IAC-RK3588-Kit User Manual

Ver.#：1.0

2024.04

Version Record

Version	Hardware Platform	Description	Date	Revisor
1.0	IAC-RK3588-MB-V1.00	Initial version, first release	2024-04	wwx

Contents

ATTN: This is the user manual for IAC-RK3588-Kit development kit

Ⅰ.Preface

Company Profile

Zhejiang Qiyang Intelligent Technology Co., Ltd., established in 2007, which locates in Hangzhou, Zhejiang, PRC. It is a high-end technological enterprise that specializes in exploitation, fabrication, and selling embedded computer mainboards . With 10 years of experiences, Qiyang has established the completed service chain from the design concept to mass production successfully.

The R&D team is organized by 30 more technical engineers. Qiyang focus on providing functional embedded hardware, software tool and customization solutions. It has been applied to Industrial Control, Internet of Things, New Retail, Smart Medical, Electricity Device, Environmental Surveillance, Charging Pile etc.

With the growth of the business, Qiyang has set up an SMT factory in Zhuji, Zhejiang province, which is 5000 ㎡, with a 2xSMT production line. The SMT factory performs the ISO9001 Quality Management System strictly. Relying on the solid production ability, the SMT factory's annual capacity is about a million sets, which totally guarantee the delivery date.

Qiyang has a thorough sales marketing network, professional sales ,and after-sales team to provide full technical support and service. The business has spread over 120 countries and areas, it helps the clients to introduce the products into the market efficiently and successfully.The combination and extension of research and development, production capacity, and market, that provide a solid foundation for Qiyang to provide specialized, globalized embedded hardware and software.

We offer:

l.Software/Hardware Mainboard

Based on the CPU solution from NXP,Rockchip,MTK,Renesas,TI,Atmel,Cirrus Logic etc, Qiyang provides the ARM development kit/system on module/industrial board and periphery products, paired tools and software for the user do further exploitation.

2.Customization Service

Fully taking the advantage of the technical accumulation on the ARM platform and Linux,

Android, Ubuntu OS, Qiyang provides the efficient OEM/ODM service to the users.

Sincerely thanks for using Qiyang's product, we will try our best to offer you the technical supports!

II. Tutorials

2.1.Guidance

IAC-RK3588-Kit provides a set of accessories as below:

No.	List		Qty
1	SOM Module	IAC-RK3588-CM	1
2	Carrier	IAC-RK3588-MB	1
4	Serial Cable	3PIN2.0 spacing with socket	3
5	USB cable	Type-c USB downloading cable	2
7	Power	DC 12V/2.5A	1

You can purchase a 7-inch LCD and accessories additionally.

You will require the below accessories while testing:

Network：100M/1000M LAN cable, and wired router;

Display: A monitor with HDMI port, and HDMI cable;

Others: USB flash disk, USB mouse, TF card ,etc.

IAC-RK3588-Kit supports eMMC boot, the board has been preloaded the firmware, it can be tested directly.

To learn more about the connector details, please refer to ***IAC-RK3588-Kit Hardware Manual V1.0.pdf***

2.2.UART DEBUG

If it is in the pre-stage testing or kernel development stage, UART DEBUG is very useful to check the system boot log, especially, the device is without GUI.

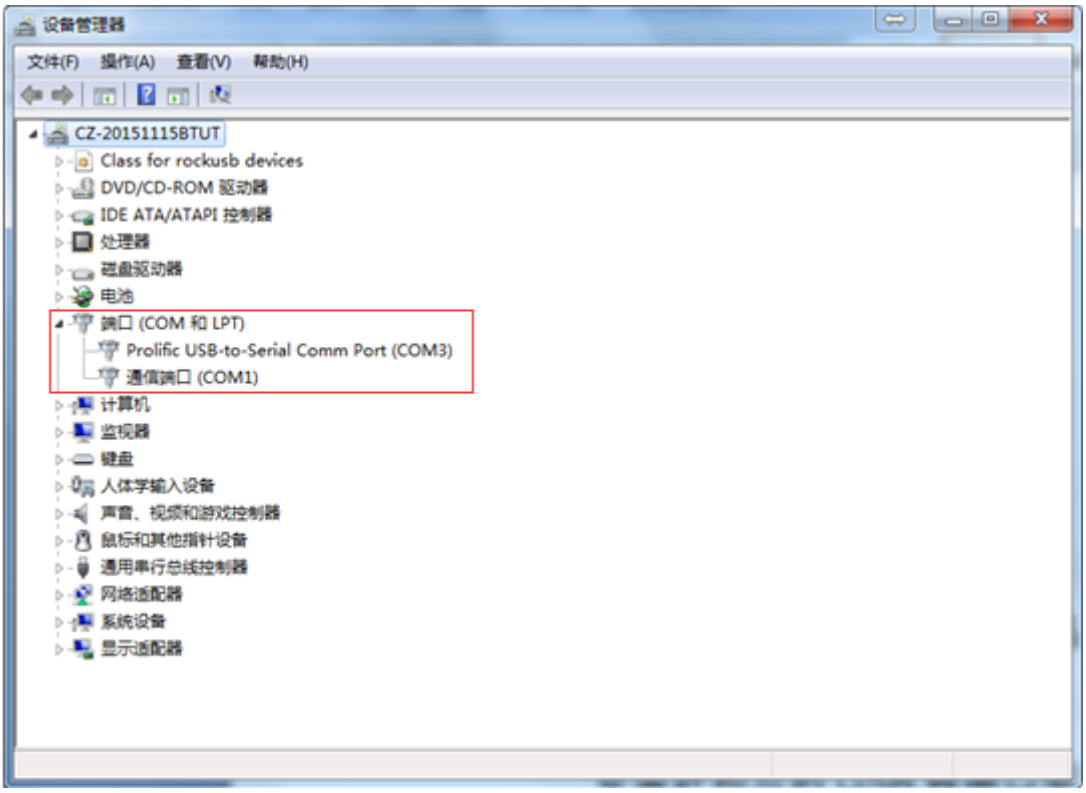
Hardware Connection:

Normally, a PC with Windows 10 has the ch340 driver, if there is not a such driver in your PC, please download it first.

UART connection: To connect the DEBUG UART (J2) to host machine via Type-c cable.

Driver Installation:

After installing the driver, please insert a Type-c cable to power on the board. The system hints the hardware starts initializing. After that, the COM port could be found in the device manager, as below picture shown:



As the above picture shown, COM1 is the UART which the PC owns, COM3 is the UART on development kit.

Use UART DEUG on Windows

Normally, Putty, SecureCRT or other UART tools are used on Windows. The specific operation method is easy to find in Internet.

Here, we take an example of SecureCRT：

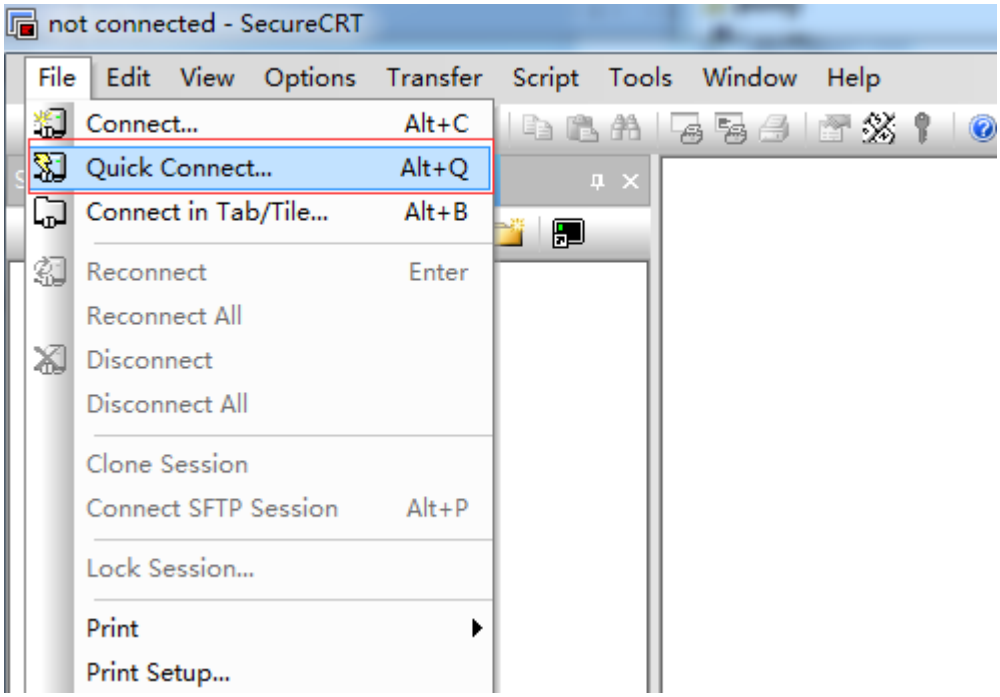
1. Open File-Quick Connect;

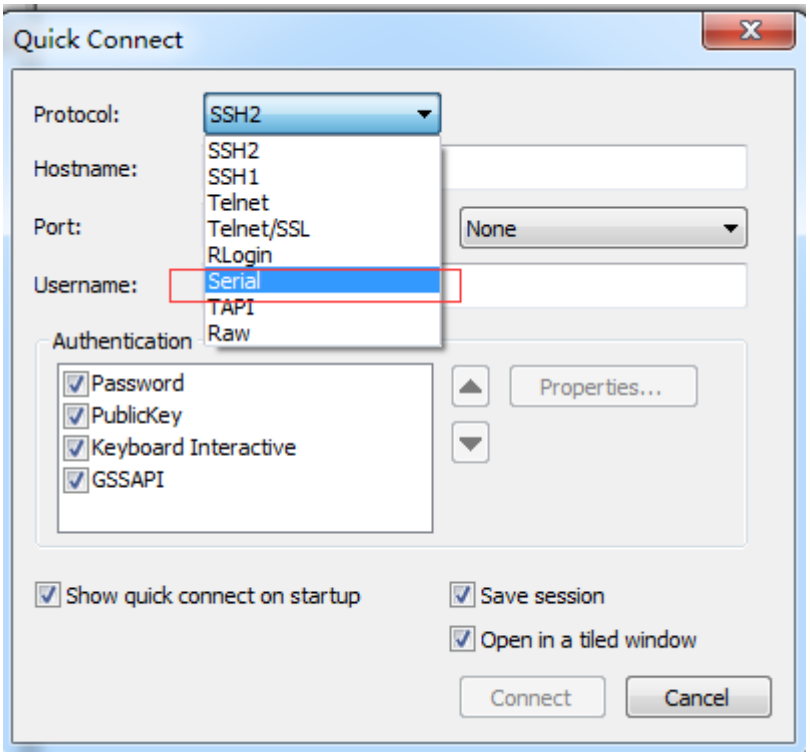
2. Select Protocol: Serial

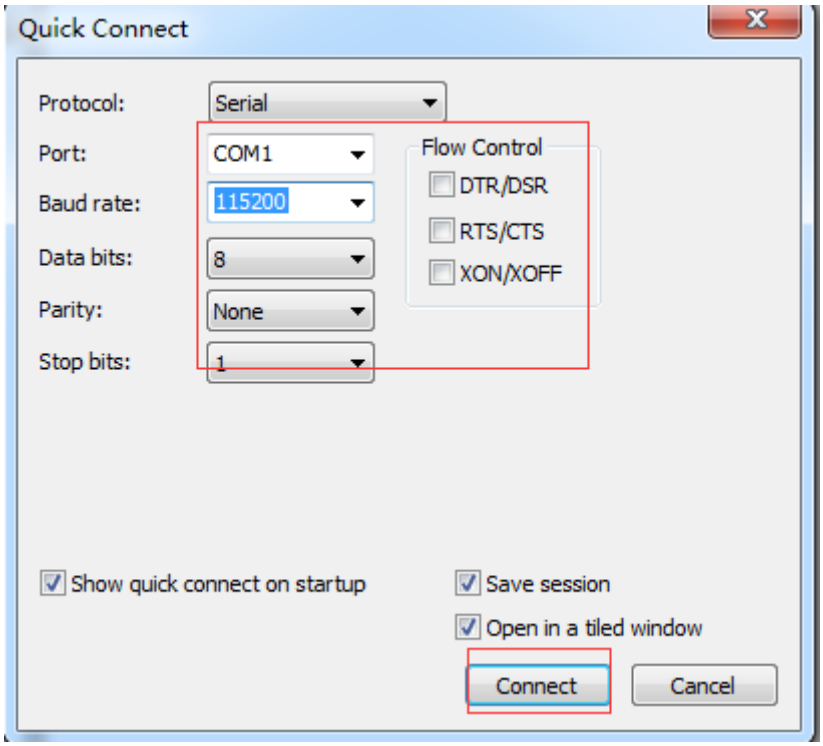
3. Modify the port as the COM port which you can find from device manager;

4.Baud Rate:115200, Data Bits:8, Stop Bits:1, Parity: None, Flow Control: None ;

5.Click ***connect***.







```
         Starting Advanced IEEE 802_/WPA/WPA2/EAP Authenticator...
[FAILED] Failed to start Advanced I_1X/WPA/WPA2/EAP Authenticator.
See 'systemctl status hostapd.service' for details.
         Starting Bluetooth service...
[  OK  ] Started Bluetooth service.
[   11.626449] ttyFIQ ttyFIQ0: tty_port_close_start: tty->count = 1 port count = 2

Debian GNU/Linux 10 linaro-alip ttyFIQ0

linaro-alip login: root (automatic login)

Last login: Wed Aug 24 08:19:43 UTC 2022 on ttyFIQ0
Linux linaro-alip 4.19.219 #207 SMP Wed Aug 24 10:13:56 CST 2022 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@linaro-alip:~# [   13.102317] EXT4-fs (mmcblk0p8): mounting ext2 file system using t
he ext4 subsystem
[   13.104008] EXT4-fs (mmcblk0p8): warning: mounting unchecked fs, running e2fsck is rec
ommended
[   13.104751] EXT4-fs (mmcblk0p8): mounted filesystem without journal. Opts: (null)
[   13.281045] EXT4-fs (mmcblk0p7): mounting ext2 file system using the ext4 subsystem
[   13.283373] EXT4-fs (mmcblk0p7): warning: mounting unchecked fs, running e2fsck is rec
ommended
[   13.284119] EXT4-fs (mmcblk0p7): mounted filesystem without journal. Opts: (null)

root@linaro-alip:~#
```

Use UART DEBUG on Ubuntu:

Normally, minicom, picocom or other UART tools will be used on the Ubuntu. The specific operation method can be easily found in Internet.

III. Firmware upgradation

3.1. OS version

Current OS version：

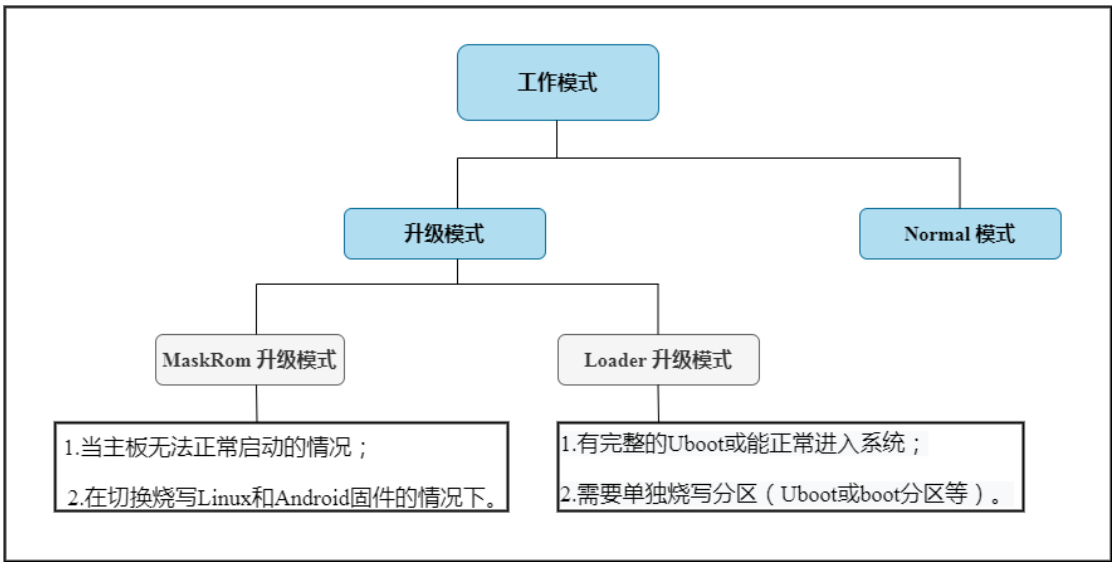
OS Version	Kernel Vesion	Support
Debian 11	Linux5.10	Yes
Android13.0	Linux5.10	Yes

3.2. Working Mode

IAC-RK3588-Kit provides two working mode, Normal mode and upgrade mode.

Under normal circumstances, the development kit starts system while enters into Normal mode. If the firmware needs to be upgraded, please select the suitable upgrade mode and firmware package.

Caution: The development kit has been preloaded the firmware before leaving the factory.



Normal Mode:

Normal mode is the normal boot process, where each component is loaded in turn and the board enters the system normally.

Upgrade Mode:

The upgrade mode includes MaskRom mode, Loader mode, and SD card mode; the SD card mode will not be clarified here.

MaskRom Mode:

MaskRom mode is used for system recovery when the bootloader is damaged. Generally, it is not necessary to enter the MaskRom mode. Only when the bootloader verification fails (the IDR block cannot be read, or the bootloader is damaged), the BootRom code will enter the MaskRom mode. At this time, the BootRom code waits for the host to transmit the bootloader code through the USB interface, load and run it.

Loader Mode:

In Loader mode, the bootloader will be in upgrade status, waiting for the commands from the host machine to upgrade the firmware, etc.

To enter Loader mode, the bootloader must detect that the RECOVERY key is pressed and the USB is connected during startup:

1. Make sure the device is connected to the power adapter and powered on;

2. Connect the mainboard with the host machine by a USB OTG cable;

3. Press and hold the RECOVERY (Recovery-SW2) key on the device;

4. Short press the RESET (Reset-SW1) button;

5. Press and hold for about 2 seconds, then release the RECOVERY (SW2) key.

Caution: If the device is still not found after pressing the RESET button, please press and hold the POWERON (SW3) button while keeping the RECOVERY button pressed, and then release the RECOVERY button.

3.3. Firmware Description

There are two types of firmware:

1. Single unified firmware: update.img, which packs the boot parameter loader, parameters and all partition images together for firmware release.

By default, a unified upgrade firmware image is provided, which can be obtained in the image folder from the shared network disk link.

2. Multiple partition images: such as kernel.img, boot.img, recovery.img and other files, which are generated in the development stage.

3.3. Firmware Flashing

IAC-RK3588-Kit ‘s firmware has been preloaded by default before leaving the factory.

Preparation:

1. IAC-RK3588-Kit development kit;

2. USB Type-c downloading cable;

3. Image file and firmware: They are available to obtain through a shared network disk link;

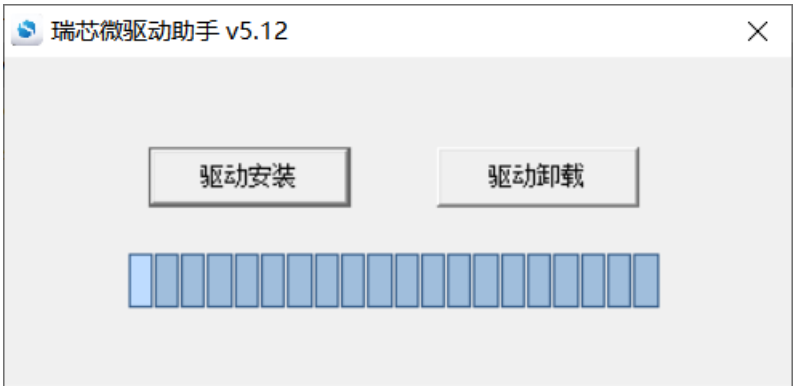
4. Host Machine (Computer) : Support Windows7 (32/64 bit), Windows10 (32/64 bit).

Firmware flashing tool: RKDevTool_Release_vx.xx (Version Number), please install RK USB driver before using the firmware flashing tool. If the driver is loaded, please skip this step.

Install RK USB driver:

Rockchip_DriverAssitant_v5.12 driver file, it could be obtained via shared network disk link.

Copy the ***Rockchip_DriverAssitant_v5.12*** driver compression package to the computer, unzip it, and then run the ***DriverInstall.exe***. In order to use the updated driver for all devices, please select ***Driver Uninstall***, and then select ***Driver Install***, as the below picture shown:



1. Make sure the device is connected to the power adapter and powered on.

2. Connect the device and the computer by a USB Type-C cable.

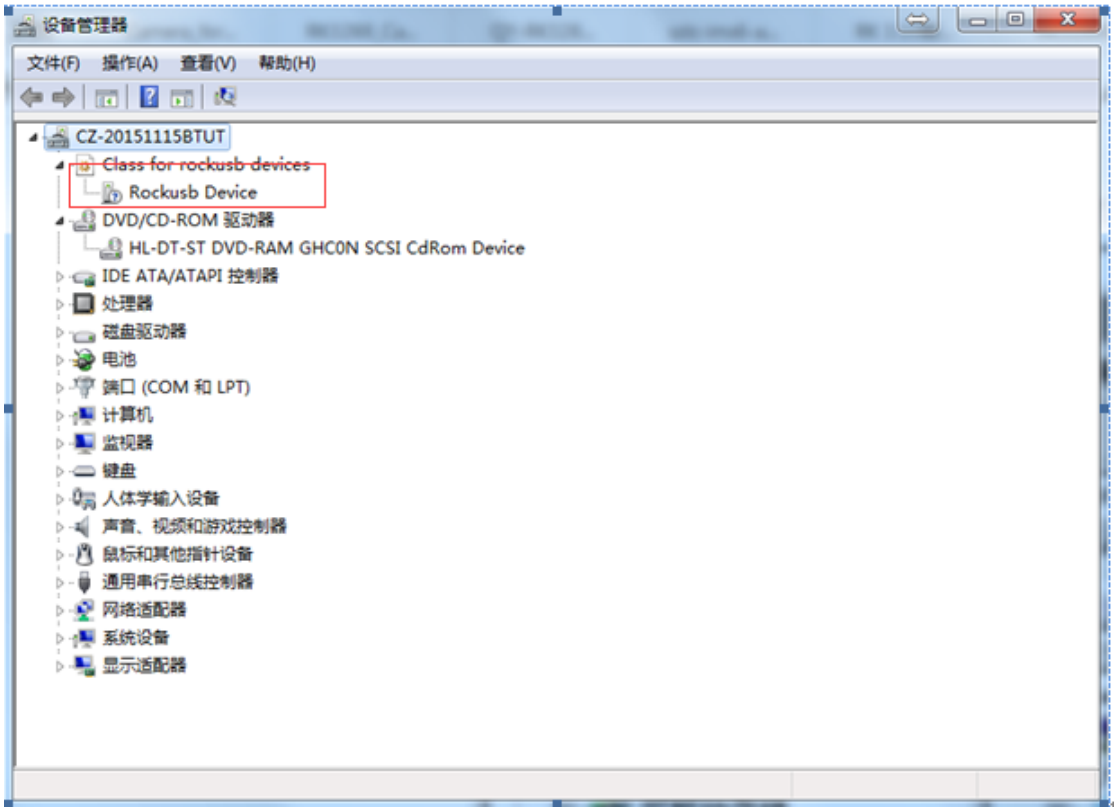
3. Press and hold the RECOVERY (SW2) key on the device.

4. Short press the RESET (SW1) key.

5. Long press and hold for about 2 seconds, release the RECOVERY key.

Note: If the device is still not found after pressing the RESET button, please press and hold the RECOVERY button while pressing the POWERON (SW3) button, and then release the RECOVERY button.

Till now, the computer should prompt that new hardware has been found and the configuration is finished. Open device manager, you will see the new device ***Rockusb Device*** appears, as the below picture shown:

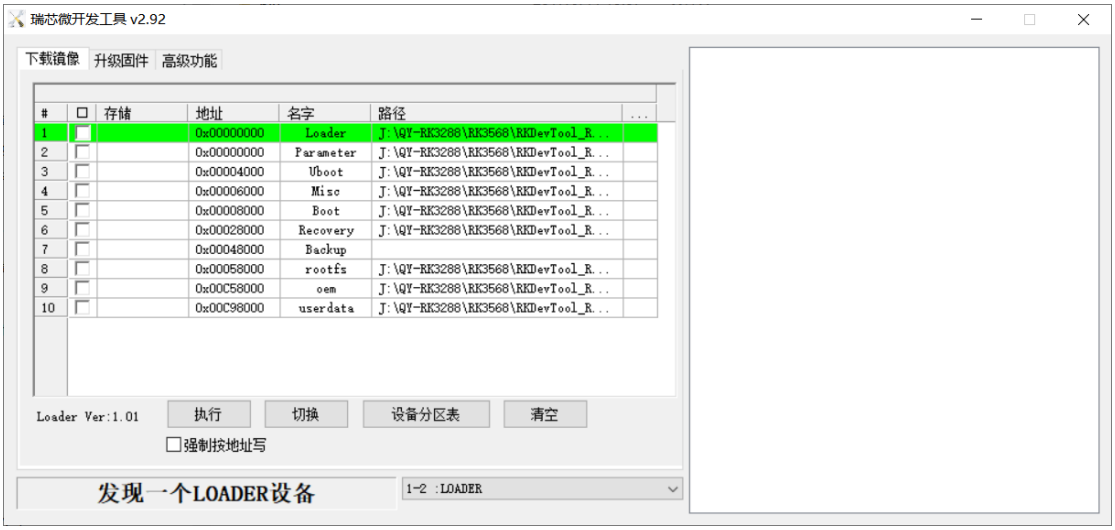


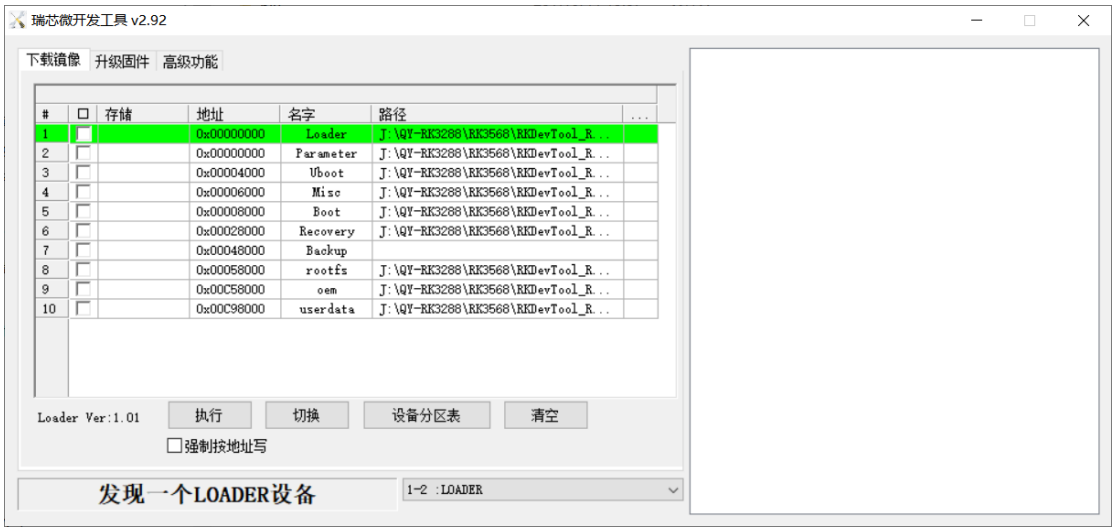
If not, please go back to the previous step to reinstall the RK USB driver.

Flash firmware:

Open Mfgtool ***RKDevTool_Release_v3.19*** or higher version, which can be obtained from the shared network disk link.

Open ***RKDevTool_Release_v3.19***, run ***RKDevTool.exe*** in the directory, and click ***Execute***, as the below picture shown:

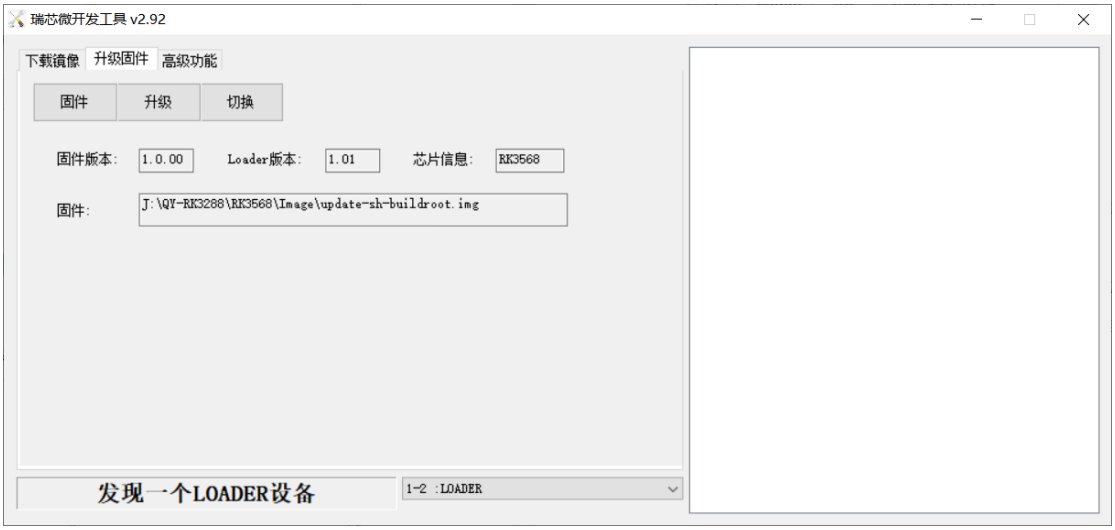




As the above picture shown: a LOADER device is found, which can be used for image flashing.

Flash unified firmware:

1. Switch to the ***Upgrade Firmware*** page, as the below picture shown



2. Press ***Firmware*** button to open the firmware file which need to upgraded (Available in network disk). The upgrade tool will show the detailed firmware information.

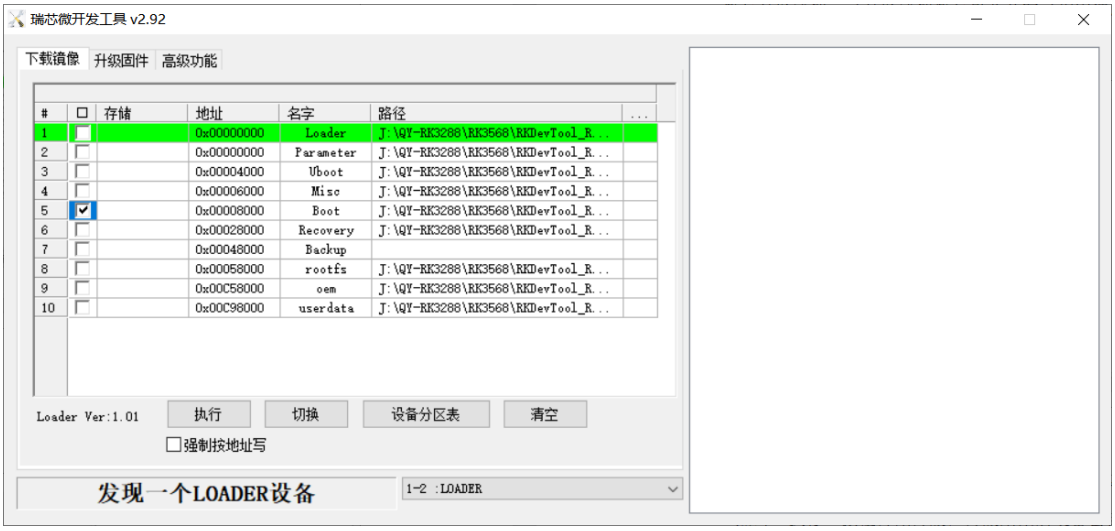
3. Press the ***Upgrade*** button to start the upgrade.

4. If upgrade fails, you can try ***Erase Flash*** button in the advanced function to erase the flash, and then upgrade.

Note: If the firmware loader version you flashed is inconsistent with the original machine, please perform ***Erase Flash*** before upgrading the firmware.

Flash partition image: (Partition image flashing can be used in the development process)

1. Switch to the ***Download Image*** page, as the below picture shown:



2. Check the partitions that need to be flashed, and you can choose multiple.

3. Make sure the path of the image file is correct. If necessary, click the blank table cell on right to re-select.

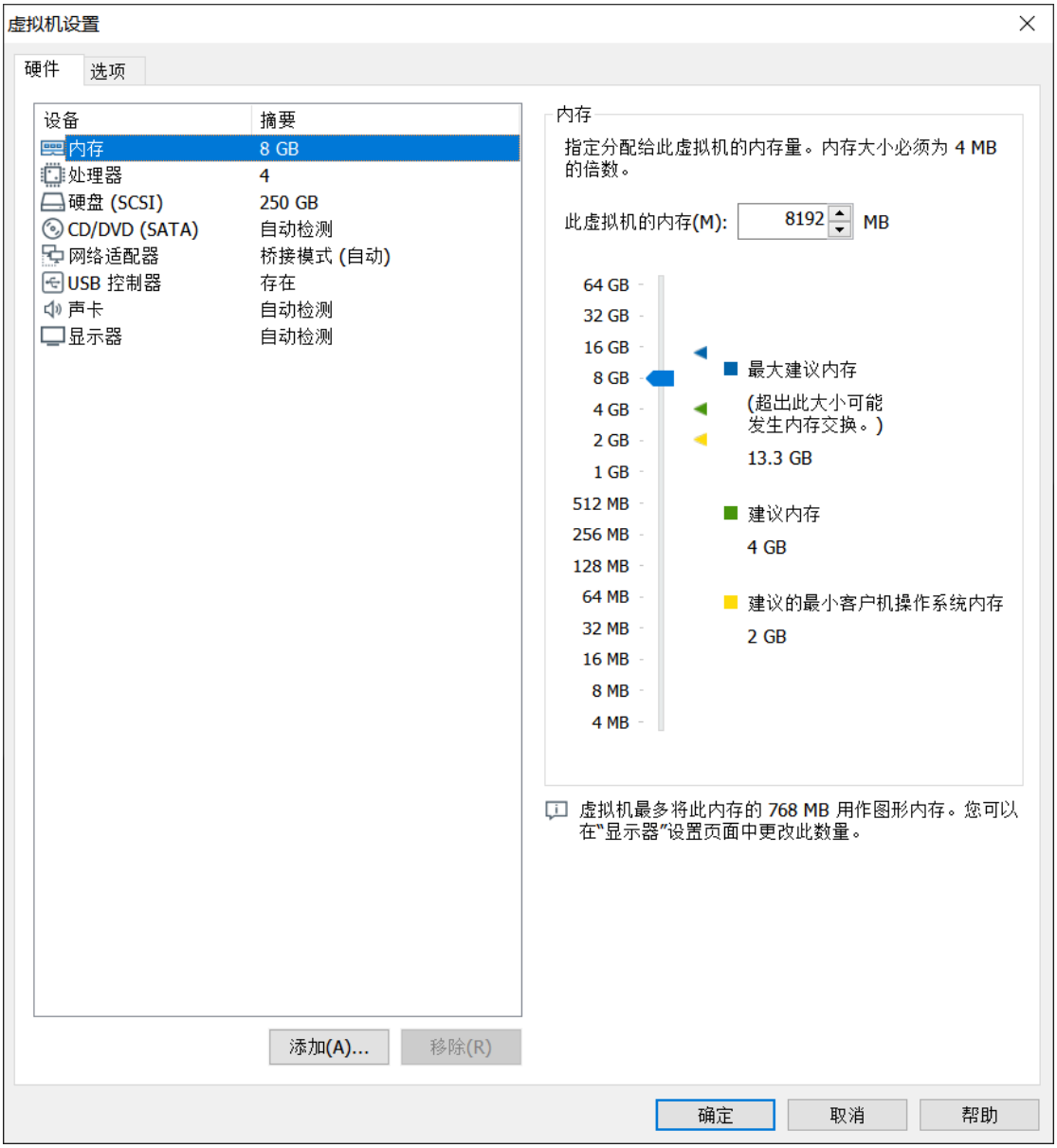
4. Click the ***Execute*** button to start the upgrade, and the device will automatically restart after the upgrade is finished.

IV. Linux Developing

4.1. Build Compilation Environment

4.1.1. Host Machine Configuration:

The PC environment is required to be ubuntu20.04 or higher version. Here is the author's virtual machine configuration information, which can be allocated according to the needs of the host.



4.1.2. Install the necessary libraries

Before compiling, you need to install some necessary libraries. The relevant libraries are as follows:

***sudo apt-get install repo git ssh make gcc libssl-dev liblz4-tool ***

***expect g++ patchelf chrpath gawk texinfo chrpath diffstat binfmt-support ***

qemu-user-static live-build bison flex fakeroot cmake gcc-multilib g++-multilib

***unzip ***

***device-tree-compiler python-pip ncurses-dev pyelftools ***

Note: Above libraries are just a small part, please install the missing libraries while the compilation reports error.

4.1.3. SDK compiler

The sdk directory contains ***u-boot kernel*** compiler and the ***buildroot*** compiler.

The two are compiled by different compilers. In fact, the ***u-boot kernel*** uses the same compiler path as below:

prebuilts/gcc/linux-x86/aarch64/gcc-arm-10.3-2021.07-x86_64-aarch64-none-linux-gnu/bin/aarch64-none-linux-gnu-

buildroot uses the compiler which is generated by buildroot, so it will be generated after buildroot is compiled. The path is as below:

buildroot/output/rockchip_rk3588/host/bin/aarch64-buildroot-linux-gnu-

4.1.4 sdk overall compiling (Default:buildroot)

Caution: Use ***make menuconfig*** command in top-level directory to select the default file system which needs compiling, and save it in character interface. And then execute ***make update-defconfig*** command and write in the configuration file.

Execute***./build.sh*** in ***sdk*** directory (Do not use root authority), it prompts the relative configuration options, the default option: *rockchip_rk3588_QIYANG_defconfig*,as the below picture shown:

```
chengsj@u20:~/Rk3588$ ./build.sh lunch

################ Rockchip Linux SDK ################

Manifest: IAC-RK3588-MB.xml
Version: linux-5.10-gen-rkr7

Log colors: message notice warning error fatal

Log saved at /home/chengsj/Rk3588/output/sessions/2024-04-24_11-31-31
Pick a defconfig:

1. rockchip_defconfig
2. rockchip_rk3588_QIYANG_defconfig
3. rockchip_rk3588_evb1_lp4_v10_defconfig
4. rockchip_rk3588_evb7_v11_defconfig
5. rockchip_rk3588s_evb1_lp4x_v10_defconfig
Which would you like? [1]:
```

After compilation, there is a sdk top-level directory in [rockdev] folder, and it generates the image file as below:

```
chengsj@u20:~/Rk3588/rockdev$ ls
boot.img  linux-headers.tar  MiniLoaderAll.bin  misc.img  oem.img  parameter.txt  recovery.img  rootfs.img  uboot.img  update.img  userdata.img
chengsj@u20:~/Rk3588/rockdev$
```

The below picture is the screenshot after successful compilation, the below repo error will not affect the result.

```
Generating new image, please wait...
Writing head info...
Writing boot file...
Writing firmware...
Generating MD5 data...
MD5 data generated successfully!
New image generated successfully!
Running mk-updateimg.sh - build_updateimg succeeded.
Images under /home/chengsj/Rk3588/output/firmware/ are ready!
Running mk-firmware.sh - build_firmware succeeded.
Saving linux-headers to /home/chengsj/Rk3588/output/firmware/linux-headers.tar
+ make -C /home/chengsj/Rk3588/kernel/ -j25 CROSS_COMPILE=/home/chengsj/Rk3588/prebuilts/gcc/linux-x86/aarch64/gcc-arm-10.3-2021.07
3588_linux.config
make: Entering directory '/home/chengsj/Rk3588/kernel'
#
# No change to .config
#
Using .config as base
Merging ./arch/arm64/configs/rk3588_linux.config
#
# merged configuration written to .config (needs make)
#
#
# configuration written to .config
#
make: Leaving directory '/home/chengsj/Rk3588/kernel'
+ make -C /home/chengsj/Rk3588/kernel/ -j25 CROSS_COMPILE=/home/chengsj/Rk3588/prebuilts/gcc/linux-x86/aarch64/gcc-arm-10.3-2021.07
make: Entering directory '/home/chengsj/Rk3588/kernel'
  SYNC    include/config/auto.conf.cmd
  CALL    scripts/atomic/check-atomics.sh
  CALL    scripts/checksyscalls.sh
  CHK     include/generated/compile.h
make: Leaving directory '/home/chengsj/Rk3588/kernel'
+ cd /home/chengsj/Rk3588/kernel
{
        # Based on kernel/scripts/package/builddeb
        find . arch/arm64 -maxdepth 1 -name Makefile\*
        find include -type f -o -type l
        find arch/arm64 -name module.lds -o -name Kbuild.platforms -o -name Platform
        find $(find arch/arm64 -name include -o -name scripts -type d) -type f
        find arch/arm64/include Module.symvers -type f
        echo .config
} | tar --no-recursion --ignore-failed-read -T -        -cf "/home/chengsj/Rk3588/output/firmware/linux-headers.tar"
+ tar -uf /home/chengsj/Rk3588/output/firmware/linux-headers.tar scripts tools
+ cd /home/chengsj/Rk3588
Running 99-all.sh - build_all succeeded.
```

Caution: If the compilation is always unable to succeed, please contact Qiyang FAE to obtain the finished virtual machine. The size is about 200GB.

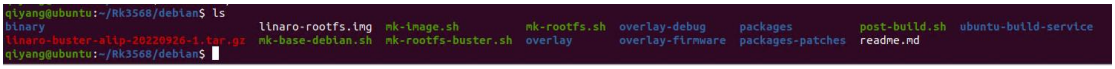
4.2. Debian firmware compiling

4.2.1. Strictly, it is not the real compilation while compiling Debian firmware, it is actually adding some RK specific packages on Debian basic package. Normally, it will not report errors, below are the compilation procedures:

Caution: It will download the offline package while compiling Debian, you can ask for the downloaded offline packages from the FAE.

./buid.sh debian

The generated file is as below:



linaro-roofts.img is the Debian image which we burnt to the board.

IV.Android developing

5.1. How to use ADB

IAC-RK3588-Kit is using ***adb debug*** as default, it is available to do ***adb debug*** by usb-otg. Whether the adb can't be detected, if yes, please check if the driver is installed normally, whether the usb device is recognized on PC.

5.2. Setup compiling environment

Based on ubuntu20.04, here, we listed the common libraries' installation. Some libraries are impossible to found or use. Please Google the similar libraries, if the sequent compilation lacks of some libraries, you can install the similar libraries based on the reported information.

sudo apt-get install git gnupg flex bison gperf libsdl1.2-dev \libesd-java libwxgtk3.0-dev squashfs-tools build-essential zip curl \libncurses5-dev zlib1g-dev pngcrush schedtool libxml2 libxml2-utils \xsltproc lzop libc6-dev schedtool g++-multilib lib32z1-dev lib32ncurses5-dev \lib32readline-dev gcc-multilib libswitch-perl libssl-dev unzip zip device-tree-compiler \liblz4-tool python-pyelftools python3-pyelftools openjdk-8-jdk

5.3. Android 13.0 firmware compiling

The Android source code is large, it has been split for several packages, after decompression, it starts compiling, the steps are as below:

Block compilation:

Kernel: export PATH=../prebuilts/clang/host/linux-x86/clang-r450784d/bin:$PATH

alias msk='make CROSS_COMPILE=aarch64-linux-gnu- LLVM=1 LLVM_IAS=1'

msk ARCH=arm64 rockchip_defconfig android-13.config pcie_wifi.config && msk ARCH=arm64 BOOT_IMG=../rockdev/Image-rk3588_t/boot.img rk3588-evb7-lp4-v10.img -j8

Android: ***1、source build/envsetup.sh 2、lunch rk3588_r-userdebug 3、make -j16***

Package command：***./mkimage.sh***

Overall Compilation:

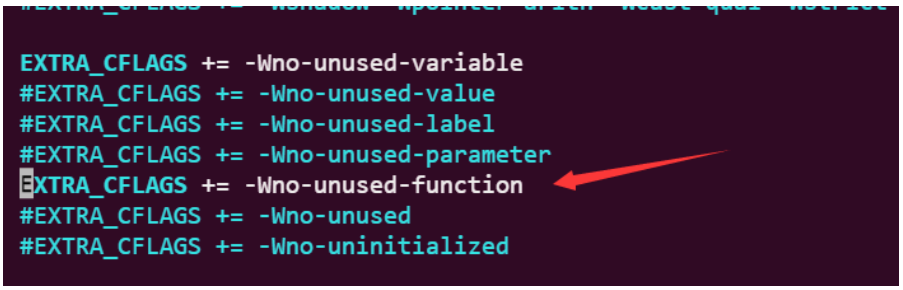
1.source build/envsetup.sh 2.lunch rk3588_r-userdebug 3. ./build.sh -UKAup

5.4. Reflash Android image file

The procedures and methods are same for firmware flashing and Android image file, the difference is to distinguish the files.

VI.FAQ

1.While compiling Debian, the WIFI/BT will be compiled. Due to the compilation regularity, it will report errors if the functions were not used. Need to revise the compilation regularity based on the reported errors, the revised contents are as below:



Zhejiang Qiyang Intelligent Technology Co., Ltd.

Tel: 86-571-87858811 / 87858822

Fax: 86-571-89935912

Technical Support： 0571-87858811 ext.805

E-MAIL： supports@qiyangtech.com

Website: **http://www.qiytech.com**

or **http://www.qiyangtech.com**

ADD: 3rd Floor, Building A, WSCG Building, NO.6 Xiyuan 8th Road, Sandun Town, Xihu District, Hangzhou City, Zhejiang China

Postal Code： 310013