

# Rockchip RK3588 Linux Edge SDK 快速入门

---

文件标识: RK-JC-YF-856

发布版本: V1.1.0

日期: 2022-05-10

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

## 免责声明

本文档按“现状”提供, 瑞芯微电子股份有限公司(“本公司”, 下同)不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

## 商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自拥有者所有。

版权所有 © 2022 瑞芯微电子股份有限公司

超越合理使用范畴, 非经本公司书面许可, 任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部, 并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

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

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: [fae@rock-chips.com](mailto:fae@rock-chips.com)

## 前言

## 概述

本文提供 Rockchip RK3588 边缘计算SDK 快速入门指导。

## 产品版本

芯片名称	内核版本
RK3588/RK3588s	5.10

## 读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

## 修订记录

版本号	作者	修改日期	修改说明
V1.0.0	Addy Ke	2022-01-07	初始版本
V1.0.1	Addy Ke	2022-01-25	增加配置：用户密码
V1.0.2	Addy Ke	2022-03-25	修改配置项目，增加烧写、串口调试和SD卡启动说明
V1.0.3	Addy Ke	2022-0426	增加SD卡启动说明，修改编译配置
V1.1.0	Addy Ke	2022-0510	发布Beta版本V0.1.0

# 目录

## Rockchip RK3588 Linux Edge SDK 快速入门

1. 前言
2. 搭建系统环境
  - 2.1 编译主机系统要求
  - 2.2 安装编译依赖基础软件
    - 2.2.1 Debain系统安装依赖
    - 2.2.2 Ubuntu系统安装依赖
  - 2.3 获取源码
3. 编译配置
  - 3.1 新增配置
    - 3.1.1 新增RK3588产品配置
    - 3.1.2 新增RK3588s产品配置
  - 3.2 设置配置信息
  - 3.3 查看配置信息
  - 3.4 配置信息说明
    - 3.4.1 系统配置
    - 3.4.2 安全启动
    - 3.4.3 分区配置
    - 3.4.4 uboot配置
    - 3.4.5 内核配置
    - 3.4.6 文件系统配置
4. 镜像编译
  - 4.1 一键编译
  - 4.2 生成分区文件
  - 4.3 编译Uboot镜像
  - 4.4 编译kernel镜像
  - 4.5 制作rootfs镜像
  - 4.6 打包update镜像
  - 4.7 生成镜像签名的公私钥
  - 4.8 查看编译帮助
5. 烧写镜像
  - 5.1 进入烧写模式
    - 5.1.1 进入loader烧写模式
    - 5.1.2 进入maskrom烧写模式
  - 5.2 查询烧写状态
    - 5.2.1 Linux主机查询
    - 5.2.2 Windows主机查询
  - 5.3 LINUX主机烧写镜像
    - 5.3.1 烧写所有镜像
    - 5.3.2 烧写uboot镜像
    - 5.3.3 烧写内核镜像
    - 5.3.4 烧写文件系统镜像
    - 5.3.5 烧写misc镜像
    - 5.3.6 烧写oem镜像
    - 5.3.7 烧写userdata镜像
    - 5.3.8 查看烧写帮助
  - 5.4 Windows主机烧写镜像
6. 串口调试
  - 6.1 串口连接
  - 6.2 Windows主机调试
    - 6.2.1 获取端口号
    - 6.2.2 配置调试串口信息
  - 6.3 Linux主机调试
    - 6.3.1 安装minicom
    - 6.3.2 配置调试窗口信息

### 6.3.3 运行minicom

## 7. SD卡启动

### 7.1 制作SD启动卡

### 7.2 从SD卡启动

## 8. 安全启动

## 9. 常见问题

# 1. 前言

---

为了方便文档描述，约定如下变量定义：

- BOARD：开发板/产品型号；如RK3588 EVB1开发板的型号是RK3588\_EVB1。
- DTB：内核设备树；如RK3588 EVB1开发板的产品内核设备树是rk3588-evb1-lp4-v10-linux。
- ROOT\_DIR：边缘计算SDK的工作目录；文档中所示的目录为/home/toybrick/work/edge。
- CHIP：开发板的芯片型号；当前边缘计算SDK支持的芯片型号为rk3588或rk3568。
- OUT\_DIR：编译生成的镜像路径\${ROOT\_DIR}/out/\${CHIP}/\${BOARD}/images，如RK3588 EVB1的路径为/home/toybrick/work/edge/out/rk3588/RK3588\_EVB1/images。

## 2. 搭建系统环境

---

### 2.1 编译主机系统要求

- Ubuntu18.04及以上和Debian11版本，内存推荐16GB及以上。
- 系统的用户名不能有中文字符。
- 只能使用普通用户搭建开发环境，不允许用root用户执行。

### 2.2 安装编译依赖基础软件

#### 2.2.1 Debain系统安装依赖

```
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
```

#### 2.2.2 Ubuntu系统安装依赖

```
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 获取源码

- 从FAE获取RK3588\_EDGE\_SDK软件包，解压到工作目录。
- 从FAE获取原生Debian11 Gnome桌面的镜像压缩文件（debian11-gnome.tar.gz）拷贝：  
\${ROOT\_DIR}/rootfs/images/aarch64/目录下（注意：不需要解压）。

## 3. 编译配置

### 3.1 新增配置

边缘计算SDK默认支持RK3588-EVB1，RK3588-EVB4，RK3588s-EVB1和TB-RK3588X0板子。如果是基于这几块板子开发，不需要新增配置，请略过本章节。

#### 3.1.1 新增RK3588产品配置

假设新增开发板型号为RK3588-MY-PRODUCT，新增开发板的内核设备树为rk3588-my-product-linux，其操作步骤如下：

1. 拷贝RK3588-EVB1到RK3588-MY-PRODUCT：

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

2. 修改vendor/rk3588/RK3588-MY-PRODUCT/config.json内容如下所示：

- board：修改board的值为RK3588-MY-PRODUCT。
- dtbname：修改kernel子集里的dtbname的值为rk3588-my-product-linux。
- 其他配置：参考《配置信息说明》修改其他配置。

#### 3.1.2 新增RK3588s产品配置

假设新增开发板型号为RK3588s-MY-PRODUCT，新增开发板的内核设备树为rk3588s-my-product-linux，其操作步骤如下：

1. 拷贝RK3588-EVB1到RK3588s-MY-PRODUCT：

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

2. 修改vendor/rk3588/RK3588s-MY-PRODUCT/config.json内容如下所示：

- board：修改board的值为RK3588s-MY-PRODUCT。
- dtbname：修改kernel子集里的dtbname的值改为rk3588s-my-product-linux。
- chip：保持不变（RK3588和RK3588s均设置为rk3588）。
- 其他配置：参考本章节的《配置信息说明》修改其他配置。

### 3.2 设置配置信息

执行如下执行命令，输入产品型号的序号（如：RK3588 EVB1开发板所对应的序号为0）设置配置信息：

```

./edge set

[EDGE DEBUG] Board list:
> rk3588
  0. RK3588s-EVB1
  1. TB-RK3588X0
  2. RK3588-EVB1
  3. RK3588-EVB4
  4. TB-RK3588SD0
> rk3568
  5. TB-RK3568X0
Enter the number of the board: 1

```

注意：每次更新或修改边缘计算**SDK**相关代码后，请重新执行此命令，更新配置。

### 3.3 查看配置信息

执行如下命令，查看当前配置信息：

```

./edge env

[EDGE DEBUG] root path: /home/toybrick/edge
[EDGE DEBUG] out path: /home/toybrick/edge/out/rk3588/RK3588s-EVB1/images
[EDGE DEBUG] board: RK3588s-EVB1
[EDGE DEBUG] chip: rk3588
[EDGE DEBUG] arch: arm64
[EDGE DEBUG] bootmode: extlinux
[EDGE DEBUG] > Secureboot:
[EDGE DEBUG]   enable: False
[EDGE DEBUG]   rollback: [0, 0]
[EDGE DEBUG]   burnkey: False
[EDGE DEBUG] > Partition:
[EDGE DEBUG]   uboot: ['0x00002000', '0x00004000']
[EDGE DEBUG]   misc: ['0x00006000', '0x00002000']
[EDGE DEBUG]   boot_linux:bootable: ['0x00008000', '0x00020000']
[EDGE DEBUG]   recovery: ['0x00028000', '0x00040000']
[EDGE DEBUG]   resource: ['0x00068000', '0x00010000']
[EDGE DEBUG]   rootfs:grow: ['0x00078000', '-']
[EDGE DEBUG] > Uboot:
[EDGE DEBUG]   config: rk3588-edge
[EDGE DEBUG] > Kernel:
[EDGE DEBUG]   version: 5.10
[EDGE DEBUG]   config: rk3588_edge.config
[EDGE DEBUG]   dtbname: rk3588s-evb1-lp4x-v10-linux
[EDGE DEBUG]   size: 72
[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]   apturl: http://repo.rock-chips.com
[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 配置信息说明

SDK配置项包含系统配置、安全启动、分区配置、uboot配置、内核配置和文件系统配置。配置说明如下：

- 公共配置保存在vendor/common/config.json。
- 板级配置保存在vendor/\${CHIP}/\${BOARD}/config.json，其值会覆盖公共配置的同名配置项的值。
- 值为“not set”的配置项必须在《板级配置》中设置；其他配置项可根据实际需要在《板级配置》中修改。

### 3.4.1 系统配置

系统配置项包含board、chip、arch和bootmode配置项：

配置项	描述	默认值	备注
board	开发板或产品型号	not set	board值必须和vendor/\${CHIP}目录下的\${BOARD}目录名一致
chip	芯片型号	not set	RK3588和RK3588s的芯片型号都设置为rk3588
arch	芯片架构	not set	arm或arm64
bootmode	Uboot的启动方式	extlinux	支持extlinux和fit两种启动方式

### 3.4.2 安全启动

安全启动的配置项为secureboot，包含enable、rollback和burnkey三个子项：

配置项	描述	默认值	备注
enable	是否使能安全启动	false	只有fit启动方式支持安全启动
rollback	防回滚的版本号	[0, 0]	0表示不启用防回滚
burnkey	是否把公钥hash烧写到OTP/eFuse	false	

说明：

1. rollback的值[N, M]表示uboot回滚的最小版本号是 N, boot和recovery回滚的最小版本号是 M。
2. burnkey使能后，系统将只能引导这把key签名的镜像，请谨慎操作。



### 3.4.3 分区配置

系统的分区信息，包括分区名，起始地址和分区大小（起始地址和分区大小的单位为block，每个block的大小为512字节）。编译脚本会根据bootmode设置的启动方式加载part-extlinux和part-fit分区表，分区表信息如下：

**part-extlinux:** extlinux启动方式的系统分区

配置项	描述	起始地址	分区大小	备注
uboot	uboot分区	0x00002000	0x00004000	必选
misc	misc分区	0x00006000	0x00002000	烧写进入recovery模式，可选
boot_linux	内核分区	0x00008000	0x00020000	必选
recovery	recovery分区	0x00028000	0x00040000	必选
resource	resource分区	0x00068000	0x00010000	保存开机LOGO，必选
rootfs	根文件系统分区	0x00078000	-	所有剩余空间，必选

**part-fit: fit**启动方式的系统分区

配置项	描述	起始地址	分区大小	备注
uboot	uboot分区	0x00002000	0x00004000	必选
misc	misc分区	0x00006000	0x00002000	烧写进入recovery模式，可选
boot	内核分区	0x00008000	0x00020000	必选
recovery	recovery分区	0x00028000	0x00040000	必选
backup	backup分区	0x00068000	0x00010000	可选
rootfs	根文件系统分区	0x00078000	0x01c00000	必选
oem	oem分区	0x01c78000	0x00040000	可选
userdata	userdata分区	0x01cb8000	-	所有剩余空间，可选

说明：

1. 如果起始地址和分区地址都为0，则脚本会忽略此分区。
2. OTG口不接USB线、长按recovery按键，系统会从recovery分区引导，并进入紧急修复模式。
3. 烧写misc镜像，系统也会从recovery分区引导，并进入紧急修复模式。

### 3.4.4 uboot配置

配置项	描述	默认值	备注
config	uboot编译配置	rk3588-edge	configs/rk3588-edge.config

### 3.4.5 内核配置

配置项	描述	默认值	备注
version	内核版本号	5.10	目前只支持5.10内核
config	内核编译配置	rk3588_edge.config	arch/arm64/configs/rk3588_edge.config
size	内核镜像的大小，单位：M	64	auto表示自动调整大小
dtbname	内核设备树文件名	not set	不包含后缀dtb
docker	内核配置是否需要支持docker	false	内核镜像会增大
debug	芯片调试口物理地址	not set	根据芯片配置

说明：内核编译配置默认会加载rockchip\_linux\_defconfig和config配置项指定的配置。

### 3.4.6 文件系统配置

配置项	描述	默认值	备注
osname	操作系统名称	debian	
version	操作系统版本号	11	
type	桌面类型	gnome	
apturl	边缘计算系统的APT源的URL基址	<a href="http://repo.rock-chips.com/">http://repo.rock-chips.com/</a>	Debian源地址： \${apturl}/edge/debian
uuid	rootfs分区的PARTUUID值	614e0000-0000-4b53-8000-1d28000054a9	不可修改
size	rootfs镜像的大小，单位：M	auto	auto表示自动调整大小
user	操作系统的管理账号	not set	
password	管理账号的密码	not set	
relver	发布版本号	0.1.0	其值会写入/etc/edge-release

说明：配置项type支持的类型：

1. server：服务类系统，不带桌面。
2. gnome：基于wayland，原生gnome桌面系统。
3. gnome-docker：基于wayland，原生gnome桌面系统，预安装docker相关组件。
4. gnome-ros1：基于wayland，原生gnome桌面系统，预安装ros1相关组件。
5. gnome-ros2：基于wayland，原生gnome桌面系统，预安装ros2(ros2-foxy)相关组件。
6. lxde：基于xorg，LXDE轻量级桌面系统。

## 4. 镜像编译

---

### 4.1 一键编译

执行如下命令编译所有镜像，并打包update.img，保存在OUT\_DIR目录：

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

### 4.2 生成分区文件

执行如下命令生成parameter.txt，保存在OUT\_DIR目录：

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

### 4.3 编译Uboot镜像

执行如下命令编译生成MiniLoaderAll.bin和uboot.img镜像，保存在OUT\_DIR目录：

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

### 4.4 编译kernel镜像

执行如下命令编译生成内核镜像，保存在OUT\_DIR目录：

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

说明：

fit启动方式：生成boot.img和recovery.img

extlinux启动方式：生成boot\_linux.img、recovery.img和resource.img

### 4.5 制作rootfs镜像

执行如下命令编译生成rootfs.img，保存在OUT\_DIR目录：

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

1. 编译制作rootfs镜像过程中，脚本会检查OUT\_DIR目录下是否存在rootfs.img，如果存在会打印如下信息，提示用户是否继续？

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

- 如果需要删除旧的rootfs镜像重新编译制作，请输入N或n。
- 如果希望基于之前编译制作的rootfs镜像继续，请输入Y或y。

2. 编译制作rootfs镜像过程中，脚本会调用vendor/common/pre-install/install.sh和vendor/\${CHIP}/\${BOARD}/pre-install/install-board.sh脚本预安装软件包和修改系统配置信息。如果需要额外安装软件包或修改系统配置信息，请修改vendor/\${CHIP}/\${BOARD}/pre-install/install-board.sh。
3. 编译过程中，如果出现提示：Do you want to accept these changes and continue updating from this repository [y/N]，请输入y。

## 4.6 打包update镜像

执行如下命令打包所有镜像，生成update.img，保存在OUT\_DIR目录：

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

## 4.7 生成镜像签名的公私钥

执行如下命令，生成镜像签名的公私钥保存在uboot/keys目录下：(仅支持fit启动方式)

```
./edge build -s
```

## 4.8 查看编译帮助

查看支持的编译参数：

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

# 5. 烧写镜像

---

## 5.1 进入烧写模式

### 5.1.1 进入loader烧写模式

1. 连接Type-C口到电脑PC端，按住主板的V+/Recovery按键不放。
2. 开发板供电12V，若已经上电，按下复位按键。
3. 当开发板进入loader模式后，松开按键。
4. 参考《查询烧写状态》章节，确认开发板进入loader模式。

### 5.1.2 进入maskrom烧写模式

- 1. 连接Type-C口到电脑PC端，按住主板的Maskrom按键不放。
- 2. 开发板供电12V，若已经上电，按下复位按键。
- 3. 当开发板进入loader模式后，松开按键。
- 4. 参考《查询烧写状态》章节，确认开发板进入maskrom模式。

## 5.2 查询烧写状态

### 5.2.1 Linux主机查询

执行如下命令查询烧写状态：

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

- none: 表示开发板未进入烧写模式。
- loader: 表示开发板进入loader烧写模式。
- maskrom: 表示开发板进入maskrom烧写模式。

### 5.2.2 Windows主机查询

双击打开tools\RKDevTool\_Release\_v2.84目录下的RKDevTool.exe，界面显示：

- 没有发现设备（如图1-1所示）：表示开发板未进入烧写模式。
- 发现一个LOADER设备（如图1-2所示）：表示开发板进入loader烧写模式。
- 发现一个MASKROM设备（如图1-3所示）：表示开发板进入maskrom烧写模式。

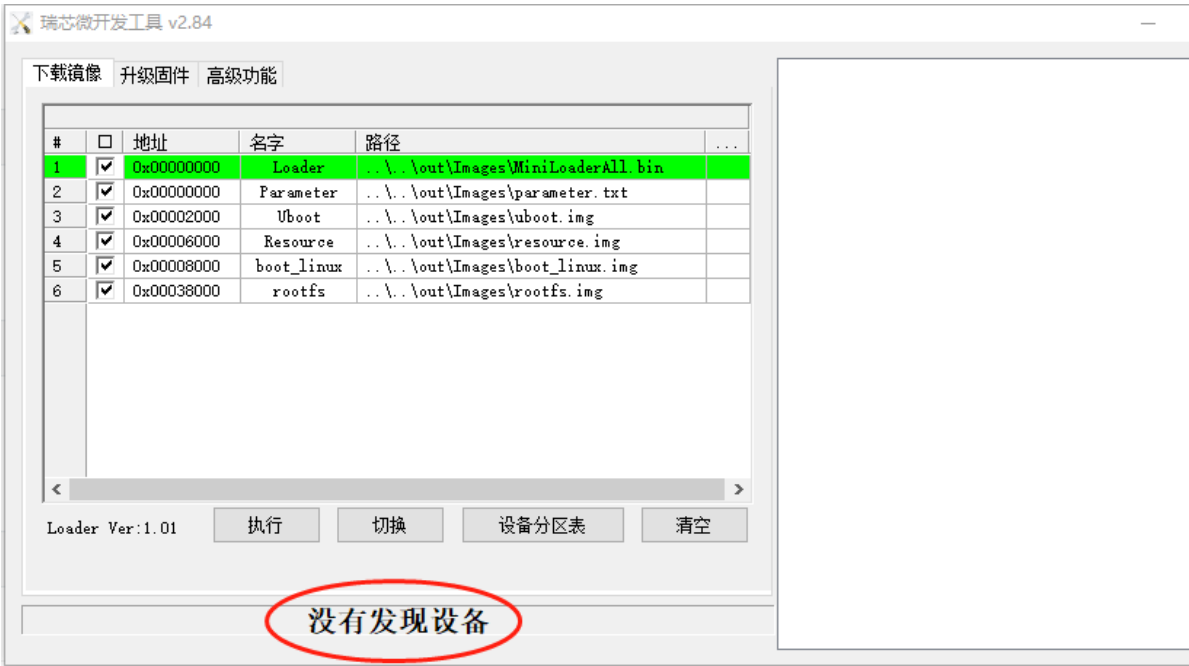


图1-1：没有发现设备

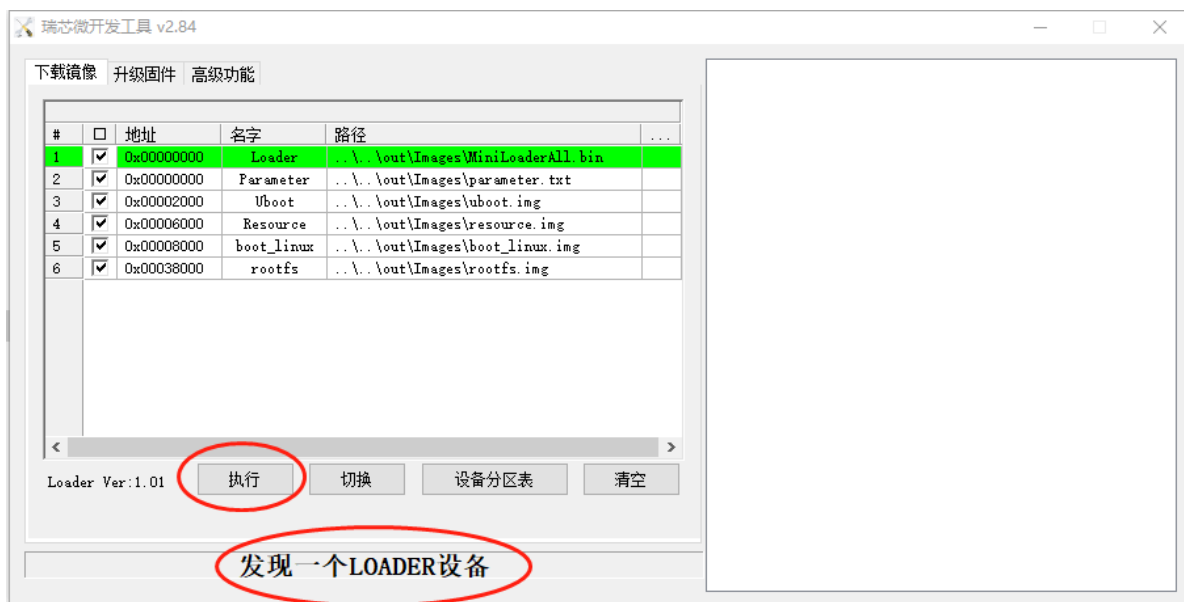


图1-2：发现一个LOADER设备

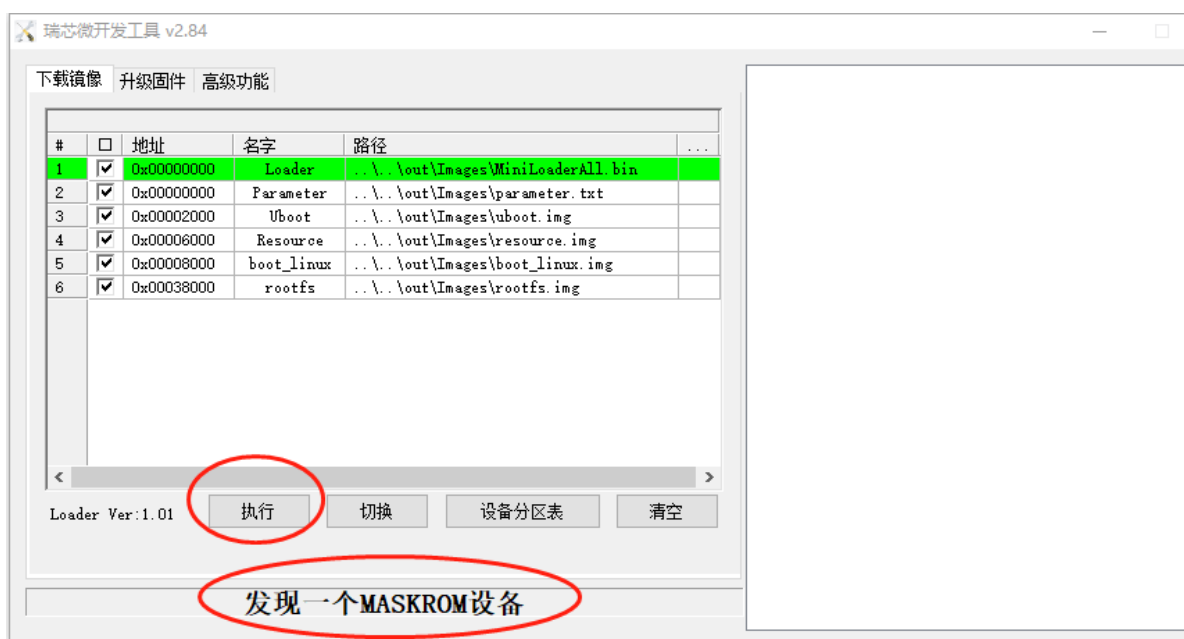


图1-3：发现一个MASKROM设备

## 5.3 LINUX主机烧写镜像

### 5.3.1 烧写所有镜像

烧写所有镜像

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

### 5.3.2 烧写uboot镜像

烧写镜像：MiniLoaderAll.bin，uboot.img和parameter.txt

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

### 5.3.3 烧写内核镜像

烧写内核镜像：

1. extlinux启动模式会烧写boot\_linux.img、recovery.img和resource.img
2. fit启动模式会烧写boot.img和recovery.img

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

### 5.3.4 烧写文件系统镜像

烧写镜像：rootfs.img

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

### 5.3.5 烧写misc镜像

烧写misc镜像： misc.img

```
./edge flash -m
```

### 5.3.6 烧写oem镜像

烧写oem镜像： oem.img (仅支持fit启动方式)

```
./edge flash -o
```

### 5.3.7 烧写userdata镜像

烧写userdata镜像： userdata.img (仅支持fit启动方式)

```
./edge flash -d
```

### 5.3.8 查看烧写帮助

查看支持的烧写参数：

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

## 5.4 Windows主机烧写镜像

1. 双击打开tools\RKDevTool\_Release\_v2.84目录下的RKDevTool.exe。
2. 在工具的空白处点击右键，选择弹出菜单的"导入配置"，如下图所示：

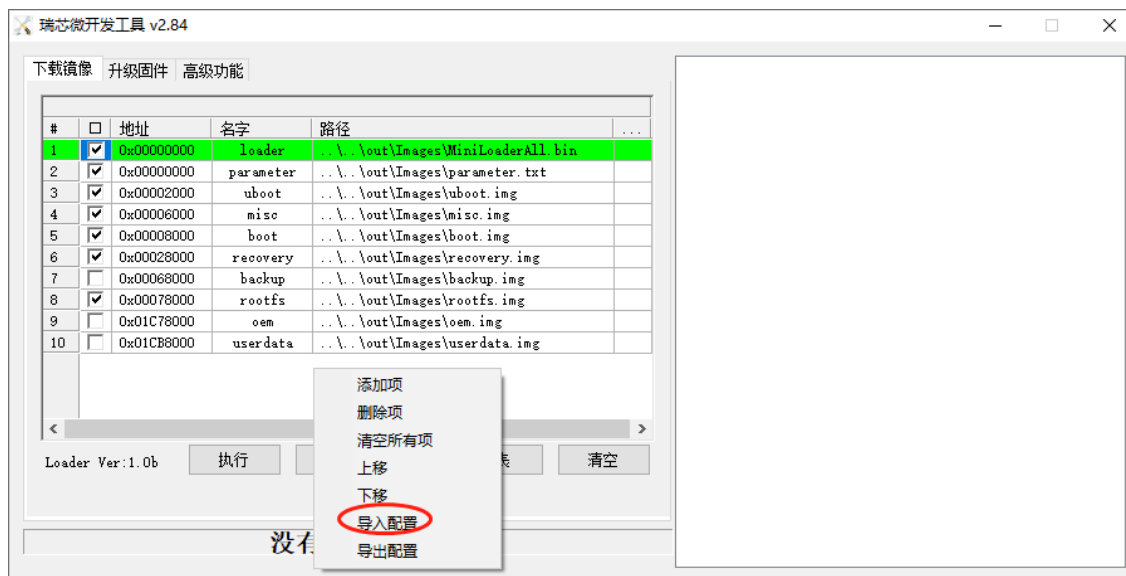
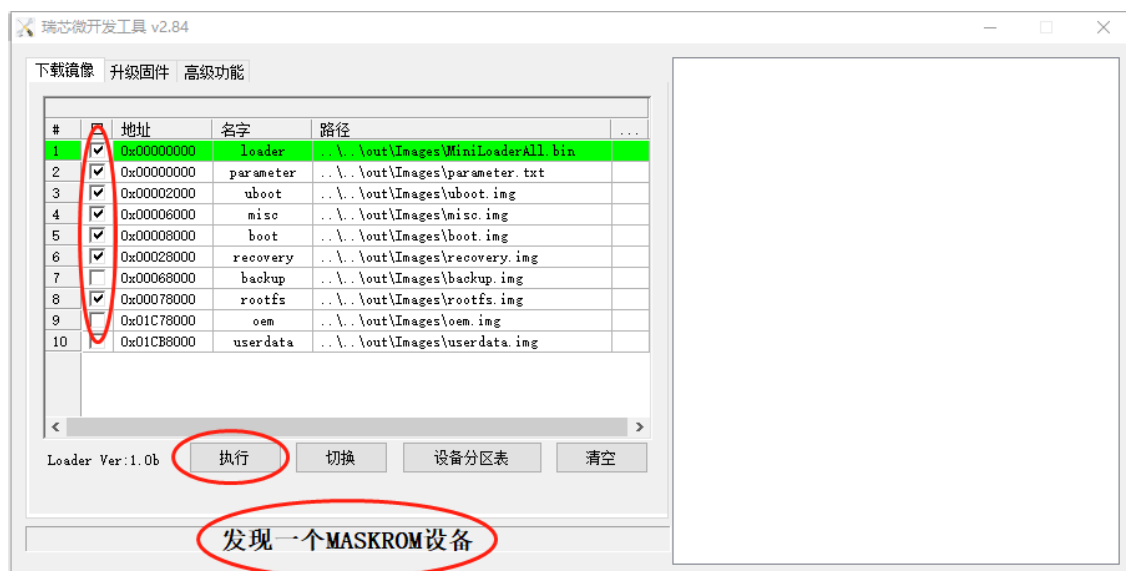


图2-1：导入配置

3. 选择要烧写的分区配置
  - config-extlinux.cfg: extlinux启动方式
  - config-fit.cfg: fit启动方式
4. 确认开发板已经进入loader或者maskrom烧写模式。
5. 打勾选择需要烧写的镜像。

注意：**Loader**和**Parameter**选项建议打勾选择，其他选项根据需要打勾选择。
6. 点击“执行”按钮，开始烧写固件（如图2-1所示）。





## 6. 串口调试

## 6.1 串口连接

用USB线连接主机的USB host口和开发板的调试口(通常开发板上的调试口边上有标有类似"DEBUG"或"UART TO USB"的丝印)。

## 6.2 Windows主机调试

### 6.2.1 获取端口号

打开设备管理器获取调试串口的端口号，如图3-1所示：

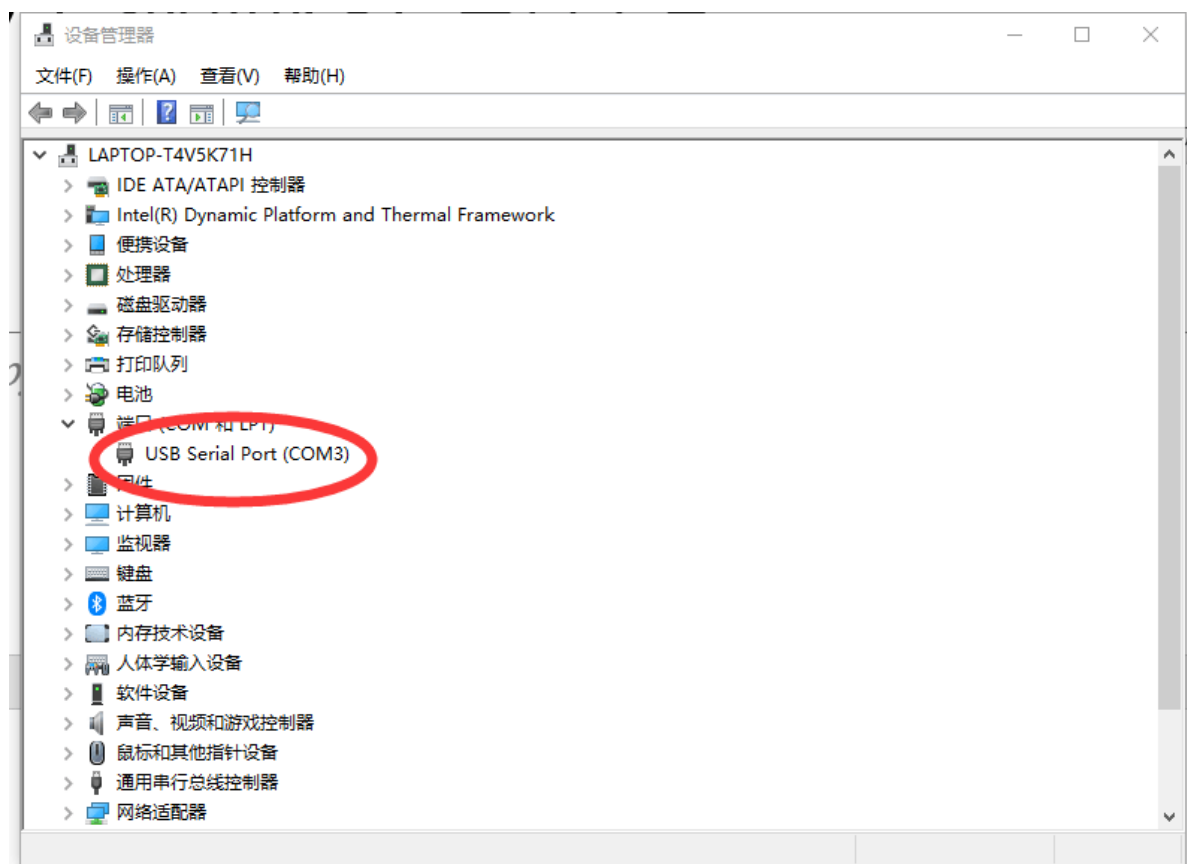


图3-1: 获取调试端口号

### 6.2.2 配置调试串口信息

打开串口工具SecureCRT, 点击"快速连接"按钮, 打开调试串口配置界面如图3-2和图3-3所示:

1. Port: 选择设备管理器显示的端口号
2. 波特率: 1500000
3. 禁止流控: 不勾选RTS/CTS

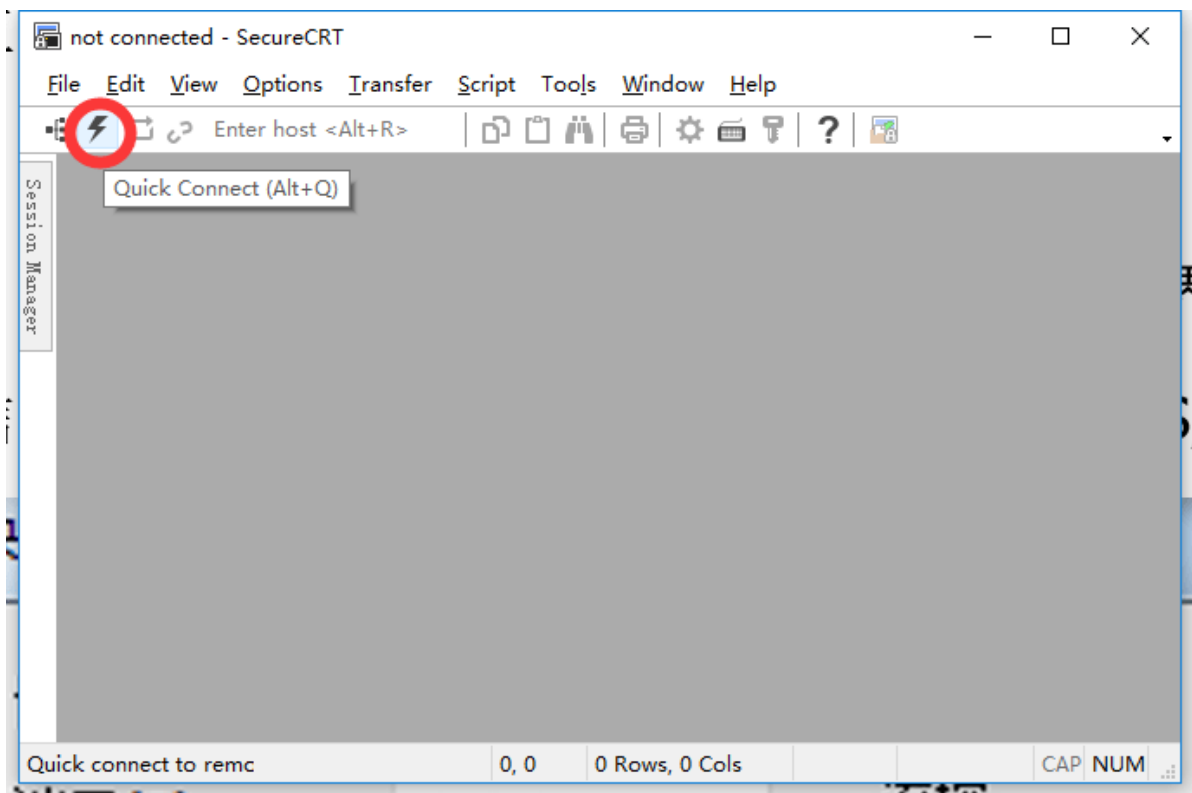


图3-2: SecureCRT

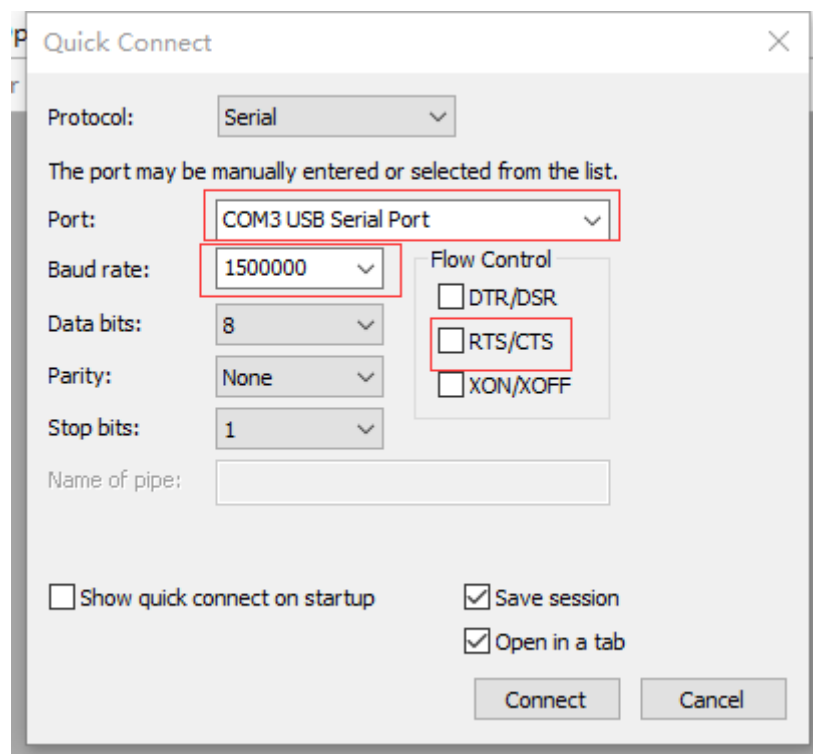


图3-3: 配置调试串口信息

## 6.3 Linux主机调试

### 6.3.1 安装minicom

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

### 6.3.2 配置调试窗口信息

按如下步骤，配置保存串口信息（此步骤执行一次即可）：

1. 执行如下命令打开minicom

```
sudo minicom -s
```

2. 进入串口设置界面：输入CTRL-A Z

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

3. 端口设置：选择"Serial port setup"
4. 设置串口设备：输入"A"，填入"/dev/ttyUSB0"，然后按回车确定
5. 禁止流控：输入"F"，按回车确定
6. 设置波特率：输入"E"，再输入"A"直到显示"Current 1500000 8N1"，然后按回车确认
7. 配置完成后，界面显示

```
+-----+
| 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. 退出端口设置：按回车
9. 保存配置：选择"Save setup as dfl"
10. 退出设置：选择"Exit"

### 6.3.3 运行minicom

```
sudo minicom
```

## 7. SD卡启动

## 7.1 制作SD启动卡

1. 准备一张SD卡（容量不小于16G），通过SD卡读卡器插到windows电脑下USB Host口。
2. 双机打开tools\SDDiskTool\_v1.69\SD\_Firmware\_Tool.exe，如图4-1所示：

按界面提示选择磁盘设备、功能模式和update.img的路径，然后点击“开始创建”按钮，开始制作SD卡。



图4-1：SDDisk界面

## 7.2 从SD卡启动

将SD卡从RK3588开发板的SD卡槽重启开机启动，即可从SD卡启动。

## 8. 安全启动

安全启动（secure boot）操作步骤如下：

1. 参考《编译配置》章节修改：
  - 将配置项启动方式bootmode的值修改为：fit
  - 修改配置项secucreboot的子项enable为：true
  - 根据需要修改子项rollback和burnkey。
2. 如果已有公私钥，将公私钥拷贝到uboot/keys目录，否则参考《镜像编译》章节生成签名公私钥。
3. 参考《镜像编译》章节编译生成带签名的镜像。
4. 参考《烧写镜像》章节烧写镜像。

## 9. 常见问题

---

- 问题：制作根文件系统出错，提示出错信息：`chroot: failed to run command '/pre-install/install.sh': Exec format error.`

解决办法：安装qemu, qemu-user-static

- 问题：编译制作rootfs镜像过程中，系统异常或按CTRL+C强制退出，导致镜像和设备节点没有卸载。
- 解决办法：执行如下命令强制卸载：

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