

Classification Level: Top secret () Secret () Internal () Public (√)

RKNN SDK Quick Start Guide

(Technology Department, Graphic & Computing Platform Center)

Mark: [] Editing [√] Released	Version:	1.5.2
	Author:	NPU
	Completion Date:	2023-8-21
	Auditor:	Vincent
	Reviewed Date:	2023-8-21

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd

(All rights reserved)

Revision History

Version	Author	Revision Date	Revision Notes	Auditor
1.3.0b0	Shuangjie Lin	2022-4-23	Initial version.	Vincent
1.3.3b0	Shuangjie Lin	2022-6-20	1. Modify the downloaded network disk address. 2. Modify the lib library path setting issue.	Vincent
1.4.0	Shuangjie Lin	2022-08-29	update version.	Vincent
1.4.2	HPC	2023-2-13	update version.	Vincent
1.5.0	HPC	2023-5-22	update version.	Vincent
1.5.2	HPC	2023-8-21	update version.	Vincent

Contents

1 Overview.....	4
2 Prepare Tools.....	4
3 Quick Start to Use RKNN-Toolkit2 And RKNPU2.....	6
3.1 Install RKNN-Toolkit2	6
3.1.1 Install and infer using docker image	6
3.1.2 Install and reason through pip install.....	8
3.2 How to Compile and Use RKNPU2	10
3.2.1 Download the tools required for compilation.....	10
3.2.2 Compilation tool path setting of demo	11
3.2.3 Update RKNN model.....	12
3.2.4 Compile rknn_yolov5_demo.....	12
3.2.5 Run rknn_yolov5_demo on the board.....	13
4 Reference Documents	14
5 Appendix.....	15
5.1 View and set the CPU, DDR and NPU frequency of the board	15
5.1.1 CPU fixed frequency command	15
5.1.2 DDR fixed frequency command.....	15
5.1.3 NPU fixed frequency command	15
5.2 The command adb devices cannot see the device.....	16

1 Overview

This document provides a detailed introduction to users on how to quickly use RKNN-Toolkit2 and RKNPU2 tools on the EVB of the ROCKCHIP chip to convert the yolov5s.onnx model into the yolov5s.rknn model and perform edge inference.

Supported platforms: RV1106, RV1103.

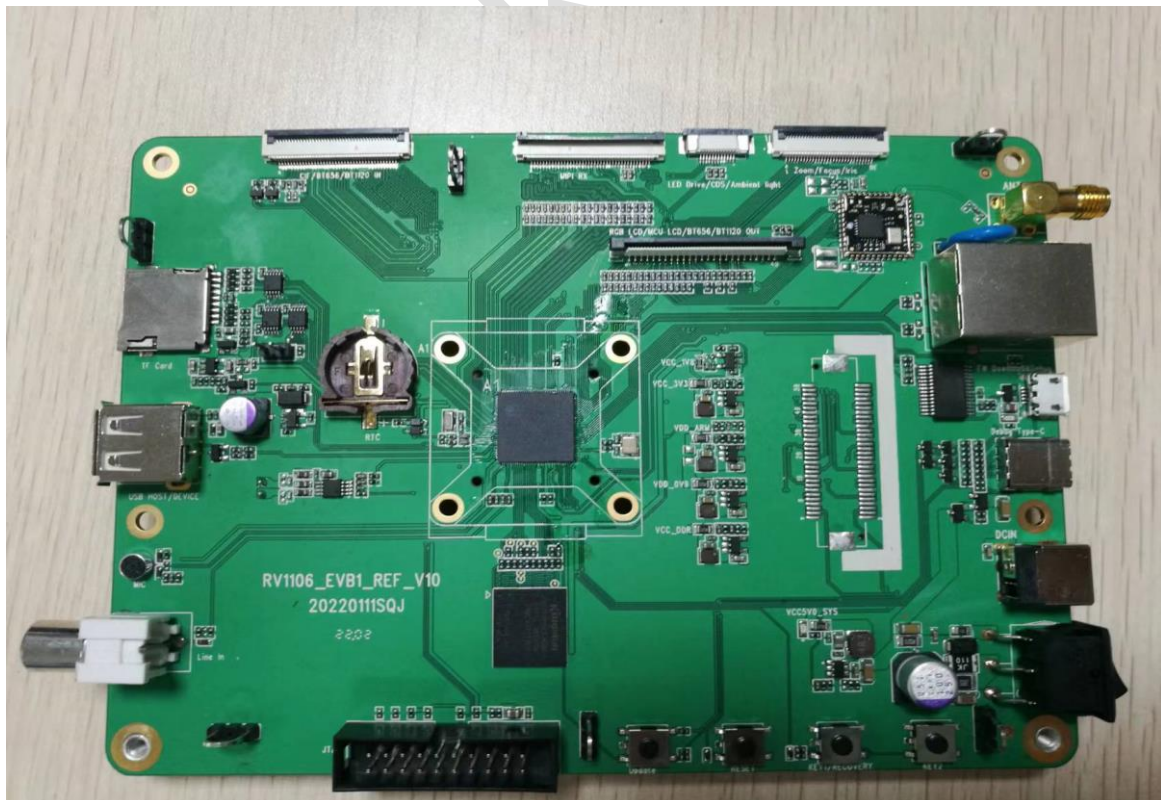
RKNPU2 project download address: <https://github.com/rockchip-linux/rknpu2>

RKNN-Toolkit2 project download address: <https://github.com/rockchip-linux/rknn-toolkit2>

2 Prepare Tools

1. A computer with Ubuntu18.04 / Ubuntu20.04 / Ubuntu22.04 operating system.
2. One EVB (RV1106, RV1103)

RV1106



-
3. A data cable connecting the board to the computer.

RV1106: Double-ended USB data cable



4. A power adapter.

RV1106: Output 12V-2A



3 Quick Start to Use RKNN-Toolkit2 And RKNPU2

3.1 Install RKNN-Toolkit2

This chapter introduces two methods of installing and using RKNN-Toolkit2, ‘installation via pip install’ and ‘installation via Docker image’. Users can choose the installation method by themselves. If the computer is not Ubuntu18.04/Ubuntu20.04/Ubuntu22.04 system, it is recommended to use the ‘installation via Docker image’ method, which has integrated all the required installation package dependencies, which is simple and reliable.

The following operations use Ubuntu18.04 and Python3.6 as examples.

3.1.1 Install and infer using docker image

1. If the Docker tool is not installed on the computer, please follow this installation tutorial (<https://mirrors.tuna.tsinghua.edu.cn/help/docker-ce/>) to install the Docker tool before proceeding to the next step.
2. Open a terminal command line window, cd into the docker folder of the RKNN-Toolkit2 project, and modify the path in the cd command according to the save path of the project.

cd <Enter the path of the docker folder in the RKNN-Toolkit2 project>

Command:

```
cd ~/Projects/rknn-toolkit2-1.x.x/docker/docker_full
ls
```

It is found that there is a docker image file rknn-toolkit2-1.x.x-cp36-docker.tar.gz in the current directory.

3. Load the docker image.

```
docker load --input rknn-toolkit2-1.x.x-cp36-docker.tar.gz
```

4. View all current docker images.

Command:

```
docker images
```

It can be found that the REPOSITORY is rknn-toolkit2, and the TAG is 1.x.x-cp36, which means the loading is successful.

5. Run docker container.

Command:

```
docker run -t -i --privileged -v /dev/bus/usb:/dev/bus/usb \  
-v ~/Projects/rknpu2/examples/RV1106_RV1103/rknn_yolov5_demo:/rknn_yolov5_demo \  
rknn-toolkit2:1.x.x-cp36 /bin/bash
```

Mapping a directory into a Docker environment can be done by appending '-v <host src folder>:<image dst folder>'.

The green part is the example/RV1106_RV1103/rknn_yolov5_demo local folder path in the rknpu2 project (modified according to the local path) mapped to the /rknn_yolov5_demo folder in the docker container.

After successfully entering the docker container, the command ls can view the folder rknn_yolov5_demo, indicating that the mapping is successful.

6. Enter the rknn_yolov5_demo/convert_rknn_demo/yolov5 directory in the docker container.

Command:

```
cd rknn_yolov5_demo/convert_rknn_demo/yolov5
```

7. Convert yolov5s.onnx to rknn model.

```
python3 onnx2rknn.py
```

Command:

```
pip3 install -r doc/requirements_cp36-1.x.x.txt
```

Note:

1) If the error 'XX version cannot be matched' occurs during the installation process, it may be caused by the pip version being too low. You can execute the following upgrade pip version command first, upgrade pip to version 21.3.1, and then execute the above installation command again.

```
python3 -m pip install --upgrade pip
```

5. Install RKNN-Toolkit2 (Python3.6 for x86_64).

Command:

```
pip3 install \
package/rknn_toolkit2-1.x.x+xxxxxxxx-cp36-cp36m-linux_x86_64.whl
```

6. Check whether RKNN-Toolkit2 is installed successfully.

Command:

```
python3
from rknn.api import RKNN
```

```
Python 3.6.9 (default, Dec 8 2021, 21:08:43)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from rknn.api import RKNN
>>>
```

If there are no errors, the installation is successful. Press and hold Ctrl+D to exit Python3.

7. cd into rknpu2/examples/RV1106_RV1103/rknn_yolov5_demo/convert_rknn_demo/yolov5 directory.

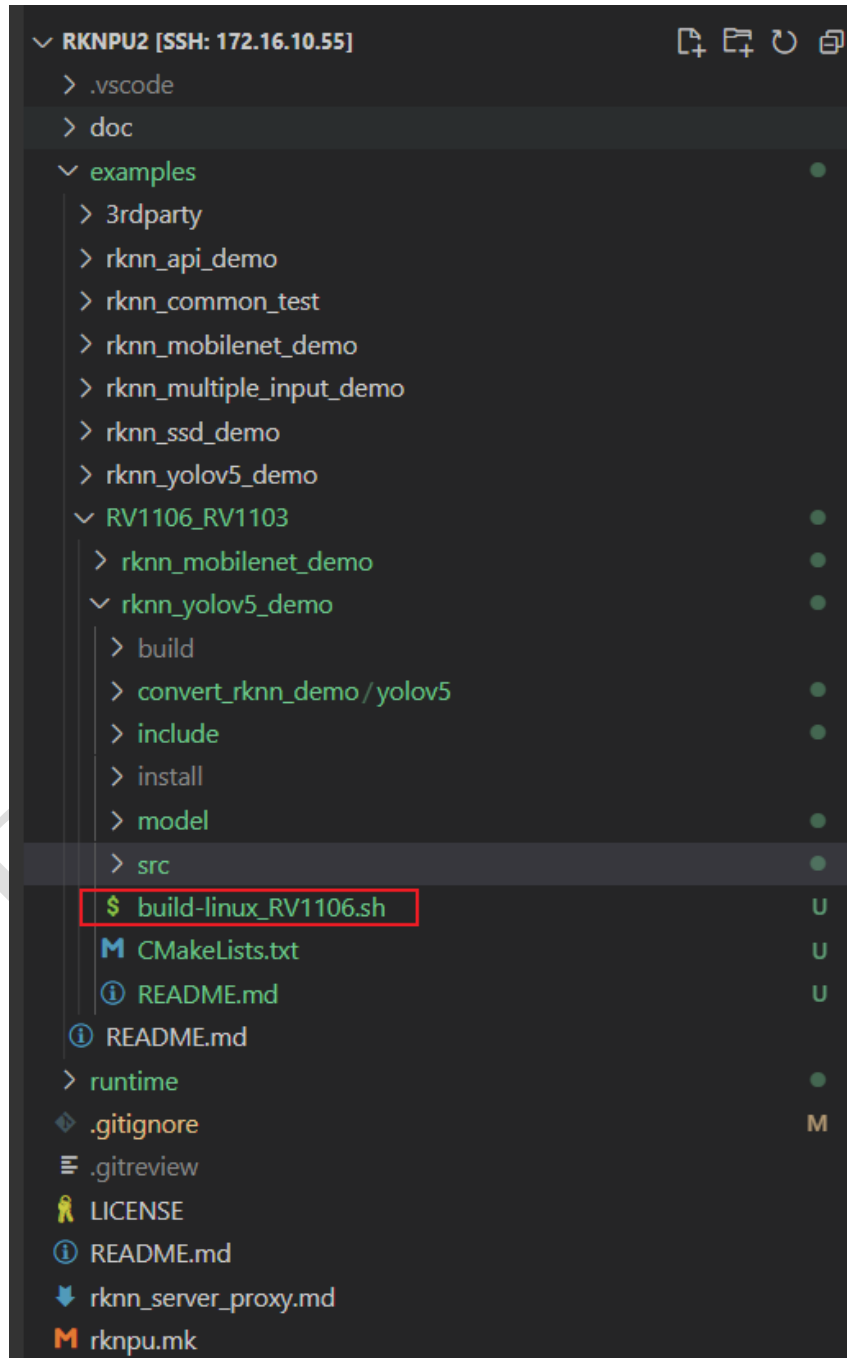
through baidu online storage shown below:

Link:<https://eyun.baidu.com/s/3qZxSDNQ>

Password:rknn

3.2.2 Compilation tool path setting of demo

Open the compile script examples/RV1106_RV1103/rknn_yolov5_demo/build-linux_RV1106.sh.



1) Linux system

Set `RK_RV1106_TOOLCHAIN` to the path of the local computer `arm-rockchip830-linux-uclibcgnueabi` and save it.

```
1  #!/bin/bash
2  set -e
3
4  if [ -z $RK_RV1106_TOOLCHAIN ]; then
5      echo "Please set the RK_RV1106_TOOLCHAIN environment variable!"
6      echo "example:"
7      echo "  export RK_RV1106_TOOLCHAIN=<path-to-your-dir/arm-rockchip830-linux-uclibcgnueabi>"
8      exit
9  fi
10
11 # for arm
12 GCC_COMPILER=$RK_RV1106_TOOLCHAIN
13
14 ROOT_PWD=$( cd "$( dirname $0 )" && cd -P "$( dirname "$SOURCE" )" && pwd )
15
```

3.2.3 Update RKNN model

Copy the converted RV1106 platform model `yolov5s-640-640.rknn` in Chapter 3.1 to the `rknpu2/examples/RV1106_RV1103/rknn_yolov5_demo/model/RV1106/` directory.

3.2.4 Compile rknn_yolov5_demo

1) Enter the `rknn_yolov5_demo` folder in the terminal command window.

Command:

```
cd examples/RV1106_RV1103/rknn_yolov5_demo/
```

2) Run the `build-linux_RK1106.sh` script to compile the program.

Command:

```
./ build-linux_RV1106.sh
```

PS:

- 1) Compiling RV1106 only supports arm linux compilation. For details, please refer to `/rknpu2/examples/RV1106_RV1103/rknn_yolov5_demo/README.md`.

-
- 2) If a cmake error occurs during compilation, you can execute the following command to install cmake and then run the compilation script.

```
sudo apt install cmake
```

3.2.5 Run rknn_yolov5_demo on the board

- 1) Upload the compiled program and the required files install/rknn_yolov5_demo_Linux folder to the /data/ folder of the board.

Command:

```
adb root
adb push install/rknn_yolov5_demo_Linux /data/
```

- 2) Enter the board system.

Command:

```
adb shell
```

- 3) cd enter the directory where the program is located.

Command:

```
cd /data/rknn_yolov5_demo_Linux/
```

- 4) Set the library file path (**special attention: the path must be an absolute full path**).

Command:

```
export LD_LIBRARY_PATH=/data/rknn_yolov5_demo_Linux/lib
```

- 5) Run the program to identify the category of the object in the corresponding picture.

Usage: ./rknn_yolov5_demo <rknn model> <jpg>

Command:

```
./rknn_yolov5_demo ./model/RV1106/yolov5s-640-640.rknn ./model/bus.jpg
```

```
# ./rknn_yolov5_demo model/RV1106/yolov5s-640-640.rknn model/bus.jpg
rknn_api/rknnrt version: 1.5.2 (a1529deb8@2023-08-22T14:57:28), driver version: 0.8.8
model input num: 1, output num: 3
input tensors:
  index=0, name=images, n_dims=4, dims=[1, 640, 640, 3], n_elems=1228800, size=1228800, fmt=NHWC, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003922
output tensors:
  index=0, name=output, n_dims=4, dims=[1, 80, 80, 255], n_elems=1632000, size=1632000, fmt=NHWC, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003860
  index=1, name=283, n_dims=4, dims=[1, 40, 40, 255], n_elems=408000, size=408000, fmt=NHWC, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003922
  index=2, name=285, n_dims=4, dims=[1, 20, 20, 255], n_elems=102000, size=102000, fmt=NHWC, type=INT8, qnt_type=AFFINE, zp=-128, scale=0.003915
custom string:
Begin perf ...
0: Elapse Time = 79.62ms, FPS = 12.56
model is NHWC input fmt
loadLabelName ./model/coco_80_labels_list.txt
person @ (208 244 286 506) 0.884136
person @ (479 238 560 526) 0.863766
person @ (110 236 230 535) 0.832498
bus @ (94 130 553 464) 0.697389
person @ (79 354 122 516) 0.349307
```

4 Reference Documents

For more detailed usage and description of RKNN-Toolkit2, please refer to ‘Rockchip_User_Guide_RKNN_Toolkit2_CN.pdf’ manual.

For more detailed usage and description of RKNPU API, please refer to ‘Rockchip_RKNPU_User_Guide_RKNN_API_CN.pdf’ manual.

5 Appendix

5.1 View and set the CPU, DDR and NPU frequency of the board

Usually, the frequency of each unit on the board is dynamically tuned. In this case, the performance of the tested model will fluctuate. In order to prevent inconsistent performance test results, it is recommended to fix the frequency of the relevant units on the board before testing during performance evaluation. The frequency viewing and setting commands of related units are as follows:

5.1.1 CPU fixed frequency command

- 1) View CPU frequency:

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

- 2) Fixed CPU frequency, example of fixing 1.6GHz on CPU (required support of firmware)

```
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/policy0/scaling_governor  
echo 1608000 > /sys/devices/system/cpu/cpu0/cpufreq/policy0/scaling_setspeed
```

5.1.2 DDR fixed frequency command

- 1) View DDR frequency:

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

- 2) Fixed DDR frequency (not settable)

5.1.3 NPU fixed frequency command

- 1) View NPU frequency (requires firmware support) :

For RV1106:

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

- 2) Fixed NPU frequency (not settable)

5.2 The command adb devices cannot see the device

1. Check whether the connection is correct, re-plug or replace the USB port of the computer.
2. When using a USB-connected board in a local computer and a docker container, only one end can use the adb server at a time. Therefore, if you cannot see the device when executing the command (adb devices) at one end, you can execute the command at the other command end.

```
adb kill-server
```

Terminate the external adb service, and then return to the original command terminal window to execute the command (adb devices) to view the device.

3. When the following error occurs, adb is not installed. You need to execute the installation command to install adb.

```
Command 'adb' not found, but can be installed with:
sudo apt install adb
```

Command:

```
sudo apt install adb
```