

一. 安卓容器

1. 下载我开启了安卓容器选项的内核进行安装或者下载我最新编译的系统不需要更新内核
[内核deb包](#)

2. 安装ppa源里的malig610固件

```
sudo apt install mali-g610-firmware
```

3. 首先安装好docker

```
curl -fsSL https://get.docker.com | bash -s docker --mirror Aliyun
```

4. 拉取安卓镜像, 使用chisbread/rk3588-gaming:redroid-firefly

```
docker pull chisbread/rk3588-gaming:redroid-firefly
```

5. 创建容器

```
docker run -itd --privileged \
    --pull always \
    -v "$(pwd)"/Android:/data \
    -v /dev:/dev \
    -v /run/dbus:/run/dbus \
    -v /var/run/dbus:/var/run/dbus \
    --mount=type=bind,source=/dev/mali0,destination=/dev/mali0 \
    -p 5555:5555 \
    chisbread/rk3588-gaming:redroid-firefly \
    androidboot.redroid_width=1080 \
    androidboot.redroid_height=1920 \
    androidboot.redroid_dpi=480 \
    androidboot.redroid_fps=30 \
    androidboot.redroid_gpu_mode=host
```

6. win使用qtscrcpy链接安卓容器即可使用

7. 5b上使用scrcpy即可链接安卓容器

声音使用sndcpy或者scrcpy-server2.0版本均不能在5b通过3.5mm耳机孔播放声音, win和mac音频传输正常

摄像头经测试无法在安卓容器内使用。但是安卓容器内显示有video文件-----等待解决

Armbian使用官方4k摄像头

下载[Camera](#)设备树

放置于/boot下，使用armbian-add-overly添加即可

二. spi点亮0.96oled屏幕

如代码所示

无法点亮，但数据收发都正常且时序正确

三. rknpu简单使用

3.1 使用x86模拟npu

1. 克隆两个demo仓库\

```
git clone https://github.com/rockchip-linux/rknpu2.git
git clone https://github.com/rockchip-linux/rknn-toolkit2.git
```

2. rknn-toolkit2是x86使用的工具，以下操作均在x86电脑上完成

```
sudo apt-get install libxslt1-dev zlib1g-dev libglib2.0 libsm6 libgl1-mesa-glx
libprotobuf-dev gcc
cd ./rknn-toolkit2          # 进入rknn-toolkit目录
pip3 install -r doc/requirements_cp36-1.4.0.txt      # py3.6用这个
pip3 install -r doc/requirements_cp38-1.4.0.txt      # py3.8用这个 其实这两个文件没
有区别，任何py版本都可以用
# py3.9也可以用
pip3 install ./packages/rknn_toolkit2-1.4.0_22dcfef4-cp36-cp36m-
linux_x86_64.whl# py3.6用这个
pip3 install ./packages/rknn_toolkit2-1.4.0_22dcfef4-cp38-cp38-linux_x86_64.whl
# py3.8用这个
```

3. 运行demo

```
cd examples/onnx/yolov5
python3 test.py
```

如果运行完后，没有结果图片，修改test.py

```
# 在这段注释后添加retval = cv2.imwrite("result.jpg",img_1)
# show output
# cv2.imshow("post process result", img_1)
# cv2.waitKey(0)
# cv2.destroyAllWindows()
retval = cv2.imwrite("result.jpg",img_1)
```

3.2 连板调试

不需要有线，没找到rknnserver源码所以必须要用到adb

Linux系统

我没有找到办法开启adb server使win能通过adb链接linux

安卓系统

安卓系统没有使用过，根据文档说明需要usb链接板子和电脑

adb无线链接没有测试

上传rknn_server和librknnrt.so到板子

运行rknn_server

```
rknn_server
# 如果没有执行权限加执行权限
sudo chmod +x ./rknn_server
```

- librknnrt.so: 是一个板端的runtime 库。
- rknn_server: 是一个运行在板子上的后台代理服务，用于接收PC 通过USB 传输过来的协议,然后执行板端runtime 对应的接口，并返回结果给PC

以下操作在x86电脑上进行

使用adb devices 查看设备ID

修改脚本target 和device_id

- rknn-toolkit2/examples/onnx/yolov5/test.py

```
rknn.config(mean_values=[[0, 0, 0]], std_values=[[255, 255, 255]])
修改为---->
rknn.config(mean_values=[[0, 0, 0]], std_values=[[255, 255,
255]],target_platform='rk3588')
ret = rknn.init_runtime()
修改为---->
ret = rknn.init_runtime(target='rk3588',device_id='your device id')
```

```
# 终端进入rknn-toolkit2/examples/onnx/yolov5目录
python3 test.py
```

3.3在板子上运行demo

安卓

查看原文档关于安卓系统下的章节

Linux

以下在板子上操作

1. 克隆rknpu2仓库

```
git clone https://github.com/rockchip-linux/rknpu2.git
cd rknpu2/examples/rknn_yolov5_demo
./build-linux_RK3588.sh
cd ./install/rknn_yolov5_demo_Linux
./rknn_yolov5_demo ./model/RK3588/yolov5s-640-640.rknn
.././convert_rknn_demo/yolov5/bus.jpg
# 生成out.jpg作为结果
```

3.4 后续学习

建议阅读rknpu2仓库中doc中的Rockchip_RKNPU_User_Guide_RKNN_API_V1.4.0_CN.pdf文档，使用yolov5_demo练习api使用

然后使用自己的模型，阅读RKNN_Compiler_Support_Operator_List_v1.4.0.pdf针对自己模型中npu不支持的算子进行重写，修剪模型。

3.5 其他重要命令

- CPU 定频命令

查看cpu频率

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
cat /sys/kernel/debug/clk/clk_summary | grep arm
```

固定cpu频率

```
# 查看CPU 可用频率
cat /sys/devices/system/cpu/cpufreq/policy0/scaling_available_frequencies
408000 600000 816000 1008000 1200000 1416000 1608000 1704000
# 设置CPU 频率，例如，设置1.7GHz
echo userspace > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
echo 1704000 > /sys/devices/system/cpu/cpufreq/policy0/scaling_setspeed
```

- DDR 定频命令

查看DDR频率

```
cat /sys/class/devfreq/dmc/cur_freq
cat /sys/kernel/debug/clk/clk_summary | grep ddr
```

固定DDR频率

```
# 查看DDR 可用频率
cat /sys/class/devfreq/dmc/available_frequencies
# 设置DDR 频率，例如，设置1560MHz
echo userspace > /sys/class/devfreq/dmc/governor
echo 1560000000 > /sys/class/devfreq/dmc/userspace/set_freq
```

o npu定频

查看npu频率

```
cat /sys/kernel/debug/clk/clk_summary | grep clk_npu_dsu0
```

固定npu频率

```
# 设置NPU 频率，例如，设置1GHz
echo 1000000000 > /sys/kernel/debug/clk/clk_npu_dsu0/clk_rate
```