



海思AI芯片培训教程

SS928/3403

2024-04-01
爱爬山的木木



个人介绍

1

十多年嵌入式软件开发

2

各种安防类产品从选型到量产经验

3

了解安防芯片生态

课程提纲



1

海思SS928 SDK介绍

2

嵌入式开发环境培训

3

SDK编译, 烧录

4

MPP培训

5

开源目标检测算法移植(YOLOV5S)



课程目标

1

了解基础AI芯片规格知识

2

入门嵌入式软件开发

3

熟悉海思ss928芯片基础

4

能够搭建NPU移植环境(PC)

5

实现Yolov5开源算法到设备
完整目标检测的Demo

能力要求



1

熟练使用Ubuntu系统

2

会C编程，有项目意识

3

会简单的python编程

4

熟悉Yolo目标检测算法

5

有基本的硬件常识



1

S S 9 2 8

S D K 介 绍



总体介绍

21AP10是一颗面向市场推出的专业超高清智能网络录像机SoC。该芯片最高支持四路sensor输入，支持最高4K60的ISP图像处理能力，支持3F WDR、多级降噪、六轴防抖、硬件拼接等多种图像增强和处理算法，为用户提供了卓越的图像处理能力。

21AP10内置四核A55，提供高效且丰富和灵活的CPU资源，以满足客户计算和控制需求。集成单核MCU，以满足某些低延时要求较高场景。

21AP10集成了高效的图像分析工具推理单元，最高10.4Tops INT8，并支持业界主流的图像分析工具框架。并内置双核Vision DSP，以满足客户一些差异化的CV计算需求。

21AP10采用先进的12nm低功耗工艺和0.65pitch封装，同时支持LPDDR4/LPDDR4x/DDR4颗粒，满足客户应用的产品小型化设计和快速量产。

21AP10配套提供的稳定、易用的SDK设计，能够支撑客户快速产品量产。

关键特性

- 智能加速
 - 10.4Tops INT8 双NN加速引擎
 - 双核Vision Q6的DSP处理
- 4K60 编解码
 - 支持4K60的H.265/H.264编码。
 - 支持10路1080p30的H.265/H.264解码。
- 支持 4 路 4M 实时拼接
 - 支持4路4Mp30机内实时硬化拼接。
- 高速接口
 - 支持USB3.0和PCIe2.0高速接口。
- 小型化封装
 - 采用23mm x 23mm FC-BGA封装形式。

名称	时间	类型	大小
ReleaseDoc	2024/4/1 20:04	文件夹	
SS928V100R001C02SPC022	2024/4/1 21:02	文件夹	
aarch64-mix210-linux.tgz	2024/4/1 20:01	WinRAR 压缩文...	254,115 KB
cc-riscv32-cfg5-musl-20211008-elf.ta...	2024/4/1 19:57	WinRAR 压缩文...	57,977 KB
SS928V100R001C02SPC022.part1.rar	2024/4/1 20:06	WinRAR 压缩文...	1,945,600...
SS928V100R001C02SPC022.part2.rar	2024/4/1 20:05	WinRAR 压缩文...	95,743 KB
SVP_PC.part1.rar	2024/4/1 20:04	WinRAR 压缩文...	1,945,600...
SVP_PC.part2.rar	2024/4/1 20:04	WinRAR 压缩文...	1,484,433...

序号	路径	说明
1	ReleaseDoc	所有的SDK文档
2	SS928V100R001C02SPC022	SDK总目录
3	aarch64-mix210-linux.tgz	交叉编译工具
4	\SS928V100R001C02SPC022\01.software\board\SS928V100_SDK_V2.0.2.2.tgz	芯片SDK源代码
5	SS928V100R001C02SPC022\ReleaseDoc\zh\00.hardware\chip\22AP70交付件\00.hardware\chip\22AP70 超高清智能网络录像机 SoC 产品简介.pdf	芯片手册简介
6	SS928V100R001C02SPC022\ReleaseDoc\zh\00.hardware\chip\22AP70交付件\00.hardware\chip\21AP10 超高清智能网络录像机 SoC 用户指南.pdf	芯片手册详细使用
7	SS928V100R001C02SPC022\SS928V100R001C02SPC022\01.software\pc\ToolPlatform\ToolPlatform-1.0.11-win32-x86_64.zip	芯片烧写工具，裸片烧写软件
8	SS928V100R001C02SPC022\ReleaseDoc\zh\01.software\board\MPP\MPP 媒体处理软件 V5.0 开发参考.pdf	Mpp 开发参考文档

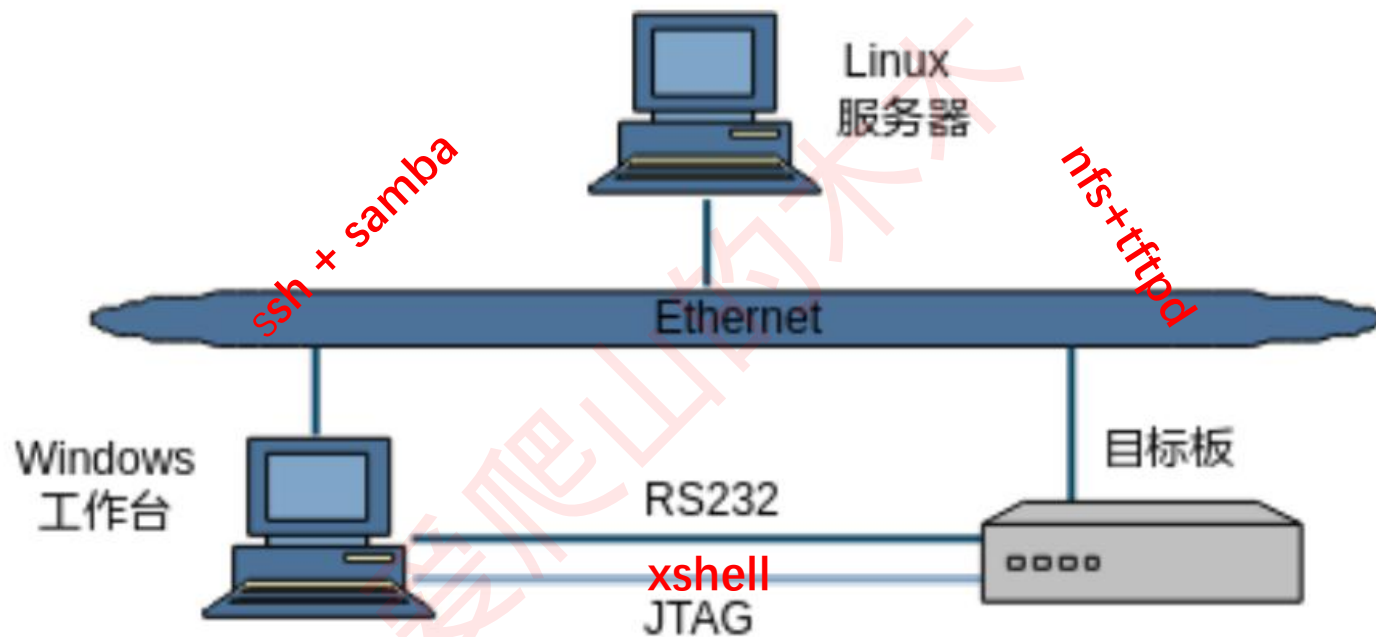


2

嵌入式开发环境培训



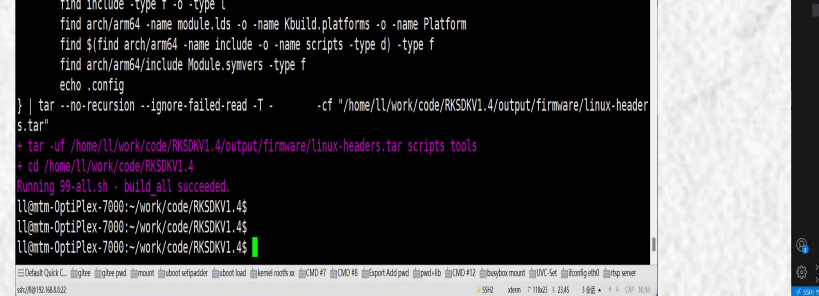
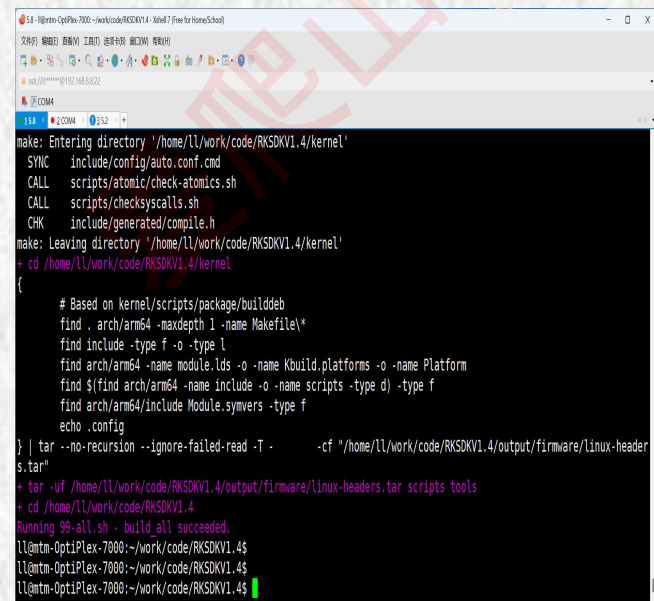
图1-2 SS928V100 开发环境



由于嵌入式单板的资源有限，不能在单板上运行开发和调试工具，通常需要交叉编译调试的方式进行开发和调试，即“宿主机 + 目标机”的形式

开发主要软件

xShell	Windows 系统， 开发工具， 写代码的插件
Ubuntu18.04 Server	开发工具， 写代码的插件
vscode	写代码的插件



2

开发环境实操作

1.安装ssh server

```
sudo apt install openssh-server
```

2.使用xshell链接linux Server

3.修改源, 更新系统

4.添加samba信息

```
sudo apt-get install samba
```

```
sudo smbpasswd -a ll #创建smb用户ll
```

```
vi /etc/samba/smb.conf
```

[mm]

```
comment = ll
```

```
path = /home/ll/work
```

```
available= yes
```

```
browseable = yes
```

```
writable = yes
```

```
guest ok = yes
```

```
valid users = ll
```

5.增加nfs配置

```
sudo apt-get install nfs-kernel-server
```

```
/etc/exports
```

```
/home/ll/work/nfs *(rw, sync, no_root_squash, no_subtree_check)
```

```
mount -t nfs -o nolock 192.168.31.125:/home/wayne/nfs /mnt/net
```

6.增加tftpd

```
vim /etc/default/tftpd-hpa
```

```
TFTP_DIRECTORY="/home/ll/work/tftpd"
```

7.创建密钥相关

```
ssh-keygen -t rsa -b 4096 #在ubuntu上创建密钥和相关文件夹
```

把Windows的公钥添加到authorized_keys

```
cat id_rsa.pub >> .ssh/authorized_keys
```




3

SDK 编译与烧录



1.安装交叉编译工具

2.编译SDK

3.烧录固件



1. ReleaseDoc\zh\01.software\board\OSDRV\SS928V100/SS927V100 开发环境用户指南.pdf
编译环境相关说明文档
2. ReleaseDoc\zh\01.software\board\SS928V100/SS927V100 SDK 安装以及升级使用说明.pdf
SDK烧录相关文档
3. SS928V100_SDK_V2.0.2.2/osdrv/readme_en.txt
SDK编译，模块编译文档

```
ll@ll:~/work/code/sdk_ss928/aarch64-mix210-linux$ tree
.
├── aarch64-mix210-linux.install
├── aarch64-mix210-linux.tar.bz2
├── readme.txt
└── runtime_lib.tgz

0 directories, 4 files
```

```
ll@ll:~/work/code/sdk_ss928/SS928V100_SDK_V2.0.2.2$ tree
.
├── package
│   ├── open_source.tgz
│   ├── osdrv.tgz
│   ├── platform.tgz
│   └── smp.tgz
├── scripts
│   └── common.sh
├── sdk.cleanup
└── sdk.unpack

2 directories, 7 files
```

3

编译 SDK

1.1 系统sh选择

```
sudo dpkg-reconfigure dash 选择 no
```

1.2 更新系统

```
sudo apt-get install make libc6-i386 lib32z1 lib32stdc++6 zlib1g-dev libncurses5-dev ncurses-term libncursesw5-dev g++ u-boot-tools texinfo texlive gawk libssl-dev openssl bc p7zip-full gperf bison flex diffutils git unzip zlib1g-dev liblzo2-dev uuid-dev pkg-config automake
```

2.解压SDK

```
cd SS928V100_SDK_V2.0.2.2/  
./sdk.unpack
```

3.修改SDK makefile

3.1 修改硬件相关内容

flash, chip

3.2 修改Uboot相关内容

```
./tools/pc/uboot_tools/SS928V100_demo_8L_LPDDR4x_3733M-8GB_32bitx2-A55_1400M-emmc.xlsm
```

3.3注释liteos, ipc相关内容

4.编译整个SDK

```
cd osdrv  
make all
```


3

单独编译模块



一、编译uboot

```
SS928V100_SDK_V2.0.2.2/osdrv  
make gslboot_build  
SS928V100_SDK_V2.0.2.2/osdrv/pub/ss928v100_emmc_image_glibc/boot_image.bin
```

二、编译内核

```
make atf  
sdk/SS928V100_SDK_V2.0.2.2/osdrv/pub/ss928v100_emmc_image_glibc/uImage_ss928v100
```

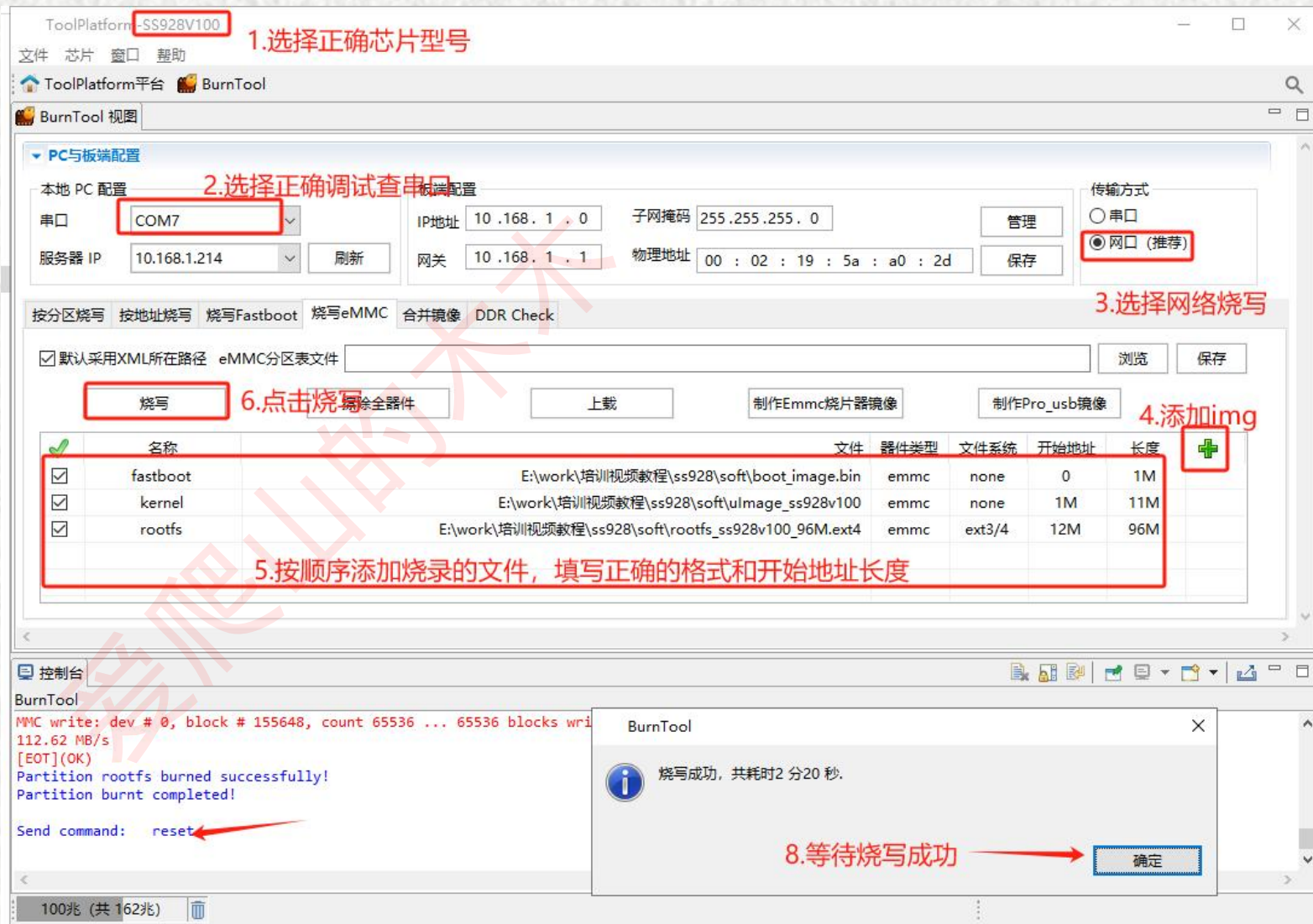
3

烧录系统

```
ll@ll:~/work/code/sdk_ss928/SS928V100_SDK_V2.0.2.2
/osdrv/pub/ss928v100_emmc_image_glibc$ tree

├── rootfs_ss928v100_128k.jffs2
├── rootfs_ss928v100_256k.jffs2
├── rootfs_ss928v100_2k_128k_32M.ubifs
├── rootfs_ss928v100_4k_256k_50M.ubifs
├── rootfs_ss928v100_64k.jffs2
├── rootfs_ss928v100_96M.ext4
├── u-boot-ss928v100.bin
└── uImage_ss928v100

0 directories, 8 files
```



SS928V100R001C02SPC022\01.software\pc\ToolPlatform\ToolPlatform-1.0.11-win32-x86_64.zip



在嵌入式系统中，bootargs是非常重要的参数。它保存着Linux内核启动的相关信息，如启动参数和控制台信息等。而在ARM架构的系统中，bootargs参数则是在U-Boot引导时设置的。

Flash布局

1MB	11MB	96MB	1024MB
boot_image.bin	kernel	rootfs	app

```
setenv bootargs 'mem=512M console=ttyAMA0,115200 clk_ignore_unused rw rootwait root=/dev/mmcblk0p3 rootfstype=ext4
blkdevparts=mmcblk0:1M(uboot.bin),11M(kernel),96M(rootfs.ext4),1024M(app)';
setenv bootcmd 'mmc read 0 0x50000000 0x800 0x5800; bootm 50000000';
sa;
```

$0x800 = 1 \times 1024 \times 1024 / 512 (\text{BlockSize}) = 2048$

$0x5800 = 0x800 \times 11$

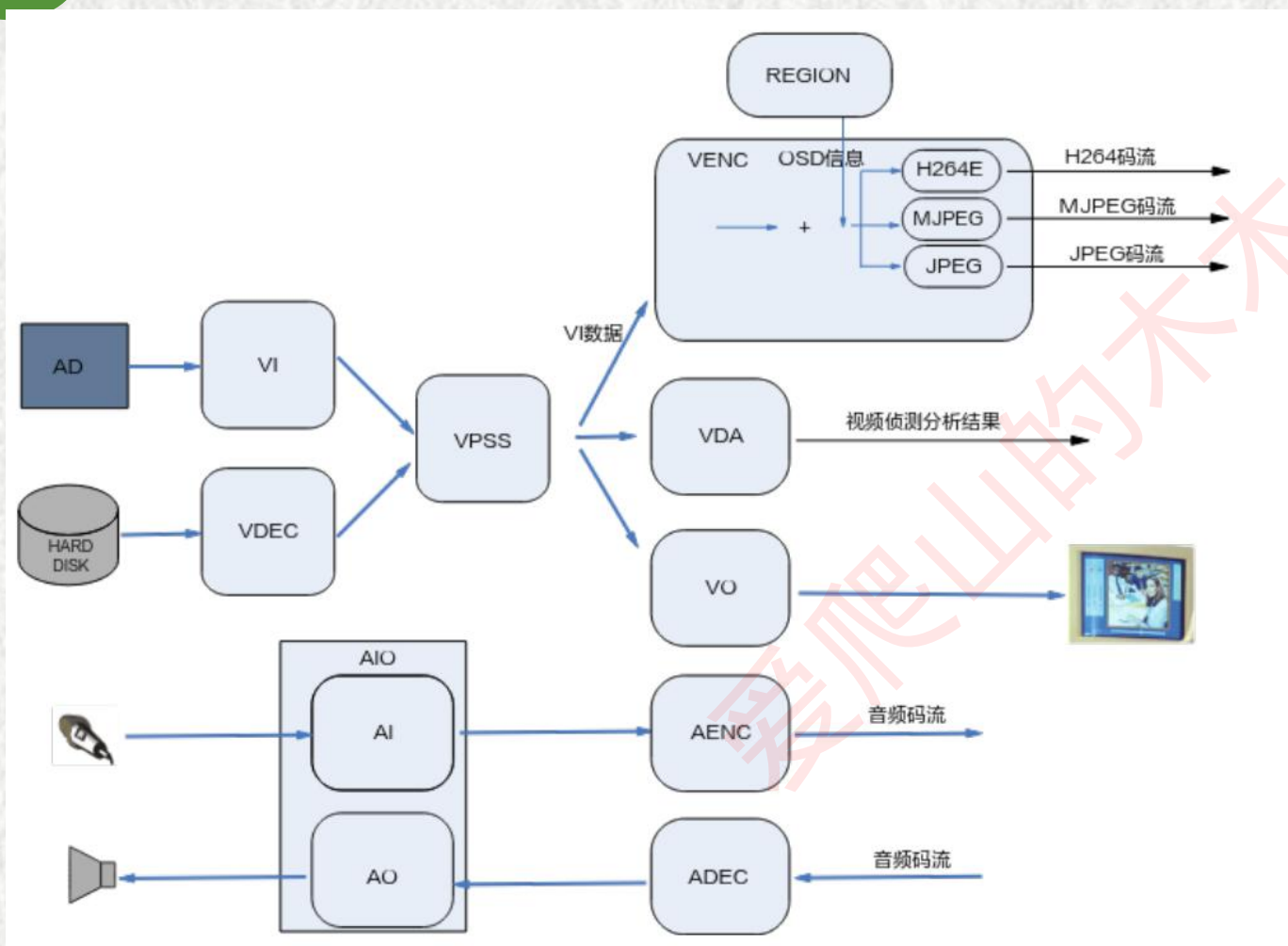


4

M P P 基础讲解

4

Mpp 内部数据流程图



MPP在系统中的位置

4

运行 Mpp Sample



1

loadko.sh 脚本

2

sample_vio 编译

3

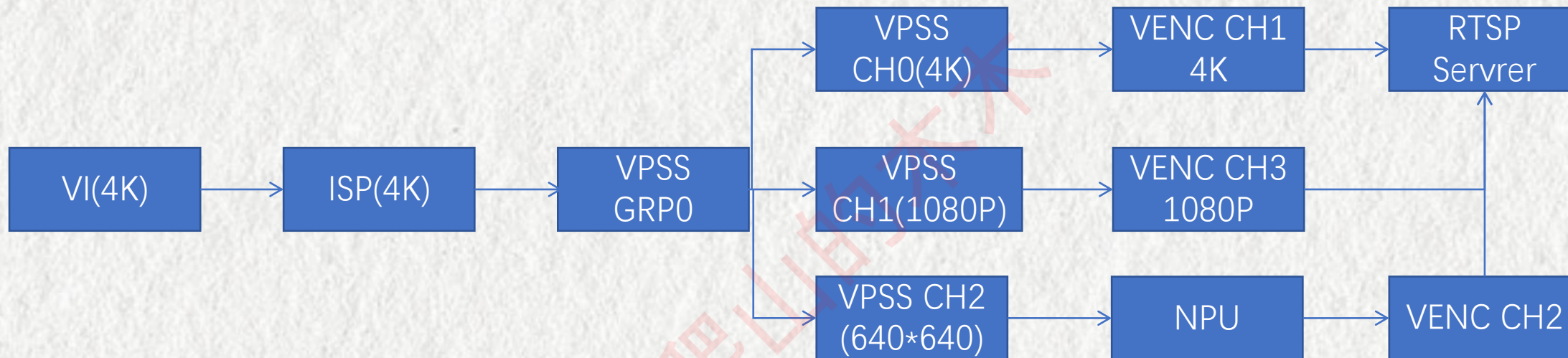
mpp proc 信息查看

4

pipeline

4

pipeline



- 1.支持一路4K主码流RTSP
- 2.支持preview码流一路
- 3.支持算法通道一路

https://gitee.com/apchy_ll/librtsp-server.git



5

开源目标检测算法移植



1

板端环境装备，npu Sample

2

ATC转换环境搭建

3

Yolov5s模型移植，后处理代码实现

5

板端环境装备，npu Sample

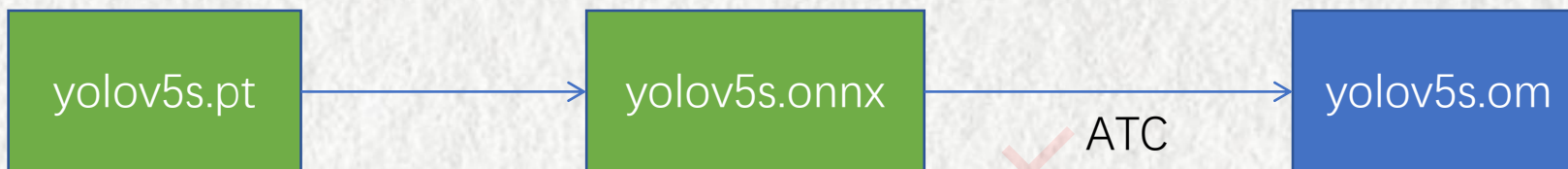


1

编译 Sample

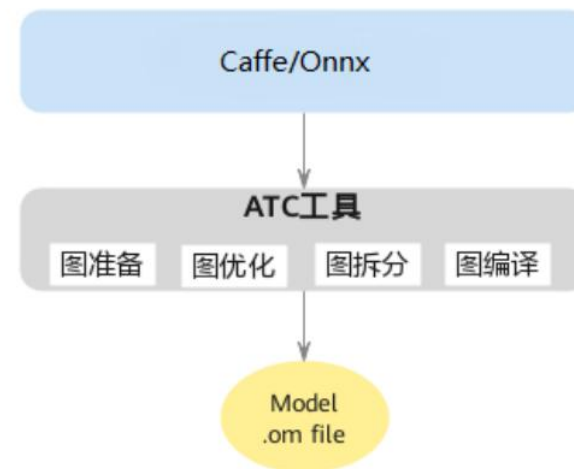
2

复制 sample、data、lib 到板端



<https://gitee.com/monkeycc/yolov5.git>

图1-1 ATC 工具功能架构



ReleaseDoc\zh\01.software\pc\SVP_NNN\驱动和开发环境安装指南.pdf

ReleaseDoc\zh\01.software\pc\SVP_NNN\ATC工具使用指南.pdf

SVP_PC\SVP_PC\NNN_PC\NNN_PC\Ascend-cann-toolkit_5.20.t6.2.b060_linux-x86_64.run



```
conda create -n atc python=3.9.2
conda activate atc
pip3 install protobuf==3.13.0 --user -i https://pypi.tuna.tsinghua.edu.cn/simple
pip3 install psutil==5.7.0 --user -i https://pypi.tuna.tsinghua.edu.cn/simple
pip3 install numpy --user -i https://pypi.tuna.tsinghua.edu.cn/simple
pip3 install scipy --user -i https://pypi.tuna.tsinghua.edu.cn/simple
pip3 install decorator==4.4.0 --user -i https://pypi.tuna.tsinghua.edu.cn/simple
pip3 install sympy==1.5.1 --user -i https://pypi.tuna.tsinghua.edu.cn/simple
pip3 install cffi==1.12.3 --user -i https://pypi.tuna.tsinghua.edu.cn/simple
pip3 install pyyaml --user -i https://pypi.tuna.tsinghua.edu.cn/simple
pip3 install pathlib2 --user -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
atc --model=yolov5s.onnx --framework=5 --output=yolov5s --input_format=NCHW --
input_shape="images:1,3,640,640" --log=debug --soc_version="OPTG"
```


5

Yolov5s 模型移植，后处理代码实现



1

移植 NCNN

2

板端 OM 运行 Demo

爱爬山的木木



- 1 对于Yolov5s执行效果有个比对
- 2 使用NCNN里面自带的图像前处理加载图片
- 3 直接使用Demo里面的后处理

<https://github.com/Tencent/ncnn.git>

5

Yolov5s 模型移植，后处理代码实现



1 ONNX 转 OM 支持 YUV420SP 输入

2 Sample 分析

3 Demo 代码讲解

https://gitee.com/apchy_ll/ss928_yolov5s.git



感谢观看

THANKS FOR WATCHING

哔哩哔哩<https://www.bilibili.com/video/BV1bH4y1u7nv/>