IAC-RK3576-Kit User Manual

Ver.#:1.0
2025.07

Company Profile

Founded in 2007 in Hangzhou, Zhejiang Qiyang Intelligent Technology Co., Ltd. is a national high-tech enterprise focusing on the research, development, production and sales of ARM embedded products. more than 10 years of accumulation and precipitation, and successfully built the product from development to mass production of the service chain.

As the core of the company, Qiyang R&D team consists of more than 30 embedded engineers, dedicated to providing users with easy-to-use embedded hardware, software tools and customized product solutions. It has been widely used in industrial control, Internet of Things, new retail, medical, electric power, environmental monitoring, charging pile and other fields.

The production base in Zhuji provides a strong guarantee for Qiyang, covering an area of 5,000 square meters, with two SMT production lines, through and strictly follow the ISO9001 quality management system certification to guide production. Relying on the strong production strength, the annual production capacity of up to 1 million sets, to ensure that the user delivery, to solve the worries.

Qiyang has a perfect sales and marketing network, professional sales and after-sales team to provide users with a full range of technical support and services. Our business has spread to more than 120 countries and regions, and we have successfully helped more than 2,000 users to bring their products to the market quickly and efficiently.

The combination and extension of R&D, production capacity and market has laid a solid foundation for Qiyang Intelligence to become a specialized and globalized embedded hardware and software supplier.

We offer you:

- **Multi-platform software/hardware products**

NXP, Rockchip, MTK, Renesas, TI, Atmel, Cirrus Logic and other multi-platform ARM development boards/core boards/industrial control boards and

peripheral hardware products, as well as supporting the user to support rapid secondary development of supporting tools and software resources.

- **Customized services**

Give full play to the accumulated technology on ARM platform and Linux, Android, Ubuntu, Debian operating systems to provide users with customized embedded product services (OEM/ODM).

Thank you for using Qiyang Intelligence's products, we will do our best to provide you with technical assistance! We wish you good luck in your work!

Technical Support

If you have questions about the documentation, you can contact us during office hours (Monday - Friday 8:30-12:00, 13:30-17:30) by one of the following methods:

Technical e-mail: supports@qiyangtech.com

Technical Support Tel: 0571-87858811-805

Website: www.qiytech.com(Chinese)/www.qiyangtech.com(English)

Updates and access to information

1. Updating of information

The information about the product is constantly being improved and updated, so please make sure that it is up to date when you use the content.

2、Update notification

Please note that Qiyang Intelligence's products update are notified via WeChat.

3. Access to information

After product purchase, please contact our business staff to get it.

4. Provision of information

Software: factory image, related kernel source code, interface test source code, cross-compiler

Hardware: Corresponding baseboard schematics, PCB source files (Allegro 16.6)

Documentation: Hardware manual, test manual, user manual, environment construction manual, IO pin comparison table, core board, chassis structure dimension drawing (dxf), original chip information, etc.

Recommendations for use

1) Before using the development board, be sure to read the hardware manual first;

2) Before use, please check the packing list in detail and test the information CD-ROM for missing documents;

3) Understand the basic structure and composition of the development board, including the allocation of hardware resources, the SOMand the base board of each pin definition, as well as the definition of the expansion pin and so on;

4) If you need to burn images, development board function test, etc., you also need to read the user manual and test manuals

5) IAC-RK3576-Kit development board, bulk order accepted.

Version Record

Version No.	Hardware Platform	Date	Revision	Revised by
V1.0	IAC-RK3576-MB-BETA-V1_00 IAC-RK3576-CM-BETA-V1_00	2025-06	Initial version	Maoh

Contents

Note: This manual mainly introduces the hardware interface of this development board.

I. Preparation

◆ Prepare a set of PC running Windows 10.

◆ A set of PC which installed VM, the virtual machine which has already installed Linux system (Ubuntu or other distributions) or a PC which installed Linux system directly. If the Linux has not installed yet, please install the VM Ubuntu 20.04 in Windows 10 refers to the ***Ubuntu Installation Instruction Manual***,

◆ UART :Connect the UART of PC to DEBUG UART (J25) on Development Board through a 3-PIN serial cable (RS232).

(If there is no UART on PC or using laptop, please prepare an USB TO UART (RS232) cable additionally.)

◆ Network: Connect the LAN port of PC to Ethernet port (J1) on development board, or through network switch.

◆ USB: Connect USB port of PC to USB Device (J6) on development board by USB cable.

◆ UART setting: Open terminal communication software in Windows or SecureCRT tool in the provided SDK package; Connect to the serial port on the development board, and set the UART with below parameters: Baud rate:115200, Data bit:8-bit, Stop bitt: 1-bit, Parity bit: None, Data flow control: None.

II. System Image Flashing (Firmware Flashing)

Remark: The image file has been preloaded to the board before delivery.

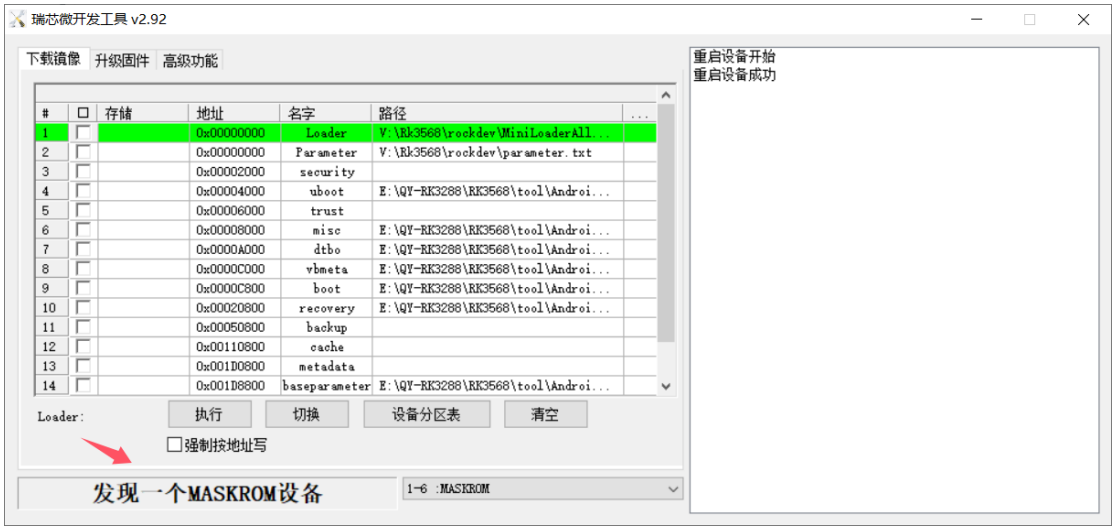
2.1.Work Mode

IAC-RK3576-Kit has three modes, boot-up mode, Maskrom mode and Loader mode. Amongst, Maskrom mode and Loader mode are for image flashing mode, the difference between them are the application areas, the details are as below:

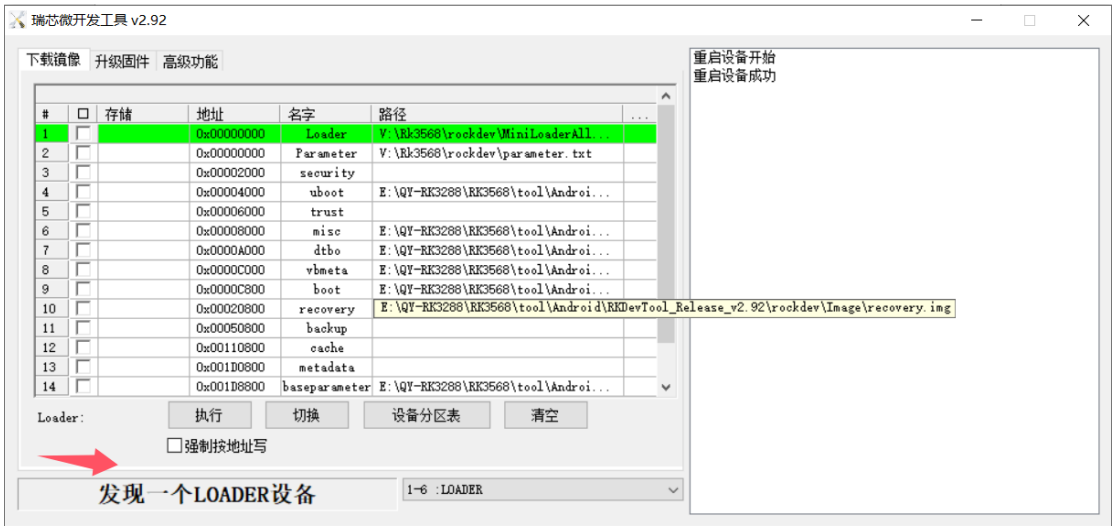
Boot-up mode: The development board boots up normally, it enters into this mode by default, serial's log prints normally, the serial Log is as below:

```
         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:~#
```

Maskrom Mode: While the SOM module doesn't have any image file or the SOM can't boot normally after erasing 'flash' ,it will cause force entry ,the SOM will enter Maskrom mode, the step for flashing software is as below:



Loader Mode: The SOM boots normally; When debugging the kernel, the full image or partition-specific image is burnt to enter this mode, the flashed software is shown as below:

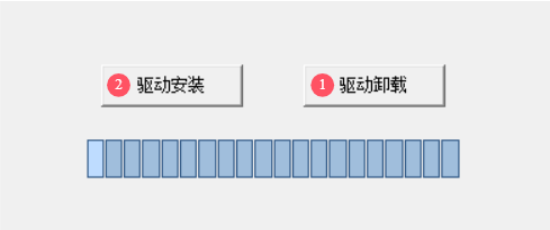


2.2.Image Introduction

There are two image files; one is a full image file package which includes all partition images; another is an individual image package which is flashed to the specific partition image when debugging.;
The file name, contents and size are different; Full image file package is named as update.img which is big ; The individual image file package is named as boot.img which is small.

2.3. Image Flashing

1.Install RK USB driver:
Extract Rockchip_DriverAssitant_zip from Onedrive; there will be an executable file ‘exe’ generated. Double-click and open it; Select ‘卸载驱动（Uninstall Driver）,then select ‘驱动安装 Install Driver’， as below picture shown:



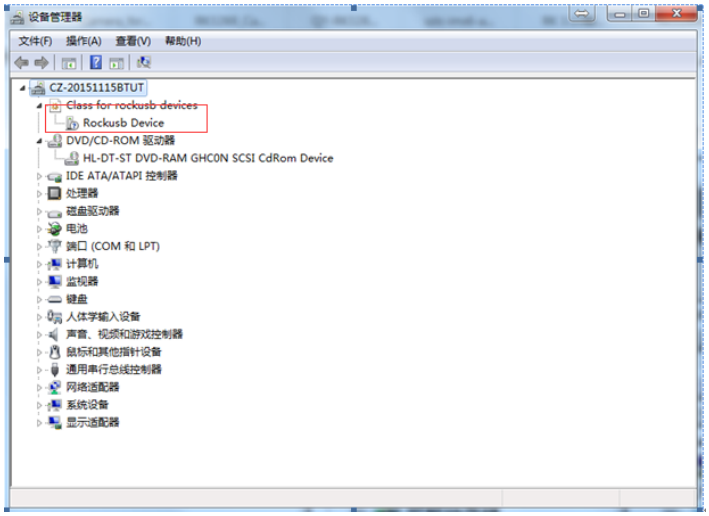
2.Enter Loader Mode

Type-C connects to IAC-RK3576-Kit (J6)

Mode 1（Hardware）: Long-press and hold Recovery (SW2) then brie fly press Reset(SW10)，release Recovery (SW2) after about 2 seconds.

Mode 2 (Software): Input ‘reboot loader’ command in serial terminal or using RKDevTool button to switch loader mode.

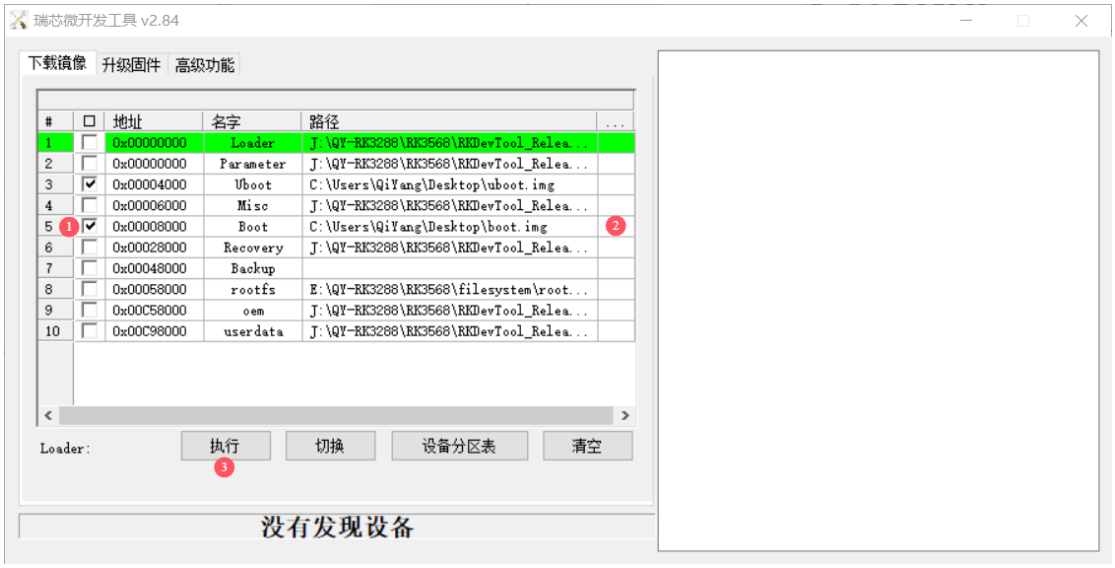
When it enters Loader mode , there will be a USB device recognized in PC device management, it shows as below：



3.Image Flashing

Full image file flashing: Extract RKDevTool_Release zip, then double-click RKDevTool.exe file, click ‘升级固件 Upgrade Firmware’->‘固件 Firmware’->’选中完整镜像包 Select the full image’->‘升级 Upgrade’

Partition Image Flashing: Extract RKDevTool_Release zip, double-click RKDevTool.exe file, to select the option box ->The blank area is loading the partition firmware-> Execute



4.Erase ‘flashing’

If you want to reflash Android system, you have to erase full image, then reflash the Android image. The steps are as below:

Enter Loader mode, then click RKDevTool.exe ->Erase Flash, then enter Maskrom mode after erasing.

III. Linux Compilation

3.1.Build PC host environment

Please refer to ***Ubuntu Setup Manual***

3.2.Install necessary libraries

It requires to install the necessary libraries after the PC host environment is set up. (Below libraries are the general libraries, please install other libraries based on it's error report during compilation.)

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 \

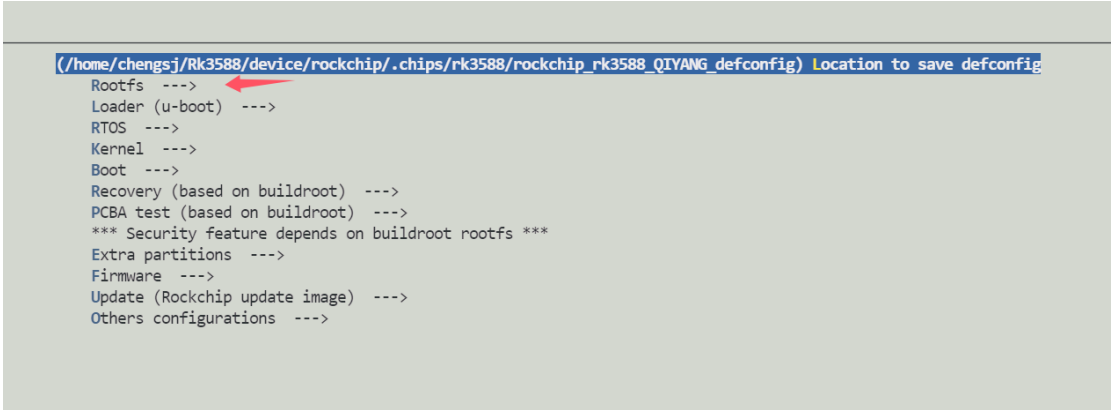
3.3.SDK Compiler Introduction

SDK directory includes u-boot kernel compiler and buildroot compiler, they are using different compilers to compile, Amongst, u-boot, kernel are using same compiler's path: prebuilts/gcc/linux-x86/aarch64/gcc-arm-10.3-2021.07-x86_64-aarch64-none-linux-gnu/bin/aarch64-none-linux-gnu-

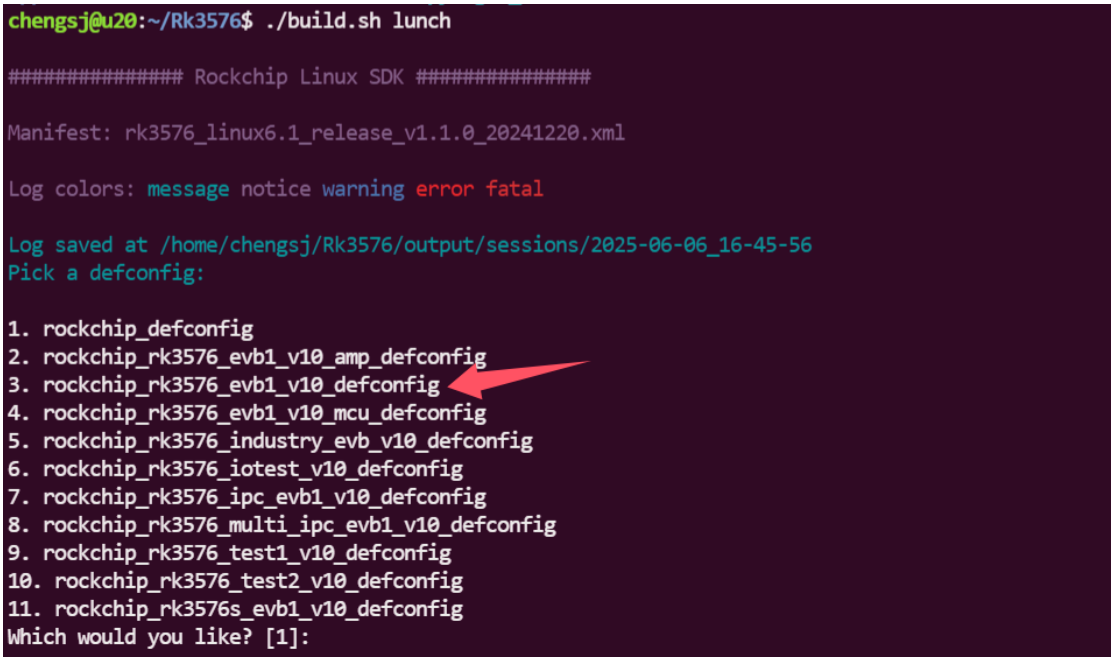
And buildroot is using the compiler which the buildroot generated; it will be generated after the buildroot finished compilation, path: buildroot/output/rockchip_rk3576/host/bin/aarch64-buildroot-linux-gnu-

3.4.SDK Compilation (Default Compilation: Debian Bookworm)

Remark: To execute ‘make menuconfig’ command in SDK top directory ,select default file system to compile as below picture shown, please confirm before compiling!



Execute ‘./build.sh’ directly in SDK top directory (DON’T use root permiss ion), it will show relative configuration options, the default option: rockchi p_rk3576_evb1_v10_defconfig,as below picture shown:



It generates ‘rockdev’folder in SDK top directory after normal compiling, it includes below image files in folder.

Attached is the screenshot after successful compilation, below repo error will not affect compilation result:

```
uboot.img(/home/chengsj/Rk3576/u-boot/uboot.img): 4.0M
userdata.img(/home/chengsj/Rk3576/output/extra-parts/userdata.img): 8.0M
Making update image...
=========================================
         Start packing update image
=========================================
Generating package-file for update:
# NAME  PATH
package-file    package-file
parameter       parameter.txt
bootloader      MiniLoaderAll.bin
uboot   uboot.img
misc    misc.img
boot    boot.img
recovery        recovery.img
backup  RESERVED
rootfs  rootfs.img
oem     oem.img
userdata        userdata.img
Packing /home/chengsj/Rk3576/output/firmware/update.img for update...
Android Firmware Package Tool v2.29
------ PACKAGE ------
Add file: ./package-file
package-file,Add file: ./package-file done,offset=0x800,size=0xe1,userspace=0x1
Add file: ./parameter.txt
parameter,Add file: ./parameter.txt done,offset=0x1000,size=0x227,userspace=0x1,flash_address=0x00000000
Add file: ./MiniLoaderAll.bin
bootloader,Add file: ./MiniLoaderAll.bin done,offset=0x1800,size=0xbc1f9,userspace=0x179
Add file: ./uboot.img
uboot,Add file: ./uboot.img done,offset=0xbe000,size=0x400000,userspace=0x800,flash_address=0x00004000
Add file: ./misc.img
misc,Add file: ./misc.img done,offset=0x4be000,size=0xc000,userspace=0x18,flash_address=0x00006000
Add file: ./boot.img
boot,Add file: ./boot.img done,offset=0x4ca000,size=0x26ee600,userspace=0x4ddd,flash_address=0x00008000
Add file: ./recovery.img
recovery,Add file: ./recovery.img done,offset=0x2bb8800,size=0x5cac000,userspace=0xb958,flash_address=0x00028000
Add file: ./rootfs.img
rootfs,Add file: ./rootfs.img done,offset=0x8864800,size=0x117700000,userspace=0x22ee00,flash_address=0x00078000
Add file: ./oem.img
oem,Add file: ./oem.img done,offset=0x11ff64800,size=0x11ef000,userspace=0x23de,flash_address=0x01c78000
Add file: ./userdata.img
userdata,Add file: ./userdata.img done,offset=0x121153800,size=0x800000,userspace=0x1000,flash_address=0x01cb8000
Add CRC...
Make firmware OK!
------ OK ------
********rkImageMaker ver 2.29********
Generating new image, please wait...
Writing head info...
Writing boot file...
Writing firmware...
```

Remark: If you can't compile it successfully after several tries, you can contact with our FAE to get the finished VM, it is about 200G. We can share through Onedrive.

3.5.Compile Debian

Remark: Below operations only operate in Debian directory.

1. Compile Debian package (If it can't compile for network reason, the finished compilation package is available in Onedrive: linaro-bookworm-alip-20250707-1.tar.gz)

RELEASE=bookworm TARGET=desktop ARCH=arm64 ./mk-base-debian.sh

2.Compile RK package

RELEASE=bookworm ARCH=arm64 ./mk-rootfs.sh

3.Packing image file

./mk-image.sh

In the end, it generates the below file:

```
chengsj@u20:~/RK3576/debian$ ls
binary                          linaro-rootfs.img  mk-image.sh      mk-rootfs-bookworm.sh  overlay          overlay-firmware  packages-patches  readme.md  ubuntu-build-service
linaro-bookworm-alip-20250707-1.tar.gz  mk-base-debian.sh  mk-iso-debian.sh  mk-rootfs.sh           overlay-debug  packages          post-build.sh     scripts
chengsj@u20:~/RK3576/debian$
```

The linaro-rootfs.img is the image file which we preloaded to the board.

IV. Android Compilation

4.1.How to use ADB

IAC-RK3576-Kit has already supported usb adb; it is available to debug through USB-OTG,If it can't be recognized, please check if the driver is installed normally, and USB device is recognized.

4.2.Install the necessary library

After the PC host environment is set up ,it has to install the necessary libraries. (Below libraries are the general libraries, please install other libraries based on it's error report during compilation.)

sudo apt-get install openjdk-11-jdk bison g++-multilib git gperf libxml2-utils make zlib1g-dev:i386 zip liblz4-tool libncurses5 libssl-dev bc flex curl python-is-python3 zlib1g-dev libelf-dev dwarves

4.3.Compile Android image

Step1. source build/envsetup.sh
Step2. lunch rk3576_u-userdebug
Step3. ./build.sh -UKAu

4.4.Flash Android Image

The steps and methods for burning Android image are same as chapter 2.3.The difference is on system, if you want to change from Linux OS to Android OS, you have to erase the previous image first, then reflash Android system.

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