

HYAI-311UAV-Orin_NX 测试说明

主题	HYAI-311UAV-Orin_NX 测试说明
文档号	1.0
创建时间	2023-04-12
最后修改	2023-05-19
版本号	1.0
文件名	HYAI-311UAV-Orin_NX 测试说明.pdf
文件格式	Portable Document Format

阅前须知

声明

北京合众恒跃科技有限公司保留随时对其产品进行修正、改进和完善的权利，客户在下单前应获取相关信息的最新版本，并验证这些信息是正确的。本文档一切解释权归北京合众恒跃科技有限公司所有。

简介

北京合众恒跃科技有限公司（简称：HZHY）成立于 2011 年，是一家专业提供嵌入式应用解决方案的高新技术企业。本企业位于中关村高科技园区，专注于嵌入式产品研发、生产、销售等工作，并有着多年的嵌入式系统软、硬件技术积累，同时拥有一支技术精湛的嵌入式研发团队。该团队在嵌入式原理设计、信号完整性、电磁兼容性、电源冗余设计、系统热设计、完整的测试体系方面具有丰富的产品研发经验。

合众恒跃以卓越的服务品质、专业安全的技术服务实力，为不同的客户群体提供更好、更优质的嵌入式产品研发及生产服务。我们一直秉持“您身边的定制专家”理念，与诸多客户保持良好的合作关系，并致力于将低功耗、高性能、稳定可靠的嵌入式产品应用于工业控制、智能仪表、智能图像分析等诸多领域。

联系方式

北京合众恒跃科技有限公司

地址：北京市海淀区安宁庄后街南 1 号 A 区一层 1020 号

电话：010-62129511-8001

传真：010-62126811

官网：<http://www.hzhytech.com>

目录

阅前须知	2
声明	2
简介	2
联系方式	2
目录	3
第 1 章 板卡测试需求说明	5
1.1 硬件需求说明	5
1.2 软件需求说明	5
1.3 程序需求说明	6
第 2 章 板卡硬件检验	7
2.1 硬件外观检验	7
2.2 硬件通电检验	7
第 3 章 板卡测试说明	11
3.1 电源测试	11
3.2 HDMI 显示测试	14
3.3 USB3.0 测试	14
3.4 USB2.0 接口测试（Type-A）	17
3.5 USB2.0 与 USB3.0 接口测试（锁扣）	18
3.6 网口测试	19
3.7 调试串口测试	20
3.8 SSD 测试	21
3.9 CAN 测试	22
3.10 RS232 测试	24

3.11 Camera 测试	25
3.12 GPIO 测试	27
3.13 BMI160 传感器测试（焊 U2 后测此项）	27
3.14 烧写端口测试	28
3.15 HYAI-311UW 测试	28
3.15.1 电源测试	28
3.15.2 4G 测试	29
3.15.3 WIFI/蓝牙测试	31
3.16 HYAI-3115g 测试	35
3.16.1 电源测试	35
3.16.2 5G 测试	36

第 1 章 板卡测试需求说明

1.1 硬件需求说明

Table 1.1-1 硬件需求表

资源名称	描述	用途及范围	数量
HYAI-310UAV_V20	782134910	Jetson Xavier NX/NANO/Ori-NX 载板	1
网线	常规五类线	将板卡和服务器连接	5
Type-C		下载	1
USB 鼠标		操作显示屏	1
HDMI 显示屏		用于显示	1
USB 转 RS232 (TTL 电平)		用于 RS232 通信	1
电源	12V/5A	给板卡供电	1
imx219 摄像头		用于测试 CSI 接口	4
CAN-bus 接口卡		用于测试 CAN	1
U 盘		用于测试 USB2.0、USB3.0	
USB3.0 工业相机		用于测试 USB3.0	

1.2 软件需求说明

Table 1.2-1 硬件需求表

资源名称	描述	用途及范围
操作系统	Win7/XP 以上	用做服务器
Ubuntu18.4/		用做服务器

SecureCRT		用于串口信息显示
UartAssist.exe		用于测试 RS232 串口

注：上述软件，在板卡配套文件中给出

1.3 程序需求说明

Table 1.3- 1 程序需求表

资源名称	描述	用途及范围
carrierTest 包		测试应用镜像

注：上述软件，在板卡配套文件中给出

第 2 章 板卡硬件检验

2.1 硬件外观检验

- a. 目测板卡整洁度；
- b. 目测原件焊接虚实；
- c. 目测是否缺焊元件；
- d. 使用数字万用表检验各电源极与公共地是否存在短路

2.2 硬件通电检验

- a. 通电后测量各个电源极输出电压是否符合标准。
- b. LD1 灯常亮

2.3 板卡程序固化

一、硬件连接

J9 的 1 脚接仿真器的 C2D(4)，2 脚接仿真器 C2CK(7)，3 脚接仿真器的 GND(2)

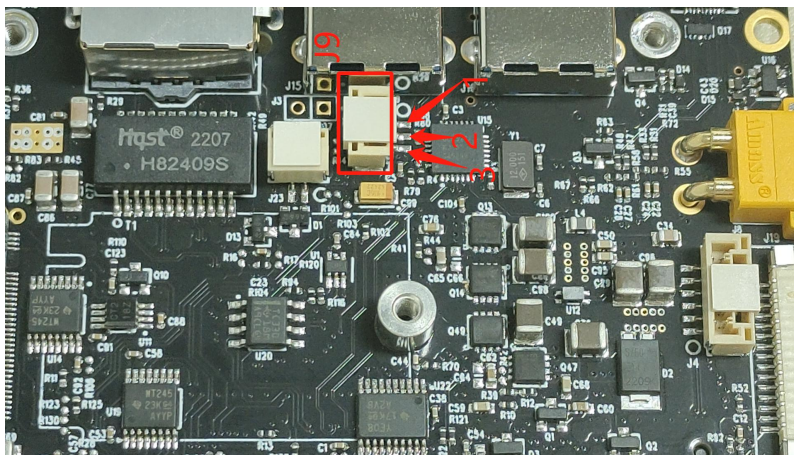


图 2-1

U-EC6 仿真器示意图

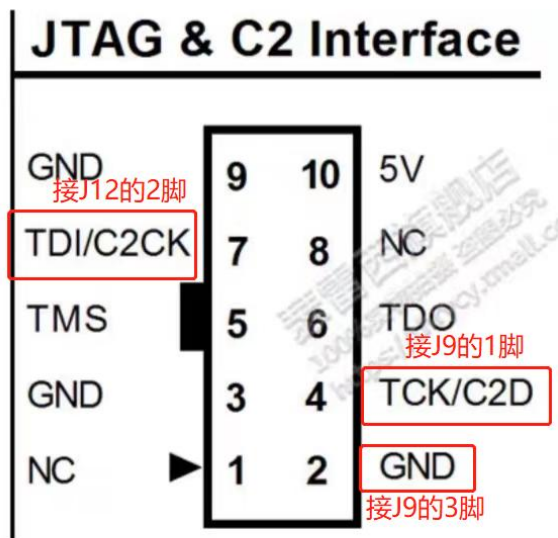
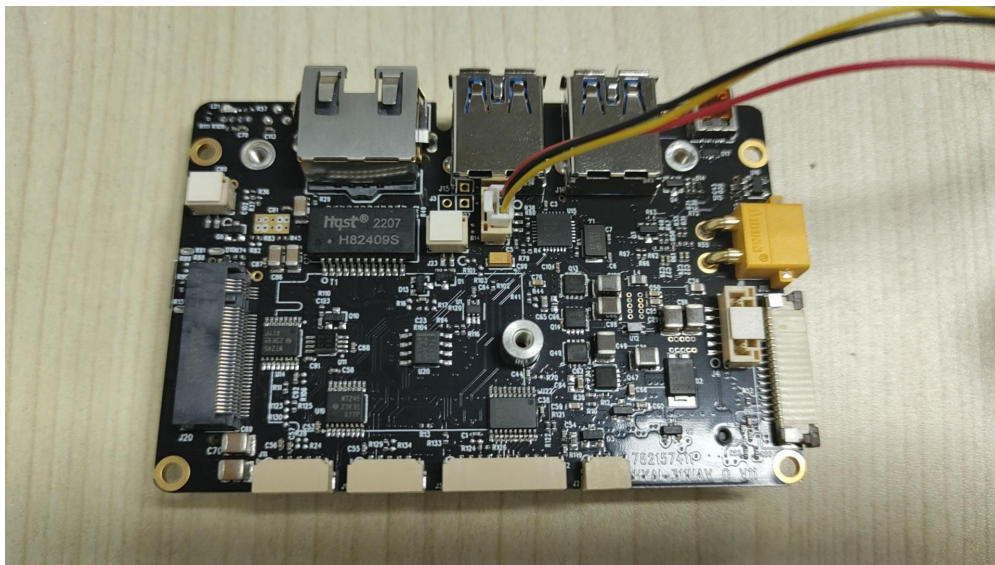


图 2-2

二、 程序烧写

1 板卡连接好仿真器并上电：



2 打开 MCUProductionProgrammer 应用程序，进行如下配置：

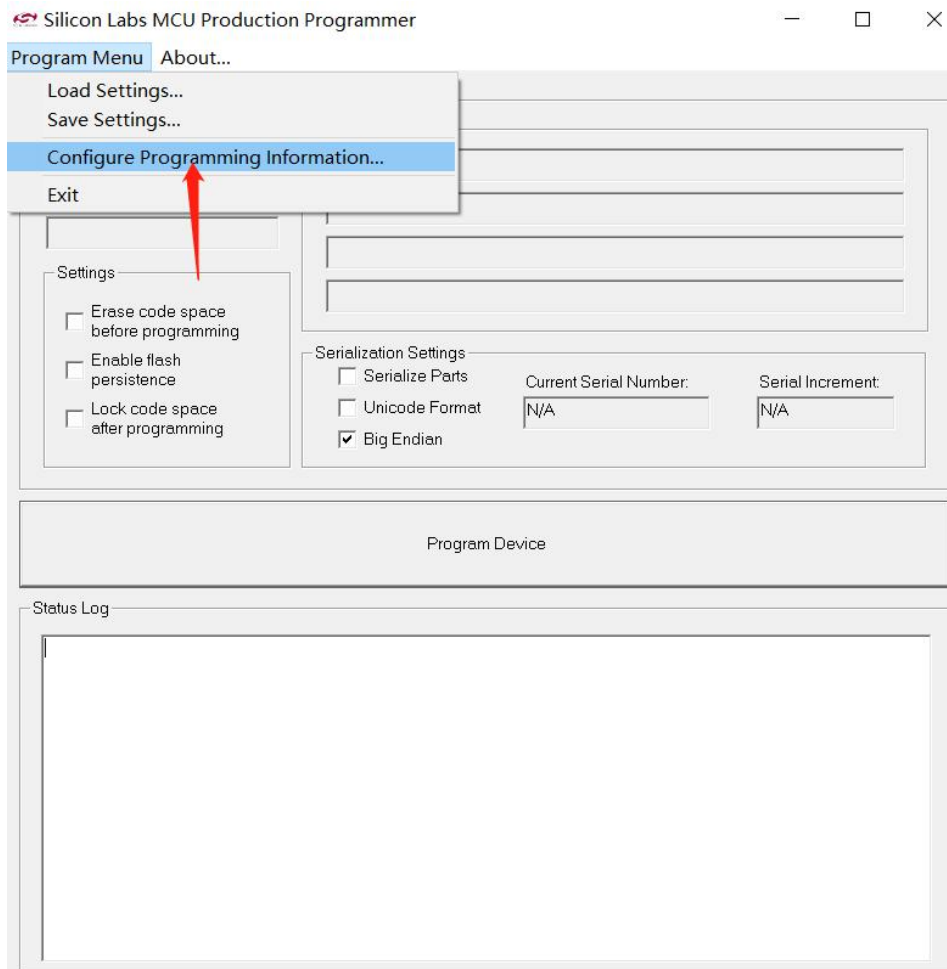


图 2-3

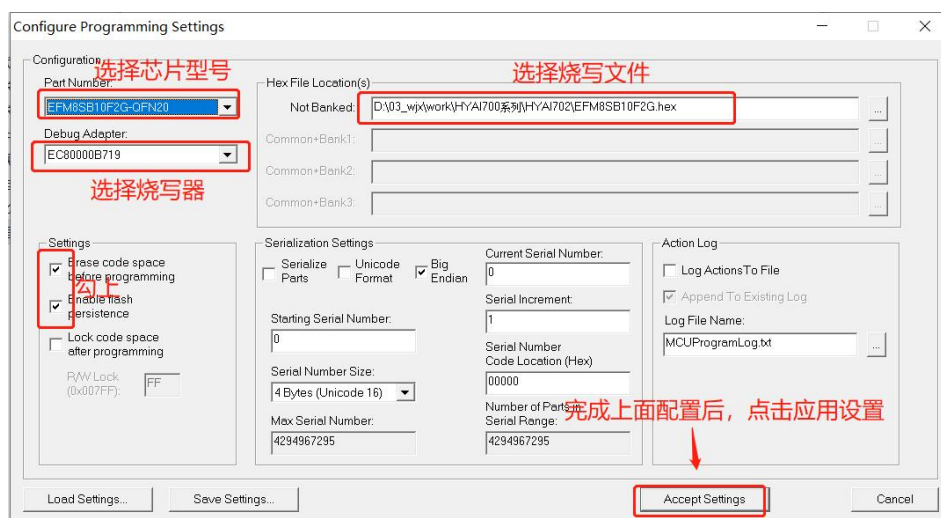


图 2-4

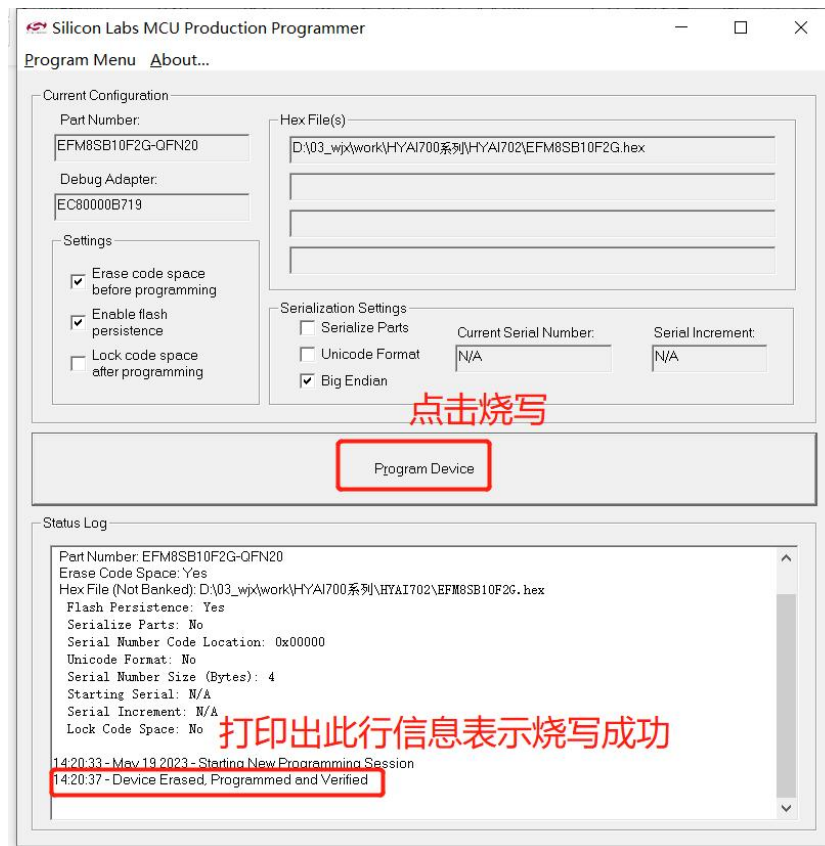


图 2-5

第 3 章 板卡测试说明

3.1 电源测试

将底板 J3 短接，不连接核心板，用万用表测量底板的 1.8V、3.3V、5V 的测试点，1v8 和 3V3 的测试点如图 3-1，5V 需要量 C98 和 C100 如图 3-2,测试正确无误后再测试下面的步骤。（测试底板电源时，需测量正确后在连接核心板。）

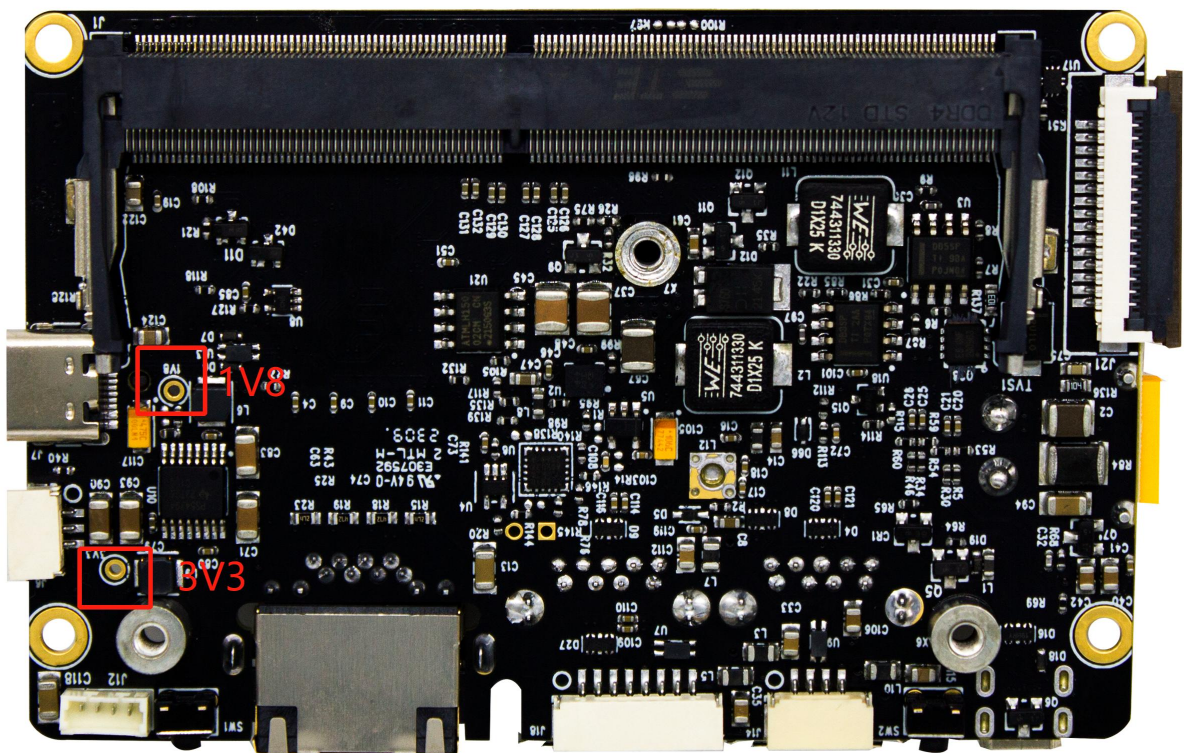


图 3-1

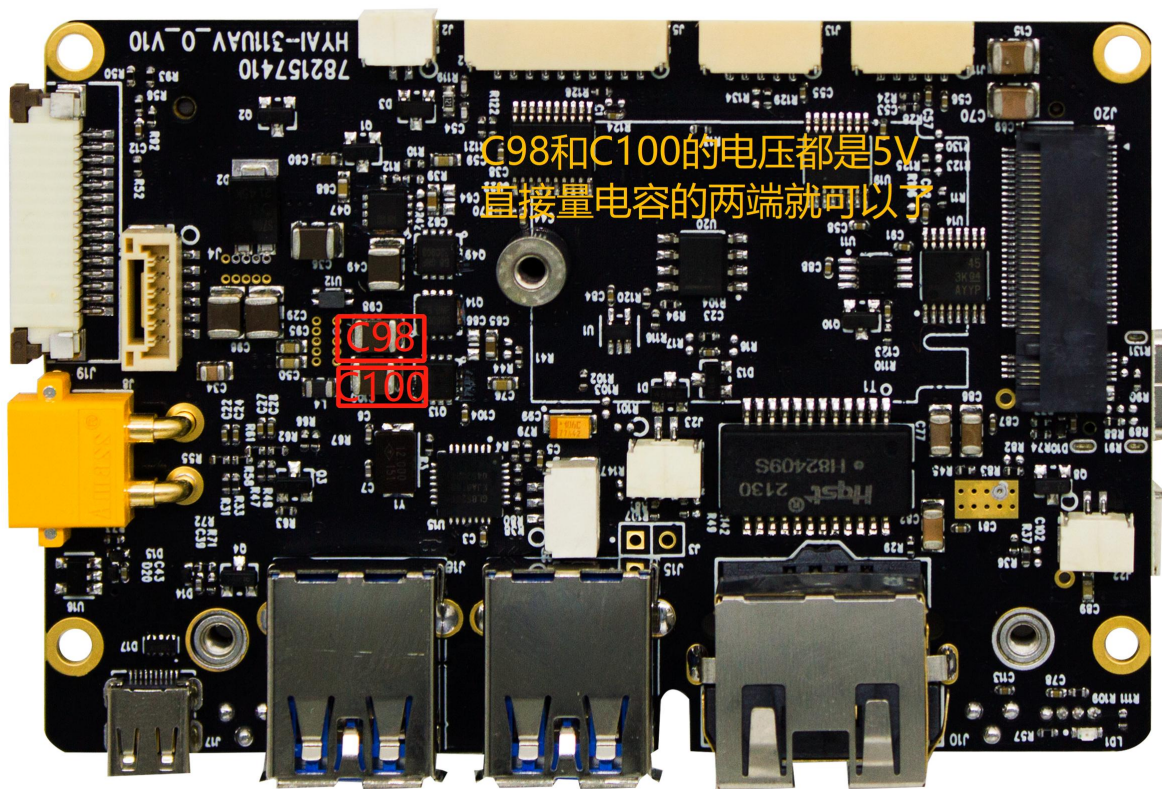


图 3-2

Orin-NX 除无 TF 功能以，外完全兼容 311UAV 板卡。

将 HDMI 显示屏连接板卡，连接 USB 鼠标、USB 键盘，连接 DEBUG 调试串口，连接 Type-C 烧写线到 PC 机，连接 2 路摄像头，连接 CAN 盒，USB2.0 连接 U 盘，连接网口到 PC 机，如图 3-3、图 3-4

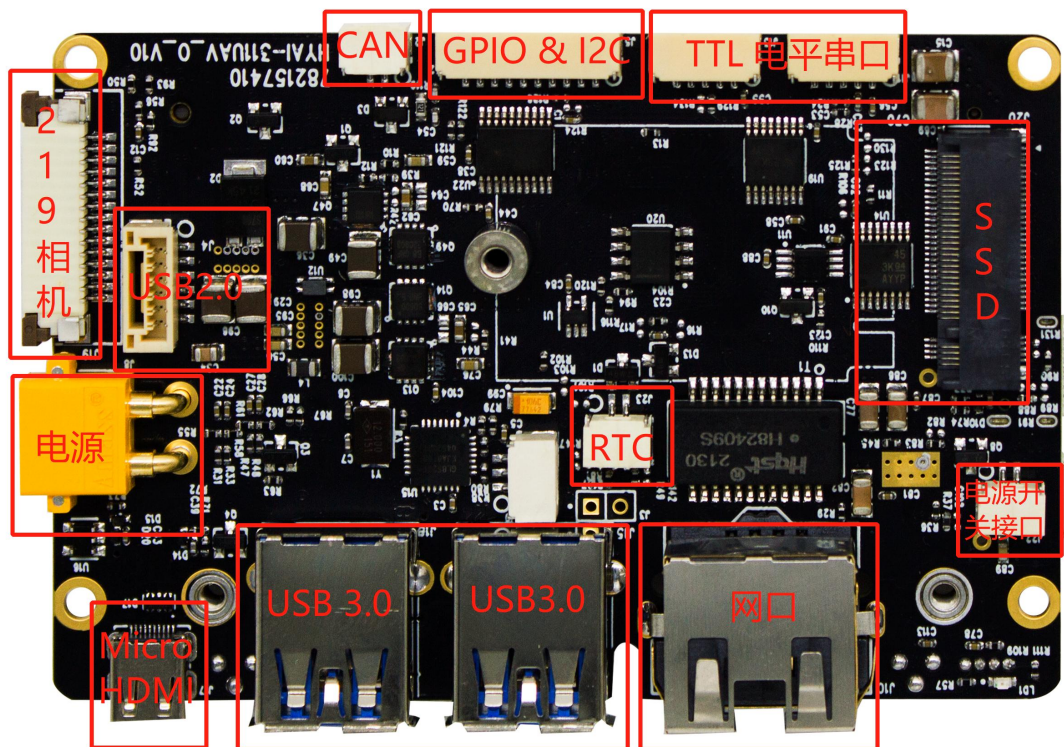


图 3-3

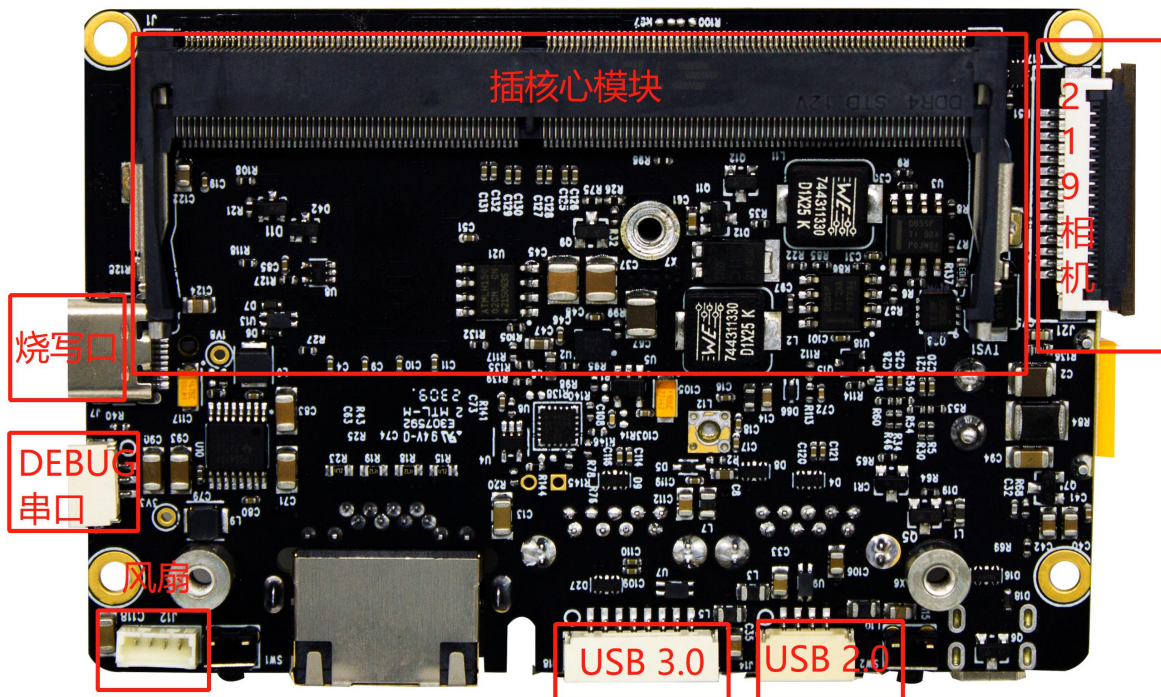


图 3-4

然后给板卡上电启动板卡，输入用户名 `nvidia` 及密码 `nvidia` 登录系统。

进入 `carrierTest` 目录。

```
nvidia@localhost:~$ cd /hzy/carrierTest/
```

3.2 HDMI 显示测试

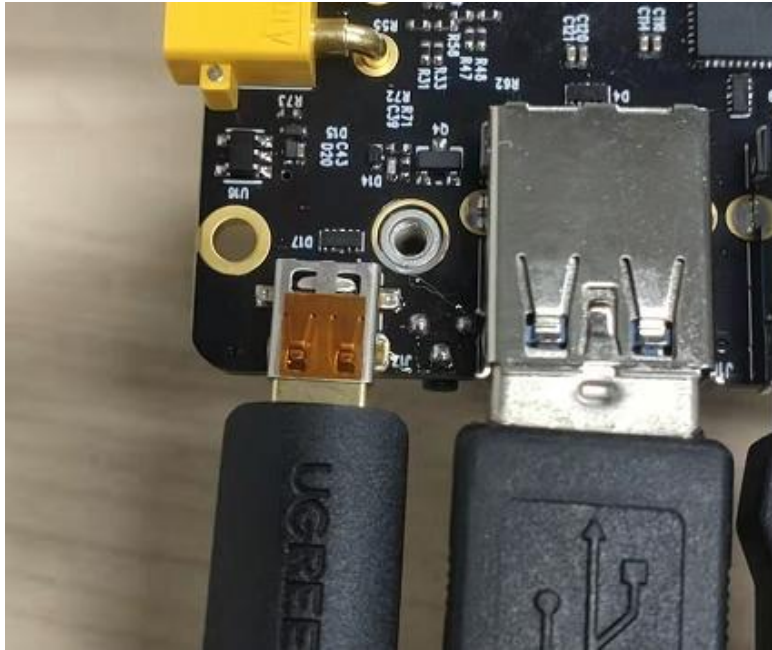


图 3-5

板卡上电后，会在 HDMI 显示屏上显示 `ubuntu` 登录系统，输入用户名和密码（烧录完成系统启动时设置的用户名及密码）登录系统。能正常登陆后，表示 HDMI 接口正常。

3.3 USB3.0 测试

方法 一 使用 USB3.0U 盘：

将载板的 USB3.0 接口，在上电后接入 USB3.0U 盘（为了测试热插拔），如图 3-6

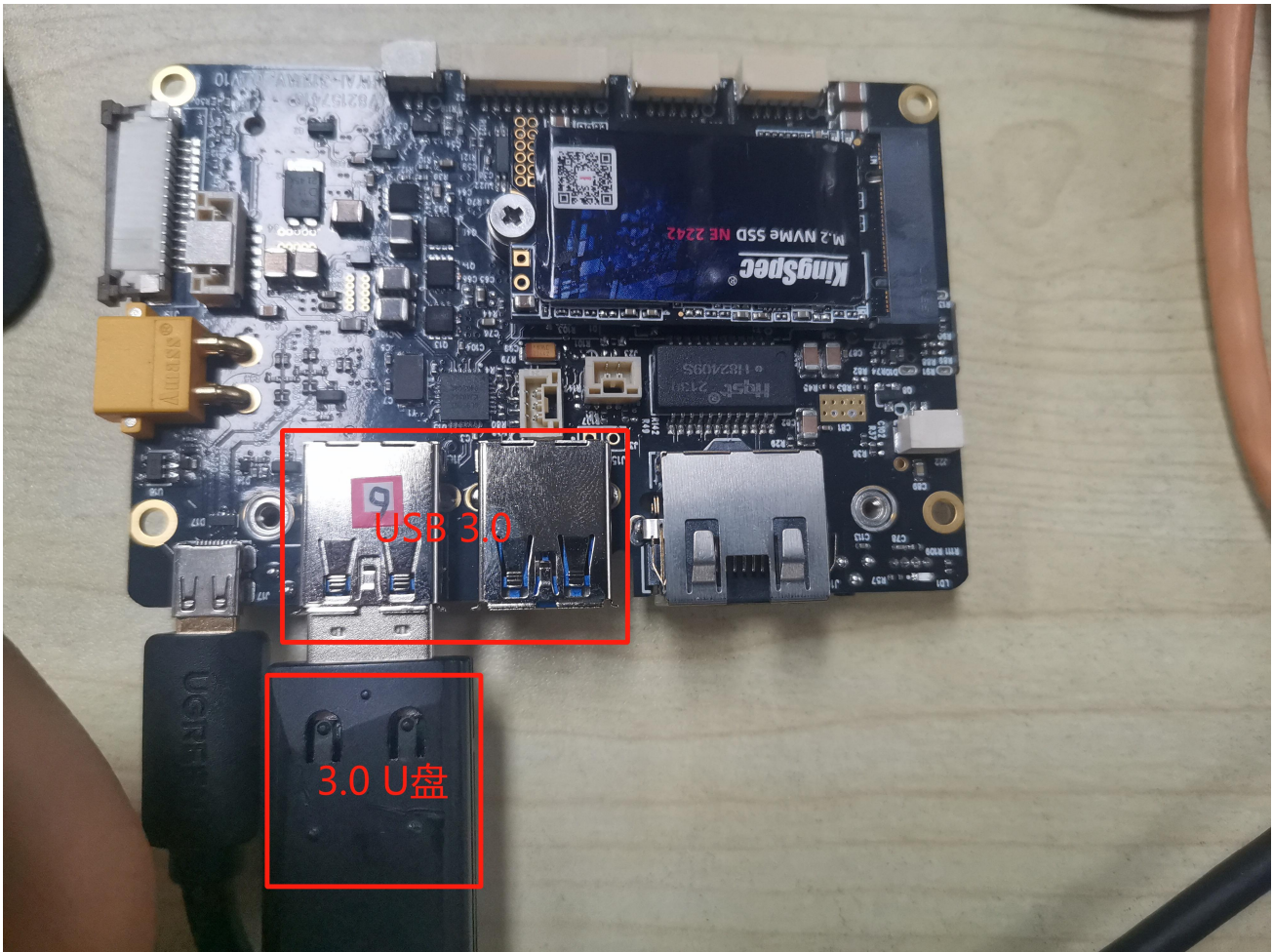


图 3-6

在串口端输入下面的命令，在最下面一行为最新的打印，如果识别到的是 SuperSpeed 则表示 USB3.0 测试成功。

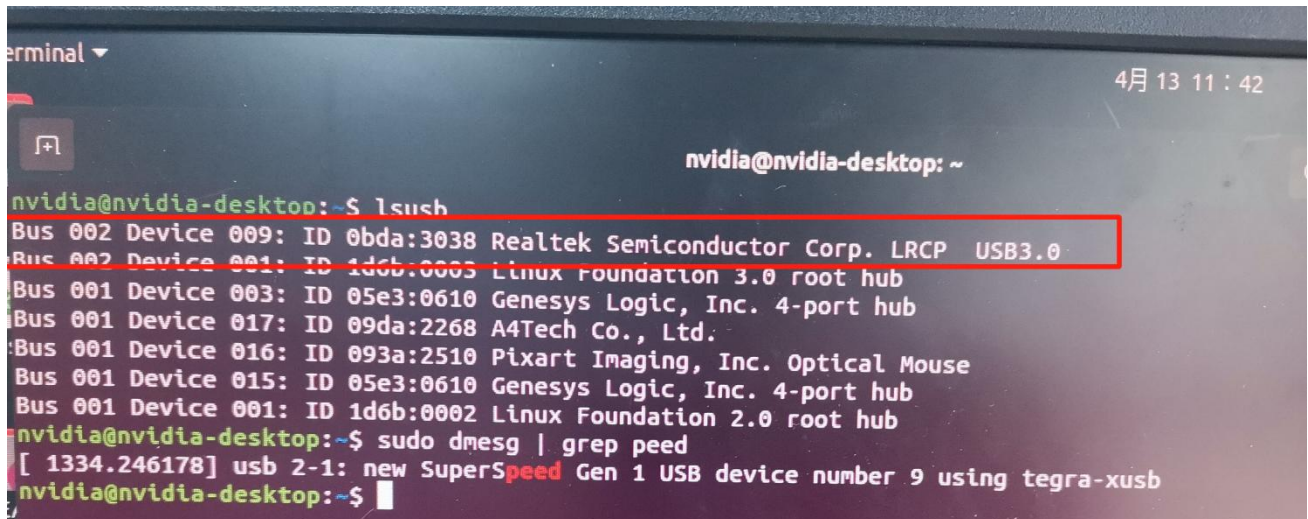
nvidia@nvidia:~\$ dmesg | grep speed

```
nvidia@nvidia:~$ dmesg | grep speed
5.180653] tegra-xusb 3610000.xhci: Host supports USB 3.1 Enhanced SuperSpeed
5.512242] usb 1-2: new high-speed USB device number 2 using tegra-xusb
5.656473] usb 2-3: new SuperSpeed USB device number 2 using tegra-xusb
5.824223] usb 1-2.1: new high-speed USB device number 3 using tegra-xusb
5.964539] usb 2-3.1: new SuperSpeed USB device number 3 using tegra-xusb
6.164265] usb 1-2.1.1: new low-speed USB device number 4 using tegra-xusb
9.046539] tegra-pcie-dw 14160000.pcie: Setting init speed to max speed
9.613246] tegra-pcie-dw 141a0000.pcie: Setting init speed to max speed
90.788525] usb 2-3.2: new SuperSpeed USB device number 4 using tegra-xusb
104.468243] usb 1-2.1: new low-speed USB device number 5 using tegra-xusb
104.964238] usb 1-2.1: new low-speed USB device number 6 using tegra-xusb
[ 222.828497] usb 2-3.2: new SuperSpeed USB device number 5 using tegra-xusb
nvidia@nvidia:~$
```

图 3-7

方法 二 使用 USB 3.0 工业相机:

首先将 USB3.0 工业相机与板卡的 Tyoe-A 3.0 接口相连，执行 lsusb 查看相机节点 3038，如图 3-8



```
terminal ▾ 4月 13 11:42
nvidia@nvidia-desktop: ~
nvidia@nvidia-desktop:~$ lsusb
Bus 002 Device 009: ID 0bda:3038 Realtek Semiconductor Corp. LRCP USB3.0
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 05e3:0610 Genesys Logic, Inc. 4-port hub
Bus 001 Device 017: ID 09da:2268 A4Tech Co., Ltd.
Bus 001 Device 016: ID 093a:2510 Pixart Imaging, Inc. Optical Mouse
Bus 001 Device 015: ID 05e3:0610 Genesys Logic, Inc. 4-port hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
nvidia@nvidia-desktop:~$ sudo dmesg | grep peed
[ 1334.246178] usb 2-1: new SuperSpeed Gen 1 USB device number 9 using tegra-xusb
nvidia@nvidia-desktop:~$
```

图 3-8

然后使用命令 **cheese** 播放，画面流畅即为 USB3.0，如图 3-9,如果是 USB2.0 就会黑屏，如图 3-10 和图 3-11。

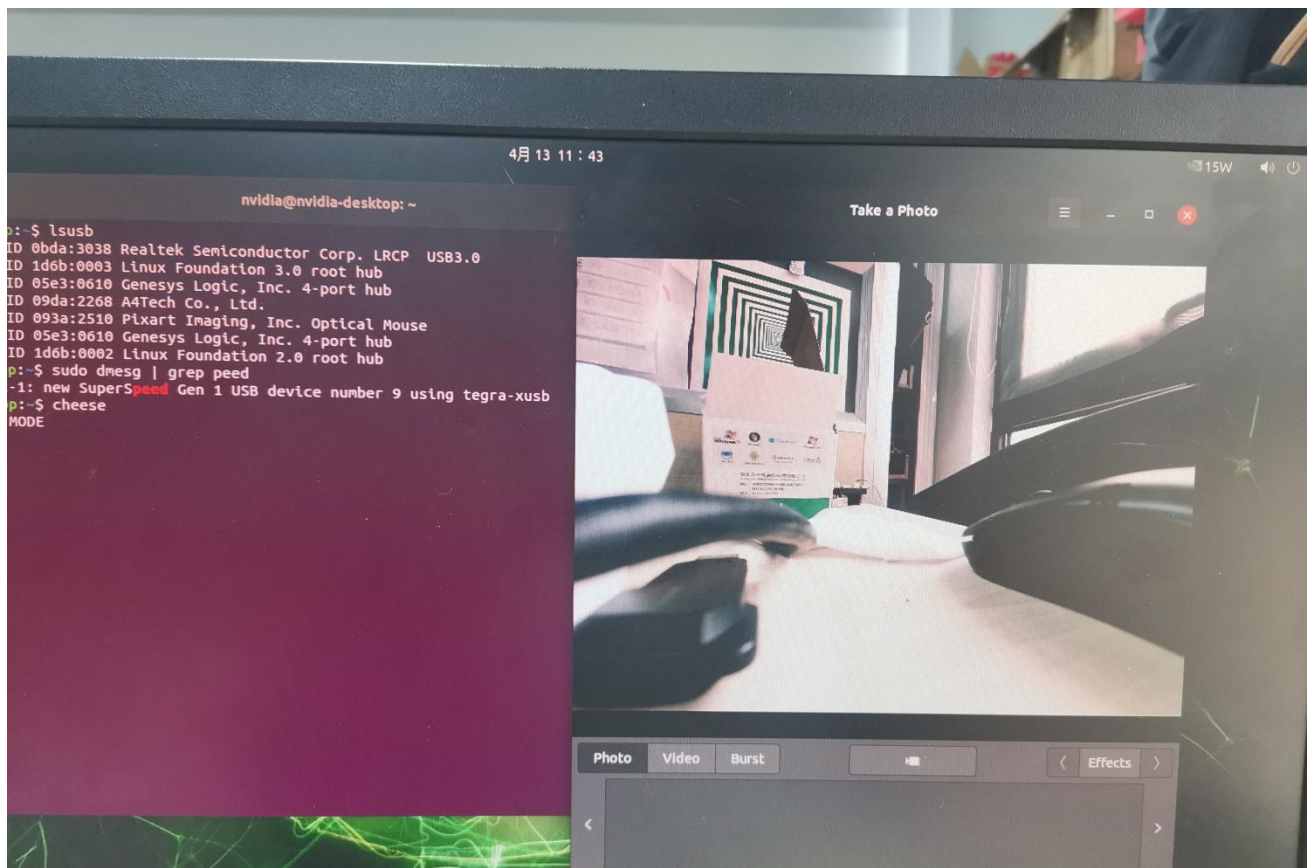


图 3-9

```
nvidia@nvidia-desktop:~$ lsusb
Bus 002 Device 010: ID 05e3:0626 Genesys Logic, Inc. USB3.1 Hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 020: ID 09da:2268 A4Tech Co., Ltd.
Bus 001 Device 019: ID 093a:2510 Pixart Imaging, Inc. Optical Mouse
Bus 001 Device 018: ID 05e3:0610 Genesys Logic, Inc. 4-port hub
Bus 001 Device 003: ID 05e3:0610 Genesys Logic, Inc. 4-port hub
Bus 001 Device 021: ID 0bda:3038 Realtek Semiconductor Corp. LRCP USB3.0
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
nvidia@nvidia-desktop:~$ sudo dmesg | grep peed
[ 1447.269161] usb 1-2: new high-speed USB device number 21 using tegra-xusb
nvidia@nvidia-desktop:~$
```

图 3-10

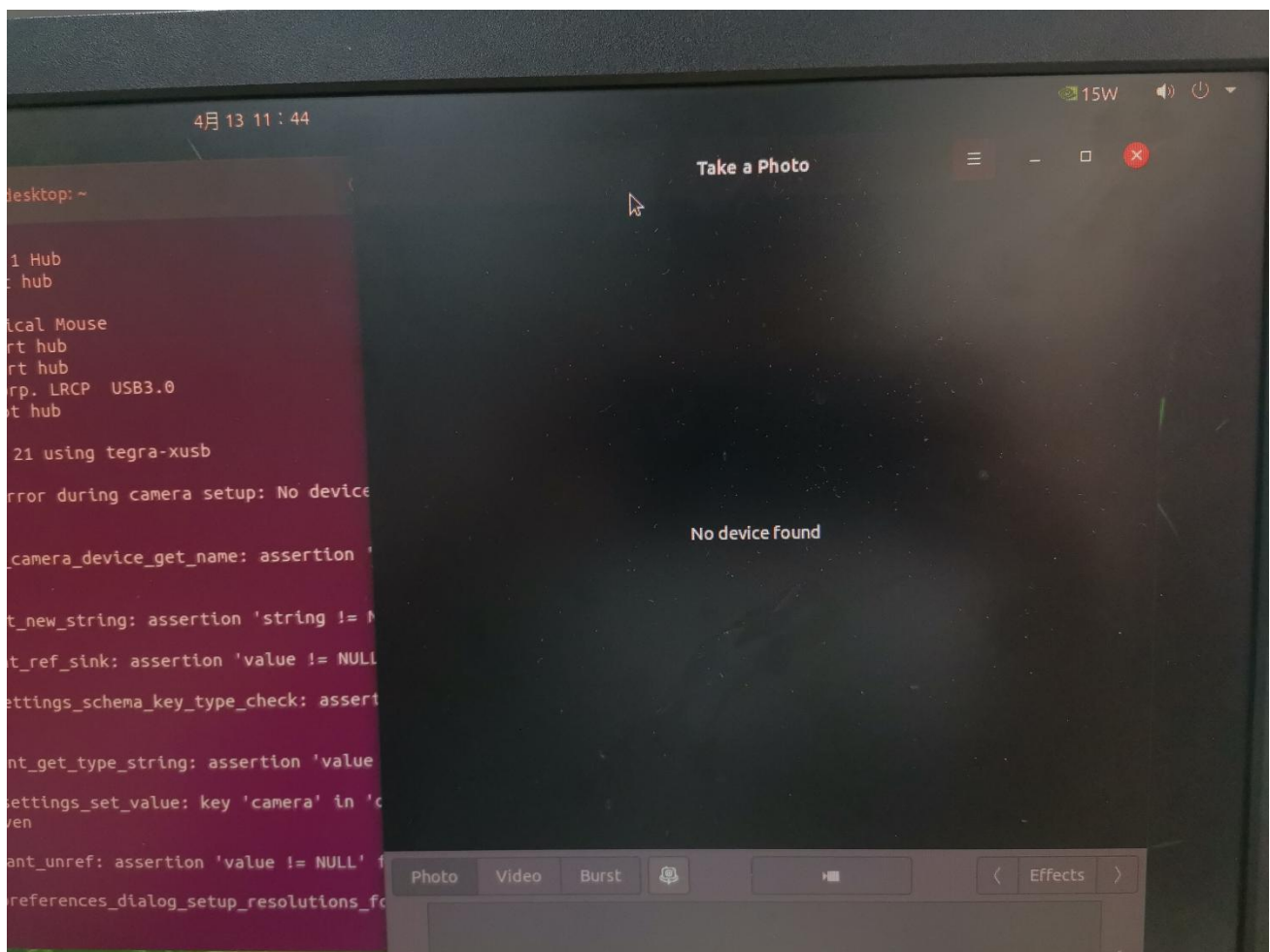


图 3-11

3.4 USB2.0 接口测试（Type-A）

USB 接口如图 3-12 所示：

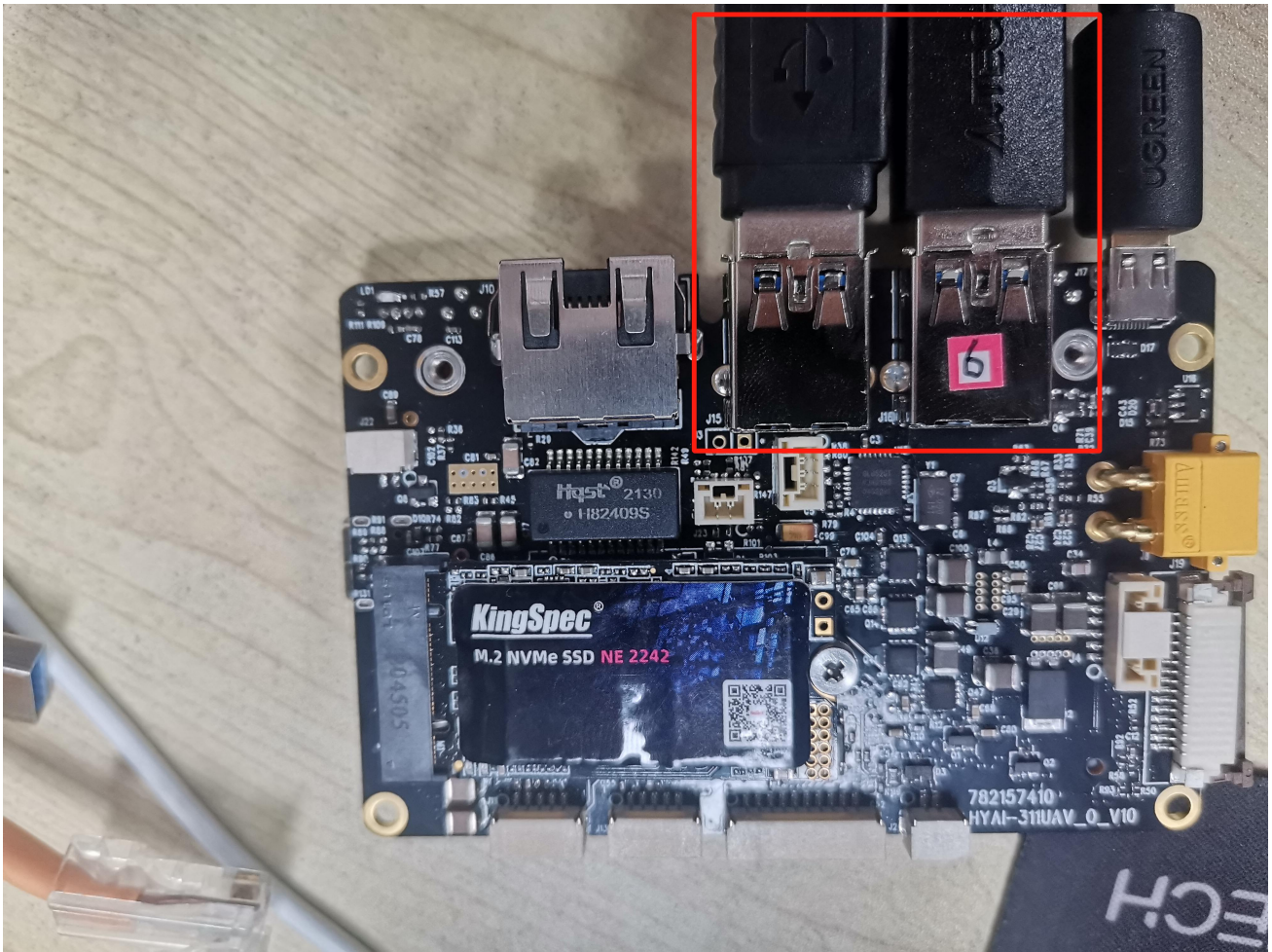


图 3-12

插入 USB 鼠标及键盘，操作鼠标及键盘，查看 HDMI 显示屏是否有动作。

3.5 USB2.0 与 USB3.0 接口测试（锁扣）

如图 3-13 所示为 USB2.0 和 3.0 的接口。

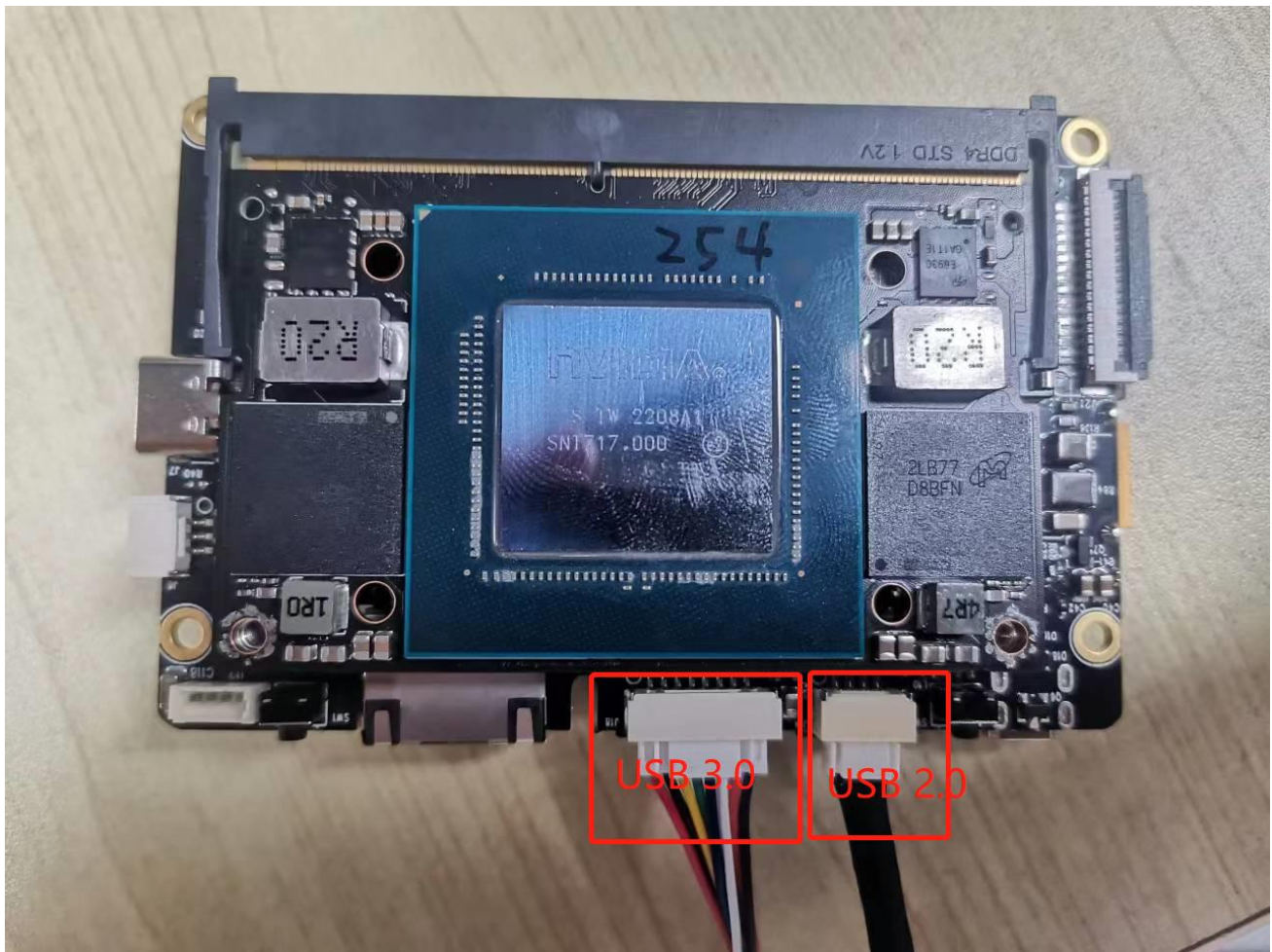


图 3-13

插入 USB 的测试工装，然后插入 U 盘，在系统的/dev 下查看是否有 U 盘设备 sda 或 sdb。HDMI 显示屏显示读有 U 盘表示测试成功。

\$ ls /dev

测试 USB3.0 时输入命令，在最下面一行为最新的打印，如果识别到的是 SuperSpeed 则表示 USB3.0 测试成功。

nvidia@nvidia:~\$ dmesg | grep peed

现象与图 3-7 类似。

注： J18 USB3.0 与原本 311UAV 线序不同们需要将 6 引脚与 7 引脚互调。

3.6 网口测试

1) 用网线分别通过交换机连接板卡的网口与 PC 机如图 3-14 所示

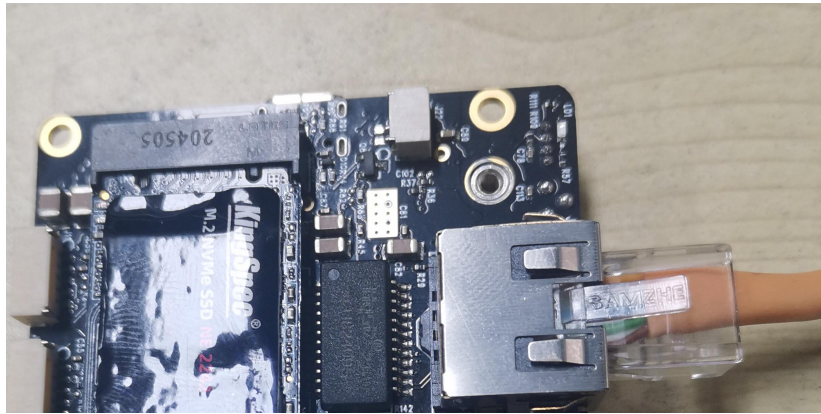


图 3-14

- 2) 使用 ifconfig 查看板卡 ip，如果没有 ip，使用 ifconfig 设置 IP（与 PC 机同网段）。例如如下设置，

```
sudo ifconfig eth0 192.168.1.210
```

- 3) 板卡 ping PC 机 IP，测试网络通信是否正常，输入下面命令

```
ping 192.168.1.47
```

3.7 调试串口测试

- 1) 使用 TTL 电平串口模块，连接板卡 J6 和 PC 机。如图 3-15



图 3-15

- 2) 如果上电之前 PC 机并没有连接调试串口到板卡，根据 PC 机调试串口连接的端口号打开 SecureCRT

并连接。

- 3) 如果上电之前 PC 机连接了调试串口，则能够看到系统启动信息。如果系统启动后连接，敲击回车，可以看到登录信息如图 3-16 所示，输入用户名及密码（nvidia）登录系统。

```
localhost login: nvidia
Password:
Last login: Thu Jun  4 14:02:32 CST 2020 on ttyTCU0
welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.9.140-tegra aarch64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

197 packages can be updated.
130 updates are security updates.

nvidia@localhost:~$
```

图 3-16

3.8 SSD 测试

Orin-NX 直接以 SSD 为系统盘，能起来系统时就是好的。

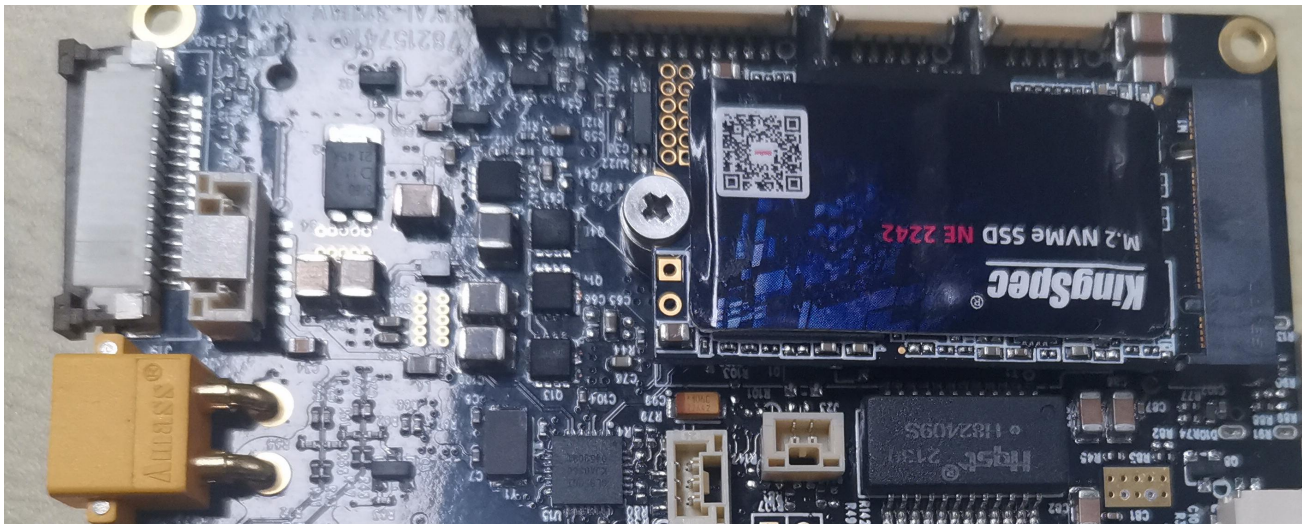


图 4.9-1

3.9 CAN 测试

连接 J2 的 1（CANL）与 2（CANH）脚到 can 盒，如图 3-18

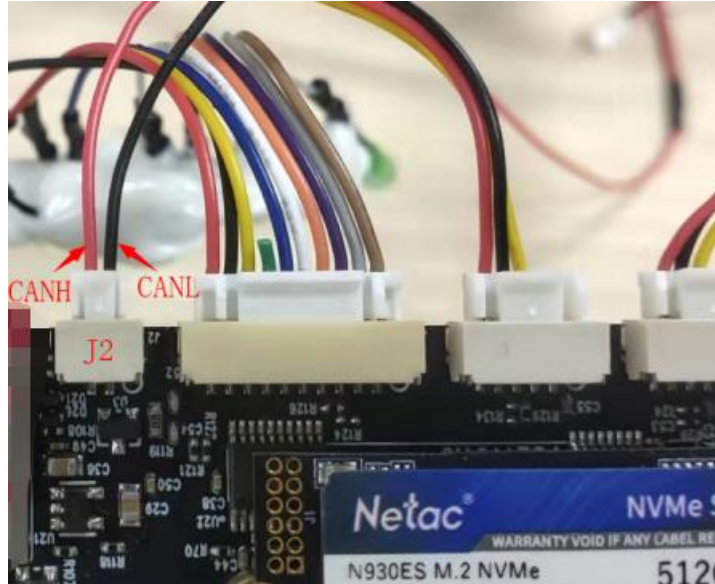


图 3-18

打开 PC 端 CAN 测试软件 CANalyst.exe，点击“打开设备”，设置波特率为 500kbps，如图 3-19 所示，点击“确定”进行连接。

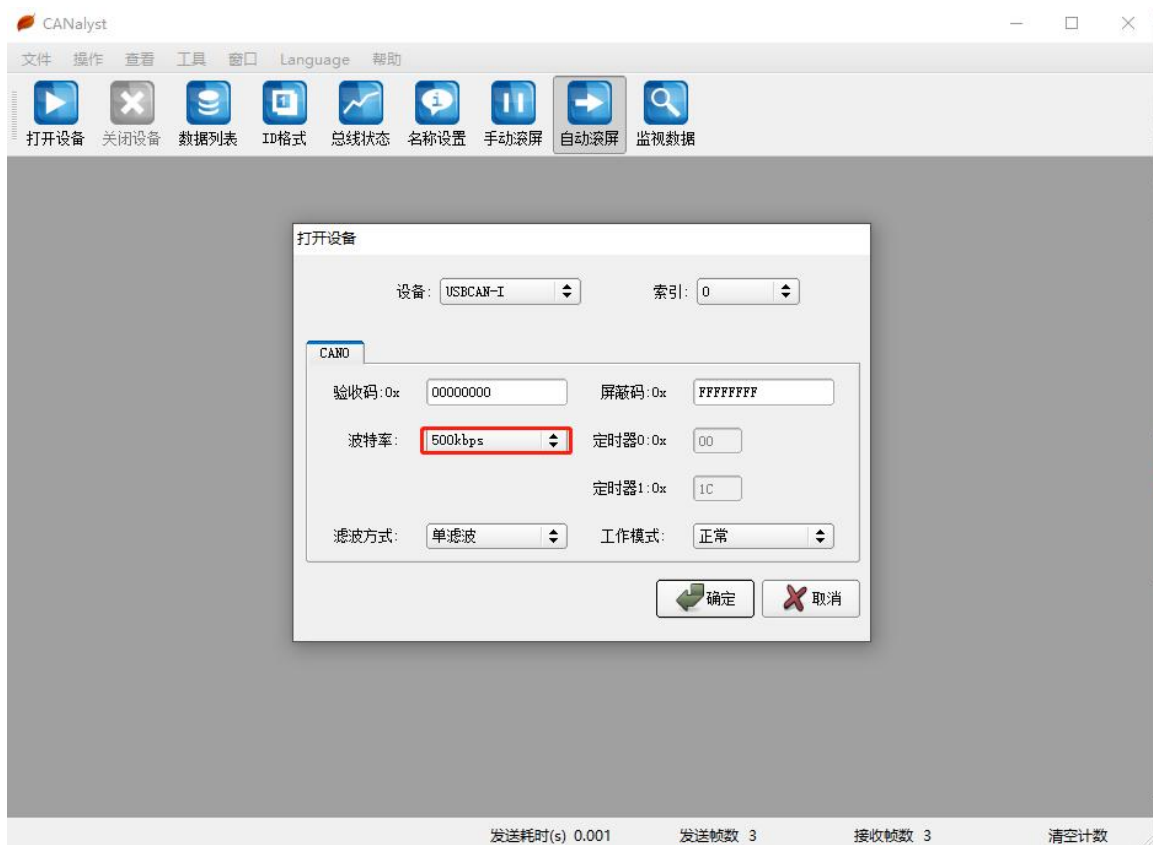


图 3-19

- 执行 ./canDriver.sh 加载 CAN 驱动:

```
$ ./canDriver.sh
```

```
nvidia@localhost:~/carrierTest$ ./canDriver.sh
nvidia@localhost:~/carrierTest$
```

图 3-20

- 执行 CAN 测试程序

```
$ ./canTest
```

- 在 PC 端软件, 点击“发送”按钮发送数据, 同时能够收到板卡发来的数据, 接收到的数据一直为“00 01 02 03 04 05 06 07”。如图 3-21 所示发送接收了 3 组数据 (可以在“数据”的框中改变要发送的数据)。

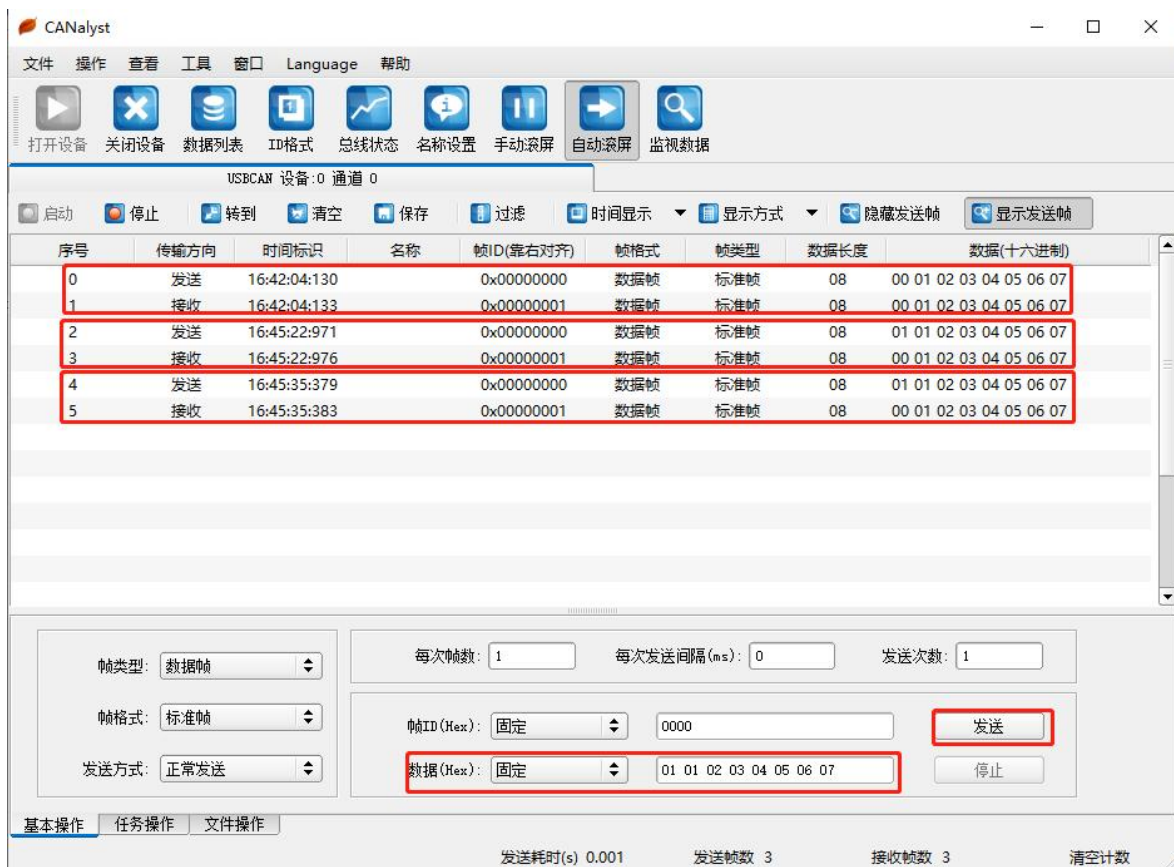


图 3-21

同时调试串口终端测试过程中打印测试次数, 如图 3-22 所示

```
nvidia@localhost:~/carrierTest$ ./canTest
1
2
3
```

图 3-22

3.10 RS232 测试

板卡共 2 个 RS232 串口 J11/J15（TTL 电平），如图 3-23

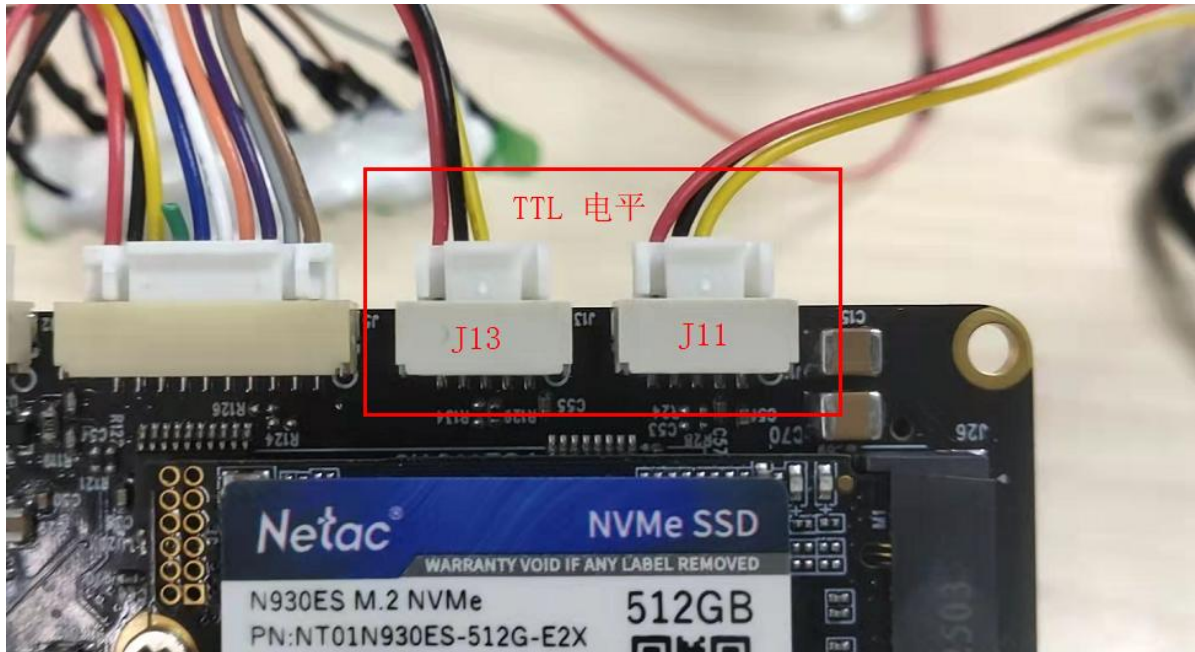


图 3-23

通过 RS232 串口模块（TTL 电平）连接 PC 机与板卡，打开 PC 端串口调试助手。

在板卡上执行串口测试程序如图 3-24

```
$ service nvgetty stop
```

```
$ sudo ./uartTest -a -e
```

```
nvidia@localhost:~/carrierTest$ sudo ./uartTest -a -e
===== Read Com info of Device =====
UART0 - Run Thread with on ...
Open UART0 /dev/ttyTHS0 ---->
UART1 - Run Thread with on ...
Open UART1 /dev/ttyTHS1 ---->
Open success(3) UART0 - RS485/RS232
SetComCfg UART0 [9600-n-8-1] success.
Open success(4) UART1 - RS485/RS232
SetComCfg UART1 [9600-n-8-1] success.
UART0 O#@ UART1 < Recv 5 data.
UART1 > Send 5 data done.
UART1 < Recv 5 data.
UART1 > Send 5 data done.
```

图 3-24

上述提示中本装置中有 2 个 UART 串口，测试配置为 9600-n-8-1，测试程序自动监视串口上的数据，并进行回显；

测试时通过电脑上串口调试助手，向 RS232 串口发送数据，如字符串“123456”；
此时串口终端会显示 UART0 收到数据数，并自动原数据发送返回，打印报文如下：

UART0 < Recv 6 data.

电脑端的 RS232 则会收到返回的数据，即刚发的字符串“123456”；
此时表示 RS232 串口在 9600bps 是收发功能正常。

3.11 Camera 测试

本载板上有 2 个摄像头，插入 imx219 摄像头，对应关如图 3-25 所示：

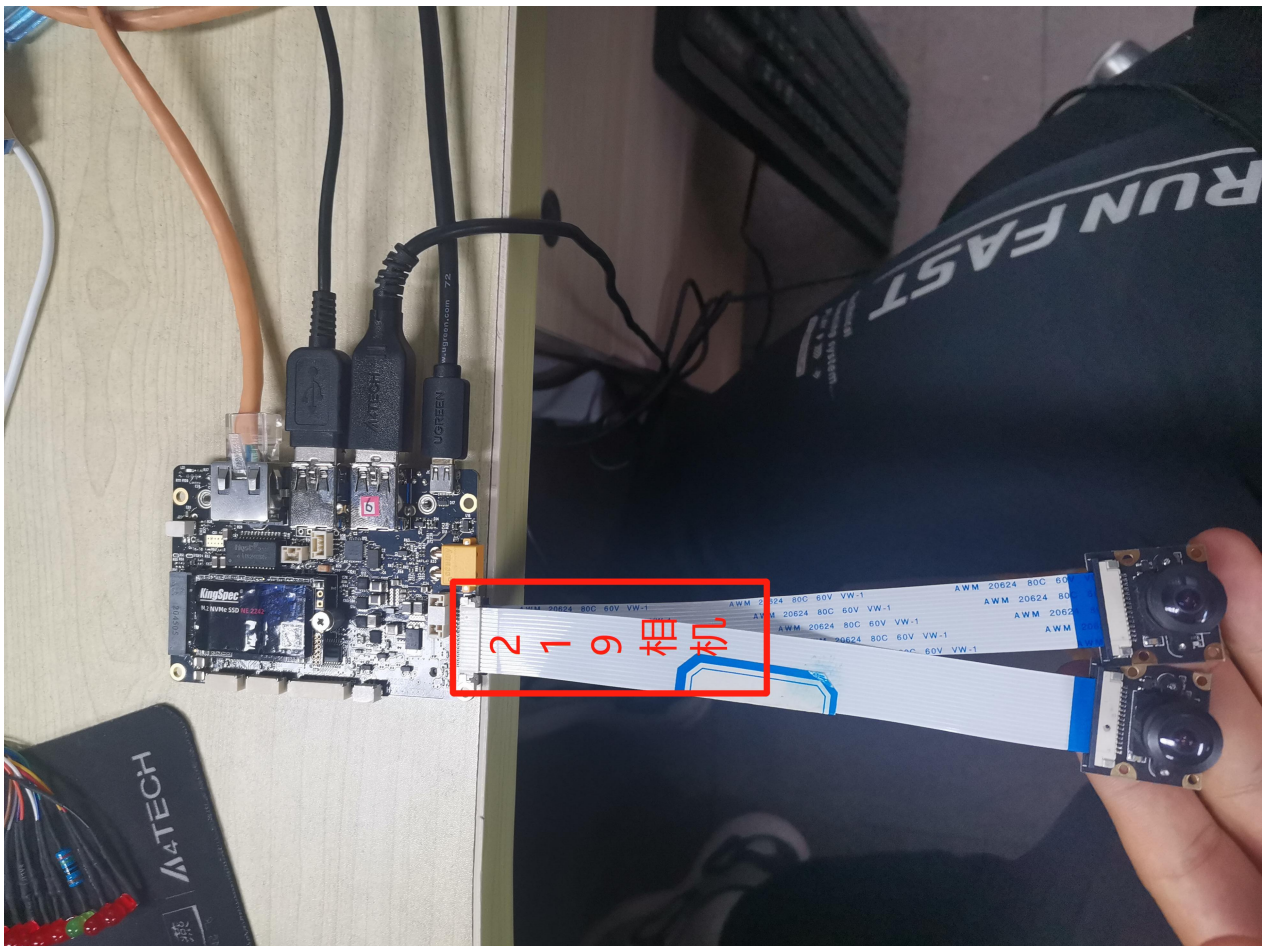


图 3-25

在 HDMI 显示屏 ubuntu 系统中，打开终端，进入 carrierTest 目录，执行 cameraTest.sh 测试 camera。
cameraTest.sh 后面跟参数 0~2，参数 0~2 表示要测试的哪个摄像头，参数 2 表示 2 个摄像头全部测

试。

执行以下指令测试 2 个摄像头，红色标记处指出了当前测试的为哪个摄像头，同时 HDMI 显示屏上弹出当前摄像头采集的画面。当测试完当前摄像头时，输入 **ctrl+c** 结束当前摄像头测试，进入下一个摄像头的测试，直到 2 个摄像头测试完成。

\$./cameraTest.sh 2

```
nvidia@localhost:~/carrierTest$ ./cameraTest.sh 4
4
Setting pipeline to PAUSED ...

Using winsys: x11
Pipeline is live and does not need PREROLL ...
Got context from element 'eglglessink0': gst.egl.EGLDisplay=context, display=(GstEGLDisplay)NULL;
Setting pipeline to PLAYING ...
New clock: GstSystemClock
GST_ARGUS: Creating output stream
CONSUMER: waiting until producer is connected...
GST_ARGUS: Available sensor modes :
GST_ARGUS: 3264 x 2464 FR = 21.000000 fps Duration = 47619048 ; Analog Gain range min 1.000000, max 10.625000; Exposure Range min 13000, max 683709000;
GST_ARGUS: 3264 x 1848 FR = 28.000001 fps Duration = 35714284 ; Analog Gain range min 1.000000, max 10.625000; Exposure Range min 13000, max 683709000;
GST_ARGUS: 1920 x 1080 FR = 29.999999 fps Duration = 33333334 ; Analog Gain range min 1.000000, max 10.625000; Exposure Range min 13000, max 683709000;
GST_ARGUS: 1280 x 720 FR = 59.999999 fps Duration = 16666667 ; Analog Gain range min 1.000000, max 10.625000; Exposure Range min 13000, max 683709000;
GST_ARGUS: 1280 x 720 FR = 120.000005 fps Duration = 8333333 ; Analog Gain range min 1.000000, max 10.625000; Exposure Range min 13000, max 683709000;
GST_ARGUS: Running with following settings:
  camera index = 0
  camera mode = 0
  Output Stream W = 3264 H = 2464
  seconds to Run = 0
  Frame Rate = 21.000000
GST_ARGUS: Setup Complete, Starting captures for 0 seconds
GST_ARGUS: Starting repeat capture requests.
CONSUMER: Producer has connected; continuing.
ERROR: from element 'GstPipeline:pipeline0/GstNvArgusCameraSrc': Internal data stream error.
Additional debug info:
gstbasesrc.c(3055): gst_base_src_loop (): /GstPipeline:pipeline0/GstNvArgusCameraSrc:nvarguscamerasrc0:
streaming stopped, reason error (-5)
EOS on shutdown enabled -- waiting for EOS after Error
waiting for EOS...
```

图 3-26

执行以下命令测试单个摄像头，参数 0~3。例如测试第 0 个摄像头：

\$./cameraTest.sh 0

注：当测试过程中发现某个摄像头没有显示画面时，板卡重新上电，单独测试此摄像头，看是否正确。

摄像头与端口是不绑定的，例如检测到两个摄像头，则摄像头对应 video0~1，不一定是上面接口中的 camera0 与 camera1。如果要确定是哪一个摄像头有问题，根据测试现象有如下分析结果：

- 1) 首先可查看串口调试信息，看摄像头是否识别，如下所示“**imx219 9-0010: board setup failed**”，表示 I2C 识别摄像头出错。这里的“**imx219 9-0010**”与摄像头的对应关系图 4.14-1 已经指出。

nvidia@localhost:~\$ dmesg | grep imx219

```
nvidia@localhost:~$ dmesg | grep imx219
[ 3.106560] ==WAZ [workspace/Nvidia/nvidia_sdk/JetPack_4.4_DP_Linux_DP_JETSON_XAVIER_NX/Linux_for_Tegra/sources/kernel/nvidia/drivers/media/i2c/imx219.c:imx219_probe]696=====
[ 3.106785] imx219 9-0010: teoracam sensor driver:imx219 v2.0.6
[ 13.142961] imx219 9-0010: imx219_board_setup: error during i2c read probe (-110)
[ 13.143226] imx219 9-0010: board setup failed
[ 13.143610] imx219: probe of 9-0010 failed with error -110
[ 13.144006] ==WAZ [workspace/Nvidia/nvidia_sdk/JetPack_4.4_DP_Linux_DP_JETSON_XAVIER_NX/Linux_for_Tegra/sources/kernel/nvidia/drivers/media/i2c/imx219.c:imx219_probe]696=====
[ 13.144705] imx219 10-0010: teoracam sensor driver:imx219 v2.0.6
[ 13.170018] imx219 10-0010: imx219_board_setup: error during i2c read probe (-121)
[ 13.170729] imx219 10-0010: board setup failed
[ 13.170938] imx219: probe of 10-0010 failed with error -121
[ 13.171363] ==WAZ [workspace/Nvidia/nvidia_sdk/JetPack_4.4_DP_Linux_DP_JETSON_XAVIER_NX/Linux_for_Tegra/sources/kernel/nvidia/drivers/media/i2c/imx219.c:imx219_probe]696=====
[ 13.171924] imx219 11-0010: teoracam sensor driver:imx219 v2.0.6
[ 13.196471] imx219 11-0010: imx219_board_setup: error during i2c read probe (-121)
[ 13.196779] imx219 11-0010: board setup failed
[ 13.196983] imx219: probe of 11-0010 failed with error -121
[ 13.197455] ==WAZ [workspace/Nvidia/nvidia_sdk/JetPack_4.4_DP_Linux_DP_JETSON_XAVIER_NX/Linux_for_Tegra/sources/kernel/nvidia/drivers/media/i2c/imx219.c:imx219_probe]696=====
[ 13.197643] imx219 12-0010: teoracam sensor driver:imx219 v2.0.6
[ 13.222569] imx219 12-0010: imx219_board_setup: error during i2c read probe (-121)
[ 13.223151] imx219 12-0010: board setup failed
[ 13.223341] imx219: probe of 12-0010 failed with error -121
```

图 3-27

- 2) 当 camera 识别正确，但当执行 camera 测试时不能显示采集画面（只显示一个框框），则表示为 camera 采集的时钟或数据线上有问题。这时要使用测试单个摄像头方法进行逐个测试。当某个摄像头不能成

功显示时，需要重启板卡在进行下一个摄像头的测试。

3.12 GPIO 测试

- 1) 将 HYAI-310UAV_V10 板卡 J5 连接灯板，如图 3-28

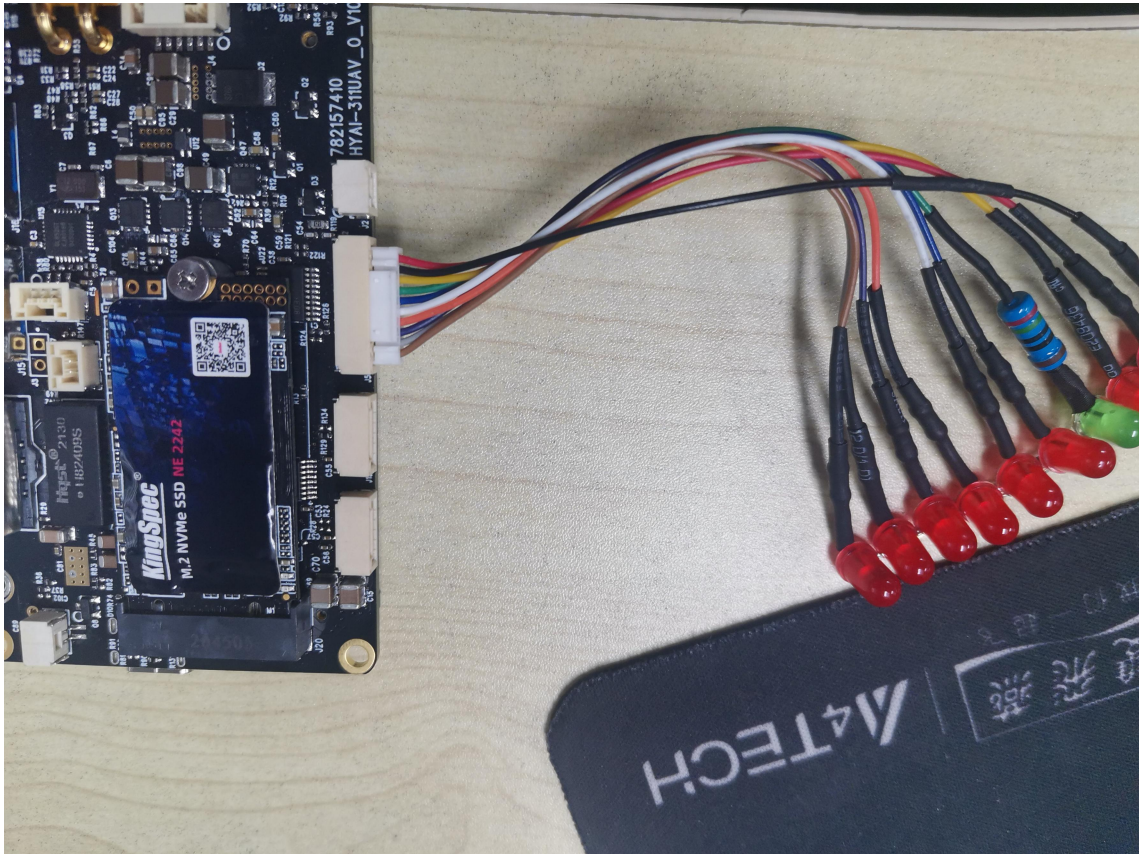


图 3-28

- 2) 在/hzhy/carrierTest 路径执行下面的命令，测试 GPIO，灯板上面 8 个灯都闪烁，表示测试成功。

```
$ ./gpio_311UAV
```

3.13 BMI160 传感器测试（焊 U2 后测此项）

输入下面的命令，如图 3-29 显示，表示测试正常

```
$ i2cdetect -y -r 8
```

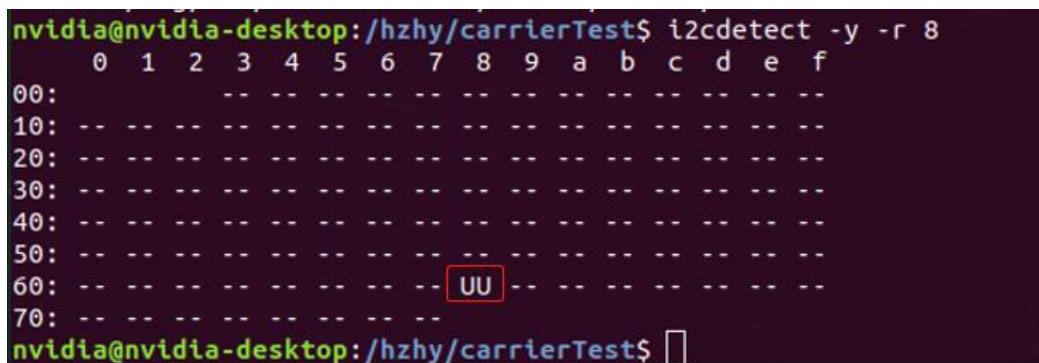


图 3-29

3.14 烧写端口测试

将板卡断电后，使用 Type-C USB 线连接板卡 J7 到 PC 机。

- 1) 按住“强制恢复（FRC）”按钮 J23。
- 2) 给板卡上电
- 3) 在 Windows 系统“设备管理器”中能够看到“APX”设备



图 3-30

- 4) 如果打开着虚拟机，将此设备连接到 ubunut 虚拟机中，执行 lsusb，能够看到 Nvidia 信息

```
hzhy@hzhy-IdeaCentre-GeekPro-14I0B:~/Orin-nx-16g/Linux_for_Tegra$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 006: ID 10c4:ea60 Cygnal Integrated Products, Inc. CP210x UART Bridge / myAVR mySmartUSB light
Bus 001 Device 005: ID 05e3:0610 Genesys Logic, Inc. 4-port hub
Bus 001 Device 004: ID 05e3:0610 Genesys Logic, Inc. 4-port hub
Bus 001 Device 003: ID 05e3:0610 Genesys Logic, Inc. 4-port hub
Bus 001 Device 008: ID 0955:7323 NVidia Corp.
Bus 001 Device 002: ID 0bda:c123 Realtek Semiconductor Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

图 3-31

3.15 HYAI-311UW 测试

3.15.1 电源测试

将底板 J3 连接跳线帽，不连接核心板，将 HYAI-311UW 小板的 J6 与底板 J4 连接，J1 与底板 J18 连接，之后给板卡 12V 上电，用万用表测量 HYAI-311UW 小板的测试点（3.3V）如图 3-32 所示，正确无误

后再进行 WIFI、4G 的测试。（测试底板电源时，需测量正确后在连接核心板和 4G 模块。）



图 3-32

3.15.2 4G 测试

- 1) 将 HYAI-311UW 扩展板插入 4G 模块（EC20），并连接 4G 天线，插入 SIM 卡，如图 3-33

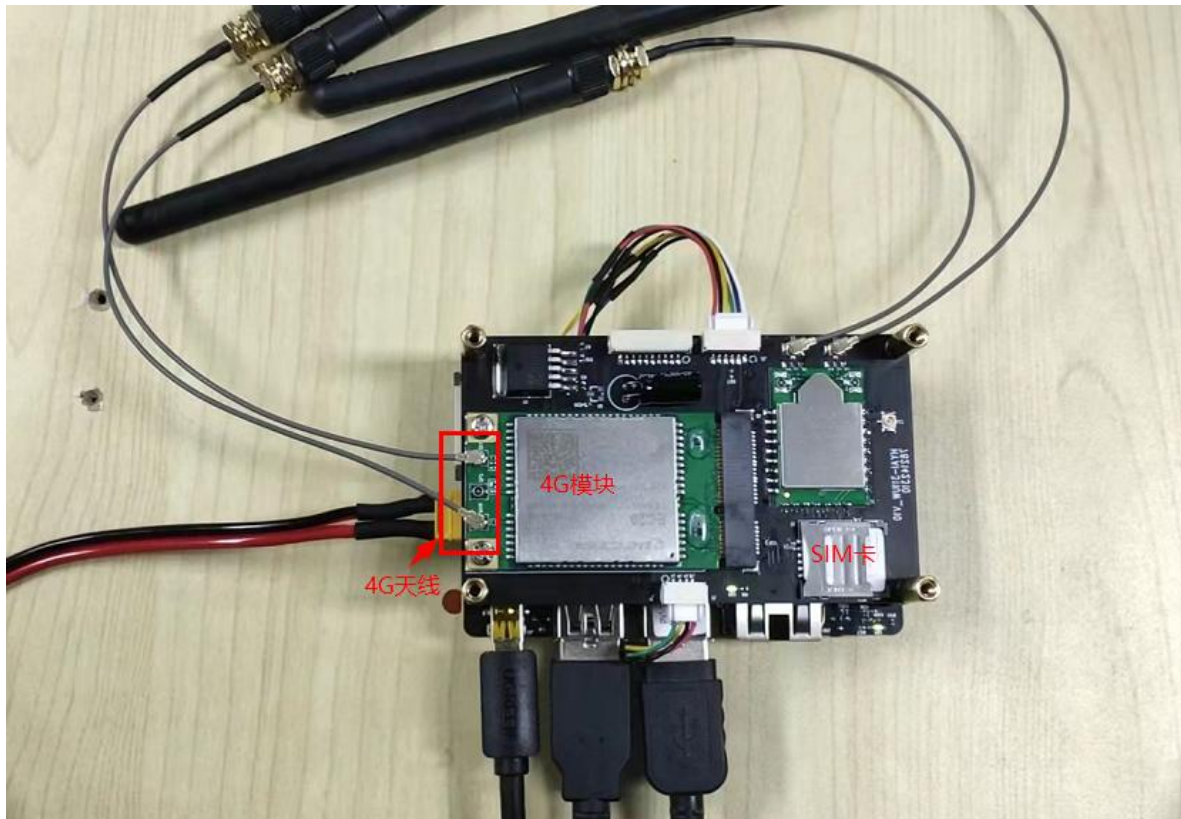


图 3-33

2) 在 /hzhy/carrierTest 目录下运行下面拨号脚本，如 图 3-34

\$ sudo ./4g.sh

```
nvidia@nvidia-desktop:/hzhy/carrierTest$ sudo ./4g.sh
[sudo] password for nvidia:
nvidia@nvidia-desktop:/hzhy/carrierTest$ [04-12_18:43:44:873] QConnectManager_Linux_V1.6.3.1
[04-12_18:43:44:873] Find /sys/bus/usb/devices/1-3.2 idVendor=0x2c7c idProduct=0x125, bus=0x001, dev=0x003
[04-12_18:43:44:874] Auto find qmichannel = /dev/cdc-wdm0
[04-12_18:43:44:874] Auto find usbnet_adapter = wwan0
[04-12_18:43:44:874] netcard driver = qmi_wwan_q, driver version = V1.2.0.14
[04-12_18:43:44:874] Modem works in QMI mode
[04-12_18:43:44:892] cdc_wdm_fd = 7
[04-12_18:43:44:970] Get_clientWDS = 7
[04-12_18:43:45:002] Get_clientDMS = 1
[04-12_18:43:45:034] Get_clientNAS = 2
[04-12_18:43:45:066] Get_clientUIM = 1
[04-12_18:43:45:098] Get_clientWDA = 1
[04-12_18:43:45:130] requestBaseBandVersion EC20CEFALGR06A03M1G
[04-12_18:43:45:258] requestGetSIMStatus SIMStatus: SIM_READY
[04-12_18:43:45:291] requestGetProfile[1] cmnet///0/IPV4V6
[04-12_18:43:45:322] requestRegistrationState2 MCC: 460, MNC: 0, PS: Attached, DataCap: LTE
[04-12_18:43:45:354] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED
[04-12_18:43:45:354] ifconfig wwan0 0.0.0.0
[04-12_18:43:45:358] ifconfig wwan0 down
[04-12_18:43:45:418] requestSetupDataCall WdsConnectionIPv4Handle: 0xe188ef50
[04-12_18:43:45:546] ifconfig wwan0 up
[04-12_18:43:45:549] busybox udhcpc -f -n -q -t 5 -i wwan0
udhcpc: started, v1.30.1
udhcpc: sending discover
udhcpc: sending select for 10.65.57.73
udhcpc: lease of 10.65.57.73 obtained, lease time 7200
[04-12_18:43:45:671] /etc/udhcpc/default.script: Resetting default routes
SIOCDLRT: No such process
[04-12_18:43:45:673] /etc/udhcpc/default.script: Adding DNS 221.179.155.209
[04-12_18:43:45:674] /etc/udhcpc/default.script: Adding DNS 221.179.155.193
```

图 3-34

3) 执行 `ping -I wwan0 baidu.com`, 如图 3-35, 可以 ping 通表示测试通过。

```
nvidia@nvidia-desktop:~$ ping -I wwan0 baidu.com
PING baidu.com (110.242.68.66) from 10.65.57.73 wwan0: 56(84) bytes of data.
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=1 ttl=48 time=64.9 ms
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=2 ttl=48 time=37.8 ms
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=3 ttl=48 time=40.8 ms
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=4 ttl=48 time=41.0 ms
64 bytes from 110.242.68.66 (110.242.68.66): icmp_seq=5 ttl=48 time=39.5 ms
```

图 3-35

3.15.3 WIFI/蓝牙测试

将 HYAI-311UW_V10 扩展板连接 WIFI 天线, 如图 3-36 所示。

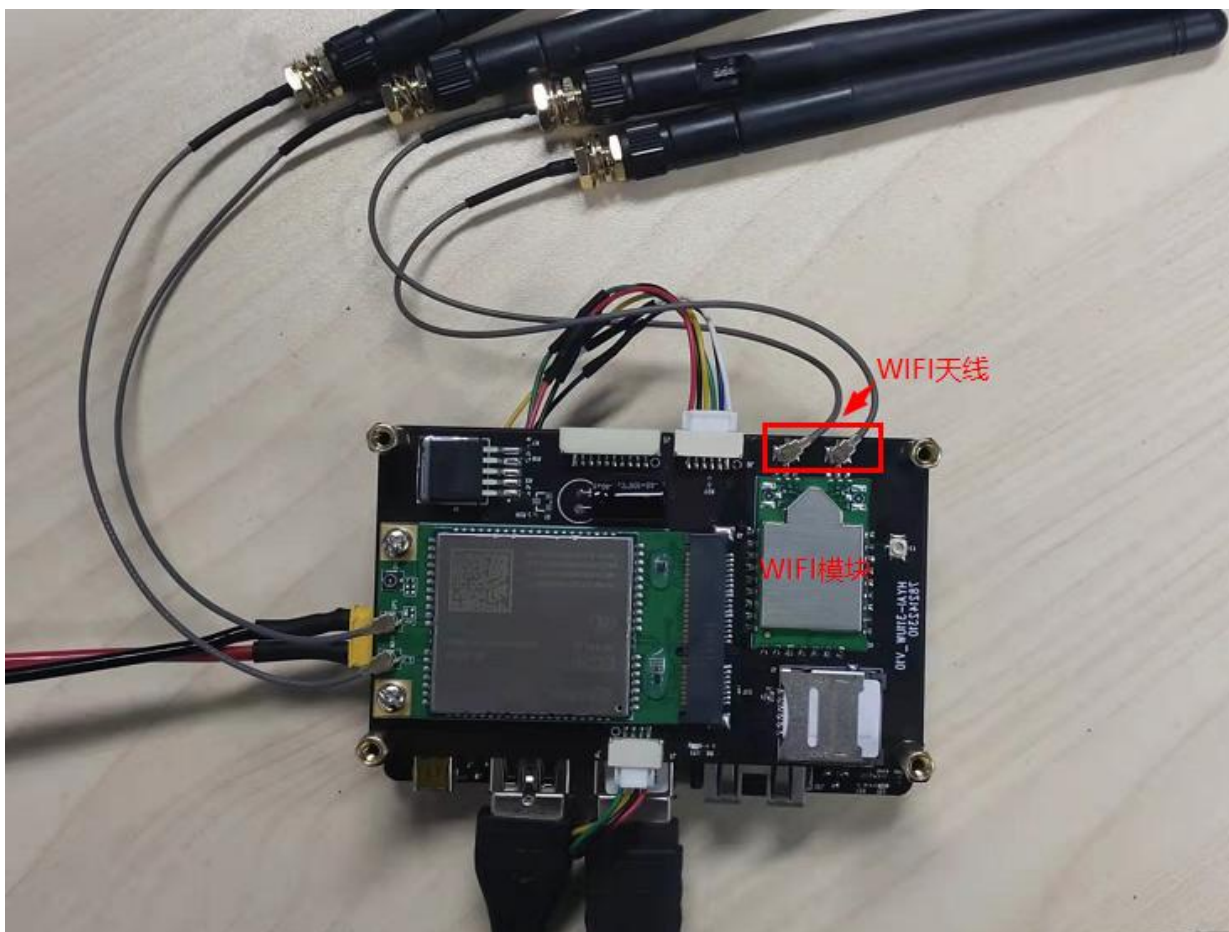


图 3-36

HDMI 显示屏上，输入用户密码进入系统。

➤ WIFI 测试

拔掉载板网线，左键点击屏幕右上角的网络连接标志，下拉显示检测到的 wifi 信号，断开 4G 连接。点击 WIFI 网络名称进行连接，输入 WIFI 密码，可以看到右上角网络连接显示变为了 wifi 连接的样式。

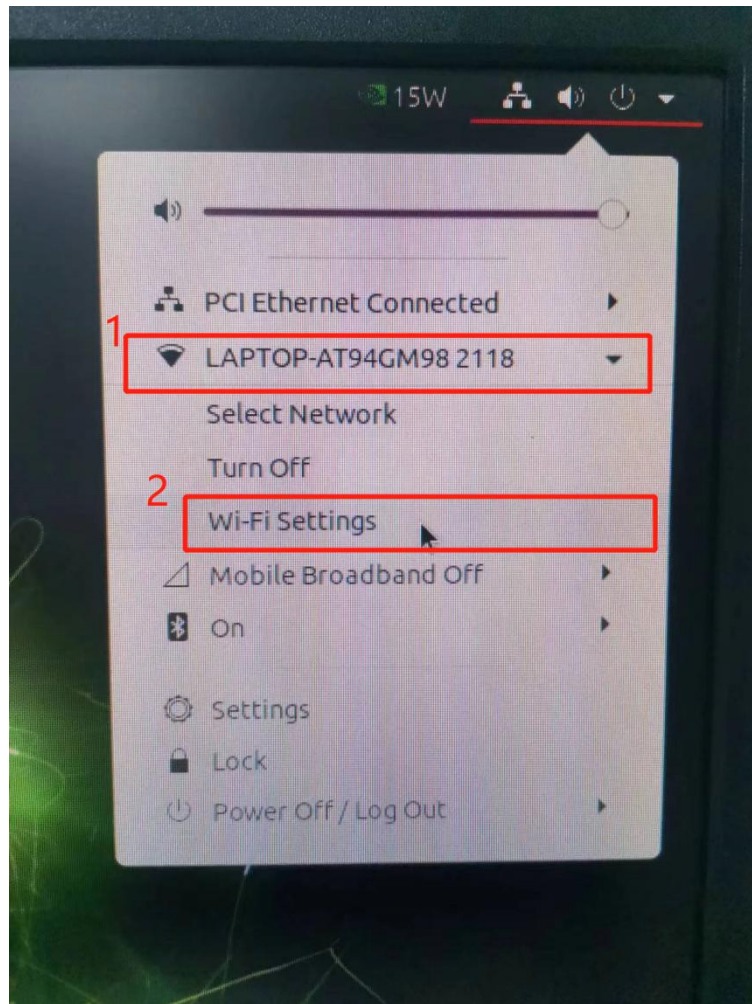


图 3-37

执行 `ping -I wlan0 baidu.com` 能够 ping 通表示测试成功，如图 3-38

```
nvidia@nvidia-desktop:~$ ping -I wlan0 baidu.com
PING baidu.com (39.156.66.10) from 192.168.137.145 wlan0: 56(84) bytes of data.
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=1 ttl=50 time=10.2 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=2 ttl=50 time=68.7 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=3 ttl=50 time=9.29 ms
```

图 3-38

➤ 蓝牙测试

1) 测试之前，先连接蓝牙天线，如图 3-39

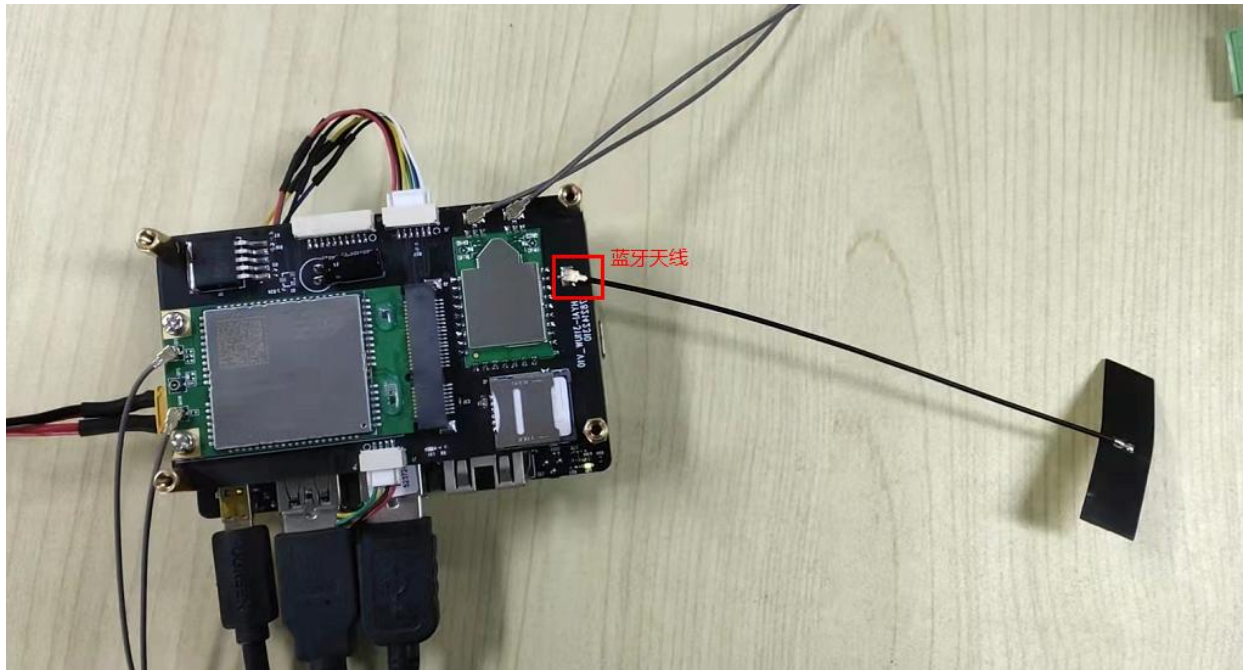


图 3-39

2) 点击屏幕左边设置，在弹出的窗口中点击“Bluetooth”。如图 3-40

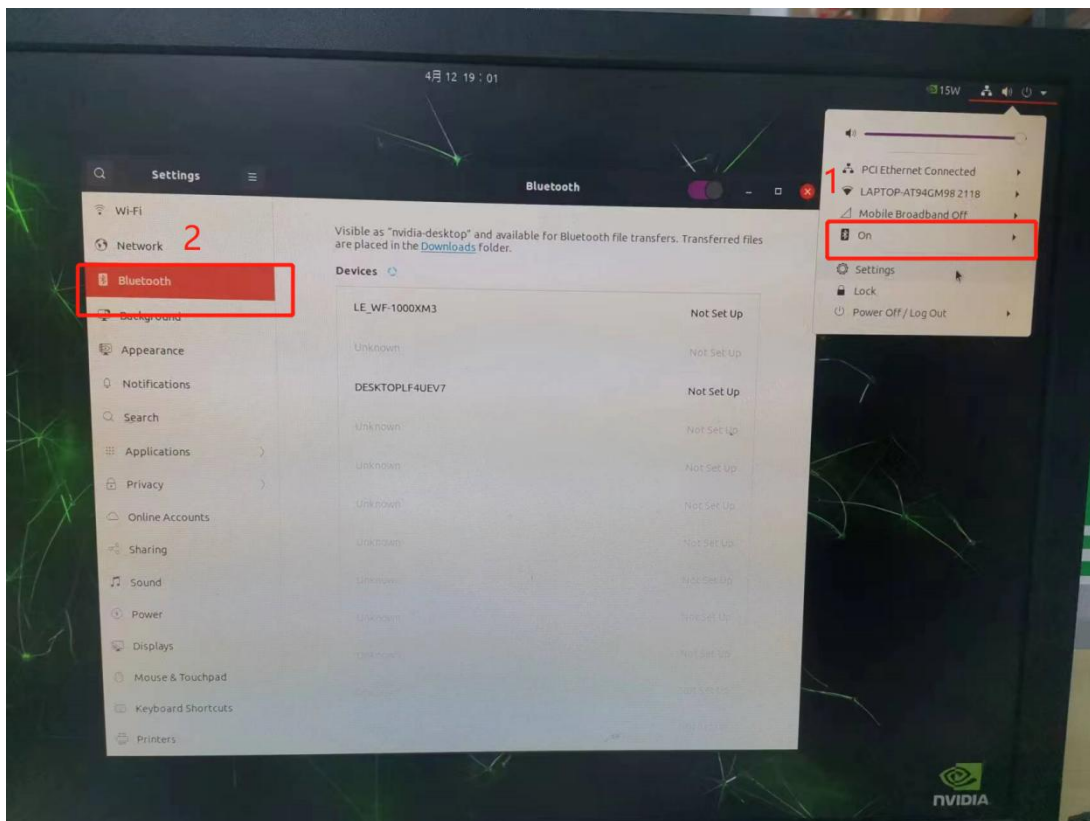


图 3-40

3) 屏幕中显示配对信息，点击要连接设备进行配对（手机端也点击“配对”）。

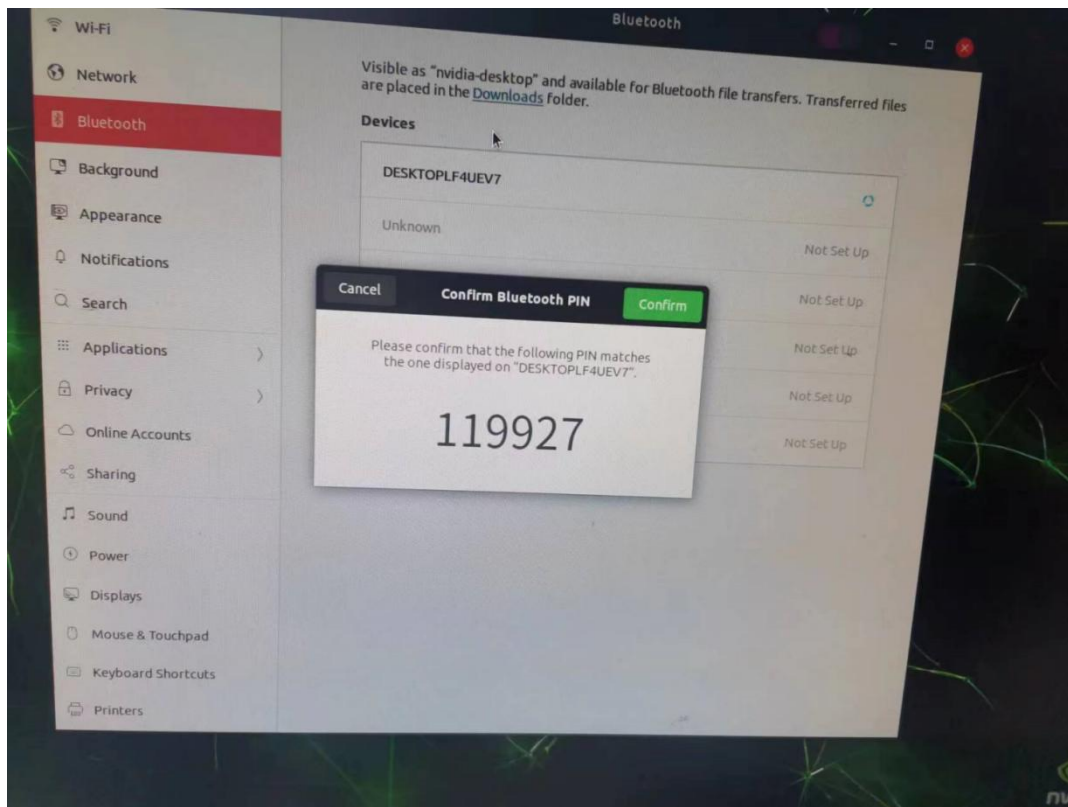


图 3-41

4) 配对成功后，屏幕对话框中显示“Disconnected”。

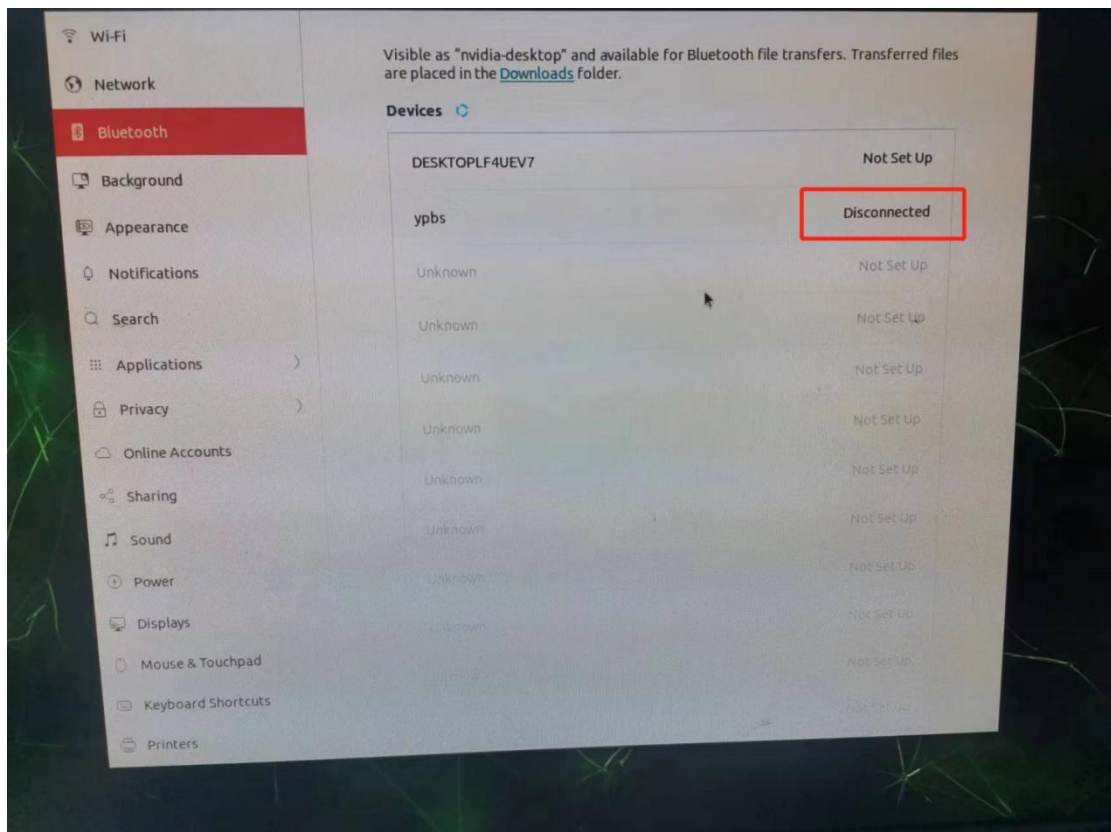
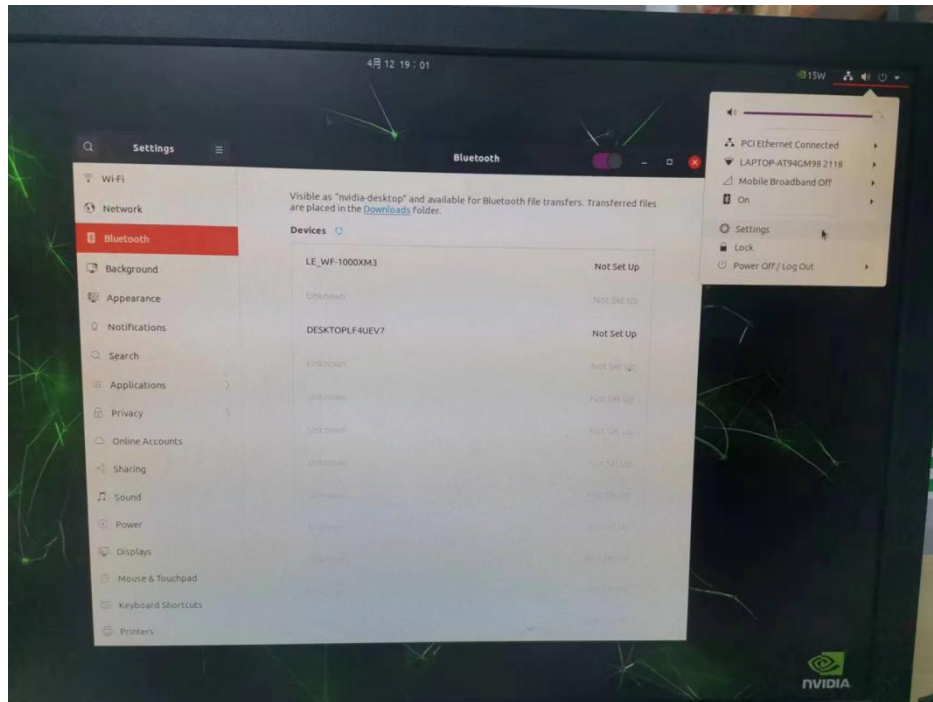


图 3-42

5) 点击右上角蓝牙图标，显示为“ON”表示蓝牙正常。测试通过。



3.16 HYAI-3115g 测试

针对 HYAI-311UAV-Orin_NX_V20 版本升级，增加了 5G 的功能，因此增加 HYAI-3115g 扩展板测试项

3.16.1 电源测试

将底板 J3 连接跳线帽，不连接核心板，将 HYAI-3115g 小板的 J4 与底板 J5 连接，J1 与底板 J18 连接，之后给板卡 12V 上电，用万用表测量 HYAI-3115g 小板的测试点（3.7V、5V）如图 3-43 所示，正确无误后再进行 5G 的测试。（测试底板电源时，需测量正确后在连接核心板和 5G 模块。）

注：J18 的线序与原本 311UAV 的线序不同，所以工装不统统用，参考用户手册的 J18 线序定义修改工装才可以使用，J18 的 6 引脚和 7 引脚互调即可。

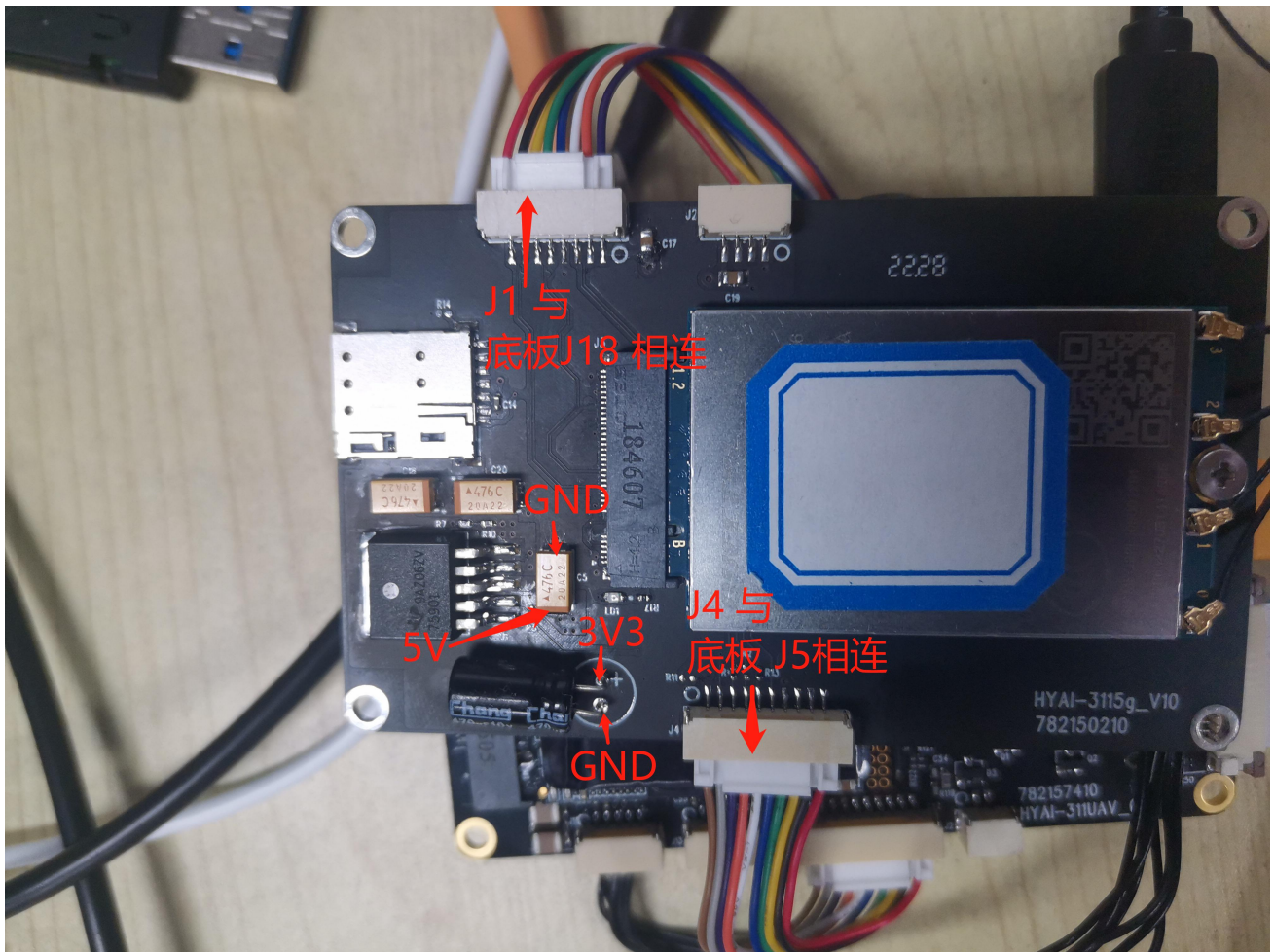


图 3-43

3.16.2 5G 测试

- 1) 将 HYAI-3115g 扩展板插入 5G 模块 (U-CN)，并连接 5G 天线，插入 SIM 卡，图 3-44

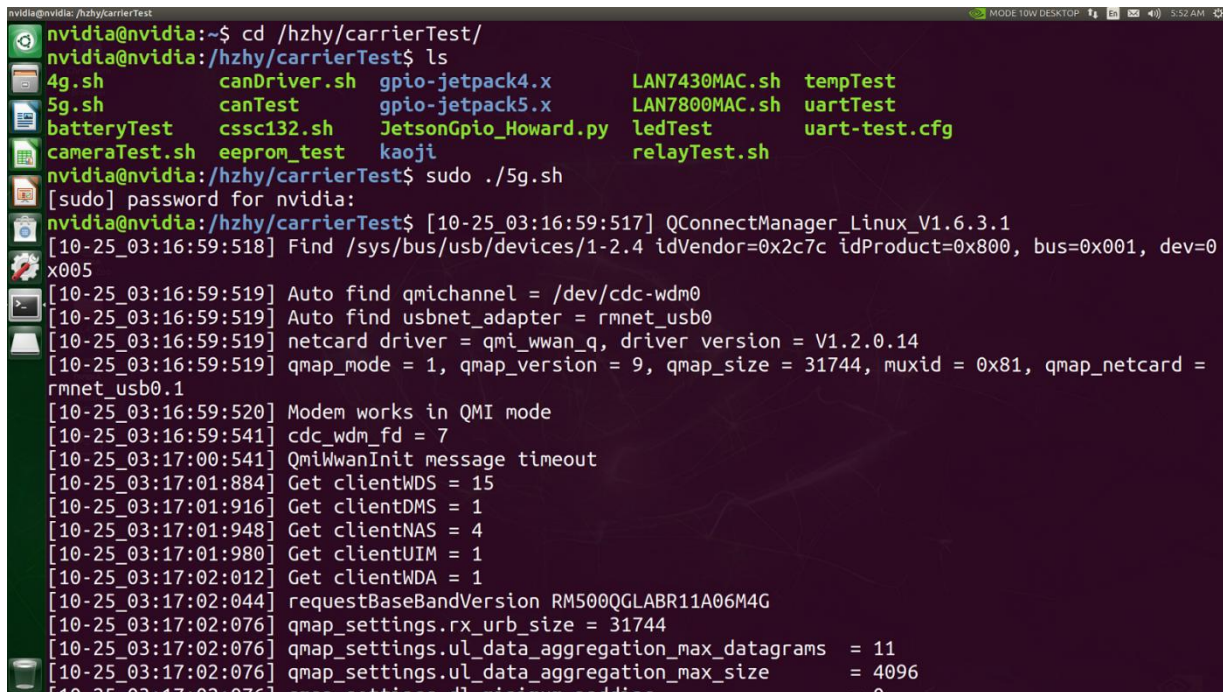


图 3-44

- 2) 载板连接 HDMI 显示屏，打开终端，进入测试程序目录，执行 5G 测试程序，如图 3-45

```
nvidia@nvidia:~$ cd /hzhy/carrierTest
```

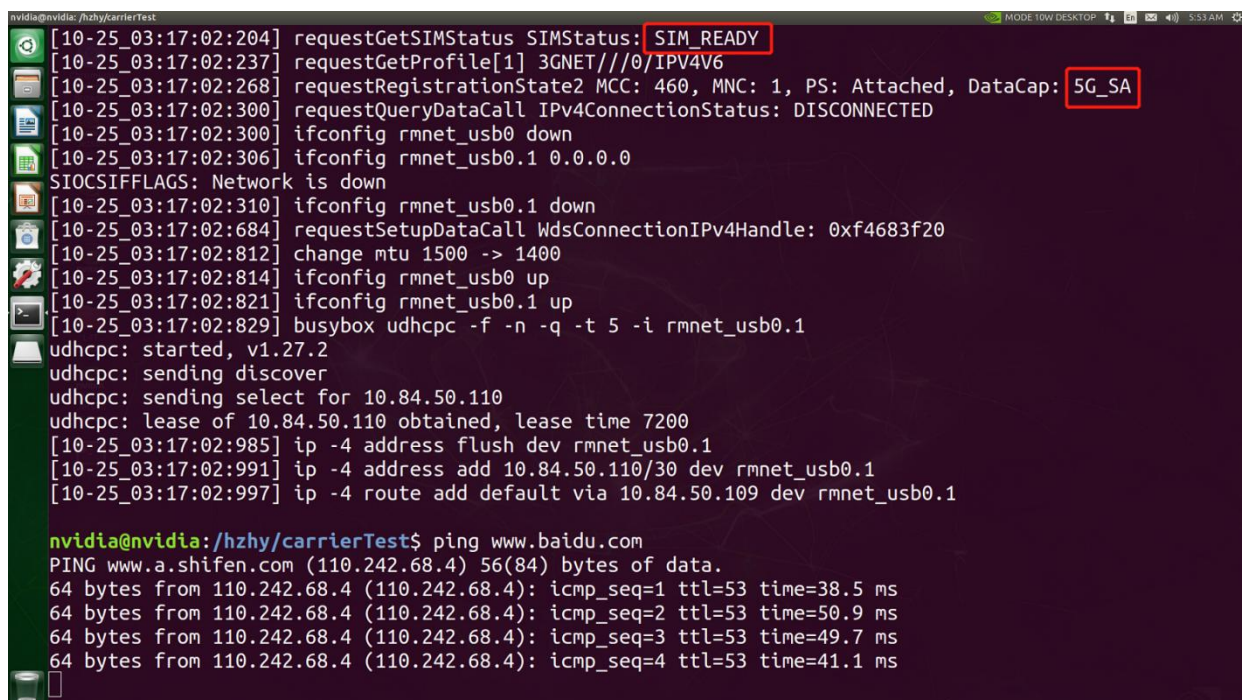
```
nvidia@nvidia:/hzhy/carrierTest$ sudo ./5g.sh
```



```
nvidia@nvidia:~$ cd /hzhy/carrierTest/
nvidia@nvidia:/hzhy/carrierTest$ ls
4g.sh          canDriver.sh  gpio-jetpack4.x  LAN7430MAC.sh  tempTest
5g.sh          canTest       gpio-jetpack5.x  LAN7800MAC.sh  uartTest
batteryTest    cssc132.sh    JetsonGpio_Howard.py  ledTest        uart-test.cfg
cameraTest.sh  eeprom_test   kaoji            relayTest.sh
nvidia@nvidia:/hzhy/carrierTest$ sudo ./5g.sh
[sudo] password for nvidia:
nvidia@nvidia:/hzhy/carrierTest$ [10-25_03:16:59:517] QConnectManager_Linux_V1.6.3.1
[10-25_03:16:59:518] Find /sys/bus/usb/devices/1-2.4 idVendor=0x2c7c idProduct=0x800, bus=0x001, dev=0x005
[10-25_03:16:59:519] Auto find qmichannel = /dev/cdc-wdm0
[10-25_03:16:59:519] Auto find usbnet_adapter = rmnet_usb0
[10-25_03:16:59:519] netcard driver = qmi_wwan_q, driver version = V1.2.0.14
[10-25_03:16:59:519] qmap_mode = 1, qmap_version = 9, qmap_size = 31744, muxid = 0x81, qmap_netcard = rmnet_usb0.1
[10-25_03:16:59:520] Modem works in QMI mode
[10-25_03:16:59:541] cdc_wdm_fd = 7
[10-25_03:17:00:541] QmiWwanInit message timeout
[10-25_03:17:01:884] Get clientWDS = 15
[10-25_03:17:01:916] Get clientDMS = 1
[10-25_03:17:01:948] Get clientNAS = 4
[10-25_03:17:01:980] Get clientUI = 1
[10-25_03:17:02:012] Get clientWDA = 1
[10-25_03:17:02:044] requestBaseBandVersion RM500QGLABR11A06M4G
[10-25_03:17:02:076] qmap_settings.rx_urb_size = 31744
[10-25_03:17:02:076] qmap_settings.ul_data_aggregation_max_datagrams = 11
[10-25_03:17:02:076] qmap_settings.ul_data_aggregation_max_size = 4096
[10-25_03:17:02:076] qmap_settings.dl_maximum_padding = 0
```

图 3-45

- 3) 可以看到 SIM 卡已经正确识别，信号为 5G，如图 3-46，可以 ping 通 baidu 表示测试通过，测试之前请确认除 5G 外无其他网络连接。或者直接 ping -I 指定的网卡。



```
nvidia@nvidia:/hzhy/carrierTest$ [10-25_03:17:02:204] requestGetSIMStatus SIMStatus: SIM_READY
[10-25_03:17:02:237] requestGetProfile[1] 3GNET///0/IPV4V6
[10-25_03:17:02:268] requestRegistrationState2 MCC: 460, MNC: 1, PS: Attached, DataCap: 5G_SA
[10-25_03:17:02:300] requestQueryDataCall IPV4ConnectionStatus: DISCONNECTED
[10-25_03:17:02:300] ifconfig rmnet_usb0 down
[10-25_03:17:02:306] ifconfig rmnet_usb0.1 0.0.0.0
[10-25_03:17:02:310] SIOCSIFFLAGS: Network is down
[10-25_03:17:02:310] ifconfig rmnet_usb0.1 down
[10-25_03:17:02:684] requestSetupDataCall WdsConnectionIPv4Handle: 0xf4683f20
[10-25_03:17:02:812] change mtu 1500 -> 1400
[10-25_03:17:02:814] ifconfig rmnet_usb0 up
[10-25_03:17:02:821] ifconfig rmnet_usb0.1 up
[10-25_03:17:02:829] busybox udhcpc -f -n -q -t 5 -i rmnet_usb0.1
udhcpc: started, v1.27.2
udhcpc: sending discover
udhcpc: sending select for 10.84.50.110
udhcpc: lease of 10.84.50.110 obtained, lease time 7200
[10-25_03:17:02:985] ip -4 address flush dev rmnet_usb0.1
[10-25_03:17:02:991] ip -4 address add 10.84.50.110/30 dev rmnet_usb0.1
[10-25_03:17:02:997] ip -4 route add default via 10.84.50.109 dev rmnet_usb0.1

nvidia@nvidia:/hzhy/carrierTest$ ping www.baidu.com
PING www.a.shifen.com (110.242.68.4) 56(84) bytes of data.
64 bytes from 110.242.68.4 (110.242.68.4): icmp_seq=1 ttl=53 time=38.5 ms
64 bytes from 110.242.68.4 (110.242.68.4): icmp_seq=2 ttl=53 time=50.9 ms
64 bytes from 110.242.68.4 (110.242.68.4): icmp_seq=3 ttl=53 time=49.7 ms
64 bytes from 110.242.68.4 (110.242.68.4): icmp_seq=4 ttl=53 time=41.1 ms
```

图 3-46