

Rockchip RK3588 Linux Edge SDK Quick Start

ID: RK-JC-YF-856

Release Version: V1.1.0

Release Date: 2022-05-10

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.

All rights reserved. ©2022. 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

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

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

The document presents the basic usage of Rockchip RK3588 EDGE linux SDK, aiming to help developers get started with RK3588 EDGE Linux SDK faster.

Product Version

Chipset	Kernel Version
RK3588/RK3588s	5.10

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Revision History

Version	Author	Date	Change Description
V1.0.0	Addy Ke	2022-01-07	Initial version
V1.0.1	Addy Ke	2022-01-25	Add configuration: user password
V1.0.2	Addy Ke	2022-03-25	Modify configuration items, added flash and debug instruction

Contents

Rockchip RK3588 Linux Edge SDK Quick Start

1. Preface
2. Set up a Development Environment
 - 2.1 System Requirement of Compile Host
 - 2.2 Install Compilation Rely Software
 - 2.2.1 Install Compilation Rely Software on Debian
 - 2.2.2 Install Compilation Rely Software on Ubuntu
 - 2.3 How to Get the SDK
3. The Configuration of Compilation
 - 3.1 Add New Configuration
 - 3.1.1 Added Configurations of RK3588 Products
 - 3.1.2 Added Configurations of RK3588S Products
 - 3.2 Setting Configuration Information
 - 3.3 Configuration Information
 - 3.4 Configuration Information Introduction
4. Image Build
 - 4.1 Automatic Build
 - 4.2 Generate Partition File
 - 4.3 Uboot Build
 - 4.4 Kernel Build
 - 4.5 Rootfs Build
 - 4.6 Update Image Package
 - 4.7 View Build Help
5. Flash Image
 - 5.1 Enter Flash Mode
 - 5.1.1 Enter Loader Flash Mode
 - 5.1.2 Enter Maskrom Flash Mode
 - 5.2 Check Flash Status
 - 5.2.1 Check Flash Status by Linux Host
 - 5.2.2 Check Flash Status by Windows Host
 - 5.3 Linux Flash Images Introduction
 - 5.3.1 Flash all Images
 - 5.3.2 Flash Uboot Image
 - 5.3.3 Flash Kernel Image
 - 5.3.4 Flash Rootfs Image
 - 5.3.5 Flash Help
 - 5.4 Windows Flash Images Introduction
6. Debug
 - 6.1 Serial Port Connection
 - 6.2 Windows Debug
 - 6.2.1 Get Port
 - 6.2.2 Port Setting
 - 6.3 Linux Debug
 - 6.3.1 Install Minicom
 - 6.3.2 Port Setting
 - 6.3.3 Running Minicom
7. Boot from SD Card
 - 7.1 Create SD Boot Card
 - 7.2 Boot From SD Card
8. Flash Image From SD Card
 - 8.1 Automatic Flash
 - 8.2 Create Partition
 - 8.3 Flash Kernel Image
 - 8.4 Flash Rootfs Image
 - 8.5 Help File

1. Preface

For documentation purposes, define the variables as follow:

- BOARD: development board/product model name; for example: TB-RK3588X0 development board model name is TB-RK3588X0.
- DTB: Kernel device tree; for example: TB-RK3588X0 development board kernel device tree is rk3588-toybrick-x0-linux.
- ROOT_DIR: TB-RK3588X Linux BSP work directory; the directory shown in the document is /home/toybrick/work/edge.
- CHIP: the chip model of development board; TB-RK3588X Linux BSP support RK3588 and RK3568.
- OUT_DIR: the path of images \${ROOT_DIR}/out/\${CHIP}/\${BOARD}/images, for example TB-RK3588X0 images path is /home/toybrick/work/edge/out/rk3588/TB-RK3588X0/images.

2. Set up a Development Environment

2.1 System Requirement of Compile Host

- It is recommend use Ubuntu18.04 or higher version and Debian11 to development, and the recommend memory greater than 16GB.
- The user name can't contain Chinese characters.
- please set up the development environment only as a common user, not as the root user.

2.2 Install Compilation Rely Software

2.2.1 Install Compilation Rely Software on Debian

```
sudo apt -y install python lz4 coreutils qemu qemu-user-static python3 \
devicetree-compiler clang bison flex lld libssl-dev bc genext2fs git make
```

2.2.2 Install Compilation Rely Software on Ubuntu

```
sudo apt -y install python lz4 coreutils qemu qemu-user-static python3 \
device-tree-compiler clang bison flex lld libssl-dev bc genext2fs git make
```

2.3 How to Get the SDK

- Get the RK3588_EDGE_SDK from FAE, and extract it into the working directory.
- Get the compressed image file of the native Debian11 Gnome desktop (debian11-gnome.tar.gz) from the FAE, and copy it to: \${ROOT_DIR}/rootfs/images/aarch64/ (Note: there is no need to decompress)

3. The Configuration of Compilation

3.1 Add New Configuration

The EDGE SDK support RK3588-EVB1, RK3588-EVB4, RK3588-IR88MX01, RK3588s-EVB1 and TB-RK3588X0 development boards by default. If you are developing on these boards and do not need to add new configurations, please skip this section.

3.1.1 Added Configurations of RK3588 Products

For example the new development board model name is RK3588-MY-PRODUCT, and the kernel device tree is rk3588-my-product-linux, the procedure is as follows:

1. Copy RK3588-EVB1 to RK3588-MY-PRODUCT:

```
cp -r vendor/rk3588/RK3588-EVB1 vendor/rk3588/RK3588-MY-PRODUCT
```

2. Modified `vendor/rk3588/RK3588-MY-PRODUCT/config.json` as follow:
 - board: modify the value of board into RK3588-MY-PRODUCT.
 - dtbname: modify the value of dtbname in kernel subset into rk3588-my-product-linux.
 - Other configuration: please refer to "Configuration information introduction" .

3.1.2 Added Configurations of RK3588S Products

For example the new development board model name is RK3588s-MY-PRODUCT, and the kernel device tree is rk3588s-my-product-linux, the procedure is as follows:

1. Copy RK3588-EVB1 to RK3588s-MY-PRODUCT:

```
cp -r vendor/rk3588/RK3588-EVB1 vendor/rk3588/RK3588s-MY-PRODUCT
```

2. Modified `vendor/rk3588/RK3588s-MY-PRODUCT/config.json` as follow:
 - board: modify the value of board into RK3588s-MY-PRODUCT.
 - dtbname: modify the value of dtbname in kernel subset into rk3588s-my-product-linux.
 - chip: don't need to modify.(RK3588 and RK3588s both set as rk3588)
 - Other configuration: please refer to "Configuration information introduction" .

3.2 Setting Configuration Information

Run the following command and enter the product model number (for example, the serial number of the RK3588 EVB1 development board is 1) to set the configuration information.

```
./edge set

[EDGE DEBUG] Board list:
> rk3588
0. RK3588-IR88MX01
1. RK3588-EVB1
2. TB-RK3588X0
3. RK3588s-EVB1
4. RK3588-EVB4
5. RK3588-MY-PRODUCT
> rk3568
6. TB-RK3568X0
Enter the number of the board: 1
```

Note: Please run this command again to update the configuration ,when you update or modify the code related to edge SDK.

3.3 Configuration Information

Run the following command to check the current configuration information.

```
./edge env

[EDGE DEBUG] board: RK3588-EVB1
[EDGE DEBUG] chip: rk3588
[EDGE DEBUG] arch: arm64
[EDGE DEBUG] > Partition:
[EDGE DEBUG]   vnm: ['0x00000000', '0x00000000']
[EDGE DEBUG]   uboot: ['0x00002000', '0x00002000']
[EDGE DEBUG]   trust: ['0x00004000', '0x00002000']
[EDGE DEBUG]   resource: ['0x00006000', '0x00002000']
[EDGE DEBUG]   boot_linux:bootable: ['0x00008000', '0x00030000']
[EDGE DEBUG]   rootfs:grow: ['0x00038000', '-']
[EDGE DEBUG] > Uboot:
[EDGE DEBUG]   config: rk3588-edge
[EDGE DEBUG] > Kernel:
[EDGE DEBUG]   version: 5.10
[EDGE DEBUG]   uuid: a2d37d82-51e0-420d-83f5-470db993dd35
[EDGE DEBUG]   config: rk3588_edge.config
[EDGE DEBUG]   dtbname: rk3588-evb1-lp4-v10-linux
[EDGE DEBUG]   size: 72
[EDGE DEBUG]   initrd: True
[EDGE DEBUG]   docker: False
[EDGE DEBUG]   debug: 0xfeb50000
[EDGE DEBUG] > Rootfs:
[EDGE DEBUG]   osname: debian
[EDGE DEBUG]   version: 11
[EDGE DEBUG]   type: gnome
```

```
[EDGE DEBUG]    uuid: 614e0000-0000-4b53-8000-1d28000054a9
[EDGE DEBUG]    size: auto
[EDGE DEBUG]    user: rockchip
[EDGE DEBUG]    password: rockchip
[EDGE DEBUG]    relver: 0.1.0
```

3.4 Configuration Information Introduction

- Common configuration is saved in vendor/common/config.json.
- Board configuration is saved in vendor/\${CHIP}/\${BOARD}/config.json, and the value will override the value of the same configuration item in the common configuration.
- The value of "not set" configuration items must be set in the "board level" ; Other configuration items can be modified in "board Configuration" as required.
- The chip value of kernel subset in configuration information of RK3588 and RK3588s both are set to rk3588.

The configuration include the following items:

1. System configuration: EDGE SDK configuration information

Configuration Item	Description	Default Value	Note
board	Development board or products model name	not set	
chip	Chip type	not set	
arch	Chip architecture	not set	

Notice: The value of board must be same as \${BOARD} directory name in vendor/\${CHIP} directory.

2. partition: Partition information, include partition name, start address and partition size (the unit of the start address and partition size is block, each block size is 512 bytes).

Configuration Item	Description	Start Address	Partition Size	note
vnvm	Vnvm partition, used to save vendor information	0x00000000	0x00000000	only valid for flash
uboot	Uboot partition	0x00002000	0x00002000	required
trust	Trust partition	0x00004000	0x00004000	optional, the value can be set as 0
resource	Resource partition, used to save startup loading LOGO	0x00006000	0x00002000	required
boot_linux	Kernel partition	0x00008000	0x00030000	required
rootfs	Root file system	0x00038000	-	all remaining storage space, and it is required

Note: If start address and partition address both are 0, then script will ignores this partition.

3. uboot: Uboot configuration information

Configuration Item	Description	Default Value	Note
config	uboot compilation configuration	rk3588-edge	configs/rk3588-edge.config

4. kernel: Kernel configuration information

Configuration Item	Description	Default Value	Note
version	Kernel version	5.10	Currently, only 5.10 kernels are supported
uuid	The PARTUUID value of kernel partition	a2d37d82-51e0-420d-83f5-470db993dd35	It is can't be modified
config	Kernel compilation configuration	rk3588_edge.config	Path: arch/arm64/configs/rk3588_edge.config
size	Kernel image size,unit: M	72	Auto: automatically adjusts the size
dtbname	Kernel device tree file name	not set	Not include dts
initrd	Whether to load initrd.img	true	Start slowly
docker	Whether the kernel configuration needs to support Docker	false	Kernel image will become larger
debug	Chip debug port address	not set	According to chip configuration

Description:

- *Initrd can push the system into emergency repair mode. More information please refer to "Rockchip_Developer_Guide_Linux_Edge_Debian_CN.pdf"*
- *The kernel compilation configuration will load rockchip_linux_defconfig and the configuration specified by the config by default.*

5. rootfs: rootfs configuration information

Configuration Item	Description	Default Value	Note
osname	OS name	debian	
version	OS version	11	
type	Desktop type, support server, gnome and lxde	gnome	
apturl	The URL base address of APT source of edge system	http://repo.rock-chips.com/	Debian source address: \${apturl}/edge/debian
uuid	The PARTUUID value of rootfs	614e0000-0000-4b53-8000-1d28000054a9	It is can't be modified
size	rootfs image size, unit: M	auto	Auto: automatically adjusts the size
user	OS administrator user name	not set	
password	Administrator password	not set	
relver	Release version	0.1.0	The value will be written in: /etc/edge-release

Description: The following are the supported desktop type.

- server: It is service system and no desktop.
- gnome: It is native gnome desktop system which based on wayland.
- gnome-docker: It is native gnome desktop system which based on wayland, docker related components are preinstalled.
- gnome-ros1: It is native gnome desktop system which based on wayland, ros1 related components are preinstalled.
- gnome-ros2: It is native gnome desktop system which based on wayland, ros2(ros2-foxy) related components are preinstalled.
- lxde: It is LXDE Lightweight desktop system which based on xorg.

4. Image Build

4.1 Automatic Build

Execute the following commands to automatically complete all build, package update.img and save it in OUT_DIR directory.

```
./edge build -a
```

Note: The P parameter indicates that patches are created before compilation

4.2 Generate Partition File

Execute the following command to generate parameter.txt and saved it in OUT_DIR directory.

```
./edge build -p
```

4.3 Uboot Build

Execute the following command to generate MiniLoaderAll.bin and uboot.img, and saved them in OUT_DIR directory.

```
./edge build -u
```

4.4 Kernel Build

Execute the following command to generate boot_linux.img and resource.img, and saved them in OUT_DIR directory.

```
./edge build -k
```

4.5 Rootfs Build

Execute the following command to generate rootfs.img, and saved it in OUT_DIR directory.

```
./edge build -r
```

1. During rootfs image compilation, the script will check whether rootfs.img exists in the OUT_DIR directory. If rootfs.img exists, the following information is displayed.

[EDGE INFO]: Rootfs already exists, do you want to continue ?[Y/n]

- If you need to delete the old rootfs image and recompile it, please enter N or n.
- If you want to continue with the rootfs image you compiled earlier, please enter Y or y.

2. During rootfs image compilation, the script will call `vendor/common/pre-install/install.sh` and `vendor/${CHIP}/${BOARD}/pre-install/install-board.sh` scripts to preinstall the software package and modify system configuration information. If you need additional software package or modify system configuration information, please modify `vendor/${CHIP}/${BOARD}/pre-install/install-board.sh`.

4.6 Update Image Package

Execute the following command to package all build, generate update.img, and save it in OUT_DIR directory.

```
./edge build -U
```

4.7 View Build Help

Execute the following command to view supported build parameters.

```
./edge build -h
```

5. Flash Image

5.1 Enter Flash Mode

5.1.1 Enter Loader Flash Mode

1. Connecting PC USB Host and development board Type-C with Type-C cable, long press V+/Recovery button.
2. The development board with 12V power supply. If it is powered on, press the reset button.
3. When development board entered loader mode, release the button.
4. Please refer to "Check the flash status" section to check whether the development board is enter loader mode.

5.1.2 Enter Maskrom Flash Mode

1. Connecting PC USB Host and development board Type-C with Type-C cable, long press Maskrom button.
2. The development board with 12V power supply. If it is powered on, press the reset button.
3. When development board entered loader mode, release the button.
4. Please refer to "Check the flash status" section to check whether the development board is enter Maskrom mode.

5.2 Check Flash Status

5.2.1 Check Flash Status by Linux Host

Execute the following command to check flash status.

```
./edge flash -q
```

- none: Indicating that the development board is not in flash mode.
- loader: Indicating that the development board entered loader mode.
- maskrom: Indicating that the development board entered maskrom mode.

5.2.2 Check Flash Status by Windows Host

Open RKDevTool.exe in tools\RKDevTool_Release_v2.84 directory.

- No device be found (as shown in Figure 1-1): Indicating that the development board is not in flash mode.
- Found a LOADER device (as shown in Figure 1-2): Indicating that the development board entered loader mode.
- Found a MASKROM device (as shown in Figure 1-3): Indicating that the development board entered maskrom mode.

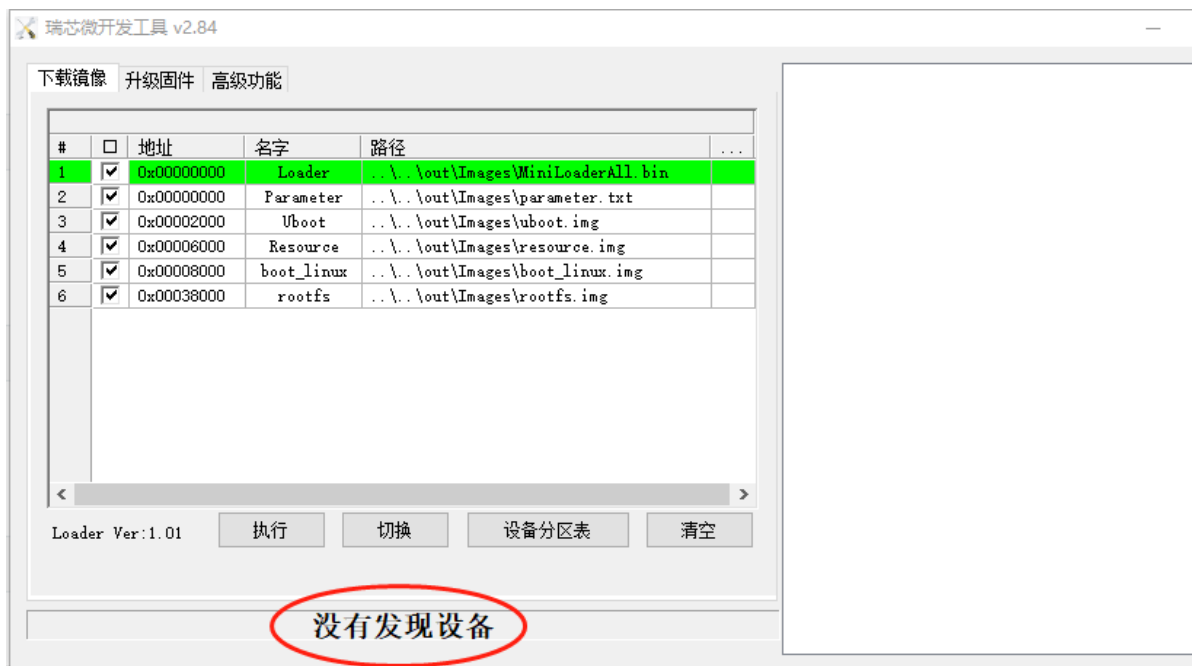


Figure 1-1: No device be found

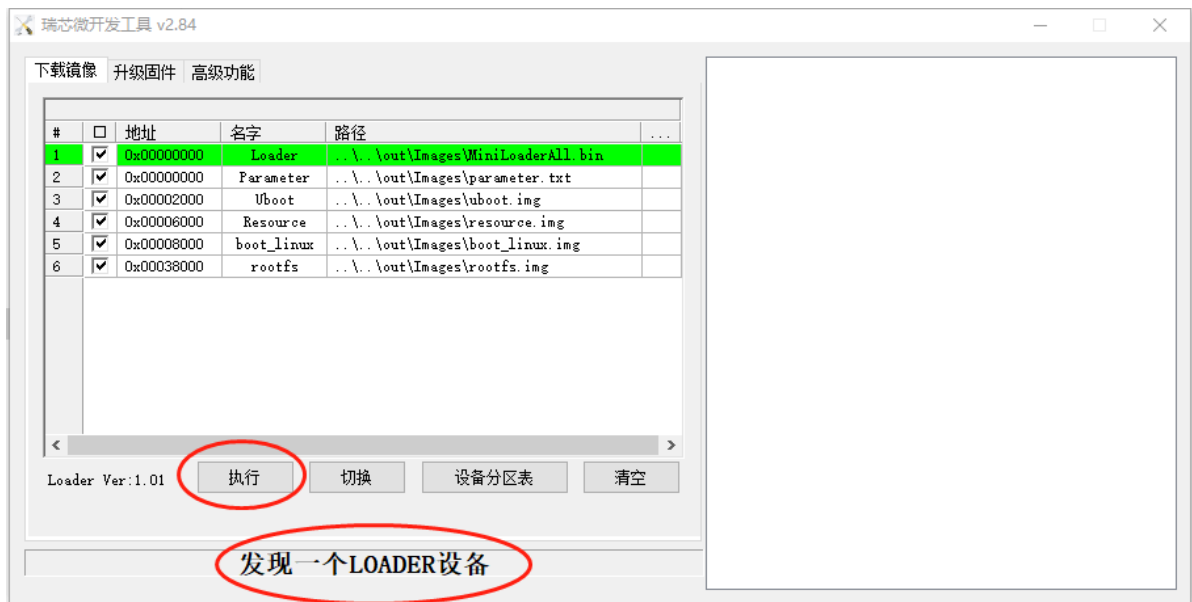


Figure 1-2: Found a LOADER device

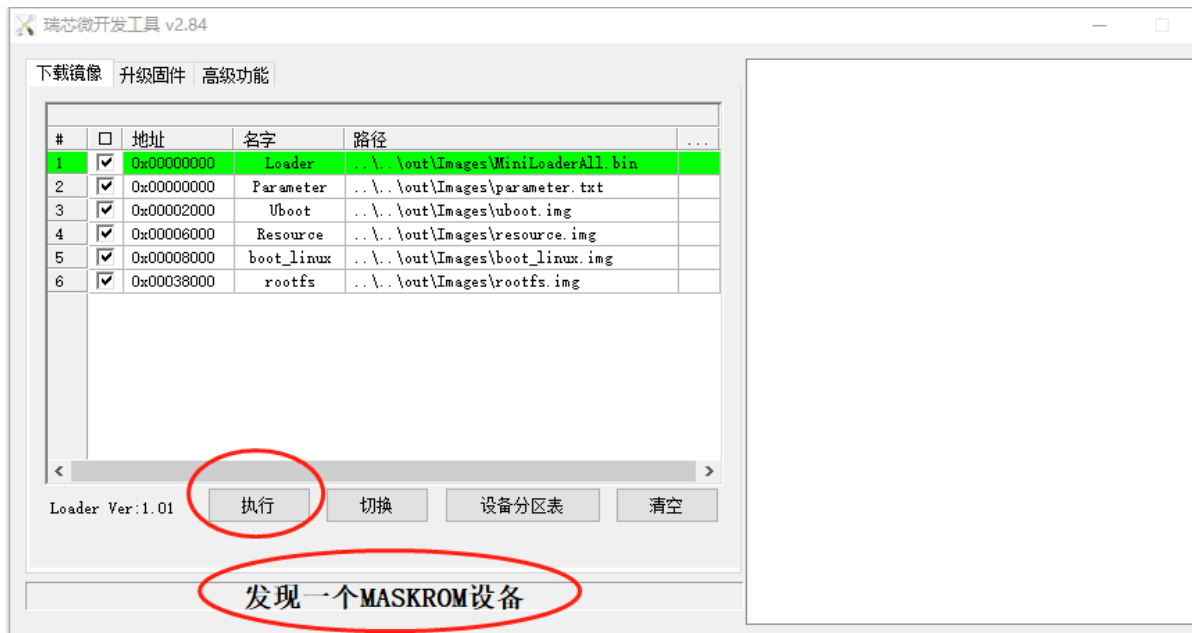


Figure 1-2: Found a MASKROM device

5.3 Linux Flash Images Introduction

5.3.1 Flash all Images

Flash those images: MiniLoaderAll.bin, uboot.img, resource.img, boot_linux.img, rootfs.img and parameter.txt

```
./edge flash -a
```

5.3.2 Flash Uboot Image

Flash those images: MiniLoaderAll.bin, uboot.img and parameter.txt

```
./edge flash -u
```

5.3.3 Flash Kernel Image

Flash those images: boot_linux.img and resource.img

```
./edge flash -k
```

5.3.4 Flash Rootfs Image

```
./edge flash -r
```

5.3.5 Flash Help

Execute the following command to view flash help.

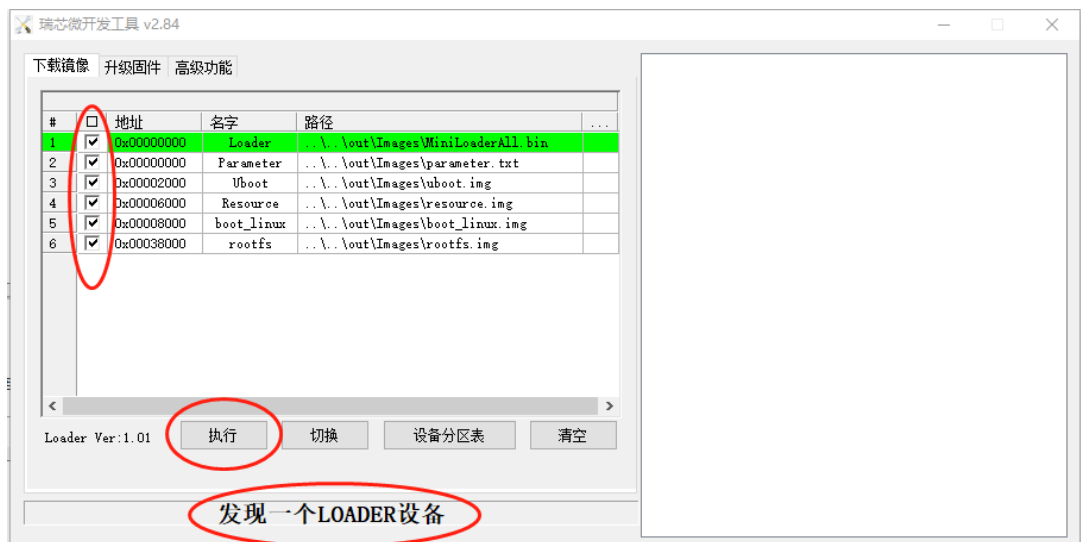
```
./edge flash -h
```

5.4 Windows Flash Images Introduction

1. Open RKDevTool.exe in tools\RKDevTool_Release_v2.84 directory.
2. Verify development board has entered loader or maskrom mode.
3. Select the needed images to flash

Notice: Loader and Parmeter are recommended, other options selected as needed.

4. click "Execute" button to start flash images. (as shown in figure 2-1).



6. Debug

6.1 Serial Port Connection

Connecting PC USB host port and development board debug port with USB cable, usually the debug port on development board has “DEBUG” or "UART TO USB" silkscreen.

6.2 Windows Debug

6.2.1 Get Port

Open Windows Device Manager to get the port of USB serial port, as shown in figure 3-1:

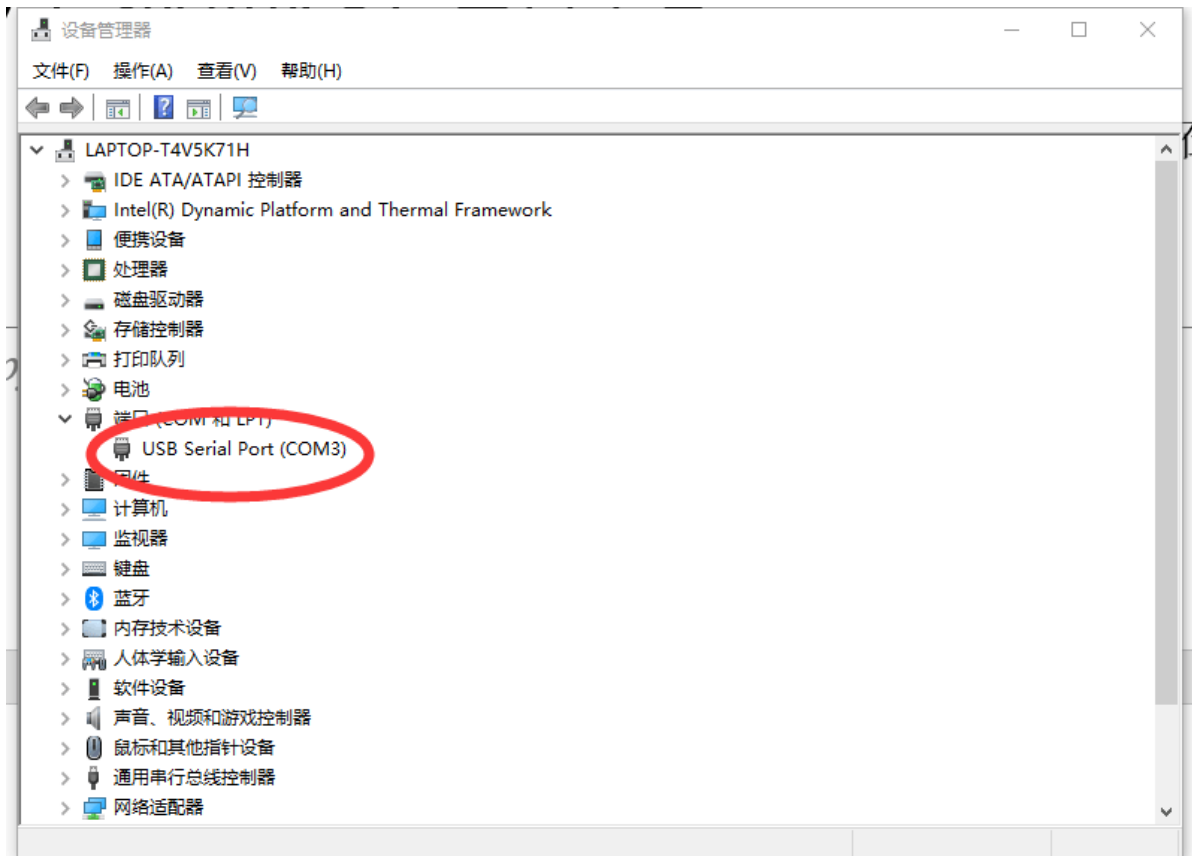


Figure 3-1: Get serial port

6.2.2 Port Setting

Open SecureCRT, click "Quick Connect" button to configure serial, as shown in figure 3-2 and figure 3-3:

1. Port: Set the port number what we get from Windows Device Manager.
2. Baud rate: Should be set 1500000
3. Disable flow control: Do not select RTS/CTS

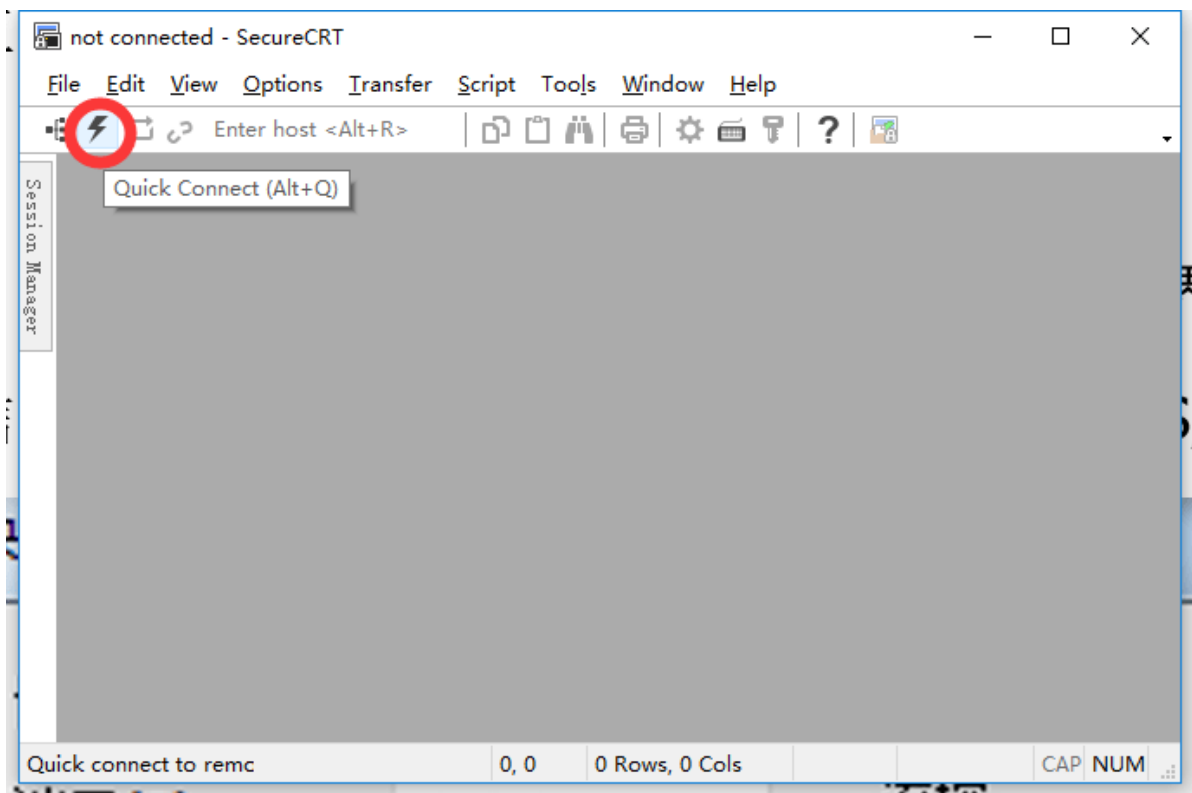


Figure 3-2: SecureCRT

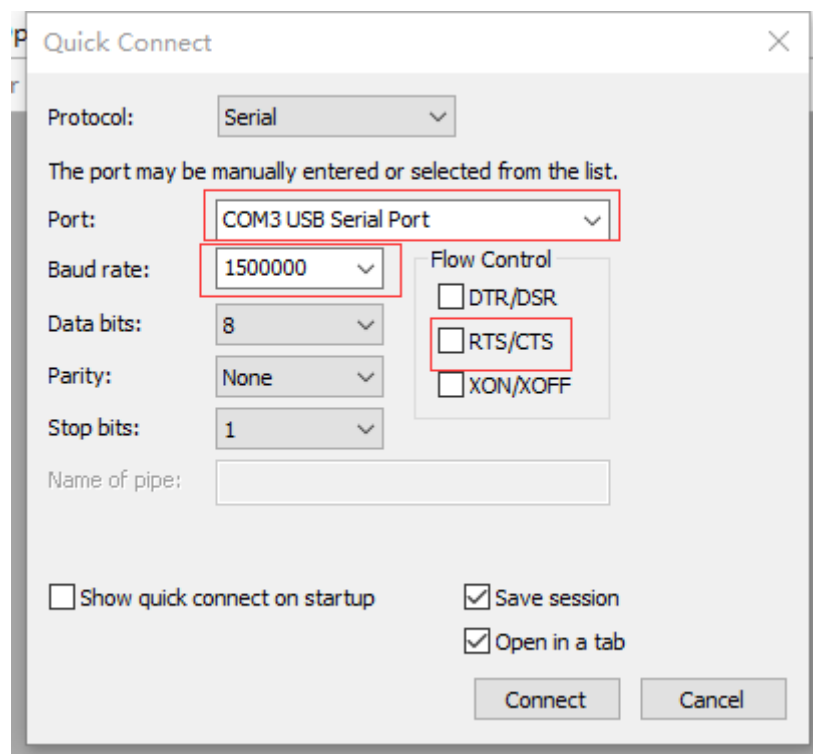


Figure3-3: port setting

6.3 Linux Debug

6.3.1 Install Minicom

```
sudo apt -y install minicom
```

6.3.2 Port Setting

According to the following steps to configure the serial port information (execute the steps for the first time).

1. Execute the following command to open minicom.

```
sudo minicom -s
```

2. Press CTRL-A Z, enter serial setting interface.

```
+-----+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup       |
| Modem and dialing       |
| Screen and keyboard     |
| Save setup as dfl       |
| Save setup as           |
| Exit                    |
+-----+
```

3. Port setting: select "Serial port setup".
4. Port device setting: Press "A", input "/dev/ttyUSB0", and then press "ENTER".
5. Disable flow control: Press "F", and then press "ENTER".
6. Baud rate setting: Press "E", and then press "A" to switch baud rate, until it shows Current 1500000 8N1", then press "ENTER".
7. Port setting as shown bellow:

```
+-----+
| A - Serial Device       : /dev/ttyUSB0 |
| B - Lockfile Location   : /var/lock   |
| C - Callin Program      :             |
| D - Callout Program     :             |
| E - Bps/Par/Bits        : 1500000 8N1 |
| F - Hardware Flow Control : No        |
| G - Software Flow Control : No        |
+-----+
```

8. Exit port setting: Press "ENTER" to return.
9. Save configuration: Select "Save setup as dfl".
10. Exit setting: Select "Exit".

6.3.3 Running Minicom

```
sudo minicom
```

7. Boot from SD Card

7.1 Create SD Boot Card

1. Prepare a microSD card and the capacity not less than 16GB. Insert microSD card into USB card reader and then into USB port of host computer.
2. Run SD_Firmware_Tool by double click the `tools\SDDiskTool_v1.69\SD_Firmware_Tool.exe`, as shown in figure 4-1:
Select correct disk device, function mode and upgrad.img path, then click "Create" button to make it.

figure 4-1:SDDisk interface

1. If the development board have FLASH/EMMC, you must flash miniloaderall. bin and uboot.img in FLASH/EMMC by flash tool firstly.

Note: RK3588 preferentially loads firmware from FLASH/EMMC. If there is no FLASH/EMMC on the board, the system will boot from the SD card.

7.2 Boot From SD Card

Insert microSD card into SD card slot of RK3588 development board and reboot the board, then it will boot from the SD card.

8. Flash Image From SD Card

8.1 Automatic Flash

Execute the following command to create two partitions, and write kernel image and rootfs image to these two partitions.

```
./edge sdcard -a
```

8.2 Create Partition

Insert microSD card into SD card reader, then execute the following command to create two partitions, and the first partition size is 100M.

```
./edge sdcard -n
```

8.3 Flash Kernel Image

Execute the following command to flash the kernel image to the first partition of the SD card.

```
./edge sdcard -k
```

8.4 Flash Rootfs Image

Execute the following command to flash the rootfs image to the second partition of the SD card.

```
./edge sdcard -r
```

8.5 Help File

Execute the following command to view the supported command parameters.

```
./edge sdcard -h
```

9. FAQ

- **Question:** Error information is displayed when building rootfs: `chroot: failed to run command '/pre-install/install.sh': Exec format error.`

Solution: Install qemu, qemu-user-static

- **Question:** During the process of compiling rootfs image, the system is abnormal or you press CTRL+C to forcibly exit. As a result, the images and device nodes are not uninstalled.
- **Solution:** Execute the following command to mandatory uninstall it.

```
./build/sripts/chroot.sh out/Images out
```