

# RKNN Demo Developer Guide

---

Document ID: RK-KF-YF-334

Release Version: V1.0.1

Release Date: 2020-07-22

Security Class: ☐Top-Secret ☐Secret ☐Internal ☒Public

## DISCLAIMER

THIS DOCUMENT IS PROVIDED "AS IS". ROCKCHIP ELECTRONICS CO., LTD. ("ROCKCHIP") DOES NOT PROVIDE ANY WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, WITH RESPECT TO THE ACCURACY, RELIABILITY, COMPLETENESS, MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT OF ANY REPRESENTATION, INFORMATION AND CONTENT IN THIS DOCUMENT. THIS DOCUMENT IS FOR REFERENCE ONLY. THIS DOCUMENT MAY BE UPDATED OR CHANGED WITHOUT ANY NOTICE AT ANY TIME DUE TO THE UPGRADES OF THE PRODUCT OR ANY OTHER REASONS.

## Trademark Statement

"Rockchip", "瑞芯微", "瑞芯" shall be Rockchip's registered trademarks and owned by Rockchip. All the other trademarks or registered trademarks mentioned in this document shall be owned by their respective owners.

**All rights reserved. © 2020. Rockchip Electronics Co., Ltd.**

Beyond the scope of fair use, neither any entity nor individual shall extract, copy, or distribute this document in any form in whole or in part without the written approval of Rockchip.

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian, PRC

Website: [www.rock-chips.com](http://www.rock-chips.com)

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: [fae@rock-chips.com](mailto:fae@rock-chips.com)

## Preface

### Overview

This document mainly introduces the usage of Rockchip Linux RKNN Demo, aiming to help engineers get started with RKNN Demo development and debugging methods faster.

### Product ID

Chipset Name	Buildroot	Debian	Yocto
RK1808	Y	Y	N
RK3399PRO	Y	Y	N

### Application Object

This document (this guide) is intended primarily for the following readers:

Field Application Engineer

Software Development Engineer

### Revision History

Date	Version	Author	Change Description
2018-12-08	V0.0.1	lhp	initial version
2019-02-15	V0.0.2	lhp	support 1808 and 3399pro
2019-06-05	V0.0.2	Caesar Wang	add rknn_demo FAQ
2020-03-12	V1.0.0	Caesar Wang	Markdown initial version
2020-07-22	V1.0.1	Ruby Zhang	Update the company name and the format of the document

## **Content**

### **RKNN Demo Developer Guide**

1. Run RKNN\_DEMO
  - 1.1 Overview
  - 1.2 Configure in buildroot
  - 1.3 NPU Related
  - 1.4 Compile and Run
2. RKNN\_DEMO Development
  - 2.1 File Directory Introduction
3. RKNN\_DEMO FAQ
  - 3.1 How to Switch Display 720p Resolution on HDMI

# 1. Run RKNN\_DEMO

---

## 1.1 Overview

The rknn\_demo module configuration directory is "/ buildroot/ package/ rockchip/ rknn\_demo" and the code directory is "/external/rknn\_demo". It is mainly used to collect images through USB camera and then send them to NPU for processing and display the results through MiniGUI. The currently supported model is "mobilenet\_ssd".

## 1.2 Configure in buildroot

The required configurations are enabled in SDK by default, the mainly dependences are RGA and USB camera. If they are not enabled, please go to kernel to check the historical changes of related config. Because RKNN interfaces and model of rk1808 and rk3399pro are different, you can configure according to chip type in the configuration file, mainly basing on BR2\_PACKAGE\_RK1808 and BR2\_PACKAGE\_RK3399PRO. When it is rk1808, the value of the macro "NEED\_RKNNAPI" used in the code is 0 and the value is 1 when it is rk3399pro.

## 1.3 NPU Related

The model files have been compiled into the board by default in the SDK. The corresponding file macro and directory are as follows:

```
1 | #define MODEL_NAME "/usr/share/rknn_demo/mobilenet_ssd.rknn"
2 | #define BOX_PRIORS_TXT_PATH "/usr/share/rknn_demo/box_priors.txt"
3 | #define LABEL_NALE_TXT_PATH "/usr/share/rknn_demo/coco_labels_list.txt"
```

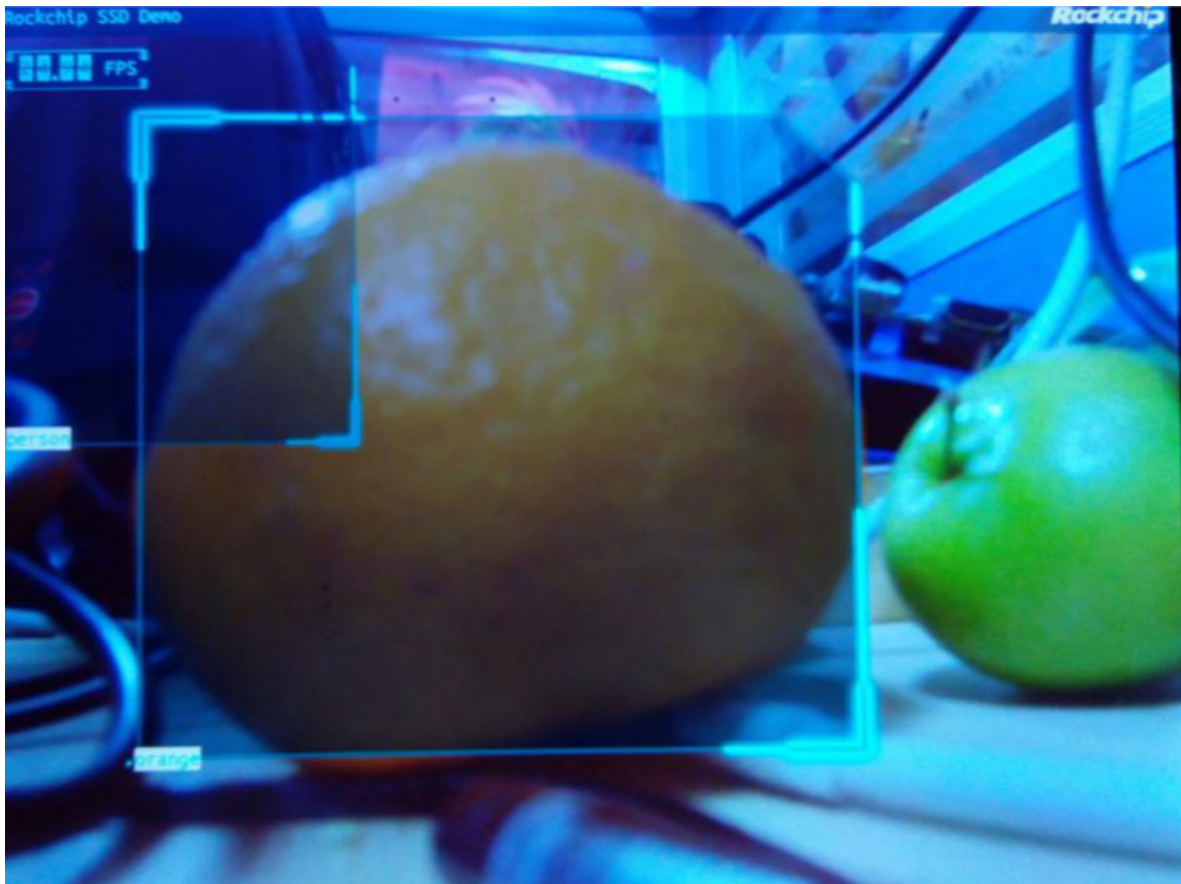
Before the model runs, make sure the related files exist.

## 1.4 Compile and Run

You can compile modules in the SDK directory with the command "make rknn\_demo" and generate the rknn\_demo executable file. Before copy to the board, make sure USB camera is plugged in, and run "rknn\_demo" command directly. Note: should not be coexist with other UIs. Please delete the related UI startup commands before starting. The board's default UI is QT, you can run the command:

```
1 | /etc/init.d/S50launcher stop
```

The normal running frame rate is around 25~30fps. If the frame rate is not enough, the USB camera input frame rate may not be enough. It is recommended to face bright light or replace the USB camera. Unstable connection of USB camera will cause abnormal operation. So please keep a stable connection. The operation result are as follows:



## 2. RKNN\_DEMO Development

---

### 2.1 File Directory Introduction

The config.in is a configuration file, the rknn\_demo.mk is the basic compilation file in which copy of resource is done. Detailed commands please refer to RKNN\_DEMO\_INSTALL\_TARGET\_CMDS. The CMakeLists.txt is compiled file in the “src/” code directory. You can add your own files to compile in RKNN\_DEMO\_SRC. The rknn\_camera.c is the main file which is used to start MiniGUI main window and initialize modules. The MiniGUIMain is the main function entry. The Rknn\_ui\_show creates a main window for MiniGUI. The rknn\_demo\_init will start two threads: post and run. Run is used for capturing image and NPU processing, and sends the result to post thread which receives the processing result of NPU and doing post-processing, and outputs the result to display. The “src/rknn/ssd” is SSD related processing file. In the ssd.c, the ssd\_run function loads the model and obtains the buf of USB camera through the cameraRun function, and outputs to the registration function ssd\_camera\_callback. In the ssd\_camera\_callback function, yuv\_draw function sends video data to MiniGUI layer for RGA synthesis of video data and UI data. YUV420toRGB24\_RGA convert video data from 640480 nv12 format to 300300 rgb888 format which will be sent to the ssd\_rknn\_process function for processing. The src/ui/ssd is UI display file for SSD. The caption\_create function paints title bar and displays it in caption\_wnd\_proc; the fps\_create function paints frame rate bar and displays it in fps\_wnd\_proc; the ssd\_paint\_object paints region of detected object and the processing result of SSD is sent here for display. Detailed MiniGUI development and processing, please refer to related open source materials.

## 3. RKNN\_DEMO FAQ

---

### 3.1 How to Switch Display 720p Resolution on HDMI

```
1 [root@rk3399pro:/]# rknn_demo
2 librga:RGA_GET_VERSION:3.02,3.020000
3 ctx=0x2607c20,ctx->rgaFd=3
4 Rga built version:version:+2017-09-28 10:12:42
5 success build
6 set plane zpos = 3 (0~3)size = 3686476, g_bo.size = 4259840
7 size = 3686476, cur_bo->size = 2129920
8 size = 3686476, cur_bo->size = 2129920
9 size = 3686476, cur_bo->size = 2129920
10 NEWGAL: Video mode smaller than requested.
```

In case of the above issue, add the debug information for debugging, as below:

```
1 external/minigui$ git diff
2 diff --git a/src/newgal/video.c b/src/newgal/video.c
3 index f32197a..5641126 100644
4 --- a/src/newgal/video.c
5 +++ b/src/newgal/video.c
6 @@ -524,6 +524,8 @@ GAL_Surface * GAL_SetVideoMode (int width, int height,
7     int bpp, Uint32 flags)
8 {
9     GAL_VideoSurface = (mode != NULL) ? mode : prev_mode;
10
11     +    GAL_SetError("NEWGAL: mode->w=%d, mode->h=%d, width=%d,
12     height=%d\n",mode->w, mode->h, width, height);
13
14 Solutions as follows:
15
16 (1) How to Switch to a Different Type of Panel
17 The "/external/minigui" is selected VOP0(VOPB) for display by default.
18 Ensure that the display device (EDP/HDMI/MIPI..)
19 is placed on VOPB.
20
21 (2) How to Switch Display Resolution
22 The default resolution is 2048x1536 on RK3399PRO EVB at present. If you need
23 to switch resolution to 1280x720, the following configuration is needed:
24 rknn_demo/minigui/MiniGUI-1280x720.cfg and ui/ssd/ssd_ui.c where the
25 resolution should be changed to 1280x720.
26 Finally change package/rockchip/rknn_demo/rknn_demo.mk in the buildroot:
27
28 ```diff
29 --- a/minigui/MiniGUI-1280x720.cfg
30 +++ b/minigui/MiniGUI-1280x720.cfg
31 @@ -48,7 +48,7 @@ defaultmode=800x600-32bpp
32 #{{ifdef _MGGAL_SHADOW
33 [shadow]
34     real_engine=drmcon
35 -defaultmode=1280x720-16bpp
36 +defaultmode=720x1280-16bpp
37     rotate_screen=ccw
38 #}}
39
40 diff --git a/ui/ssd/ssd_ui.c b/ui/ssd/ssd_ui.c
```

```

36 | index 8e9884d..310e682 100644
37 | --- a/ui/ssd/ssd_ui.c
38 | +++ b/ui/ssd/ssd_ui.c
39 | @@ -15,8 +15,8 @@
40 | #define DST_W          300
41 | #define DST_H          300
42 | #if NEED_RKNNAPI
43 | -#define DISP_W          2048
44 | -#define DISP_H          1536
45 | +#define DISP_W          720
46 | +#define DISP_H          1280

```

Secondly, change the package/rockchip/rknn\_demo/rknn\_demo.mk in buildroot, as follows:

```

1 | ifeq ($(BR2_PACKAGE_RK3399PRO),y)
2 | -RKNN_DEMO_MINIGUI_CFG=minigui/MiniGUI-2048x1536.cfg
3 | +RKNN_DEMO_MINIGUI_CFG=minigui/MiniGUI-1280x720.cfg
4 | endif

```

Lastly, you need rebuild or clean the external/rknn\_demo and external/minigui projects.

```

1 | rm buildroot/output/rockchip_rk3399pro_combine/build/rknn_demo-1.0.0/ -rf
2 | rm buildroot/output/rockchip_rk3399pro_combine/build/rknn_demo-1.0.0/ -rf
3 | ./build.sh

```