

IoT.DT 万物纵横

www.iotdt.com

DA160S

产品使用手册 V1.2

四川万物纵横科技股份有限公司

版本修订记录

版本	修改人	文档说明	日期
V1.0	廖		2025-2-13
V1.1	代	更新各接口使用说明	2025-7-1
V1.2	廖	更新 4G/5G 与 GPS 说明	2026-6-10

目 录

第一章 配件清单.....	1
1.1. 配件清单.....	1
1.2. 操作系统.....	1
1.3. 外设支持.....	1
第二章 配套系统服务和接口测试说明.....	2
2.1. 调试工具使用.....	2
2.2. 千兆网口使用说明.....	3
2.3. Ssh 远程使用说明.....	4
2.4. Sftp 服务实用说明.....	5
2.5. LED 功能说明.....	6
2.6. WDT(看门狗).....	6
2.7. USB 使用说明.....	7
2.8. 继电器使用说明.....	7
2.9. TF 使用说明.....	8
2.10. RTC 使用说明.....	9
2.11. 音频使用说明.....	10
2.12. HDMI-OUT 使用说明.....	11
2.12.1. HDMI 视频使用.....	11
2.13. RS-232/RS-485/IO 使用说明.....	11
2.13.1. 串口使用说明.....	11
2.13.2. GPIO 使用说明.....	14
2.14. CAN 接口使用说明.....	15

2.15. SSD 读写测试(选配).....	17
2.16. SATA 读写测试.....	19
2.17. 4G/5G 模块测试.....	20
2.18. GNSS 使用说明	24
2.19. docker 使用说明.....	25
第三章 系统升级.....	26
3.1. TF 卡方式烧写	26
3.2. 内存空间分配.....	28

第一章 配件清单

1.1. 配件清单

默认标配如下：

配件名称	数量	说明
DA160S盒子	1个	
电源适配器	1个	输入：100V-240VAC, 50Hz/60Hz, 0.4A Max 输出：12VDC, 3A
1.5m国标AC线	1个	标准电压：250VAC，电流：10A，单相电 源（无需三相电源）
接线端子	2个	插拔式
螺丝钉	4个	

1.2. 操作系统

DA160S 预装 Ubuntu 22.04，用户可以通过 ssh 登录或者串口登录。

1.3. 外设支持

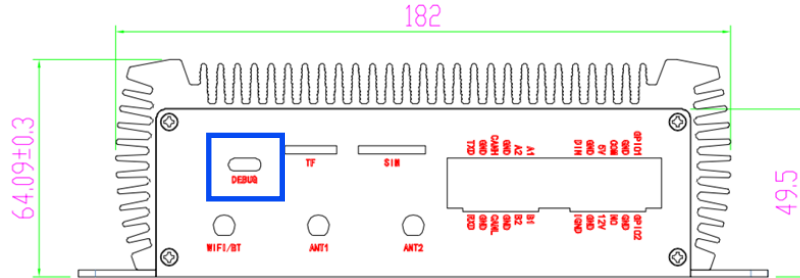
DA160S 可以连接鼠标、键盘等设备进行使用。

1.4. 注意：如果需要刷机，请使用万物纵横对应版本刷机包， 否则会导致升级后启动异常或功能异常！

第二章 配套系统服务和接口测试说明

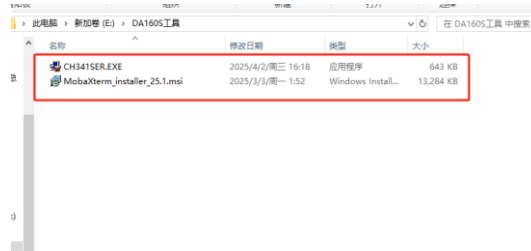
2.1. 调试工具使用

DA160S 盒子调试需要 USB Type-C 线搭配串口工具使用。

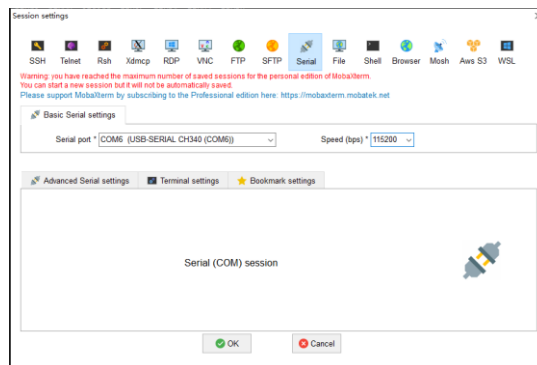


➤ 操作步骤

- 1)硬件先使用接上盒子的 DEBUG 口连接到电脑 PC。
- 2)安装调试工具包下的 CH340SER 串口驱动；



- 3)查看工具的 Log;例：MobaXterm 使用方法步骤如下图所示：(选择对应 debug 串口，波特率:115200)用户名：linaro 密码：linaro



```

Ubuntu 22.04.5 LTS sophon tty50
sophon login: linaro
Password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.10.4-sophon-custom aarch64)

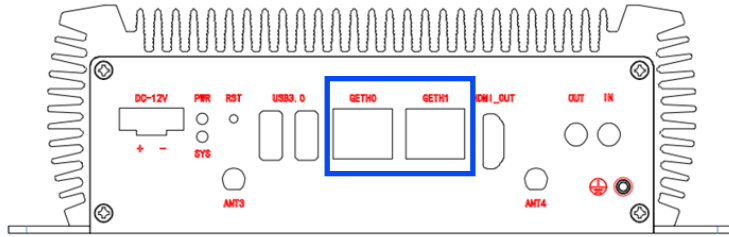
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro
overlay / overlay rw,relatime,lowerdir=/media/root-ro,upperdir=/media/root-rw,workdir=/media/root-rw/overlay-workdir 0 0
/dev/mmcblk0p5 /media/root-rw ext4 rw,relatime 0 0
/dev/mmcblk0p4 /media/root-ro ext4 ro,relatime 0 0

Last login: Tue Jun 10 19:02:32 CST 2025 on tty50
[ OK ] Started User Manager for UID 1000.
[ OK ] Started Session c1 of User linaro.
[ OK ] Finished setup sophon runtime env..
[ OK ] Started Sophon HDMI Service.
Stopping Getty on tty1..
[ OK ] Stopped Getty on tty1.
linaro@sophon:~$ [ OK ] Started Docker Application Container Engine.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
Starting Record Runlevel Change in UTMP...
u[ OK ] Finished Record Runlevel Change in UTMP.
do -s
[sudo] password for linaro:
root@sophon:/home/linaro# bm_version
SophonSDK(CV186AH) 1.9
sophon-soc-libsophon : 0.4.11
sophon-soc-libsophon-dev : 0.4.11
sophon-media-soc-sophon-ffmpeg : 1.9.0
sophon-media-soc-sophon-opencv : 1.9.0
BL2 cv186ah:pd37de75 2025-05-29T09:35:30+08:00
BL31 cv186ah:pd37de75 2025-05-29T09:35:30+08:00
U-Boot 2021.10-dirty (May 29 2025 - 09:35:08 +0800) cvitek_sophgo
KernelVersion : Linux sophon 5.10.4-sophon-custom #7 SMP Thu May 29 11:48:10 CST 2025 aarch64 aarch64 aarch64 GNU/Linux
Mode: soc mode
NCUVersion: 16.16.1
root@sophon:/home/linaro#

```

登录后，确认版本号：bm_version

2.2. 千兆网口使用说明



◆ 说明

- 1、有两个网卡，其中 GETH0 对应 eth0，GETH1 对应 eth1。
- 2、修改网络 IP 配置，可以在/etc/netplan 目录下 01-netcfg.yaml 配置文件。
- 3、eth0 默认 DHCP 模式，eth1 默认静态 IP：192.168.1.254。

➤ 操作步骤

- 步骤 1 请将网线接上 eth0 口和交换机，eth1 口和电脑(或交换机)，然后，电脑加入 192.168.1.xxx 网段；
- 步骤 2 vi /etc/netplan/01-netcfg.yaml
- 步骤 3 修改 IP 配置文件，默认配置和参考配置：

```
root@sophon:/etc/netplan# cat 01-netcfg.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      dhcp4: yes
      addresses: []
      optional: yes
      dhcp-identifier: mac
    eth1:
      dhcp4: no
      addresses: [192.168.158.1/24]
      optional: yes
```

- **步骤 4** 修改完，netplan apply 或重启设备生效网络配置，
- **步骤 5** 用 ifconfig 查看，确认 IP。

2.3. Ssh 远程使用说明

◆ 说明

DA160S 支持通过 ssh 协议(openssh-server)，通过 IP 地址 ssh 远程登录控制台；

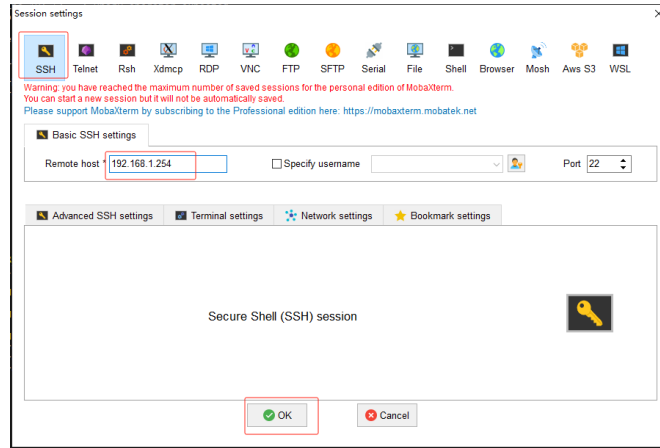
➤ 操作步骤

- **步骤 1** PC 和 DA160S 接入同一网段，192.168.1.1/24，DA160S 使用 eth1
- **步骤 2** 如：PC IP 配置为 192.168.1.2，保证能 ping 通 DA160S（192.168.1.254）
- **步骤 1** 使用软件工具登录设置的 IP 地址：**192.168.1.254**。

端口号：22

账号：linaro

密码：linaro



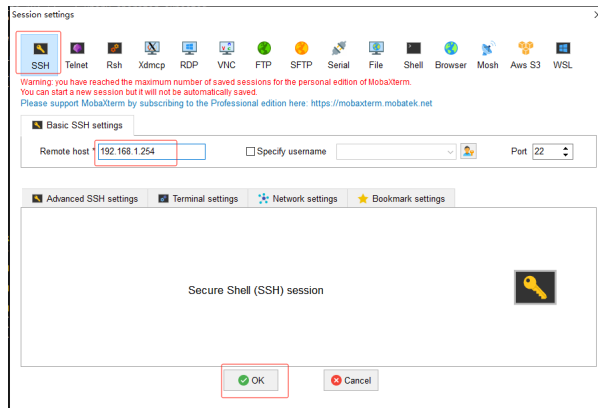
2.4. Sftp 服务实用说明

◆ 说明

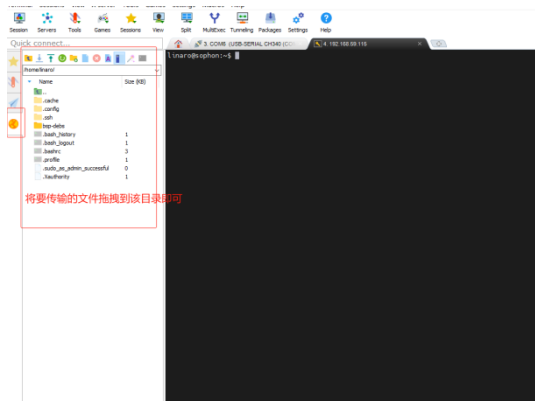
使用 ssh 的 scp 协议方式，允许客户通过远程传输文件；

➤ 操作步骤

- 步骤 1 使用 linaro 用户，远程登录 ssh。



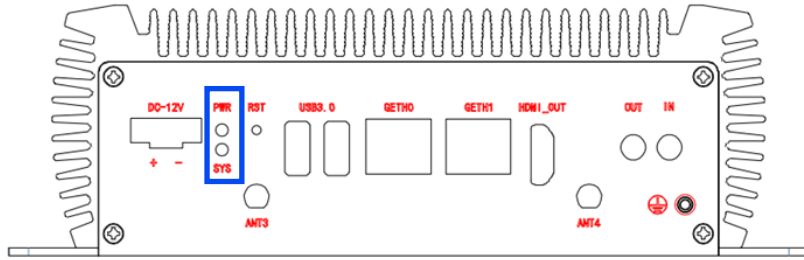
- 步骤 2 传输和下载文件到指定文件夹，将文件拖拽到左侧目录即可上传



•步骤 3 传输完成。

2.5. LED 功能说明

◆ 说明



DA160S 有 2 个 LED，功能如下：

LED_PWR	电源灯	亮：通电 灭：断电
LED_SYS	系统状态灯	

2.6. WDT (看门狗)

DA160S 有一个硬件看门狗，在系统中看门狗默认是关闭状态。

如果需要启用，按照如下流程进行测试：

喂狗

```
# echo a > /dev/watchdog
```

超时时间默认为 30S，如果 30S 内再次执行此操作，则重置看门狗超时时间。

在 30S 内不再喂狗，则看门狗会触发系统重启。

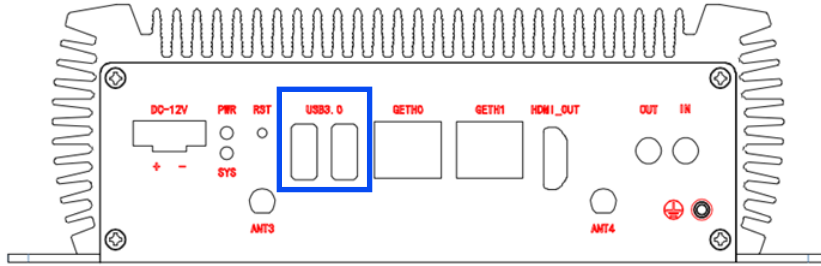
停止喂狗：

```
# echo V > /dev/watchdog
```

2.7. USB 使用说明

◆ 说明

上下两个 USB3.0 接口均为标准的 USB 接口；可接入 U 盘、USB 鼠标、USB 键盘等设备，以 U 盘为例，进行测试：



➤ 操作步骤

- 步骤 1 接入 USB 设备，然后上电，使用 `fdisk -l` 系统会有如下打印：

注意：新使用的 U 盘、TF、mSATA 卡等存储设备需要先使用 `fdisk` 新增分区并进行 `mkfs.ext4`、`mkfs.vfat` 等工具进行格式化，才能进行挂载使用。

- 步骤 2 挂载方式：`mount /dev/sda1/mnt`
- 步骤 3 写测试：`dd if=/dev/urandom of=/mnt/test2 bs=4096 count=1048574`
- 步骤 4 读测试：`dd if=/mnt/test2 of=/dev/null bs=4096 count=1048574`

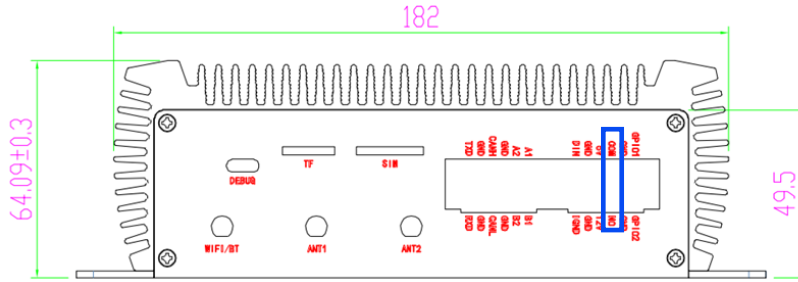
```
root@sophon:/# dd if=/dev/urandom of=/mnt/test2 bs=4096 count=1048574
1048574+0 records in
1048574+0 records out
4294959104 bytes (4.3 GB, 4.0 GiB) copied, 58.9828 s, 72.8 MB/s
root@sophon:/#
root@sophon:/#
root@sophon:/# dd if=/mnt/test2 of=/dev/null bs=4096 count=1048574
1048574+0 records in
1048574+0 records out
4294959104 bytes (4.3 GB, 4.0 GiB) copied, 30.9957 s, 139 MB/s
```

- 步骤 5 若是使用 USB 扩展串口、声卡之类得转接设备，需提前做好驱动适配。

2.8. 继电器使用说明

◆ 说明

电磁继电器开关用于控制继电器，盒子硬件已经预装 1 组；



➤ 操作步骤

• 步骤 1

```
echo 391 > /sys/class/gpio/export
```

```
echo out > /sys/class/gpio/gpio391/direction
```

闭合继电器

```
echo 1 > /sys/class/gpio/gpio391/value
```

断开继电器

```
echo 0 > /sys/class/gpio/gpio391/value
```

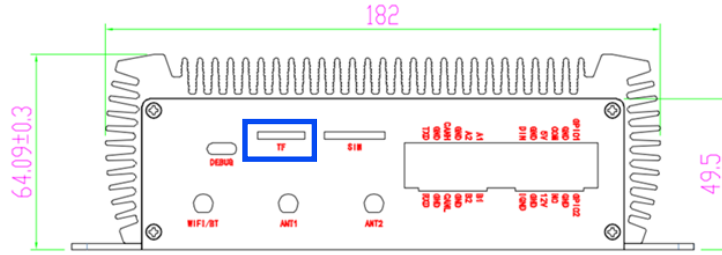
测试现象：继电器的测试，value 值由高低变化时，继电器会有滴答切换声。

2.9. TF 使用说明

◆ 说明

DA160S 盒子前盖板有 1 个 TF 卡接口，直接接上 class10 及以上的 TF 卡，需要重新上电使用，用于升级功能。

(注意：新使用的 U 盘、TF 卡等存储设备需要先使用 fdisk 新增分区，才能进行挂载使用，fdisk 的使用方法请自行在网上寻找。)



➤ 操作步骤

- 步骤 1 格式化 1 个分区：`sudo mkfs.vfat -l /dev/mmcblk1`
- 步骤 2 挂载 TF 卡：`mount /dev/mmcblk1 /mnt`
- 步骤 3 写测试：`dd if=/dev/urandom of=/mnt/test2 bs=4096 count=1048574`
- 步骤 4 读测试：`dd if=/mnt/test2 of=/dev/null bs=4096 count=1048574`

```

root@sophon:~# dd if=/dev/urandom of=/mnt/test1 bs=4096 count=1048576
1048576+0 records in
1048576+0 records out
4294967296 bytes (4.3 GB, 4.0 GiB) copied, 217.367 s, 19.8 MB/s
root@sophon:~#
root@sophon:~#
root@sophon:~# dd if=/mnt/test1 of=/dev/null bs=4096 count=1048574
1048574+0 records in
1048574+0 records out
4294959104 bytes (4.3 GB, 4.0 GiB) copied, 55.589 s, 77.3 MB/s
root@sophon:~#

```

2.10. RTC 使用说明

◆ 说明

DA160S 出厂时，会搭配着纽扣电池，给 RTC 时钟供电；支持 RTC 硬件时间调整。

➤ 操作步骤

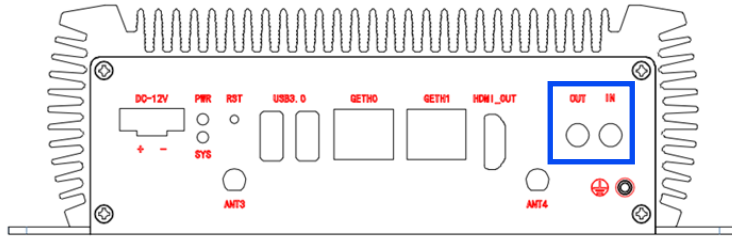
设置 RTC 时间：

- 步骤 1 设置系统时间，`date -s "2025-07-01 13:11:00"`
- 步骤 2 时间保存，写入 RTC 硬件，`hwclock -w`
- 步骤 3 查询 RTC 时间是否设置成功，`hwclock -r`

2.11. 音频使用说明

◆ 说明

DA160S 支持接入 MIC 输入，喇叭输出。通过程序使用喇叭播放 MIC 输入的音频信号。



➤ 操作步骤

• **步骤 1** 盒子上预装了 tinyalsa，但为了方便测试，使用网络在线安装 alsa 库
`apt install alsa-base`

• **步骤 1** `aplay -l`，查询 USBaudio 的设备号，如下图：

`arecord -l`，查询 USB audio 的设备号，如下图：

```

root@ophon:/home/linaro# alsa
alsa      alsa-info  alsabat  alsabat-test  alsactl  alsaloop  alsamixer  alsatplg  alsaucm
root@ophon:/home/linaro# alsa-C
root@ophon:/home/linaro#
root@ophon:/home/linaro# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 1: cv186xadc [cv186x_adc], device 0: cviteka-dac 2810a000.dac-0
Subdevices: 1/1
Subdevice #0: subdevice #0
card 1: cv186xadc [cv186x_adc], device 1: cviteka-dac1 28109000.dac-1 [cviteka-dac1 28109000.dac-1]
Subdevices: 1/1
Subdevice #0: subdevice #0
card 2: DWHDMI [DW-HDMI], device 0: DW HDMI [dw-hdmi-ahb-audio]
Subdevices: 1/1
Subdevice #0: subdevice #0
root@ophon:/home/linaro# arecord -l
**** List of CAPTURE Hardware Devices ****
card 0: cv186xadc [cv186x_adc], device 0: cviteka-adc 2810a100.adc-0 [cviteka-adc 2810a100.adc-0]
Subdevices: 1/1
Subdevice #0: subdevice #0
card 0: cv186xadc [cv186x_adc], device 1: cviteka-adc1 28109100.adc-1 [cviteka-adc1 28109100.adc-1]
Subdevices: 1/1
Subdevice #0: subdevice #0
root@ophon:/home/linaro#

```

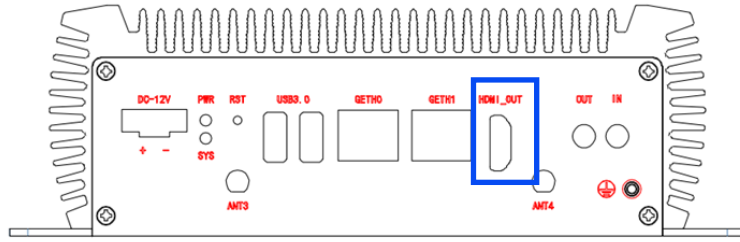
• **步骤 2** 根据查询到的设备号，修改 `arecord` 和 `aplay` 的操作对象

• **步骤 3** Audio 音频设备，录音：`arecord -D hw:0,1 -f dat wavtest.wav -d 10`

• **步骤 4** Audio 音频设备，播音：`aplay -D hw:1,0 -f dat wavetest.wav`

• **步骤 5** Audio 音频设备，回环：`arecord -D hw:0,1 -f dat | aplay -D hw:1,0`

2.12. HDMI-OUT 使用说明



2.12.1. HDMI 视频使用

◆ 说明

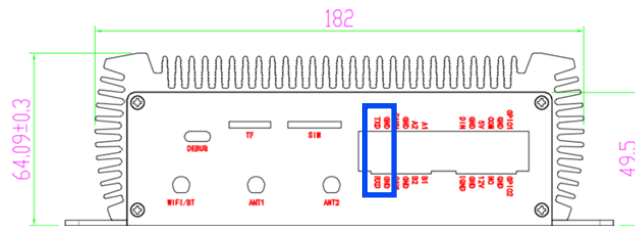
DA160S 的 HDMI 接口，支持使用 HDMI-OUT 播放视频信号，测试过的分辨率最大为 4K@60fps。

➤ 操作步骤(播放软件默认的 QT 界面信息)

- 步骤 1 将 HDMI 口，默认接到 4K@60fps 的显示器上。进入系统
- 步骤 2 查看 HDMI 输出。
- 步骤 3 显示器的分辨率正确，会输出系统信息和网络信息。

2.13. RS-232/RS-485/I0 使用说明

2.13.1. 串口使用说明



名称	数量	说明
RS-232	1路	接入RXD、TXD、GND，对应控制/dev/ttyS1
RS-485	2路	RS485接入A、B、GND，RS485_1：对应控制/dev/ttyS2；

		RS485_2: 对应控制/dev/ttyS3
IO	3个	2路输入/输出: 对应GPIO421、GPIO422 1路输入DIN: DIN: GPIO389
继电器	1路	接入COM、NO, 对应控制GPIO391, 接入设备<=36V

➤ 操作步骤(串口 232)

- **步骤 1** 准备杜邦线，RS-232USB 转串口线一根，开始收发测试。
- **步骤 2** 将盒子的 TX、RX、GND 分别接到 RS-232 串口线的 RX、TX、GND，串口线接到 PC；
- **步骤 3** 打开串口工具(sscom)，锁上 COM 口，波特率 115200；
- **步骤 4** PC 端打开串口助手，设置参数 115200-8N-1



- **步骤 5 板端发送数据测试**

板端执行如下命令发送数据，需确定板端 RS232 串口，假设为/dev/ttyS2:

设置串口参数为: 115200-8N-1

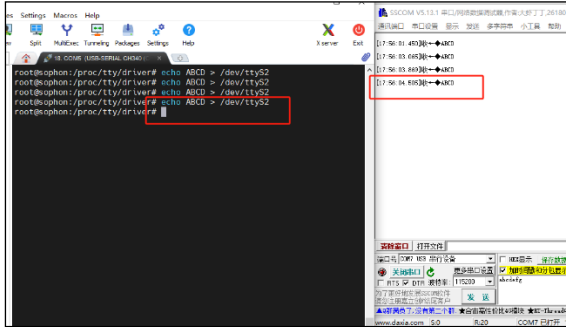
```
# stty -F /dev/ttyS2 raw
```

```
# stty -F /dev/ttyS2 -echo -iexten
```

```
# stty -F /dev/ttyS2 raw 115200 cs8 -parenb -cstopb
```

发送数据

```
# echo ABCD > /dev/ttyS2
```



● 板端接收数据测试

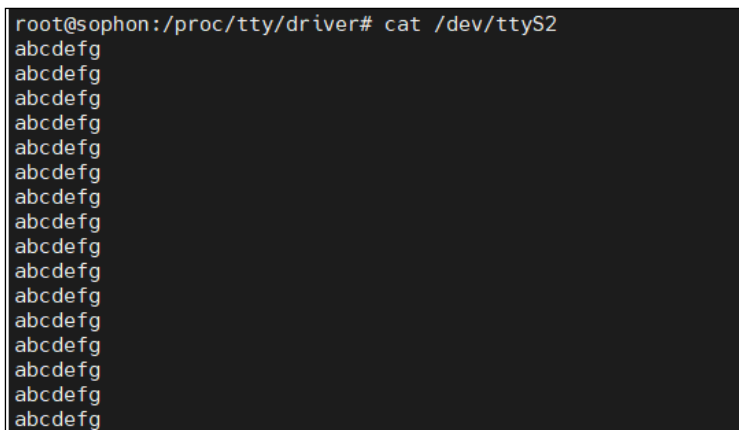
板端执行如下命令等待接收数据：

```
# cat /dev/ttyS1
```

PC 在串口工具中发送数据，如发送 abcdefg，需要发送回车换行：

预期结果：

发送后，板端可以收到 PC 端发送的数据 abcdefg



● 步骤 7 确认盒子收发数据与预期结果一致。

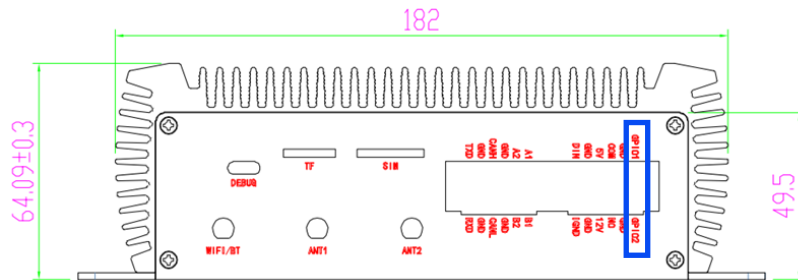
➤ 操作步骤(串口 485)

- **步骤 1** 准备杜邦线，RS-485、USB 转串口线一根，开始收发测试；
- **步骤 2** 将盒子的 A、B、GND 分别接到 RS-485 串口线的 A、B、GND，串口线接到 PC；
- **步骤 3** 打开串口工具(sscom)，锁上 COM 口，波特率 115200；
- **步骤 4** 串口收发测试和 RS232 流程一致，需要修改的是发送和接收对应的/dev/ttySx， 这里 ttySx 根据设备决定，RS485_1 对应 ttyS2、RS485_2 对应 ttyS3。

2.13.2. GPIO 使用说明

◆ 说明

本小节主要讲述外围设备 GPIO 的使用方法：DA160S 提供 4 组可控的外围 GPIO；



GPIO1: gpio421

GPIO2: gpio422

➤ 操作步骤(输入 GPIO421)

• 步骤

```
echo 421 > /sys/class/gpio/export
```

```
echo in > /sys/class/gpio/gpio421/direction
```

```
cat /sys/class/gpio/gpio421/value
```

1)输出模式的 GPIO，可以通过供地方式，将外接设备的高低电平信号接入

IN 口。

2)查看 value 值的变化。

➤ 操作步骤(输出 GPIO422)

连接方法：OUT1 用杜邦线接到 IN1。

•步骤

```
echo 422 >/sys/class/gpio/export
```

```
echo out > /sys/class/gpio/gpio422/direction
```

```
echo 1 > /sys/class/gpio/gpio422/value
```

```
echo 0 > /sys/class/gpio/gpio422/value
```

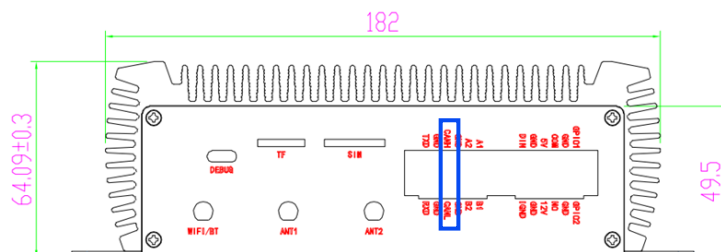
1)当 gpio422/value 设置为 1 时；cat /sys/class/gpio/gpio422/value，也为 1，此时外部输出为高电平。

2)当 gpio480/value 设置为 0 时；cat /sys/class/gpio/gpio482/value，也为 0，此时外部输出为低电平。

2.14. CAN 接口使用说明

◆ 说明

DA160S 支持 CAN 总线协议，允许 PC 等上位机通过 CAN 与盒子连接通信。



➤ 操作步骤

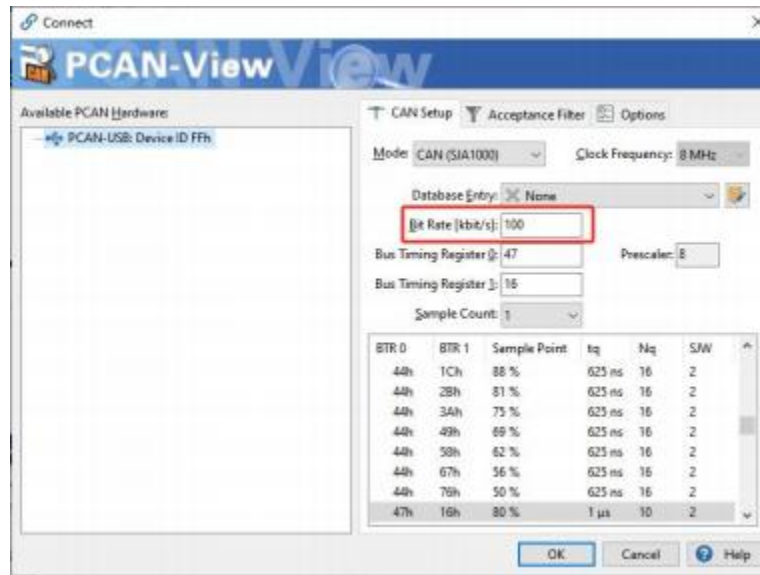
•步骤 1 windows 安装 PEAK-System 驱动包，然后安装 pcanview 工

具包。

•步骤 2 将 USBCAN 适配器的 USB 端接入电脑 USB 口;

•步骤 3 将 USBCAN 适配器的 CAN 端 L 接入盒子的 CANL 口, H 接入盒子的 CANH 口

•步骤 4 将打开 pcانview 工具, 按如下配置:



•步骤 5 将在 debug 口输入串口命令, 如下:

1、查询当前网络设备: `ifconfig can0`, 查看 can0 设备

2、设置比特率 100KHz: `ip link set can0 up type can bitrate 100000`

3、启动 CAN: `ip link set can0 up`

4、发送数据:

```
cansend can0 -i 0x123 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
```

```
cansend can0 -i 0x234 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
```

5、接收数据: `can dump can0`

•步骤 6 将在 debug 口输入串口命令, 如下:


```

root@sophon:/home/linaro# fdisk /dev/nvme0n1
Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xe2342089.

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): 默认
First sector (2048-500118191, default 2048): 默认
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-500118191, default 500118191): +100G

Created a new partition 1 of type 'Linux' and of size 100 GiB.

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

```

最后输入 w 保存配置

```

Created a new partition 1 of type 'Linux' and of size 100 GiB.
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

```

成功执行后，可以查看到分区文件：

```

root@sophon:/home/linaro# ls /dev/nvme0*
/dev/nvme0 /dev/nvme0n1 /dev/nvme0n1p1
root@sophon:/home/linaro#

```

格式化为 fat32

```
# mkfs.vfat /dev/nvme0n1p1
```

创建挂载目录

```
# mkdir /mnt/share
```

挂载设备

```
# mount /dev/nvme0n1p1 /mnt/share
```

```

root@sophon:/home/linaro# mkfs.vfat /dev/nvme0n1p1 格式化 fat32
root@sophon:/home/linaro# mkfs.vfat /dev/nvme0n1p1
root@sophon:/home/linaro# mount /dev/nvme0n1p1 /mnt/share/ 挂载SSD
root@sophon:/home/linaro#

```

确认设备挂载情况(容量 100G，挂载位置/mnt/share)

```
root@sophon:/home/linaro# lsblk
NAME        MAJ:MIN RM  SIZE RO  TYPE MOUNTPOINT
mmcblk0     179:0    0  29.1G  0  disk
├─mmcblk0p1 179:1    0   128M  0  part /boot
├─mmcblk0p2 179:2    0   138M  0  part /recovery
├─mmcblk0p3 179:3    0    10M   0  part
├─mmcblk0p4 179:4    0    3.4G  0  part /media/root-ro
├─mmcblk0p5 179:5    0    8.9G  0  part /media/root-rw
├─mmcblk0p6 179:6    0   16.6G  0  part /data
mmcblk0boot0 179:8    0     4M   1  disk
mmcblk0boot1 179:16   0     4M   1  disk
nvme0n1     259:0    0  238.5G  0  disk
├─nvme0n1p1 259:1    0   100G  0  part /mnt/share  查看设备挂载情况
root@sophon:/home/linaro#
```

● SSD 读写流程

写入数据：

```
# echo ABCD > /mnt/share/a.txt
```

```
# sync
```

读取写入的数据：

```
# cat /mnt/share/a.txt
```

```
root@sophon:/home/linaro# echo ABCD > /mnt/share/a.txt
root@sophon:/home/linaro#
root@sophon:/home/linaro# sync
root@sophon:/home/linaro#
root@sophon:/home/linaro# cat /mnt/share/a.txt
ABCD
root@sophon:/home/linaro#
```

写一行ABCD到硬盘

从硬盘读取文件

● 测试完成后卸载设备

```
# umount /mnt/share
```

2.16. SATA 读写测试

准备一个 SATA 接口的硬盘用于测试

假设 SATA 硬盘为/dev/sda1

挂载 SATA 硬盘设备

```
# umount /mnt/share
```

读写测试参考 `ssd` 读写流程，将访问的设备名 `/dev/nvme0n1p1` 改成 `/dev/sda1`

读写完后卸载设备

```
# umount /mnt/share
```

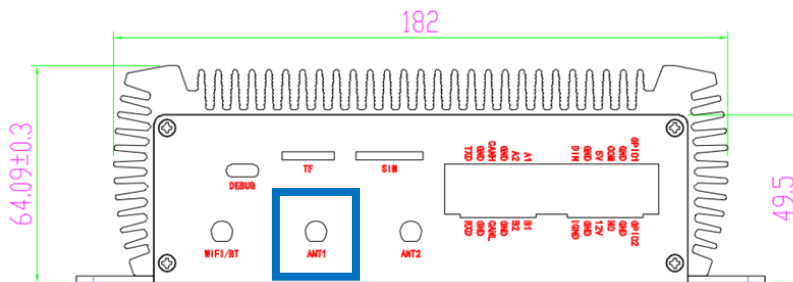
2.17. 4G/5G 模块测试

4G 模块名：EC200A-CN （USB 接口）

5G 模块名：RM500U-CNV （USB 接口）

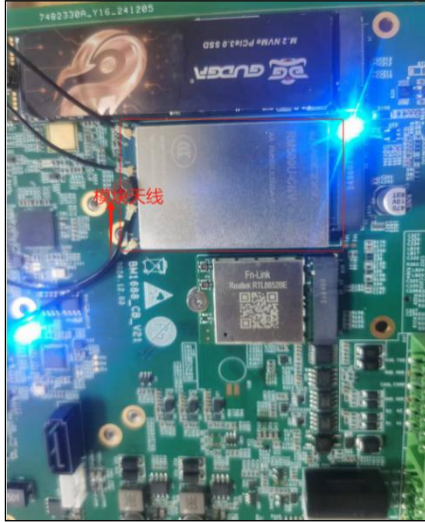
注意 SIM 不支持热插拔，需要断电后，进行 SIM 插拔。

下图为 4G/5G 天线接口

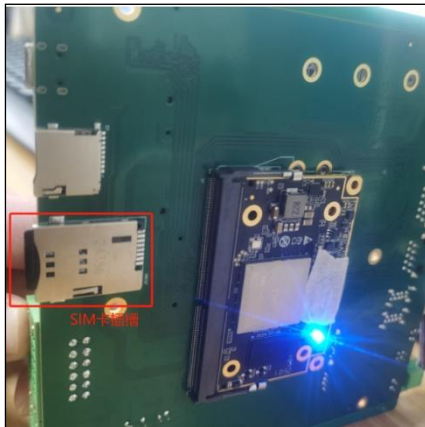


- 插入 5G 模块并接上天线（并插入上网卡(sim 卡或物联网卡)）

5G 模块含天线截图：



插入上网卡



- 在板端默认终端（终端 A），查看设备识别情况

lsub

```

root@sophon:/home/linaro# lsub
Bus 004 Device 005: ID 174c:1153 ASMedia Technology, Inc. ASM1153 SATA 3Gb/s bridge
Bus 004 Device 004: ID 2c7c:0900 Quectel Wireless Solutions Co., Ltd. USB3.0 Hub
Bus 004 Device 003: ID 05e3:0620 Genesys Logic, Inc. USB3.0 Hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 002 Device 005: ID 0bda:b85b Realtek Semiconductor Corp.
Bus 002 Device 003: ID 05e3:0610 Genesys Logic, Inc. 4-port hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@sophon:/home/linaro#
    
```

- 查询对应的 USB 网卡

注意，在 Ubuntu20.04LTS 网卡名以 enx 开头，在 Ubuntu22.04LTS 网卡名以 usb 命名。下面我们以 Ubuntu20.04 为例进行介绍：

```
# ifconfig -a
```

```
root@sqphon:/home/linaro# ifconfig -a
can0: flags=128<NOARP> mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 28

can1: flags=128<NOARP> mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 29

docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:57:dd:96:49 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

dummy0: flags=130<BROADCAST,NOARP> mtu 1500
    ether 06:ac:41:93:a7:0f txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enx9dcd2a0478: flags=4090<BROADCAST,MULTICAST> mtu 1500
    ether be:9d:cd:2a:04:78 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- 使能网卡

```
# ifconfig enx9dcd2a0478 up
```

- 启动一个 ssh 终端登录板端（终端 B），以 root 用户启动 minicom

```
# sudo -s
```

```
# minicom -D /dev/ttyUSB2 -s 115200
```

- 在终端 B 上，配置模块参数

查询卡是否插入

```
# at+cpin?
```

查询信号

```
# at+qeng="servingcell"
```

空闲情况下，信号搜索成功并驻网成功，输出如下信息。

```
+QNETDEVSTATUS: 1,1,"IPV4V6",0
at+qeng="servingcell"
+QENG: "servingcell", "NOCONN", "LTE", "FDD", 460,00,BA6C999,372,1300,3,5,5,8309,-83,-6,-62,15,0,0,42
```

此时，使用 at+CSQ 可以查到正常的信号质量

```
#at+CSQ
```

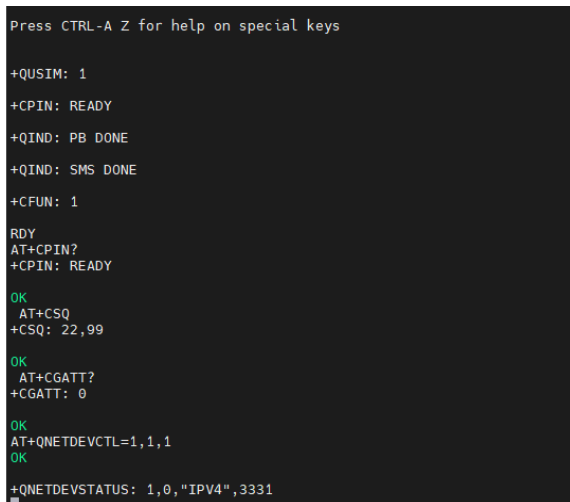
返回: +CSQ: 22,99

第一个数字 22, 代表信号质量 22, 如果第一个数字为 99, 则认为无信号。使用 `at+qeng="servicingell"` 查询网络状态。

拨号

```
#at+QNETDEVCTL=1,1,1
```

正常注册截图示例:



```
Press CTRL-A Z for help on special keys
+QUSIM: 1
+CPIN: READY
+QIND: PB DONE
+QIND: SMS DONE
+CFUN: 1
RDY
AT+CPIN?
+CPIN: READY
OK
AT+CSQ
+CSQ: 22,99
OK
AT+CGATT?
+CGATT: 0
OK
AT+QNETDEVCTL=1,1,1
OK
+QNETDEVSTATUS: 1,0,"IPV4",3331
```

- 在终端 A 上, 给网卡分配 IP 并测试

```
# dhclient enxbe9dcd2a0478
```

命令说明:

dhclient 网卡名

测试指定 USB 网卡访问网络:

```
# ping 8.8.8.8 -Ienxbe9dcd2a0478
```

命令说明:

ping IP 地址 -I 网卡名


```

$GPRMC,060458.00,A,3032.279873,N,10403.833894,E,0.0,200.2,190526,1.9,W,A,V*5A
$GPGSA,A,3,06,11,21,29,,,,,,,,,4.0,3.9,0.9,1*26
$GPGSV,2,1,08,05,64,270,18,06,19,080,35,09,04,036,18,11,50,045,36,1*64
$GPGSV,2,2,08,12,29,230,19,15,13,194,32,21,70,009,24,29,22,317,28,1*69
$GPGGA,060459.00,3032.279916,N,10403.833928,E,1.04,3.9,465.4,M,-31.0,M,,*40
$GPVTG,200.2,T,202.2,M,0.0,N,0.0,K,A*21
$GPRMC,060459.00,A,3032.279916,N,10403.833928,E,0.0,200.2,190526,1.9,W,A,V*5F
$GPGSA,A,3,06,11,21,29,,,,,,,,,4.0,3.9,0.9,1*26
$GPGSV,2,1,08,05,64,270,20,06,19,080,34,09,04,036,18,11,50,045,36,1*6E
$GPGSV,2,2,08,12,29,230,19,15,13,194,33,21,70,009,23,29,22,317,31,1*67
$GPGGA,060500.00,3032.279948,N,10403.833966,E,1.04,3.9,465.5,M,-31.0,M,,*40
$GPVTG,200.2,T,202.2,M,0.0,N,0.0,K,A*21
$GPRMC,060500.00,A,3032.279948,N,10403.833966,E,0.0,200.2,190526,1.9,W,A,V*53
$GPGSA,A,3,06,11,21,29,,,,,,,,,4.0,3.9,0.9,1*26
$GPGSV,2,1,08,05,64,270,20,06,19,080,34,09,04,036,18,11,50,045,36,1*6E
$GPGSV,2,2,08,12,29,230,19,15,13,194,33,21,70,009,25,29,22,317,31,1*61
$GPGGA,060501.00,3032.279981,N,10403.834086,E,1.04,3.9,465.5,M,-31.0,M,,*4C
$GPVTG,200.2,T,202.2,M,0.0,N,0.0,K,A*21
$GPRMC,060501.00,A,3032.279981,N,10403.834086,E,0.0,200.2,190526,1.9,W,A,V*5F
$GPGSA,A,3,06,11,21,29,,,,,,,,,4.0,3.9,0.9,1*26
$GPGSV,2,1,08,05,64,270,20,06,19,080,34,09,04,036,18,11,50,045,37,1*6F
$GPGSV,2,2,08,12,29,230,19,15,13,194,33,21,69,009,25,29,22,317,32,1*6A
$GPGGA,060502.00,3032.280014,N,10403.834047,E,1.04,3.9,465.6,M,-31.0,M,,*4A
$GPVTG,200.2,T,202.2,M,0.0,N,0.0,K,A*21
$GPRMC,060502.00,A,3032.280014,N,10403.834047,E,0.0,200.2,190526,1.9,W,A,V*5A
$GPGSA,A,3,06,11,21,29,,,,,,,,,4.0,3.9,0.9,1*26
$GPGSV,2,1,08,05,64,270,20,06,19,080,34,09,04,036,18,11,50,045,37,1*6F
$GPGSV,2,2,08,12,29,230,19,15,13,194,33,21,69,009,27,29,22,317,31,1*68
$GPGGA,060503.00,3032.280045,N,10403.834091,E,1.04,3.9,465.7,M,-31.0,M,,*45
$GPVTG,200.2,T,202.2,M,0.0,N,0.0,K,A*21
$GPRMC,060503.00,A,3032.280045,N,10403.834091,E,0.0,200.2,190526,1.9,W,A,V*54
$GPGSA,A,3,06,11,21,29,,,,,,,,,4.0,3.9,0.9,1*26
    
```

2.19. docker 使用说明

◆ 说明

DA160S 系统集成已预装 docker 服务，您可以用 dockerinfo 命令查看状态。注意 docker 的根目录被配置到了 /data/docker 目录下，与默认设置不同,docker 命令的使用请自行在网上查找。

```

root@bm1684:~# docker info
Client:
 Debug Mode: false

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 19.03.8
 Storage Driver: overlay2
  Backing Filesystem: <unknown>
  Supports d_type: true
  Native overlay diff: true
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version:
 runc version:
 init version:
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 5.4.217-bm1684
 Operating System: ubuntu 20.04 LTS
 OSType: linux
    
```

第三章 系统升级

3.1. TF 卡方式烧写

◆ 说明

需要准备一张 16GB 以上的 TF 卡(推荐使用闪迪 TF 卡),并将卡格式为 fat32 格式, (如果 SD 卡上有多个分区, 只能使用第一个分区)。

► 操作步骤

Ubuntu 系统(版本 18.04、20.04)通过命令行的方式

- 步骤 1 `lsblk -l` 找到 U 盘的序列号, 如: `/dev/sdb`
- 步骤 2 `sudo umount /dev/sdb` 解除 U 盘挂载
- 步骤 3 `sudo mkfs.vfat -I /dev/sdb` 格式化 U 盘为 fat32 格式, 只有一个分

区

如果出现 attribute“partition”notfound, 不要在意, 格式化完毕以后将 TF 卡拔出。

```

root@ena:~# df -h
文件系统 容量 已用 可用 已用% 挂载点
udev          0.3G     0 0.3G    0% /dev
tmpfs         1.3G  2.0M  1.3G    1% /run
/dev/sda1    198G  174G  22G   94% /
tmpfs         0.4G     0 0.4G    0% /dev/shm
tmpfs         5.0M  4.0K  5.0M    1% /run/lock
tmpfs         0.4G     0 0.4G    0% /sys/fs/cgroup
/dev/loop0   128K  128K    0 100% /snap/barefs
/dev/loop4    7M   7M    0 100% /snap/core22/608
/dev/loop2   347M  347M    0 100% /snap/gnome-3-38-2004/115
/dev/loop9    92M   92M    0 100% /snap/gtk-common-themes/1535
/dev/loop2    64M   64M    0 100% /snap/core20/1738
/dev/loop31   45M   45M    0 100% /snap/snap-store/599
/dev/loop8    9.7M  9.7M    0 100% /snap/ltop/1417
/dev/loop12   50M   50M    0 100% /snap/snap/17033
/dev/loop10   46M   46M    0 100% /snap/snap-store/630
/dev/loop6    347M  347M    0 100% /snap/gnome-3-38-2004/119
/dev/loop7    65M   65M    0 100% /snap/gtk-common-themes/1519
/dev/sda2    513M  4.9K  513M    1% /boot/efi
tmpfs         1.3G  38K  1.3G    1% /run/user/1000
/dev/loop9   3.2G  3.2G    0 100% /var/lib/docker/overlay2/922bc8f674a58f0e3774384e5e7cf08a252b9618cda319986076c8b4a8f12ab/merged
vmbgfc-fuse  716G  393G  324G   55% /mnt/sgfs
/dev/loop13    7M   7M    0 100% /snap/core22/608
/dev/loop3    64M   64M    0 100% /snap/core20/1778
/dev/sdb      38G  38K  38G    1% /media/ena/b41c-7816
root@ena:~# sudo umount /dev/sdb
root@ena:~# sudo mkfs.vfat -F 32 /dev/sdb
mkfs.fat 4.1 (2017-01-24)
attribute "partition" not found
root@ena:~#

```

WIN10 系统格式化

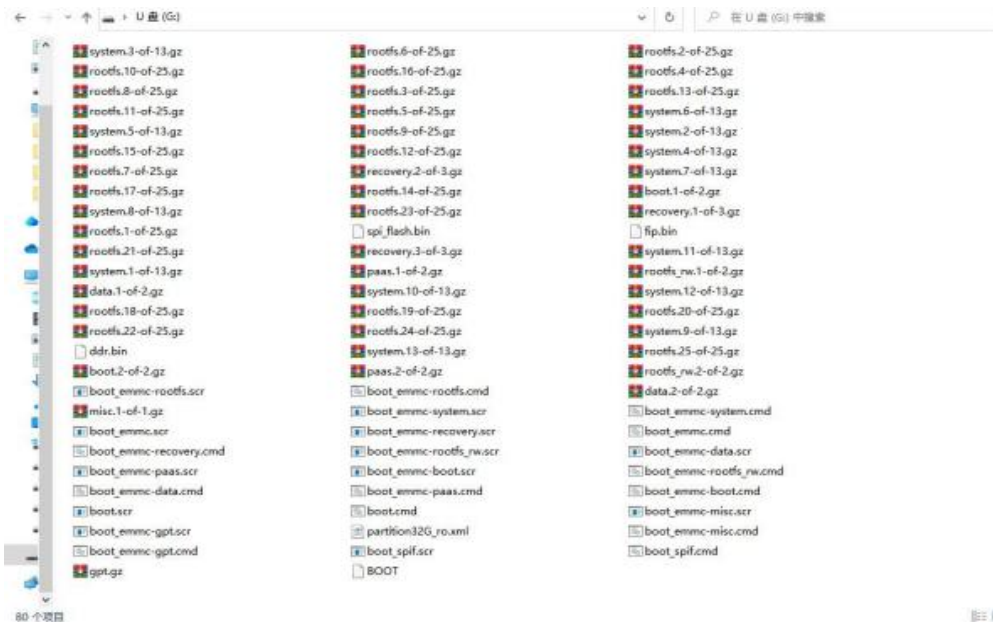
- 步骤 1 win10 右键格式化为 fat32 格式即可



升级包放置和烧录

•步骤 1 将升级文件放置到 TF 卡里面

下载压缩包，并找到固件升级包(例如:，烧写包版本以最新为准),并将其解压后的所有文件拷贝到 TF 卡的根目录下。



•步骤 2 插入 TF 卡，重新上电，待烧写完成，会看到拔掉 TF 卡并重启的提示，我们拔出 TF，重新上电即可进入新的系统。

```

MMC write: dev # 0, block # 2048, count 2048 ... 2048 blocks written: OK in 48 ms (20.8 MiB/s)
Program fip.bin done
switch to partitions #0, OK
mmc0(part 0) is current device
update @MMC
fs reading: //boot_emmc.scr
1405 bytes read in 5 ms (274.4 KiB/s)
## Executing script at 120000000
switch to partitions #0, OK
mmc0(part 0) is current device
fs reading: //boot_emmc-gpt.scr
736 bytes read in 5 ms (143.6 KiB/s)
## Executing script at 120000000
fs reading: //gpt.gz
418 bytes read in 4 ms (101.6 KiB/s)

Uncompressed size: 17408 = 0x4400

MMC write: dev # 0, block # 0, count 34 ... 34 blocks written: OK in 2 ms (8.3 MiB/s)

fs reading: //boot_emmc-boot.scr
1362 bytes read in 6 ms (221.7 KiB/s)
## Executing script at 120000000
fs reading: //boot-1-of-2.gz
15341892 bytes read in 1662 ms (8.8 MiB/s)

Uncompressed size: 102760448 = 0x6200000

MMC write: dev # 0, block # 8192, count 200704 ... 200704 blocks written: OK in 4563 ms (21.5 MiB/s)

fs reading: //boot-2-of-2.gz
30566 bytes read in 9 ms (3.2 MiB/s)

Uncompressed size: 31457280 = 0x1E00000

MMC write: dev # 0, block # 208896, count 61440 ... 61440 blocks written: OK in 1398 ms (21.5 MiB/s)

fs reading: //boot_emmc-recovery.scr
798 bytes read in 6 ms (129.9 KiB/s)
## Executing script at 120000000
fs reading: //recovery-1-of-1.gz
19872361 bytes read in 2151 ms (8.8 MiB/s)

Uncompressed size: 45600768 = 0x2B70000

MMC write: dev # 0, block # 270336, count 89064 ... █

```

```

Please remove the installation medium, then reboot
LED 'status' not found (err=-19)
Please remove the installation medium, then reboot
LED 'status' not found (err=-19)
Please remove the installation medium, then reboot
LED 'status' not found (err=-19)
Please remove the installation medium, then reboot
LED 'status' not found (err=-19)
Please remove the installation medium, then reboot
█

```

baXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

出现上述 Log,代表刷机完成

- 步骤 7 拔掉 SD 卡，重新给 DA160S 上电，成功进入系统，即升级完成。

登录账号: **linaro**

登录密码: **linaro**

3.2. 内存空间分配

◆ 说明

- DA160S 默认版本的内存，采用 LPDDR4x，总容量为 8GB，分为 4GB、4GB、共 2 颗 DDR 颗粒。

- 采用连续地址分配方式，其中

NPU 对应的 DDR 颗粒专门用于 TPU 计算加速，默认分配为 2G 内存。

VPP、JPU 对应的 DDR 颗粒专门用于视频图像编解码加速。BM1688VPU 为 0。其中 VPU 默认分配为 2G 内存，其余模块占用几百兆内存。

- 用户可将 Kernel 对应的 DDR 颗粒用于系统内存使用。CPU 的内存为 4GB(实际应该为 3.1G)，可通过 free 命令获取内存空间使用情况。

总 DDR 内存 8G：NPU+VPP+CPU=8G；