Rockchip QFacialGate Instruction

ID: RK-SM-YF-374

Release Version: V1.1.1

Release Date: 2021-03-15

Security Level: □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.

Rockchip Electronics Co., Ltd.

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

Website: 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

Preface

Overview

This document is intended to introduce the usage of each module of QFicialGate application, which is based on librkfacial.so. For the specific interface usage, please refer to "Rockchip_Instruction_Rkfacial_CN.pdf". The source code and document are located in SDK/external/rkfacial.

Product Version

Platform	**Kernel Version**
RK1808, RK1806	Linux 4.4
RV1109, RV1126	Linux 4.19

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Revision History

Date	**Version**	**Author**	**Change Description**
2020-07-24	V1.0.0	CTF	Initial version
2020-08-31	V1.1.0	CTF	Add Qt configuration introduction
2021-03-15	V1.1.1	ZLX	Update product version information

Contents

1. General Introduction

1.1 Application Introduction

The QFicialGate application takes RK's own algorithm rockface through librkfacial.so to realize face detection, face feature point extraction, face recognition, and live detection process.

It includes the following functions:

- Get RGB camera image data for face recognition, and IR camera image data for live detection .
- Use SQLITE3 as a database to store facial feature values and user names.
- Use Qt to realize user registration, registration data deletion, face frame tracking, user name display and other operations.
- Use ALSA interface to realize the audio broadcast function of each process.

Note: you need RK authorization to use rockface, please refer to sdk/external/rockface/auth/README to apply for authorization. and refer to external/rkfacial/doc/Rockchip_Instruction_Rkfacial_CN.pdf for librkfacial.so usage.

1.2 Usage

QFacialGate -f num

-f: indicates the maximum number of face database supported. The default face database supports a maximum of 1000 when no configuration.

2. Qt Configuration

2.1 Configuration

- Run `make menuconfig` in the root directory to open the following configuration

```
BR2_PACKAGE_QT5=y
BR2_PACKAGE_QT5_VERSION_5_9=y
BR2_PACKAGE_QT5BASE_EXAMPLES=n        //Qt examples
BR2_PACKAGE_QT5BASE_WIDGETS=y
BR2_PACKAGE_QT5BASE_GIF=y
BR2_PACKAGE_QT5BASE_JPEG=y
BR2_PACKAGE_QT5BASE_PNG=y
BR2_PACKAGE_QT5MULTIMEDIA=y
BR2_PACKAGE_QT5QUICKCONTROLS=y
BR2_PACKAGE_QT5QUICKCONTROLS2=y
BR2_PACKAGE_QT5BASE_LINUXFB_ARGB32=y
BR2_PACKAGE_QT5BASE_USE_RGA=y         //RGA optimization, see chapter 3.4.2 for details, turn off this configuration when running Qt examples
```

```

# Fonts                              //Fonts configuration
BR2_PACKAGE_BITSTREAM_VERA=y
BR2_PACKAGE_CANTARELL=y
BR2_PACKAGE_DEJAVU=y
BR2_PACKAGE_FONT_AWESOME=y
BR2_PACKAGE_GHOSTSCRIPT_FONTS=y
BR2_PACKAGE_INCONSOLATA=y
BR2_PACKAGE_LIBERATION=y
BR2_PACKAGE_QT5BASE_FONTCONFIG=y
BR2_PACKAGE_SOURCE_HAN_SANS_CN=y
```

If you want to run the sample program that comes with Qt, you can turn on `BR2_PACKAGE_QT5BASE_EXAMPLES`, and the corresponding sample program will be generated in the `usr/lib/qt/examples` directory after compilation.

- When finishing configuration, you need to run `make savedefconfig` to save the configuration to the corresponding xxx_defconfig file in the `buildroot/configs` directory.

2.2 Build

- You can directly use `make qt5base-dirclean && make qt5base-rebuild` to build by the configuration of `make menuconfig` and `make savedefconfig`.
- If you add configuration options directly in `buildroot/configs/xxx_defconfig`, you must build with `./build.sh rootfs` for the configuration to take effect.

2.3 Run

- Take QFacialGate running as an example:

```
//Config operation display by using drm or fb api, fb is inefficient and not optimized, and both are configured to 1.
export QT_QPA_FB_DRM=1
//Display terminal configuration: linuxfb display, no rotation; set the screen rotation angle through rotation, which can be configured as: 0、90、180、270
export QT_QPA_PLATFORM=linuxfb:rotation=0
//Set the face base library to support a maximum of 30,000
QFacialGate -f 30000 &
```

3. UI Introduction

3.1 UI Controls

- Register button: register facial feature values collected by the camera to the database in real time.
- Delete button: delete the facial feature values collected by the camera from the database in real time.

- RGB/IR button: RGB/IR camera display switch button; when the button is switched to RGB, the screen displays RGB image and face detection result; when the button is switched to IR, the screen only displays IR image, without face detection result.
- Capture button: save the 30-frame image data currently displayed on the screen. The saved file is named after the current time. The RGB image is saved in the /data/rgb/ directory, and the IR image is saved in the /data/ir/ directory.
- Setting icon button: set IP address, click to pop up an IP address setting window, enter the IP address, subnet mask, and gateway address.
- Face frame: red means it is not alive, blue means it has not been registered in the database, green means it is alive and in the white-list registered in the database; black means it is alive and in the black-list registered in the database.
- Information display area at the bottom: display time, detected user information, if the device is connected to the Ethernet, it will also display the IP address, enter the IP address in the PC browser, you can log in to the web management tool, please refer to the web management tool for details: Docs/Linux/ApplicationNote/Rockchip_Instructions_Linux_Web_Configuration_CN.pdf.

3.2 Camera Image Display

- QFacialGate in the RV1109 platform is only responsible for the display of UI controls. Camera image data is directly displayed in librkfacial.so through DRM interface. Please refer to the TWO_PLANE macro control process in the code for details.
- The VOP of RK1808/1806 platform is only one layer, so the camera image data is integrated in QFacialGate through RGA and UI controls and then sent to display. Please refer to the process of ONE_PLANE macro control in the code for details.

3.3 Code Modules Introduction

3.3.1 class desktopview

- QFacialGate entry class, implement UI layout management, librkfacial.so initialization.
- `initRkfacial`

  Librkfacial.so initialization function, call set_isp_param and set_cif_param to set the corresponding camera parameters, and camera image data callback; call register_rkfacial_paint_box to register the face frame coordinate callback; call register_rkfacial_paint_info to register the user information callback.

3.3.2 class videoitem

- Realize the display of face frame, detected user information, time, and IP address. For RK1808/1806 platform, it also includes camera image data display.
- `rgaDrawImage`

  RGA synthesis function, please refer to chapter 2.4.1 of this document for details. For RGA usage, please refer to: external/linux-rga/Linux rga instructions.pdf.
- `drawBox`

  Draw face frame
- `drawSnapshot`

When a liveness registered in the database is detected, the photo of the liveness in the database is synthesized using RGA.

- `drawInfoBox`

  Draw the bottom information display area, including time and IP address display; when a liveness registered to the database is detected, the user name and the photo of the living body in the database will also be displayed.

3.3.3 class snapshotthread

- Get the images of the liveness in the database, get the image information by calling turbojpeg_decode_get, and release the resources by turbojpeg_decode_put.

3.3.4 class savethread

- Save the image data currently displayed on the screen, click the Capture button to start saving, and automatically stop after saving 30 frames.

3.3.5 class qtkeyboard

- Customize keyboard. Support 0~9, 26 letters (both upper and lower case), delete, space, slash and other common symbol keys. The keyboard layout is located in qtkeyboard.ui

3.4 Performance Optimization

3.4.1 QFacialGate Optimization

- Use RGA synthesis instead of directly using drawRect and drawImage of Qt, including:
  1. Synthesize the semi-transparent shadow frame in the bottom information display area.
  2. When a living registered in the database is detected, it is also used to synthesize the living images in the database.
  3. For RK1808/1806 platform also includes camera image data synthesis.
- Comparison test display: RGA synthesis reduces CPU usage, and the frame rate increases significantly. If UI has a similar large area shadow or image data display, please refer to the rgaDrawImage api in videoItem.cpp to use RGA synthesis.

3.4.2 Qt Optimization

- Use RGA synthesis to optimize drawImage, increase frame rate and reduce CPU, and controlled by BR2_PACKAGE_QT5BASE_USE_RGA macro. This macro must be turned on, otherwise the frame rate will drop severely and the image will be noticeably stuck.
- UI data is drawn directly to Linuxfb buffer, skipping blacking and two neon synthesis, further reducing CPU, controlled by the BR2_PACKAGE_QT5BASE_LINUXFB_DIRECT_PAINTING macro, this optimization is only effective in a single window. If UI uses multi-window display, you can turn off this macro, but CPU usage will increase slightly, which may cause a small decrease in frame rate.

- All of the above macro switches can run `make menuconfig` configuration in the root directory. After modification, you need `make savedefconfig` to save the configuration, and run `make qt5base-dirclean && make qt5base-rebuild` to recompile Qt, and run `make QFacialGate-dirclean && QFacialGate-rebuild` to recompile QFacialGate to make the configuration take effect.