EVO TX2 GMSL2用户手册

安全警示及使用注意事项

请在使用本产品前仔细阅读本手册，未经授权的操作会导致错误或意外。制造商对因错误操作而导致设备出现的任何问题均不负责。

- 避免热插拔设备接口。
- 要正确关闭电源，请先关闭Ubuntu系统，然后再切断电源。由于Ubuntu系统的特殊性，在Nvidia的开发板上，如启动未完成的时候强行断电，会有0.03%的概率出现异常，进而导致设备无法启动。由于使用Ubuntu系统，米文的设备上也会存在同样的问题。
- 请勿使用本手册提及以外的线缆。
- 避免在强磁场环境下使用本设备。
- 长期不使用及运输前需要对数据进行备份。
- 推荐使用原包装进行运输。
- 警告！ 此为A级产品，在生活环境中，该产品可能会造成无线电干扰。在这种情况下，可能需要用户对干扰采取切实可行的措施。

简介 Brief

米文EVO TX2 GMSL2是一款专为工业场景设计的嵌入式边缘计算设备。它搭载的NVIDIA® Jetson TX2能够以15W的低功耗提供高达1.3TFlops (FP16)的算力。EVO TX2 GMSL2采取高效能的被动散热设计，可以在苛刻的工业环境中稳定正常工作。同时鉴于其紧固的嵌入式设计，可以达到很高的抗振等级。EVO TX2 GMSL2提供了丰富的I/O功能，可以满足多种专用传感器的接入需求,特别是6路GSML2接口，可以满足6路摄像机的接入。同时EVO TX2 GMSL2还在内部设计了多种扩展接口，提供更多通讯以及存储扩展方案。

产品清单

-EVO TX2 GMSL2 x 1

-电源配适器 x 1

-电源转接线 x 1

-扩展安装部件 x 2

-安装螺丝 若干

-快速上手指南及保修卡 x 1

-合格证 x 1

产品规格 Specifications

处理器模组 Processor

	Specification
Processor	NVIDIA Jetson TX2
AI Performance	Up to 1.3 TFLOPS (FP16)
CPU	Dual-Core NVIDIA Denver 2 64-Bit CPUQuad-Core ARM® Cortex®-A57 MPCore
GPU	256-core NVIDIA Pascal™ GPU architecture with 256 NVIDIA CUDA cores
Memory	8GB 128-bit LPDDR4 Memory1866 MHz - 59.7 GB/s
Storage	32GB eMMC 5.1
Video Encode	1x 4Kp60 3x 4Kp30 4x 1080p60 8x 1080p30 (H.265) 1x 4Kp60 3x 4Kp30 7x 1080p60 14x 1080p30 (H.264)
Video Decode	2x 4Kp60 4x 4Kp30 7x 1080p60 14x 1080p30 (H.265 & H.264)

接口 I/O

	Interface	Quantity	Note
Function KEY	Recovery Button	1	
Network	Ethernet	2×Gigabit Port	2 independent Gigabit Ethernet port RJ45
Camera	Camera	6×GMSL FAKRA Z TYPE	10V Transmission distance up to 15 meters GMSL2,compatible with GMSL1
Video output	HDMI	1×HDMI 2.0 TYPE A	5V 1A
USB	USB	3×USB 3.0 TYPE A 1×USB 2.0 TYPE A	USB 5V, 1A USB 2.0 Flashing Port
I/O	UART	2xRS232 1xRS485	DB9 Terminal
	CAN	2	Two CAN in One DB9 Terminal With CAN chip, terminal resistor 120
	GPIO	2	DB9 Terminal
	SYNC IO	1	DB9 Terminal
User Expansion	TF Socket	1xTF Slot	MicroSD card supported
	M.2	1×M.2 M Key	2242 SIZE NVME SSD
	Mini PCIe	1	For 4G or WiFi expansion
	Nano SIM Socket	1	For Nano SIM Card

供电 Power Supply

Power Supply	Spec
Input Type	DC
Input Voltage	Wide input 12V-30V DC
Maximum Consumption	15W

结构 Mechanical

Mechanical	Spec
Dimensions (W×H×D)	178mm×70mm×110mm (I/O ports and mounting holes excluded)
Weight	1.2Kg

环境 Environmental

Environmental	Spec
Operating Temperature	-20℃-60℃, 0.2~0.3m/s air flow[1]

Storage Temperature	-25℃-80℃
Storage Humidity	10%-90% non-condensing
Vibration	5gn,10Hz~150Hz,3 Axis[2]
Protection	IP5X
ESD	Touch 6KV, Air 8KV
TVS	500V

认证 Certification

Certification	Status
CCC, CE, FCC, RoHS, SRRC	Processing

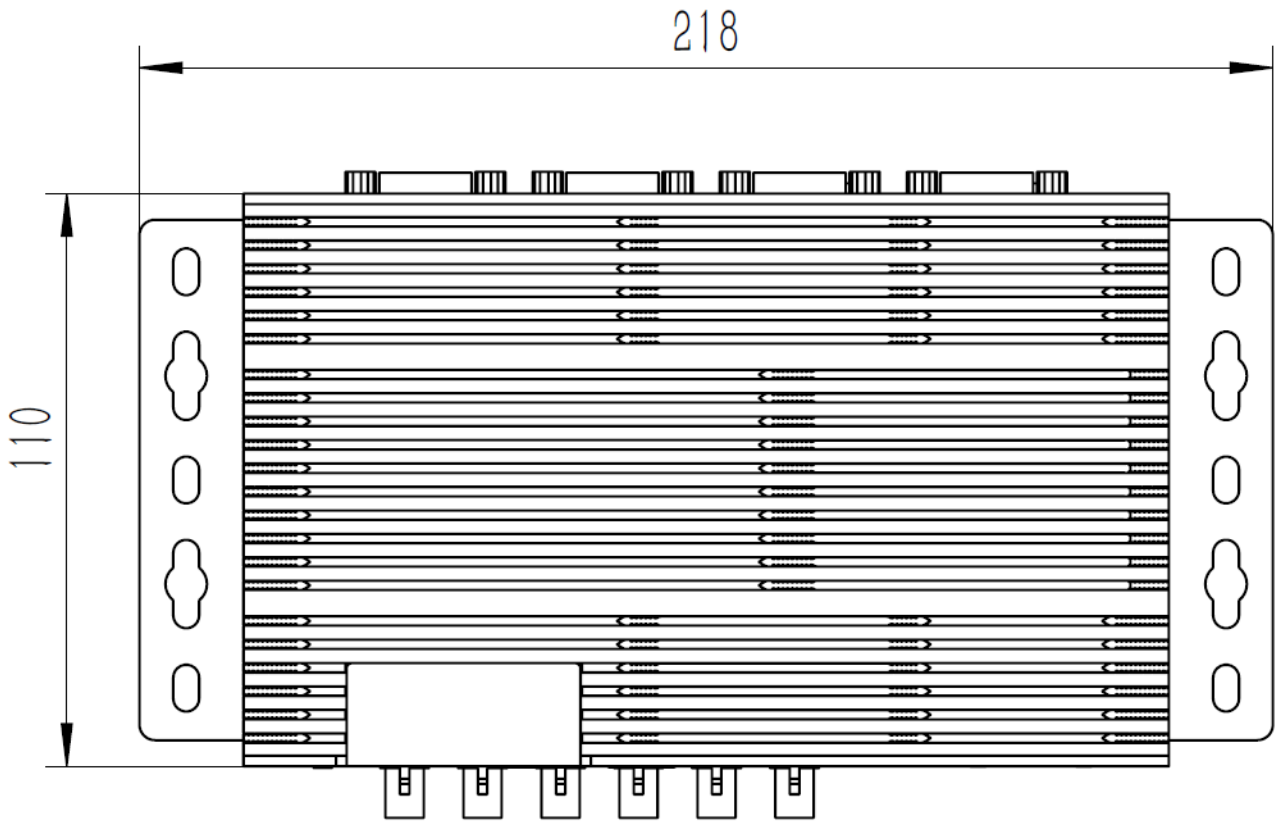
[1] According to GB/T 2423-2008 60℃以上运行时，运行频率降低 Working frequency is subject to change after temperature reaches 60℃
[2] According to GB/T 2423.10-2008

尺寸及安装 Install Dimension

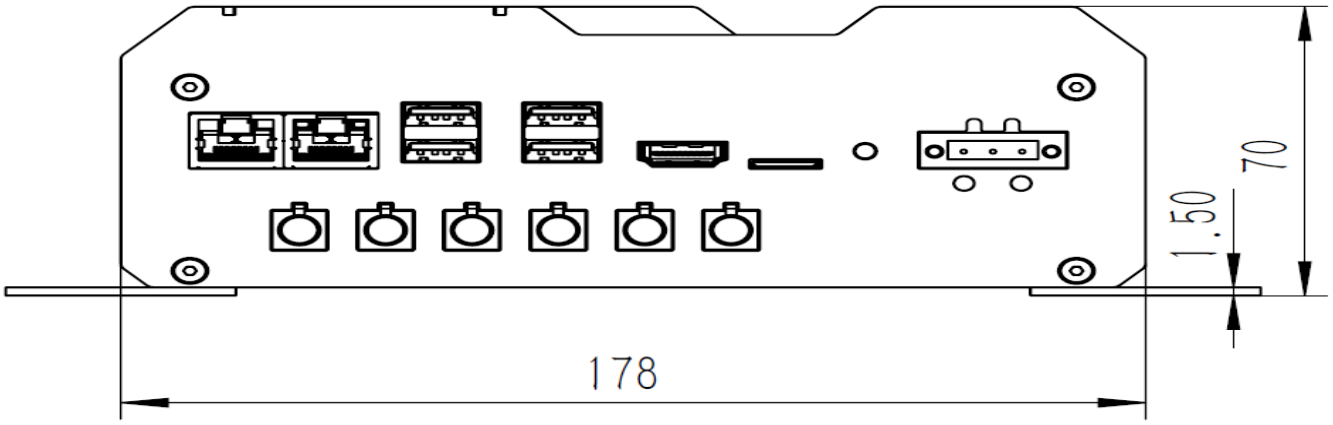
EVO TX2 GMSL2主体尺寸及安装孔位尺寸如图:

Dimensions and mounting hole position as below:

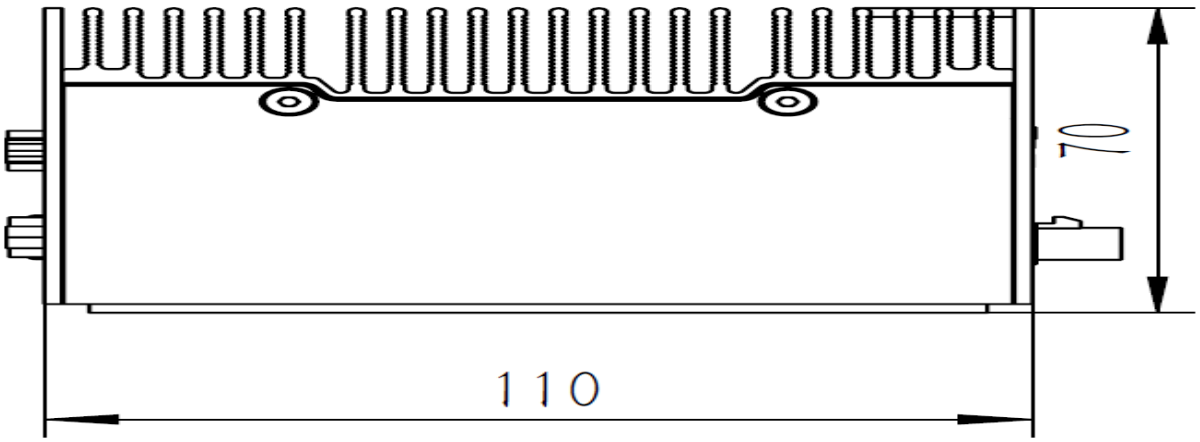
俯视图 Up view(Unit:mm)



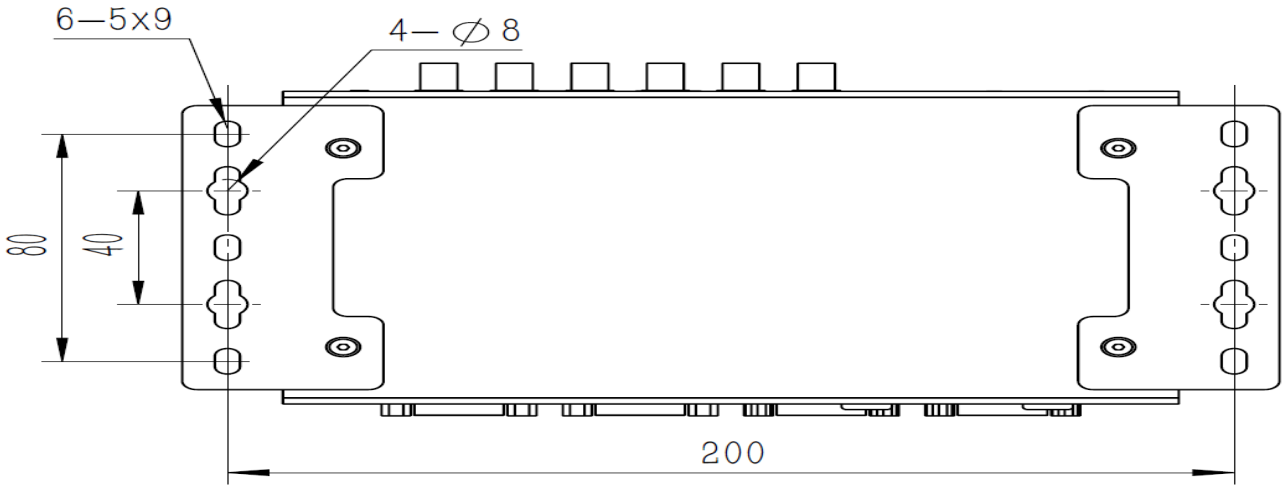
主视图 Front view(Unit:mm)



左视图 Left view(Unit:mm)



安装孔位图1 Mounting Hole(Unit:mm)



安装孔位图2 Mounting Hole(Unit:mm)

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，请发问题到email:helpdesk@miivii.com，我们将帮助您解决问题。也可访问米文技术论坛http://forum.miivii.com，搜索我们的知识库，以查找常见问题的解决方案。

保修

保修期：米文设备保修期为自购买之日起一年。 保修条例：保修期内产品，若出现非人为损坏的故障米文将进行免费保修。请联系helpdesk@miivii.com获取保修协助。

接口说明及扩展安装方式

接口说明

正面接口

图 EVO TX2 GMSL2正面接口示意图

接口	接口名称	接口说明
LAN_1,LAN_2	千兆网口	两路独立千兆网口
USB	三路USB 3.0接口与单路USB 2.0接口	3路USB 3.0接口 1路USB2.0接口，USB2.0口在Recovery模式下可用于烧写镜像 5V 1A
GMSL	GMSL 输入	可接入6路GMSL2协议的摄像头 GMSL2接口兼容GMSL1
HDMI	HDMI接口	HDMI 2.0 TYPE A 5V 1A
SD	TF卡槽	可以扩展TF卡 3.3V 1A
RECOVERY	Recovery模式按钮	按下后上电开机，可以进入Recovery模式
DC IN	电源接口	支持宽电压12V-30VDC输入
PWR	电源指示灯	载板上电：指示灯为黄色 载板启动：指示灯为白色常亮 载板错误：指示灯为红色常亮

STATUS	系统状态指示灯	系统启动前：指示灯为红色 系统启动后：指示灯为蓝色常亮

背面接口

图 EVO TX2 GMSL2背面接口示意图

接口	接口名称	接口说明
CAN_1&2	CAN接口1号与2号	包含两路CAN信 7.5V Max@48mA Max 带有CAN芯片，终端电阻120
RS485_1	RS485串口1号	差分输出驱动电压 2.0VDC Min，驱动A电流1mA Max
RS232_1	RS232串口1号	逻辑1: -3V~-12V，逻辑0: 3V~12V，电流1.6mA Max
RS232_2	RS232串口2号	逻辑1: -3V~-12V，逻辑0: 3V~12V，电流1.6mA Max
GPIO_1&2	GPIO 接口	2×GPIO_IN High 1V-12V, Low 0V-0.8V 2×GPIO_OUT 3.3V
Sync IO	Sync IO	1路Sync in同步功能 1路Sync out同步功能 1路PPS同步功能

CAN接口信号定义

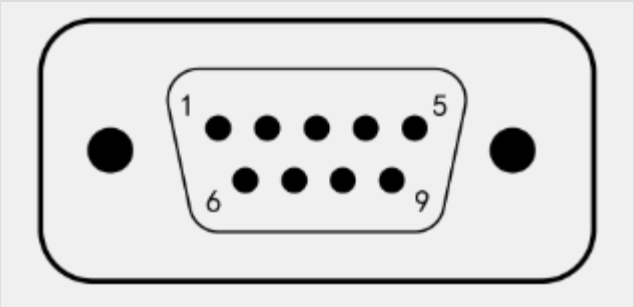


图 CAN_1&2接口序号图

接口名称	引脚序号	信号定义	接口说明
CAN_1&2	1	TX2_CAN1_L	TX2_CAN_1 L端
	2	TX2_CAN0_L	TX2_CAN_0 L端
	3	GND	地
	4-5	NC	空脚
	6	GND	地
	7	TX2_CAN0_H	TX2_CAN_0 H端
	8	TX2_CAN1_H	TX2_CAN_1 H端
	9	NC	空脚

RS485接口信号定义

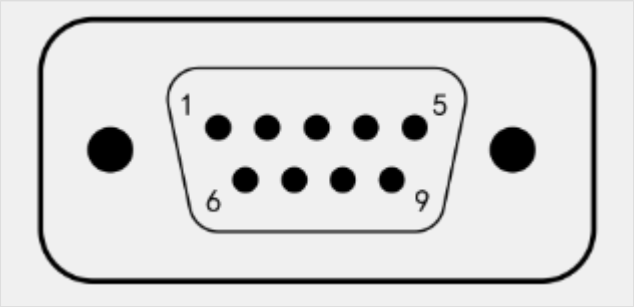


图 RS485接口序号图

接口名称	引脚序号	信号定义	接口说明
RS485_1	1	NC	空脚
	2	RS_485A	RS485_1号A端
	3	RS_485B	RS485_1号B端
	4	NC	空脚
	5	GND	地
	6-9	NC	空脚

RS232接口信号定义

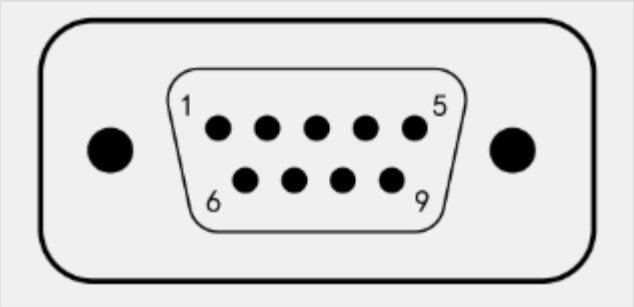


图 RS232接口序号图

接口名称	引脚序号	信号定义	接口说明
RS232_1	1	NC	空脚
	2	USB_UAR8_RXD	RS232_1号接收
	3	USB_UAR8_TXD	RS232_1号发送
	4	NC	空脚
	5	GND	地
	6-9	NC	空脚
RS232_2	1	NC	空脚
	2	USB_UART9_RXD	RS232_2号接收
	3	USB_UART9_TXD	RS232_2号发送
	4	NC	空脚
	5	GND	地
	6-9	NC	空脚

串口节点说明

UART接口的设备节点对应关系如下：

UART接口	设备节点
RS485_1	ttyUART_485_1
RS232_1	ttyUART_232_1
RS232_2	ttyUART_232_2

GPIO接口信号定义

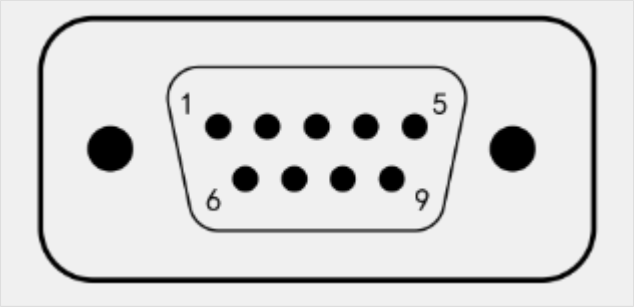


图 GPIO接口序号图

接口名称	DB9针脚序号	信号定义	接口说明
GPIO_D	1	GPIO3_PY.00_3V3	GPIO IN
GPIO_E	2	GPIO3_PY.06_3V3	GPIO IN
GPIO_B	3	GPIO3_PB.04_3V3	GPIO OUT
GPIO_C	4	GPIO3_PI.05_3V3	GPIO OUT
NC	5,9	NC	空
GND	6-8	GND	地

GPIO接口引脚号

GPIO接口引脚号对应关系如下：

GPIO名称	默认配置	引脚号
GPIO_B	GPIO_OUT	332
GPIO_C	GPIO_OUT	389
GPIO_D	GPIO_IN	480
GPIO_E	GPIO_IN	486

SYNC_IO接口信号定义

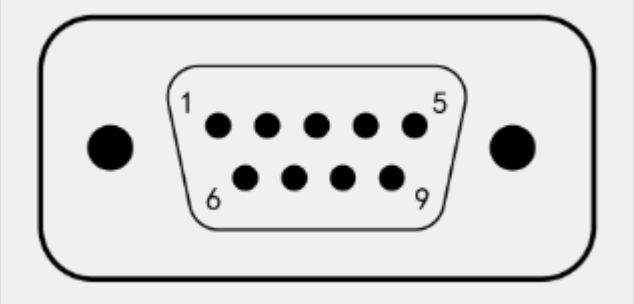


图 SYNC_IO接口序号图

接口名称	DB9针脚序号	信号定义	接口说明
PPS同步接口	2	PPSA_UART_RX	PPSA_UART(TTL/232)信号：RX
	3	PPSA_UART_TX	PPSA_UART(TTL/232)信号：TX
	5	GND	GND
	6	AO_DMIC_IN_3V3	PPS_A Pluse秒脉冲信号 3.3V
GND	1	GND	地
Sync out同步接口	9	GPIO_PW5_3V3	Sync out信号
Sync in同步接口	7	GPIO_PAA02_3V3	Sync in信号

Debug接口位置

米文EVO TX2 GMSL2的Debug(RS232)接口位于载板背面，位置如图。其中PIN 8为RX引脚，PIN 10为TX引脚

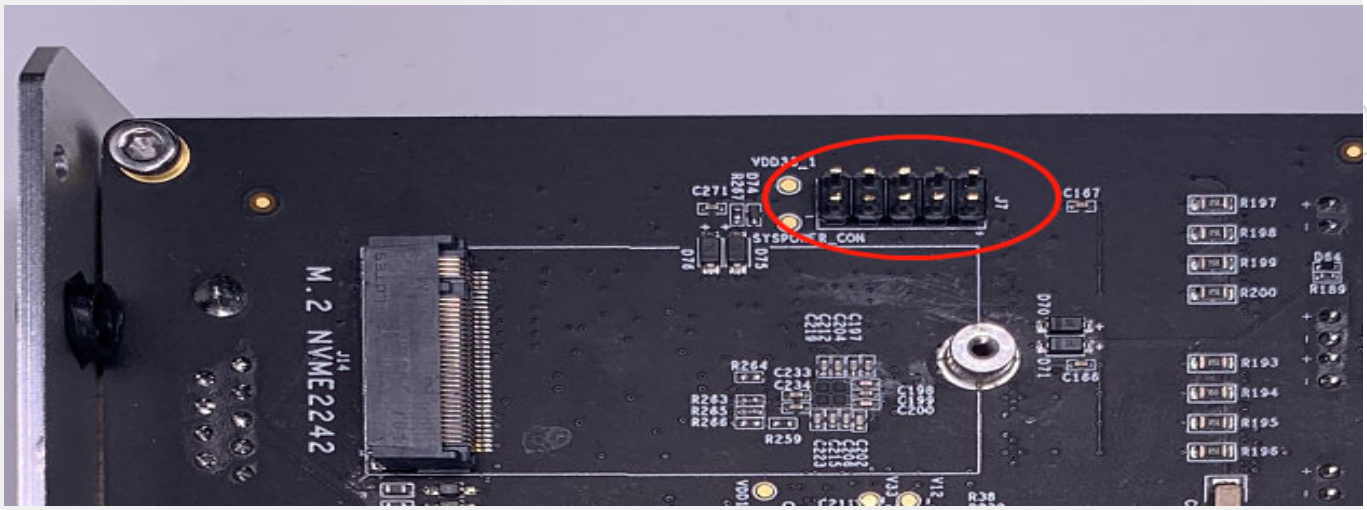


图 Debug接口位置图

引脚序号	信号定义	接口说明
6	GND	地
8	DGB_UART0_RX_232	调试串口RX信号
10	DGB_UART0_TX_232	调试串口TX信号

扩展设备安装方式

扩展设备安装方式

EVO TX2 GMSL2提供M.2 M Key, mini PCIe接口作为存储以及通讯扩展设备使用

安装扩展设备时，需要先拧开如下8个螺丝从而打开EVO TX2 GMSL2的底盖，如图所示：

图 EVO TX2 GMSL2螺丝孔位置1

图 EVO TX2 GMSL2螺丝孔位置2

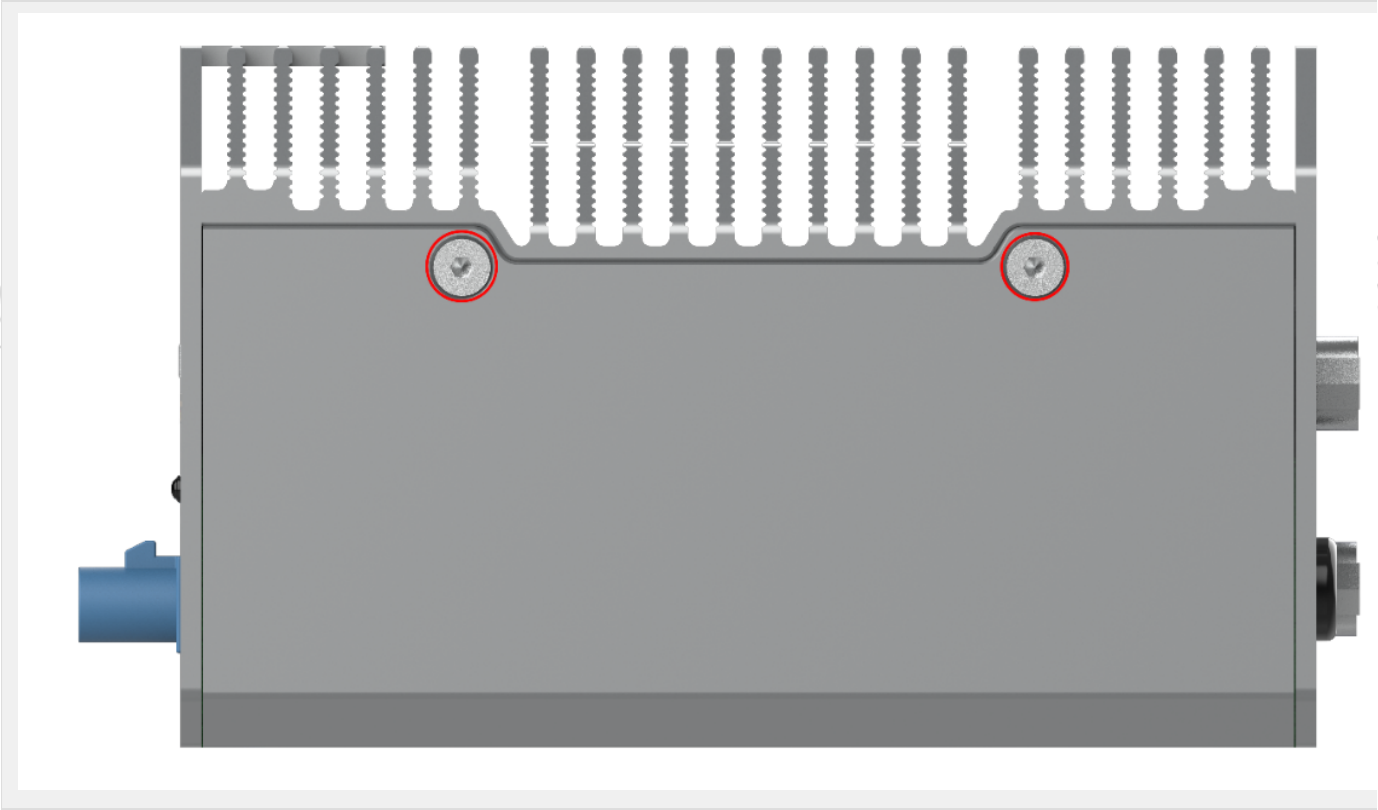


图 EVO TX2 GMSL2螺丝孔位置3

注意拧开左右侧板对称的4个螺丝

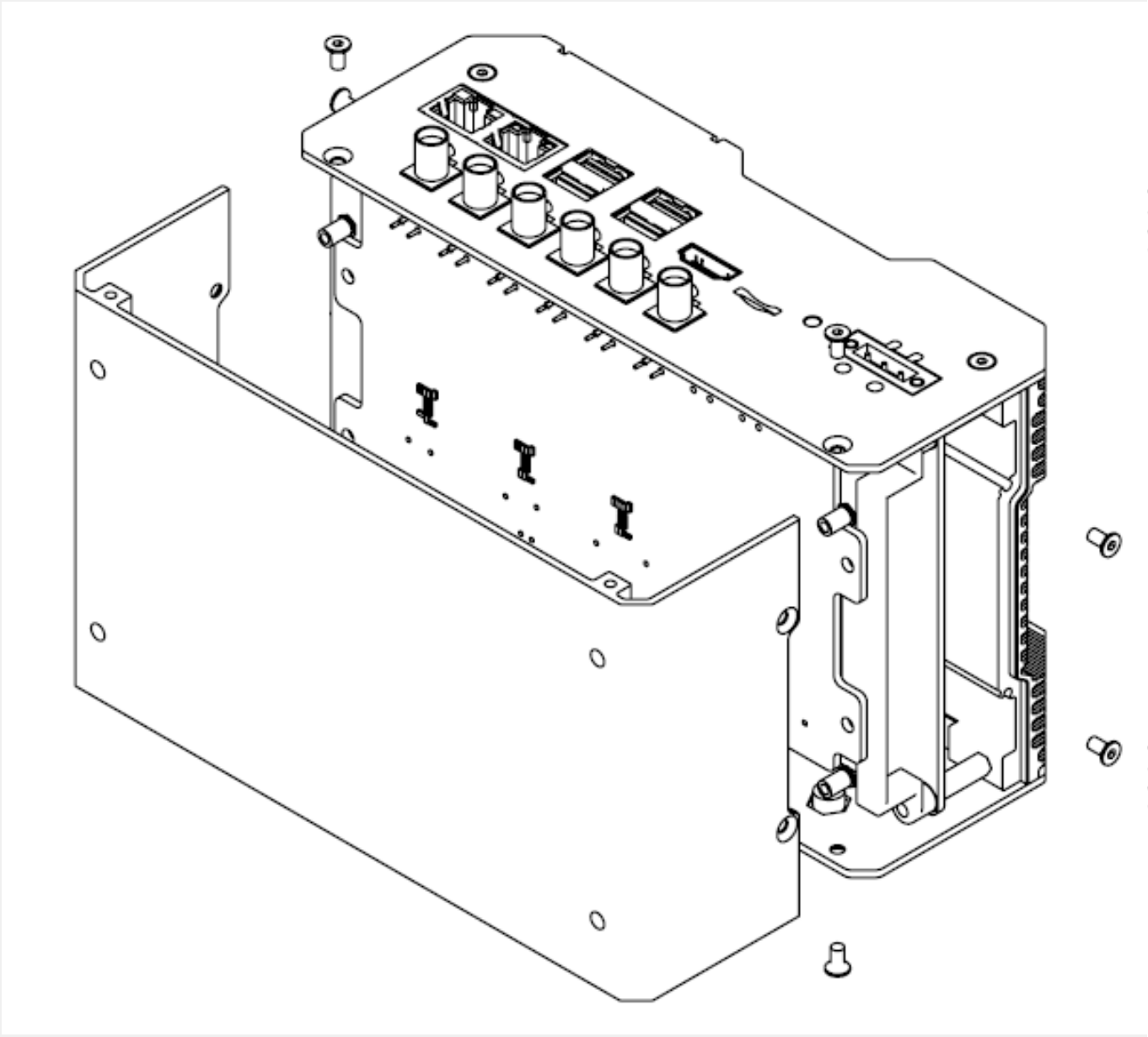


图 打开底盖

打开底盖后，可扩展接口位置如下图：

图 可扩展接口位置图

按照扩展需求，进行设备的安装，并用相应的螺丝固定扩展模块。注意如果需要安装mini PCIe接口的4G模块，请先将SIM卡插入到对应接口的Nano SIM卡槽中，再安装4G模块。

图 模块安装示意图

若采用WiFi及4G模块，需要安装天线以确保信号稳定。天线一端安装在模块上，另一端固定到EVO TX2后面板的预留孔位上。

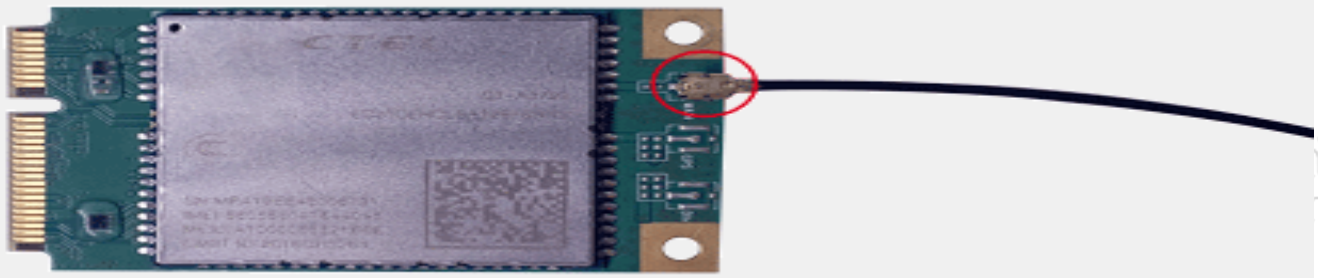


图 天线安装示意图1

图 天线安装示意图2

扩展设备安装完毕后，请参考第一步的方式，将设备及固定螺丝恢复原位置，将设备重新装好以便后续使用

米文设备固定方式

若需要将米文设备安装在其他设备上，请先安装附件中的安装固定板

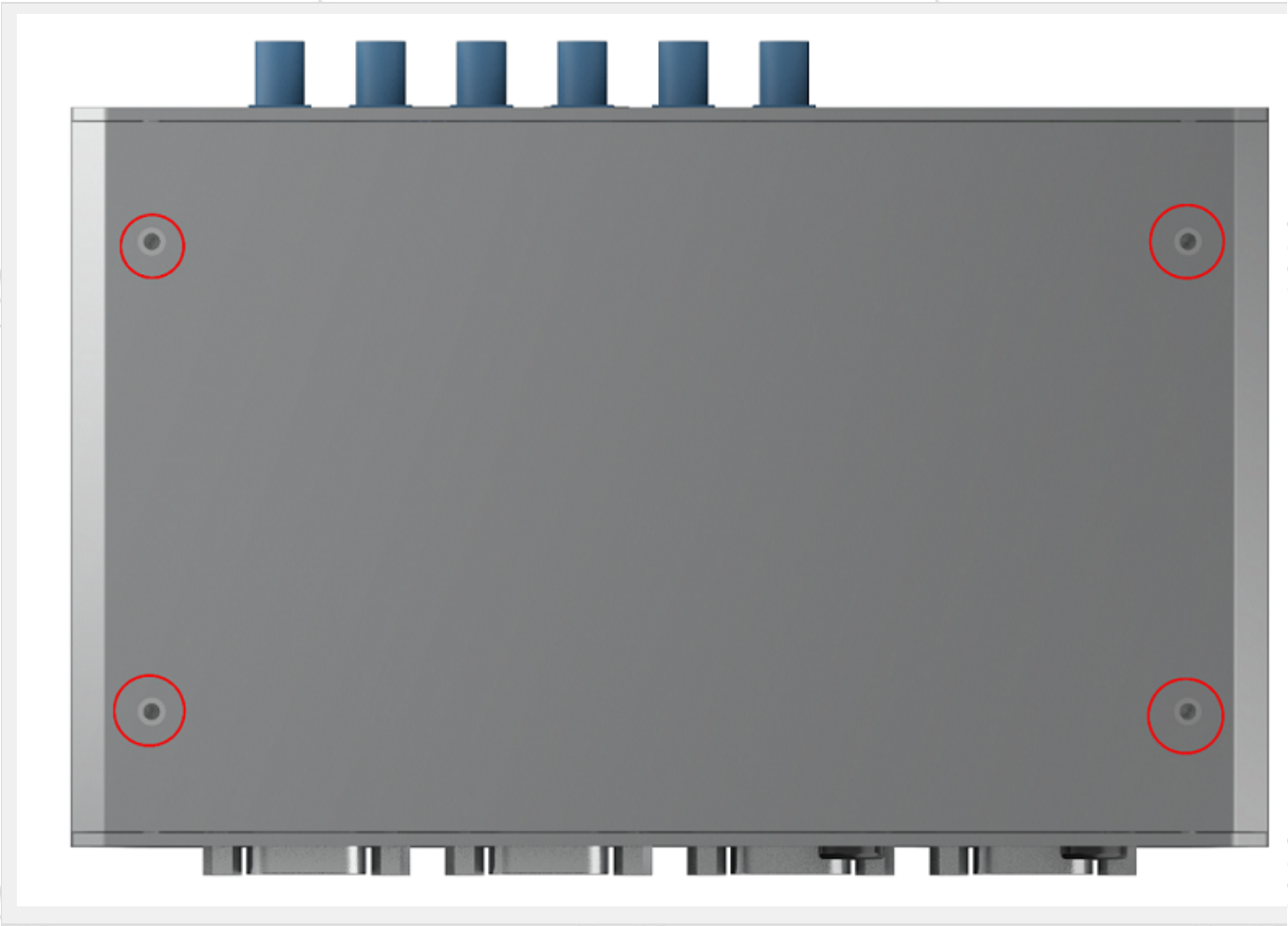


图 安装固定板示意图

通过安装固定板上的孔位，根据实际情况将米文设备固定到其他设备上

图 EVO TX2 GMSL2固定示意图

Mini PCIe4G支持清单

序号	品牌	产品型号	支持方式	使用接口	模块功能	工作温度	规格	备注
1	移远	EC20-CEHCLG-MINIPCIE-CB	正式	Mini PCIe	4G	-40ºC 至 80ºC	**规格**全网通**速度**最大130Mbps (下载)/最大30Mbps (上传)	
2	移远	EC20-CEHC-MINIPCIE-CB	Beta	Mini PCIe	4G	-40ºC 至 80ºC	**规格**全网通**速度**最大130Mbps (下载)/最大30Mbps (上传)	
3	移远	EC20-CEHCLG-MINIPCIE-C	Beta	Mini PCIe	4G	-40ºC 至 80ºC	**规格**全网通**速度**最大130Mbps (下载)/最大30Mbps (上传)	

注:

1. 正式支持：每次米文系统版本升级，会在米文设备上进行验证。
2. BETA支持：米文调试过，但不会在每次米文系统版本升级中验证，如使用过程中需要进一步支持请联系对应的销售工程师或客户经理。

功能介绍

通用使用方法

系统介绍

米文设备采用Ubuntu系统。默认用户名：nvidia； 密码：nvidia

烧写镜像

请访问米文技术论坛http://forum.miivii.com/来获取烧写工具，烧写工具说明及相应镜像。

开关机

开机：米文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接口与米文设备相连，开机画面如图所示：

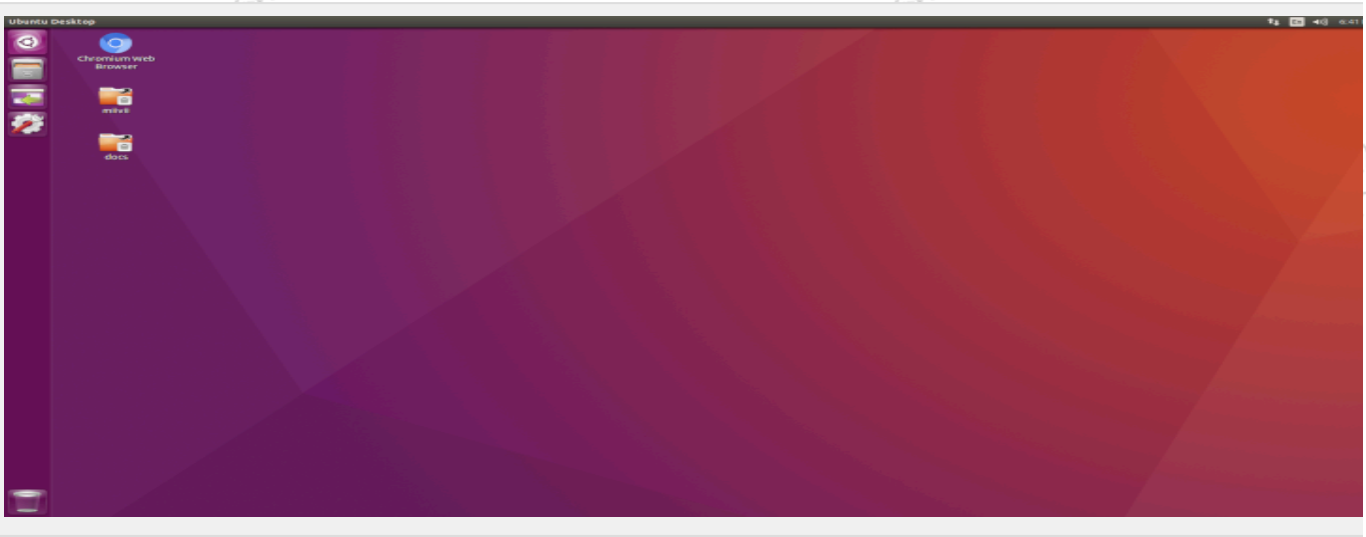


图 开机画面

关机：长按POWER键/ON KEY按钮关机。或在命令行中执行$ sudo poweroff，完成软关机 重启：在命令行中执行$ sudo reboot，完成重启

MIIVII SETTINGS的使用说明

简介

MIIVII SETTINGS（又称米文设置），是米文为了简化对于设备进行设置，而提供的工具。

提供诸如系统状态检测、远程访问、远程登陆等等功能。

Jetpack 4.4版本及以下镜像MIIVII SETTINGS的使用说明

使用视频

请参考：https://doc.miivii.com/pages/viewpage.action?pageId=5275982

登陆

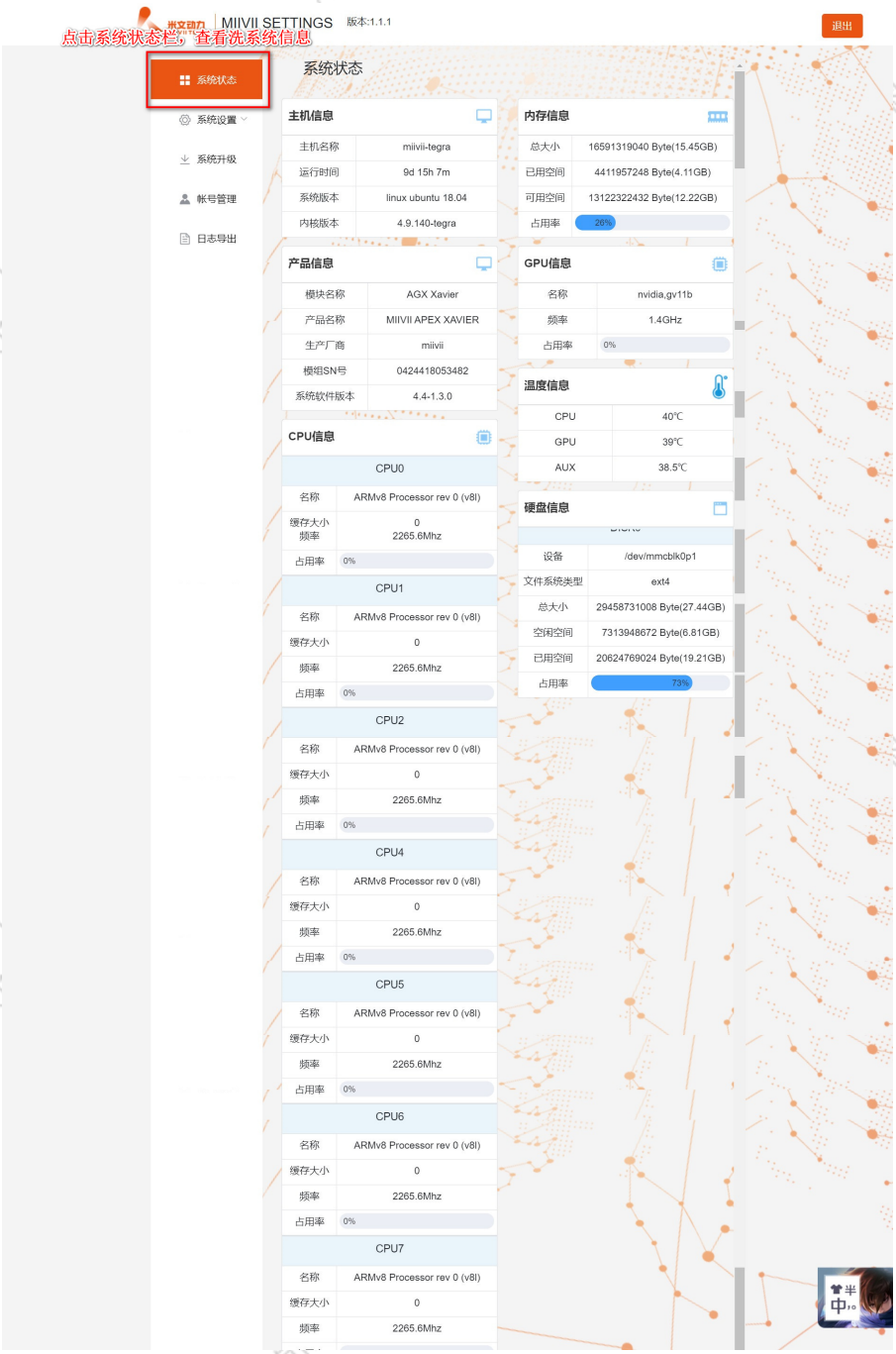
默认端口号为3000

用户名密码，为系统用户名密码。具有sudo权限的用户方可登录。无sudo权限用户无法登陆。

功能说明

系统状态

用于查看当前系统CPU占比、内存占比，存储占比等基本信息。



所有设备配置软件均含有基本信息显示功能，在此可以看到设备的系统版本，也可以通过命令行查看系统版本。

```
cat /etc/miivii_release
APEX 4.2.2-1.5.0
```

系统设置

对系统基本功能进行设置，如系统授时设置，GMSL相机设置等等。

- GMSL设置

Apex Xavier以及S2Pro包含GMSL摄像头设置功能，

Apex Xavier II, EVO TX2 GMSL2包含GMSL2摄像头设置功能，

EVO Xavier, EVO TX2以及S2不包含GMSL/GMSL2摄像头设置功能。

该功能可以按照GMSL/GMSL2摄像头的接插位置，分别设定摄像头的品牌。

设备Apex有两组GMSL摄像头接口，分别记为GMSL_A与GMSL_B,

设备S2Pro只有一组记为GMSL_A,

设备EVO TX2 GMSL2有6个独立GMSL2摄像头接口，记为GMSL_0-5

设备Apex Xavier II有8个独立GMSL2摄像头接口，记为GMSL_0-7

GMSL摄像头配置方法

在初次接入GMSL摄像头以及更换GMSL摄像头型号时需要对配置文件进行更改，并重启设备。配置文件路径：/opt/miivii/config/gmsl_camera/camera.cfgMVGCB-001A： EntronMVGCB-002A： CalmcarMVGCB-003A：AdayoMVGCB-006A：Sensing默认配置：A组和B组摄像头接口默认配置都是MVGCB-001A

- 系统授时设置

Apex Xavier, Apex Xavier II, EVO Xavier, EVO TX2 GMSL2, S2Pro包含同步功能设置, , EVO TX2 ,S2不包含同步功能设置。

其中，NTP为默认模式。NTP网络授时模式，此时设备接入网络，可以被NTP服务授时。同时，设备可作为同步源，外接支持同步功能的传感器，并进行同步；GPS为GPS外界授时模式，此时设备外接GPS，可以被GPS授时。同时，设备可作为同步源，外接支持同步功能的传感器，并进行同步；None为不同步模式，此时设备不被外界授时，但可以作为同步源。同时，也可在此调节Sync out输出频率，注意此处并非是GMSL的频率。

设备授时模式与Sync out输出频率配置方法设备授时模式与Sync out输出频率的调整需要对配置文件进行修改，并重启设备。配置文件路径：/opt/miivii/config/sync/sync.cfg授时模式是通过修改其中"sync_type:X" 的X数值来实现。0：GPS外界授时模式1：NTP网络授时模式2：不同步模式Sync out输出频率通过修改其中"sync_out_freq:XX"的 XX数值实现Sync out频率调节。该调节仅支持整数。

```
cat /opt/miivii/config/sync/sync.cfg
sync_out_freq:25
sync_type:2
/*
note:
sync_out_freq---the frequency is 25 for sync out time
sync_type---0 is for GPS calibrate time
1 is for SYS calibrate time
2 can not calibrate time
```

- 远程访问设置

系统升级

对cuda相关包，以及miivii提供的相关包和系统进行升级及回退。

账号管理

用于绑定米文边缘服务。

日志导出

用于导出/var/log/中的日志，便于进行售后分析。

功率模式设定

搭载Jetson TX2的米文设备有多工作模式。可以通过右上角的NVIDIA绿色标志设置进行调整。 米文设备的默认模式为2：MAXP CORE ALL

图 设置图标

点击下拉菜单即可对米文设备的工作模式进行修改，工作模式的细节详见下表：

Mode	Mode Name	Denver2	Frequency	ARM A57	Frequency	GPU Frequency
0	Max-N	2	2.0GHz	4	2.0GHz	1.30 GHz
1	Max-Q	0		4	1.2GHz	0.85 GHz
2	Max-P Core-All	2	1.4GHz	4	1.4GHz	1.12 GHz
3	Max-P ARM	0		4	2.0GHz	1.12 GHz

也可采用命令行调整：

```
#
sudo nvpmodel -q verbose
#
sudo nvpmodel -m <MODE ID>
#
sudo jetson_clocks
#
sudo jetson_clocks --show
```

IO使用方法

GPIO接口配置方法

对GPIO接口使用的示例如下，请将< >中的信息修改为想要调整的GPIO节点号，具体对应关系请参考【接口说明】部分

```
# root
sudo su -
# (DO)
echo 1 > /sys/class/gpio/<gpio339>/value
# (DO)
echo 0 > /sys/class/gpio/<gpio339>/value
# (DI)
cat /sys/class/gpio/<gpio339>/value
```

若需要关机后保留配置，可以将以上命令写入/etc/rc.local 文件

注：GPIO外接方式说明 DO为开漏输出（开漏输出就是不输出电压，控制输出低电平时引脚接地，控制输出高电平时引脚既不输出高电平，也不输出低电平，为高阻态。如果外接上拉电阻，则在输出高电平时电压会拉到上拉电阻的电源电压） 设置为高电平时，DO脚与外接的电压相同（0V~40V）； 设置为低电平时，DO脚为地。

UART接口配置方法

打开/dev/(folder)下面对应的设备节点，设置波特率，停止位，奇偶校验位，数据位等。可以使用stty命令配置串口的波特率，停止位，奇偶校验位，数据位等，详细见stty命令说明。

命令示例，请将< >中的信息修改为想要调整的串口节点号，具体对应关系请参考【接口说明】部分

```
sudo stty -F /dev/<UART_XXX> speed 115200 cs8 -parenb -cstopb -echo
```

输出数据测试

```
sudo echo “miivii tty debug” > /dev/<UART_XXX>
```

使用下面命令接收输入数据

```
sudo cat /dev/<UART_XXX>
```

GPS 对设备授时使用方法

GPS对设备授时功能优点：设备通过GPS设备从GPS卫星上获取当地标准的时间信号，从而精准定位设备时间。

GPS支持型号

串口支持修改波特率，默认波特率为9600

支持GPS品牌型号：所有符合GPRMC数据标准格式输出的GPS设备，且必须要有PPS秒脉冲输出的GPS设备

连接方式

```
“”
```

授时功能配置

在初次接入GPS时需要在MiiVii Setting配置软件中进行系统配置，将Sync Mode选项配置成GPS模式，重启系统。MiiVii Setting具体方法请参考"米文配置软件介绍"部分。

检查授时是否成功

修改系统时间，输入命令

```
sudo date -s "2018-10-1"
```

等待2~3s，查看当前时间，输入命令

```
date
```

若显示时间为："2018-10-1"，说明授时失败

若显示时间为："当前时间"，说明授时成功

故障排查

若授时失败，需进行故障排查

1.查看GPS是否有输出

输入命令

```
cat /dev/ttyTHS1
```

终端收到带有GPRMC字段的输出，例如：

GPRMC,014600.00,A,2237.496474,N,11356.089515,E,0.0,225.5,310518,2.3,W,A*23

2.查看GPS的pps信号是否有输出

输入命令

```
hexdump /dev/miivii-sync-in-a
```

终端有十六进制的数据输出，例如：

0000400 02fe 9f40 490e 562d 1647 004e 0000 0000

3.识别方法

如果以上"1"&"2"没有输出，说明GPS工作不正常，可以把GPS放到窗外或是到户外测试，或更换GPS进行测试

如果"1"&"2"输出正常，检查MiiVii Setting配置是否为GPS模式，如果不是，更改模式后重新启动

执行以上操作之后，GPS授时依然不成功，输入命令

```
hexdump /dev/miivii-sync-out
```

终端有十六进制的数据输出，例如：

0000400 02fe 9f40 490e 562d 1647 004e 0000 0000

如果没有数据输出，可能是没有用匹配的刷机工具和镜像刷机，建议检查镜像和刷机工具重新刷机

如果有数据输出，可能是设备硬件问题，建议联系售后维修处理

CAN口配置方法

CAN10设备具体使用方法，参考https://github.com/linux-can/can-utils中的cansend.c和candump.c

测试命令：

```
sudo modprobe can

sudo modprobe can_raw

sudo modprobe mttcan

sudo ip link set can0 type can bitrate 500000 sjw 4 berr-reporting on
loopback off

sudo ip link set up can0

sudo cansend can0 123#abcdabcd

sudo candump can0

sudo ip -details -statistics link show can0

sudo ifconfig can0 down
```

CAN FD配置使用方法：

```
sudo modprobe can

sudo modprobe can_raw

sudo modprobe mttcan

sudo ip link set can0 type can bitrate 500000 sjw 4 dbitrate 2000000 dsjw
4 berr-reporting on fd on

sudo ip link set up can0

sudo cansend can0 213##011
```

[10] CAN FD和CAN 2.0的区别：

1)

```
sudo ip link set can0 type can bitrate 500000 dbitrate 2000000 berr-
reporting on fd on
```

其中bitrate为can2.0模式下的波特率； dbitrate为can fd模式下的波特率，根据官方文档，这个值最大可配置为5M，一般应用最好采用2M;

2)

```
sudo cansend can0 213##011
```

发送命令中，id与数据之间多了一个#，并且##后的第一个字节(0)为canfd_frame.flags的值，范围为0~F； canfd_frame.flags后面的字节(11)为第一个数据，一次最多可以传输64个字节。

扩展设备配置方法

扩展SSD硬盘使用

查看硬盘信息:

```
sudo fdisk -lu
```

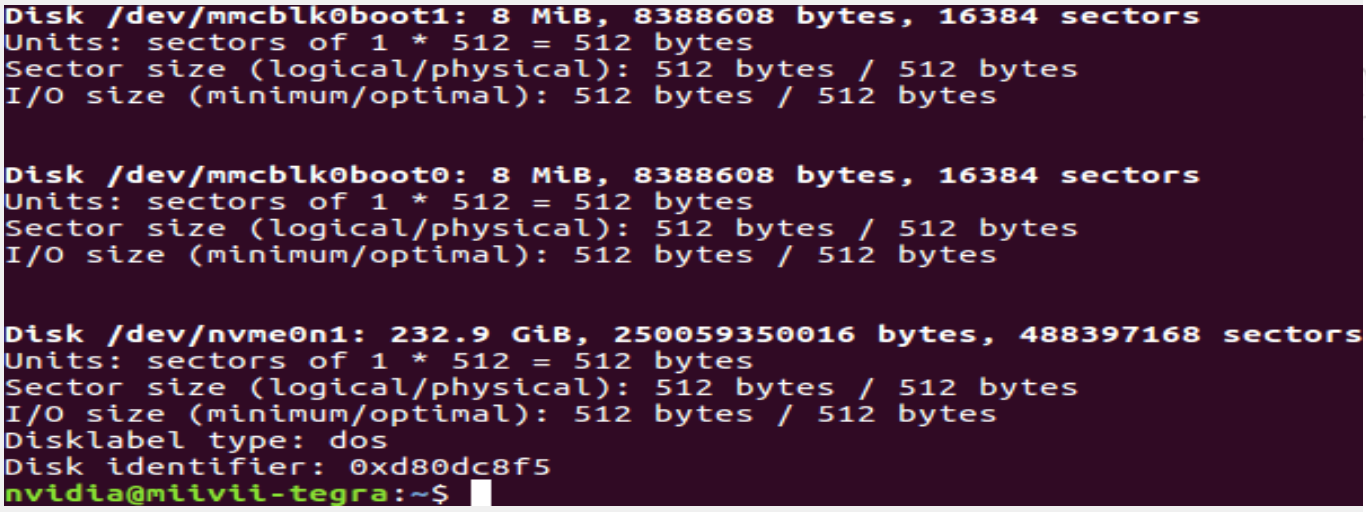


图 查看硬盘信息页面截图

格式化硬盘:

```
sudo mkfs -t ext4 /dev/nvme0n1
```

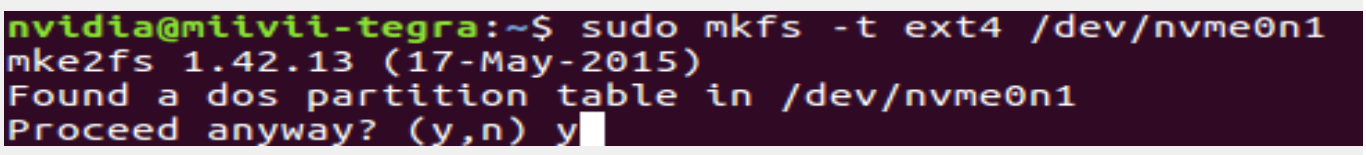


图 格式化硬盘截图

查看硬盘UUID:

```
sudo blkid /dev/nvme0n1
```

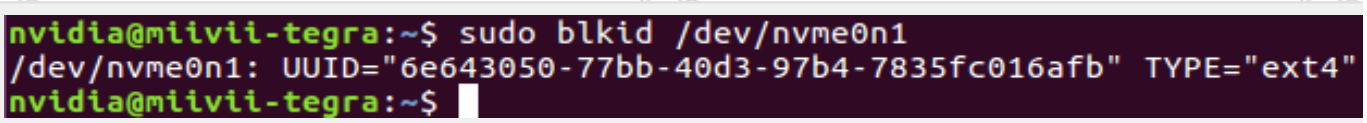


图 查看硬盘UUID 截图

开机自动挂载硬盘的设置方法： **在/etc/systemd/system路径下创建一个systemd服务，用来开机自动执行挂载硬盘，如：** miivii_mount_ssd.service

```
#miivii_mount_ssd.service
vim miivii_mount_ssd.service
[Unit]
Description=MIIVII specific script
After=udev.service

[Service]
ExecStart=/etc/systemd/miivii_mount_ssd.sh

[Install]
WantedBy=multi-user.target
```

在/etc/systemd/路径下创建一个脚本，用来挂载硬盘，如： miivii_mount_ssd.sh

```
#miivii_mount_ssd.sh
vim miivii_mount_ssd.sh
#!/bin/bash
mount -o rw /dev/nvme0n1 /home/nvidia/workspace
```

为创建的脚本文件添加可执行权限

```
sudo chmod +x miivii_mount_ssd.sh
```

将挂载硬盘的服务设置为开机自启动

```
sudo systemctl enable miivii_mount_ssd.service
```

无线设备配置方法

WiFi配置方法

米文S2，S2Pro，EVO TX2 ，EVO TX2 GMSL2自带WiFi功能。米文Apex Xavier，Apex Xavier II，EVO Xavier, Lite NX，Lite Nano的WiFi功能由外接扩展模块提供，请按照【扩展设备安装方式】的内容对WiFi模块进行安装。请在开机Ubuntu系统桌面右上角网络连接图标中，找到要连接的WiFi名称并点击，然后在弹出的密码框输入密码并点击连接即可。

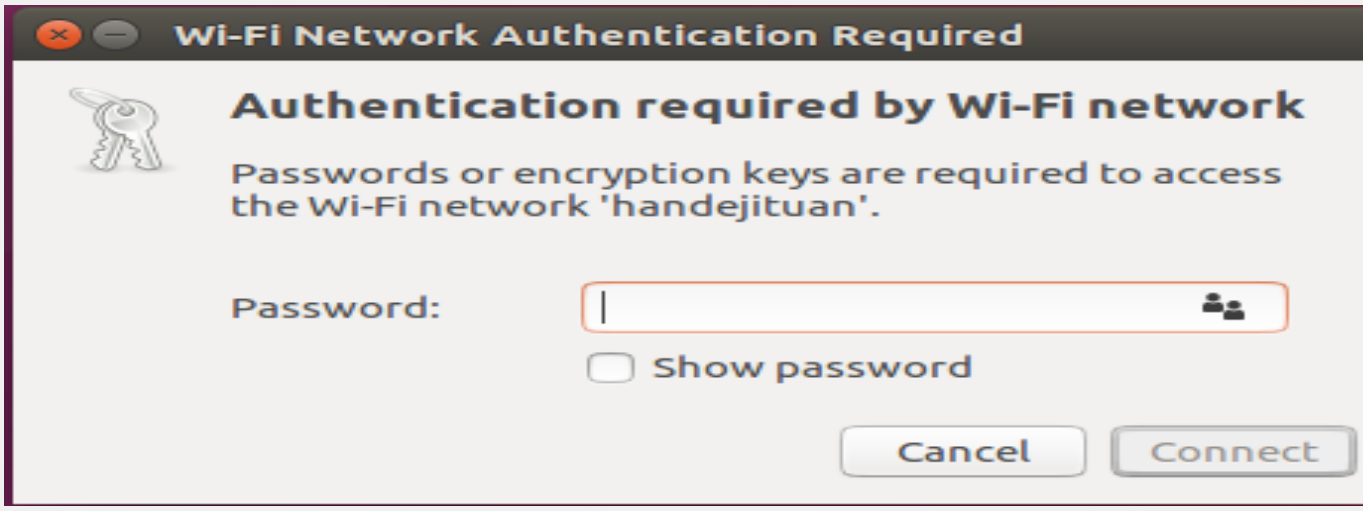


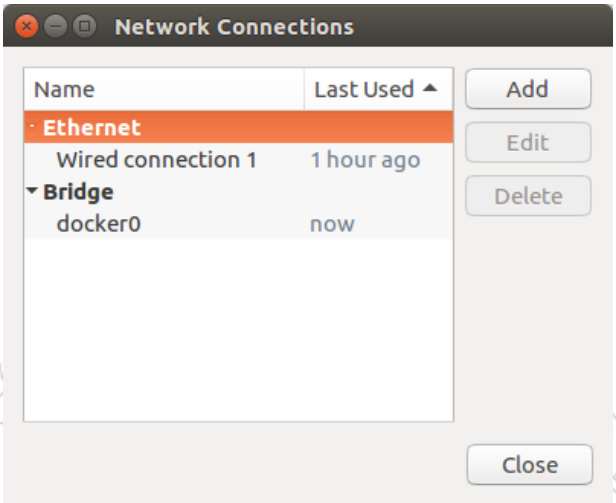
图 WiFi连接截图

4G模块配置方法

米文所有标准产品中不包含4G模块，需要用户自行另配。请按照【扩展设备安装方式】的内容对SIM卡以及4G模块进行安装。请注意如果您使用的是物联网SIM卡，则会出现SIM卡与设备硬件绑定的问题，请提前与通讯供应商确认。

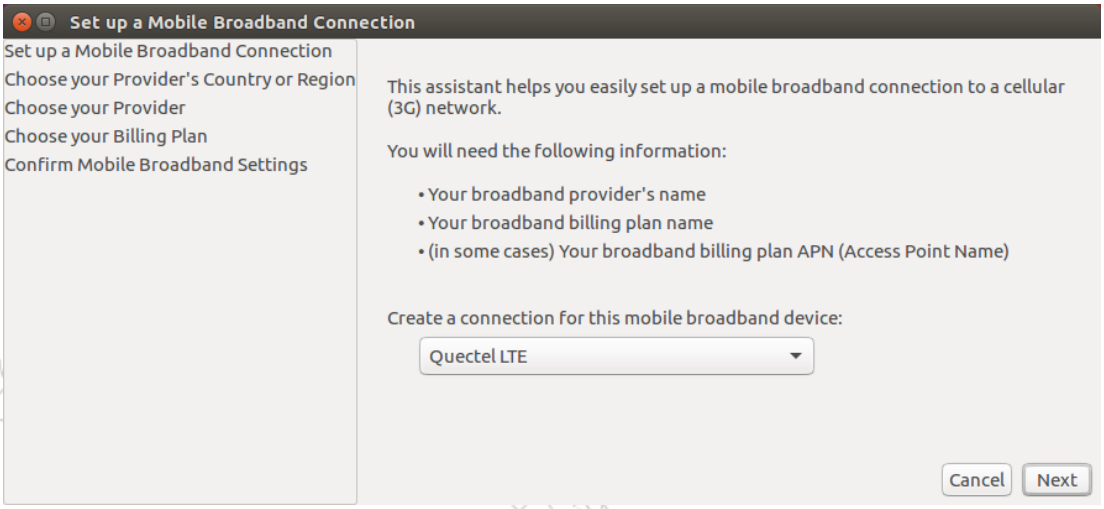
米文的系统镜像中，整合了相应4G模块驱动。安装好4G模块后系统会自动识别。查看/dev目录，会看到/dev/ttyUSB0~/dev/ttyUSB3，一共4个设备。

在桌面右上方网络连接图标中，找到Edit Connections，点击add，如图所示：

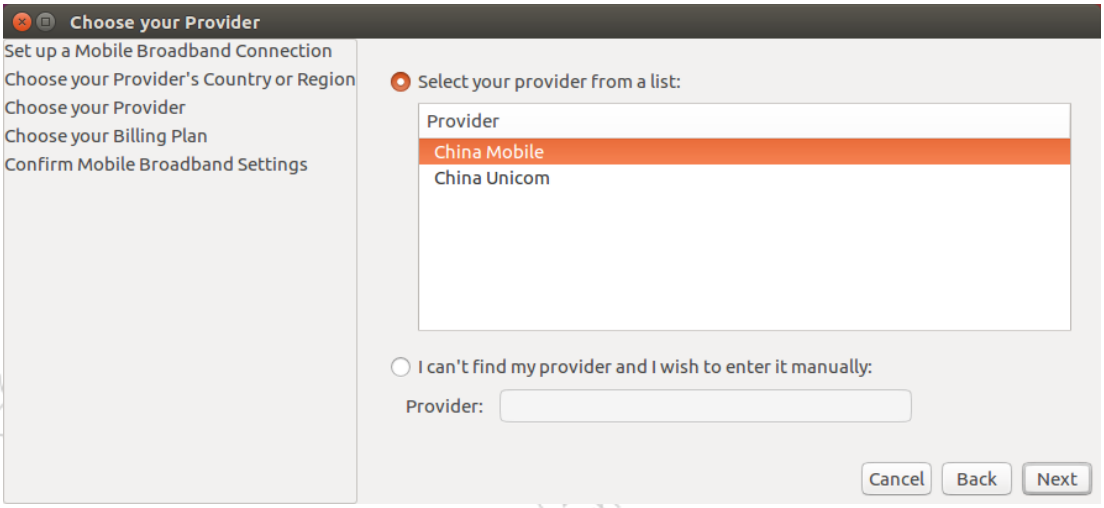


选择连接类型 Mobile Broadband

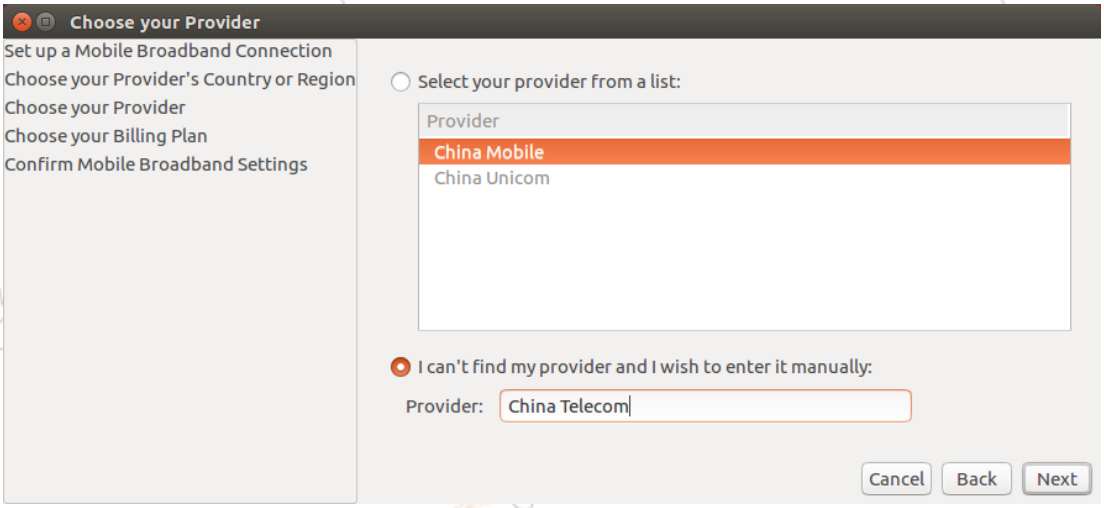
选择Next（选项Quectel LTE，Fibocom NL668 Modem，Android与Any deviceetc依据不同型号的4G模块显示不同信息，可直接点击Next）



选择国家为China，然后根据SIM卡选择运营商：中国移动是China Mobile，中国联通是China Unicom



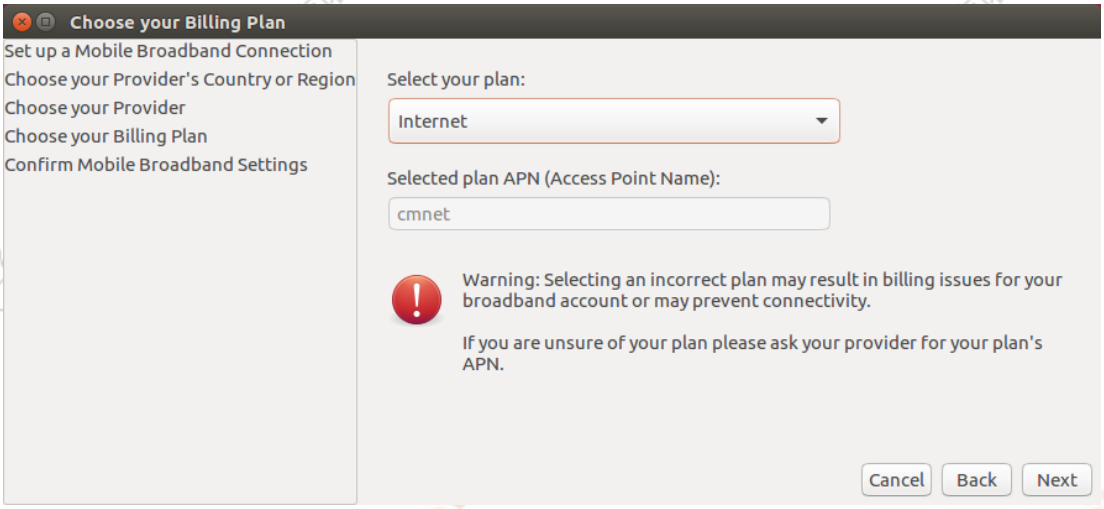
如果选择的运营商是中国电信，则点击手动新建运营商 China Telecom



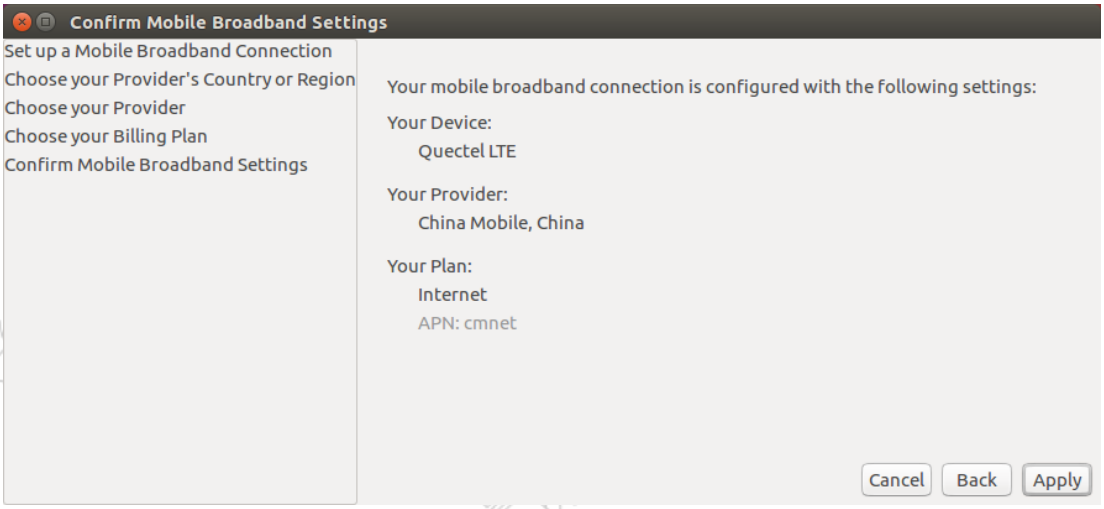
选择你的Plan

请根据SIM卡信息选择。移动选Internet，联通和电信选default

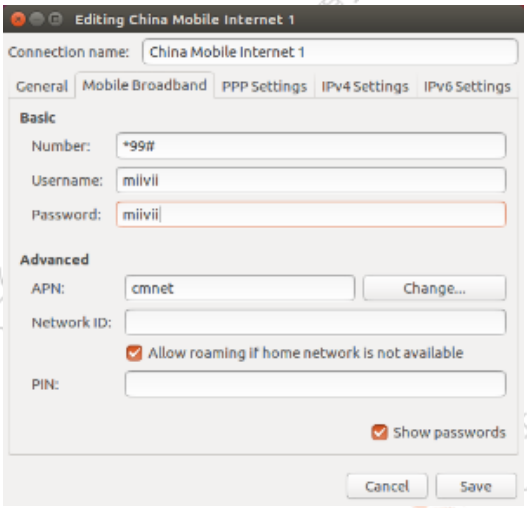
这里注意一下APN移动为cmnet，联通为3gnet，电信为ctnet



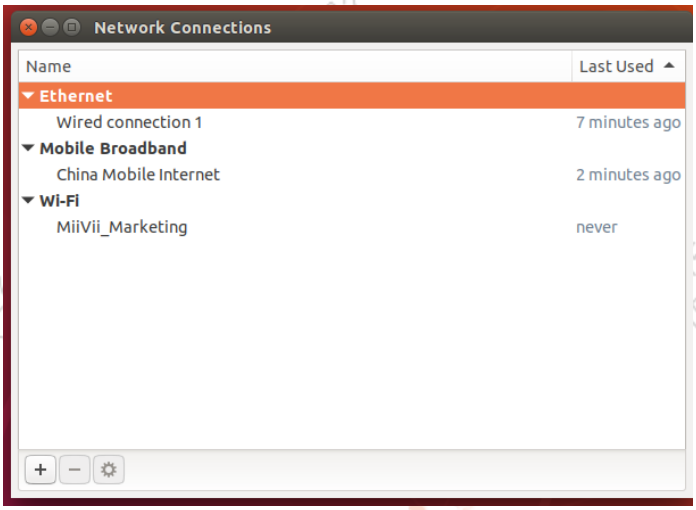
检查已创建的信息，如正确无误，则点击Apply



设定用户名和密码，点击save

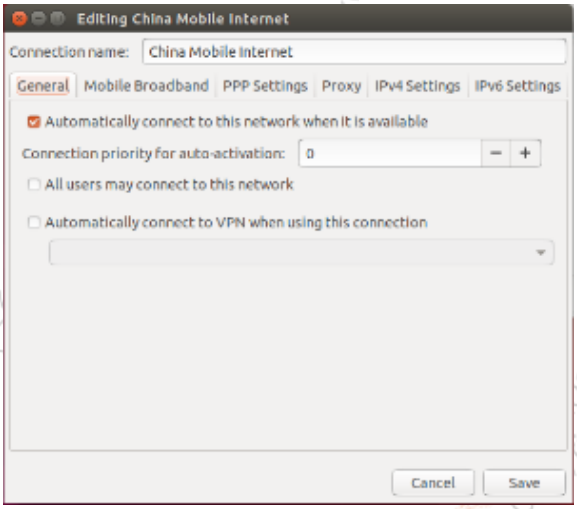


网络创建完成后，在桌面右上方网络连接图标中选中新建的网络连接，就能够正常上网了。若需要4G网络开机自动连接设置，已移动为例，建立好连接文件China Mobile Internet后操作如下： 点击桌面上方网络连接图标，在下拉菜单中点击Edit Connections选项。在弹出的窗口中选中China Mobile Internet选项，点击下方设置图标



弹出的窗口中选择General选项，并勾选Automatically connect to this network when it is available选项

之后保存退出。重启米文设备，就可以在输入开机密码后自动重连4G网络



同步功能使用说明

同步功能介绍

设备支持三种同步方式，分别是：PPS，Sync in和Sync out同步。同步误差为0.1-1s。（同步误差测试方法详见附录）

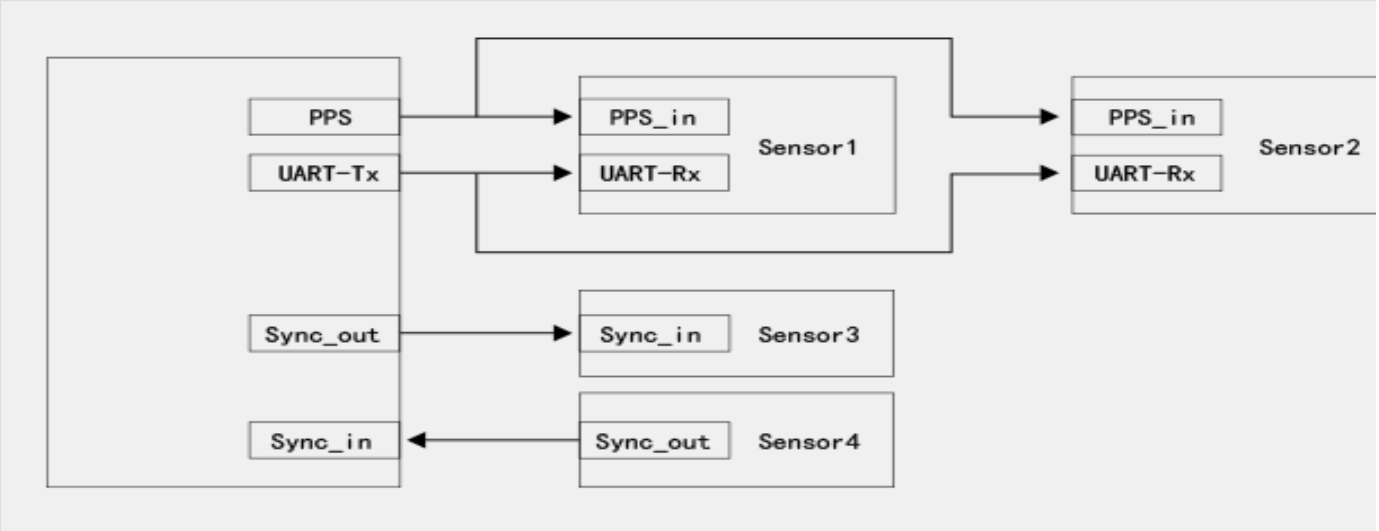


图 设备同步接线原理图

同步功能使用方法

PPS同步模式

设备输出PPS信号[1]（每秒产生一个脉冲，脉宽50ms），并通过串口（UART/RS232）的Tx引脚发送该脉冲上升沿产生时间的NMEA GPRMC消息，消息示例：

```
$GPRMC,060249.000,A,3949.63046,N,11616.48565,E,0.296,,291118,,,A*4d
```

[1] PPS 信号的硬件连接方式请见"IO转接线说明"中的"PPS连接线及引脚定义"部分

其中"060249.000"为每秒产生脉冲时的时间戳(UTC时间)，格式为"时分秒.000"，正常时间都是整秒格式。支持PPS同步模式的传感器会通过收到的PPS以及GPRMC消息对自身时钟系统进行校时，使之与设备的系统时钟保持一致。传感器的采样时间会作为时间戳（timestamp），与数据一起被发送至设备。至此，系统获取了传感器采样的系统时间，完成同步。

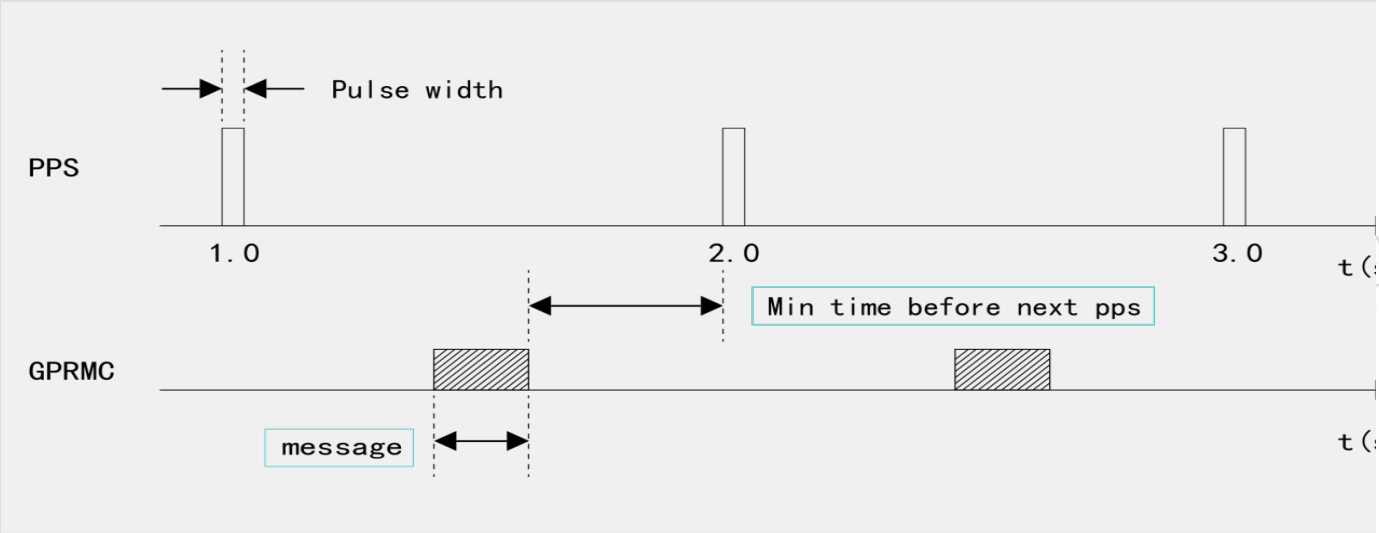


图 PPS同步原理图

同步功能验证方法（以RS-LiDAR-16激光雷达为例）：

当传感器只有数据输入接口与设备相连，未连接设备的PPS_A_SYNC口和PPS_A_TX时，传感器的ROS node向操作系统上传数据中的时间戳为硬件时间戳（hardware timestamp），即传感器内部时钟计器的时间（多数传感器会设定一个固定的初始时间作为计时起点，每次上电后开始计时）。此时在Ubuntu操作系统中打印该硬件时间戳，并与设备的系统时间进行比较，可发现二者的偏差较大。

```
[ INFO] [1544594922.929736448]: Got param time_mode: gps
ros time:1544594922.876364231
lidar time:1483229224.839233000

[ INFO] [1544594923.026582016]: Got param time_mode: gps
ros time:1544594922.977174997
lidar time:1483229224.940033000

[ INFO] [1544594923.134761184]: Got param time_mode: gps
ros time:1544594923.078029156
lidar time:1483229225.040833000

[ INFO] [1544594923.229261088]: Got param time_mode: gps
ros time:1544594923.178819418
lidar time:1483229225.141632000

[ INFO] [1544594923.332589472]: Got param time_mode: gps
ros time:1544594923.279591560
lidar time:1483229225.242432000
```

图 RS-LiDAR-16未同步时硬件时间戳与系统时间戳对比

当传感器连接设备的PPS_A_SYNC及PPS_A_TX后，传感器的ROS node向操作系统上传数据中的硬件时间戳为传感器内部时钟经过PPS授时后的时间，与设备的系统时间一致。此时在Ubuntu操作系统中打印接收到的硬件时间戳，并与收到该数据时的系统时间（ros::time::now）比较，当二者的差值小于100ms时，说明PPS功能生效。

```
[ INFO] [1544601870.404294176]: Got param time_mode: gps
ros time:1544601870.349020720
lidar time:1544601870.347184000

[ INFO] [1544601870.503913024]: Got param time_mode: gps
ros time:1544601870.449862003
lidar time:1544601870.447984000

[ INFO] [1544601870.602763072]: Got param time_mode: gps
ros time:1544601870.550686121
lidar time:1544601870.548784000
```

图 数据时间戳与系统时间对比

Sync out 同步模式

设备支持Sync out同步信号[2]
[2] Sync out信号的硬件连接方式请见"IO转接线说明"中的"SYNC连接线及引脚定义"部分

设备可以通过Sync out引脚输出1-30Hz，脉宽5ms的脉冲信号，用于触发外部传感器启动采样。同时设备会记录该脉冲上升沿的产生时间。传感器完成采样后，设备会将记录的时间与本次传感传回的数据做关联，作为该数据的时间戳，至此，系统获取了传感器采样的系统时间，完成同步。

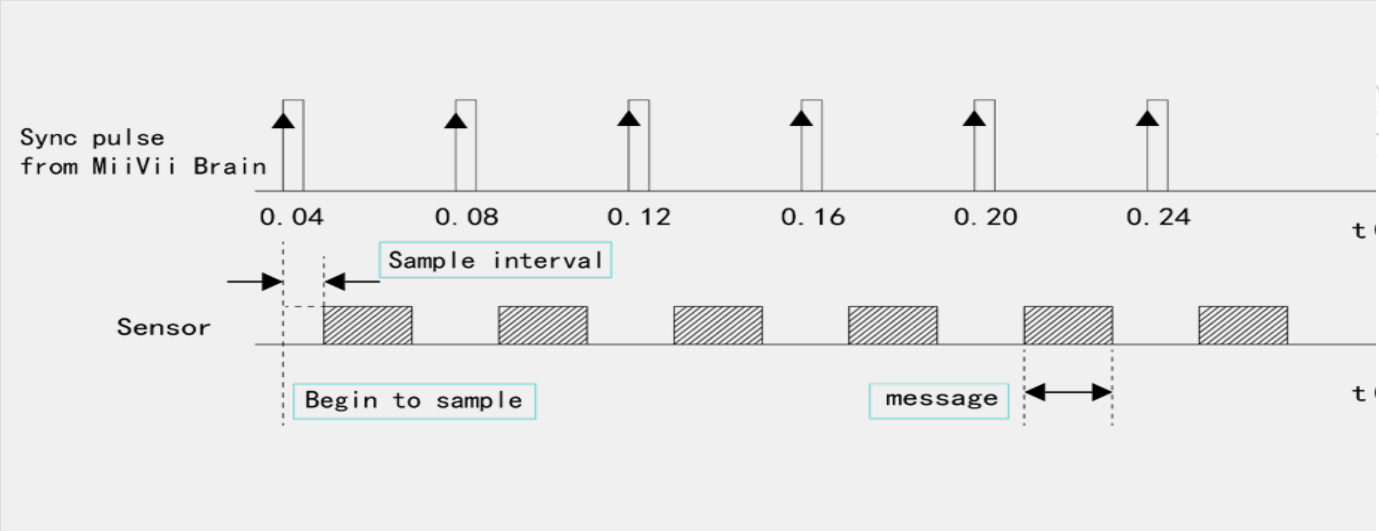


图 Sync out同步原理图（25Hz）

与此同时，设备还提供GMSL接口的Sync out同步功能，详见GMSL摄像头章节

同步功能验证方法： 配置传感器为外部触发同步模式，通过ROSbag抓包确认传感器触发频率是否为sync.cfg中所设定的频率值。如果偏差小于1Hz，则说明Sync out功能生效。

数据传输时间分析

测试说明：设定Sync out 发出信号的频率为10Hz，测量设备发出的Sync out信号的上升沿到设备接收到视频帧之间的时间间隔（transfer time）。

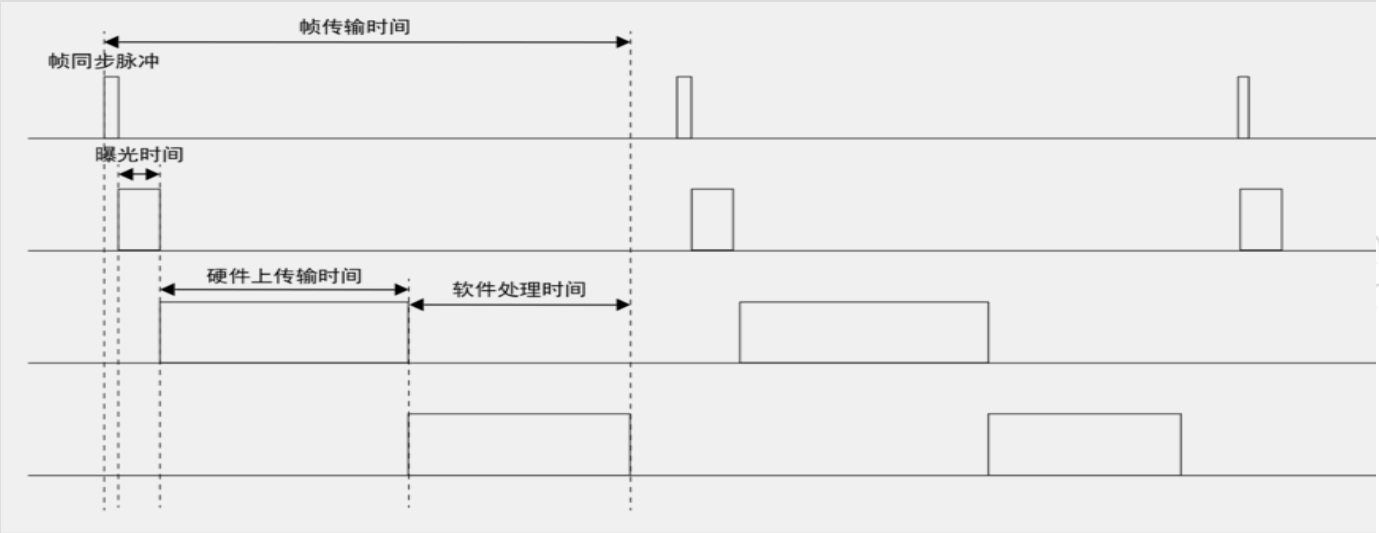


图 IO帧传输时序示意图

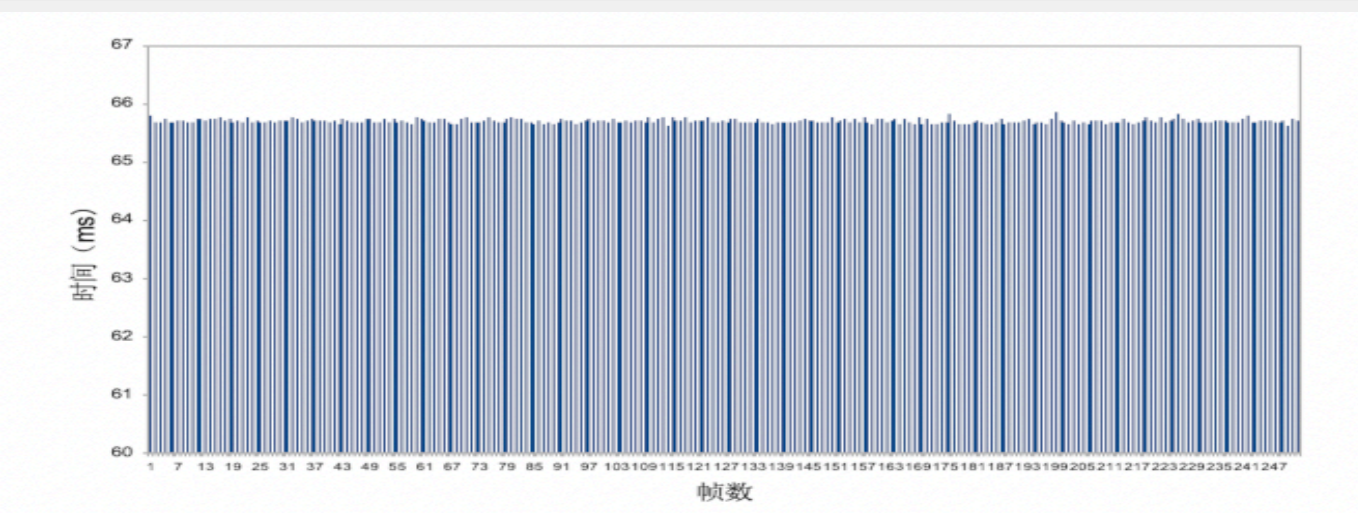


图 帧传输时间测试结果

测量结果发现帧传输时间的平均值为65.70ms。

Sync in 同步模式

设备支持Sync in同步信号[3]。
[3] Sync in同步信号的硬件连接方式请见"IO转接线说明"中的"SYNC连接线及引脚定义"部分

支持Sync in同步模式的传感器，在启动采样的时刻会产生并发出一个脉冲信号。设备通过SYNC_IN引脚接收脉冲信号，并记录该脉冲上升沿的产生时间。传感器完成采样后，设备会将记录的时间与本次传感传回的数据做关联，作为该数据的时间戳。至此，系统获取了传感器采样的系统时间，完成同步。

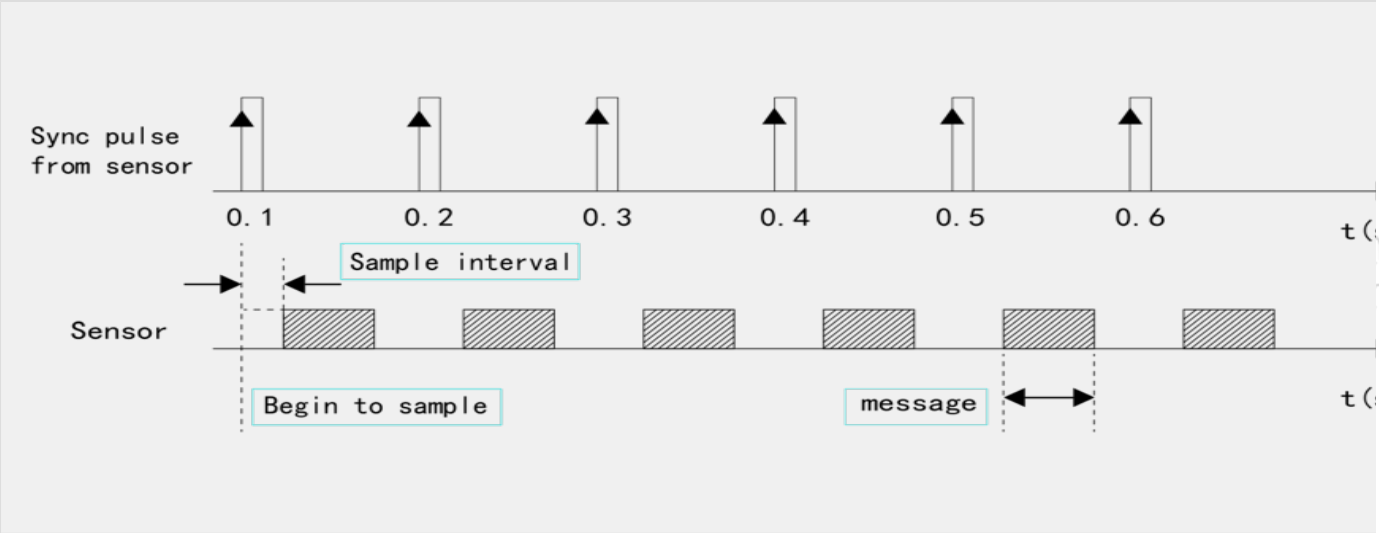


图 Sync in同步原理图(10Hz)

同步功能验证方法： 在Ubuntu操作系统中打印SYNC_IN引脚接收到脉冲信号的时间戳，将该时间戳与收到传感器数据帧的系统时间（ros::time::now）相比较，如果二者的差值小于100ms，说明Sync in功能生效。

同步误差测试方法

通过示波器测量PPS脉冲间隔

测量结果：

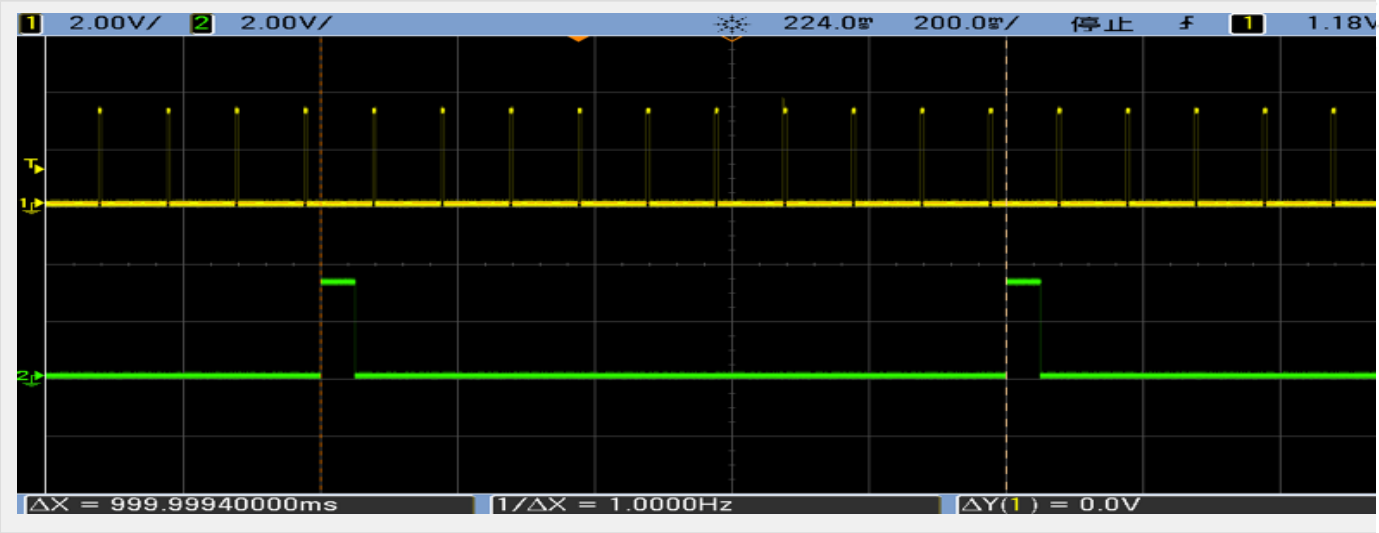


图 PPS脉冲间隔

	理论值（s）	实测值（s）	误差（s）
PPS	1000000	999999.4	0.6

通过示波器测量Sync out脉冲间隔

用示波器测量两个Sync out（10Hz）脉冲之间的间隔并与理论值做比较。

测量结果：

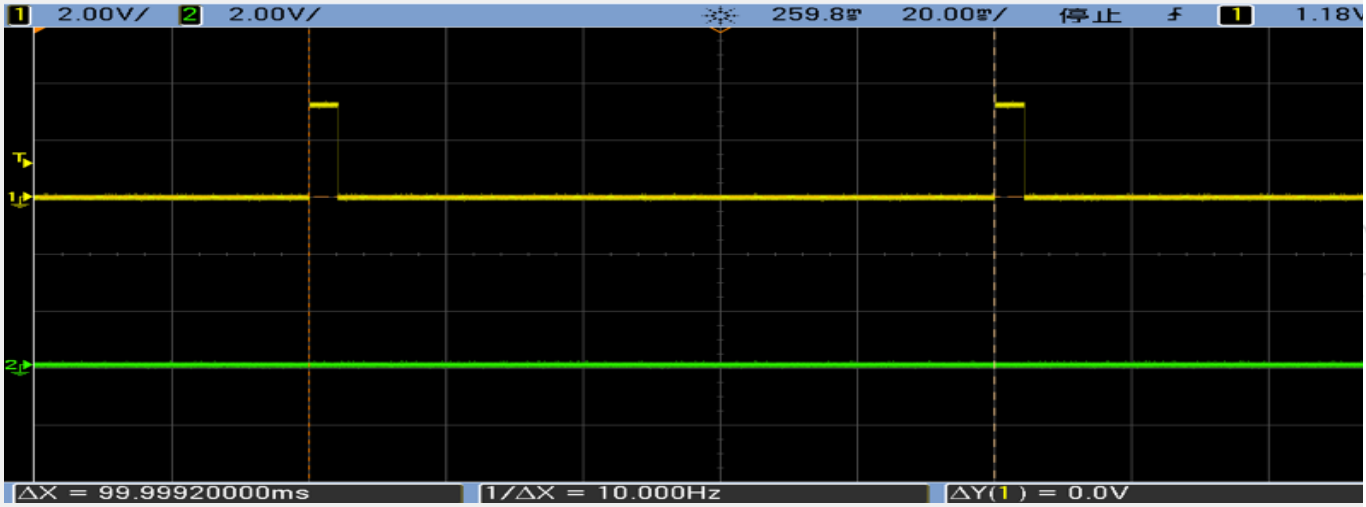


图 Sync out脉冲间距

	理论值（s）	实测值（s）	误差（s）
Sync out	100000	99999.2	0.8

自行评估同步效果的方法

用户可以通过时间戳测量jitter，来自行评测设备同步效果。

同步sample code使用说明

MiiVii提供的sample code用于自行评估设备同步性能，其使用方法如下:

```
#

cd /opt/miivii/feature/sync_test/bin`

#sync out

./sync_out_test

#sync in

./sync_in_test

#pps

./pps_test
```

Sync out jitter测量

利用MiiVii提供的sample code（sync_out_test），实时分析统计接收到的时间戳（timestamp），得到Sync out信号的频率、周期、平均误差、最大误差、方差等值，并实时打印。

```
You are checking the sync out mode
Time interval: the interval between two frames
Frequency measured : the Frequency measured by the Time interval
Max deviation : the difference between the maximum and the average
standard deviation : calculated by statistical data
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 32(ns) standard deviation: 9.0553
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 32(ns) standard deviation: 9.0553
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 32(ns) standard deviation: 9.0553
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 32(ns) standard deviation: 9.0553
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 32(ns) standard deviation: 9.0553
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 32(ns) standard deviation: 9.0553
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 32(ns) standard deviation: 9.0553
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 32(ns) standard deviation: 9.0553
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 32(ns) standard deviation: 9.0553
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 32(ns) standard deviation: 9.0553
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 32(ns) standard deviation: 9.0553
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 32(ns) standard deviation: 9.0553
```

图 Sync out示例测试结果

Sync in jitter测量

由外部设备（如信号发生器）输出固定周期的脉冲信号，接入到设备的SYNC_IN引脚。再利用MiiVii提供的sample code（sync_in_test）实时分析统计接收到的时间戳（timestamp），得到Sync in信号的：频率、周期、平均误差、最大误差、方差等值，并实时打印。

在没有信号发生器的情况下可以将设备的SYNC_IN与SYNC_OUT的引脚相连，以SYNC_OUT的输出作为25Hz的输入信号。

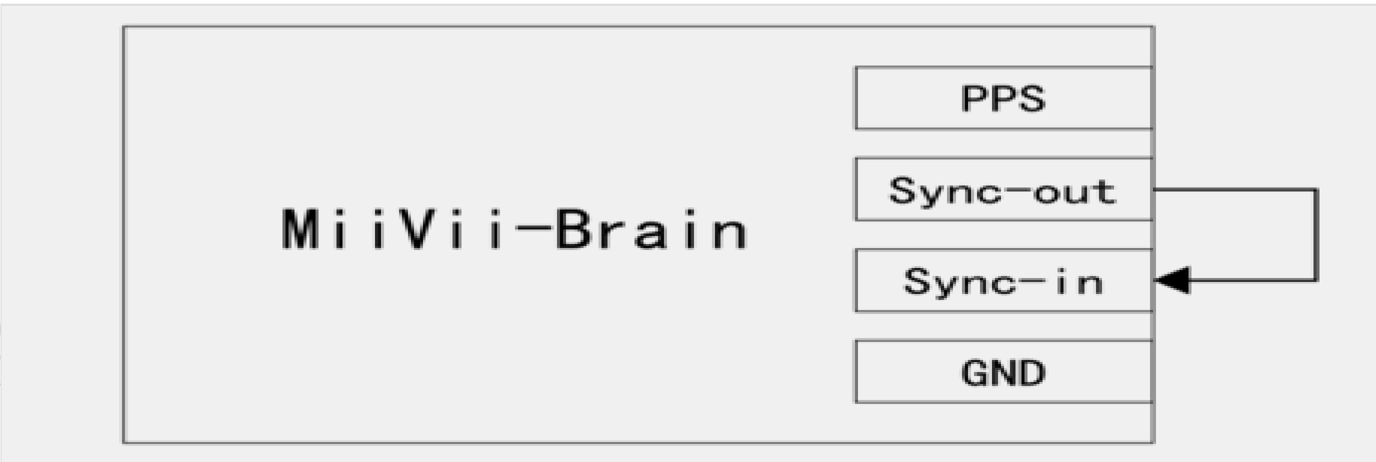


图 IO接线示意图

```
You are checking the sync in mode,pleale check the connecting line
Time interval: the interval between two frames
Frequency measured : the Frequency measured by the Time interval
Max deviation : the difference between the maximum and the average
standard deviation : calculated by statistical data
Time interval: 39999999(ns) Frequency measured: 25.000001(hz)  Max deviation: 193(ns) standard deviation: 76.583
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 193(ns) standard deviation: 65.984
Time interval: 39999999(ns) Frequency measured: 25.000001(hz)  Max deviation: 193(ns) standard deviation: 71.700
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 193(ns) standard deviation: 71.693
Time interval: 39999999(ns) Frequency measured: 25.000001(hz)  Max deviation: 193(ns) standard deviation: 69.942
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 193(ns) standard deviation: 61.269
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 193(ns) standard deviation: 68.673
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 193(ns) standard deviation: 78.115
Time interval: 39999999(ns) Frequency measured: 25.000001(hz)  Max deviation: 193(ns) standard deviation: 66.143
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 193(ns) standard deviation: 77.051
Time interval: 40000000(ns) Frequency measured: 25.000000(hz)  Max deviation: 193(ns) standard deviation: 67.535
```

图 Sync in示例测试结果

PPS jitter测量

将设备的的PPS与SYNC_IN引脚相连，利用MiiVii提供的sample code（pps_test），实时分析统计接收到的时间戳（timestamp），得到PPS信号的频率、周期、平均误差、最大误差、方差等值，并实时打印。

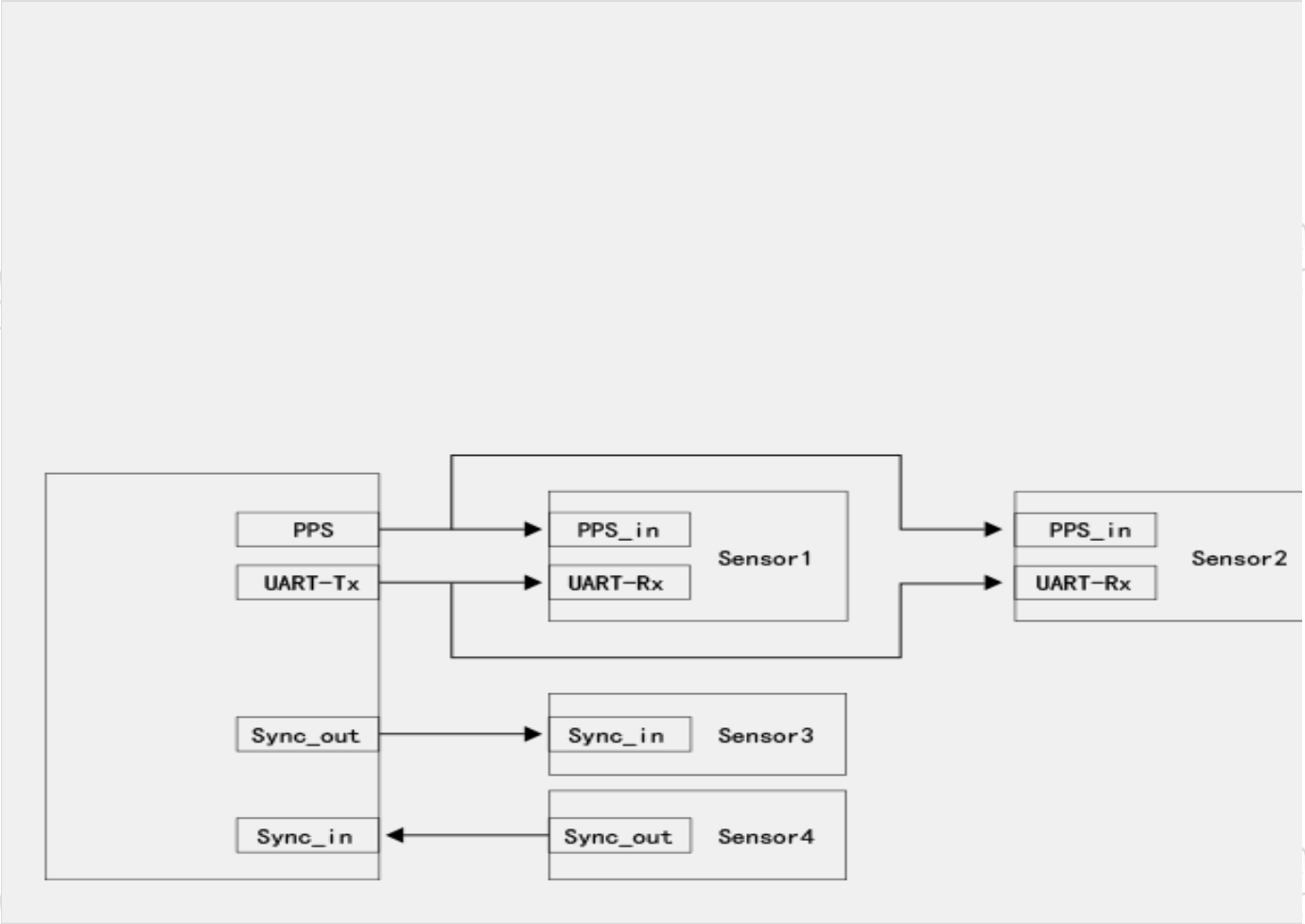


图 IO接线示意图

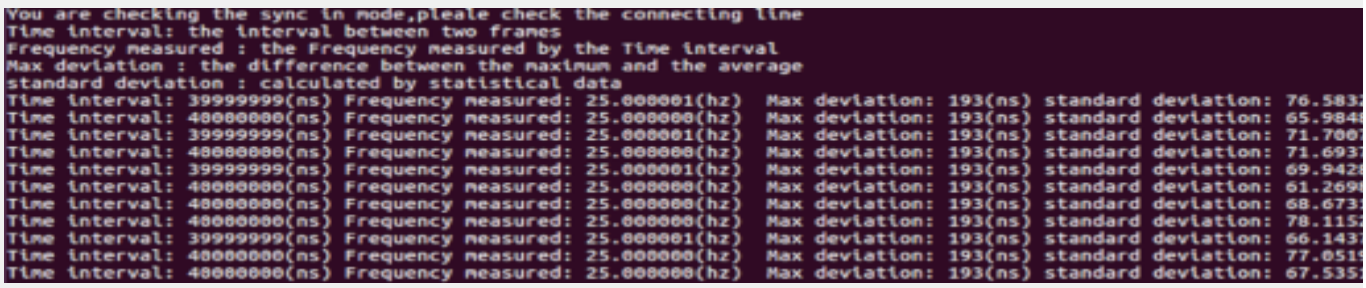


图 PPS测试结果

GMSL2摄像头使用方法

接口特性

- **不支持热插拔**。
- 支持最长15米同轴电缆的信号传输。
- 推荐支持输出分辨率为720p，1080p，4k等多种分辨率的相机。

GMSL2摄像头支持

序号	品牌	产品型号	支持类型	快门类型	分辨率	帧率
1	Entron	S001A	正式	卷帘	1280*720	30fps
2	Sensing	SG1-AR0143-0101-GMSL	正式	卷帘	1280*720	30fps
3	丽景	LC022A1（IMX390-5200-GMSL2）	Beta	卷帘	1920*1080	30fps
4	Sensing	SG1-AR0147-0101-GMSL	Beta	卷帘	1280*720	30fps
5	Sensing	SG2-AR0231-0202-GMSL	Beta	卷帘	1920*1080	22fps
6	Sensing	SG2-AR0233-GW5200-GMSL2	Beta	卷帘	1920*1080	30-60fps
7	Sensing	SG1-AR0144C-8310-GMSL	Beta	全局	1280*720	30fps
8	Sensing	SG1-AR0144M-8310-GMSL	Beta	全局	1280*720	30fps
9	Sensing	SG8-AR0820C-5300-GMSL2	Beta	卷帘	3840*2160	30fps
10	Sensing	SG2-IMX390C-5200-GMSL2	Beta	卷帘	1920*1080	30fps
11	英睿	Xsafe_A6D	Beta	/	640*512	25fps
12	丽景	LC008A1（AR0233-5200-GMSL2）	Beta	卷帘	1920*1080	30fps

注：

1. 正式支持：每次米文系统版本升级，会在米文设备上进行验证。
2. BETA支持：米文调试过，但不会在每次米文系统版本升级中验证，如使用过程中需要进一步支持请联系对应的销售工程师或客户经理。

连线方式

连接方式请参考"接口说明"部分

名词解释

名词	解释	备注
自触发	指摄像头不受外部触发信号控制，自己进行快门。	一般相机都支持这种模式。只要外部不给触发信号就是工作在这种模式。
同步触发	指所有摄像头受同时触发信号控制，在同时进行快门。	**需要相机固件支持。**购买相机的时候请和厂商确认是否支持。 同时需要外部给触发信号。（如使用我们的SDK）

摄像头配置

在初次接入GMSL摄像头以及更换GMSL摄像头型号时，

对于Jetpack 4.4及以前，可以参考：通用-Jetpack 4.4版本及以下镜像MIIVII SETTINGS的使用说明

对于Jetpack 4.5及以后，可以参考：通用-Jetpack 4.5版本及以上镜像MIIVII SETTINGS的使用说明

快速验证

您可以使用Linux下的Cheese工具，进行快速验证出图是否正常。

您可以参照如下视频：

1. 确认GMSL相机设置正确
2. 打开Cheese
3. 确认分辨率设置正常
4. 选中Cheese相机对应的设备名字。（视频中选中的是video4）

请注意，由于Cheese并不能设置图像格式，由于相机输出的格式并不一定和Cheese默认格式匹配，因此可能存在色差。这个属于Cheese软件的问题。

视频输出

1.为了方便使用，设备提供cameras_egl_demo，cameras_opencv_demo，cameras_sdk_demo三个可执行文件来显示GMSL摄像头图像。具体请参考 /opt/miivii/features/gmsl_camera

非SDK用法（**强烈推荐！**）：OpenCV Python Demo，OpenCV C++ Demo

cameras_opencv_demo：OpenCV Demo 直接使用v4l和opencv来获取摄像头图像。

cameras_sdk_demo：SDK Demo，兼容GMSL1的SDK，使用ASIC来进行图像格式的转换，运行效率高。并且可以通过SDK获取触发快门的时间戳。（如需时间戳，则推荐，否则不推荐）

cameras_egl_demo：EGL Demo，使用egl来作为显示部分实现，显示部分效率高。

注：EVO TX2 GMSL2产品无cameras_opencv_demo

2.设备支持使用gstreamer输出视频流，图像获取与显示使用方法如下：

720P：

```
gst-launch-1.0 -v v4l2src device="/dev/video1" ! video/x-raw,framerate=30
/1,width=1280,height=720,format=UYVY ! xvimagesink
```

1080P：

```
gst-launch-1.0 -v v4l2src device="/dev/video0" ! video/x-raw,framerate=60
/1,width=1920,height=1080,format=UYVY ! xvimagesink
```

请注意，由于相机输出的格式并不一定和gstreamer默认格式匹配，因此可能存在色差。这个属于相机固件与gstreamer格式匹配的问题。

3.视频输出分为自触发模式，同步模式两种应用场景：

(1).**自触发模式**：一般所有相机都支持这种模式。是指相机根据相机本身最高帧率输出，此时无法获取相机的时间戳，相机之间的图片也无法同步。

在使用自触发模式的情况下，无需使用任何SDK。可以参考：OpenCV Python Demo，OpenCV C++ Demo

但要注意需要自己进行格式转换。如从YUYV转成BGR格式等。

(2).**同步模式**：需要相机固件支持外部触发。是指所有相机都被同一触发信号触发，快门时间几乎严格同步。

如果你只是希望使用同步模式，但是不关心相机的时间戳，我们仍然建议您使用非SDK的示例。OpenCV Python Demo，OpenCV C++ Demo

在同步模式下，使用OpenCVdemo显示/dev/video0的摄像头画面。**请注意，1280x720需要和摄像头真实分辨率匹配。如1080p的相机，则为1920x1080。**

```
./bin/cameras_opencv_demo -s 1280x720 -d /dev/video0
```

在同步模式下，使用SDK demo显示/dev/video0的摄像头画面。**请注意，1280x720需要和摄像头真实分辨率匹配。如1080p的相机，则为1920x1080。**

```
./bin/cameras_sdk_demo -s 1280x720 -d /dev/video0
```

命令示例：

第一步：编译

```
cp -r /opt/miivii/features ~/
cd ~/features/gmsl_camera
sudo make;
```

运行OpenCV Demo**（请注意分辨率和相机真实分辨率必须一致）**

```
./bin/cameras_opencv_demo -s 1280x720 -d /dev/video0
```

运行SDK Demo**（请注意分辨率和相机真实分辨率必须一致）**

```
./bin/cameras_sdk_demo -s 1280x720 -d /dev/video0
```

运行EGL Demo**（请注意分辨率和相机真实分辨率必须一致）**

```
./bin/cameras_egl_demo -s 1280x720 -d /dev/video0
```

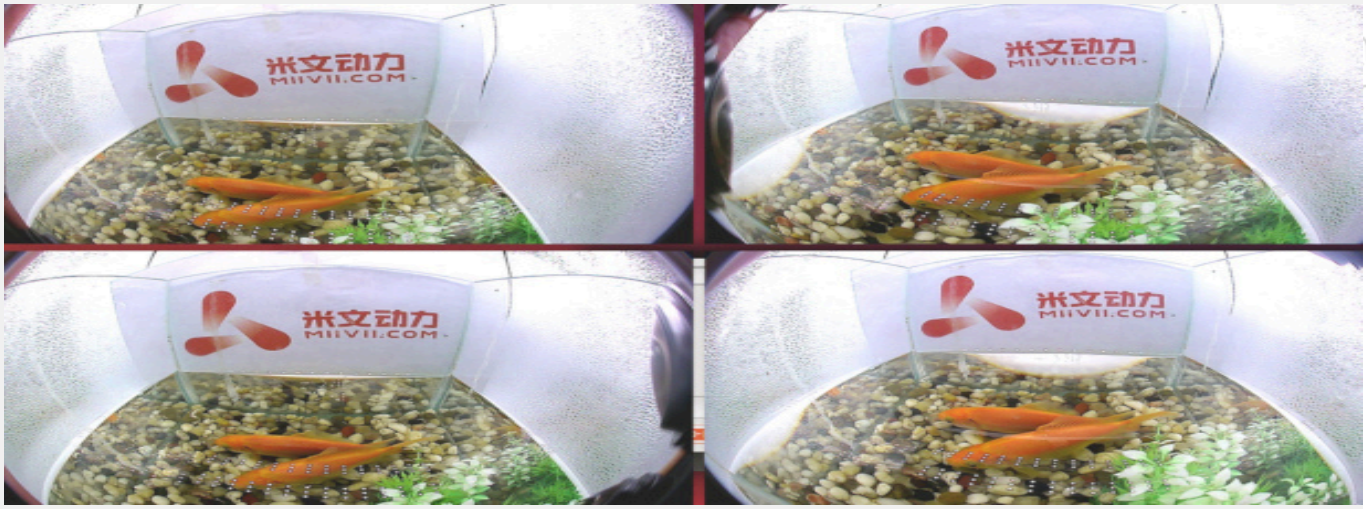


图 摄像头图像

在同一进程，打开多路相机，并且获取时间戳：

命令	结果示例
Apex2 上在同时打开2路相机 `./bin/cameras_sdk_demo -s 1280x720 -d /dev/video0 -m 2` -m表示打开的摄像头节点数量。	会显示2路独立视频，并以独立窗口显示。

GMSL/GMSL2时间戳相关测试方法

如何获取详细日志及日志说明?

命令	结果

`# export CHECK_TIME=1 log log`
`sudo jetson_clocks`
`rm /tmp/cameras_sdk_demo.log`
`./bin/cameras_sdk_demo -s 1280x720 -d /dev/video0`
	1. 在屏幕打印中会增加打印日志
	2. 在/tmp/下，会生成cameras_sdk_demo.log文件

日志文件说明

字段	单位	物理含义	测试方法
Timestamp	纳秒	触发该帧的时间	根据传输延迟，从队列中获得的触发时间
FrameInterval	纳秒	帧间隔 两帧之间的触发时间间隔	与前一Timestamp的差值
FrameTransferDelay	纳秒	帧传输延迟	LinuxGetFrameTime - Timestamp
LinuxGetFrameTime	纳秒	Linux获取帧时的系统时间	收到帧时候的Linux系统时间
LinuxFrameInterval	纳秒	Linux获取帧的Linux时间间隔	与前一LinuxGetFrameTime的差值

如何确认时间戳是否准确?

在代码中，会通过检查FrameInterval的方式，来确认时间戳是否准确。

执行命令:

```
# export CHECK_TIME=1   log log
sudo jetson_clocks
rm /tmp/cameras_sdk_demo.log
./bin/cameras_sdk_demo -s 1280x720 -d /dev/video0
```

如果时间戳正常，则在/tmp/cameras_sdk_demo.log的内容如下，**只有一行**:

```
Timestamp : 1620897955083817280 FrameInterval : 1620897955083817280
FrameTransferDelay : 66992720 LinuxGetFrameTime : 1620897955150810000
LinuxFrameInterval : 1620897955150810000
```

如果时间戳异常，则在/tmp/cameras_sdk_demo.log中会记录多组时间戳，**为多行**:

```
Timestamp : 1620958367246484576 FrameInterval  :  1620958367246484576
FrameTransferDelay : 67111424 LinuxGetFrameTime : 1620958367313596000
LinuxFrameInterval : 1620958367313596000
Timestamp : 1620958739646034432 FrameInterval  :  1620958739646034432
FrameTransferDelay : 67403568 LinuxGetFrameTime : 1620958739713438000
LinuxFrameInterval : 1620958739713438000
Timestamp : 1620958748796023808 FrameInterval  :  1620958748796023808
FrameTransferDelay : 80901192 LinuxGetFrameTime : 1620958748876925000
LinuxFrameInterval : 1620958748876925000
Timestamp : 1620958789795973504 FrameInterval  :  1620958789795973504
FrameTransferDelay : 72186496 LinuxGetFrameTime : 1620958789868160000
LinuxFrameInterval : 1620958789868160000
Timestamp : 1620959793244763712 FrameInterval  :  1620959793244763712
FrameTransferDelay : 73185288 LinuxGetFrameTime : 1620959793317949000
LinuxFrameInterval : 1620959793317949000
Timestamp : 1620959854794691840 FrameInterval  :  1620959854794691840
FrameTransferDelay : 68099160 LinuxGetFrameTime : 1620959854862791000
LinuxFrameInterval : 1620959854862791000
Timestamp : 1620960274844196896 FrameInterval  :  1620960274844196896
FrameTransferDelay : 68391104 LinuxGetFrameTime : 1620960274912588000
LinuxFrameInterval : 1620960274912588000
Timestamp : 1620960283994186240 FrameInterval  :  1620960283994186240
FrameTransferDelay : 71857760 LinuxGetFrameTime : 1620960284066044000
LinuxFrameInterval : 1620960284066044000
Timestamp : 1620960291394178080 FrameInterval  :  1620960291394178080
FrameTransferDelay : 68419920 LinuxGetFrameTime : 1620960291462598000
LinuxFrameInterval : 1620960291462598000
```

如何确认时间戳精度?

命令	确认方式

把屏幕日志存储到文件中
`# export CHECK_TIME=1  log`
`log`
`sudo jetson_clocks`
`./bin/cameras_sdk_demo -s 1280x720 -d /dev/video0 > log`
	1. 把log导入excel
	2. 对FrameInterval取最大最小值
	3. 通过最大最小值的差，可以获取时间戳精度。

如何确认图像帧传输延迟是否稳定?

确认摄像头图像帧传输延迟

先验知识

GMSL类型	图像分辨率	典型相机	传输延迟
GMSL1	720p	SG1-AR0143-0101-GMSL-Hxxx	60ms左右
GMSL1	1080p	SG2-AR0231-0202-GMSL-Hxxx	100ms左右

使用低于传输延迟的帧率，打开摄像头。由于此时传输延迟小于帧间隔，因此时间戳的缓存为1，因此不会因为软件引入其他问题，所测的的就是真实的物理延迟。

请注意不要使用传输延迟附近所对应的帧率，传输延迟的抖动，会造成时间戳不准。

命令	确认方式

Apex Xavier II
`# export CHECK_TIME=1 log`
`sudo jetson_clocks`
`./bin/cameras_sdk_demo -s 1280x720 -d /dev/video0 -r 10-0`
`-r 10-010`
Apex Xavier和EVO TX2 GMSL2
`# export CHECK_TIME=1 log`
`sudo jetson_clocks`
`./bin/cameras_sdk_demo -s 1280x720 -d /dev/video0 -r 10`
`-r 1010`
	通过FrameTransferDelay确认得到实际的传输延迟

应用功能使用

米文设备提供多种样例，方便客户进行开发和快速验证

算法：米文设备提供算法库，目前提供行人，车辆，自行车三分类的检测。详情请参考/opt/miivii/features/algorithm 中的三分类检测算法

加速SDK：米文设备提供基于Yolo v3识别网络的加速SDK。详情请参考/opt/miivii/features/miivii-accelerator

ROS范例：米文设备提供基于ROS的DEMO。详情请参考米文动力Githubhttps://github.com/MiiViiDynamics

除此之外，米文动力还为开发者提供了部分开源代码，请于米文动力Github查看 https://github.com/MiiViiDynamics

附录

异常处理

如在开发过程中出现异常情况，可先通过DEBUG串口打印log自行判断问题。具体操作如下:

第一步：根据【接口说明】部分中的信息，找到DEBUG接口的具体位置

第二步：用一根UART-USB转接线[1]，将DEBUG接口与上位机PC相连接

第三步：在上位机PC端，下载串口调试工具，将波特率调整为115200 Baud

第四步：在串口调试工具中抓取串口log以便分析异常问题

[1]：可根据【接口说明】部分中的信息，选择RS232-USB转接线或者TTL-USB转接线。

系统在线升级（OTA）的使用说明

概述

系统在线升级，通常又是OTA，是米文针对所有设备提供的软件服务。

即可以不进行刷机来更新系统固件。

从Jetpack 4.5开始，所有的米文设备都支持系统在线升级。

使用方式

方法一（推荐）：使用MIIVII SETTINGS进行版本升级和回退；

1. 在设备上打开浏览器输入http://127.0.0.1:3000，或者远程PC浏览器上输入http://<device ip>:3000
2. 使用系统登录账号登录到MIIVII SETTINGS界面；

3. 选择系统升级功能，点击"检查更新"检查是否有新版本；

4. 检测到有升级版本时，可以点击"系统升级"来升级安装包

5. 完成升级后系统会记录升级时间，可以查看该时间的升级记录

6. 然后点击回退即可回退到历史版本。

7. 升级完成后重启系统以确保升级内容生效

方法二：使用命令行进行升级或者升级指定安装包

使用命令行进行升级

1. 执行下面命令更新源

```
sudo apt update
```

2. 执行下面命令升级系统

```
sudo apt upgrade -y
```

3. 升级完成后重启系统以确保升级内容生效

升级指定安装包

- 执行下面命令升级指定安装包

```
sudo apt install -y miivii-websettings
```

Jetpack 4.4版本及以下镜像烧录

请参考：Jetpack 4.4版本及以下镜像烧录

Jetpack 4.5版本及以上镜像烧录

1.功能介绍

米文刷机工具，适用于米文系列产品。

米文刷机工具，是为了方便进行米文设备的烧写、克隆，小批量生产而提供的工具软件。

您可以通过X86架构PC作为烧写主机，给米文设备烧写米文动力官方镜像。在开发米文设备一段时间后，可以将现有设备镜像克隆来保存开发进度，并单台或小批量烧写到其他米文设备中。

核心功能

- 自动检测使用环境
- 自动检测最新镜像
- 内置镜像下载器，无需手动下载镜像
- 支持批量烧写
- 支持镜像克隆（但要注意Clone后再烧写需要使用同一Jetpack版本）

2.准备软件硬件

2.1. 烧写主机准备

需要将烧写主机与米文设备连接方能烧写镜像。烧写主机推荐配置如下：

- CPU采用X86架构的Intel酷睿系列处理器
- 内存8GB ddr3及以上
- 空余硬盘容量40G 及以上
- 系统为Ubuntu Linux x64 v16.04或v18.04

2.2. 烧写软件环境准备

- sudo apt install python2.7python3python

2.3. 准备米文烧写工具和米文设备镜像

2.3.1.刷机工具安装

- 准备PC主机，系统为：Ubuntu 16.04或者Ubuntu 18.04
- 安装key

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
05BE38FE8ADA7CD12E3281B52FC7A8453C3B8F24
```

- 在本地 ubuntu 系统中添加源

```
sudo sh -c 'echo "deb http://upgrade.miivii.com/miiviitools/
mvtools main" > /etc/apt/sources.list.d/miivii-l4t-apt-source.list'
```

- 手动更新

```
sudo apt update
```

- apt-get 安装 刷机工具 Deb 包

```
sudo apt-get install miivii-ftool
```

- 安装完成后(在18.04系统中点击Show Applications或者在16.04系统中点击Search Your Computer)会发现如下快捷方式

- 双击快捷方式，输入密码：您的sudo密码。

2.4. 准备硬件

● 米文设备及电源, USB 数据线

3.操作

3.1. 硬件连接

● 通过 USB 数据线将米文设备烧写口与烧写主机相连;

● 按住米文设备的RECOVERY按钮，之后给米文设备上电开机，进入FORCE_Recovery烧写模式。

3.2软件使用

3.2.1. 镜像烧写

3.2.1.1在线模式镜像烧写

● 点击"在线模式"复选框，选择Jetpack版本及下载路径，并点击下一步，开始下载选择版本当前最新的刷机环境及设备镜像

●这里需要选择下载完成后是否自动开始刷机，选择自动后，下载完成后会自动解压、校验、刷机

●下载速度取决于所在环境的网速，一般可达5M/s

●开始刷机后通常需要15分钟以上完成，请耐心等待

3.2.1.2离线模式镜像烧写

● 点击"离线模式"复选框，选择已经下载好的刷机环境及设备镜像，并点击下一步直接开始烧录。

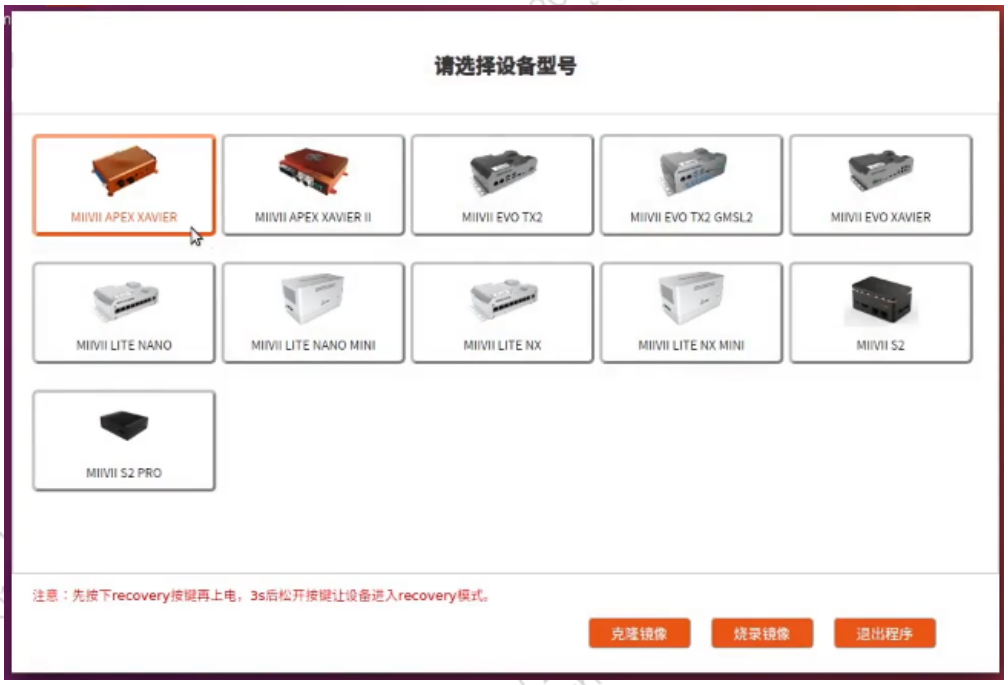
刷机工具使用教程.mp4

3.2.2. 镜像克隆

● 将打算克隆的米文设备按照3.1的方法进入FORCE_Recovery模式，打开烧写工具

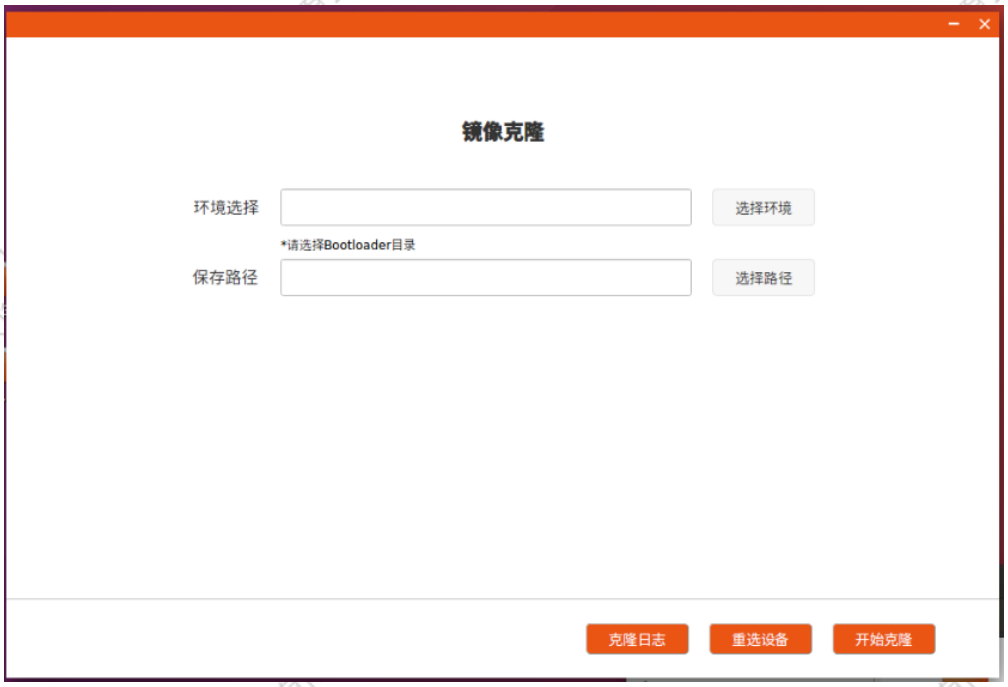
● 点击【输入上位机密码】按钮，**输入当前烧写主机的开机密码**

● 点击【克隆镜像】按钮，进入克隆操作



●修改克隆文件保存在烧写主机中的路径和名称*，并点击开始克隆

注：文件存储路径中不能有中文或特殊字符



- 镜像克隆通常需要30分钟以上才能完成：
- 克隆完成，会生成克隆镜像与MD5文件，再次烧写请按照3.2.1步骤进行操作

注：如在镜像烧写，克隆过程中遇到问题，请联系米文动力售后邮箱需求帮助：helpdesk@miivii.com

附1. 烧写问题自检

如果遇到烧写问题，请先按照如下条目进行自检：

- 检查是否在烧写工具左上角输入了上位机开机密码
- 检查是否进入到Recovery模式，可以通过lsusb命令鉴定
- 检查Micro USB、双Type A线缆质量是否达标，是否只是用于充电的双芯线
- 检查上位机，是否为X86-64架构台式机，笔记本。（服务器，嵌入式设备，虚拟机等其他设备暂不支持）
- 检查上位机系统是否为 Linux 1604 1804
- 检查磁盘格式，烧写主机的磁盘格式推荐为EXT4
- 检查上位机容量是否足够
- 镜像和烧写工具存储路径中不能有中文或其他特殊字符