

# EAXVA04 使用手册

## 目录

EAXVA04 使用手册.....	1
一、 产品介绍 .....	2
二、 系统框图 .....	3
2.1 系统框图 .....	3
2.2 端口介绍 .....	3
三、 快速开始 .....	4
3.1 提前准备 .....	4
3.2 基础知识 .....	5
3.3 使用设备 .....	5
四、 软件介绍 .....	5
3.1 系统信息 .....	5
3.2 默认安装库.....	5
3.3 查看系统版本信息 .....	6
四、Xavier 端功能使用 .....	7
4.1 RS232 以及 RS485 的使用.....	7
4.2 CAN 使用 .....	9
4.3 普通网口使用 .....	11
4.4 车载以太网使用.....	13
4.5 摄像头使用 .....	14
4.6 固态硬盘的使用.....	17
4.7 PPS 功能 .....	19
4.8 风扇调速 .....	19
五、注意事项.....	20

# 一、 产品介绍

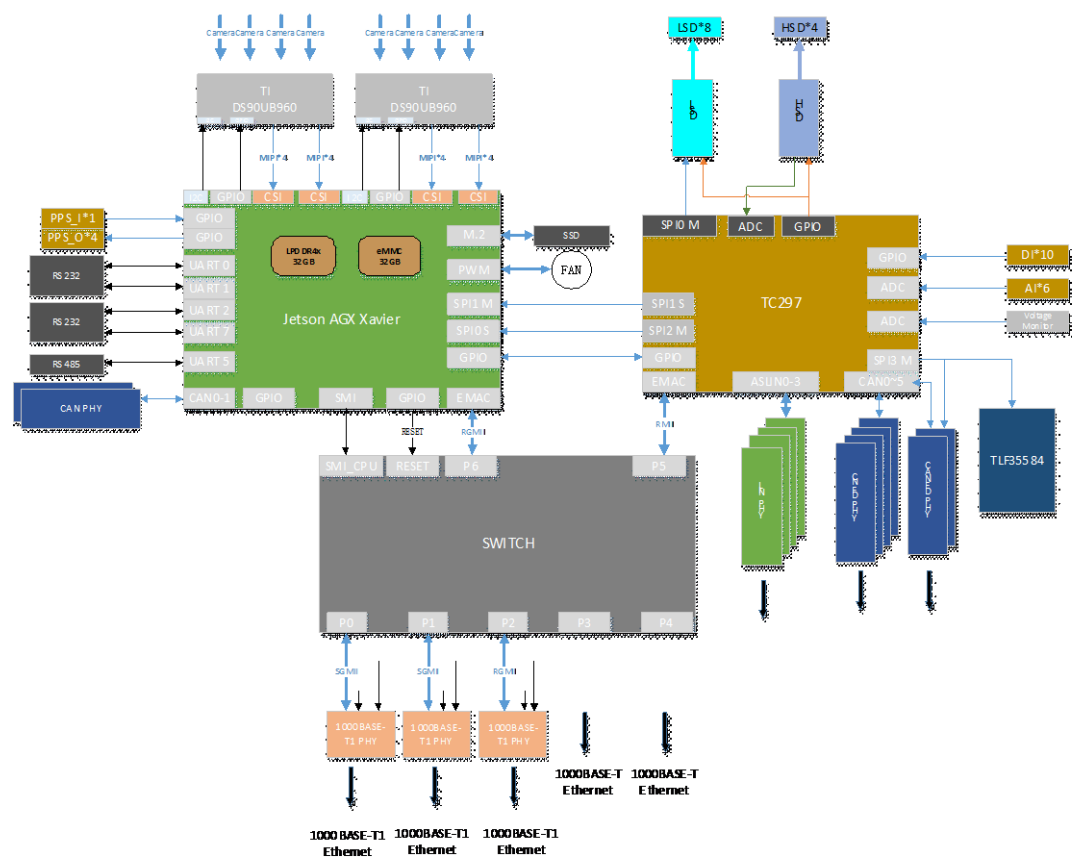
## 产品描述

优控 EAXVA04 控制器是一款基于 NVIDIA JETSON XAVIER 处理器、面向智能边缘计算应用场景的解决方案系统。

EAXVA04 控制器 XAVIER 端具有千兆网口、CAN 总线、RS232、RS485、M.2 NVME 硬盘、车载以太网接口等丰富的外设接口，支持常见的激光雷达、网络相机、FPDLINK 接口相机、毫米波雷达、超声波雷达等传感器的接入；支持 1T 容量内置固态硬盘，用于存储数据；EAXVA04 控制器面向复杂恶劣的工业场景设计：防护等级 IP67；满载运行可工作在-25~70℃ 的环境温度中。

## 二、系统框图

### 2.1 系统框图



### 2.2 端口介绍

接口类型	数量	功能	内部芯片
M.2 KEY M	1	扩展存储	SOC
摄像头接口	8	FPDlink III	SOC
千兆普通以太网	2	100BASE-T/1000BASE-T 标准	交换机
千兆车载以太网	3	100Base-T1/1000Base-T1	交换机
USB	1	1 路 USB TypeA	SOC

HDMI	1		SOC
RS232	4	其中 1 路用于 Debug	SOC
RS485	1		SOC
CAN	2		SOC
PPS_IN	1		SOC
PPS_OUT	4		SOC
CANFD	6	2 路具备特定帧唤醒	MCU
LIN	4	不需要唤醒功能	MCU
数字输入	10	默认配置，5 路高有效（其中 2 路硬 线唤醒）， 3 路低有效, 2 路 IPWM	MCU
模拟输入	6	默认配置，2 路 5V 电压型 2 路 36V 电压型，2 路电阻型	MCU
数字低边输出	8	8 路 @ 250 mA，	MCU
数字高边输出	4	4 路 @ 1 A，	MCU
5V 传感器电源	2	最大电流 200mA	MCU

## 三、快速开始

### 3.1 提前准备

使用本设备之前，请准备以下几项

- 稳定的供电电源，12VDC/ 10A min
- USB 转 RS-232
- 笔记本电脑

### 3.2 基础知识

如果你是 Linux 初学者，也许先学习一些有关 Linux 命令行工具的快速使用教程会对你很有帮助，这里有两篇很好的教程：[中文教程](#)，[英文教程](#)。

### 3.3 使用设备

#### 1. 连接

将设备的正负极连接直流电源，通过 USB 转 RS-232，将设备的 RS232-3 连接到电脑，确保电脑能够正常使用串口设备。

#### 2. 配置

配置串口：波特率 115200，8 个数据位，无奇偶校验，1 个停止位。

使用 Putty 或者 Minicom，打开串口

#### 3. 启动

打开设备的 KeyOn 开关，启动设备电源，设备首先启动 Bootloader，然后运行 Linux 系统，通过串口终端能够看到系统正常启动，之后可以登录，默认用户名：nvidia，密码：nvidia。

## 四、 软件介绍

### 3.1 系统信息

登录信息

用户名	密码	权限
nvidia	nvidia	管理员
root	nvidia	超级管理员，不可从界面登录

### 3.2 默认安装库

软件名称	版本信息	
Jetpack	jetpack 4.2.1	jetpack 4.4

CUDA	10.0.326	10.2
cuDNN	7.5.0.56	8.0
TensorRT	5.1.6.1	7.1
OpenCV	3.3.1	4.1.1
TensorFlow	1.14	
VisionWorks	1.6	
Container	0.9.0	
Multimedia_API	32.2	32.4
deepstream	1.14	5.0
VPI		0.3.7

### 3.3 查看系统版本信息

打开终端输入命令 `cat /etc/EAXVA04.m(cat /etc/EAXVA04_VERSION.m)` 查看 Xavier 镜像的版  
号

```
nvidia@EAXVA04:~$ cat /etc/EAXVA04_VERSION.m
function [EAXVA04_IMG, JetPack, L4T] = EAXVA04_VERSION()

EAXVA04_IMG='2010';
JetPack='4.2.1';
L4T='32.2';
CUDA='10.0';
cuDNN='7.5';
TensorRT='5.1';
OpenCV='3.3.1';
VisionWorks='1.6';
Container='0.9.0';
Multimedia_API='32.2';
DeepStream='4.0';
TensorFlow='1.14';

end
nvidia@EAXVA04:~$
```

nvidia 计算库版本: `jetson_release`

```
nvidia@EAXVA04:~$ jetson_release
- NVIDIA Jetson AGX Xavier [16GB]
* Jetpack 4.2.1 [L4T 32.2.0]
* NV Power Mode: MAXN - Type: 0
* jetson_stats.service: active
- Libraries:
* CUDA: 10.0.326
* cuDNN: 7.5.0.56
* TensorRT: 5.1.6.1
* Visionworks: 1.6.0.500n
* OpenCV: 3.3.1 compiled CUDA: NO
* VPI: NOT_INSTALLED
* Vulkan: 1.1.70
nvidia@EAXVA04:~$
```

kernel 版本信息 : cat /proc/version

```
nvidia@EAXVA04:~$ cat /proc/version
Linux version 4.9.140-tegra (root@li) (gcc version 7.3.1 20180425 [li
naro-7.3-2018.05 revision d29120a424ecfbc167ef90065c0eeb7f91977701] (
Linaro GCC 7.3-2018.05) ) #58 SMP PREEMPT Thu Apr 15 10:48:57 CST 202
1
nvidia@EAXVA04:~$
```

四、Xavier 端功能使用

4.1RS232 以及 RS485 的使用

RS232_1_TXD	121P-52	RS-232 串口 1	Xavier <i>ttyTHS0</i>
RS232_1_RXD	121P-71		
RS232_2_TXD	121P-69	RS-232 串口 2	Xavier <i>ttyTHS1</i>
RS232_2_RXD	121P-50		
RS232_3_TXD	121P-51	RS-232 串口 3	Xavier ttyTCU0 默认用于 Debug
RS232_3_RXD	121P-70		
RS232_4_TXD	121P-68	RS-232 串口 4	Xavier <i>ttyTHS6</i>
RS232_4_RXD	121P-49		
RS485_A	121P-34	RS485	Xavier <i>ttyTHS4</i>
RS485_B	121P-33		

工具：usb 转串口线、RS485 转串口线

测试步骤：

- 1、 RS232 和 PC 用 usb 转串口线相连，可以先连接线束上的 232-1 接口，在 pc 端打开串口调试助手软件，比如 XCOM。



- 2、测试程序是 a04 设备的/opt/eco-ev/demo/uart\_test/uart\_test.sh 脚本文件。

```
nvidia@EAXVA04:/opt/eco-ev/demo/uart_test$ ls
rs232  rs232_test.c  rs485  rs485.c  uart_test.sh
```

- 3、在终端运行./uart\_test.sh 程序,如下图所示:

```
nvidia@EAXVA04:/opt/eco-ev/demo/uart_test$ ./uart_test.sh
-----
uart port test  Baud Rate=9600
-----
1 : RS232-1
2 : RS232-2
3 : RS485
4 : RS232-4
5 : set Baud Rate
q : exit
choice [1,2,3,4,5,q]-->
```

以 RS232-1 串口为例:

- 1) 执行./uart\_test.sh 程序后,输入 1,回车,如下图所示:

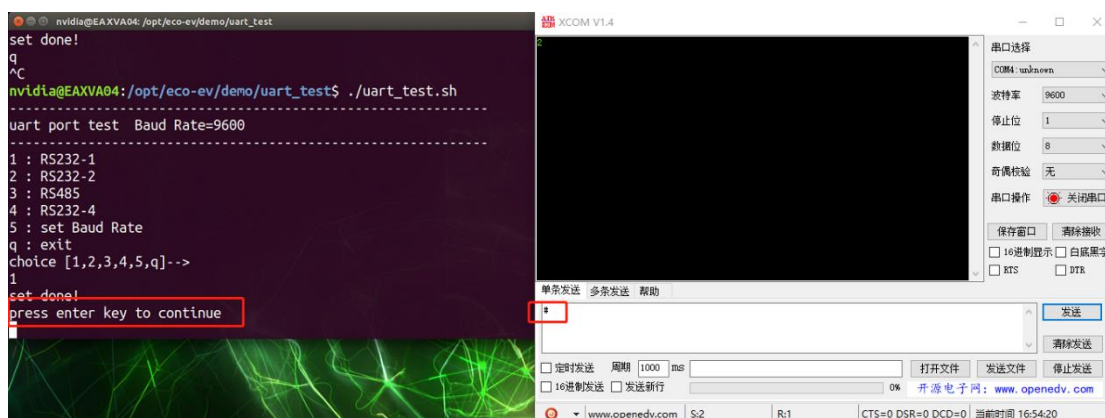
```
nvidia@EAXVA04:/opt/eco-ev/demo/uart_test$ ./uart_test.sh
-----
uart port test  Baud Rate=9600
-----
1 : RS232-1
2 : RS232-2
3 : RS485
4 : RS232-4
5 : set Baud Rate
q : exit
choice [1,2,3,4,5,q]-->
1
set done!
```

- 2) 出现 set done!,说明设置完成。
- 3) 在打开的 XCOM 软件中输入任意字符,查看是否能立即返回字符+1 后的字符。比如输入 1,应该返回字符 2.如下图所示:





4) 然后在 XCOM 中输入 “#” 字符，结束这次测试，如下图所示：



A04 终端下出现 “press enter key to continue”，说明本次测试完成，按任意键进入下一次测试。可以重新选择其他串口进行测试。

5) 选择 1、2、4 选项对应的三路 RS232 测试，选择 3 项进行 485 测试（线束 33 口与 34 口分别连接 485 转换器 T/R-与 T/R+，485 转换器和 PC 的 usb 口相连），然后在此前已打开的 XCOM 串口调试工具界面输入任意字符查看是否能立即返回字符值+1 后的字符。

注：波特率默认 9600，更改其他波特率需要 ./uart\_test.sh 程序选择 5 选项进行设置。

6) 修改波特率，执行 “./uart\_test.sh” 程序，输入 5 选项进行波特率设置，设置完成后，重复执行前 4 步，进行串口测试。

```
nvidia@EAXVA04:/opt/eco-ev/demo/uart_test$ ./uart_test.sh
-----
uart port test  Baud Rate=9600
-----
1 : RS232-1
2 : RS232-2
3 : RS485
4 : RS232-4
5 : set Baud Rate
q : exit
choice [1,2,3,4,5,q]-->
5
115200
press enter key to continue
```

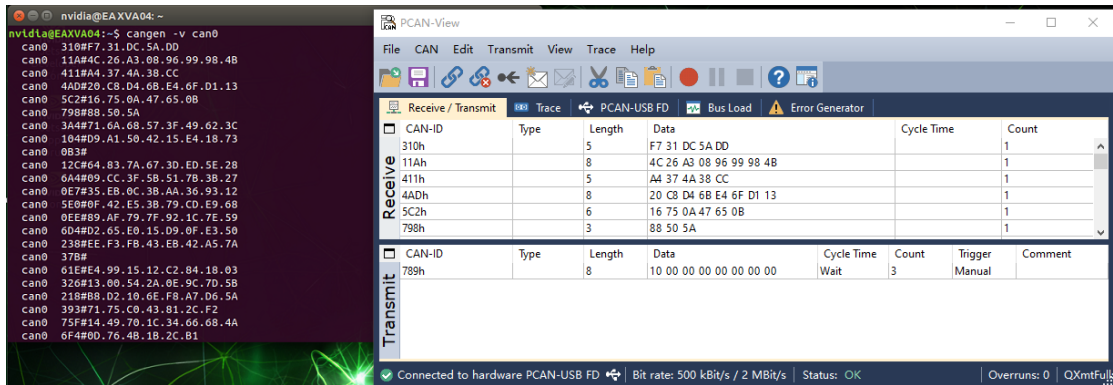
7) 串口测试完成后，按 “q” 键退出程序。

## 4.2 CAN 使用

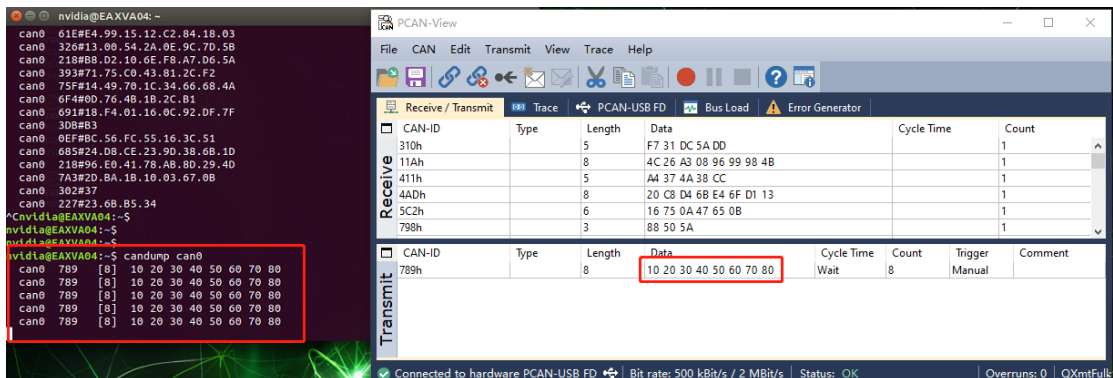
CAN_X0_H	121P-47	包含 120 $\Omega$ 终端电阻	终端电阻可选配 对应 Xavier 中的 CAN0
CAN_X0_L	121P-66		
CAN_X1_H	121P-48	包含 120 $\Omega$ 终端电阻	终端电阻可选配 对应 Xavier 中的 CAN1
CAN_X1_L	121P-67		

测试 can 接口有两种方式：1、通过 PCAN 工具将 PC 和 A04 设备相连。2、A04 设备两个 can 相连互相测试。

1、通过 PCAN 工具将 PC 和 A04 设备相连，在 Windows 下使用 PCAN-View 软件，然后在 A04 设备下打开终端，执行发送命令“cangen -v can0”，在 PCAN-View 软件可接受到数据，如下图所示：

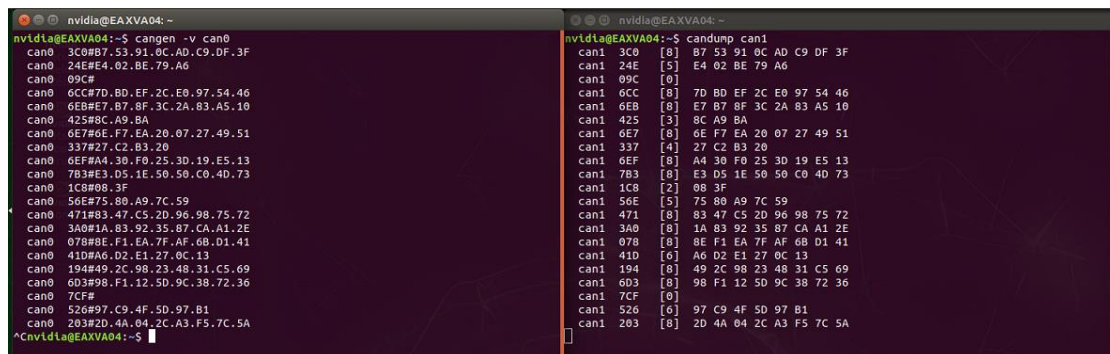


在 A04 设备终端执行“candump can0”用来接收数据，Windows 下发送数据，如下图显示：



2、通过接线端子将 A04 的两个 can 接口短接，互相测试。

1) 打开两个终端，分别输入“cangen -v can0”和“candump can1”，can0 随机发送数据，can1 接收数据。如下图所示：



2) 然后分别输入“cangen -v can1”和“candump can0”，can1 随机发送数据，can0 接收数

据。即可完成 can0 和 can1 的收发测试。

1、串口终端中输入指令“cangen -v can0”，PC ubuntu 运行./can-read.sh，查看 PC 是否能够接收到 CAN\_X0 通道发送的随机 CAN 数据帧

2、PC ubuntu 运行./can-write.sh，串口终端中输入指令“candump can0”，查看 CAN\_X0 通道是否能够接收到 PC 发送的随机 CAN 数据帧

## 4.3 普通网口使用

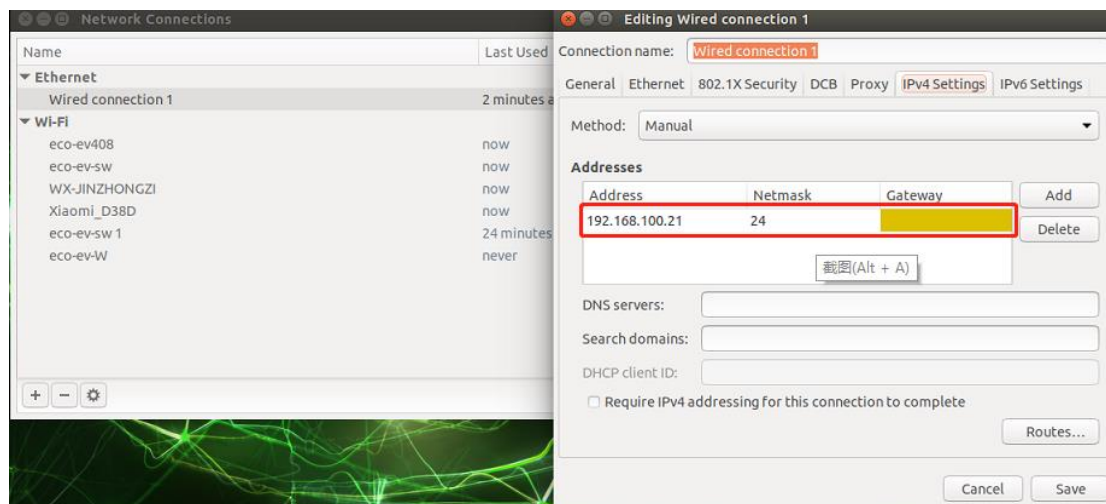
NPort3_BI_DD+	EEG2-1	普通以太网 3	100BASE-TX/1000BASE-T
NPort3_BI_DD-	EEG2-2		
NPort3_BI_DC+	EEG2-3		
NPort3_BI_DC-	EEG2-4		
NPort3_BI_DB+	EEG2-5		
NPort3_BI_DB-	EEG2-6		
NPort3_BI_DA+	EEG2-7		
NPort3_BI_DA-	EEG2-8		

注意：

- 1) 测试网络前需要使用“ifconfig eth0”命令提前查看下 Xavier 的网络 ip 地址。
- 2) pc 端 ip 地址需要和 Xavier 设备在同一网段下

### Xavier 端 IP 地址修改

1、通过系统设置配置以太网的 IP 地址，如下图所示：



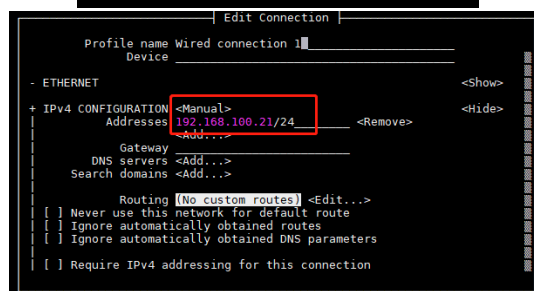
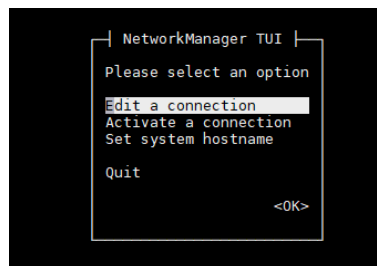
Xavier 的 IP 地址修改成功后，打开终端使用“ping 192.168.100.100”命令，检测以太网是否正常使用，如下图所示：

```
nvidia@EAXVA04:~$ ping 192.168.100.100
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data:
64 bytes from 192.168.100.100: icmp_seq=1 ttl=64 time=1.08 ms
64 bytes from 192.168.100.100: icmp_seq=2 ttl=64 time=0.189 ms
64 bytes from 192.168.100.100: icmp_seq=3 ttl=64 time=0.318 ms
64 bytes from 192.168.100.100: icmp_seq=4 ttl=64 time=0.280 ms
64 bytes from 192.168.100.100: icmp_seq=5 ttl=64 time=0.208 ms
64 bytes from 192.168.100.100: icmp_seq=6 ttl=64 time=0.226 ms
64 bytes from 192.168.100.100: icmp_seq=7 ttl=64 time=0.280 ms
64 bytes from 192.168.100.100: icmp_seq=8 ttl=64 time=0.653 ms
64 bytes from 192.168.100.100: icmp_seq=9 ttl=64 time=0.291 ms
64 bytes from 192.168.100.100: icmp_seq=10 ttl=64 time=0.233 ms
```

2、在没有接显示器的情况下，可以通过调试串口来修改和确认 Xavier 的 IP 地址。

- 1) 通过 usb 转串口线连接 pc 和 Xavier 设备上的 232-3 接口，在 pc 上使用串口调试助手打开 232-3 接口。
- 2) 在串口终端输入“export LANG=C”命令，防止字符乱码，然后输入 `sudo nmtui`，进入菜单选择 Edit a connection 选项，选择 Wired connection 2 回车进入，将 IPv4 改成手动模式，配置 ip 地址为 **192.168.100.21/24**，然后将光标移动到最下面的“OK”选项，回车一直选择“OK”选择，直到退出配置。至此网络配通。

注意：可以按照自己的需求修改 IP 地址。







## 4.5 摄像头使用

Camera-1	FAKRA-1	FPD-Link III 串行摄像头接口 1	Z Type
Camera-2	FAKRA-2	FPD-Link III 串行摄像头接口 2	Z Type
Camera-3	FAKRA-3	FPD-Link III 串行摄像头接口 3	Z Type
Camera-4	FAKRA-4	FPD-Link III 串行摄像头接口 4	Z Type
Camera-5	FAKRA-5	FPD-Link III 串行摄像头接口 5	Z Type
Camera-6	FAKRA-6	FPD-Link III 串行摄像头接口 6	Z Type
Camera-7	FAKRA-7	FPD-Link III 串行摄像头接口 7	Z Type
Camera-8	FAKRA-8	FPD-Link III 串行摄像头接口 8	Z Type

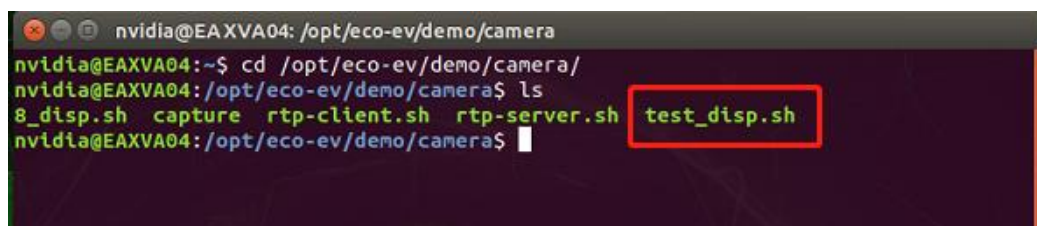
Xavier 共有 8 路摄像头, 分别对应设备上的 FAKRA0-7 接口。摄像头分为带 ISP 功能和不带 ISP 功能的两种摄像头。

**注意:** 下文中提到的 `rtp-server.sh` 和 `rtp-client.sh` 程序, 对于带 ISP 和不带 ISP 功能的摄像头来说是不一样的脚本测试程序, 具体可以询问相关技术人员。

带 ISP 功能摄像头使用:

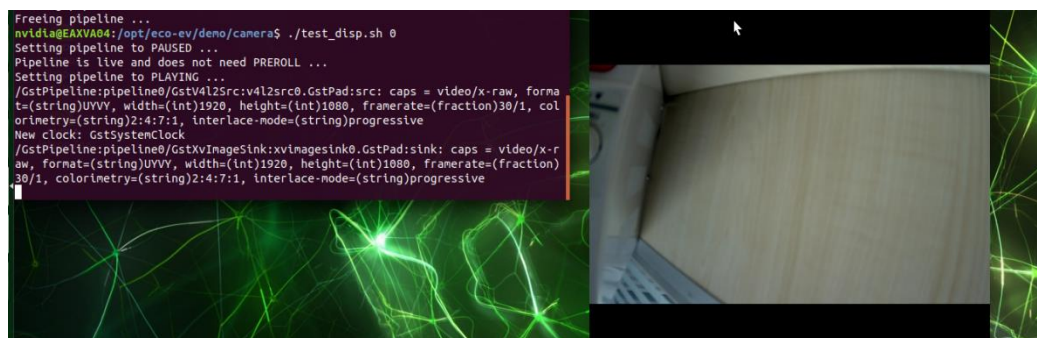
接显示器的情况:

- 1) 打开终端, 进入到 `/opt/eco-ev/demo/camera` 目录, 目录下 `test_disp.sh` 文件是摄像头测试脚本文件。如下图所示:



```
nvidia@EAXVA04: /opt/eco-ev/demo/camera
nvidia@EAXVA04:~$ cd /opt/eco-ev/demo/camera/
nvidia@EAXVA04:/opt/eco-ev/demo/camera$ ls
8_disp.sh  capture  rtp-client.sh  rtp-server.sh  test_disp.sh
nvidia@EAXVA04:/opt/eco-ev/demo/camera$
```

- 2) 连接好摄像头, 例如摄像头接到第一个接口上, 在终端执行 `./test_disp.sh 0` 命令, 即可打开摄像头。如下图所示:



不接显示器的情况下:

- 1) 连接普通网口和上位机 ubuntu 系统, 确保上位机是否能够 PING 通 Xavier

- 2) 可以通过调试串口或者 ssh 方式登录 xavier 设备，在这使用 ssh 方式登录 xavier 设备。

使用“ssh nvidia@192.168.1.123”命令登录 xavier 设备，进入到测试脚本所在目录“/opt/eco-ev/demo/camera/”如下所示：

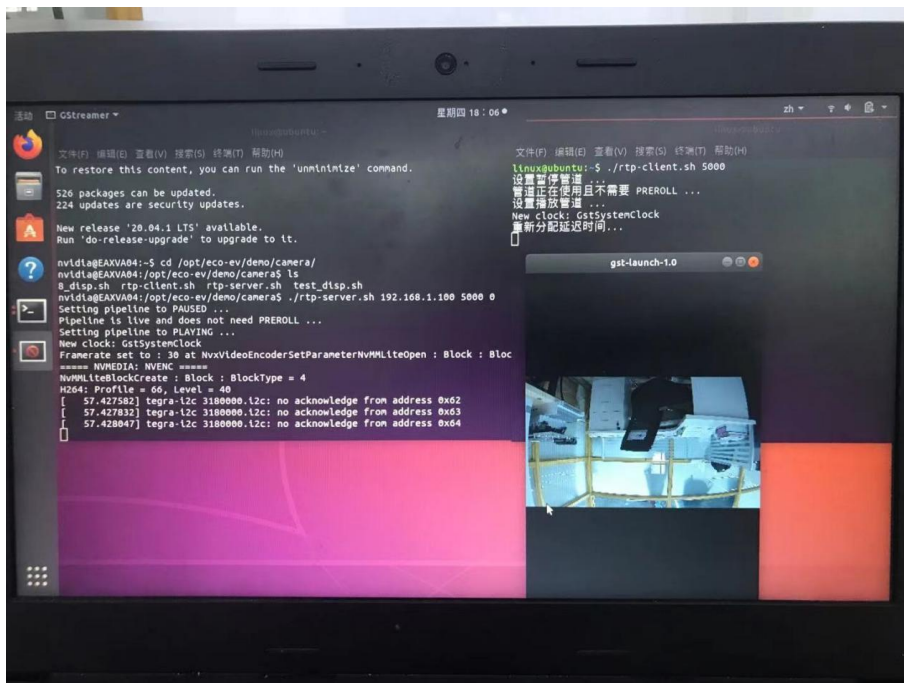
```
nvidia@EAXVA04:/opt/eco-ev/demo/camera$ ls
8_disp.sh  rtp-client.sh  rtp-server.sh  test_disp.sh
nvidia@EAXVA04:/opt/eco-ev/demo/camera$
```

- 3) 在终端运行“./rtp-server.sh 192.168.1.102 5000 0”命令打开第一个摄像头接口。如下图所示：

注：192.168.1.102 为上位机 ip，需要与自己 pc 的 ip 一致；0 是 FAKRA0 接口摄像头，用于设置要检测的 FAKRA0-7 接口的摄像头号；5000 为端口号，server、client 两端保持一致。

```
nvidia@EAXVA04:/opt/eco-ev/demo/camera$ ./rtp-server.sh 192.168.1.102 5000 0
Setting pipeline to PAUSED ...
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
Framerate set to : 30 at NvxBVideoEncoderSetParameterNvMMLiteOpen : Block : Block
Type = 4
===== NVMEDIA: NVENC =====
NvMMLiteBlockCreate : Block : BlockType = 4
H264: Profile = 66, Level = 40
```

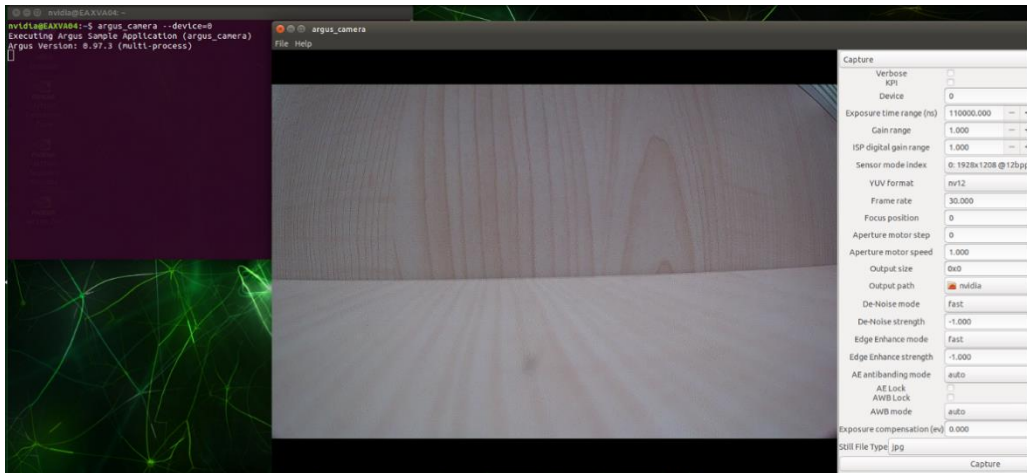
- 4) 上位机 ubuntu 运行“./rtp-client.sh 5000”，可以看到动态、清晰的图像，如下图所示：



不带 ISP 功能摄像头使用：

接显示器情况：

连接摄像头到第一个接口，打开终端，执行命令“argus\_camera --device=0”（其中 0~7，分别表示 8 个摄像头接口），摄像头即可显示图像，如下图所示：



不接显示器的情况:

- 5) 连接普通网口和上位机 ubuntu 系统，确保上位机是否能够 PING 通 Xavier
- 6) 可以通过调试串口或者 ssh 方式登录 xavier 设备，在这使用 ssh 方式登录 xavier 设备。使用“ssh nvidia@192.168.1.123”命令登录 xavier 设备，进入到测试脚本所在目录“/opt/eco-ev/demo/camera/”如下所示:

```
nvidia@EAXVA04:/opt/eco-ev/demo/camera$ ls
8_disp.sh  rtp-client.sh  rtp-server.sh  test_disp.sh
nvidia@EAXVA04:/opt/eco-ev/demo/camera$
```

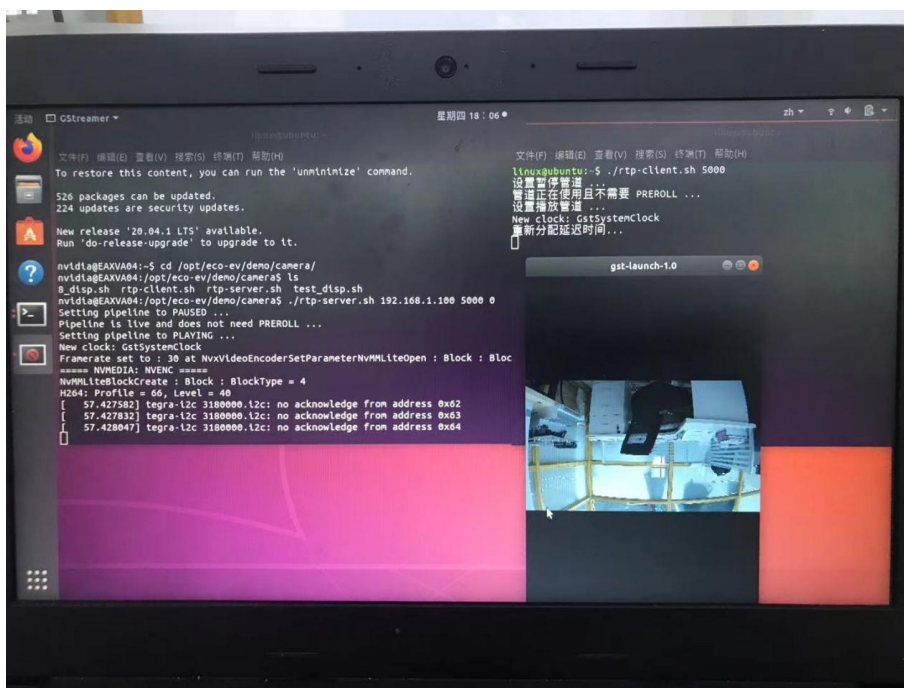
- 7) 在终端运行“./rtp-server.sh 192.168.1.102 5000 0”命令打开第一个摄像头接口。如下图所示:

注: 192.168.1.102 为上位机 ip, 需要与自己 pc 的 ip 一致; 0 是 FAKRA0 接口摄像头, 用于设置要检测的 FAKRA0-7 接口的摄像头号; 5000 为端口号, server、client 两端保持一致。

```
nvidia@EAXVA04:/opt/eco-ev/demo/camera$ ./rtp-server.sh 192.168.1.102 5000 0
Setting pipeline to PAUSED ...
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
Framerate set to : 30 at NvxFVideoEncoderSetParameterNvMMLiteOpen : Block : Block
Type = 4
===== NVMEDIA: NVENC =====
NvMMLiteBlockCreate : Block : BlockType = 4
H264: Profile = 66, Level = 40
```

- 8) 上位机 ubuntu 运行“./rtp-client.sh 5000”, 可以看到动态、清晰的图像, 如下图所示:





## 4.6 固态硬盘的使用

- 1) 查看硬盘是否识别成功：在终端中输入 `lsblk` 查看块设备找到固态硬盘名 `nvme0n1`

```
nvidia@EAXVA04:~$ lsblk | grep nvme
nvme0n1      259:6      0 465.8G  0 disk
nvidia@EAXVA04:~$
```

- 2) 第一次使用硬盘需要给硬盘创建分区，并进行格式化：

```
sudo fdisk /dev/nvme0n1
```

输入 `n` 新建分区，选择 `primary` 分区，过程中一至按回车，结束后输入 `w` 保存。

```

nvidia@EAXVA04:~$ sudo fdisk /dev/nvme0n1

Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): p
Disk /dev/nvme0n1: 465.8 GiB, 500107862016 bytes, 976773168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x48781477

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-976773167, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-976773167, default 976773167):

Created a new partition 1 of type 'Linux' and of size 465.8 GiB.
Partition #1 contains a ext4 signature.

Do you want to remove the signature? [Y]es/[N]o: y
The signature will be removed by a write command.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

nvidia@EAXVA04:~$

```

### 3) 格式化分区

`sudo mkfs.ext4 /dev/nvme0n1p1`

```

nvidia@EAXVA04:~$ sudo mkfs.ext4 /dev/nvme0n1p1
mke2fs 1.44.1 (24-Mar-2018)
Discarding device blocks: done
Creating filesystem with 122096390 4k blocks and 30531584 inodes
Filesystem UUID: 77b87e2b-fd67-4912-ab9e-55a2ec65b0e8
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
    102400000

Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done

```

### 4) 挂载硬盘

`sudo mount /dev/nvme0n1p1 /mnt`

### 5) 读写测试

`cd /mnt`

`sudo vi test.txt`

创建 `test.txt` 文件并写入随意信息，保存信息并退出后再次打开文件查看信息是否能保存

## 4.7 PPS 功能

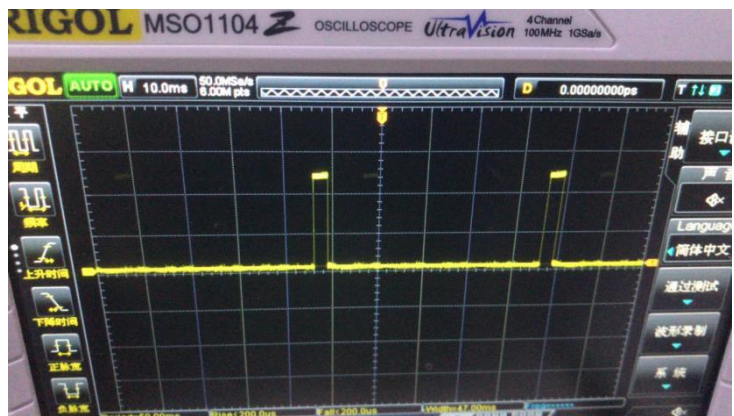
PPS_IN	121P-23	秒脉冲同步输入信号	Xavier, 支持 3.3V-16V, 硬件配置
PPS_OUT1	121P-81	秒脉冲同步输出信号	Xavier, 12V 输出
PPS_OUT2	121P-80	秒脉冲同步输出信号	Xavier, 12V 输出
PPS_OUT3	121P-79	秒脉冲同步输出信号	Xavier, 3.3V 或者 5V 输出
PPS_OUT4	121P-78	秒脉冲同步输出信号	Xavier, 3.3V 或者 5V 输出

Xavier 有 4 路 PPS\_OUT 和一路 PPS\_IN 功能。

- 1) 将线束 78、79 分别连接线束 23 口，终端界面运行“sudo ppstest /dev/pps0”，结果如下所示：

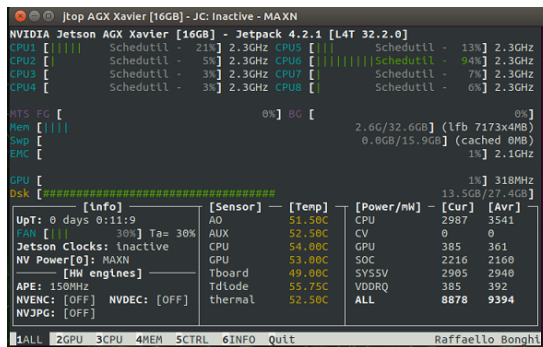
```
nvidia@EAXVA04:~$ sudo ppstest /dev/pps0
trying PPS source "/dev/pps0"
found PPS source "/dev/pps0"
ok, found 1 source(s), now start fetching data...
time_pps_fetch() error -1 (Connection timed out)
time_pps_fetch() error -1 (Connection timed out)
time_pps_fetch() error -1 (Connection timed out)
source 0 - assert 1517156308.240051571, sequence: 1813 - clear 0.000000000, sequence: 0
source 0 - assert 1517156308.290046605, sequence: 1814 - clear 0.000000000, sequence: 0
source 0 - assert 1517156308.340037255, sequence: 1815 - clear 0.000000000, sequence: 0
source 0 - assert 1517156308.390045217, sequence: 1816 - clear 0.000000000, sequence: 0
source 0 - assert 1517156308.440046556, sequence: 1817 - clear 0.000000000, sequence: 0
```

- 2) 使用示波器连接线束 80 或者 81；80 或者 81 口示波器显示频率为 20hz，80 或者 81 为 12v 电压（上下电压偏置 5V），如下图所示：

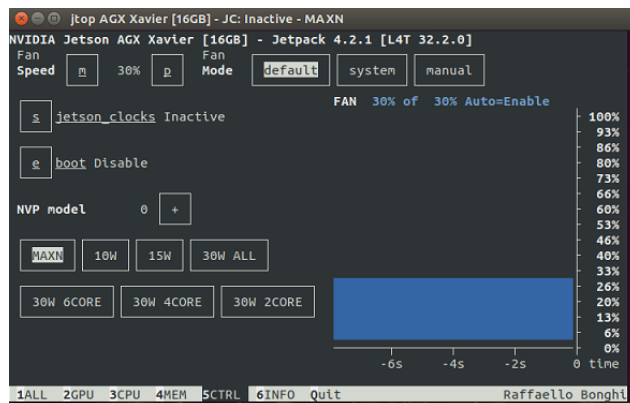


## 4.8 风扇调速

- 1) 打开终端输入“jtop”命令，如下所示：



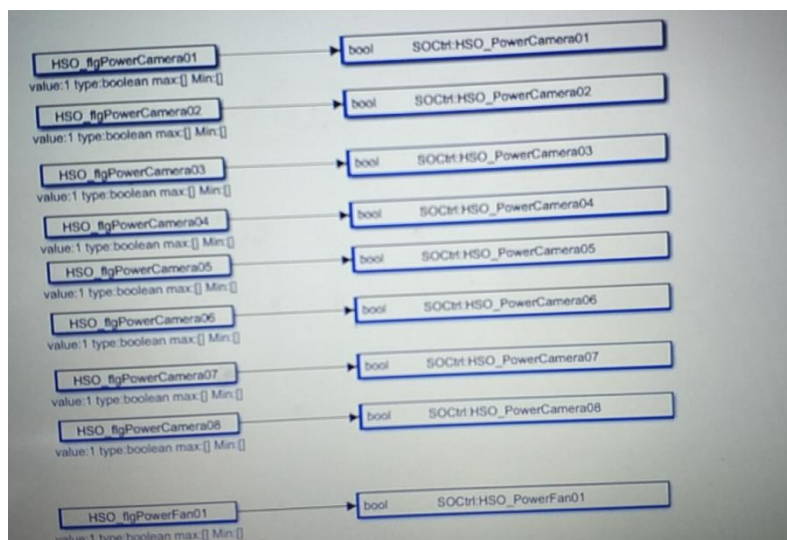
2) 点击最下面一行的选项 5，进入风扇控制界面，如下所示：



从键盘输入“m”和“p”，即可控制风扇转速。测试完成后按“q”退出。

## 五、注意事项

5.1 如果用户要根据自己需求更改 MCU 程序，需在 MATLAB 模型里添加以下端口，并打开。这些端口分别用来控制摄像头和风扇供电。



5.2 如果用户要更新系统 apt upgrade，请与优控确认是否可以更新。

5.3 如果用户使用 JetPack4.6 系统，打开摄像头时需遵循 7-0 的顺序，依次打开摄像头。