USB_DP及RS232_RXD
E	GND	接地
F	VCC_NC	电源使能，高电平/悬空模块工作，低电平模块关闭

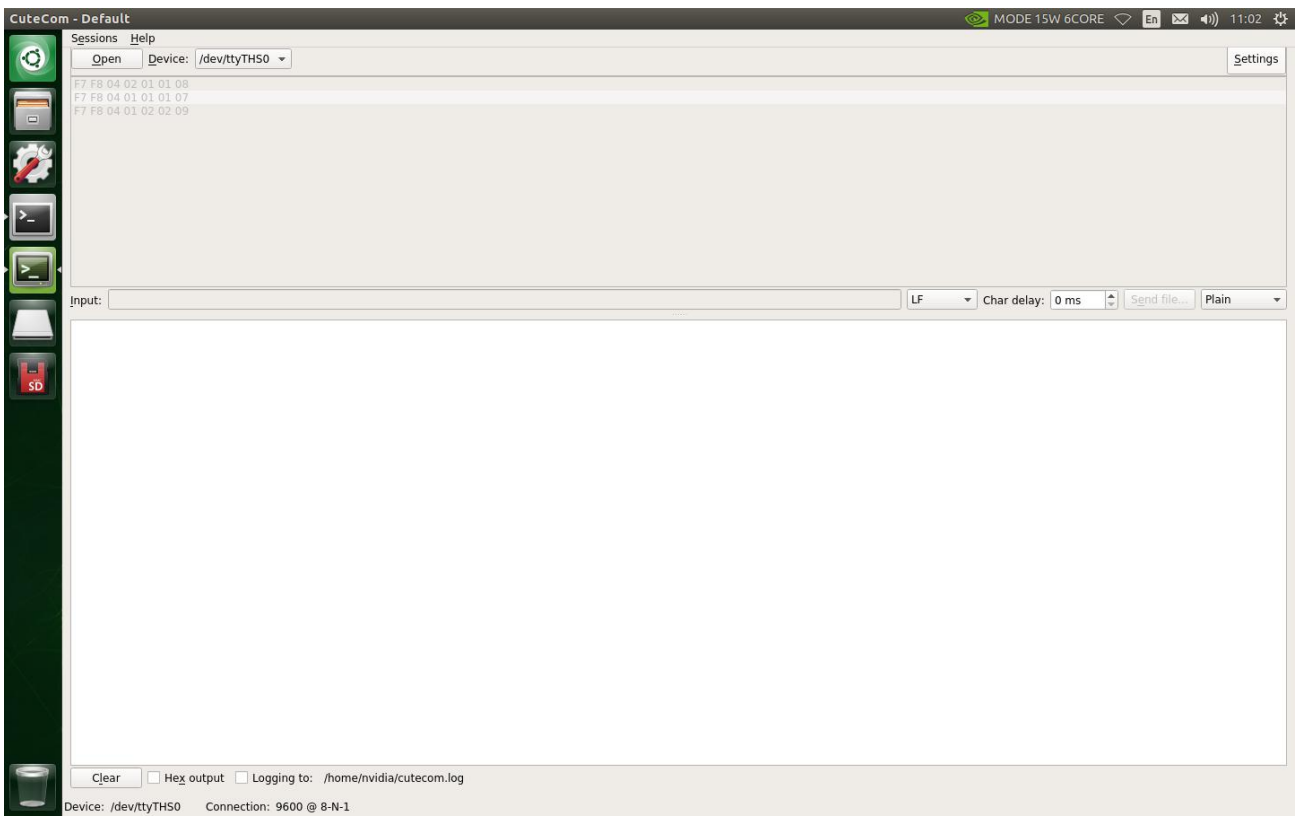
1.接线完毕后,将 GPS 模块天线放置到开阔位置,

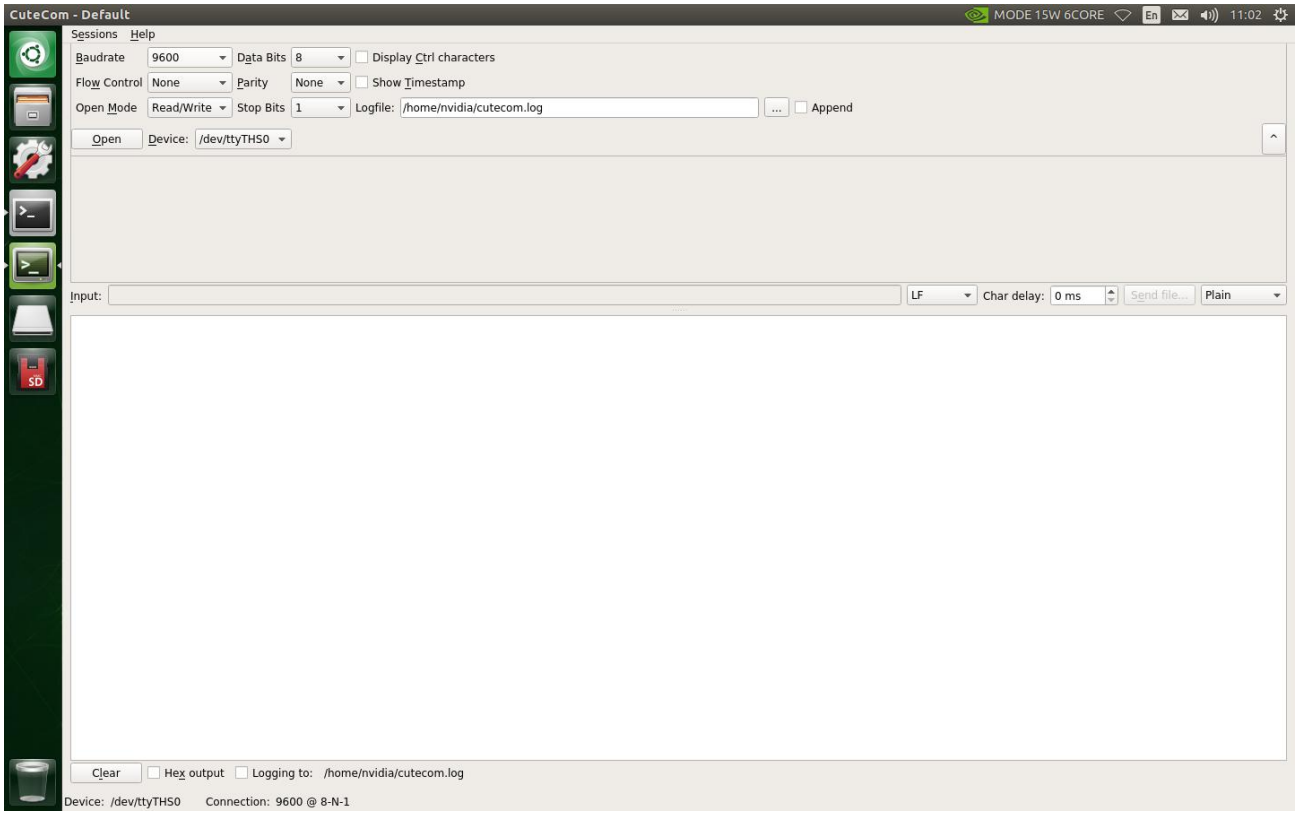
2.我们打开设备安装的串口调试工具 cutecom,

打开终端输入:sudo cutecom

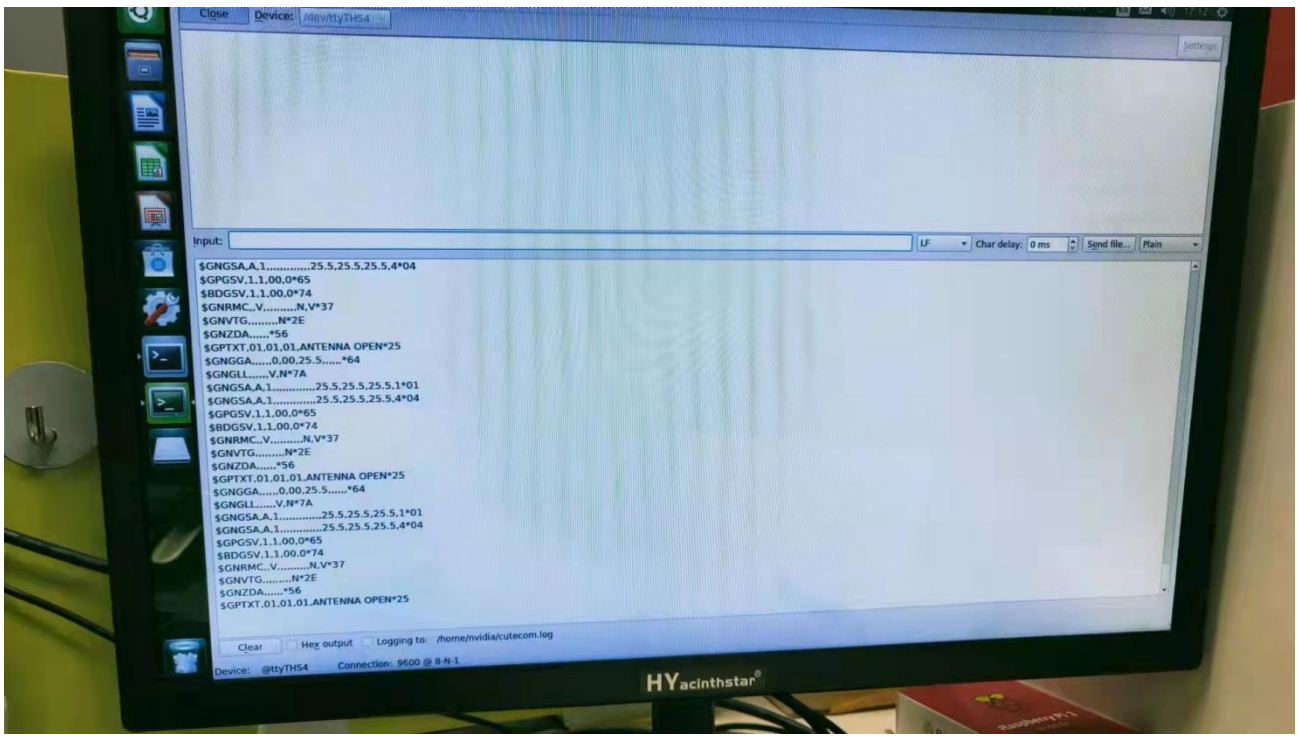


3. 串口设备号选择/dev/ttyTHS0,然后点击 settings 设置波特率为 9600,其他无需更改,然后打开串口,





4. 底下输出栏显示效果如图



出现此信息说明连接无误,GPS 模块与设备能正常通信.

5. 最后可通过运行如下命令，查看 PPS 信号信息

通过PPS工具查看PPS 信号信息：

sudo apt install pps-tools

sudo ppstest /dev/pps*

```
nvidia@Xavier:~$ sudo ppstest /dev/pps*
trying PPS source "/dev/pps0"
found PPS source "/dev/pps0"
ok, found 1 source(s), now start fetching data...
source 0 - assert 1603870031.622015376, sequence: 309 - clear  0.000000000, sequence: 0
source 0 - assert 1603870032.621971512, sequence: 310 - clear  0.000000000, sequence: 0
source 0 - assert 1603870033.621927520, sequence: 311 - clear  0.000000000, sequence: 0
```

sudo ppswatch -a /dev/pps0

```
nvidia@Xavier:~$ sudo ppswatch -a /dev/pps0
trying PPS source "/dev/pps0"
found PPS source "/dev/pps0"
timestamp: 1603869956, sequence: 233, offset: -374650317
timestamp: 1603869957, sequence: 234, offset: -374698885
timestamp: 1603869958, sequence: 235, offset: -374742653
timestamp: 1603869959, sequence: 236, offset: -374781173
timestamp: 1603869960, sequence: 237, offset: -374830957
```

背面接口

TW-609 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
4G	4G 天线接口	外接 4G 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
LAN(RJ45)X2	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
GMSL2	摄像头接口	8xGMSL2

摄像头支持类型

目前根据我司研发人员的不断努力,实现在 jetson-xavier-agx 设备上的 GMSL 摄像头的接入,目前我们这款 T609 支持以下四款摄像头,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,而是一款摄像头一种驱动,每个设备也只搭载一款驱动.

相机规格：

LI-IMX390-GMSL2

IMX390_RAW: RGB/1920x1080/30fps（制造商：猎豹 Leopard imaging）

LI-AR0231-GMSL2

AR0231_RAW: RGB/1928x1208/30fps （制造商：猎豹 Leopard imaging）

2MP 摄像模组

_GMSL2_AR0233

SG2-AR0233C-5200-GMSL2-Hxxx

AR0233/GW5200: UYVY/1920x1080/30fps（制造商：森云智能）

2MP 摄像模组

_GMSL2_IMX390

SG2-IMX390C-5200-GMSL2-Hxxx

IMX390/GW5200: UYVY/1920x1080/30fps（制造商：森云智能）

下文均已森云 IMX390 摄像头为例,猎豹摄像头使用另外方法,具体请联系我司.

首先确认设备是否读取到摄像头:

ls /dev/video*

预览执行如下命令:

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

& gst-launch-1.0 v4l2src device=/dev/video1 ! 'video/x-raw,format=UYVY,width=1920,height=1080' ! videoconvert ! fpsdisplaysink video-sink=xvimagesink sync=false (这是两个摄像头的命令,加一个摄像头就加一句这个命令,修改 id+1)

预览八个摄像头的画面:

同步测试:

• 输入 PWM 信号可动态修改频率，采用 96Pin_PWM2/GP_PWM1/GPIO.PN01

cd /sys/class/pwm/pwmchip0

echo 0 > export

cd pwm0

echo 40000000 > period

echo 4000000 > duty_cycle

echo 1 > enable #25Hz 10%

• IMX390/GW5200 仅支持 30fps 同步，不支持 30fps 以下帧率同步，比如若设置 PWM 为 25Hz，则显示黑屏但不会卡死，改为 30fps 后则正常显示

cd /sys/class/pwm/pwmchip0

echo 0 > export

cd pwm0

echo 33333333 > period

echo 3333333 > duty_cycle

echo 1 > enable #30Hz 10%

• AR0233/GW5200 支持 5~30fps 同步，即可以设置不同 PWM 频率改变同步输出的帧率 fps

cd /sys/class/pwm/pwmchip0

echo 0 > export

cd pwm0

设置 30fps

echo 33333333 > period

echo 3333333 > duty_cycle

echo 1 > enable #30Hz 10%

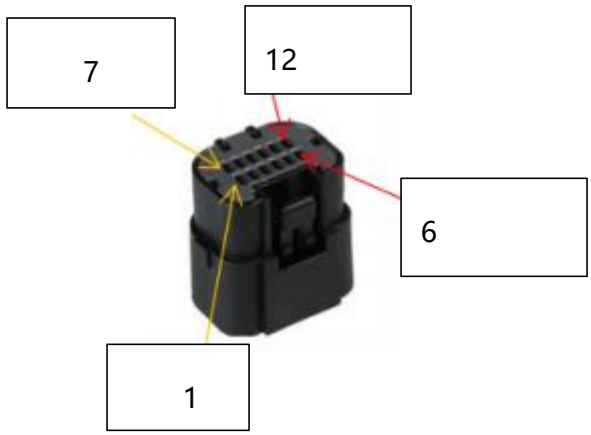
设置 25fps

echo 40000000 > period

echo 4000000 > duty_cycle

echo 1 > enable #25Hz 10%

多功能接口



6	5	4	3	2	1
VCC-IN 红色	GPIO4 橙色	GND 黑色	CAN-H1 蓝色	CAN-H0 蓝色	GPIO17 橙色
12	11	10	9	8	7
VCC-IN 红色	GPIO2 橙色	GND 黑色	CAN-L1	CAN-L0	GPIO10 橙色

接口定义:

PIN 定义			
PIN#	定义	PIN#	定义
1	GPIO17(Bidir 3.3V)_GPIO3_PQ.01(GPIO 417)	2	CAN_H0
3	CAN_H1	4	GND
5	GPIO04(OUT 3.3V)_GPIO3_PA.02 (GPIO 290)	6	VDDIN(12-24V)
7	GPIO10(Bidir 3.3V)_GPIO_PBB.02(GPIO 258)	8	CAN_L0
9	CAN_L1	10	GND
11	GPIO02(Bidir 3.3V)_GPIO_PG.07 (GPIO 343)	12	VDDIN(12-24V)

CAN 功能测试

内部原生 can 测试:**(由我司配置的电源线目前已默认将 2 路 can 焊接好,主要是测试两路 can 的相互收发情况)**

echo 'nvidia' | sudo -s sudo modprobe can

echo 'nvidia' | sudo -s sudo modprobe can_raw

echo 'nvidia' | sudo -s sudo modprobe mttcan

sudo ip link set can0 type can bitrate 80000

sudo ip link set up can0

sudo ip link set can1 type can bitrate 80000

sudo ip link set up can1

candump can0&

cangen -v can1

输完上述命令后,在终端会相继出现 CAN0,CAN1 的收发信息.

如需接入外部 can 设备测试,请将焊接线解开,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

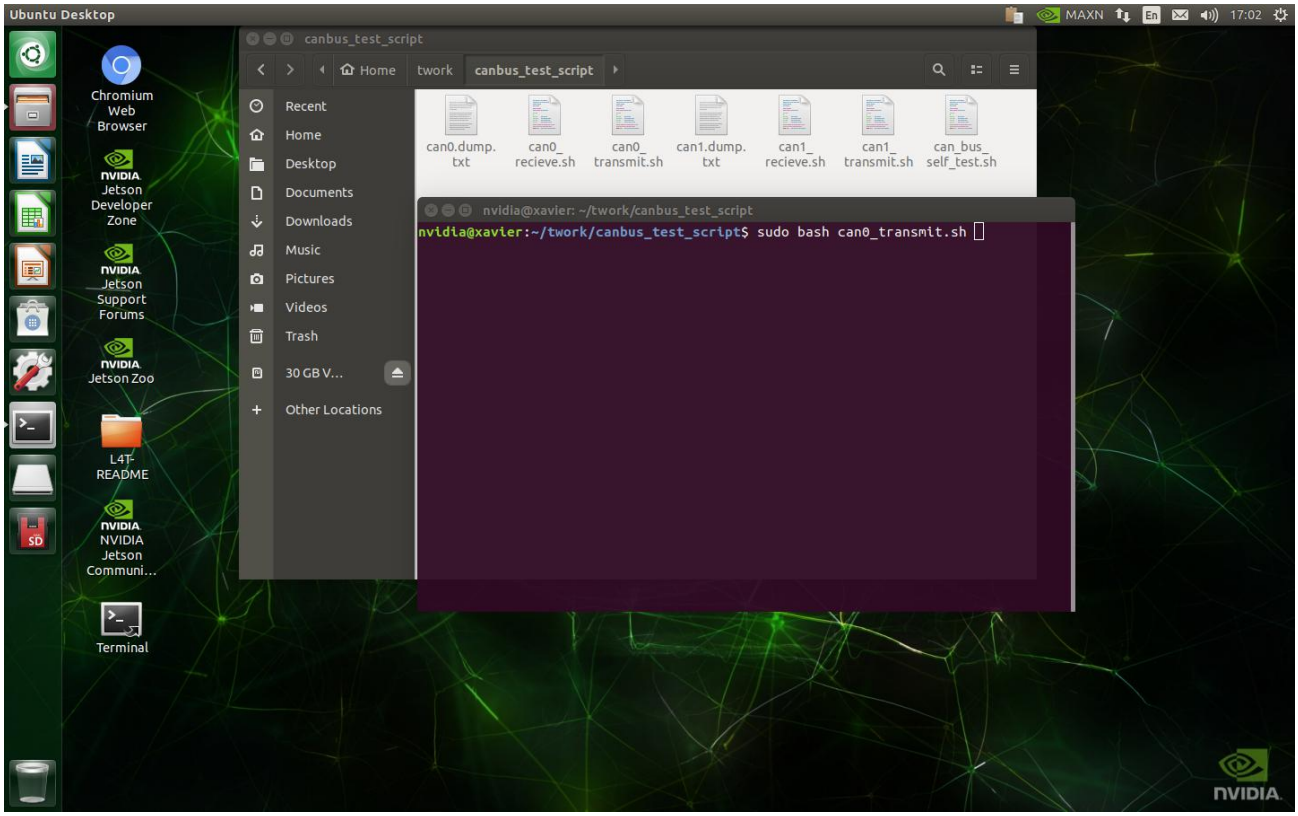
exit

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
```

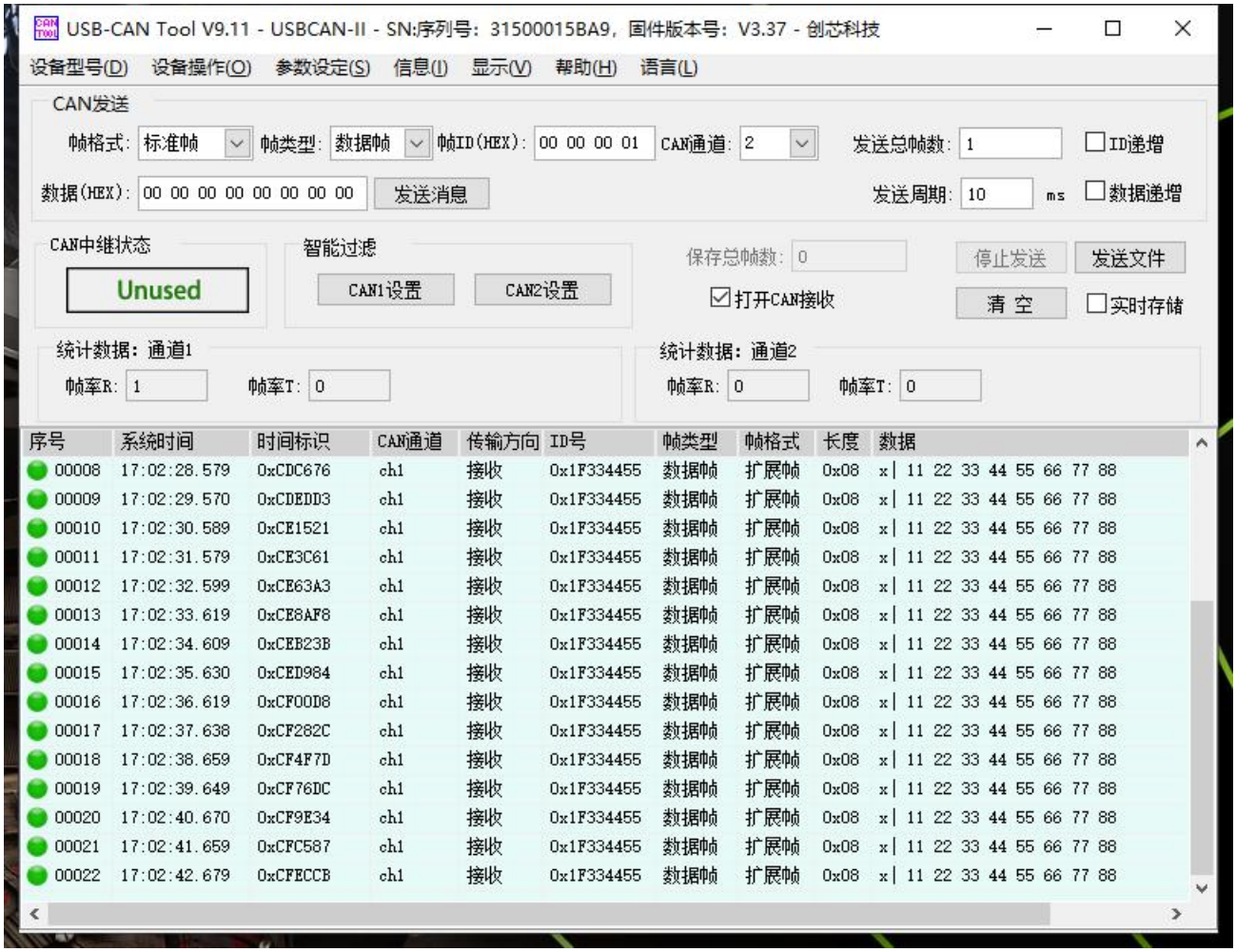
实际测试如下

终端运行发送脚本



另一端接入下图分析仪(此工具为我司测试工具,T609 整套设备不包含此工具),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



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. 编写以及运行下列 Python 参考示例代码

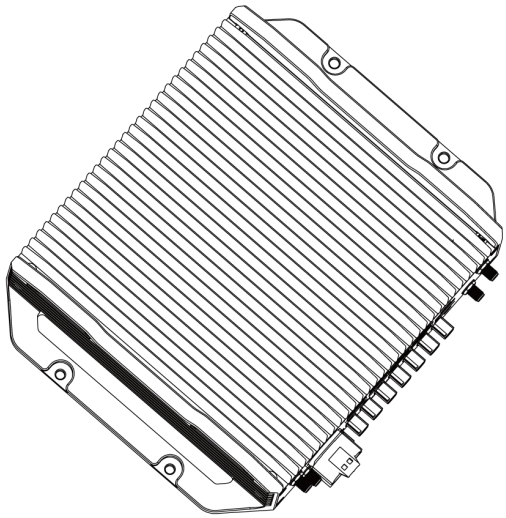
```
import time
import gpio

tst_gpio_pin = 417        #417 对应 T609 扩展的 GPIO17
#tst_gpio_pin = 290       #290 对应 T609 扩展的 GPIO04
#tst_gpio_pin = 258       #258 对应 T609 扩展的 GPIO10
#tst_gpio_pin = 343       #343 对应 T609 扩展的 GPIO02

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()
```

3. 通过如下查看设置 HI 和 LO

sudo cat /sys/kernel/debug/gpio| grep '417'

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。

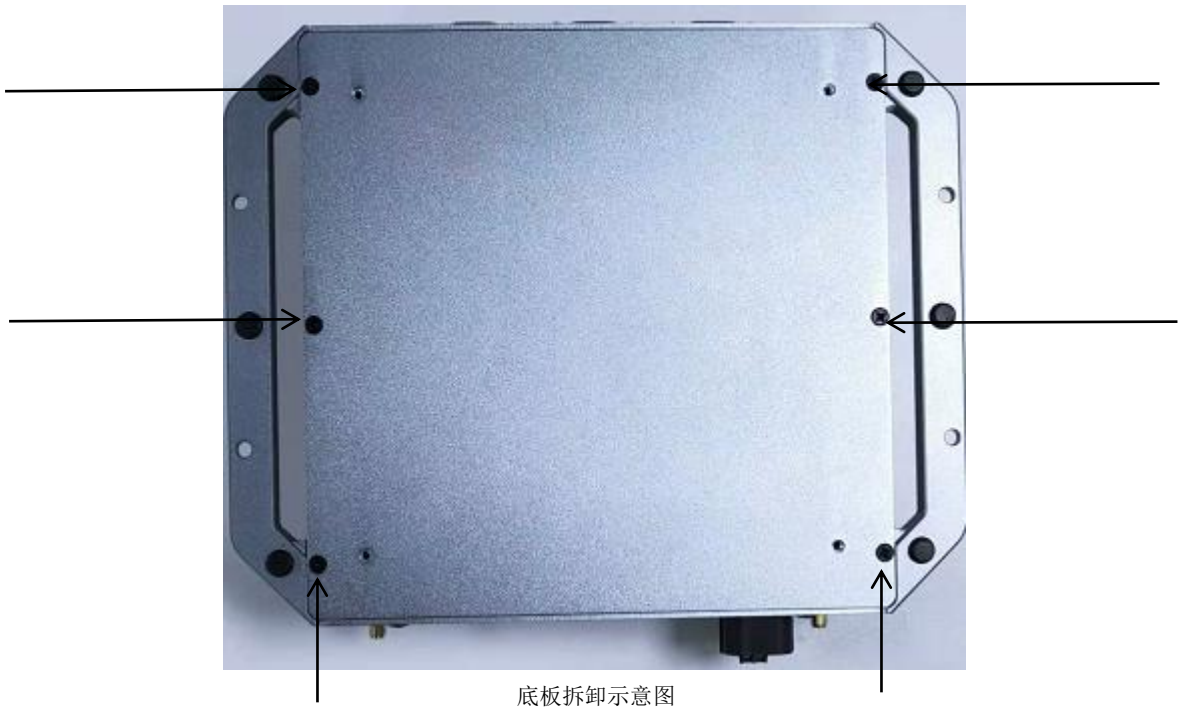
2.将盒子固定在您想要挂载盒子的地方。

Wifi/4g 天线安装

4G 模块安装

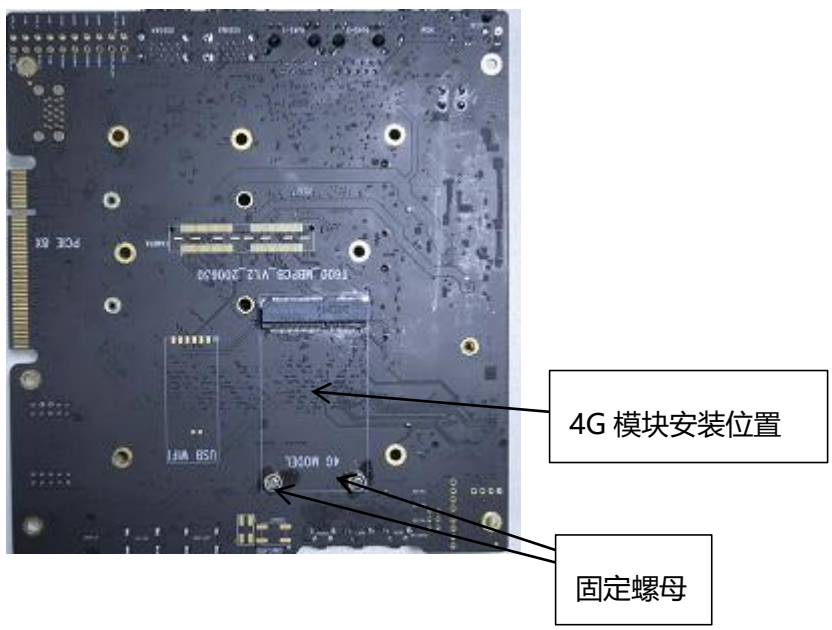
本产品标准件不搭载 4g 模块，该模块为选配安装，如需自己独立安装请参考本节内容.（建议拆卸前将原装螺母的位置拍照，进行相关记录，螺母螺柱使用盒子或者其他物品收集好避免丢失，过程需小心谨慎，切勿暴力拆卸，如因拆卸不当造成的损失，我司不承担相应责任）

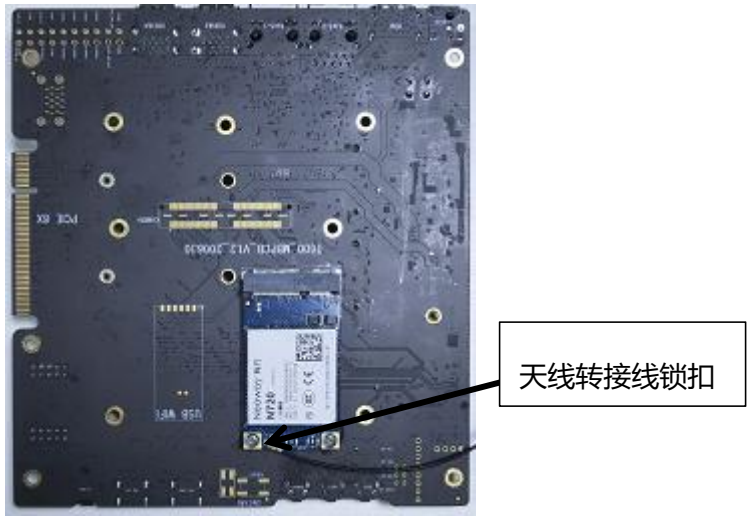
步骤一：如图箭头所示，准备好十字螺丝刀，依次拆下固定螺母，使用镊子或者螺丝刀将底层盖板取下；



底板拆卸示意图

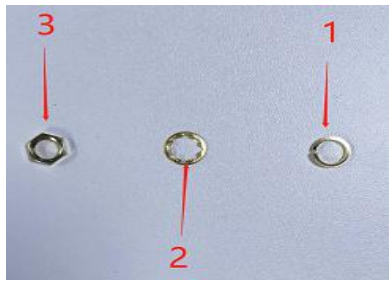
步骤二：取下 4g 模块底座的固定螺母，将 4g 模块装入如图所示的位置，找准模块上锁扣天线的位置，将天线转接线安装在模块上；





模块安装示意图

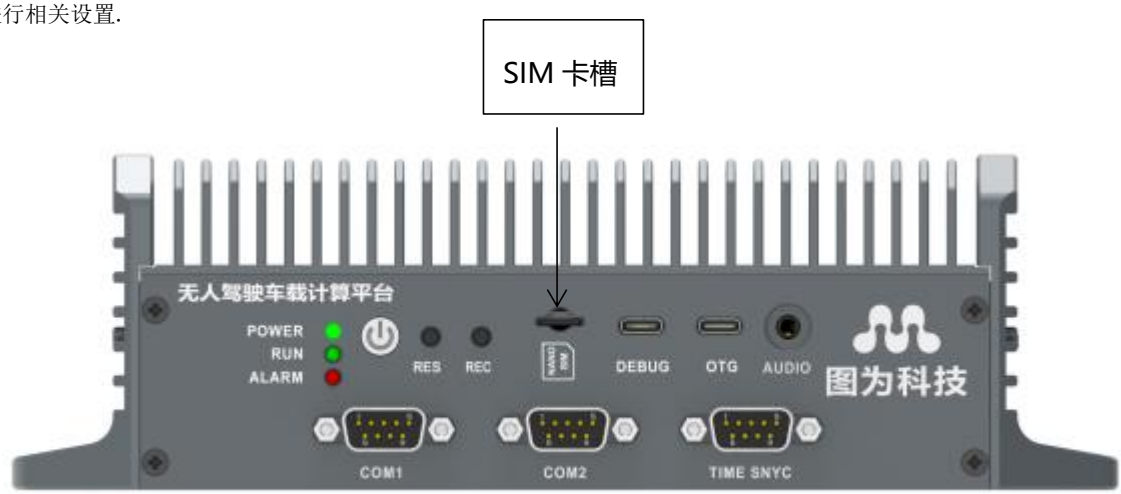
步骤三:如图所示依次将天线转接头固定好,外接天线棒;

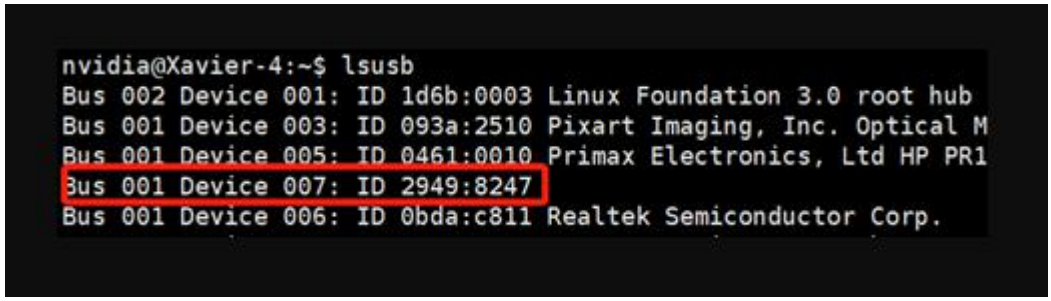


零件安装顺序图

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.



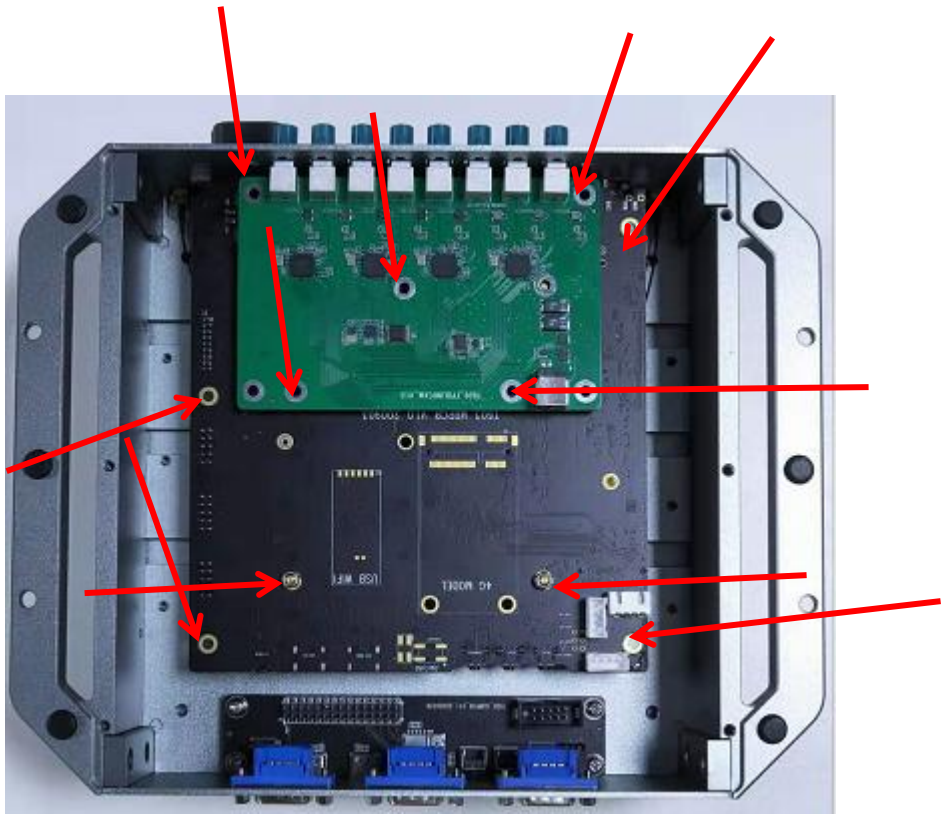

nvidia@Xavier-4:~$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 093a:2510 Pixart Imaging, Inc. Optical M
Bus 001 Device 005: ID 0461:0010 Primax Electronics, Ltd HP PR1
Bus 001 Device 007: ID 2949:8247
Bus 001 Device 006: ID 0bda:c811 Realtek Semiconductor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

Wifi 模块安装

本产品标准件不搭载 wifi 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）

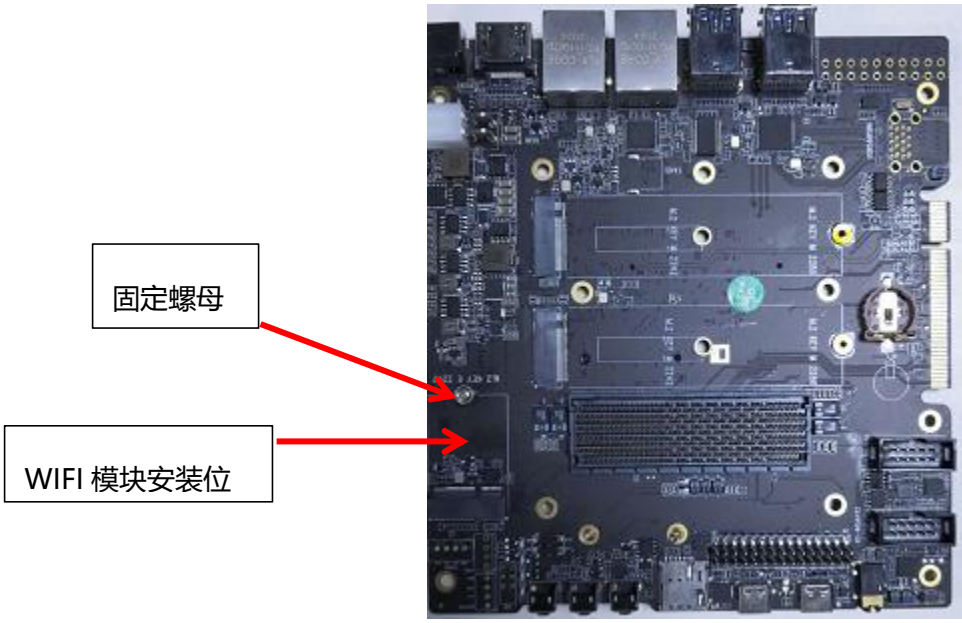
步骤一：参考 4g 模块安装步骤一；

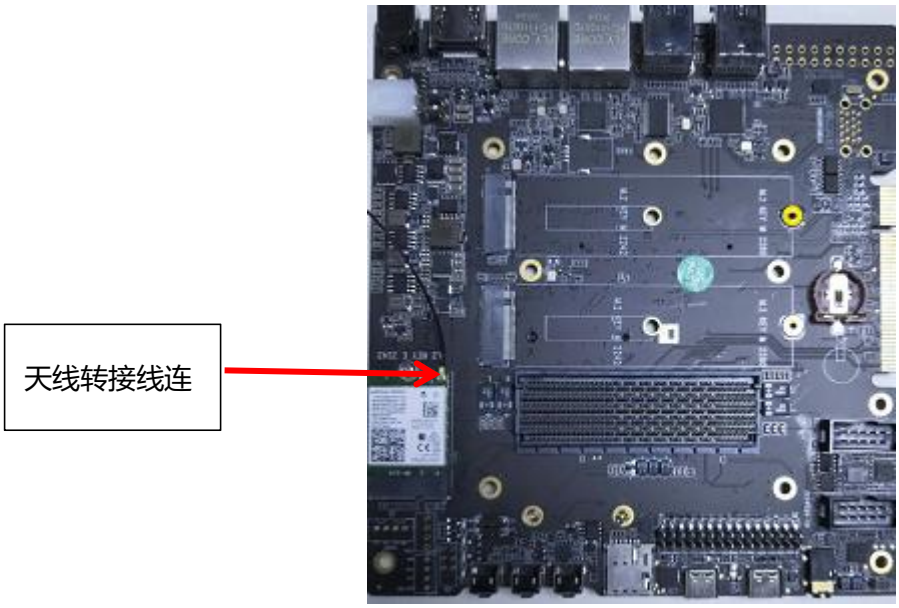
步骤二：依次取下固定载板的螺母，以及固定副板的螺母；



步骤三：先拆下副板，然后拧下在副板下面固定主板的两颗螺母，取下背面接口板，然后拆下主板

步骤四：找到如图所示的位置，取下固定螺母. 将 wifi 模块安装在底座上，锁好固定螺母，找到模块锁扣天线转接线的位置，锁扣好转接线；





步骤六:找到如图所示位置参考 4g 模块安装步骤三安装好转接头,连接天线棒;

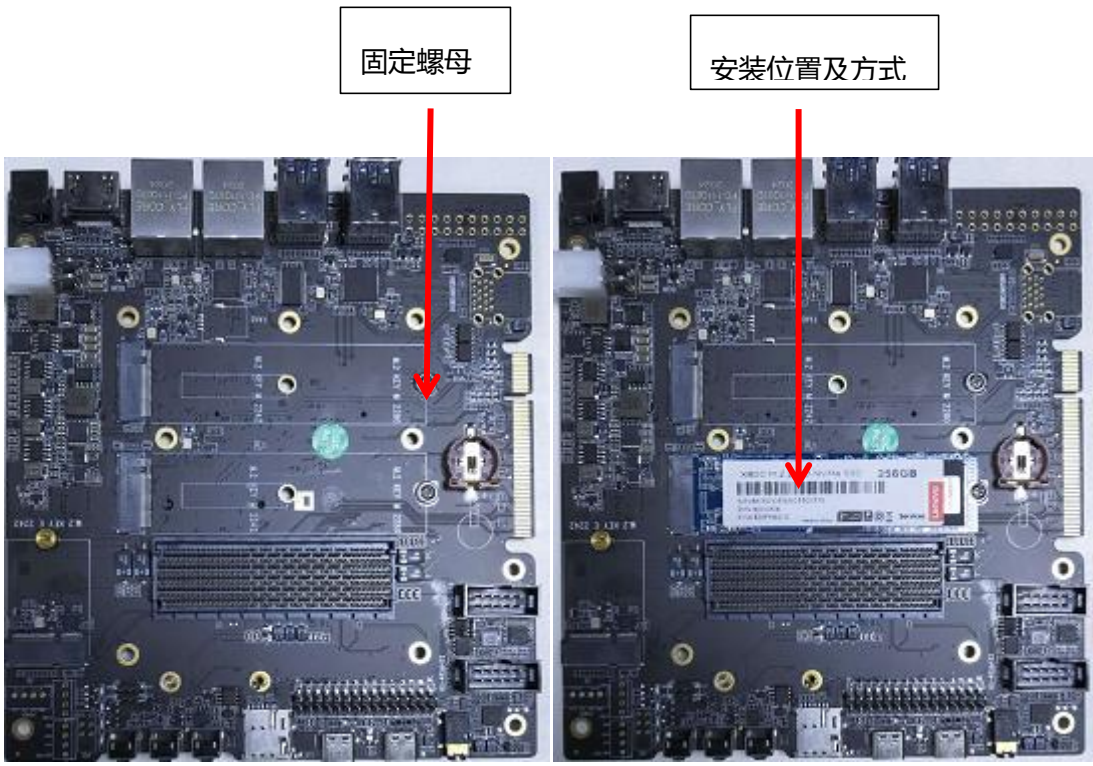
步骤七:将主板依次恢复原装方式(锁扣好核心板即可进行开机测试,避免因安装不当导致模块无法识别,需重新拆卸安装),安装完毕后设备连接电源开机测试,点击右上角网络选择,查看是否有搜索到 wifi 网络,测试通过后按照照片将设备恢复原状.

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)

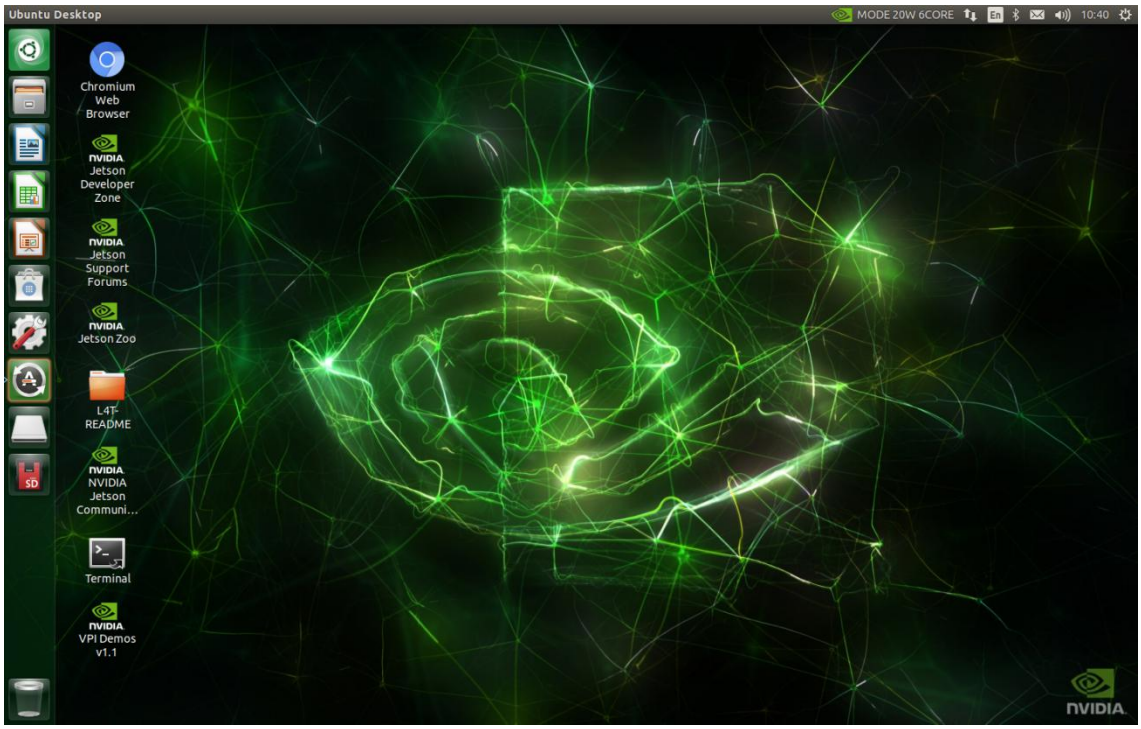
步骤一:参考 wifi 模块安装步骤一到步骤四,取下主板;

步骤二:找到如图所示的 ssd 安装位置(两个安装位置无区别),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.

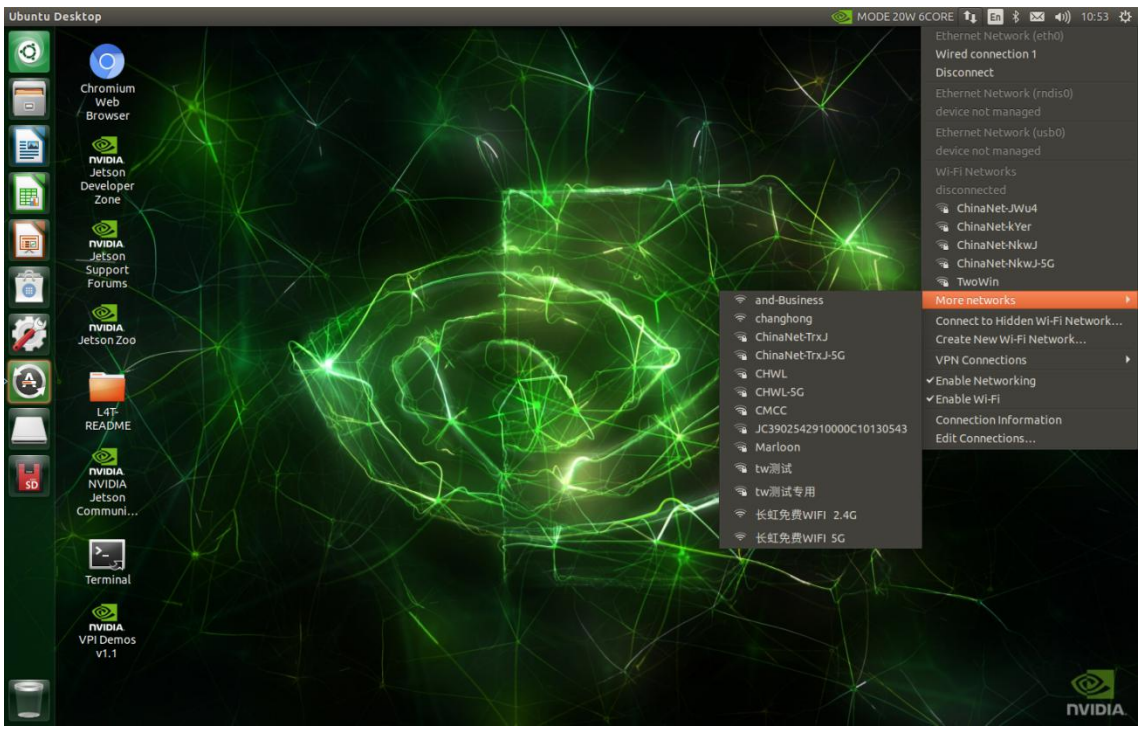


WiFi 连接

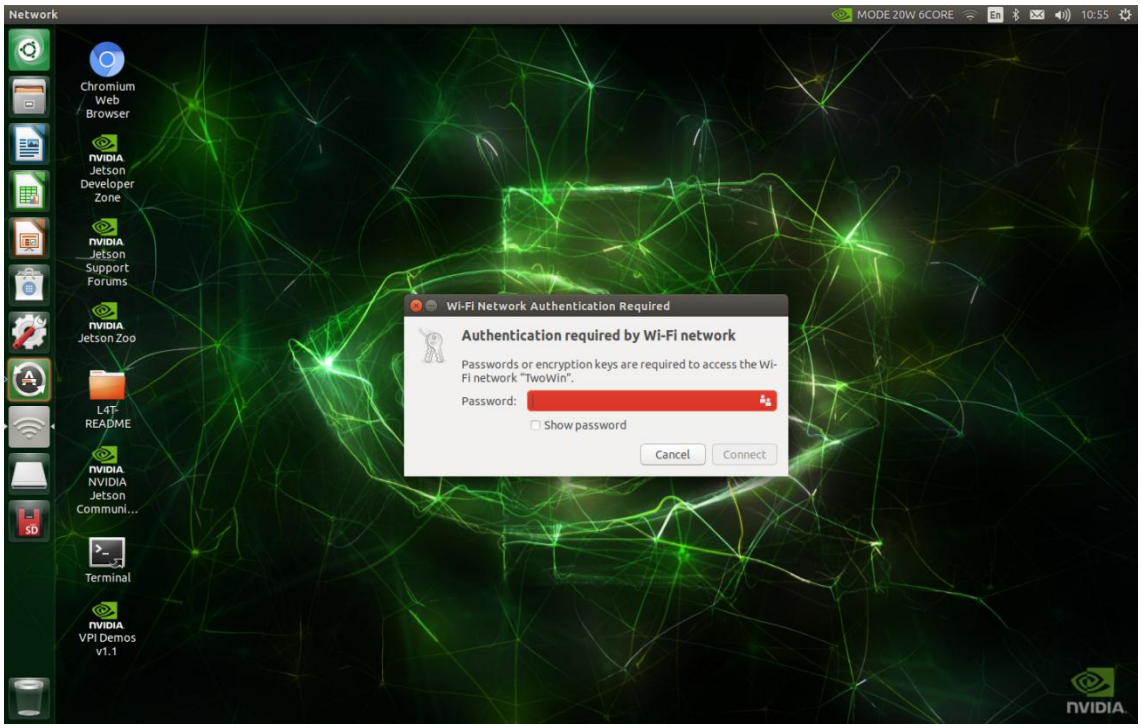
1. 点击箭头所示网络图标



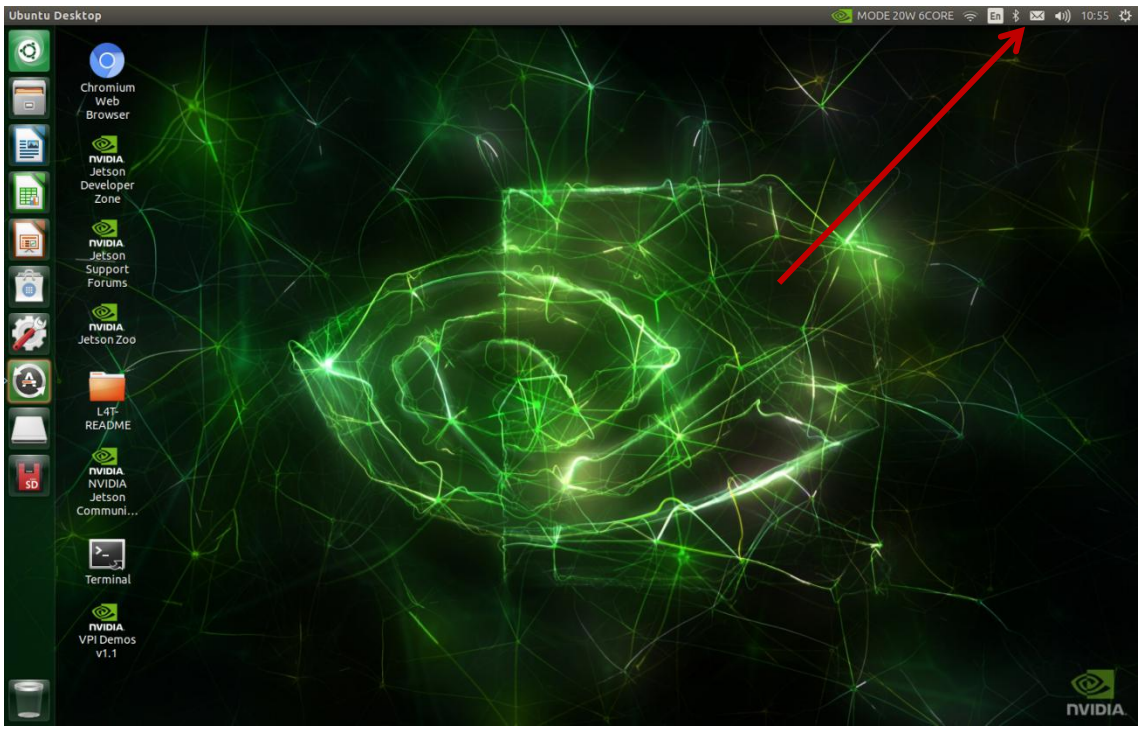
2. 搜寻可连接的 wifi 选项



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
MODE 20W 6CORE
10:56
百度一下，你就知道
baidu.com
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
Bai du 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

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/1iFOiKqKTGYuNTDZ5w_I9Mg

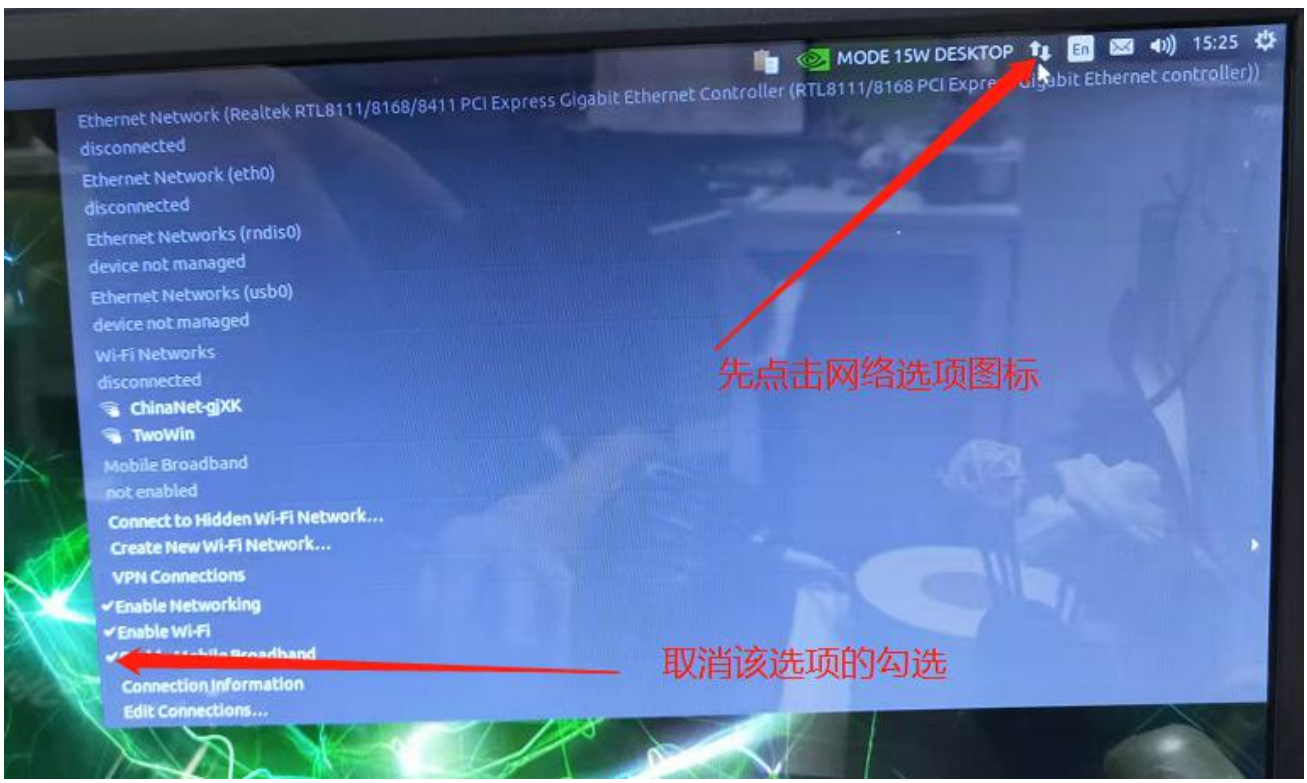
提取码：33sm

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

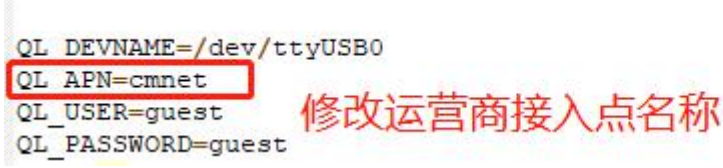
```
sudo touch /etc/rc.local
sudo chmod 755 /etc/rc.local
sudo gedit /etc/rc.local
```

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



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.

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

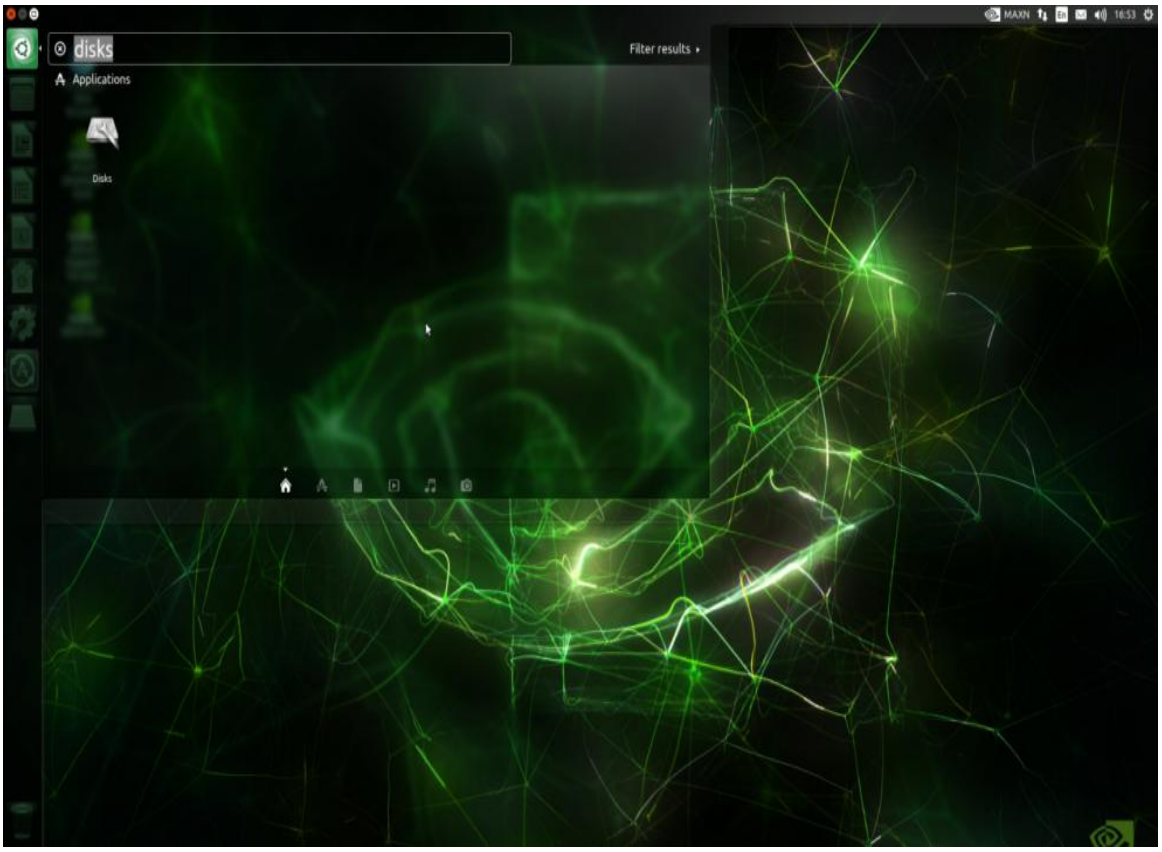
1.准备 M.2 Key M SSD

《本例采用的是铠侠 CL1-8D128 SSD》

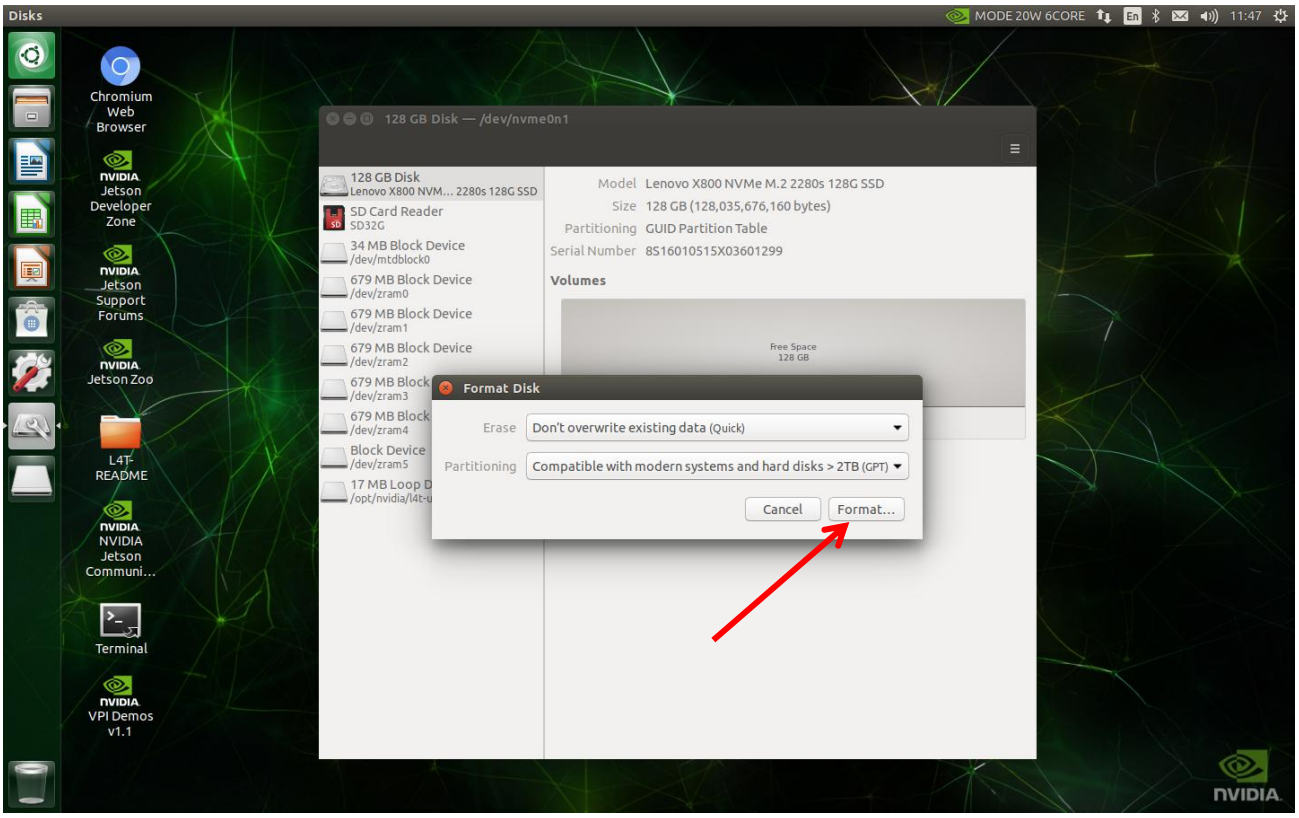
2.打开 Ubuntu18.04 自带 Disks 工具()，找到安装的 ssd 硬盘,首先按键”Ctrl+F”将其快速格式化为.

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):

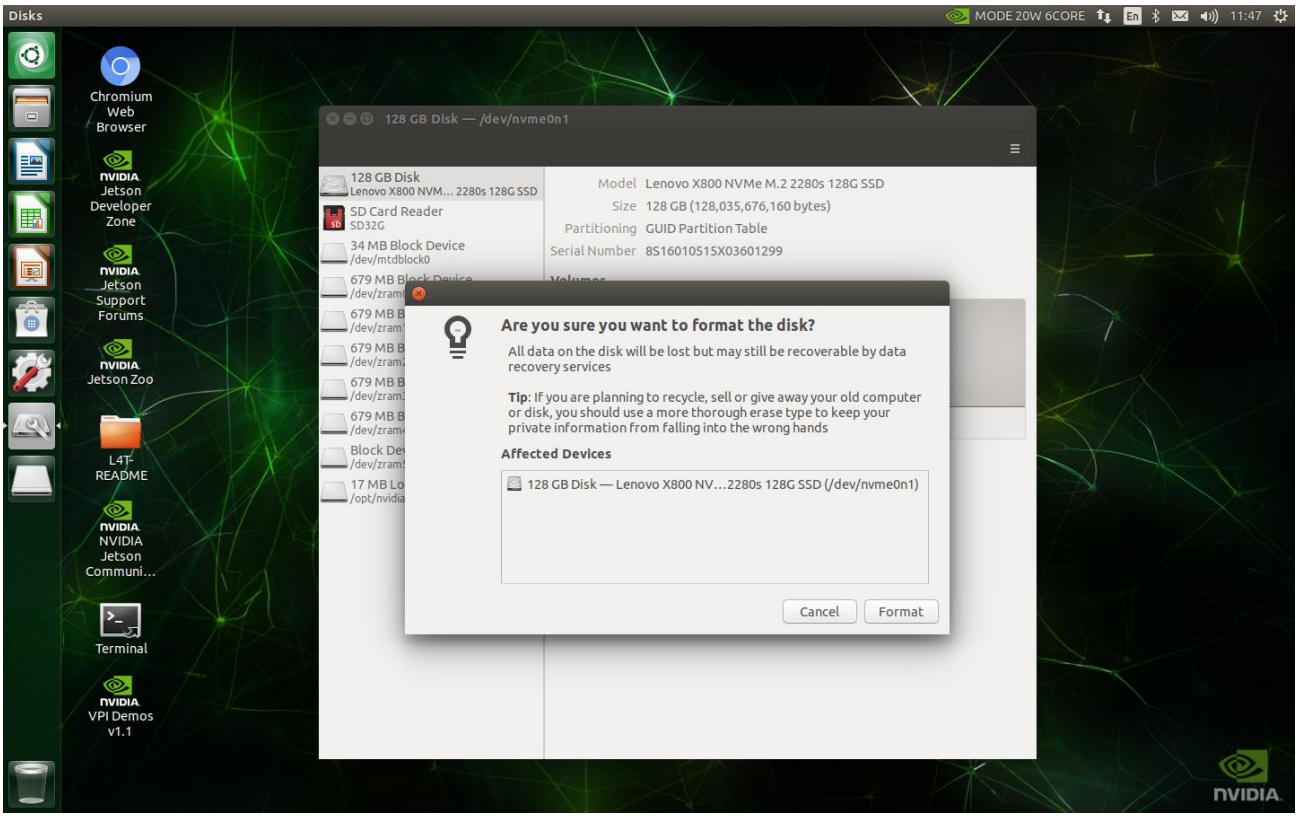
1)首先打开方式为按 ctrl 键出现搜索框,搜索”disks”工具;



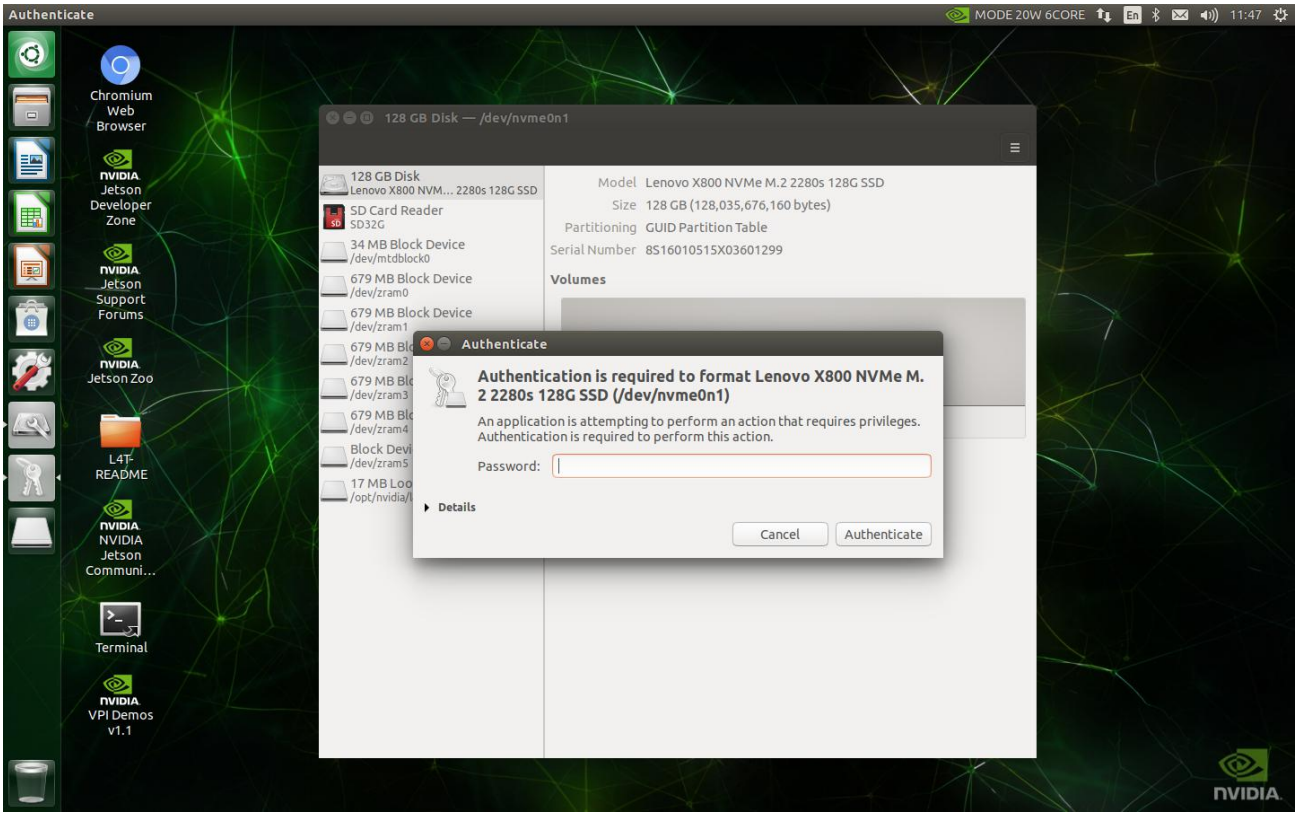
2)进入 disks,找到安装的 ssd,然后组合按键”ctrl+F”,对硬盘进行快速格式化,点击 Formaat;



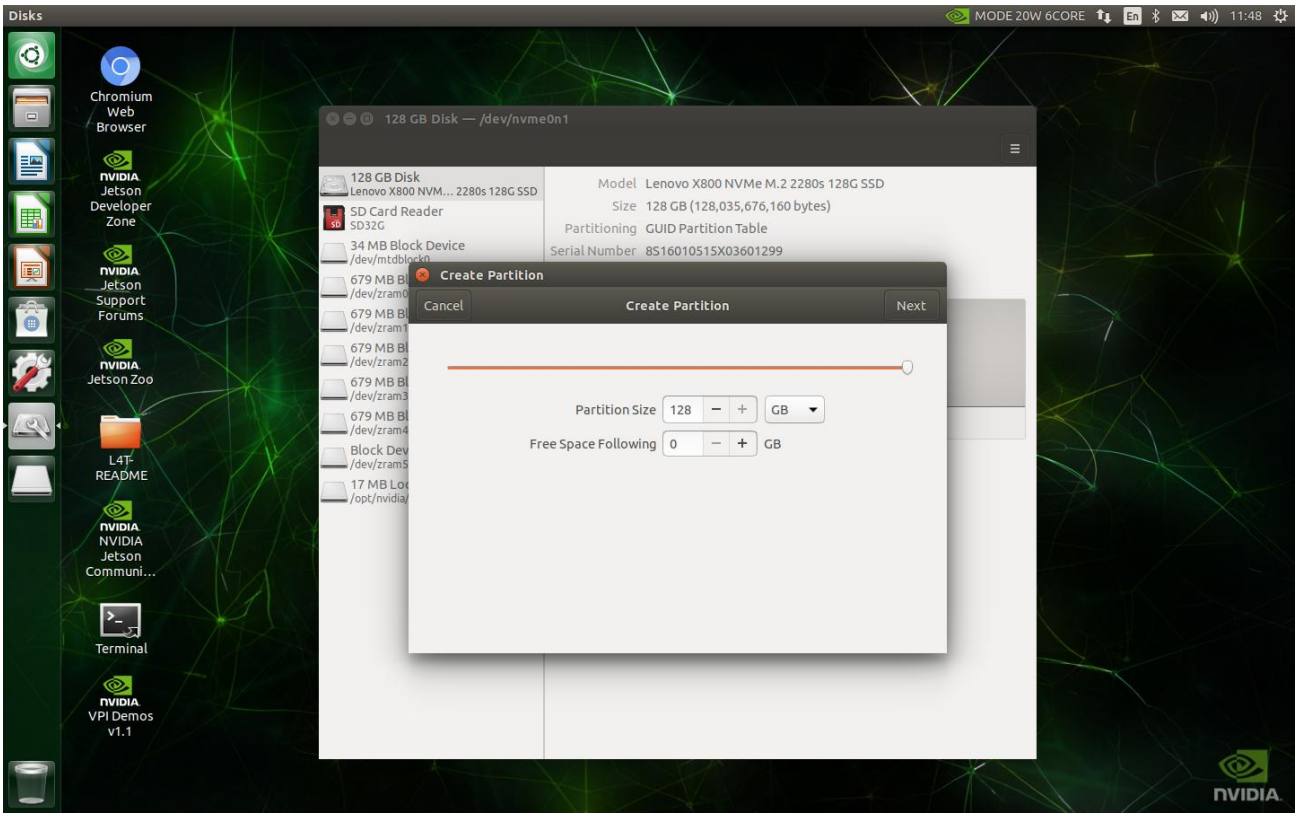
3)继续点击 format



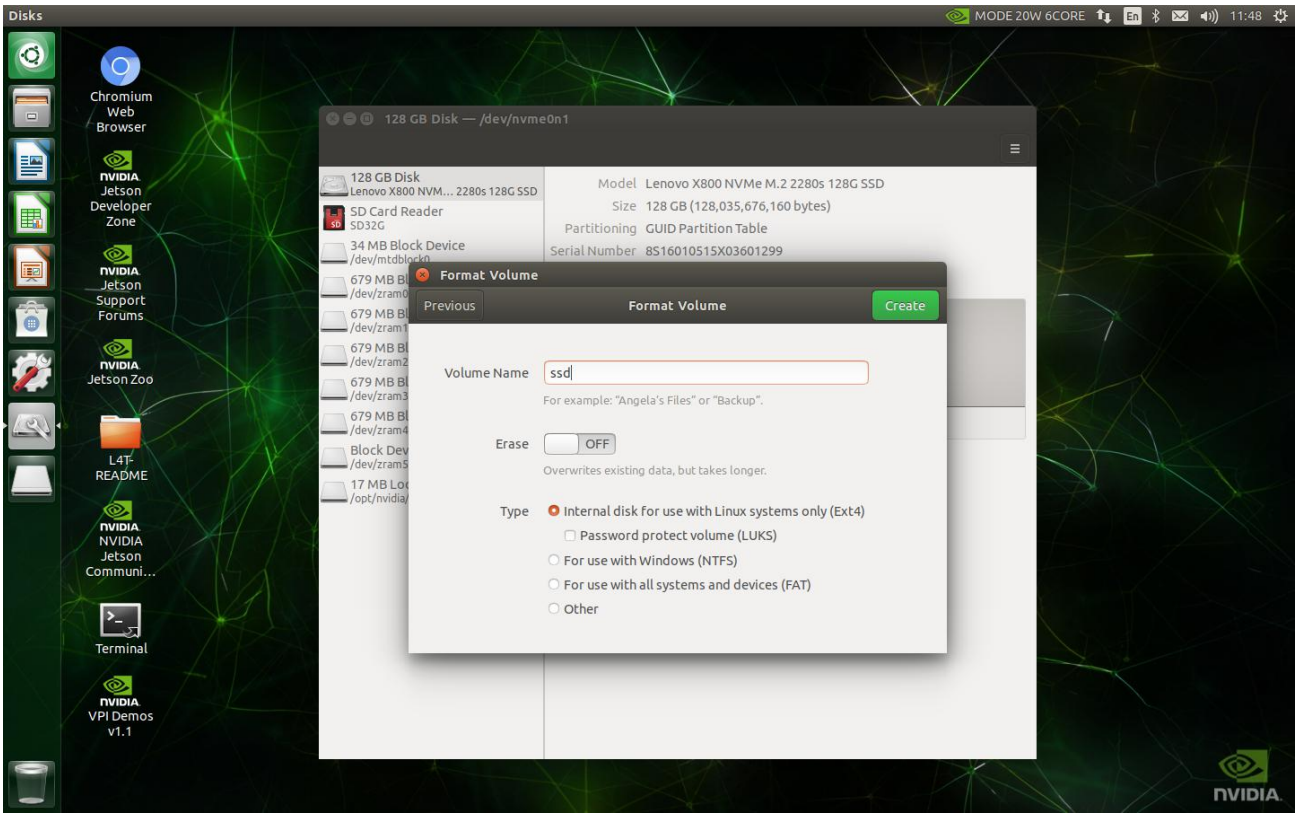
4)输入密码,若未自己修改密码,则输入默认密码”nvidia”(注意大小写)



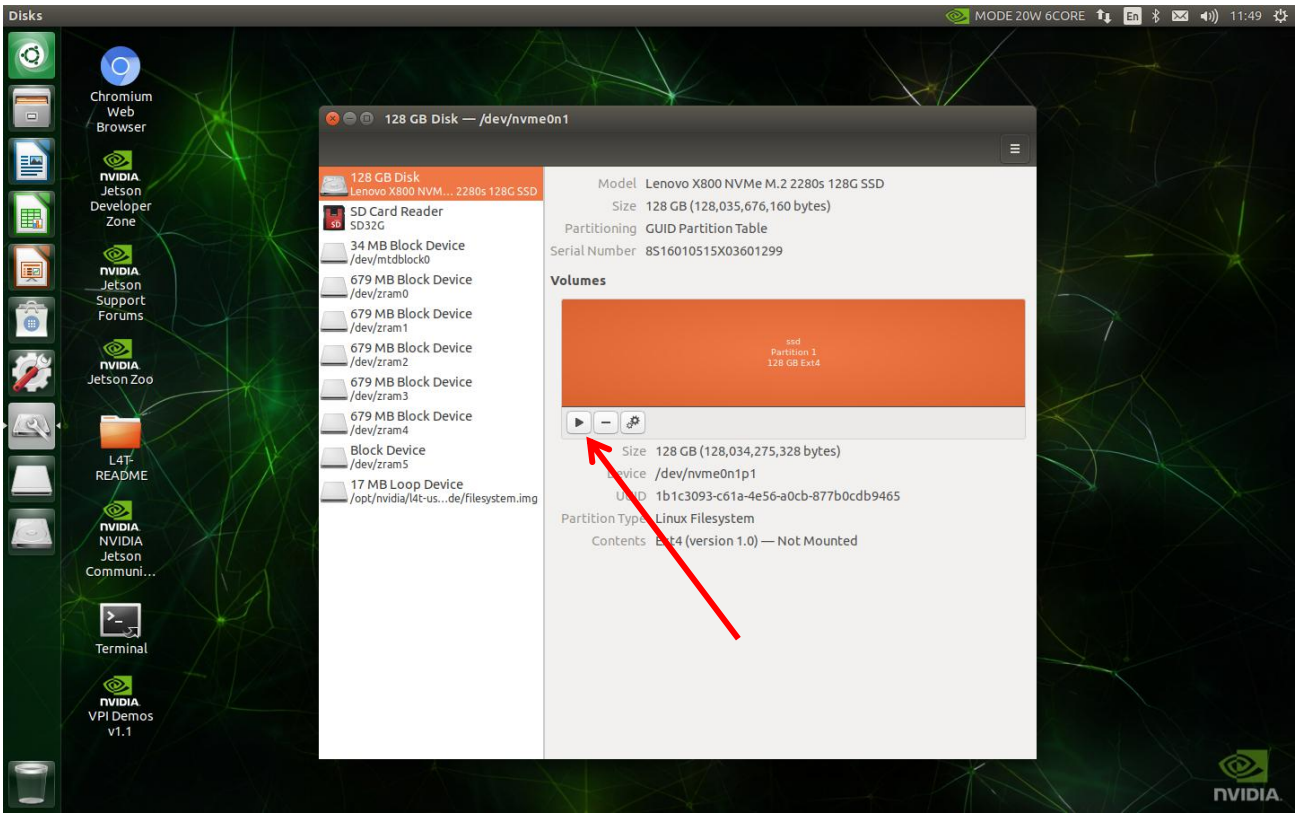
5)可以自己划分大小占比,下图是默认最大分区,点击 next



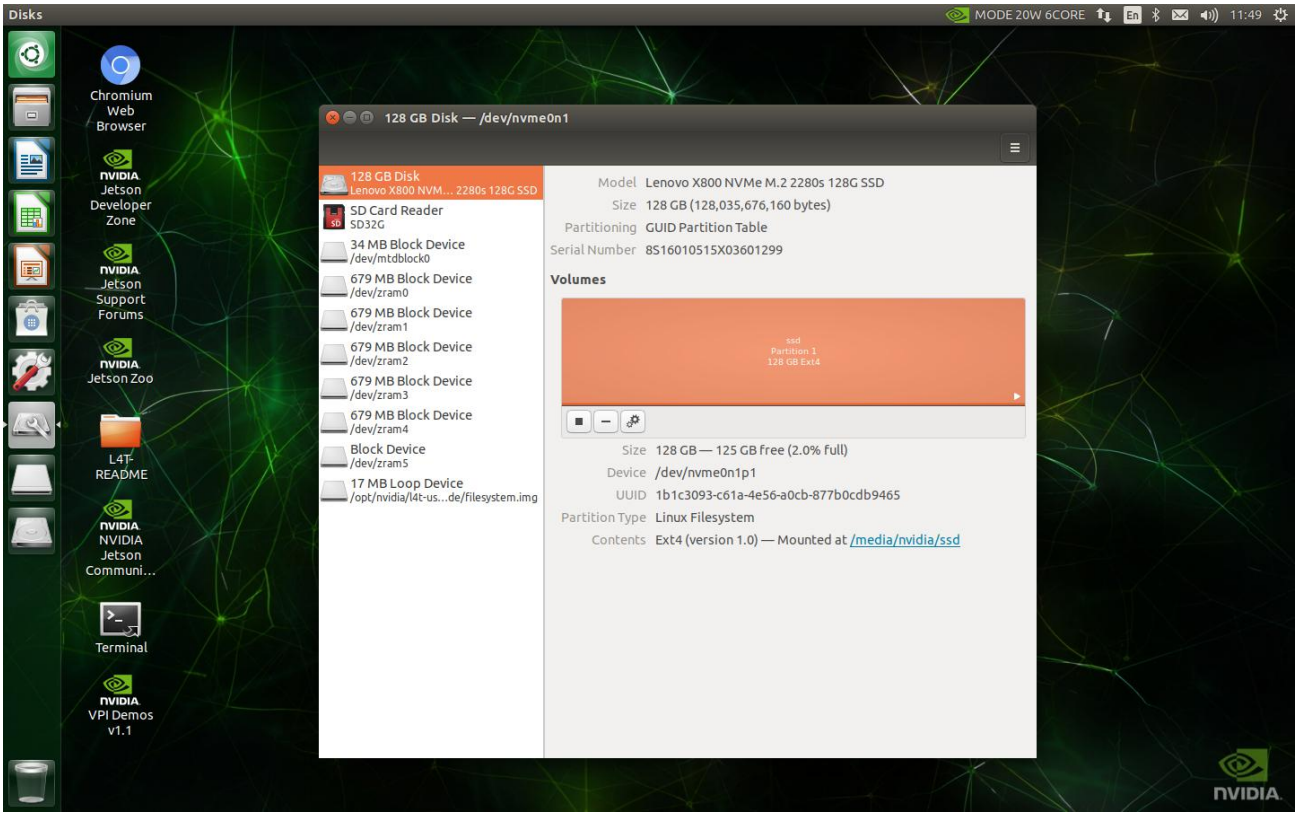
6)给分区起名字,其他均为默认选项无需修改,然后点击 create;



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

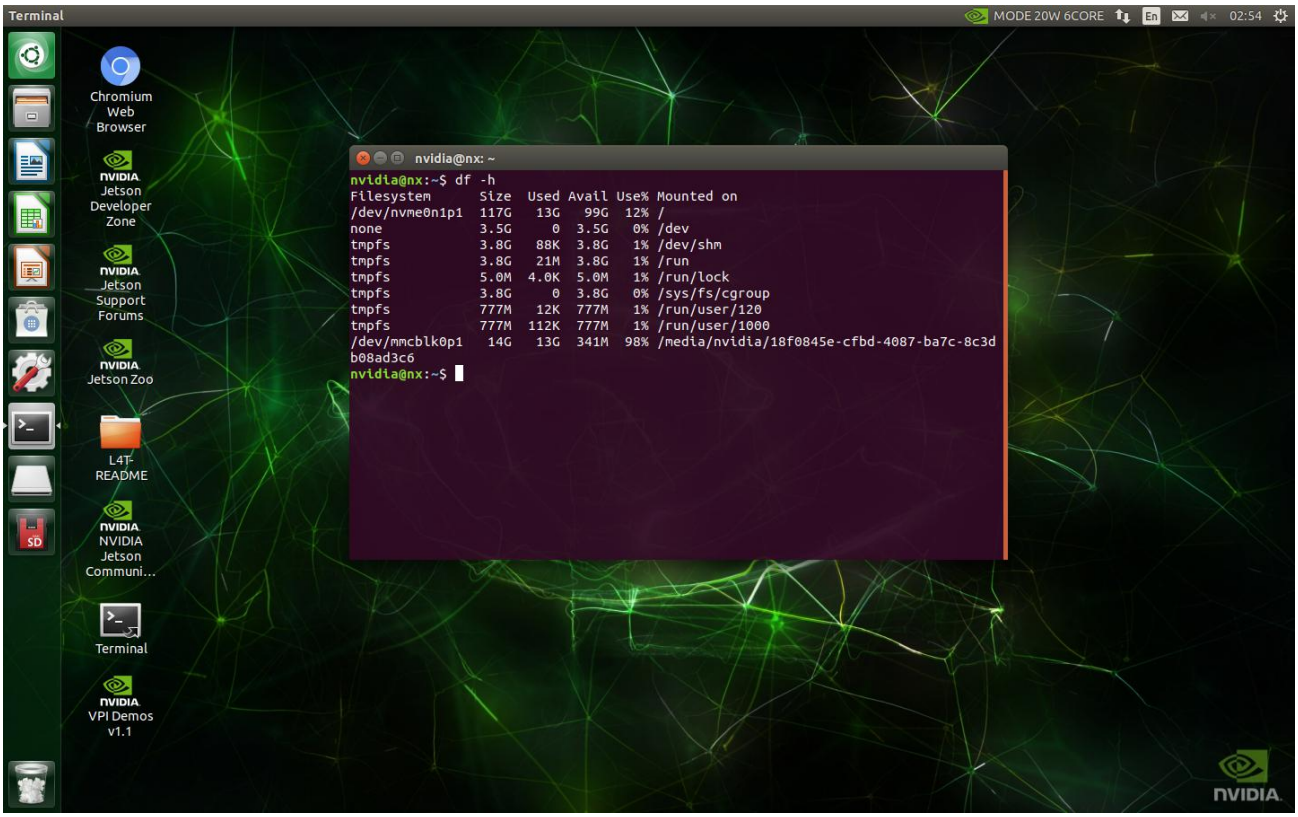
./copy-rootfs-ssd.sh(这个时间比较久,请耐心等待执行完毕再执行下一步)

./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



系统安装

我司 T609 产品系公司自行研发主板搭配 nvidia agx xavier 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

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 t609 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。

T609 进入 recovery 模式的步骤:

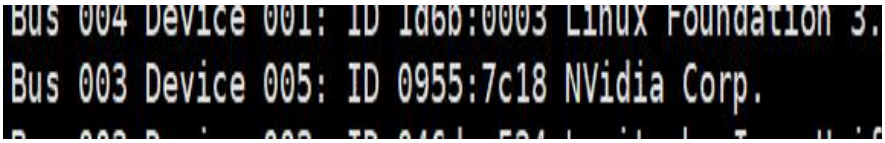
1）连接 t609 系统电源;

2）使用 usb type-c 的数据线连接 jetson 和 ubuntu host 主机(一端插在 t609 的 OTG 口,一端插在 ubuntu host 主机的 usb 3.0 插口);

3）使用 t609 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 t609 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）



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/18riNOT38uq2APqbNNQ9GVA

提取码：pr38

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:

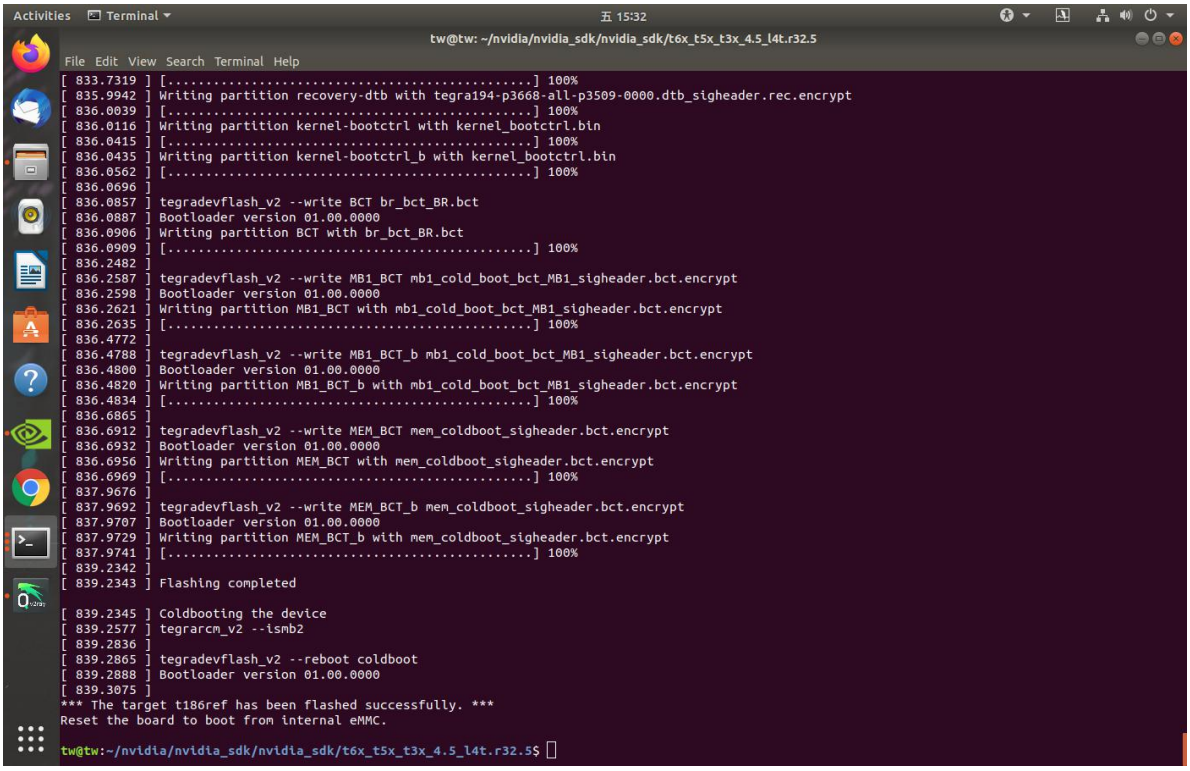
合并压缩包:sudo cat T609.JP4.5.tar.gz* > T609.4.5.tar.gz

解压:tar -zxvf T609.JP4.5.tar.gz

cd T609.JP4.5

执行刷机脚本 ./t609_gmsl2_senyun_imx390ar0233gw5200_.v1.2.flash.sh(如果执行不了,先输入 sudo chmod +x t609_gmsl2_senyun_imx390ar0233gw5200_.v1.2.flash.sh)

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.



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：

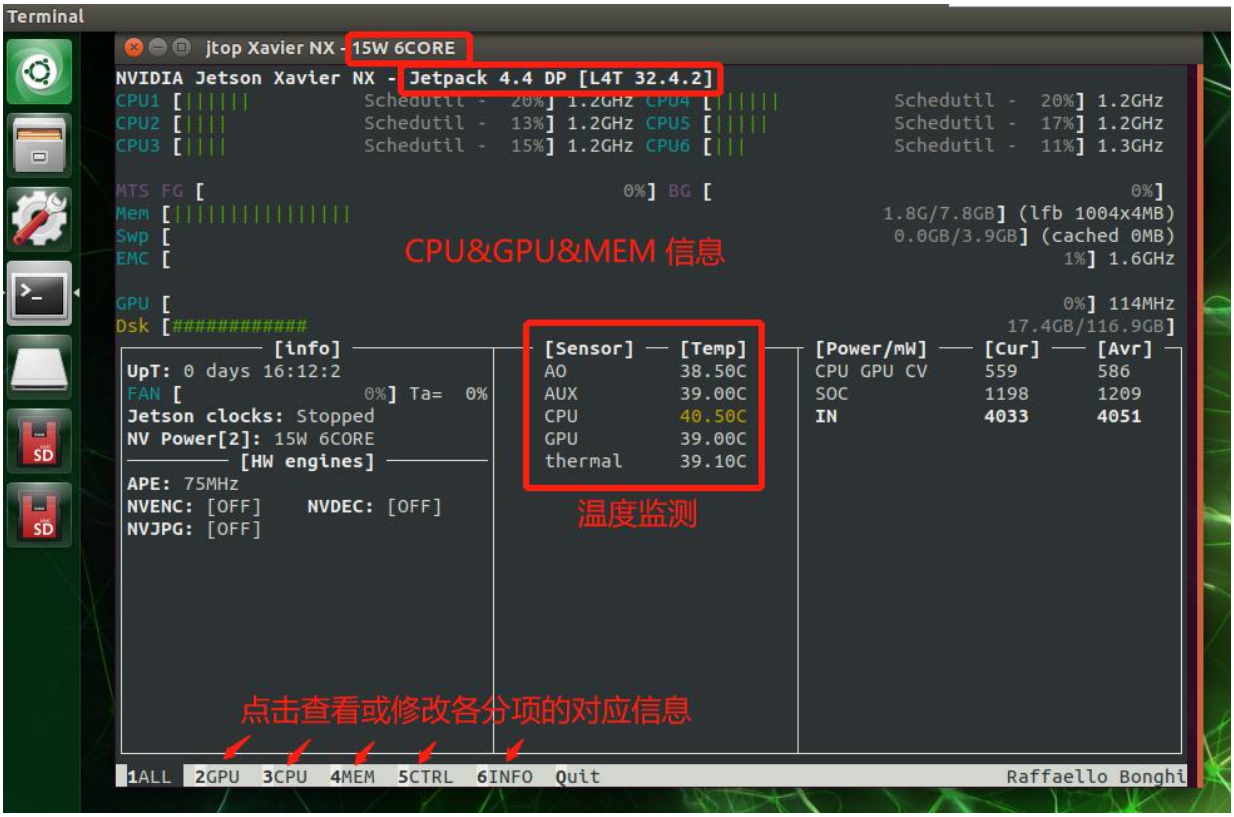
安装及运行

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、nvpmodel 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

```
nvidia@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 (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%@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 gvncviewer 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/1v2cc4ro6zvsbu0p8h6qcuaqco1qcsif.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 zlib1g-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/1J2DeI9HMtOMkH-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 拥有多层级结构，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。

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/redist/jp/v45 tensorflow# TF-1.15
sudo pip3 install --pre --extra-index-url https://developer.download.nvidia.com/compute/redist/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/redist/jp/v44 tensorflow==2.3.1+nv20.12
# TF-1.15
sudo pip3 install --pre --extra-index-url https://developer.download.nvidia.com/compute/redist/jp/v44 'tensorflow<2'
```

TensorFlow 安装包下载地址:

链接：https://pan.baidu.com/s/1gLC-lZ5pC4OASezrpfY2Rw

提取码：6iyv

保修条例

重要提示

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。

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。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）。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。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 陨坏。

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。