

TADC-D52-域控FAQ_V1.0

1、烧录

1.1 烧录准备

1.1.1 Q:怎么进行烧录

A:确认烧录的模块(mcu, j5, x9u), 从天准支持系统(221.224.80.18:1431)网址中获取该模块的烧录工具和烧录文档, 在文档的指导下进行烧录

1.1.2 Q:烧录文档的接线和实际使用的机器不同怎么办

A:烧录文档使用的是统一的烧录方法, 接的串口线束等可能会因为不同型号的机器不同的架构导致位置不同, 可以通过机器对应的用户手册找到需要使用的串口或接口, 用户手册也是从支持系统上可以下载到

1.1.3 Q:电脑什么系统支持烧录

A:推荐Ubuntu使用18.04以上进行烧录, Windows系统要使用Win10进行烧录, Win11烧录X9U和E3的工具有一部分问题

1.2 j5烧录报错

1.2.1 Q:uboot烧录和uart烧录有什么区别

1. Uoot

- a. 优点: 直接往EMMC烧录, 跳过DDR模块, 烧录时间较uart更短
- b. 缺点: 只能适用于有系统的J5, J5变砖时无法使用。且有概率倒计时结束仍未进入uboot模式(参考操作介绍)

2. Uart

- a. 优点: J5变砖时可以烧录进系统
- b. 缺点: 烧录时间较uboot更长, 有较高概率进入不了烧录模式, 需再次尝试。

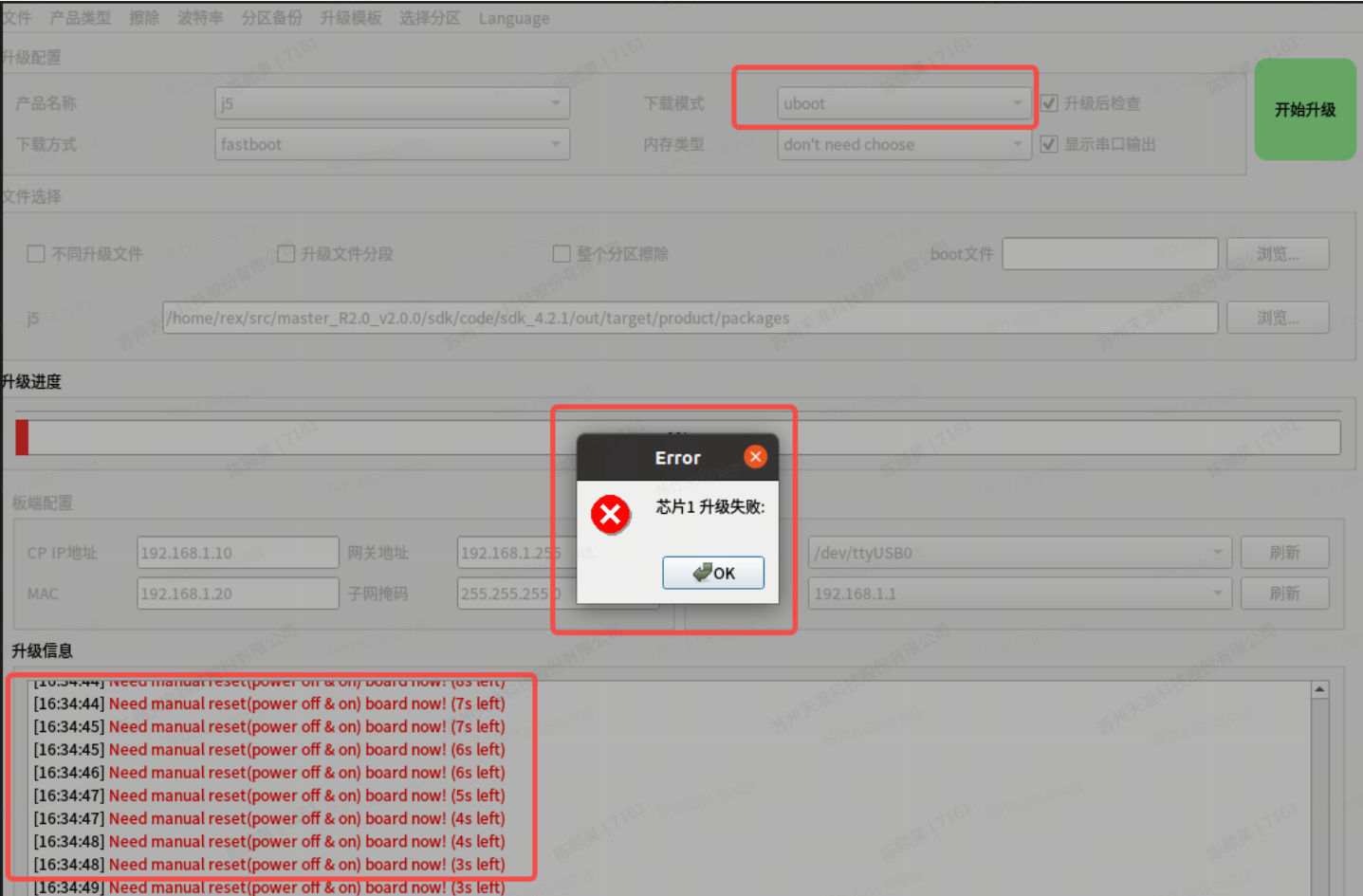
1.2.2 Q:Windows下烧录工具无法打开报错, 显示Windows无法访问指定设备, 路径或文件没有适当的权限访问该项目。

A:按照下面顺序进行排查:

1. 确认下载的压缩包是否完整

- 2. 确认Windows是否存在加密软件，工具被加密
- 3. 被杀毒工具认定为病毒，需要关闭杀毒工具
- 4. 确认是否有权限打开

1.2.3 Q:烧录到%1时重复提示Need manual reset(power off & on) board now!
直至报错，烧录失败



A:报错原因是无法识别到复位，问题排查如下：

- 1. 查看/ttyUSB0是否被minicon打开，导致烧录工具无法调用串口；
- 2. 查看串口线是否连接/有无松动

正常情况如下：

文件

产品类型

擦除

波特率

分区备份

升级模板

选择分区

Language

升级配置

产品名称

j5

下载模式

uboot

☒ 升级后检查

下载方式

fastboot

内存类型

don't need choose

☒ 显示串口输出

开始升级

文件选择

☐ 不同升级文件

☐ 升级文件分段

☐ 整个分区擦除

boot文件

浏览...

j5

/home/rex/src/master_R2.0_v2.0.0/sdk/code/sdk_4.2.1/out/target/product/packages

浏览...

升级进度

1%

板端配置

CP IP地址

192.168.1.10

网关地址

192.168.1.255

MAC

192.168.1.20

子网掩码

255.255.255.0

PC配置

串口

/dev/ttyUSB0

刷新

UDP IP

192.168.1.1

刷新

升级信息

[17:03:13] Need manual reset(power off & on) board now! (59s left)

[17:03:13] [com] tztek@TADC-D52:~#

[17:03:13] [com] tztek@TADC-D52:~#

[17:03:13] Need manual reset(power off & on) board now! (59s left)

[17:03:14] [com] tztek@TADC-D52:~#

[17:03:14] [com] tztek@TADC-D52:~#

[17:03:14] Need manual reset(power off & on) board now! (58s left)

[17:03:14] [com] tztek@TADC-D52:~#

[17:03:14] [com] tztek@TADC-D52:~#

[17:03:14] Need manual reset(power off & on) board now! (57s left)

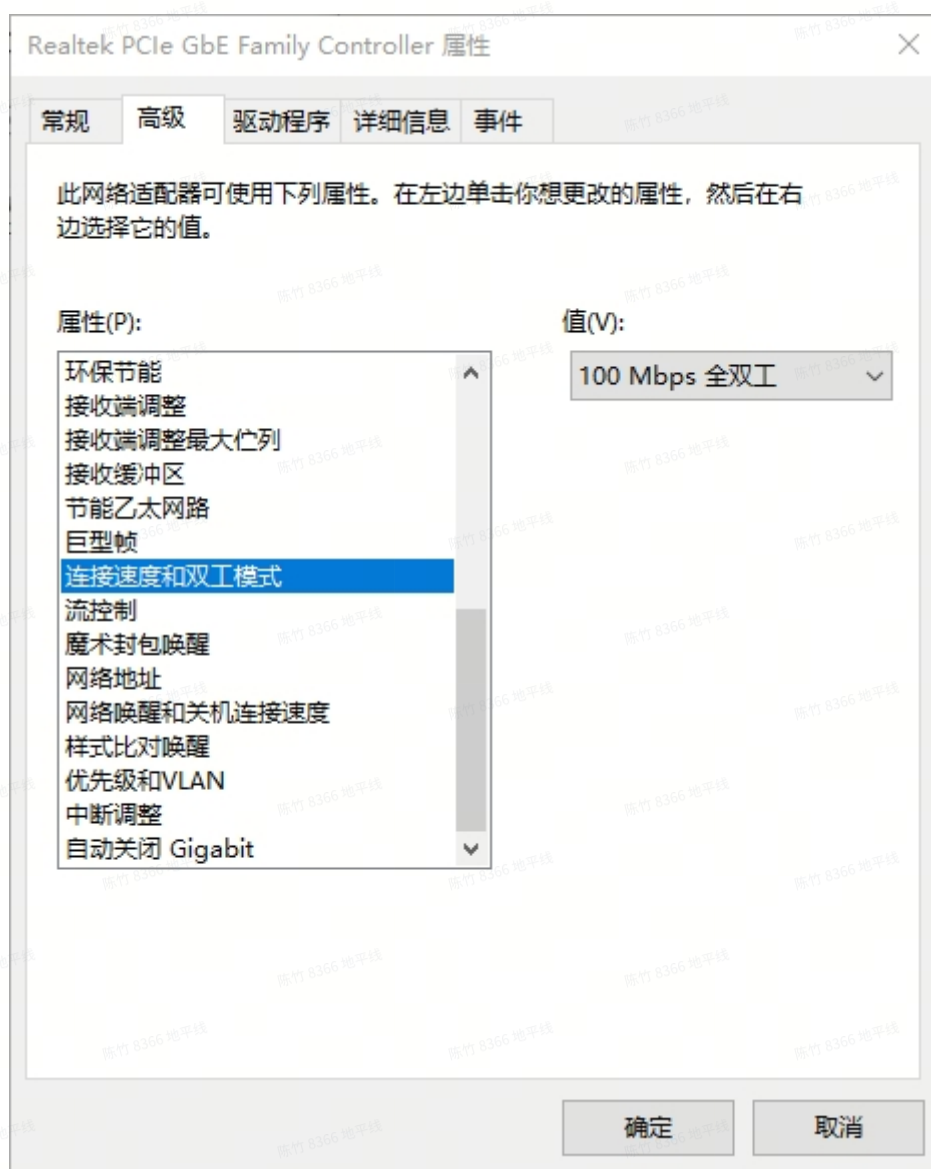
1.2.4 Q:烧录到 35% 报错，烧录失败

A:板端的J5和PC网络没有连通，排查方向如下：

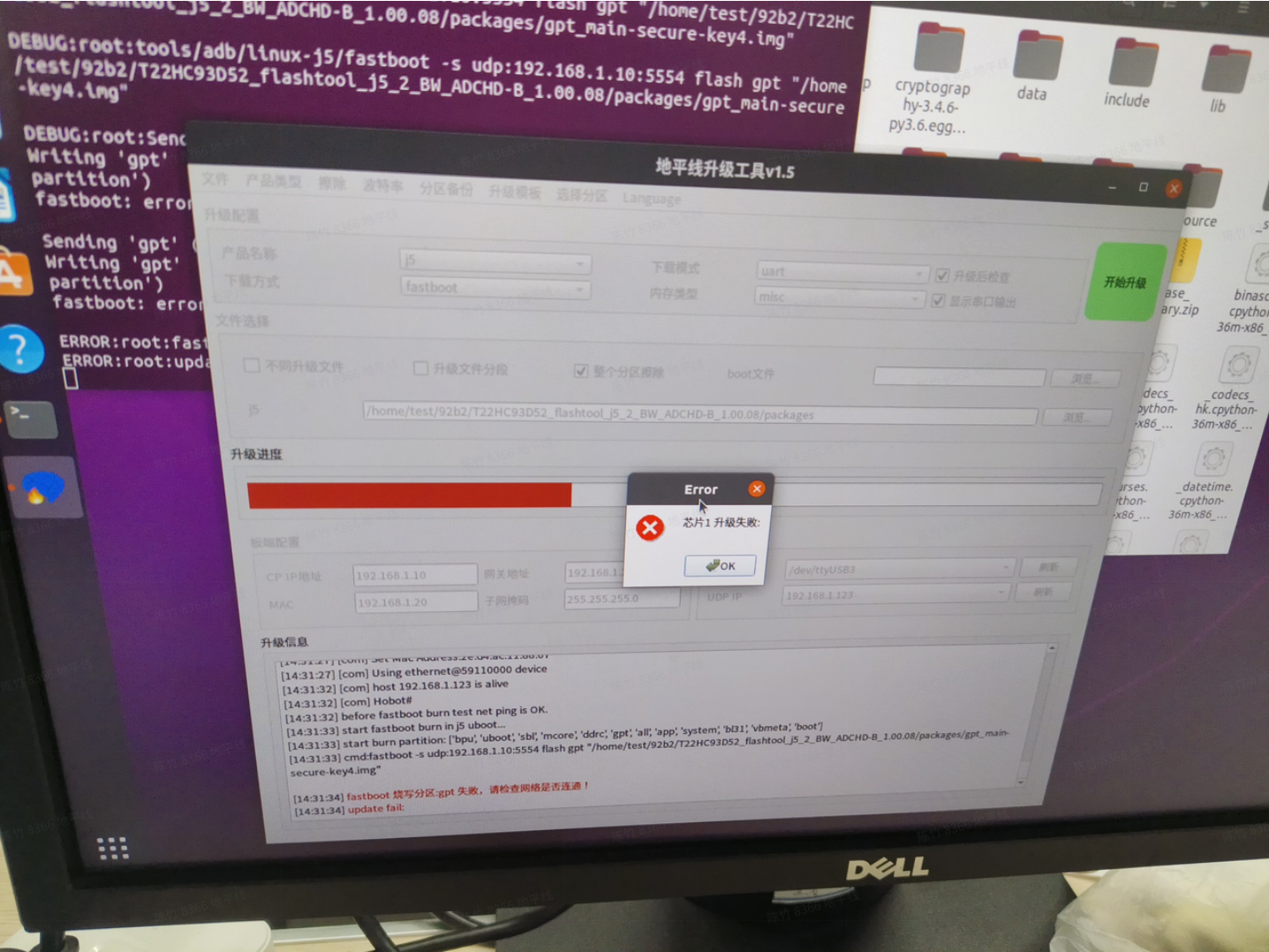
1. 排查网线和车载以太网线是否连接错误（主用左上从用右下）；
2. PC IP地址是否设置为烧录1网段，烧录工具的IP、mac等设置为1网段；
3. 车载盒的master 和 slave 模式是否设置正确，PC网口是否支持1000M，使用的车载以太网盒1000M和100M是否选择正确；
4. Windows下需要关闭防火墙



5. Windows下可在下图路径中重新配置一下连接速度和双工模式



1.2.4.1 Q:上述烧录通过，但是烧录第一个文件后报错



A:如图所示，说明已经开始烧录，但是往EMMC烧录失败，可能是烧录包与要烧录的硬件不一致，比如A1的硬件烧录B1的烧录包

1.2.4.2 Q:烧录几个文件后停止或烧录失败

文件 产品类型 擦除 波特率 分区备份 升级模板 选择分区 Language

升级配置

产品名称	j5	下载模式	uart	☒ 升级后检查
下载方式	fastboot	内存类型	misc	☒ 显示串口输出

开始升级

文件选择

☐ 不同升级文件 ☐ 升级文件分段 ☐ 整个分区擦除

boot文件

j5

升级进度

35%

板端配置

CP IP地址	192.168.1.10	网关地址	192.168.1.255
MAC	192.168.1.20	子网掩码	255.255.255.0

PC配置

串口	/dev/ttyUSB0	刷新
UDP IP	192.168.1.1	刷新

升级信息

```
gpt_main-secure.img"
[16:09:11] cmd:fastboot -s udp:192.168.1.10:5554 flash sbl "/home/rex/src/t22hc88_r1.0_v1.0.0/sdk/code/j5_sdk_1027/out/target/product/packages/
spl.img0.bin"
[16:09:12] cmd:fastboot -s udp:192.168.1.10:5554 flash sblbak0 "/home/rex/src/t22hc88_r1.0_v1.0.0/sdk/code/j5_sdk_1027/out/target/product/packages/
spl.img0.bin"
[16:09:12] cmd:fastboot -s udp:192.168.1.10:5554 flash sblbak1 "/home/rex/src/t22hc88_r1.0_v1.0.0/sdk/code/j5_sdk_1027/out/target/product/packages/
spl.img0.bin"
```

A:35%之后使用的是网络烧录的方式，所以可能和网络环境有关，问题排查步骤如下：

1. 网线连接松动或网线存在问题
2. PC的IP地址变动
3. 板子电源松动
4. 没有做好散热，导致板子温度过高

1.2.5 Q:can not into uboot!

文件产品类型擦除波特率分区备份升级模板选择分区Language

升级配置

产品名称j5

下载方式fastboot

下载模式uart

内存类型misc

☒ 升级后检查

☒ 显示串口输出

开始升级

文件选择

☐ 不同升级文件☐ 升级文件分段☐ 整个分区擦除

boot文件

浏览...

j5

/home/rex/src/t22hc88_r1.0_v1.0.0/sdk/code/j5_sdk_1027/out/target/product/packages

浏览...

升级进度

68%

板端配置

CP IP地址192.168.1.10网关地址192.168.1.255

MAC192.168.1.20子网掩码255.255.255.0

PC配置

串口/dev/ttyUSB1刷新

UDP IP192.168.1.1刷新

升级信息

[16:18:42] [com] Flashing sparse image at offset 4900032

[16:18:42] [com] Flashing Sparse Image

[16:18:42] [com] wrote 33595392 bytes to 'system_b'

[16:18:42] [com] Starting download of 33550848 bytes

[16:18:42] [com]

[16:18:42] [com]

[16:18:42] [com]

[16:18:42] [com]

[16:18:42] [com] downloading of 33550848 bytes finished

[16:18:42] [com] Flashing sparse image at offset 4908

[16:18:42] can not into uboot!

[16:18:42] update fail-

A:烧录到70%左右或以上出现如图报错和“can not into uboot!”提示，此为烧录工具bug，实际已经烧录成功

1.3 X9U烧录

支持系统有相关的烧录文档和视频，参照学习烧录即可

1.4 E3烧录

支持系统有相关的烧录文档和视频，参照学习烧录即可

1.5 烧录后查看

1.5.1 Q:如何查看版本号

A:使用如下命令查看：

```
1 cat /proc/version
```

1.5.2 Q:如何查看主从信息

A:在双J5机器中，存在master和slave，查看主从信息命令如下：

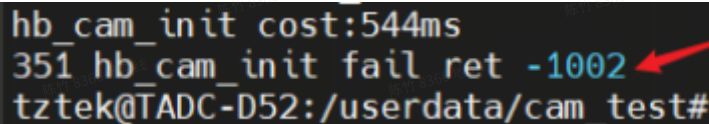
```
1 cat /proc/dtb
```

2、相机

此文档只针对由天准适配后释放给客户的相机问题的排查方法

2.1常见错误码及解决办法

2.1.1 Q:error_code: -1002



```
hb_cam_init cost:544ms
351 hb_cam_init fail ret -1002
tztek@TADC-D52:/userdata/cam_test#
```

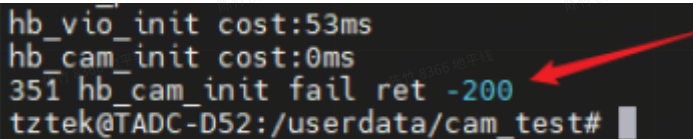
A red arrow points to the error code -1002.

错误原因：寄存器写入错误

解决方法：在往相机写入寄存器时报错，i2c 没有识别到相机，有以下原因：

- a. 相机接法有误导致，请检查相机接法；
- b. 可能是由于开机自启动脚本跑了一些相机相关的程序，导致相机节点被占用。
- c. 相机配置文件使用错误，确定使用的相机配置是否与相机匹配；

2.1.2 Q:error_code: -200

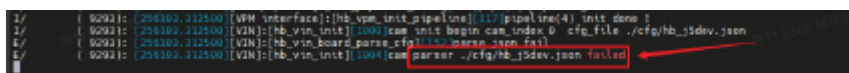


```
hb_vio_init cost:53ms
hb_cam_init cost:0ms
351 hb_cam_init fail ret -200
tztek@TADC-D52:/userdata/cam_test#
```

A red arrow points to the error code -200.

错误原因：cfg 中 json 配置文件格式不对

解决方法：大概率是由于修改配置文件的时候少加了逗号、冒号等导致，可以通过 logcat查看具体是哪个文件不对，如下图所示：



```
00031: [202302.112500] [VIM interface]:[hb_vpm_init_pipeline][11] pipeline(4) init done !
00031: [202302.112500] [V2N]:[hb_vpm_init][1000] cam_init begin cam_index 0 cfg_file ./cfg/hb_j5dev.json
00031: [202302.112500] [V2N]:[hb_vpm_board_parse_cfg][11] parse cfg fail
00031: [202302.112500] [V2N]:[hb_vpm_init][1004] cam parser ./cfg/hb_j5dev.json failed
```

A red arrow points to the log message "cam parser ./cfg/hb_j5dev.json failed".

可以看到解析 cfg/hb_j5dev.json 报错，请检查 hb_j5dev.json 格式或者使用我司提供的没有修改的配置尝试是否能点亮，若能点亮对比差异点。

也可能是 cfg/hb_j5dev.json 的 mipi 配置文件找不到，如下图所示：

```
"data_path": "./cfg/cim_config.json",  
"config_path": "./cfg/hb_mipi_ovx3c_raw12_%dfps_%dP.json"
```

将对应mipi 配置文件放到设定位置即可。

2.1.3 Q:error_code: -5018

```
reset_power cost:1136ms  
hb_vio_init cost:87ms  
341 hb_vio_init fail ret -5018  
tztek@TADC-D52:/userdata/cam_test#
```

错误原因：找不到 isp 库文件

解决方法：在 cfg/vpm_config.json 里面由关于 isp 库的路径，如下图所示：

```
"isp_dma_output_format": 8,  
"calib mode": 1,  
"calib_lname": "./cfg/lib_x3c60.so",  
"isp_algo_state": 1  
},  
"vpm": {
```

通过logcat 也可以看到打印找不到 isp 库文件，如下图所示：

```
(30633): hb_cal_open_lib -- 112  
(30633): ./cfg/lib_x3c100.so dlopen is failed: ./cfg/lib_x3c100.so cannot open shared object file: No such file or directory  
(30633): get ./cfg/lib_x3c100.so is failed, [null]  
(30633): [267711.460250][ISP]:[isp_entity_init][285]Pipe(0) isp channel 4, isp_calib update failed.
```

设定路径没有 isp 库，一般是由于误删导致，请将 isp 库放到指定目录。

2.1.4 Q:error_code: -4006

```
hb_vio_init cost:51ms  
hb_cam_init cost:3ms  
351 hb_cam_init fail ret -4006  
tztek@TADC-D52:/userdata/cam_test#
```

错误原因：找不到 cim_config.json 文件或者 cim_config.json 格式不对

解决方法：我们使用 logcat 查看具体是由于什么报错，一般有两种情况：第一种为找不到 cim_config.json，下图为 logcat 打印：

```
(11316): [267443.250000][hb cim]:cim cfg: ./cfg/cim_config.json  
(11316): [267443.250000][hb cim]:vio open ./cfg/cim_config.json fail!!  
(11316): [267443.250000][VIN]:[hb vin_init][1053]hb vin cim init error! ret -4006
```

在cfg/hb_j5dev.json 由关于 cim_config.json 的路径，对应路径下没有 cim_config.json 文件，将 cim_config.json 放到设定路径即可：如下图所示：

```
"deserial_port": 3,  
"data_path": "./cfg/cim_config.json",  
"config_path": "./cfg/hb_mipi_ovx3c_raw12_%dfps_%dP.json"  
}
```

第二种为 cim_config.json 文件格式有问题，我们检查我们是不是少了逗号或者其他格式上的问题，logcat 会有如下图所示打印：

```
(13138): [267748.000000][hb_cim]:cim_cfg: ./Cfg/cim_config.json  
(13138): [267748.000000][hb_cim]:Parse rootinfo json failed!  
(13138): [267748.000000][VIN]:[hb_vin_init][1053]hb_vin_cim_init error! ret -4006
```

2.1.5 Q:error_code: -301

```
hb_vio_init cost:0ms  
341 hb_vio_init fail ret -301  
tztek@TADC-D52:/userdata/cam_test#
```

错误原因：vpm_config.json 文件格式错误

解决方法：检查 vpm_config.json 文件是否哪里少个逗号或者其他格式问题。

2.1.6 Q:error_code: -302

```
reset_power cost:1137ms  
hb_vio_init cost:0ms  
341 hb_vio_init fail ret -302  
tztek@TADC-D52:/userdata/cam_test#
```

错误原因：cfg 里面没有 vpm_config.json 配置文件

解决方法：将对应 vpm_config.json 放到 cfg 文件夹中。

2.1.7 Q:error_code: -2000

```
sensor_start do not use GPIO_PORTB, extra_mode:32  
175 hb_cam_start pipe_line_id:1 fail ret -2000  
201 hb_vio_get_data , pipe_line_id:1 ret -6504  
grabImage 184 fail,ret -6504
```

错误原因：很多原因会导致，较复杂

解决方法：如果之前点亮过请使用我司提供的原始的点亮文件尝试点亮；如果第一次就遇到该报错（大概率是由于相机未寄到我司进行适配，虽然相机型号一样但模组厂有更新过固件等导致）

2.1.8 Q:error_code: -6504

错误原因：很多原因会导致，较复杂

解决方法：如果之前点亮过请使用我司提供的原始的点亮文件尝试点亮；如果第一次就遇到该报错（大概率是由于相机未寄到我司进行适配，虽然相机型号一样但模组厂有更新过固件等导致）

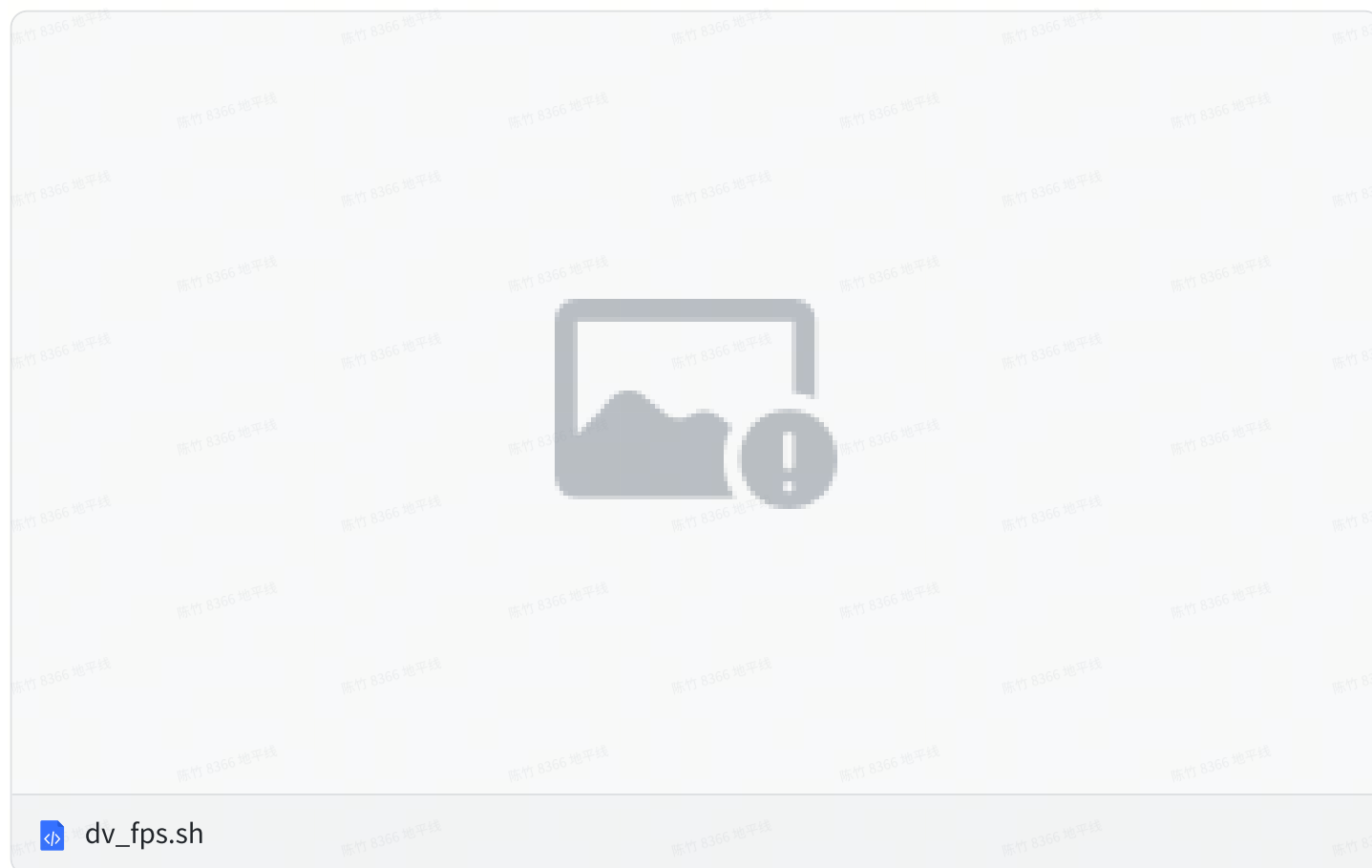
2.1.9 Q:通过错误码还是没有找到有问题的地方

A:将拿到的相机包重新解压一下，不要修改直接跑，跑通后再做其他修改，且注意修改的内容，可以知道问题可能出在哪里

2.2 相机丢帧问题

2.2.1 Q:修改帧率导致

A:如果是由我们进行适配后的相机，基本不会出现丢帧问题。如果客户自行修改相机帧率，导致超出J5的承载能力，会导致丢帧。统计帧率一版我们使用如下脚本



此脚本统计了所有通道的帧率（包含输入输出、pym、cim、isp、cimdma等）所有模块的帧率，没有error即为正常，没有丢帧现象。

2.2.2 Q:编解码导致取图丢帧

A:如果使用J5的编解码功能，编码和录制处理不会直接影响相机正常点亮的帧率，但是如果持有pym buffer不释放，可能导致出图丢帧。因此客户在编解码过程中要注意及时释放掉pym buffer。

2.2.3 Q:超过编码性能上限导致丢帧

A:如果超出J5的编解码性能，还进行编解码的话，一定会导致相机出图丢帧。（一颗J5的编码性能上限是4K*60）

2.3 Q:相机能点亮但是出图异常

A:我们提供了相关保存每路相机原始图像的可执行程序，用来排查是否在原始图像就出现问题，执行时只需要执行camera_mul_2.0 1 就可以，添加1为参数，可将每路相机图像保存在当前目录下（nv12

格式)。其次要检查是否是test.json下的每一路出图的分辨率不对导致推拉流异常

2.4 Q: 如何修改相机的帧率?

A:

1. 首先相机能调整的帧率范围在于固件所能支持的范围。
2. 相机配置文件中有关相机帧率设置的配置文件为hb_j5dev.json、hb_mipi_xxx.json文件

```
1  # cli
2  > cat config.json
3  {
4    "config_number": 1,
5    "board_name": "j5dev",
6    "comment_0": "1080p 2880*1800 20fps on host0 max9200",
7    "comment_1": "1080p 2880*1800 20fps on host2 and host3 max9212",
8    "interface_type": "mipi",
9    "ipwm_num": 4,
10   "ipwm_0": {
11     "host": "host0/3",
12   },
13   "ipwm_1": {
14     "host": "host2",
15   },
16   "ipwm_2": {
17     "host": "host3",
18   },
19   "ipwm_3": {}
20   "ipwm_index": 1,
21   "ipwm_enable": "test",
22   "trigger_mode": 4
23 }
```

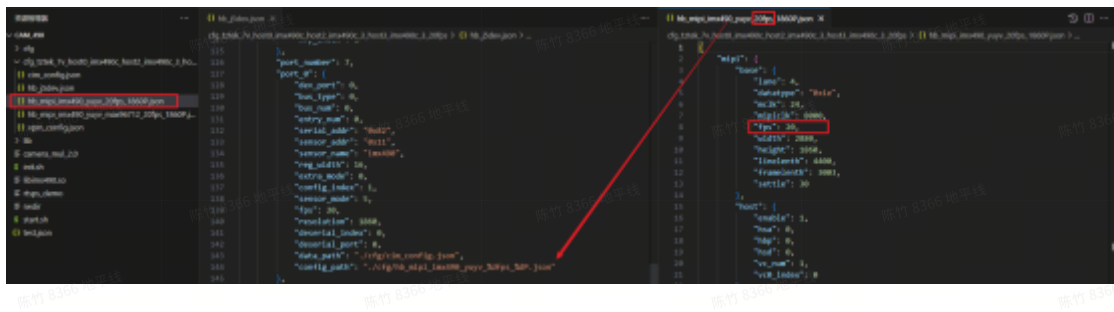
用于配置hwpwm触发信号，其中的period_us需通过相机帧率计算得出

```
1  "ipwm_0": {
2    "ipwm_index": 0,
3    "ipwm_enable": "0xf",
4    "trigger_mode": 4,
5    "offset_us": [
6      1,
7      1,
8      1,
9      1
10   ],
11   "period_us": [
12     50000,
13     50000,
14     50000,
15     50000
16   ],
17   "duty_us": [
18     10,
19     10,
20     10,
21     10
22   ]
23 },
```

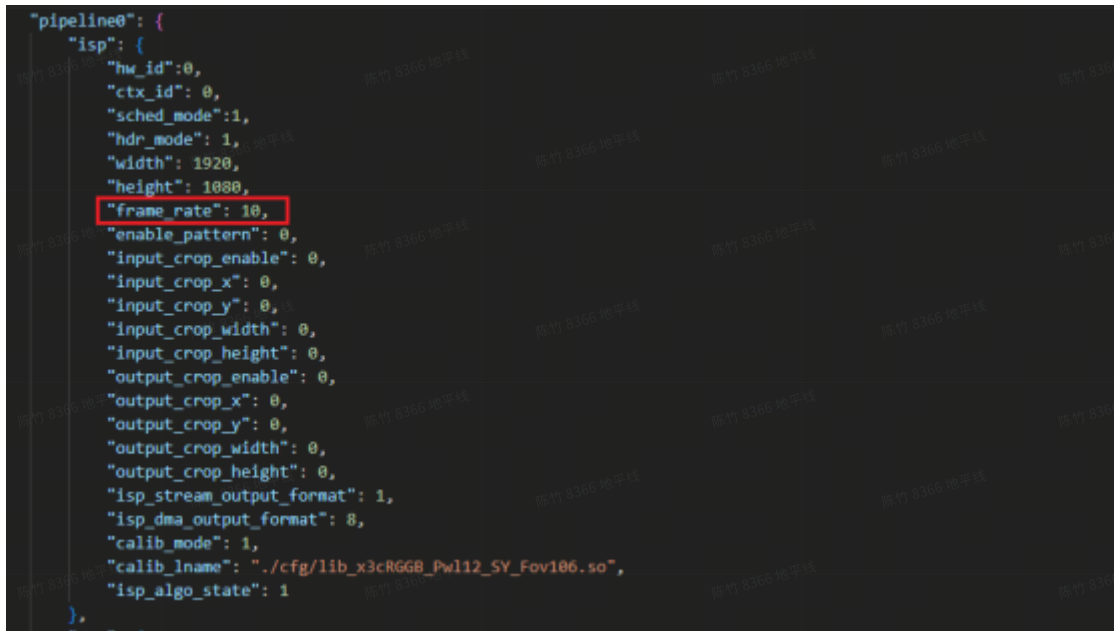
如现在20fps，而这个参数的单位为us
 $1000000/20 = 50000$

```
1  # cli
2  > cat config.json
3  {
4    "config_number": 1,
5    "board_name": "j5dev",
6    "comment_0": "1080p 2880*1800 20fps on host0 max9200",
7    "comment_1": "1080p 2880*1800 20fps on host2 and host3 max9212",
8    "interface_type": "mipi",
9    "ipwm_num": 4,
10   "ipwm_0": {
11     "host": "host0/3",
12   },
13   "ipwm_1": {
14     "host": "host2",
15   },
16   "ipwm_2": {
17     "host": "host3",
18   },
19   "ipwm_3": {}
20   "ipwm_index": 1,
21   "ipwm_enable": "test",
22   "trigger_mode": 4
23 }
```

其次还需要调整mipi配置文件中的帧率大小同时修改mipi文件的命名



Vpm_config.json中的frame_rate字段修改为对应帧率大小:



至此,调整好这些配置文件后,相机的输出帧率就会发生变化。

3、授时

3.1 GPRMC授时

3.1.1 Q:如何使用GPRMC进行授时

A:想通过GPRMC进行授时的操作具体如下:

前提:正确连接GNSS外部授时惯导工具,且直接接入惯导工具可以看到授时信息

1. 以设备和芯星通UM680A为例,正确配置后授时服务后接到MCU-RS232-1上
2. 查看GPS消息:进入J5 输入命令:

- 1 # 配置ttyS2这个节点用于接收授时信息
- 2 `stty -F /dev/ttyS2 ispeed 115200 ospeed 115200 cs8`
- 3 # 查看节点是否接收到数据
- 4 `cat /dev/ttyS2`
- 5 # 出现如下数据即为正常,有数据但是为乱码则为波特率不对

3. 查看授时服务:以该设备为例, 修改/etc/chrony.conf

```
1 # 打开chrony的配置文件
2 vi /etc/chrony.conf
```

4. 执行上面命令打开文件删除原先内容后, 替换成以下内容:

```
1 makestep 0.1 3
2 rtsync
3 allow
4 leapsectz right/UTC
5 driftfile /var/lib/chrony/drift
6 dumpdir /var/run/chrony
7 refclock PPS /dev/pps2 refid PPS lock GPSD prefer precision 1e-7 poll 2
8 refclock SHM 0 refid GPSD precision 1e-1 offset 0.9999 delay 0.2 poll 2
```

5. 最后再在J5终端执行如下命令:

```
1 # 使用gpsd接收节点收到的数据
2 gpsd -n /dev/ttyS2
3 # 重启chrony进程, 使新配置生效
4 killall -9 chronyd
5 chronyd -f /etc/chrony.conf
6 # 查看当前系统时间是否一致, 同步过来需要一点时间, 多查看几次
7 date
```

6. 判断标准: 查看J5打印出的时间是否与当前一致;

4、感知部署

4.1 Sensor

4.1.1 Q:sensor center模块启动失败

A:6v/7v/11v感知中sensor center目前只接入了相机, 因此sensor center模块无法启动主要排查相机:

1. 检查相机型号是否与指定相机型号一致;

- 2. 检查相机接法是否与相机的配置文件一致，要注意2.0和3.0的接法存在差异（相机具体的接法可参考对应的是实车搭建文档）；
- 3. 检查J5中/userdata下的init.sh脚本中是否有与点亮相机相关的程序，如果有的话需要将init.sh脚本中的进程注释掉，重启域控或者kill掉相关进程。

4.2 Vehicleio

4.2.1 Q: vehicleio模块获取不到can数据

A:vehiclio模块收到不到can数据可能的原因：

- 1. J5就没有收到底盘的can数据，需要检查硬件接线。
- 2. 检查车辆是否是启动状态，只有在启动状态底盘才会有can数据输出。
- 3. 因为2.0域控是J5有对外的can接口的，可以使用candump验证J5是否能够收到底盘can数据；而3.0域控是J5没有对外的can接口的,所以需要接mcu的can口，通过mcu转发给J5（所以3.0域控需要烧录指定的mcu版本），在3.0的感知包中，集成了相关的测试工具，可以使用工具去测试J5是能够收到can数据。

如下图所示为工具测试工具所在的目录，使用方法可参考readme文件：

组织

新建

打开

选择

uto7V_R3.0_v2.0.2_release.tar

TZTEK_Auto7V_R3.0_v2.0.2_release

test_tools

horizon

odo_debug

名称

修改日期

类型

大小

.vscode

2000/1/1 10:51

文件夹

build

2000/1/1 10:51

文件夹

cmake

2000/1/1 10:51

文件夹

config

2000/1/1 10:51

文件夹

example

2000/1/1 10:51

文件夹

include

2000/1/1 10:51

文件夹

src

2000/1/1 10:51

文件夹

CMakeLists.txt

2000/1/1 12:45

文本文档

3 KB

CMakeUserPresets.json

2000/1/1 12:45

JSON 源文件

1 KB

conanfile.py

2000/1/1 12:45

Python 源文件

4 KB

README.md

2000/1/1 12:45

Markdown 源文件

5 KB

README.pdf

2000/1/1 12:45

Adobe Acrobat ...

804 KB

- 4. 检查vehicleio的中的配置是否正确

名称	修改日期	类型	大小
dataflow	2000/1/1 10:51	文件夹	
communication_service.json	2000/1/1 8:44	JSON 源文件	2 KB
signalthreshold.json	2000/1/1 8:44	JSON 源文件	1 KB
timer.json	2000/1/1 8:44	JSON 源文件	1 KB
user_config.json	2024/1/16 20:15	JSON 源文件	1 KB
user_config_1j5.json	2000/1/1 8:44	JSON 源文件	1 KB
user_config_fillback.json	2000/1/1 8:44	JSON 源文件	1 KB
user_config_fillback_1j5.json	2000/1/1 8:44	JSON 源文件	1 KB
vehicleio_chassis_ipcf.json	2000/1/1 8:44	JSON 源文件	6 KB
vehicleio_chassis_socketcan.json	2000/1/1 8:44	JSON 源文件	6 KB
vehicleio_chassis_spi.json	2023/5/12 5:48	JSON 源文件	6 KB
vehicleio_chassis_spi_1j5.json	2000/1/1 8:44	JSON 源文件	6 KB
vehicleio_fillback.json	2000/1/1 8:44	JSON 源文件	6 KB
vehicleio_fillback_1j5.json	2000/1/1 8:44	JSON 源文件	6 KB
vehicleio_radar_eth.json	2000/1/1 8:44	JSON 源文件	6 KB

需要注意的是，2.0域控使用socket_can通信，3.0使用的是eth通信具体需要修改的地方如下图所示：

```

ata failed
[E][1680][06-10][17:18:50:625][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
h[E][1680][06-10][17:18:50:705][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:50:705][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
[E][1680][06-10][17:18:50:786][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:50:786][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
[E][1680][06-10][17:18:52:393][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:393][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
[E][1680][06-10][17:18:52:433][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:433][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
[F][1680][06-10][17:18:52:473][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:473][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
d[E][1680][06-10][17:18:52:513][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:514][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
e[E][1680][06-10][17:18:52:554][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:554][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
[E][1680][06-10][17:18:52:594][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:594][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
[E][1680][06-10][17:18:52:634][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:634][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
[E][1680][06-10][17:18:52:674][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:674][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
[E][1680][06-10][17:18:52:715][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:715][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
[E][1680][06-10][17:18:52:755][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:755][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
[E][1680][06-10][17:18:52:795][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:795][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
[E][1680][06-10][17:18:52:835][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:835][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
[E][1680][06-10][17:18:52:875][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:875][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
[E][1680][06-10][17:18:52:916][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:916][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!
[E][1680][06-10][17:18:52:956][spiha_intf.cpp:703][vehicle_service][vehicleio] SPI : Get CAN/CANFD raw data failed
[E][1680][06-10][17:18:52:956][canframes_spi.cpp:44][vehicle_service][vehicleio] GetRawDataNew : rev error!

```

- a. vehicle_service启动默认日志等级是4，只打印error错误，调试时可以先修改init脚本中的日志等级为0

修改方法：

```

Java
1 #!/bin/sh
2
3 ROOT_DIR="/mnt/vehicleio"
4
5 export LD_LIBRARY_PATH=$ROOT_DIR/lib:$LD_LIBRARY_PATH
6 #export LD_LIBRARY_PATH=/app/lib:$LD_LIBRARY_PATH
7
8 $ROOT_DIR/bin/vehicle_service $ROOT_DIR/config/car/vehicleio.json 2 0 > /dev/null 2>&1 &

```

- b. 先获取vehicleio获取can信号的通信方式，vehicleio支持SPI、ETH、SocketCAN（J5直连Can总线）

- c. 确认vehicleio的配置是否正确，需要注意can_source与hal层的通信方式对应，比如配置为4，则为socketcan，需要确认hal层的socket_can配置是否正确

vehicleio.json:

```
1 {
2   "version": "0.0.48",
3   .....
4   "running_mode" : 1,
5   "running_mode_comment" : "bit 000 , represent example rawdata",
6   "source_comment" : "0 zmq 1 eth 2 spi4 3 virtual 4 socketcan",
7   "ts_source_comment" : "SENSOR = 0, OS = 1",
8   "dump_origin_data" : 0,
9   "car" : {
10    // "0 zmq 1 eth 2 spi4 3 virtual 4 socketcan"
11    "can_source" : 4,
12    // "SENSOR = 0, OS = 1"
13    "ts_source" : 1,
14    // 根据dbc文件生成的can_xxx.json文件路径
15    "filterfile" : "/mnt/6V/vehicleio/config/car/can_xxx.json",
16    "use_internal" : 0,
17    "filter_frame" : 0
18  },
19  .....
20  "hal": {
21    "ethintf" : { // tcp参数配置, 指定MCU的IP地址和端口号
22      "ip": "192.168.2.50",
23      "port" : 7500,
24      "connect_type" : "client",
25      "timeout" : 50
26    },
27    "spi" : { // spi参数配置
28      "libspi_path" : "/system/lib/libhbspi4.so",
29      "canpercept_rcv_time" : 10,
30      "canextra_rcv_time" : 10
31    },
32    "socket_can" : { // socketcan参数配置, 修改socketcan节点
33      "dev_name" : "can3",
34      "filter_canid" : [
35      ]
36    }
37  }
```

d. 接下来可以参照4.1.3.3章节

4.2.2 Q:Vehicleio模块无法正常启动

A:可能是误改了感知包中的vehicleio的配置文件, 可以重新解压一份, 替换vehicleio模块。

4.2.3 Q: 3.0域控的感知包中vehiclio不支持解析can的扩展帧?

A: 需要修改mcu端协议中的frame type的值; 如下图协议中规定, 当frame type中的这个值为0时, 支持解析can的标准帧和拓展帧, 值为2时, 支持解析canfd的标准帧和拓展帧。

can2eth 协议

can2eth_msg 字段说明:

字段	长度(Byte)	说明
startFlag0	1	固定值 0xa5
startFlag1	1	固定值 0x3c
startFlag2	1	固定值 0x96
startFlag3	1	固定值 0x5a
startFlag4	1	固定值 0xa5
frameType	1	can=0,canfd=2

 can2eth协议.docx

count	1	count from 0-255
frameType	1	CanIdFormat type:

		CAN_STANDARD=0 CAN_EXTENDED=0 CAN_FD_STANDARD=2 CAN_FD_EXTENDED=2
channel	1	2 represents CAN2Eth

can3 0.pac 这个mcu固件就是只支持解析can的标准帧和拓展帧。



can3 0.pac
645.17KB



4.3 Odometry

4.3.1 Q:Odometry模块无输出

1. Odometry模块主要进行can数据的解析，是以vehicleio模块的输出作为输入，首先根据4.2检查vehiclio的是否有输出；
2. 检查odo模块的配置文件是否正确，odo模块使用的配置文件与vehicleio模块一致，注意检查配置的路径是否正确；
3. 需要注意的是，2.0域控使用socket_can通信，3.0使用的是eth通信具体需要修改的地方如下图所示：

```
知包 > TV > TZTEK_AutoTV_R3.0_v2.0.2_release.tar > TZTEK_AutoTV_R3.0_v2.0.2_release > horizon > odometry > configs > vehicleio > car > SuperDrive > {} vehicleio.json > ...  
{  
  "version": "0.4.7",  
  "running_mode": 1,  
  "running_mode_comment": "bit 000, represent example rawdata",  
  "source_comment": "0 zmq 1 eth 2 spi4 3 virtual 4 socket_can",  
  "ts_source_comment": "SENSOR = 0, OS = 1",  
  "car": {  
    "can_source": 1,  
    "ts_source": 0,  
    "filterfile": [  
      "/app_param/vehicle_io/can_byd.json",  
      "configs/vehicleio/car/SuperDrive/can_byd.json",  
      "configs/vehicleio/car/SuperDrive/can_lixiang.json",  
      "configs/vehicleio/car/SuperDrive/can_changan.json"  
    ],  
    "use_internal": 0,  
    "filter_frame": 1  
  },  
  "radar": {  
    "can_source": 1,  
    "ts_source": 1,  
    "filterfile": [  

```

```
    "support_parse": 1  
  },  
  "hal": {  
    "ethintf": {  
      "ip": "192.168.55.110",  
      "port": 9527,  
      "connect_type": "client",  
      "timeout": 50,  
      "wait_reconnect": 2000  
    },  
    "spi": {  
      "libspi_path": "/lib/libhbspi4.so",  
      "canpercept_recv_time": 10,  
      "canextra_recv_time": 10  
    },  
    "socket_can": {  
      "dev_name": "can0",  
      "filter_canid": [  
      ]  
    },  
    "protocol": 1  

```

4. 没有进行时间同步，odo会丢弃掉can数据，导致无输出。

4.3.2 Q:Odo模块输出的can信号异常

A:主要分为以下两种情况:

- 1. 某些can信号获取不到, 可能原因:
 - a. 底盘发出的can数据不全;
 - b. dbc转出的json文件中canID配置错误。
- 2. can信号值与实际不一致, 可能原因:
 - a. dbc转出的json文件中有误, 需要根据车辆原始的dbc文件进行检查修改, 确保每个字段都与dbc描述一致;
 - b. dbc文件中大小端的问题;
 - c. 提供的dbc文件有问题。

4.3.3 Q:odo测试轨迹异常

A:odo轨迹异常主要是因为odo解析出来的数据不对, 需要检查dbc转出的json文件是否正确; 可以通过感知包中的测试工具进行验证, 具体的测试方法可参考《ODO调试流程规范》。

https://alidocs.dingtalk.com/i/nodes/AR4GpnMqJzYM10n0c79KQYq6VKe0xE3?utm_scene=team_space

组织	新建	打开	选择
Auto7V_R3.0_v2.0.2_release.tar	TZTEK_Auto7V_R3.0_v2.0.2_release	test_tools	horizon
		odo_debug	
名称	修改日期	类型	大小
.vscode	2000/1/1 10:51	文件夹	
build	2000/1/1 10:51	文件夹	
cmake	2000/1/1 10:51	文件夹	
config	2000/1/1 10:51	文件夹	
example	2000/1/1 10:51	文件夹	
include	2000/1/1 10:51	文件夹	
src	2000/1/1 10:51	文件夹	
CMakeLists.txt	2000/1/1 12:45	文本文档	3 KB
CMakeUserPresets.json	2000/1/1 12:45	JSON 源文件	1 KB
conanfile.py	2000/1/1 12:45	Python 源文件	4 KB
README.md	2000/1/1 12:45	Markdown 源文件	5 KB
README.pdf	2000/1/1 12:45	Adobe Acrobat ...	804 KB

如果开始测试odo轨迹正常, 接入像激光雷达这些设备, 会导致网络负载过大, 增加延迟, 从而导致odo轨迹异常。

4.3.4 Q:没有接入can数据的时候, 怎么模拟ODO数据

A:在odometry模块的configs/global.json文件中, 将run_mode字段改成1 (原始是0), 这个是模拟ODO数据的接收, 直接从ODO模块模拟一条odo数据出来

```
D: > 域控 > 6V感知部署 > 6v_perception > app > 6V > odometry > configs > {} global.json > ...
1  {}
2  "log": {
3    "level": 0,
4    "tag": "odom",
5    "enable_save": 1,
6    "file_size": 2000000,
7    "file_count": 40,
8    "save_path": "/log/app/odometry",
9    "log_file_name": "J5_Odometry.Log"
10 },
11 "car_type": 0,
12 "car_type_desc": "0-pilot5, 1-superdrive",
13 "run_mode": 0,
14 "run_mode_desc": "0-realcar, 1-produce_data, 2-fillback_full, 3-fillback_independence",
15 "fillback": {
16   "enable_eol_imu_calib": 0,
17   "enable_onl_imu_calib": 0
18 },
19 "cpu_core": [0, 1, 2, 3, 4, 5, 6, 7],
20 "vehicleio": "configs/vehicleio/car/Galaxy/vehicleio.json",
21 "egomotion": "configs/egomotion_config.json",
22 "eol_param_cfg": "configs/config_eol.yaml",
23 "onl_param_cfg": "configs/config_onl.yaml",
24 "imu_onl_calib_cfg": "/data/calib_params/imu_onl_calib.json",
25 "imu_eol_calib_cfg": "/data/calib_params/imu_eol_calib.json"
```

改为1, 模拟ODO接收数据

4.4 adas

4.4.1 感知进程启动失败

4.4.1.1 Q:感知启动后有一路没有图像

现象：执行./init.sh一键启动脚本，感知启动之后，使用matrix-client工具查看出图，发现某一路没有图像；单独跑sensor模块,相机都是可以启动的；然后去分别启动前视、后视和周视的感知进程，发现没有出图的那一路对应的感知进程报如如下图所示错误：

```
2. 192.168.1.10 (roc) 3. 192.168.1.10 (roc) 4. 192.168.1.10 (roc) 5. 192.168.1.10 (roc) 6. 192.168.1.10 (roc) 7. 192.168.1.10 (roc)
I0101 08:03:12.971755 2501 mono_state_manager.cpp:2357] ----- create vehicle-3 thread
Use ClassName.json config: /userdata/app/horizon/adas/adas-rt/online/config_f2v/config-2/PostInputModule.json
I0101 08:03:13.223963 2210 post_input_module.cpp:678] PostInputModule::ADASModule Init finished
I0101 08:03:13.232998 2210 runtime_env_msg.cpp:320] Device Version : mono
I0101 08:03:13.233650 2210 runtime_env_msg.cpp:356] APP Version : SD-APP-7V.V6.1.4.2023.0522.1701
I0101 08:03:13.233725 2210 runtime_env_msg.cpp:394] ADAS internal Version : 1.10.128.0.0.c634d16c
I0101 08:03:13.233765 2210 hobot_adas.cpp:4689] Model version fsd_v7.0.0_byd_20230512-200931_chen.jia
I0101 08:03:13.233814 2210 runtime_env_msg.cpp:375] Model Version : fsd_v7.0.0_byd_20230512-200931_chen.jia
I0101 08:03:13.233860 2210 main.cpp:2473] start process: 2
I0101 08:03:13.233922 2210 hobot_adas.cpp:4117] Version: 1.10.128
I0101 08:03:13.235085 2210 hobot_adas.cpp:7398] BPUModelBase::ModelID_Det_Mosaic_Multitask
I0101 08:03:13.240644 2210 hobot_adas.cpp:2804] no watermark_msg
I0101 08:03:13.240732 2210 hobot_adas.cpp:4338] Enable type 28, but message null
I0101 08:03:13.240777 2210 hobot_adas.cpp:4338] Enable type 41, but message null
I0101 08:03:13.240817 2210 hobot_adas.cpp:4338] Enable type 25, but message null
I0101 08:03:13.240851 2210 hobot_adas.cpp:4338] Enable type 22, but message null
I0101 08:03:13.240963 2210 hobot_adas.cpp:4338] Enable type 31, but message null
I0101 08:03:13.241001 2210 hobot_adas.cpp:4338] Enable type 26, but message null
I0101 08:03:13.241037 2210 hobot_adas.cpp:4338] Enable type 5, but message null
I0101 08:03:13.241068 2210 hobot_adas.cpp:4338] Enable type 38, but message null
I0101 08:03:13.241094 2210 hobot_adas.cpp:4338] Enable type 23, but message null
I0101 08:03:13.241122 2210 hobot_adas.cpp:4338] Enable type 82, but message null
I0101 08:03:13.241151 2210 hobot_adas.cpp:4338] Enable type 84, but message null
E0101 08:03:13.251222 2210 adas_state_handler.cpp:231] duplicate callback, ignore
Use ClassName.json config: /userdata/app/horizon/adas/adas-rt/online/config_f2v/config-1/BPUResultReader.json
I0101 08:03:13.279455 2210 main.cpp:1350] Seg Fault by signal: 11
E0101 08:03:13.285116 2210 main.cpp:1368] Error: signal 11 8
E0101 08:03:13.285196 2210 main.cpp:1370] ./hb-f2v(+0x4707fc) [0xaaaa82907fc]
E0101 08:03:13.285252 2210 main.cpp:1370] linux-vdso.so.1[0x00000000]
E0101 08:03:13.285300 2210 main.cpp:1370] ./hb-f2v(+0x554cb4) [0xaaaa8374cb4]
E0101 08:03:13.285388 2210 main.cpp:1370] ./hb-f2v(+0x1b3a000) [0xaaaa8995a000]
E0101 08:03:13.285437 2210 main.cpp:1370] ./hb-f2v(+0x4e967c) [0xaaaa830967c]
E0101 08:03:13.285548 2210 main.cpp:1370] ./hb-f2v(+0x402e24) [0xaaaa8222e24]
E0101 08:03:13.285598 2210 main.cpp:1370] /lib/libc.so.6[0x00000000]
E0101 08:03:13.285650 2210 main.cpp:1370] ./hb-f2v(+0x46aed8) [0xaaaa828aed8]
E0101 08:03:13.285696 2210 main.cpp:1372] Catch SIGSEGV
Segmentation fault (core dumped)
tztek@TADC-D52:/userdata/app/horizon/adas# ./hobot-adas-workflow-f2v.sh
```

A: 下图中copy_backup_enable，参数为1的情况下，camera配置文件首先查找file_name对应的配置文件，如右边截图的/app_param/camera/camera-5/camera_0.json文件，若没有这个文件再使用默认的redirect字段对应的文件；参数为0的情况下，只会查找file_name对应的配置文件，若这个文件没有，则进程直接报错退出

```
global.json x
app > adas > adas-rt > online > config_f2v > config-0 > {} global.json > {} HobotADASContext > [ ] camera > {} 0 > # copy_backup_enable
53 },
54 "track-config" : [
55 {
56 "redirect" : "track_conf.json"
57 }
58 ],
59 "fpga-model" : {
60 "redirect" : "unet_model_config.json"
61 },
62 "camera" : [
63 {
64 "camera_id": 5,
65 "file_name": "/app_param/camera/camera-5/camera_0.json",
66 "redirecta" : "front_C.json",
67 "redirectb" : "camera_0.json",
68 "redirect" : "front bydovx8b.json",
69 "copy_backup_enable" : 0
70 },
71 ],
72 "veh-info" : {
73 "redirect" : "vehicle_info.json"
74 },
75 },
76 "BPUModelPreprocess" : {
```

4.4.1.2 Q:负载均衡服务段错误，导致感知进程启动失败

```
Segmentation fault (core dumped)  
tztek@TADC-D52:/userdata/app/lb_service# ./init.sh  
killall: lb_service: no process killed  
Segmentation fault (core dumped)  
tztek@TADC-D52:/userdata/app/lb_service#
```

A:将/log/app下面的*.lock删除后解决问题。注意*.lock为隐藏文件，通过以下指令删除：

```
1 # 删除所有隐藏lock文件  
2 rm -rf .*lock
```

4.4.1.2 Q:3.0 域控跑7V感知，sensor可以正常启动，周视感知进程无法启动

A:检查app_param下，周视相机的json文件格式是否有误；

4.5 感知结果相关

4.5.1 Q:感知必须接入can，融合端口才能输出吗？

A:融合端口需要接入所有摄像头数据+ODO数据，才会有输出的。

4.5.2 Q:7v 感知会输出周视的障碍物结果吗？能在matrix-client显示出来吗？

A:周视是有障碍物的，只是它没有输出，融合进程在接收到所有的摄像头数据和ODO数据后，将所有动态障碍物数据的特征合并，确认障碍物是不是同一个障碍物，若是同一个障碍物就会将障碍物信息合并成一条障碍物，同时更新其障碍物信息。

4.5.3 Q:地平线输出的障碍物结果，同一目标输出了两次

A:这两个障碍物是同一个，但是检测的部位不一样，一个类型是车尾框，另一个是全车框。

3.1 ObstacleType

ObstacleType 描述动态障碍物所具有的枚举类型

Nginx

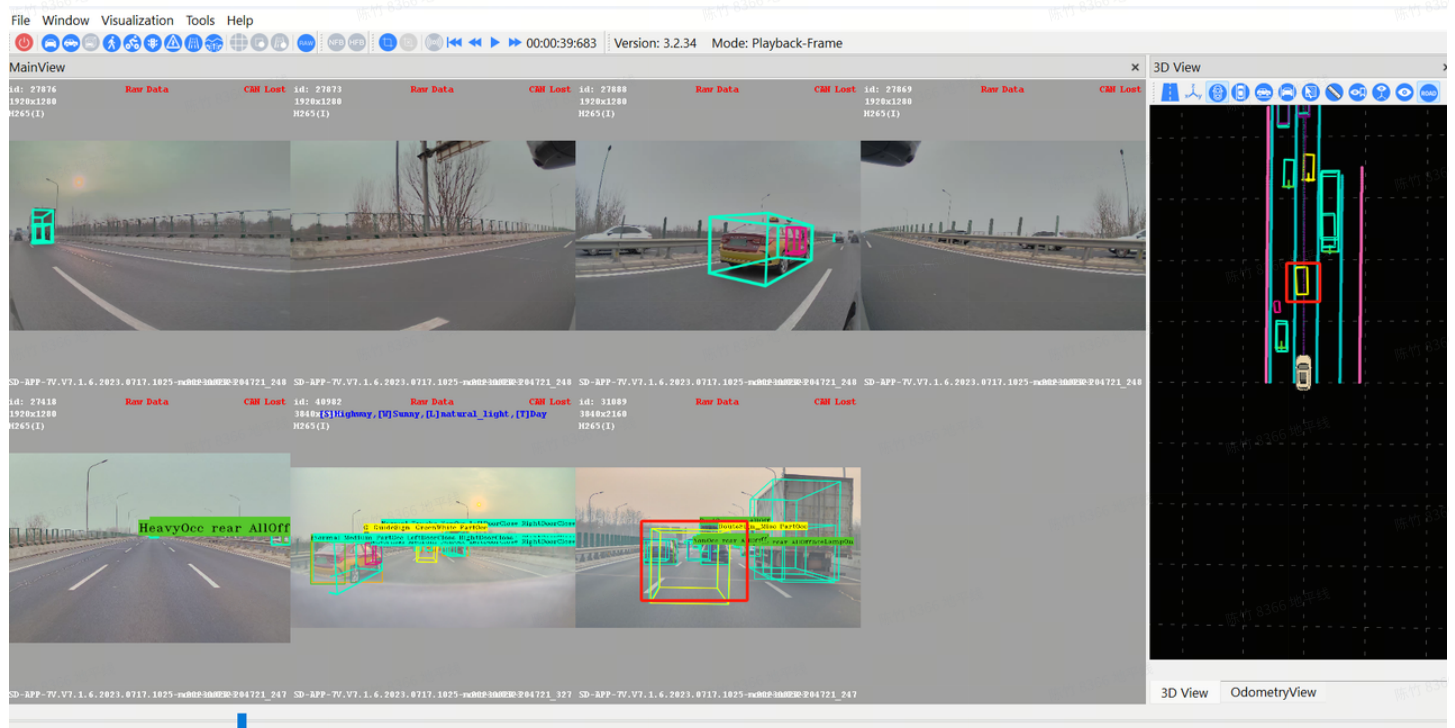
```
1 enum ObstacleType{
2   ObstacleType_VehicleRear = 0;
3   ObstacleType_VehicleFull = 1;
4   ObstacleType_Pedestrian = 2;
5   ObstacleType_Cyclist = 18;
6   ObstacleType_Bicycle = 55;
7 }
```

- **0: ObstacleType_VehicleRear(车尾框)**
- **1: ObstacleType_VehicleFull(全车框)**
- **2: ObstacleType_Pedestrian(行人)**
- **18: ObstacleType_Cyclist(骑车人)**
- **55: ObstacleType_Bicycle(无人两轮车)**

4.5.4 Q:7V感知启动之后，无法识别到路上停着的自行车

A:停着的自行车不输出,因为模型本身不检测，可以识别到动态障碍物的骑车人。

4.5.5 Q:matrix中的3D view显示虚假目标或者出现幽灵框



A:相机的内外参配置文件路径错误导致，主要是config_bev中config1-7都需要配置为对应相机的内外参文件的存放路径

v2.0.2_release > horizon > adas > adas-rt > online > config_bev >

名称	修改日期	类型	大小
config-0	2000/1/1 10:51	文件夹	
config-1	2000/1/1 10:51	文件夹	
config-2	2000/1/1 10:51	文件夹	
config-3	2000/1/1 10:51	文件夹	
config-4	2000/1/1 10:51	文件夹	
config-5	2000/1/1 10:51	文件夹	
config-6	2000/1/1 10:51	文件夹	
config-7	2000/1/1 10:51	文件夹	
config-post	创建日期: 2024/1/3 11:12 大小: 11.1 KB 文件: BEVStageOutputModule.json, BPUResultReader_97.json, ...		
global_user_co			1 KB
load_balance.json			4 KB
process.json	2000/1/1 8:44	JSON 源文件	1 KB
runtime_context.json	2000/1/1 8:44	JSON 源文件	135 KB
schedule_config.json	2000/1/1 8:44	JSON 源文件	1 KB
state_manager.json	2000/1/1 8:44	JSON 源文件	1 KB

根据global.json中的redirect字段可以判断出对应的相机，如下图所示frontnarrow_bydovx8b.json对应的是前窄相机，所以file_name要配置为前窄相机对应的内外参的json路径。

```

包 > TV > TZTEK_Auto7V_R3.0_v2.0.2_release.tar > TZTEK_Auto7V_R3.0_v2.0.2_release > horizon > adas > adas-rt > online > config_bev > config-5 > global.json > ...
    "redirect" : "engine.json"
  },
  "track-config" : [
    {
      "redirect" : "track_conf.json"
    }
  ],
  "fpga-model" : {
    "redirect" : "unet_model_config.json"
  },
  "camera" : [
    {
      "camera_id": 5,
      "file_name": "/app_param/camera/camera-5/camera_0.json",
      "redirecta" : "front_C.json",
      "redirectb" : "camera_0.json",
      "redirect" : "front_bydovx8b.json",
      "copy_backup_enable" : 1
    }
  ],
  "veh-info" : {
    "redirect" : "vehicle_info.json"
  }
},
"BPUResultPreprocess" : {
  "enable_merge_lane_parsing" : 1
},
"Real3DProcessModule": {
  "coordinate_type": 0,
  "undistort_point_method": 1
}
}

```

4.5.6 Q:地平线上位机显示的感知红绿灯结果和实际发出的红绿灯数据不一致

A:这种情况一般发生在红绿灯状态改变的时候，实际中红绿灯由红变绿，但是感知输出的红绿灯结果会在4、5帧之后才会跟着发生改变，应是地平线感知算法需要通过这几帧去判断红绿灯状态发生变化，感知输出的结果会比实际情况延后几帧。

4.6 DBC转json文件

4.6.1 Q:dbc转can.json文件失败

A: 问题排查步骤如下：

1. 首先排查，dbc文件中的字段是否填写错误；
2. dbc文件中的canID过长，导致无法解析；
3. 可能是转换工具的版本过旧，更换新版本

4.6.2 Q：有些客户的车，底盘信号中的yawrate是由两个信号描述的，一个是yawrate 数值绝对值，一个是方向的左右。这种情况要怎么处理呢？

A:需要修改下config.toml文件，加上yawratedirection，然后使用dbc转json的工具进行抓换;然后检查转出来的json文件中，yawratedirection这个信号中的value_table值按照下图修改，Right对应Neg，left对应Pos； wheelangle有这个问题，按照同样方法修改，wheelangle是 wheelangledirection；

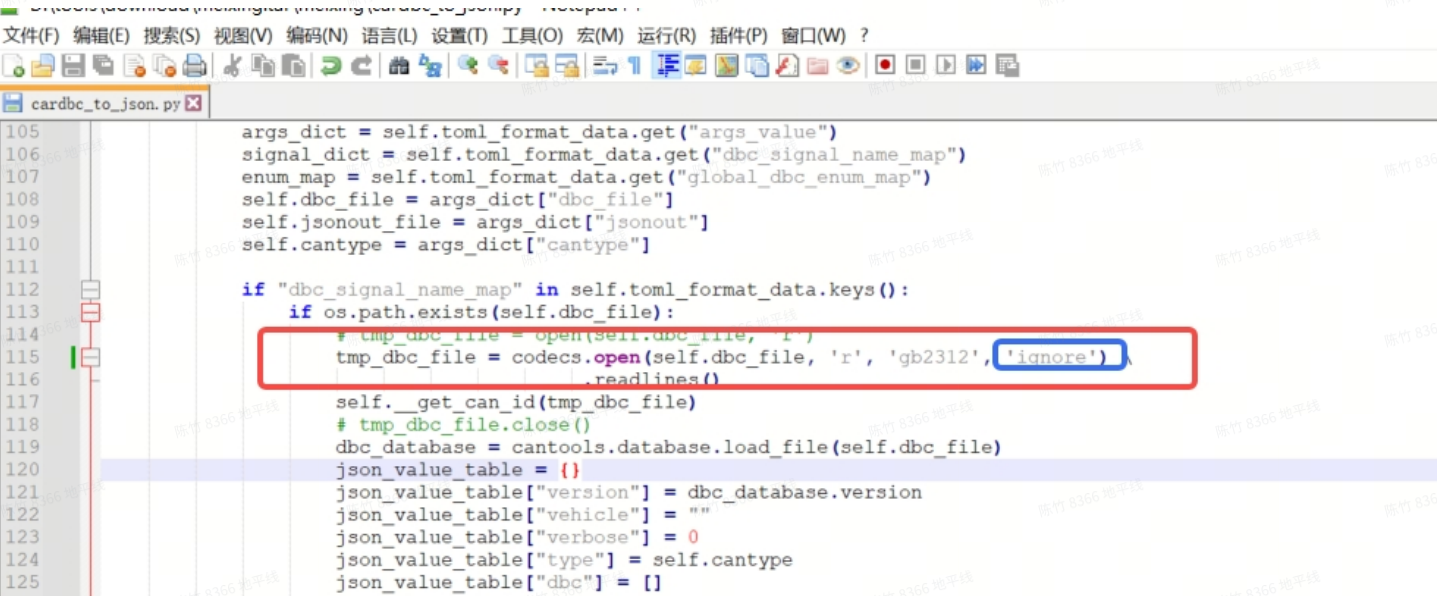
```
README.md vehicleio_chassis_spi.json vehicleio.json config.toml can_byd.json rawdata: a5
> dbc转换 > meixing > config.toml
14 yawrate_rad_s = false # dbc文件中的横摆角速度单位是否为rad/s; 若为度/s, 则填与false.
15 wheelangle_deg = true # dbc文件中的横摆角速度单位是否为度; 若为弧度, 则填写false.
16
17 # dbc signal enum name -> json enum name
18 [global_dbc_enum_map]
19 'Not Valid' = 'Invalid'
20 'invalid' = 'Invalid'
21 'valid' = 'Valid'
22 'Abnormal' = 'Invalid'
23 'Normal' = 'Valid'
24 'initial' = 'Invalid'
25 'Invalid' = 'Invalid'
26 'Stop' = 'Standstill'
27 'undefinable' = 'Invalid'
28
29 # dbc signal name -> json signal anme
30 [dbc_signal_name_map]
31 # basic signal map
32 'GE_Fahrstufe' = 'gear'
33 #'hb_GearPositionValid' = 'gearvalid'
34 'ESP_v_Signal' = 'speed'
35 #'hb_VehicleSpeedValid' = 'speedvalid'
36 'LWI_Lenkradwinkel' = 'wheelangle'
37 'LWI_QBit_Lenkradwinkel' = 'wheelanglevalid'
38 'ESP_Gierrate' = 'yawrate'
39 #'hb_YawRateValid' = 'yawratevalid'
40 'ESP_VZ_Gierrate' = 'yawratedirection'
41 'LWI_VZ_Lenkradwinkel' = 'wheelangledirection'
42
```

```
92 },
93 {
94   "name": "yawratedirect",
95   "value_type": "Enum",
96   "start_bit": 38,
97   "bit_length": 1,
98   "scale": 1,
99   "offset": 0,
100   "parser_flag": 1,
101   "timeout_time": 1000,
102   "comments": "None",
103   "range": {
104     "min": 0,
105     "max": 1
106   },
107   "value_table": [
108     {
109       "value": "0x1",
110       "desp": "Right"
111     },
112     {
113       "value": "0x0",
114       "desp": "Left"
115     }
116   ]
117 },
118 {
119   "name": "acclateral",
120   "value_type": "Enum",
121   "start_bit": 38,
122   "bit_length": 1,
123   "scale": 1,
124   "offset": 0,
125   "parser_flag": 1,
126   "timeout_time": 1000,
127   "comments": "None",
128   "range": {
129     "min": 0,
130     "max": 1
131   },
132   "value_table": [
133     {
134       "value": "0x1",
135       "desp": "Neg"
136     },
137     {
138       "value": "0x0",
139       "desp": "Pos"
140     }
141   ]
142 },
143 {
144   "name": "acclateral",
```

4.6.3 Q:dbc转json文件时报编码格式的错误？

```
cardbc_to_json.py config.toml dazhong.dbc match.txt
wc@wc:/media/wc/KIOXIA/meixing$ python3 cardbc_to_json.py config.toml
Traceback (most recent call last):
  File "cardbc_to_json.py", line 231, in <module>
    data_to_json.output_to_json()
  File "cardbc_to_json.py", line 221, in output_to_json
    json_value_table = self._dbc_to_json()
  File "cardbc_to_json.py", line 115, in _dbc_to_json
    tmp_dbc_file = codecs.open(self.dbc_file, 'r', 'gb2312') \
  File "/usr/lib/python3.6/codecs.py", line 708, in readlines
    return self.reader.readlines(sizehint)
UnicodeDecodeError: 'gb2312' codec can't decode byte 0xe4 in position 148043: illegal multibyte sequence
wc@wc:/media/wc/KIOXIA/meixing$
```

A:使用的dbc转json工具版本过旧，更换新的版本或者在工具中添加一行，如下图所示：



```
105 args_dict = self.toml_format_data.get("args_value")
106 signal_dict = self.toml_format_data.get("dbc_signal_name_map")
107 enum_map = self.toml_format_data.get("global_dbc_enum_map")
108 self.dbc_file = args_dict["dbc_file"]
109 self.jsonout_file = args_dict["jsonout"]
110 self.cantype = args_dict["cantype"]
111
112 if "dbc_signal_name_map" in self.toml_format_data.keys():
113     if os.path.exists(self.dbc_file):
114         # tmp_dbc_file = open(self.dbc_file, 'r')
115         tmp_dbc_file = codecs.open(self.dbc_file, 'r', 'gb2312', 'ignore') \
116             .readlines()
117         self._get_can_id(tmp_dbc_file)
118         # tmp_dbc_file.close()
119         dbc_database = cantools.database.load_file(self.dbc_file)
120         json_value_table = {}
121         json_value_table["version"] = dbc_database.version
122         json_value_table["vehicle"] = ""
123         json_value_table["verbose"] = 0
124         json_value_table["type"] = self.cantype
125         json_value_table["dbc"] = []
```

4.6.4 Q:车厂提供的dbc文件中有很多档位信息，转成json之后都是invalid，对最后的轨迹是否有影响？

A:感知包中用到的就P、R、N、D四个，其他的可以都给invalid.

4.7 相机相关

4.7.1 感知包都适配哪些相机型号？

A:如下图所示：

厂商	模组类型	Sensor	cam_mode	Fomat	Serializer	分辨率	FOV (°)	EFL (mm)
森云6V	前广 120°	AR0820 (RGGB)		RAW12	MAXIM MAX9295A	3840*2160	H:120° /V:66°	4.01
	周视4V 100°	AR0233 (RGGB)		RAW12	MAXIM MAX9295A	1920*1080	H:100° /V:53°	3.55
	后视 100°	AR0233 (RGGB)		RAW12	MAXIM MAX9295A	1920*1080	H:100° /V:53°	3.55
舜宇6V								
舜宇7V	前窄 30°	OX08B40 (RGGB)	BF8V105A	RAW12	MAX96717	3840*2160	D:34.4° /H:100° /V:53°	15.34
	前广 120°	OX08B40 (RGGB)	BF8V107A	RAW12	MAX96717	3840*2160	D:140° /H:120° /V:65.6°	4.1
	周视4V 100°	OX03C10	SS2V003C	RAW	MAX96717F	1920*1280	D:120° /H:100° /V:65°	3.425
	后视 60°	OX03C10	SS2V003C	RAW	MAX96717F	1920*1280	D:74.2° /H:59.7° /V:38.5°	5.84
联创7V	前窄 30°	X8B	LC870B	RAW	MAX96717	3840*2160	H:29.8°/V:16.8°	15.48
	前广 120°	X8B	LC863B	RAW	MAX96717	3840*2160	H121.2°/V67.2°	3.9
	周视4V 100°	X3C	LC287A	RAW	MAX96717F	1920*1280	H103°/V66.88°	3.326
	后视 60°	X3C	LC033P	RAW	MAX96717F	1920*1280	H:59.7°/V:38.5°	5.83
11V (舜宇7V+森云4V)	前窄 30°	OX08B40 (RGGB)	BF8V105A	RAW12	MAX96717	3840*2160	D:34.4° /H:100° /V:53°	15.34
	前广 120°	OX08B40 (RGGB)	BF8V107A	RAW12	MAX96717	3840*2160	D:140° /H:120° /V:65.6°	4.1
	周视4V 100°	OX03C10	SS2V003C	RAW	MAX96717F	1920*1280	D:120° /H:100° /V:65°	3.425
	后视 60°	OX03C10	SS2V003C	RAW	MAX96717F	1920*1280	D:74.2° /H:59.7° /V:38.5°	5.84
	环视4V 195°	ISX031(森云)	YUV422	MAX96717F		1920*1536	H:195°/V:154.5°	1.51

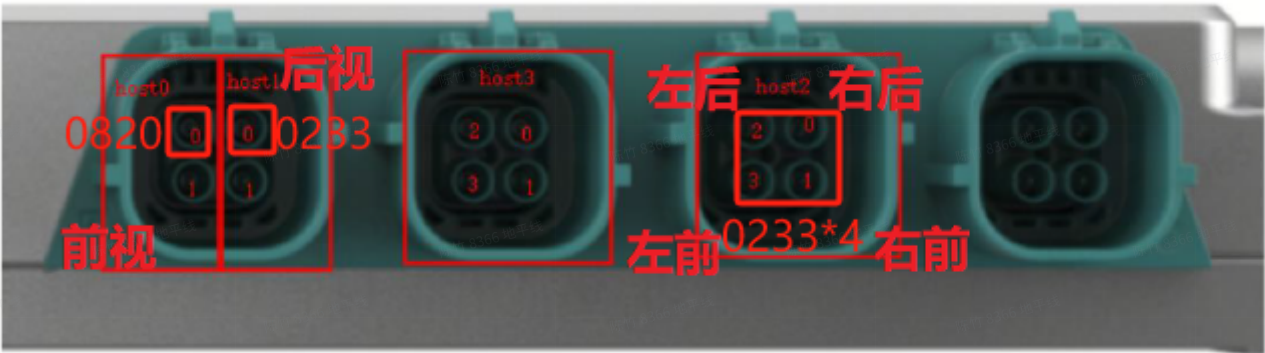
链接：https://alidocs.dingtalk.com/i/nodes/ydxXB52LJqe71rkrCbMwMe0RJqjMp697?utm_scene=team_space

4.7.2 6V/7V/11V感知相机怎样接入？

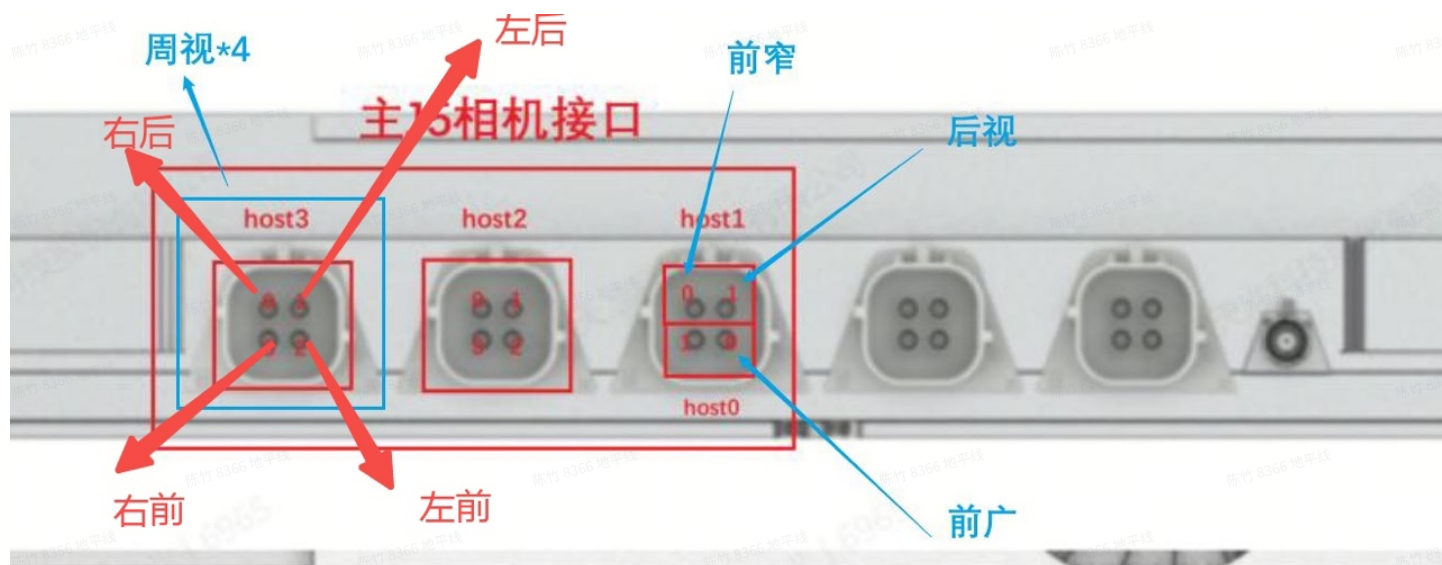
A:A:2.0/3.0 域控不同感知包下相机接法如下所示：



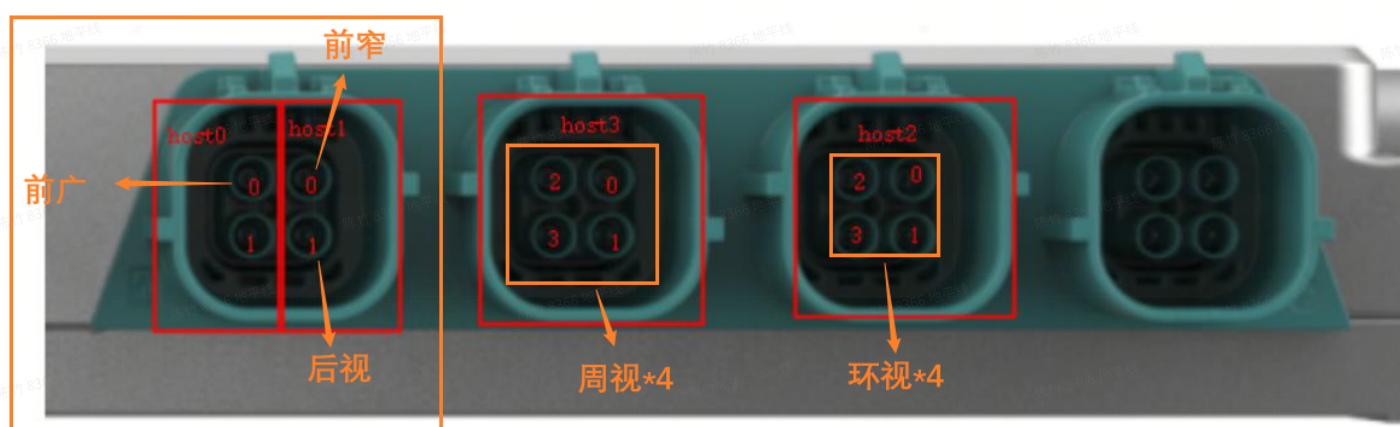
2.0域控6V相机接法



3.0域控6V相机接法

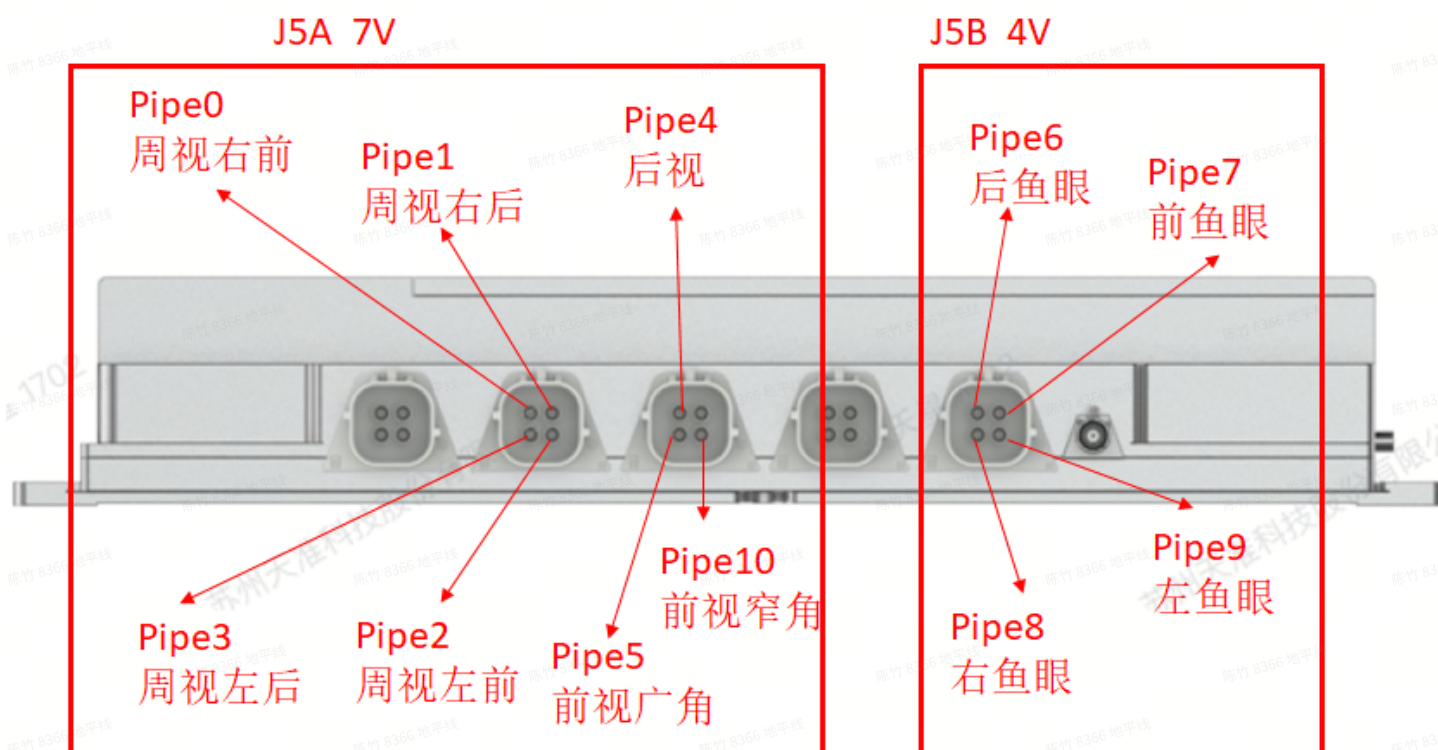


2.0 域控7V相机接法

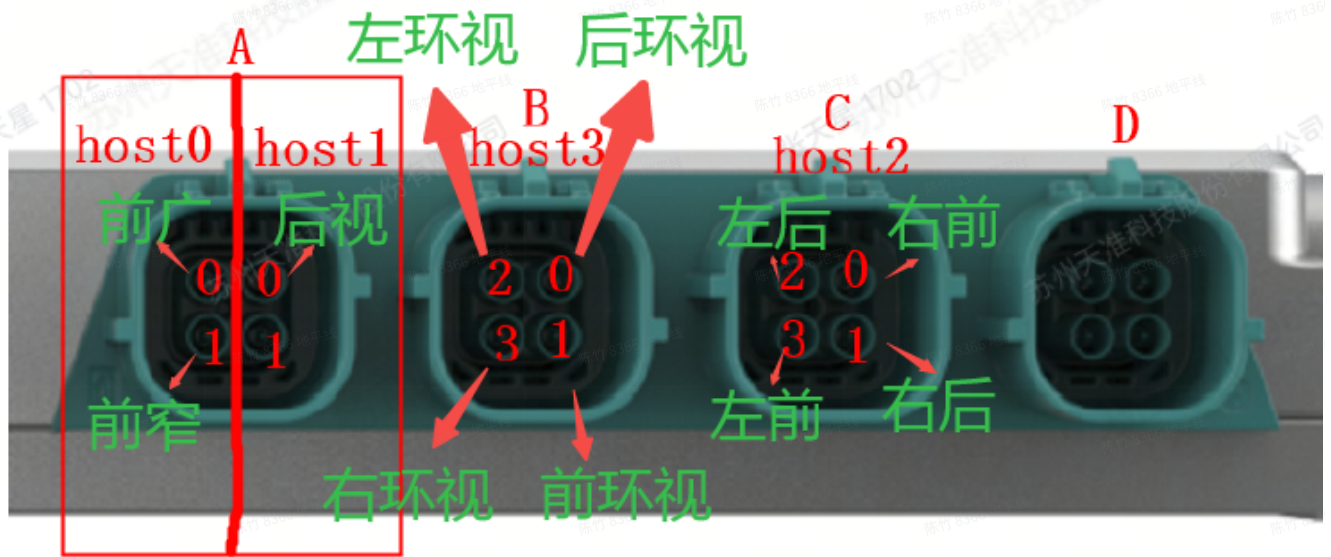


相较之前接法只改变了前广、前窄、后视接法，周视、环视不用改变

3.0域控7V相机接法



2.0域控11V相机接法



3.0域控11V相机接法

4.7.3 相机

4.8 matrix-client使用相关问题

4.8.1 matrix-client怎么使用，有没有使用文档

A: 可通过此链接下载，若无法访问请联系FAE

https://alidocs.dingtalk.com/i/nodes/MNDoBb60VL9rovnvFQE5By2gJlemrZQ3?utm_scene=team_space

4.8.2 matrix-client查看感知图像，只有一个会有检测框？

A: matrix-client中只有前视广角会有检测框，其他的相机图像是没有检测框的。

4.9 感知结果获取

4.9.1 Q: Sample code解出来的障碍物位置是 (0,0,0)

A: 查看录制的pack是否有障碍物的位置信息的，可以通过融合端口输出的是position去获取障碍物的位置信息。

4.9.2 Q: 与自车平行的障碍物车，在走近的过程中，yaw角不准

A: 此问题跟地平线感知有关，yaw值动态目标物可以根据相邻帧之间的位置变换计算yaw值，而静态目标物只能通过车辆的车头车尾识别，而这种识别是存在一定的误识别。

4.9.3 Q: 静止的障碍物会识别出速度？

A: 静止目标物的速度，因为自车是一直在运动，会存在计算出目标物低速的状况，此问题无法解决。

5、中间件部署

5.1 下载问题

5.1.1 Q:到哪里下载TROS源码包

A:通过ftp下载，可以使用filezilla工具下载，此工具在Windows和Ubuntu系统中皆可用，ftp地址如下：

地址	221.224.80.18:8021
账号	GEAC_GUEST
密码	geac_guest
路径	/new/J5/TROS/vx.x.x/

5.1.2 Q:到哪里下载Docker部署包

A:通过ftp下载，ftp地址参考5.1.1 Q，路径更改为/new/J5/TROS/TROS_Docker_Stable.tgz

5.1.3 Q:哪里可以获取到TROS的相关文档

A:使用ftp下载好TROS源码包后，相关的介绍文档就在其解压后的文件夹中，目录大致为TROS-doc\TROS.A用户使用指南\TROS.A用户使用指南，其中html文件为官方介绍文档，其中包含了TROS环境的搭建、各模块的介绍以及相关工具的使用等等，内容紧实可用，请在认真研读文档后再使用TROS的相关功能。

5.2 编译问题

5.2.1 Q:编译TROS代码为什么报错

A:拿到TROS代码后，需要搭建其编译环境，有三种方式可供选择：windows、linux和docker，这边建议使用docker环境进行编译（我方皆采用docker环境编译），因为操作简单方便，且不会影响你电脑中的其他代码或工具的使用。其编译环境的搭建具体参考TROS文档中安装指南->搭建Docker开发环境

5.2.2 Q:怎么搭建Docker环境

A:搭建docker环境步骤参考[TROS文档](#)，此文档由地平线提供更新，建议参考，docker相关命令和使用也可查阅资料

5.2.3 Q:怎么在windows下进行编译

A:想在windows下编译TROS源码的话，需要具备编译环境的条件，具体参考[TROS文档](#)中环境要求构建

5.2.4 Q:怎么在linux环境下编译

A:想在linux环境下编译TROS源码的话，需要具备编译环境的条件，具体参考[TROS文档](#)中环境要求构建

5.2.5 Q:TROS源码和Docker环境是什么关系

A:Docker是一个容器，里面部署好了TROS编译所需的工具链、库文件等，在部署好Docker后只需要将TROS所在的文件夹目录映射到Docker内，在Docker内编译生成的文件等都会出现在对应的本地目录下

5.2.6 Q:TROS源码如何编译

A:编译命令参考[TROS文档](#)中模块功能介绍中communication->安装编译介绍

5.2.7 Q:为什么编译不通过

A:先确认使用的编译命令正确之后，在确认自己需要编译的系统的sdk是否有推到环境中，例如，想要编译在J5下运行的通信程序，就要先将J5的sdk通过platform_sdk中的readme的方法将j5 sdk环境推送到编译环境中

5.3 使用问题

5.3.1 Q:如何使用生成的文件

A:在您编译成功后生成的文件，首先取决于你编译的是哪个系统下的源码，拿J5举例，编译生成的文件就将其和其依赖的配置文件、库文件一起打包放到J5域控中解压，直接执行即可

5.3.2 Q:怎么创建一个新的工程

```
1 mkdir hello && cd hello
2 # 建一个bole工程
3 conan new hello/1.0 -m bole
4 mkdir build && cd build
5 # 安装linux依赖 (可选)
6 conan install .. -pr linux
7 # 安装conan J5依赖 (可选)
8 conan install .. -pr aarch64_gcc9.3 -o *:soc=j5 -o *:hbmam=True
```

```
9 # 上述依赖可以选择系统，但是在编译之前必须有一项系统依赖
10 # 编译
11 conan build ..
```

5.3.2.1 Q:创建一个新工程编译报错

A:可能是自动生成的文件中的依赖库的版本不对，使用conan命令查看当前库文件的版本，修改同级目录中conanfile.py文件重新编译即可，例如

```
default_options = {}

# requires = [("dataflow/[~1.0]@bole/stable")] 原生成
requires = ["dataflow/1.3.4@bole/stable",  hello里存在
            ("hobot-bole-protocol/1.3.2@bole/stable", "private")]
generators = "cmake_find_package"
```

此问题报错如下

```
1 root@9fde3732b191:/work/TROS.A-preview/hello_world/test/build# conan install
.. -pr linux
2 Configuration:
3 [settings]
4 arch=x86_64
5 build_type=Release
6 compiler=gcc
7 compiler.libcxx=libstdc++11
8 compiler.version=5.4
9 os=Linux
10 [options]
11 [build_requires]
12 [env]
13
14 ERROR: Version range '~1.0' from requirement 'dataflow/[~1.0]@bole/stable'
required by 'conanfile.py (test/1.0)' could not be resolved in local cache
```

5.3.2.2 Q:怎么查看所有库文件的版本

```
1 # 使用如下命令查看库文件版本信息
2 conan search
```

```
root@f3ec347968da:/work/platform_sdk/conan_sdk/x9u# conan search  
Existing package recipes:
```

```
communication/1.6.4@bole/stable  
dataflow/1.6.3@bole/stable  
datatools/1.6.5@bole/stable  
fmt/9.0.0@thirdparty/stable  
gtest/1.8.0@thirdparty/stable  
hlog/1.6.3@bole/stable  
hobot-bole-protocol/1.4.10@bole/stable  
hthreadpool/0.4@thirdparty/stable  
libjpeg-turbo/1.5.2@thirdparty/stable  
lz4/1.9.4.1@thirdparty/stable  
lzma/5.2.5@thirdparty/stable  
message/1.6.1@bole/stable  
nlohmann_json/3.7.3@thirdparty/stable  
pack_sdk/1.6.3@bole/stable  
protocol/3.19.3@thirdparty/stable  
schedulegroup/1.6.3@bole/stable  
sensors/1.6.2@bole/stable  
serial/1.2.2@thirdparty/stable  
time/1.6.3@bole/stable  
tinyalsa/1.1.1@thirdparty/stable  
unwind/1.6.0@thirdparty/stable  
vehicleio/1.6.2@bole/stable  
zeromq/4.3.4@thirdparty/stable
```

进行图形截图

5.4 通信问题

5.4.1 Q:运行程序跑不起来，直接报错

A:可能是当前的运行环境下没有运行所需的库文件，找到之后放到当前目录下并声明环境变量，也可以统一管理并声明环境变量，报错截图如下

```

ubfolder/bin$ ./publisher 2
participant id hasn't been input, default value will be use
[I][9394][01-29][11:16:50:378][types.cpp:457][publisher][communication] COMMUNICATION_CACHE_DIR not set!,
the files generated by communication in runtime will store in /tmp/communication/ in unix environment, C
:/Windows/Temp/communication/ in windows
[I][9394][01-29][11:16:50:378][types.cpp:470][publisher][communication] WorkOnScheduleGroup not set! Sche
duleGroup threads management will not be using
[I][9394][01-29][11:16:50:378][types.cpp:547][publisher][communication] Communication init start config_f
ile:communication.json ...
[I][9394][01-29][11:16:50:378][global_config.h:154][publisher][communication] DebugOptionInfo:
enable: 0
dumping_config:
    dumping_enable:1
    dumping_path: ./comm_debug.log
    dumping_period:1
topic_list_include:[]
topic_list_exclude:[]
service_list_include:[]
service_list_exclude:[]
hbmем_tracing_config:
    hbmем_tracing_enable: 0
    hbmем_tracing_logging_path:
[I][9394][01-29][11:16:50:378][global_config.cpp:1211][publisher][communication] A total of 3 plug-in lib
raries are loaded:
[I][9394][01-29][11:16:50:378][global_config.cpp:1216][publisher][communication] Protocols id 2, name zmq
_tcp, plugin_file libplugin-zmq
[I][9394][01-29][11:16:50:378][global_config.cpp:1216][publisher][communication] Protocols id 3, name zmq
_epgm, plugin_file libplugin-zmq
[I][9394][01-29][11:16:50:378][global_config.cpp:1216][publisher][communication] Protocols id 4, name zmq
_ipc, plugin_file libplugin-zmq
communication version is vl.7.3
create publisher failed[E][9394][01-29][11:16:50:379][plugin_manager.cpp:795][publisher][communication]
Fail to open plugin file libplugin-zmq.so, reason libplugin-zmq.so: cannot open shared object file: No su
ch file or directory
[E][9394][01-29][11:16:50:379][types.cpp:1305][publisher][communication] communication is uninitialized!
Please call the hobot::communication::Init() first
[I][9394][01-29][11:16:50:379][types.cpp:626][publisher][communication] Communication deinit start ...
[I][9394][01-29][11:16:50:379][types.cpp:641][publisher][communication] Communication deinit finish

```

如上图报错信息为，不能打开文件libplugin-zmq.so这个库，预测原因有三种，运行环境和编译生成的系统不是相同的、库文件没有在环境变量当中、库文件不存在，我的编译为linux版本，运行在服务器本地，通过find命令找到库文件就在当前目录，所以没有设置好环境变量，通过下面命令设置当前目录加入环境变量当中

```
1 export LD_LIBRARY_PATH=./:$LD_LIBRARY_PATH
```

声明好环境变量之后，在运行程序时使其可以成功找到所需的库文件，运行成功

```

fae@fae:/data/fae/TZTEK_TROS_R3.0_v1.3.0/samples/component_samples/communication_ex/example/build/build_s
ubfolder/bin$ ./publisher 2
participant id hasn't been input, default value will be use
[I][10036][01-29][11:24:48:710][types.cpp:457][publisher][communication] COMMUNICATION_CACHE_DIR not set!
, the files generated by communication in runtime will store in /tmp/communication/ in unix environment,
C:/Windows/Temp/communication/ in windows
[I][10036][01-29][11:24:48:710][types.cpp:470][publisher][communication] WorkOnScheduleGroup not set! Sch
eduleGroup threads management will not be using
[I][10036][01-29][11:24:48:710][types.cpp:547][publisher][communication] Communication init start config_
file:communication.json ...
[I][10036][01-29][11:24:48:710][global_config.h:154][publisher][communication] DebugOptionInfo:
enable: 0
dumping_config:
    dumping_enable:1
    dumping_path: ./comm_debug.log
    dumping_period:1
topic_list_include:[]
topic_list_exclude:[]
service_list_include:[]
service_list_exclude:[]
hbmeme_tracing_config:
    hbmeme_tracing_enable: 0
    hbmeme_tracing_logging_path:
[I][10036][01-29][11:24:48:710][global_config.cpp:1211][publisher][communication] A total of 3 plug-in li
braries are loaded:
[I][10036][01-29][11:24:48:710][global_config.cpp:1216][publisher][communication] Protocols id 2, name zm
q_tcp, plugin_file libplugin-zmq
[I][10036][01-29][11:24:48:710][global_config.cpp:1216][publisher][communication] Protocols id 3, name zm
q_epgm, plugin_file libplugin-zmq
[I][10036][01-29][11:24:48:710][global_config.cpp:1216][publisher][communication] Protocols id 4, name zm
q_ipc, plugin_file libplugin-zmq
[I][10036][01-29][11:24:48:710][discovery_manager.cpp:70][publisher][communication] DiscoveryManager init
begin ...
[I][10036][01-29][11:24:48:711][discovery_manager.cpp:99][publisher][communication] DiscoveryManager::ini
t eth_ip 192.168.37.73 mask_ip 255.255.255.0
[I][10036][01-29][11:24:48:711][discovery_manager.cpp:169][publisher][communication] DiscoveryManager::in
it group_attr ip 224.0.0.110 port 12435
[I][10036][01-29][11:24:48:711][discovery_manager.cpp:201][publisher][communication] DiscoveryManager::in
it SingleSocketAttr ip 192.168.37.73 port 17411
communication version is v1.7.3
[I][10036][01-29][11:24:48:731][discovery_manager.cpp:222][publisher][communication] DiscoveryManager ini
t finish
[I][10036][01-29][11:24:48:731][types.cpp:608][publisher][communication] Communication init OK
[I][10036][01-29][11:24:48:731][publisher.h:99][publisher][communication] Create a new publisher, topic=t
opic, protocol=2, domain_id=0
[I][10036][01-29][11:24:48:932][publisher.h:215][publisher][communication] Publisher initialized successf
ully!, topic:topic
[I][10036][01-29][11:24:49:948][publisher_example.cpp:301][publisher][communication] publish all process
finished!
[I][10036][01-29][11:24:49:949][types.cpp:626][publisher][communication] Communication deinit start ...
[I][10036][01-29][11:24:49:972][types.cpp:641][publisher][communication] Communication deinit finish

```

5.4.2 Q:运行pub和sub进行通信，无法通信

A:

此问题一般应该出现在不同的运行位置，通常我们要在使用时要先进行IP的配置，在同级目录的communication.json文件中"participants"中“link_info”的信息，如下图

```

{
    "id": 2,
    "protocol": "zmq_tcp",
    "link_info": "127.0.0.1:6788",
    "link_type": ""
},

```

默认的IP为127.0.0.1是本地IP的意思，端口为6788，所以在本地测试时通信是可以互通的，在使用其他系统或者机器运行时需要根据机器的实际IP进行配置，端口号只要不占用正在使用端口即可。将pub

端的此字段IP改为0.0.0.0，sub端的IP改为pub所在机器的实际IP，端口号一致，再运行即可。

5.4.3 Q:运行两个sub报错

A:在进行一发多收，也就是运行一个pub，运行两个及两个以上的sub测试时，sub端会报错并且都无法运行，原因是在代码中的deinit函数限制了sub的初始化，在代码中将此部分注释掉或者修改后重新编译即可

5.4.4 Q:进行多发多收测试失败

A:在使用通信程序做多发多收测试时，如果出现只有一组通信成功或者都不能通信的情况，是因为在代码中配置的文件是唯一的，所以多个pub和sub在运行前读取的配置信息冲突，解决的办法是在代码读取的配置文件部分，修改对应的配置文件编号名称，使每一组通信对应一个communication.json的配置文件

5.4.5 Q:是否支持不同系统间的通信

A:是否支持不同系统间的通信，比如编译J5版本pub和X9U版本的sub，想进行通信，是可以的，只要配置好网络，对应的通信都是可以实现的

5.4.5 Q:可以使用通信传输图片么

A:是可以修改代码传输图片的，创建一个buffer后，将图片读到里面，然后传输，接收端以相对应的发送方式进行接收

6、BSP问题

6.1 使用问题

6.1.1 Q:接外网后手动设置IP后域名解析不了(如何设置静态IP并访问外网)

A:执行命令:

```
1 cp /app/bin/ip /sbin/ip //有版本中/sbin/ip工具不正常
2 /app/bin/ip addr flush eth0 //配置eth0，清除eth0信息
3 dhclient -4 eth0 //动态获取eth0 ip
4 ping baidu.com //正常情况下能ping通
5 ifconfig eth0 XXX.XXX.XXX.XXX//可以手动更改IP地址，如192.168.37.33改成
  192.168.37.44，此时ping baidu.com ping不通
6 route add default gw 192.168.37.254 //网关以实际为准，配置网关后可ping baidu.com
```

7. 域控使用

7.1 网络访问

7.1.1 Q:无法通过网络访问J5

A: (1) 检查接线是否正确;

(2) 检查车载盒的配置是否与所连接的网口配置一致;

(3) pc的IP是否与J5的ip设置的为同一网段, 可以通过串口进入J5, ifconfig查看J5的ip;

(4) 可能是交换机坏了, 可以尝试通过串口进入J5, 看是否能够ping通其它的soc.

(5) 车载盒是否通过网线与电脑直连, 如果使用拓展坞, 可能会导致无法访问J5, 可以换台带网口的pc, 通过网线直连尝试。

7.2 域控温度状态

7.2.1 Q:2.0 TADC液冷域控上面有温度传感器吗? 和对应的温度状态信息和报警吗?

A: 域控上有温度传感器, 没有报警的功能;

J5芯片温度查询指令: hrut_somstatus

X9 Cpu温度查询: cat /sys/devices/virtual/thermal/thermal_zone0/temp

7.3 授时相关

7.3.1 Q: 2.0和3.0域控都如何通过外部惯导授时?

A: 对于2.0域控, 默认是支持内部gps授时的, 如果需要切换为外部惯导授时, 可以通过以下两种方式:

(1) 通过can盒给mcu发送指令, 切换为外部惯导授时模式, 此方法在设备重启之后会恢复为内部授时模式, 不建议使用。

(2) 重新烧录支持外部授时的mcu固件版本, 0x15这个版本就是默认支持外部授时的。



ospi_R2.0V2.0_0x15.pac
1012.08KB



对于3.0域控, mcu固件默认就是支持外部授时模式的, 不需要切换。

7.3.2 Q: 域控接入惯导的pps之后收不到pps信号?

A:

- (1) 确认mcu固件是否是支持外部授时的，如果不支持外部授时，则需要烧录支持的mcu固件版本；
- (2) 确认惯导是否有pps信号输出，可以通过示波器去测量波形；
- (3) 接线问题，让硬件同事确认接线是否正确，具体的接线方式请参考对应的用户手册；
- (4) 确认惯导的型号，如果使用2.0域控，接入的惯导型号是导远570D的，则需要将设备寄回，焊接一个下拉电阻；所以如果确定客户使用的导远570D的惯导，域控型号2.0的话，最好在发货前焊接一个下拉电阻，避免返工。