

YGW-R1 产品 使用手册

目录

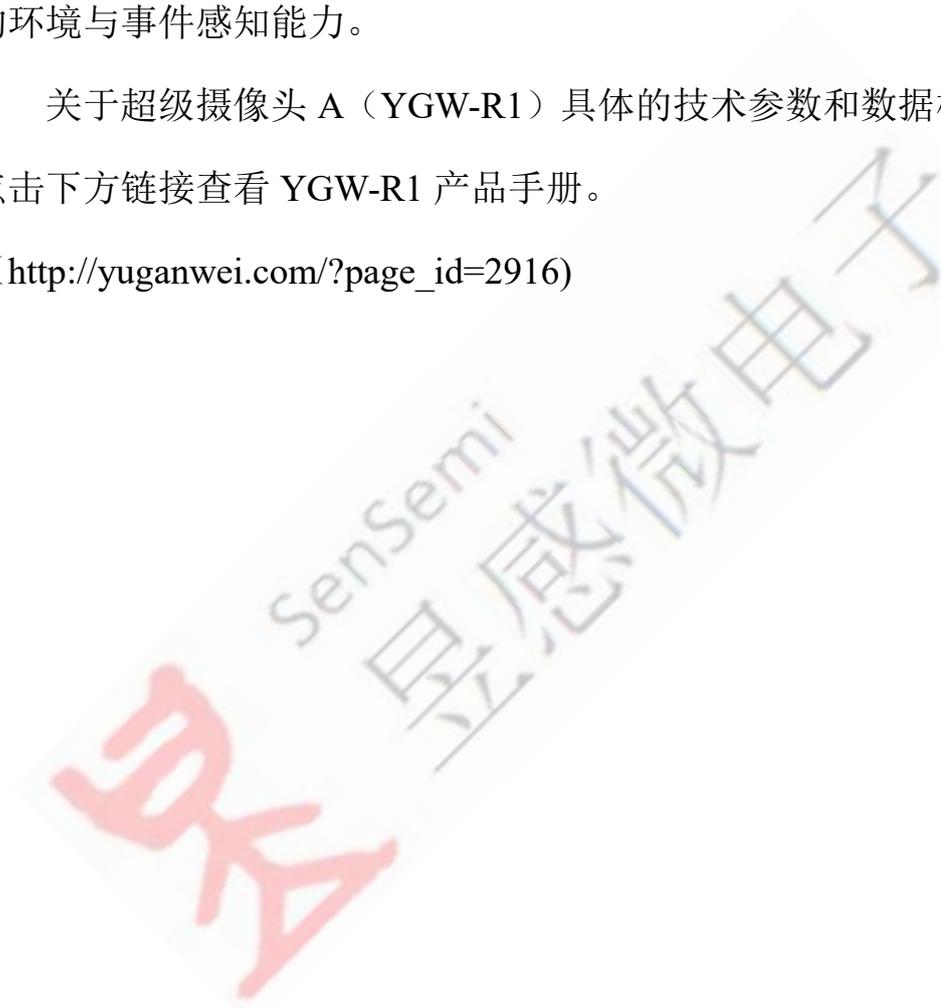
一、产品说明	3
二、技术参数	4
1. 红外摄像头设备规格	4
2. 可见光摄像设备规格	5
3. 毫米波雷达规格	5
4.标准 fakra 接口	6
5.TE 接口	7
三、设备连接及使用方法	9
3.1 设备上电连接步骤	9
3.2 设备下电拆分步骤	11
3.3 数据采集设备连接	11
四、数据采集和解析	13
4.1 数据采集设备环境	13
4.2 数据采集设备初始化	14
4.3 融合帧数据显示	18
4.4 融合帧数据采集	19
4.5 融合帧数据解析	19
4.5 融合帧保存成.avi 视频文件	20
五、YGW-R1 使用流程参考视频	21

一、产品简介

昱感微超级摄像头 A (YGW-R1) 集成了毫米波雷达, 可见光摄像头, 红外摄像头, 多传感器融合后生成时空对齐的多维像素数据, 通过 GMSL 接口发出。本品为客户提供更加直接、高效、和可扩展的环境与事件感知能力。

关于超级摄像头 A (YGW-R1) 具体的技术参数和数据格式, 请点击下方链接查看 YGW-R1 产品手册。

(http://yuganwei.com/?page_id=2916)



二、技术参数

产品前面板如图 2-1 所示，红色为标准传感器序号：

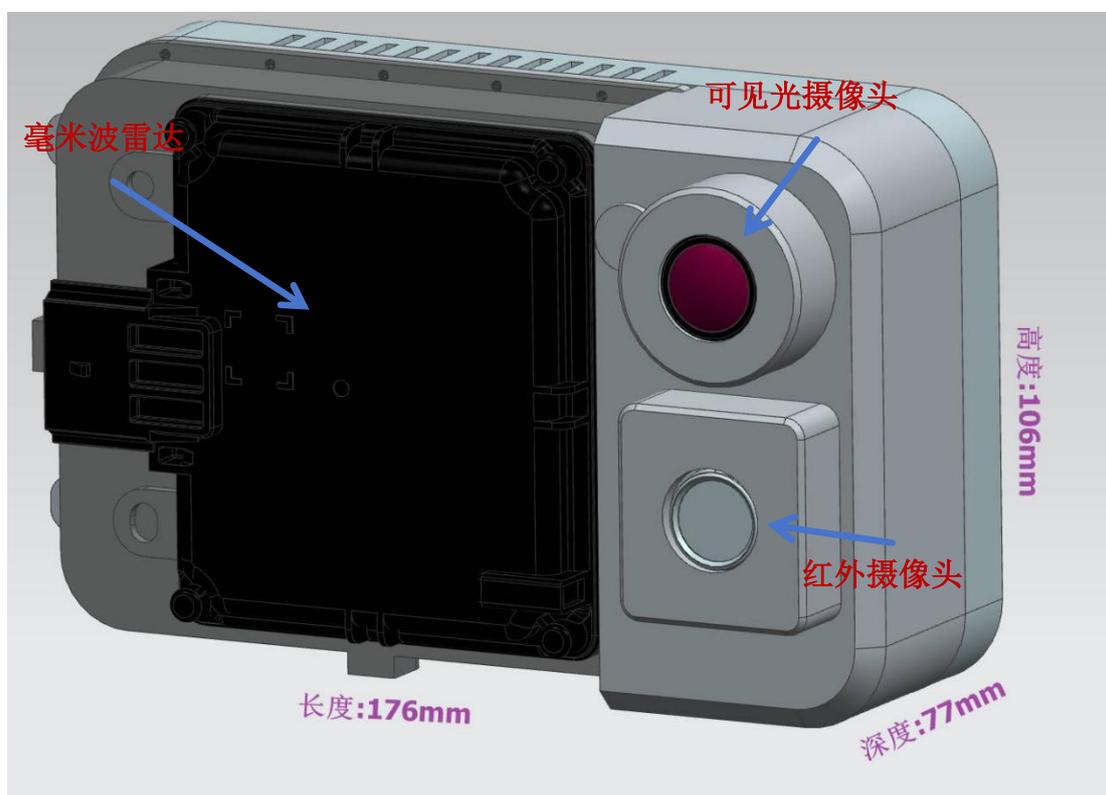


图 2-1 产品前面板

1. 红外摄像头设备规格

表 2-1 红外摄像头设备规格

分辨率	640*512
帧率	30fps
FOV	$\geq 46^{\circ} \times 37^{\circ}$
焦距	9.1mm/F1.0

2. 可见光摄像设备规格

表 2-2 可见光摄像头设备规格

分辨率	3840*2160
帧率	30fps

3. 毫米波雷达规格

表 2-3 毫米波雷达规格

简介	77GHz 成像雷达传感器, 6 发 8 收, 48 通道先进级联雷达天线设计
测距	0.25m ~285m
距离精度	$\pm 0.4\text{m}$ (>150m) $\pm 0.2\text{m}$ (>75m and <150m) $\pm 0.1\text{m}$ (<75m)
速度范围	-115~55 (m/s)
速度分辨率	0.13 m/s
速度精度	$\pm 0.06\text{m/s}$
水平角度范围	$\pm 12\text{deg}$ (230m) $\pm 60\text{deg}$ (40m)
俯仰角度范围	$\pm 10\text{deg}$ (200m)
角度精度	Az: $\pm 0.2\text{deg}$ @ $\pm 12\text{deg}$ $\pm 0.8\text{deg}$ @ $\pm 60\text{deg}$

	$\pm 0.4\text{deg}@ \pm 45\text{deg}$ E1: $\pm 0.5\text{deg}@ \pm 10\text{deg}$
刷新周期	55ms

产品后面板如图 2-2 所示，红色为标注对外接口序号：

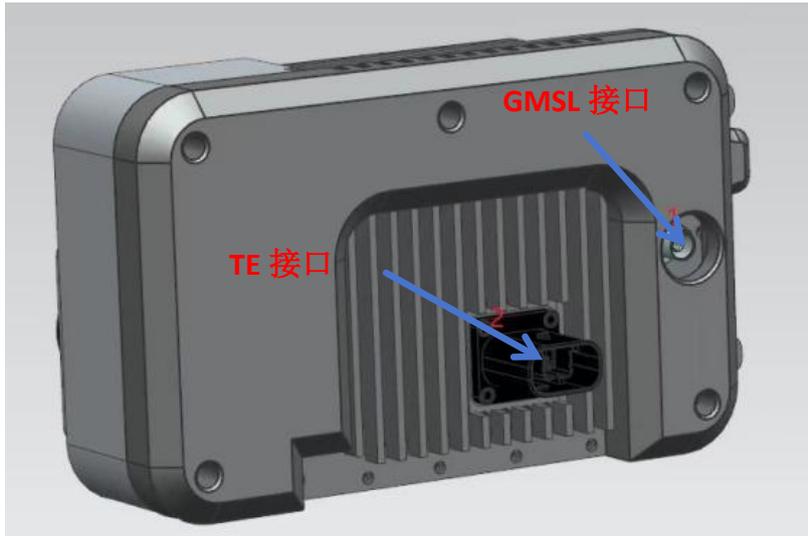


图 2-2 产品后面板

4.标准 fakra 接口

支持 GMSL2 协议。产品中会提供一根 3m 双母口的 Fakra 线缆。



图 2-3 标准 fakra 接口

5. TE 接口

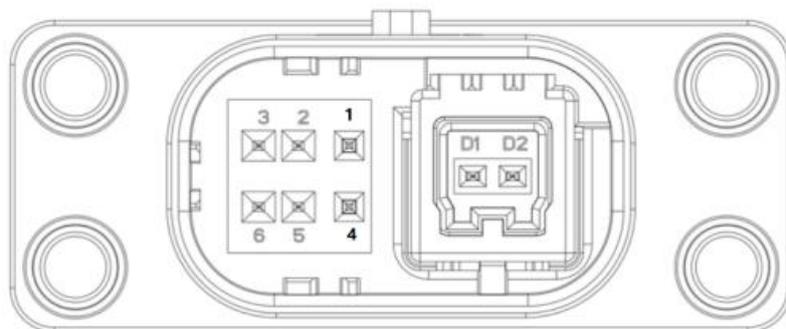


图 2-4 TE 接口

表 2-4 TE 接口信号

编号	信号	说明
1	VCC	12V 电源正
2	TTL-Tx	3.3V 电平
3	预留	
4	GND	12V 电源负
5	TTL-Rx	3.3V 电平
6	预留	
D1	MDI-N	车载以太网
D2	MDI-P	车载以太网

车载以太网接口，目前主要是以下 3 个功能，目前不支持融合产品的数据输出。

- 授时；
- 固件升级；
- 调试（调试目前暂不对客户开发）；

TE 接口线缆如图 2-5 所示，产品中会自带一根此线缆。



图 2-5 TE 接口线缆

整机规格：

- 尺寸（长高深）：176mm * 106mm * 77mm
- 功耗：26W
- 供电要求：12V 3A
- 工作温度范围：-40℃~85℃
- 对时方式：本品支持 ptp/gptp 对时，域控制器运行 ptp/gptp 授时服务通过网络给本品授时

三、设备连接及使用方法

3.1 设备上电连接步骤

- 1) 使用配套 TE 线束连接产品的后面板“TE 接口 2”，GSML 采集线连接产品后面板的“GMSL 接口 1”；
- 2) 电源转接盒正反面如图 3-1 所示，主要使用到 DC12V、串口和 TE 三个接口；将 TE 线束接到电源转换盒上（只连接黑色的 MX3.0mm-4P 的端子即可，绿色的 KF2EDG3.81mm-2P 端子可不连接）；

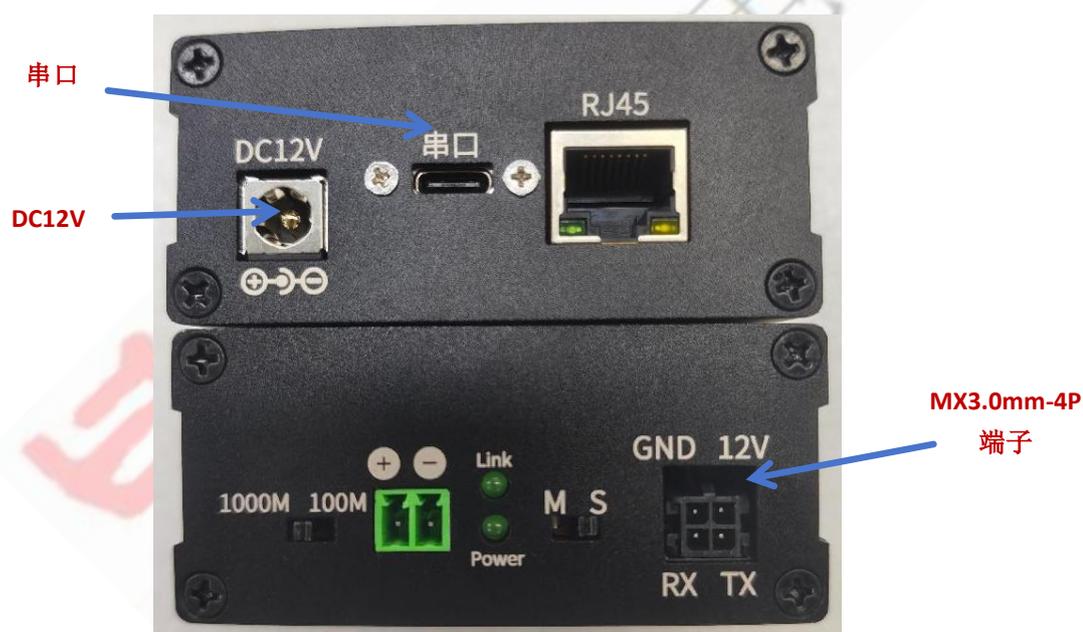


图 3-1 电源转接盒（正反面）

- 3) 将电源转换盒的串口（Type-C 接口）连接到电脑的 USB 接口，波特率设置 115200，给电源转换盒通电，电源要求电压为 12V、电流 $\geq 3A$ 的电源(本产品标配 12V 5A 的电源适配器)；如下图 3-2 和图 3-3 分别为 UART 串口连接设置以及设备完整连接图。

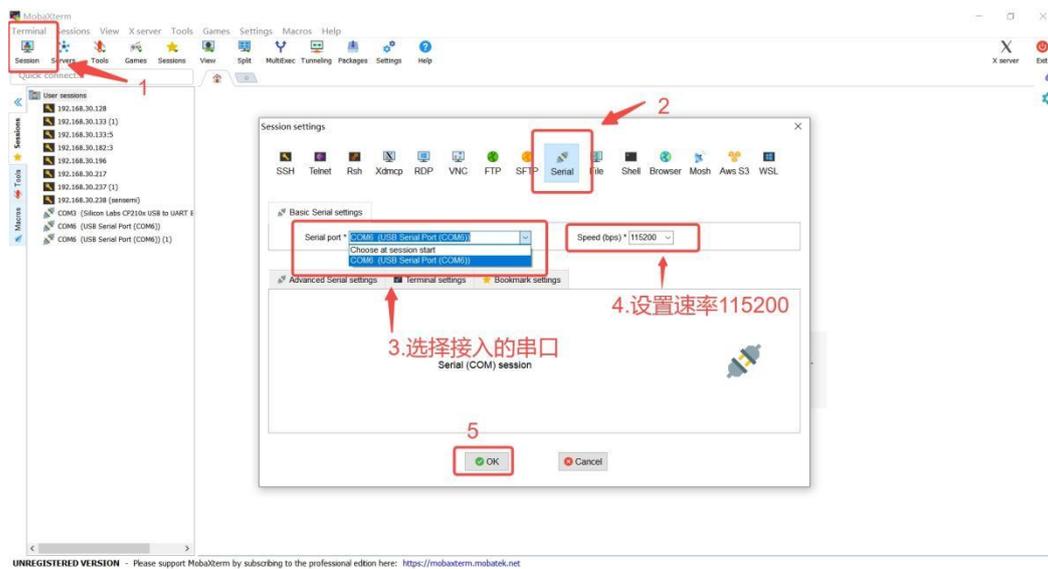


图 3-2 UART 串口连接设置



图 3-3 设备完整连接效果

- 4) 上电后串口会持续打印系统初始化信息，约一分钟左右此时串口打印“TX START”字样并不再打印，如图 3-4 所示，表明产品初始化完成，开始从 GSML2 口（接口 1）输出融合后的多维像素数据。

```
tx vdma out:4:0x61471000
tx vdma out:5:0x62dee000
tx vdma out:6:0x6476b000
tx vdma out:7:0x660e8000
tx vdma out:8:0x67a65000
vdma device id:0
pVdmaInfo:0x40ccb280 tx:0x40ccb280
vsize(hight):2320, hsize(width):7680, stride:7680
==vdmaSetup fusion 0xa0040000 finish==

[ 26.570529] autorunapp[562]: End Auto Run p002-3eg-a53...
==gdmapi file address:0x3ED1EB04==
intc-pl2ps-warp-done set before get priority = 0xA0, trigger = 0x1
intc-pl2ps-warp-done set after get priority = 0xA0, trigger = 0x3
trigger: internal
p_ddr_res_warpOut->nums_buf:9
set-before:
set-after:11111111
warp out buffer now is all released
-----TX START-----
```

图 3-4 上电自启动完成

3.2 设备下电拆分步骤

- 1) 先将 GSML 采集线从产品后面板的“GMSL 接口 1”移除；
 - 2) 再将电源转换盒的 12V 电源断开,后拔除电源转换盒上的串口线。
- 完成上述操作后表示将 YGW-R1 产品完全下电，才可以对产品进行后续操作。（以免出现静电烧毁等情况）

3.3 数据采集设备连接

数据采集设备准备：

- PC 机（整机的 CPU 主板等硬件配置最好为市场近三年内产品，否则可能出现由于性能较低导致的丢帧等问题）；
- PCIE 视频采集卡（这里已艾利光 PCIE 视频采集卡为例进行描述说明），如下图 3-5 所示；



图 3-5 艾利光 PCIe 视频采集卡

PC 机断电情况下将艾利光 PCIe 视频采集卡插入到主板对应的 PCIe 插槽中并用螺丝进行固定，如图 3-6 所示，再将 GMSL 采集线连接 YGW-R1 模组后面板的 GMSL 接口 1 和 PCIe 视频采集卡的任一 CH 口，即可完成数据采集设备的连接。

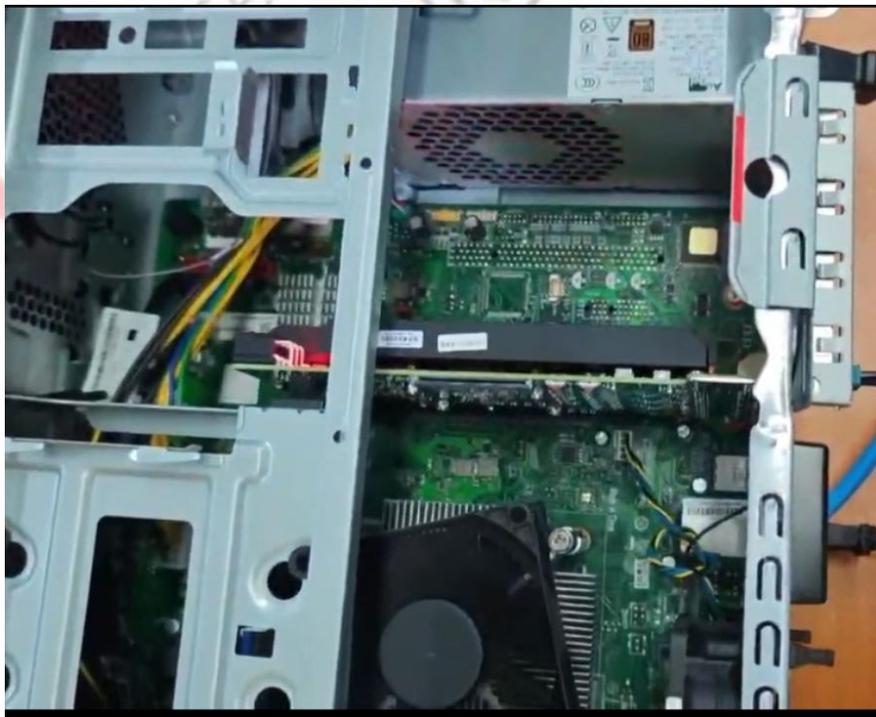
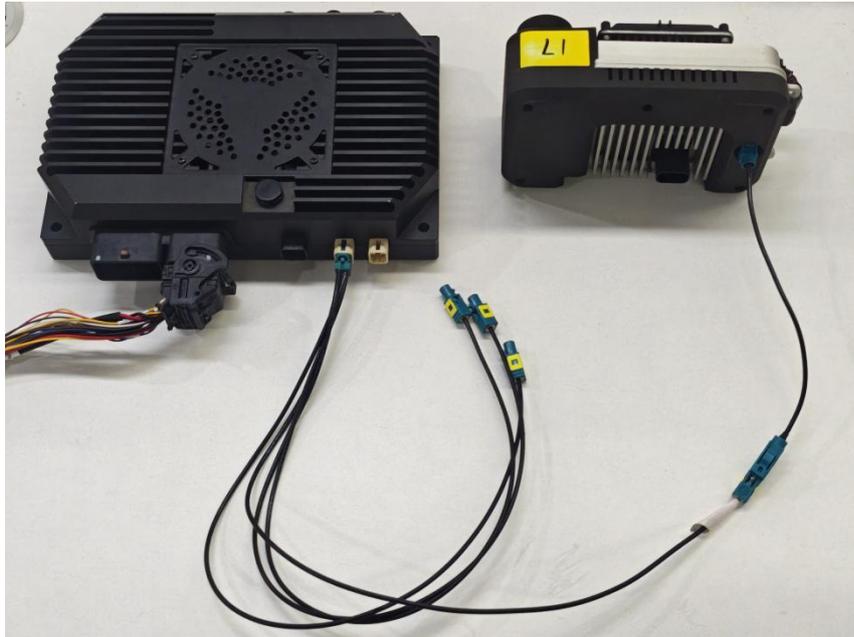


图 3-6 PC 机固定 PCIe 视频采集卡

四、数据采集和解析

本产品可直接通过标准的 serdes 接口连接域控（如下图），也可通过 PCIE 转接卡连接电脑。



以下章节主要讲述使用 PCIE 采集卡连接电脑，在进行数据采集和解析介绍时涉及到的 SDK Demo 文件、Python 脚本以及操作文档都会提供。

4.1 数据采集设备环境

- 1) PC 环境系统为 ubuntu18.04，附带对应编译器及版本要求为 GCC 7.5，CMake 3.5，Python3 等，安装时可以通过 Linux 源或者提前下载安装包进行安装。
- 2) PCIE 驱动、显示和 SDK 接口环境安装，完成《视频采集服务器环境搭建与使用操作简易操作手册》中的 PCIE 采集卡驱动安装、

pcie_sdk_demo 编译和安装，环境安装过程中可能会遇到缺少库的情况，根据提示补全缺少的库即可，其中 GStreamer 和 V4l2loopback 库环境需要使用本公司提供的文件进行安装。

4.2 数据采集设备初始化

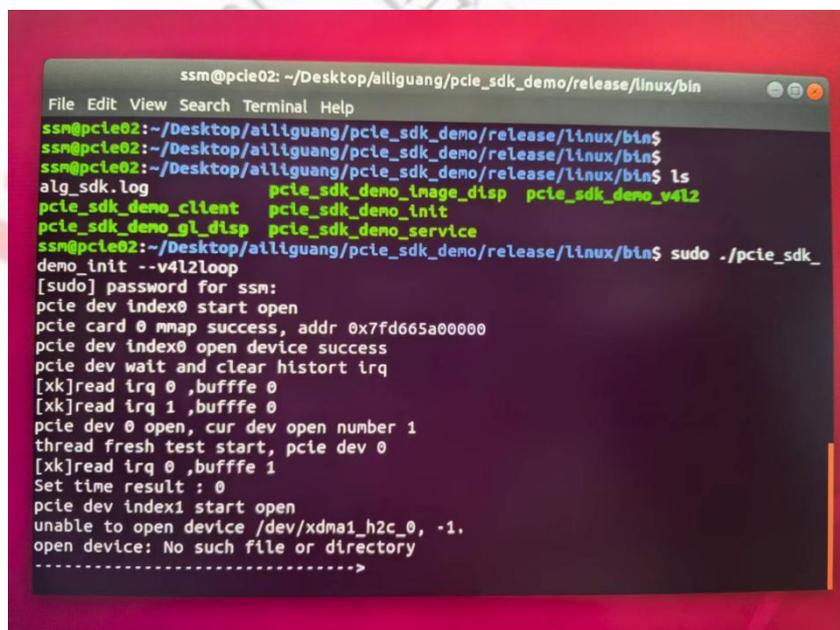
数据采集设备初始化会使用到 pcie_sdk_demo 文件目录对 PCIE 视频采集卡进行配置，以艾利光 PCIE 视频采集卡为例，具体操作步骤如下：

1) 进入 pcie_sdk_demo 目录，

```
cd xxx/pcie_demo_sdk/release/linux/bin
```

执行 `sudo ./pcie_sdk_demo_init --v4l2loop`

完成初始化，并保持该窗口默认，一段时间后可以看到窗口完成初始化的打印信息如图 4-1 所示。



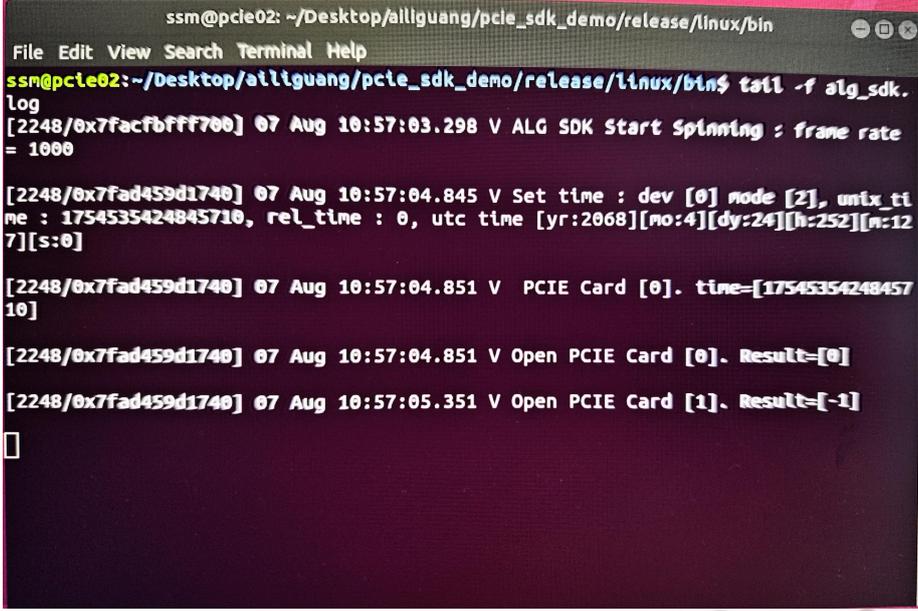
```
ssm@pcie02: ~/Desktop/alliguang/pcie_sdk_demo/release/linux/bin
File Edit View Search Terminal Help
ssm@pcie02:~/Desktop/alliguang/pcie_sdk_demo/release/linux/bin$
ssm@pcie02:~/Desktop/alliguang/pcie_sdk_demo/release/linux/bin$
ssm@pcie02:~/Desktop/alliguang/pcie_sdk_demo/release/linux/bin$ ls
alg_sdk.log          pcie_sdk_demo_image_disp  pcie_sdk_demo_v4l2
pcie_sdk_demo_client  pcie_sdk_demo_init
pcie_sdk_demo_gl_disp  pcie_sdk_demo_service
ssm@pcie02:~/Desktop/alliguang/pcie_sdk_demo/release/linux/bin$ sudo ./pcie_sdk_
demo_init --v4l2loop
[sudo] password for ssm:
pcie dev index0 start open
pcie card 0 mmap success, addr 0x7fd665a00000
pcie dev index0 open device success
pcie dev wait and clear histort irq
[xk]read irq 0 ,bufffe 0
[xk]read irq 1 ,bufffe 0
pcie dev 0 open, cur dev open number 1
thread fresh test start, pcie dev 0
[xk]read irq 0 ,bufffe 1
Set time result : 0
pcie dev index1 start open
unable to open device /dev/xdma1_h2c_0, -1.
open device: No such file or directory
----->
```

图 4-1 PCIE 视频采集卡初始化完成

2) 打开一个同目录的新窗口，用于显示 SDK 接口日志

执行 `tail -f alg_sdk.log`，可以看到每次操作 PCIE 视频采集卡的日志信

息如图 4-2 所示。



```
ssm@pcie02: ~/Desktop/alliguang/pcie_sdk_demo/release/linux/bin
File Edit View Search Terminal Help
ssm@pcie02:~/Desktop/alliguang/pcie_sdk_demo/release/linux/bin$ tail -f alg_sdk.
log
[2248/0x7facfbff700] 07 Aug 10:57:03.298 V ALG SDK Start Spinning ; frame rate
= 1000

[2248/0x7fad459d1740] 07 Aug 10:57:04.845 V Set time : dev [0] node [2], unix_ti
me : 1754535424845710, rel_time : 0, utc time [yr:2068][mo:4][dy:24][h:252][m:12
7][s:0]

[2248/0x7fad459d1740] 07 Aug 10:57:04.851 V PCIe Card [0]. time=[17545354248457
10]

[2248/0x7fad459d1740] 07 Aug 10:57:04.851 V Open PCIe Card [0]. Result=[0]

[2248/0x7fad459d1740] 07 Aug 10:57:05.351 V Open PCIe Card [1]. Result=[-1]

█
```

图 4-2 PCIe 视频采集卡操作日志

3) 完成初始化和日志显示后，需要对 PCIe 视频采集卡的 GMSL 芯片以及通道进行配置，提供了 C 和 Python 两种方式，这里推荐使用 python 脚本对 PCIe 采集卡进行配置（C 需要在修改代码后进行额外编译操作）；

打开一个新窗口，cd xxx/pcie_demo_sdk/src/python，

可以看到如图 4-3 中的如下 Python 脚本文件；这里主要使用 set_sensor_from_json.py 和 tream_on_by_channel.py 两个文件分别用于配置 GMSL 芯片和激活数据通道；

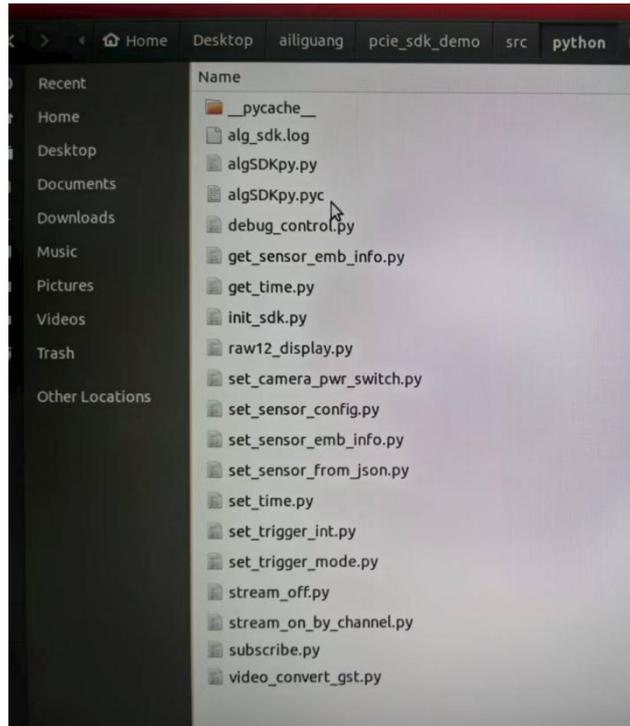


图 4-3 Python 脚本

目录下执行 `python3 set_sensor_from_json.py --json_file="/home/xxx/pcie_demo_sdk/config/alg_ox08b_1_ch.json" --channel= 0` (channel 0~7 以实际连接为准)，执行后可以在在初始化和 SDK 日志窗口看到相关打印信息如图 4-4 和 4-5 所示，图 4-4 表示 GMSL 芯片 IIC 配置信息，图 4-5 表示 Python 脚本操作 PCIE 视频采集卡日志；（alg_ox08b_1_ch.json 中默认调用的 IIC 配置是 4k 分辨率的内容，需要将 IIC 配置的 TXT 文档替换为 YGW-R1 输出的 3840*2320 分辨率的配置，该文件在 alg_viewer-0.2.16.1 中提供过）；

```
ssm@pcie02: ~/Desktop/ailiguang/pcie_sdk_demo/release/linux/bin
File Edit View Search Terminal Help
open device: No such file or directory
----->
Cam number 1
[xk] into down cam cfg deseres_id 0
[xk]read irq 0 ,bufffe 1
i2c index 0 addr 0 reg 0 data ff fmt 808
i2c index 1 addr 80 reg 2d6 data 80 fmt 1608
i2c index 2 addr 80 reg 2c7 data 80 fmt 1608
i2c index 3 addr 80 reg 2ca data 80 fmt 1608
i2c index 4 addr 80 reg 2be data 80 fmt 1608
i2c index 5 addr 0 reg 0 data f fmt 808
i2c index 6 addr 90 reg 330 data 4 fmt 1608
i2c index 7 addr 90 reg 2 data f3 fmt 1608
i2c index 8 addr 90 reg 51 data 2 fmt 1608
i2c index 9 addr 90 reg 52 data 1 fmt 1608
i2c index 10 addr 90 reg 44a data d0 fmt 1608
i2c index 11 addr 90 reg 320 data 2c fmt 1608
i2c index 12 addr 90 reg 2b0 data 83 fmt 1608
i2c index 13 addr 90 reg 2b1 data 0 fmt 1608
i2c index 14 addr 0 reg 0 data ff fmt 808
i2c index 15 addr 80 reg 383 data 0 fmt 1608
i2c index 16 addr 80 reg 318 data 5e fmt 1608
i2c index 17 addr 80 reg 570 data c fmt 1608
i2c index 18 addr 80 reg 2c7 data 90 fmt 1608
```

图 4-4 初始化界面打印信息

```
ssm@pcie02: ~/Desktop/ailiguang/pcie_sdk_demo/src/python
File Edit View Search Terminal Help
ssm@pcie02:~/Desktop/ailiguang/pcie_sdk_demo/release/linux/bin$ cd ../../..
ssm@pcie02:~/Desktop/ailiguang/pcie_sdk_demo$ cd src/python/
ssm@pcie02:~/Desktop/ailiguang/pcie_sdk_demo/src/python$ python3 set_sensor_from
_json.py --json_file="/home/ssm/Desktop/ailiguang/pcie_card.txt
ssm@pcie02:~/Desktop/ailiguang/pcie_sdk_demo/src/python$ python3 set_sensor_from
_json.py --json_file="/home/ssm/Desktop/ailiguang/pcie_sdk_demo/config/alg_ox08b
_1_ch_96717.json" --channel=0
Set for channel 0
Request took 3205 milliseconds.
result = 0
-----finish-----
ssm@pcie02:~/Desktop/ailiguang/pcie_sdk_demo/src/python$ python3 stream_on_by_ch
annel.py --channel=0
stream on channel [0]
Request took 101 milliseconds.
result = 0, ack code [0]
-----finish-----
ssm@pcie02:~/Desktop/ailiguang/pcie_sdk_demo/src/python$
```

图 4-5 Python 脚本操作打印

执行 `python3 stream_on_by_channel.py --channel=0`，表示打开对应数据连接通道,执行后可以在 SDK 日志窗口中看到帧号和帧率等信息，如图 4-6 所示。

```
ssm@pcie02: ~/Desktop/alliguang/pcie_sdk_demo/release/linux/bin
File Edit View Search Terminal Help
[2236/0x7f24f93a1700] 29 Jul 14:04:51.260 V V4L2LOOPBACK_CTL_ADD Done, ch:0
[2236/0x7f24f93a1700] 29 Jul 14:04:51.260 V Control V4l2Loopback device done, ch
:0, card_label:AILI_PCIE_CH0
[2236/0x7f24e7fff700] 29 Jul 14:04:51.289 V Send reply message it to work queue

[2236/0x7f24f93a1700] 29 Jul 14:04:51.290 V alg_sdk_v4l2loop_writer Init Success
, ch:0
[2236/0x7f24e77fe700] 29 Jul 14:04:52.234 V [Monitor] Frame Rate : [Channel 0] [
Index 465] [fps (HW) = 29.472443] [fps (SW) = 29.732409]

[2236/0x7f24e77fe700] 29 Jul 14:04:53.239 V [Monitor] Frame Rate : [Channel 0] [
Index 495] [fps (HW) = 29.472443] [fps (SW) = 29.850746]

[2236/0x7f24e77fe700] 29 Jul 14:04:54.246 V [Monitor] Frame Rate : [Channel 0] [
Index 525] [fps (HW) = 29.472443] [fps (SW) = 29.791460]

[2236/0x7f24e77fe700] 29 Jul 14:04:55.252 V [Monitor] Frame Rate : [Channel 0] [
Index 555] [fps (HW) = 29.726517] [fps (SW) = 29.821074]

[2236/0x7f24e77fe700] 29 Jul 14:04:56.256 V [Monitor] Frame Rate : [Channel 0] [
Index 585] [fps (HW) = 29.726517] [fps (SW) = 29.880480]

[2236/0x7f24e77fe700] 29 Jul 14:04:57.259 V [Monitor] Frame Rate : [Channel 0] [
Index 615] [fps (HW) = 29.726517] [fps (SW) = 29.910269]
```

图 4-6 SDK 日志界面打印信息

上述操作完成后表示接口和通道配置完成，YGW-R1 模组的融合帧数据进入到 PCIE 采集卡中，后续可以根据需求对数据进行显示和下载操作。

4.3 融合帧数据显示

融合帧数据显示分为两种方式：

一是 cd xxx/pcie_demo_sdk/release/linux/bin 目录中，执行 sudo ./pcie_sdk_demo_init_v4l2 /dev/video100，可以看到融合帧显示视频结果；（该显示方式需要 PC 机内置显卡）

二是本司提供的显示 Yolo 脚本目录下，包含 YGW_R1_huayu.py 和 Yolo8n.pt；，执行 python3 YGW_R1_huayu.py，可以看到四个窗口分别显示 Camera，IR，Camera（Yolo）以及全融合结果（Yolo）的结果如图 4-7 所示；（执行 YGW_R1_huayu.py 脚本时可能会出现 Camera open failed 的情况，查看脚本中 143 行 cap = cv2.VideoCapture("

/dev/video1"), 通过 `ls /dev/video*` 查看 V4l2loop 对应生成的 Video 后缀, 如果生成为 Video100, 修改为 `ap = cv2.VideoCapture("/dev/video100")` 即可)

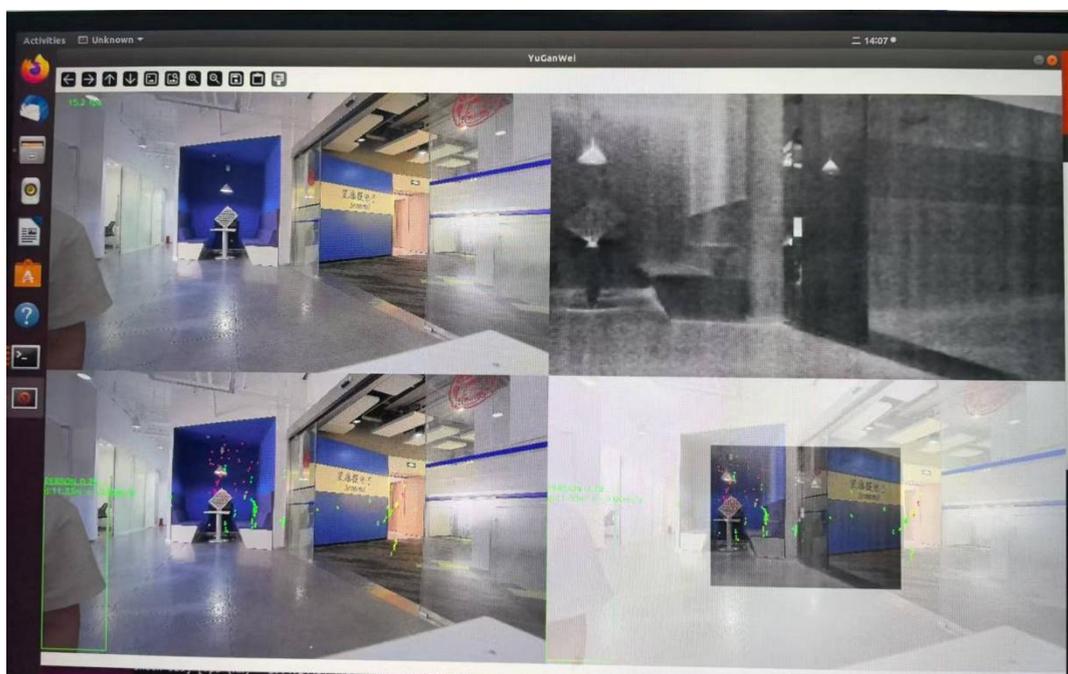


图 4-7 SDK 融合帧四宫格显示效果

4.4 融合帧数据采集

融合帧数据采集使用到本司提供的 Python 脚本为 `capture_save.py`, 运行该脚本能够从当前时刻实时保存融合帧数据到运行脚本的本地设备, 退出脚本运行表示结束保存; 运行该脚本得到的融合帧数据的原始 RAW 格式文件, 数据格式为 YUV422。

4.5 融合帧数据解析

融合帧数据可以通过本司提供 `file_process_huayu.py` 脚本将融合

帧中的 Camera 和 IR 保存成 JPG 图片格式，同时将雷达的数据保存为 CSV 格式的文件。

4.5 融合帧保存成.avi 视频文件

融合帧数据可以通过本公司提供 capture_v10_huayu_avi.py 脚本生成 xxx.avi 文件。此 xxx.avi 文件生成的目录 capture_v10_huayu_avi.py 所在目录相同。如需播放请在 windows 平台下播放。



五、YGW-R1 使用流程参考视频

根据第三章和第四章对于 YGW-R1 模组的产品信息、设备连接使用以及数据采集解析等流程介绍，能够实现多传感器融合数据以时空对齐的方式被后端接收处理。除了上述介绍的数据采集和解析方法外，借助艾利光 PCIE 视频采集卡配套工具，提供实现全流程的操作视频《YGW-R1 产品使用流程》以供参考

