

君正®

## T32 Zeratul 烧录说明

---

文档版本 V1.0.0

发布日期 2024-06-20



北京君正集成电路股份有限公司  
Ingenic Semiconductor Co., Ltd.

版权所有 © 北京君正集成电路股份有限公司 2005-2024 。保留一切权利

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。

## 商标声明



、君正和其他君正商标均为北京君正集成电路股份有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

君正提醒您，本文件夹所提供的交付件及数据仅供您开发参考。在使用本文件夹所提供的交付件及数据前，请您务必仔细阅读并透彻理解本声明。您可以选择不使用本文件夹中的交付件及数据，但如果您使用，您的使用行为将被视为对本声明全部内容无条件且不可撤销的认可。

本文件夹所提供的交付件及数据仍然是“按现状”提供，不附有任何种类的陈述与保证，在法律许可的最大范围内，君正声明免除所有明示或默示的陈述与保证，包括但不限于有关适销、非侵权和适用于某种特定用途的陈述与保证。本目录中所提供的交付件及数据在不同的使用环境下可能存在一定的偏差，您因使用本文件夹中所提供的交付件及数据而导致的所有设计、开发错误，或其他一切因使用本文件夹中所提供的交付件及数据而导致的意外、疏忽、侵权及其造成的全部损失（包括但不限于直接的、间接的、后果性的或附带性的损失），其后果由您自行承担，君正对其概不负责，亦不承担任何法律责任。

北京君正集成电路股份有限公司

地址：北京市海淀区西北旺东路 10 号院东区 14 号楼君正大厦

电话：**(86-10)56345000**

传真：**(86-10)56345001**

<http://www.ingenic.com.cn/>

# 前言

## 概述

本文为 T32 固件烧录说明，帮助使用者快速了解 T 系列开发板固件烧录操作使用。

## 产品版本

与本文档相对应的产品版本如下。

产品名称
T32

## 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

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

日期	版本	修订章节
2024-06-20	V1.0.0	第一次正式版本发布

---

# 目录

---

<b>1 启动卡烧录步骤</b> .....	<b>1</b>
1.1 制作启动卡 .....	1
1.2 使用启动卡烧录完整固件 .....	4
1.3 使用启动卡烧录各个分区内容 .....	7
<b>2 启动卡烧录步骤</b> .....	<b>9</b>
2.1 使用 tftp 烧录 .....	9
2.1.1 使用 tftp 烧录完整固件 .....	9
2.1.2 使用 tftp 烧录各个分区内容 .....	10
2.2 使用 flashcp 烧录 .....	11
2.2.1 nfs、tftp 方式 .....	11
2.2.2 进入系统挂载 sd 卡烧录 .....	12
<b>3 USB 烧录步骤</b> .....	<b>13</b>
3.1 USB 烧录常见异常 .....	17
3.2 USB 烧录分区 .....	18
<b>4 串口烧录步骤</b> .....	<b>21</b>

---

## 图片目录

---

图 1	sdk 格式变换 .....	2
图 2	卡起 boot 操作流程 .....	4
图 3	卡起烧录 .....	5
图 4	nor flash 烧录 .....	6
图 5	window usb 烧录准备 .....	13
图 6	打开烧录配置 .....	15
图 7	烧录分区选择 .....	16
图 8	flash 型号选择 .....	17
图 9	开始烧录 .....	17
图 10	串口烧录 .....	22
图 11	minicom 配置 .....	24

# 1 启动卡烧录步骤

## 1.1 制作启动卡

注意：

确定插入你的电脑上 TF 卡对应的设备节点文件是不是 sdb(有可能是 sdc 或者 sdd)；如果对应错了可能会破坏电脑的硬盘文件系统

对于第一次烧录 Nor Flash，或者没有网络的情况来说，需要从 SD 卡启动或者USB烧录软件，并将 uboot 烧录到 Nor Flash 中。之后的烧录就可以通过 tftp 进行。

Step1, 格式化 sd 卡

此步骤只需执行一次，若不更新卡起 uboot 则不需要再次执行此步骤。经过此步骤处理的 SD 卡可当作正常卡使用

Step2, 将 SD 卡插入电脑卸载 sd 卡

```
sudo umount /media/user/*
```

Step3, 将 SD 卡重新分区输入如下命令：

```
sudo fdisk /dev/sdb
```

```
> o --创建一个新的分区表
> n --添加一个分区 n
--此时会提示：
Partition type:
  p primary (0 primary, 0 extended, 4 free)
```

```

e extended
Select (default p):
Using default response p
--这里回车默认是 p, 创建主分区
分区号 (1-4, 默认为 1):
将使用默认值 1
--这里回车默认创建 1 号分区
起始 sector (2048-15130623, 默认为 2048):
将使用默认值 2048
Last sector, +扇区 or +size{K,M,G} (2048-15130623, 默认为 15130623):
将使用默认值 15130623
--2048 号扇区在 1MB 的位置, 给 uboot 空出了空间
最后一步, 输入 w, 向 SD 卡中写入分区表
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.

```

图 1 sdk 格式变换

```

:~$ sudo fdisk /dev/sdb
欢迎使用 fdisk (util-linux 2.31.1)。
更改将停留在内存中, 直到您决定将更改写入磁盘。
使用写入命令前请三思。

命令(输入 m 获取帮助): o ← Enter o
创建了一个磁盘标识符为 0xd42b6e8e 的新 DOS 磁盘标签。

命令(输入 m 获取帮助): n ← Enter n
分区类型
  p 主分区 (0个主分区, 0个扩展分区, 4空闲)
  e 扩展分区 (逻辑分区容器)
选择(默认 p): p ← Enter p
分区号 (1-4, 默认 1): ← Enter
第一个扇区 (2048-15745023, 默认 2048): ← Enter
上个扇区, +sectors 或 +size{K,M,G,T,P} (2048-15745023, 默认 15745023): ← Enter

创建了一个新分区 1, 类型为“Linux”, 大小为 7.5 GiB。

命令(输入 m 获取帮助): w ← Enter w
分区表已调整。
将调用 ioctl() 来重新读分区表。
正在同步磁盘。

```

Step 4 执行 `sync`

Step 5 格式化 sd 卡为通用的 VFat 文件系统，PC 终端输入如下命令

```
sudo mkfs.vfat /dev/sdb1
```

Step 6 编译卡启动 uboot, 并烧录进 sd 卡的 17KB 偏移处

相应卡启动 uboot 已经编译好，进入 firmware/sdcard\_uboot/ 目录找到 SOC 对

应串口的卡启动 uboot，T32ZX 对应 T32zx\_sdcard\_uboot\_msc0\_uart1.bin。

```
sudo dd if=u-boot.bin of=/dev/sdb bs=1024 seek=17
```

操作步骤如下：

图 2 卡起 boot 操作流程

```
~$ sudo umount /media/user/62BC-2357
~$ sudo fdisk /dev/sdb

欢迎使用 fdisk (util-linux 2.31.1)。
更改将停留在内存中，直到您决定将更改写入磁盘。
使用写入命令前请三思。

命令(输入 m 获取帮助): o
创建了一个磁盘标识符为 0x87b5eb1e 的新 DOS 磁盘标签。

命令(输入 m 获取帮助): n
分区类型
  p 主分区 (0个主分区, 0个扩展分区, 4空闲)
  e 扩展分区 (逻辑分区容器)
选择 (默认 p): p
分区号 (1-4, 默认 1):
第一个扇区 (2048-15745023, 默认 2048):
上个扇区, +sectors 或 +size{K,M,G,T,P} (2048-15745023, 默认 15745023):

创建了一个新分区 1, 类型为“Linux”, 大小为 7.5 GiB。
分区 #1 包含一个 vfat 签名。

您想移除该签名吗? 是[Y]/否[N]: n

命令(输入 m 获取帮助): w

分区表已调整。
将调用 ioctl() 来重新读分区表。
正在同步磁盘。

~$
~$ sync
~$ sudo mkfs.vfat /dev/sdb1
mkfs.fat 4.1 (2017-01-24)
~$ sudo dd if=u-boot-with-spl.bin of=/dev/sdb bs=1024 seek=17
记录了226+1 的读入
记录了226+1 的写出
231792 bytes (232 kB, 226 KiB) copied, 0.113483 s, 2.0 MB/s
~$
```

## 1.2 使用启动卡烧录完整固件

制作启动卡进入 boot 手动烧录，按住 BOOTSEL 按键，然后上电进入卡启动 uboot，如下图，通过输入以下命令进行一个完整固件的烧录。

图 3 卡起烧录完整固件

```
U-Boot SPL 2013.07-00035-ge57a219dc-dirty (Aug 08 2022 - 15:39:50)
Timer init

apll_freq = 1300000000
mpll_freq = 1200000000
vppll_freq = 1188000000

cclk = 1300000000
l2clk = 650000000
h0clk = 240000000
h2clk = 240000000
pclk = 120000000
cpu clk source: apll
ddr clk source: mpll
sdram init start
DDR clk rate 600000000
board_init_r
image entry point: 0x80100000
Board: ISVP (Ingenic XBurst T41 SoC)
DRAM: ddr size is 0x80000000
128 MiB
Top of RAM usable for U-Boot at: 88000000
Reserving 431k for U-Boot at: 87f94000
Reserving 32800k for malloc() at: 85f8c000
Reserving 32 Bytes for Board Info at: 85f8bfe0
Reserving 124 Bytes for Global Data at: 85f8bf64
Reserving 128k for boot params() at: 85f6bf64
Stack Pointer at: 85f6bf48
Now running in RAM - U-Boot at: 87f94000
MMC: MSC: 0
Data transfer: serial
Net: Jz4775-9161
Hit any key to stop autoboot: 0
isvp_t41#
isvp_t41#
isvp_t41#
isvp_t41#
isvp_t41#
```

Step 1:第三个参数是大小,这里清了 16MB(注:这里必须擦,否则文件系统可能报错)

```
mw.b 0x80600000 0xff 0x1000000
```

Step 2:ls SD 卡中的文件

```
fatls mmc 0
```

Step 3:load 文件到内存

```
fatload mmc 0 0x80600000 zrt_prj007zn_gc2063.bin
```

Step 4:然后，将内存中的数据写入 Nor Flash

```
sf0 probe;sf0 erase 0x0 0x1000000;sf0 write 0x80600000 0x0 0x1000000
```

图 4 nor flash 烧录

```
isvp_t41# mw.b 0x80600000 0xff 0x1000000
isvp_t41# fatls mmc 0
    10700    baize_test
            .trash-1000/
    1302    main.c
    1627252    uclibc_gdbserver
    8816    pdmac
    78828    iperf-mips-720
    342848    iperf3_static
    2215654    uimage.lzma
    11159510    uimage.lzo
```

```
isvp_t41# fatload mmc 0 0x80600000 zrt_soc_t41zn_jxf23_fw.bin
reading zrt_soc_t41zn_jxf23_fw.bin
13123584 bytes read in 571 ms (21.9 MiB/s)
isvp_t41#
isvp_t41#
isvp_t41# sf0 probe;sf0 erase 0x0 0x1000000;sf0 write 0x80600000 0x0 0x1000000
SF: Detected GD25Q128C, manufacturer id: c8
--->probe spend 4 ms
SF: 16777216 bytes @ 0x0 Erased: OK
--->erase spend 14477 ms
SF: 16777216 bytes @ 0x0 Written: OK
--->write spend 20493 ms
isvp_t41#
isvp_t41#
isvp_t41#
```

## 1.3 使用启动卡烧录各个分区内容

这里以常规的 Zeratul T32 系统分区为例：

如下各分区起始地址：

Start		end	
0x0	~	0x40000	boot
0x40000	~	0xc0000	tag
0xc0000	~	0x4c0000	kernel
0x4c0000	~	0x8c0000	rootfs
0x8c0000	~	0xbc0000	recovery
0xbc0000	~	0xec00000	system

进入 boot 命令行后, 输入如下命令：

烧录 uboot:

```
fatload mmc 0 0x80600000 boot.bin;sf0 probe;sf0 erase 0x0 0x40000;sf0 write 0x80600000 0x0 0x40000
```

烧录 tag:

```
fatload mmc 0 0x80600000 tag.bin;sf0 probe;sf0 erase 0x40000 0x80000;sf0 write 0x80600000 0x40000 0x80000
```

烧录 kernel:

```
fatload mmc 0 0x80600000 uimage.lzo;sf0 probe;sf0 erase 0xc0000 0x400000;sf0 write 0x80600000 0xc0000 0x400000
```

烧录 rootfs 根文件系统

```
fatload mmc 0 0x80600000 rootfs_camera.cpio.lzo;sf probe;sf erase 0x4c0000 0x400000;sf write 0x80600000 0x4c0000 0x400000
```

烧录 recovery

```
fatload mmc 0 0x80600000 recovery.bin;sf probe;sf erase 0x8c0000 0x300000;sf write 0x80600000 0x8c0000 0x300000
```

烧录 system 分区

```
fatload mmc 0 0x80600000 system.bin;sf probe;sf erase 0xbc0000 0x300000;sf write  
0x80600000 0xbc0000 0x300000
```

## 2 启动卡烧录步骤

### 2.1 使用 tftp 烧录

先通过网络 TFTP 将文件 load 到内存上,再烧录到 flash 中。前提：硬件：开发板上网卡；板子通过网线连接到路由器或者交换机上，PC 也连到该路由器或者交换机上，并且处于同一网段上；软件：PC 端安装并设置好 TFTP 服务，把相应的 u-boot 等文件放到 TFTP 根目录下；uboot 中支持网卡驱动并且支持相应的 TFTP 操作。方法：在 uboot 中，执行 `$ tftpboot mem_addr file_name`；就可以将文件 `file_name` 传送到 Uboot 的内存地址 `mem_addr` 中了。

#### 2.1.1 使用 tftp 烧录完整固件

Step 1:第三个参数是大小，这里清了 16MB(注：这里必须擦，否则文件系统可能报错)

```
mw.b 0x80600000 0xff 0x1000000
```

Step 2:tftp load 文件到内存

```
tftpboot 0x80600000 ZRT_PRJ007ZN_GC2063.bin
```

Step 3:然后，将内存中的数据写入 Nor Flash

```
sf0 probe;sf0 erase 0x0 0x1000000;sf0 write 0x80600000 0x0 0x1000000
```

## 2.1.2 使用 tftp 烧录各个分区内容

这里以常规的 Zeratul T32 系统分区为例：

如下各分区起始地址：

Start		end	
0x0	~	0x40000	boot
0x40000	~	0xc0000	tag
0xc0000	~	0x4c0000	kernel
0x4c0000	~	0x8c0000	rootfs
0x8c0000	~	0xbc0000	recovery
0xbc0000	~	0xec00000	system

进入 boot 命令行后, 输入如下命令：

烧录 uboot:

```
tftpboot 0x80600000 boot.bin;sf0 probe;sf0 erase 0x0 0x40000;sf0 write 0x80600000 0x0 0x40000
```

烧录 tag:

```
tftpboot 0x80600000 tag.bin;sf0 probe;sf0 erase 0x40000 0x80000;sf0 write 0x80600000 0x40000 0x80000
```

烧录 kernel

```
tftpboot 0x80600000 uimage.lzo;sf0 probe;sf0 erase 0xc0000 0x400000;sf0 write 0x80600000 0xc0000 0x400000
```

烧录 rootfs 根文件系统

```
tftpboot 0x80600000 rootfs_camera.cpio.lzo;sf probe;sf erase 0x4c0000 0x300000;sf write 0x80600000 0x4c0000 0x400000
```

烧录 recovery

```
tftpboot 0x80600000 recovery.bin;sf probe;sf erase 0x8c0000 0x300000;sf write 0x80600000 0x8c0000 0x300000
```

烧录 system 分区

```
tftpboot 0x80600000 system.bin;sf probe;sf erase 0xbc0000 0x300000;sf write 0x80600000 0xbc0000 0x300000
```

## 2.2 使用 flashcp 烧录

大部分的烧录可以在进入 shell 后进行。方法为：使用 sd 卡存放烧录固件或者挂载 nfs 或者通过 tftp 等方式将文件 download 到/tmp/下。烧写到分区 flashcp <src\_file> <dest\_partition>, 其中分区为/dev/mtdX。X 代表分区数, 例如 uboot 分区是 0, tag 分区是 1, 以此类推。

### 2.2.1 nfs、tftp 方式

创建目录:

```
mkdir -p /tmp/nfs
```

挂载主机目录:

```
mount -o nolock 192.168.1.100:/nfsroot /tmp/nfs
```

烧录 uboot 分区:

```
flashcp /tmp/nfs/boot.bin /dev/mtd0
```

或者烧录 tag 分区

```
flashcp /tmp/nfs/tag.bin /dev/mtd1
```

或者烧录 kerner 分区

```
flashcp /tmp/nfs/uImage.jz1zma /dev/mtd2
```

以此类推

## 2.2.2 进入系统挂载 sd 卡烧录

Step 1、基础系统挂载 sd 卡：

```
insmod_vfat  
insmod_mmc
```

Step 2、烧录分区按如上命令执行

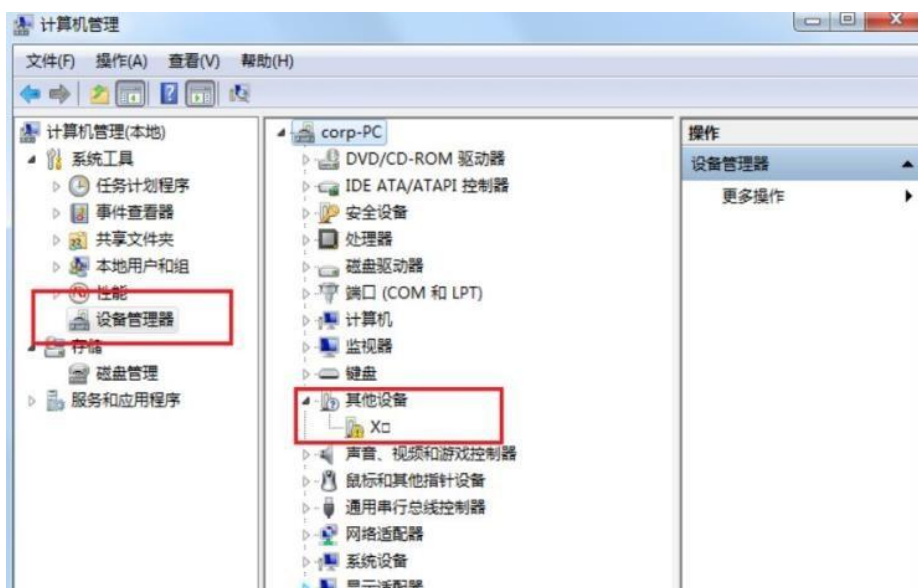
Step 3、烧录完成输入命令 `sync`

## 3 USB 烧录步骤

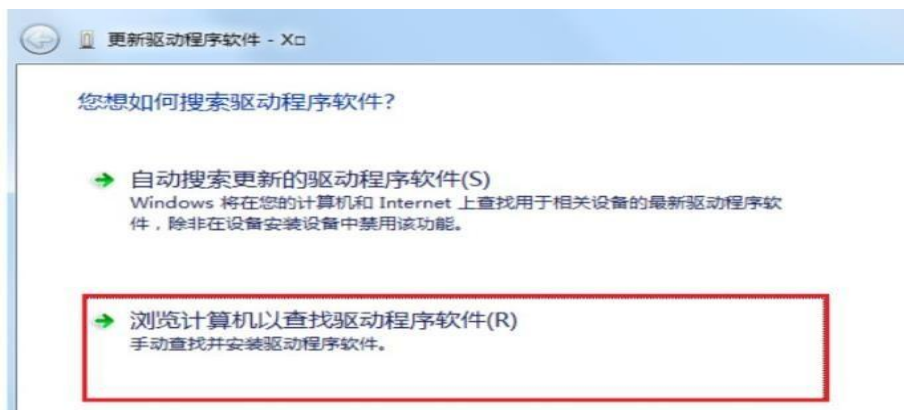
USBCloner 有 Linux 和 Windows 版本。Linux 下使用 USBCloner 不需要安装驱动。在 Windows 8 以上版本安装驱动时，首先需要配置系统“禁用驱动程序强制签名”。具体步骤查看《win10 禁用数字签名方法》文档。

板子进入烧录模式（USB 接口通过 USB 数据线连接到 PC，按住 BOOTSEL 按钮，重新上电），PC 上会自动搜索驱动安装，并且提示驱动安装失败。右键点击“我的电脑”进入“管理”，在设备管理器中的其他设备下可以找到当前没有成功安装 USB 驱动的设备，如下图。

图 5 window usb 烧录准备



在当前 USB 设备上右键点击，选择“更新驱动程序软件”弹出如下图



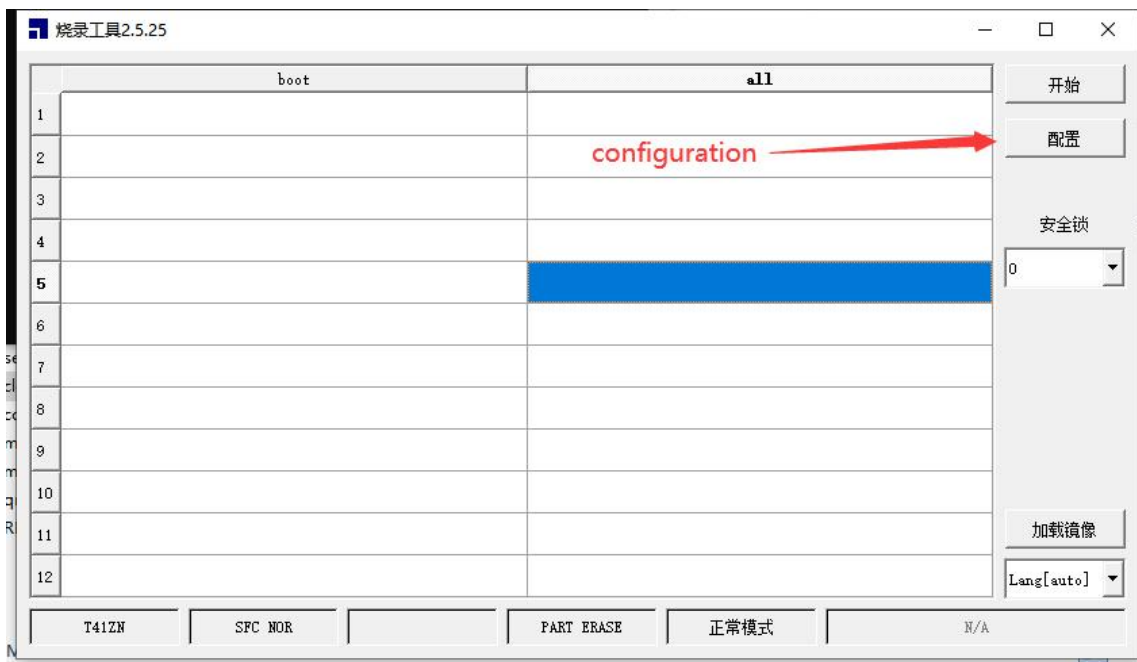
选择“浏览计算机以查找驱动程序软件”，点击“浏览”然后在所下载的烧录工具解压目录中选择驱动所在目录 cloner-win32-driver，安装成功 设备管理器可以看到如下界面：



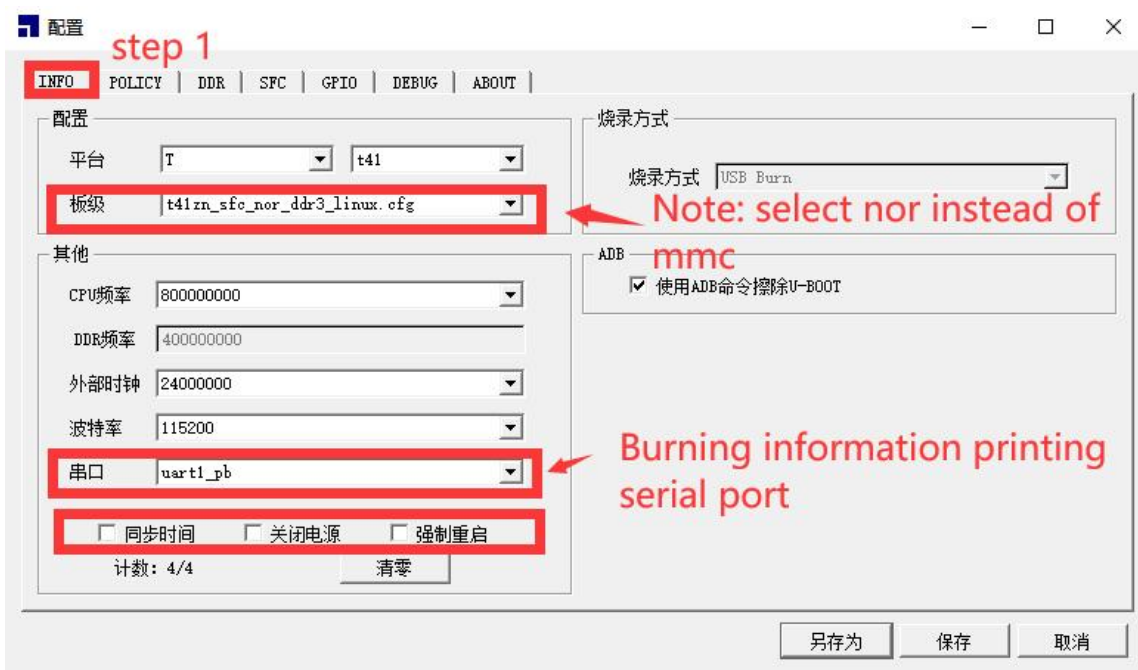
以上正确后，就可以打开 USB 烧录工具，按照如下配置，以 T41ZN 为例：

#### Step 1、打开烧录配置

图 6 打开烧录配置

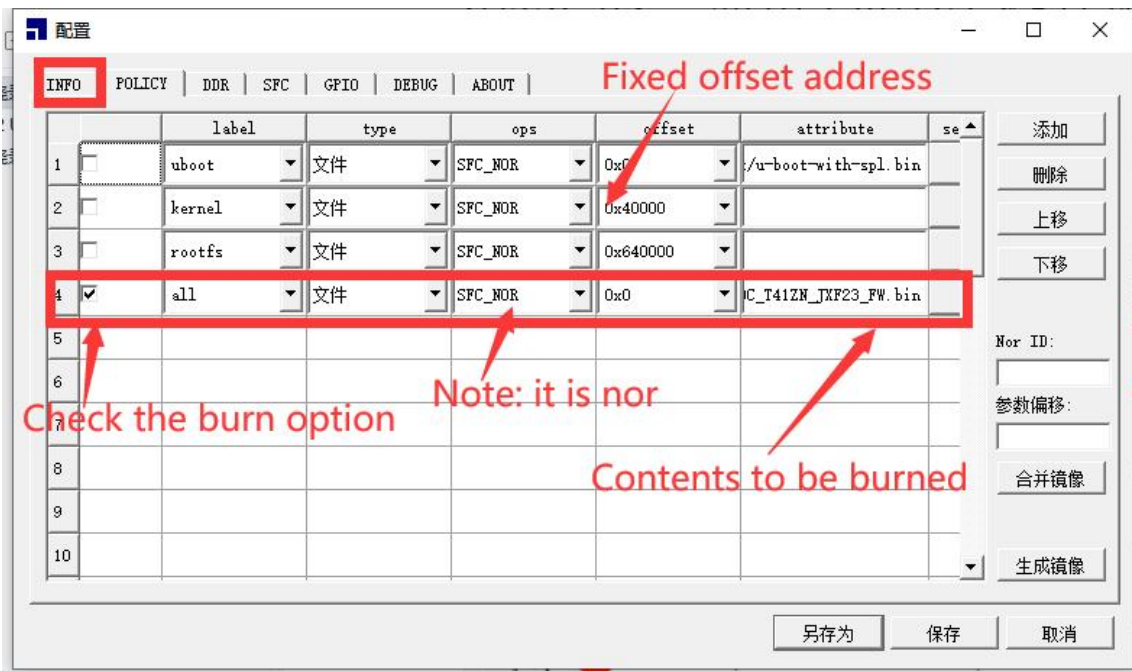


Step 2、设置烧录配置

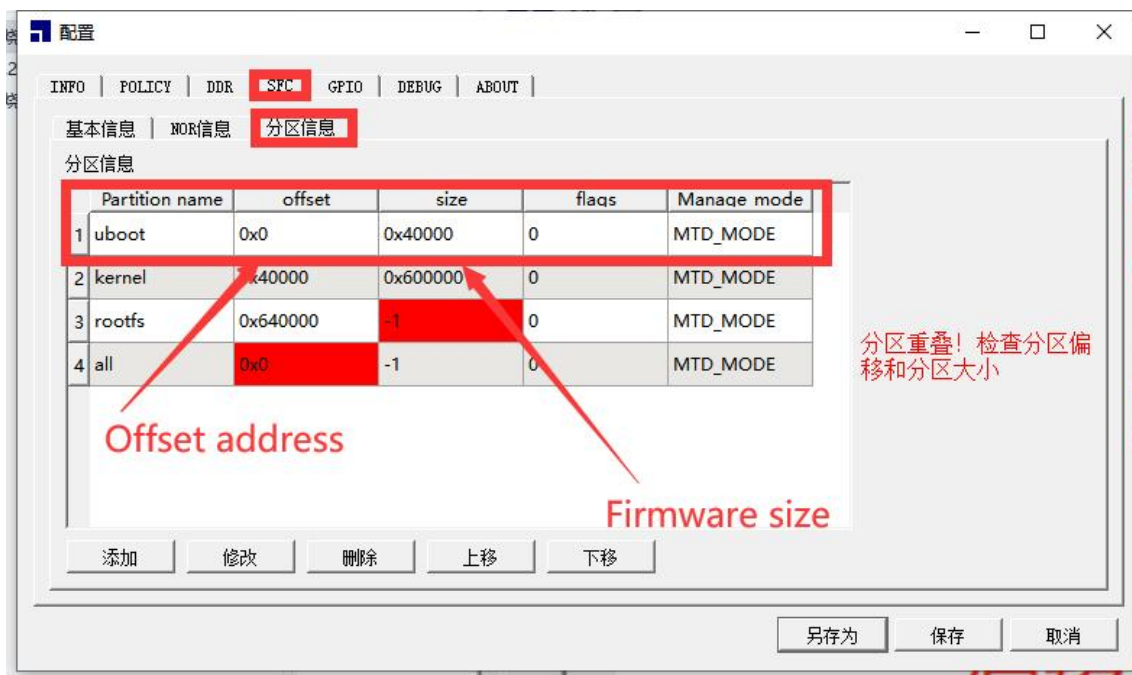


Step 3、 点击 POLICY 选择烧录分区如下：

图 7 烧录分区选择

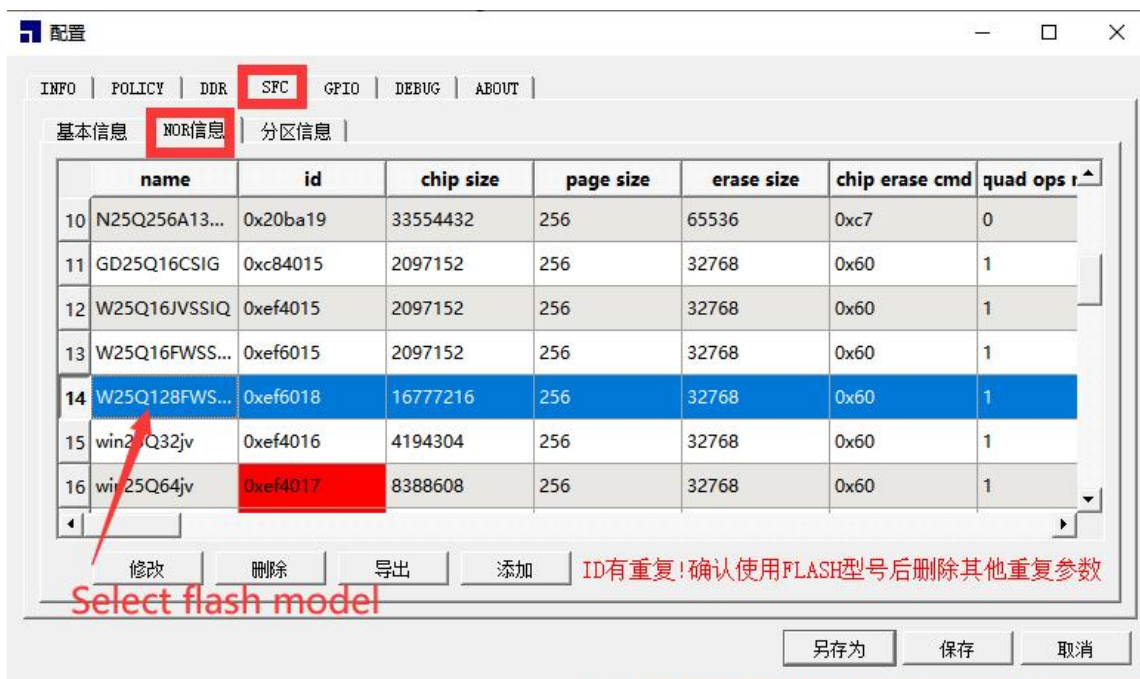


Step 4、 配置烧录分区信息



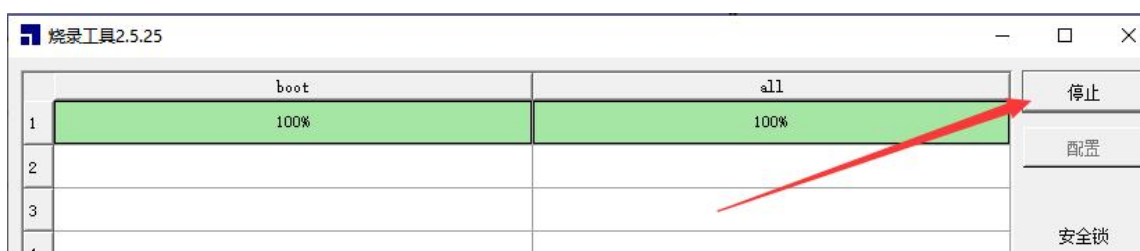
Step 5、SFC NOR 信息选择 flash 型号（默认可以忽略，自动识别）

图 8 flash 型号选择



Step6、设置完成点击保存，保存配置信息后，点击开始界面会显示进度提示如下图所示，烧录完成点击停止（注意先点击开始，按住 bootse1 按键，然后上电）

图 9 开始烧录



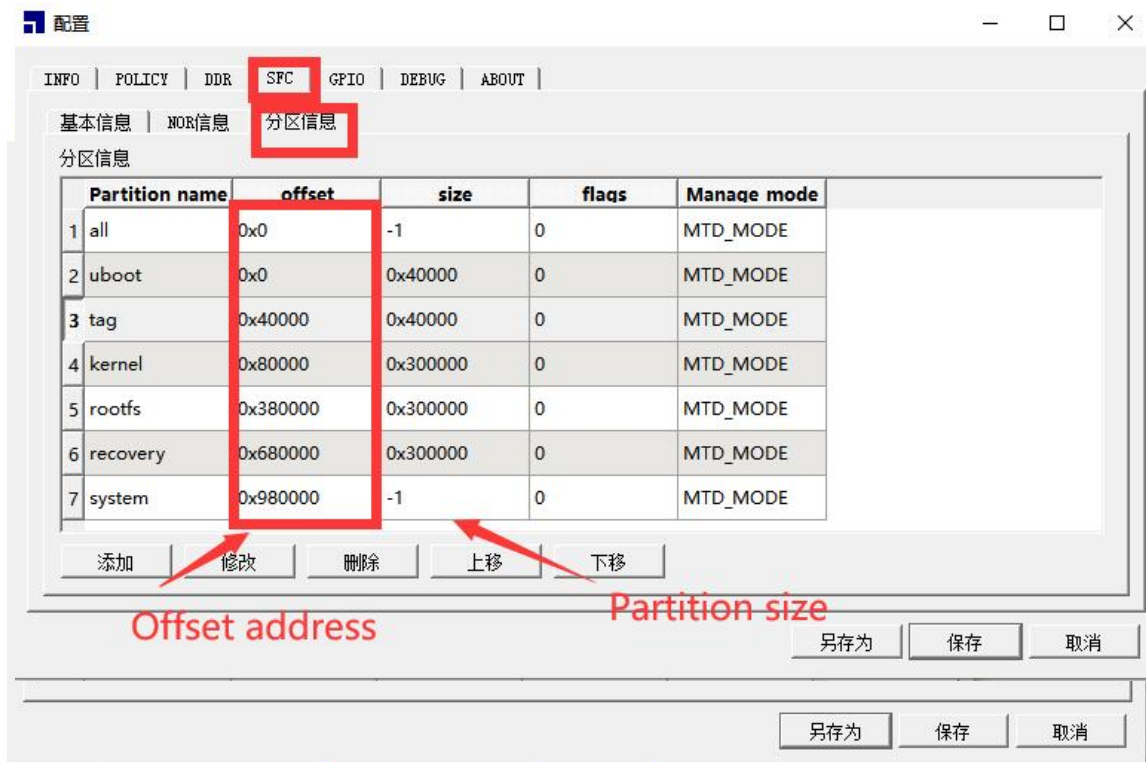
### 3.1 USB 烧录常见异常

Boot 阶段 40% —— spl 初始化异常。

Boot 阶段 60% —— uboot 运行异常。

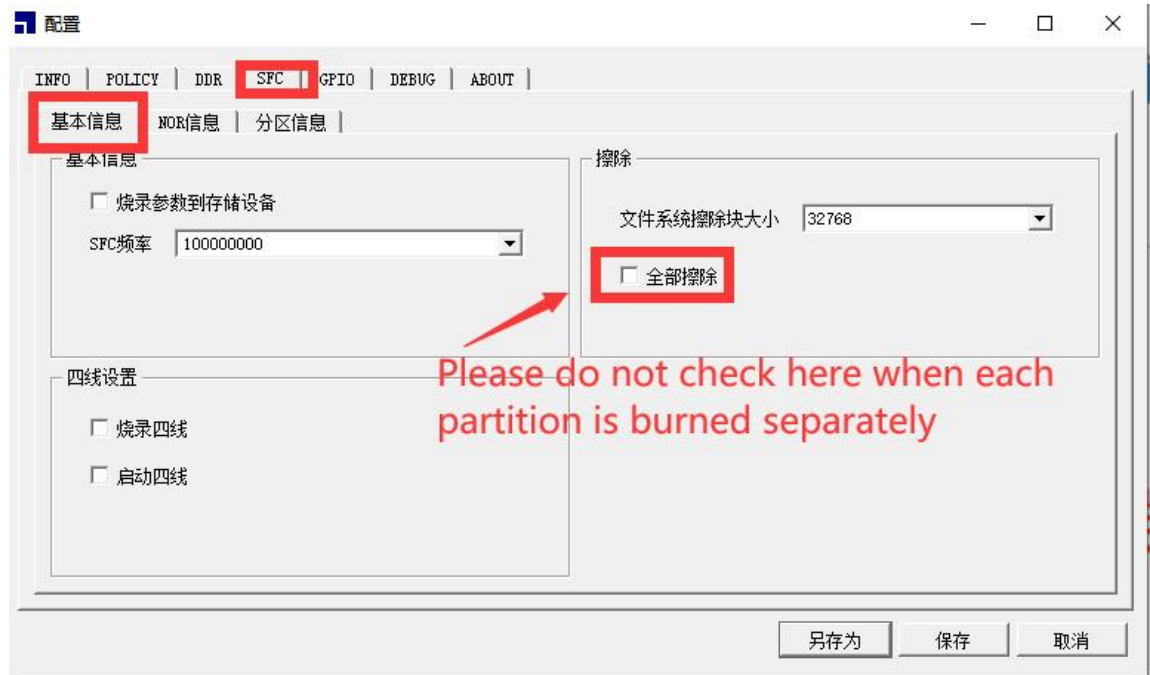
烧录阶段 0% —— flash 擦除或写入错误

## 3.2 烧录分区



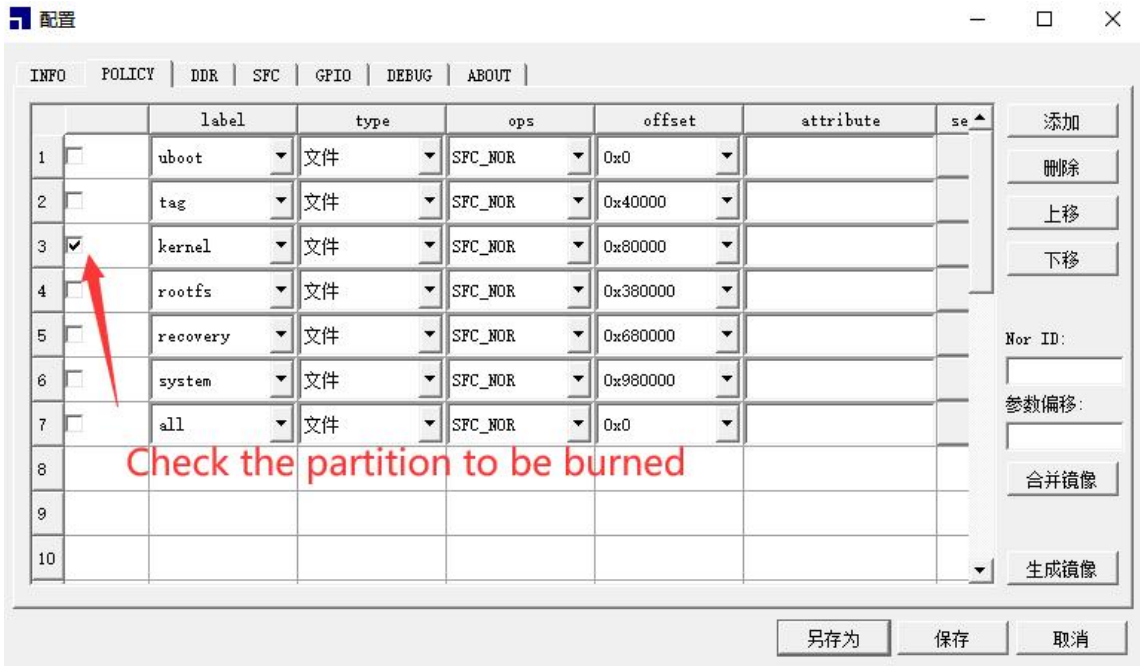
下面介绍如何烧录各个分区的内容，例如下图所示，这里要注意的是 offset 的设置，要根据分区情况使用，不能偏移重叠

然后找到分区信息，点击“添加”添加各个分区的内容，特别要注意 offset 和 size 的设置，如果重叠会显示红色。



然后找到基本信息一栏，去掉“全部擦除”。因为烧录某个分区，但是把其他分区擦除是不行的。

下面介绍单独烧录 kernel 分区为例，找到“POLICY” 勾选 kernel，例如下图。



点击开始，按 bootsel 按键，上电。烧录完成如下图所示。



## 4 串口烧录步骤

BOOT\_SEL1 配置为 1, BOOT\_SEL0 配置为 0 进入 UART 启动。

PRJ007\_zn\_msc0 配置可以支持启动后烧录 SD/EMMC 等 MSC 接口设备。

PRJ007\_zn\_sfcnor 配置可以支持启动后烧录 flash 等 SFC 接口设备。

### 1) Xshell

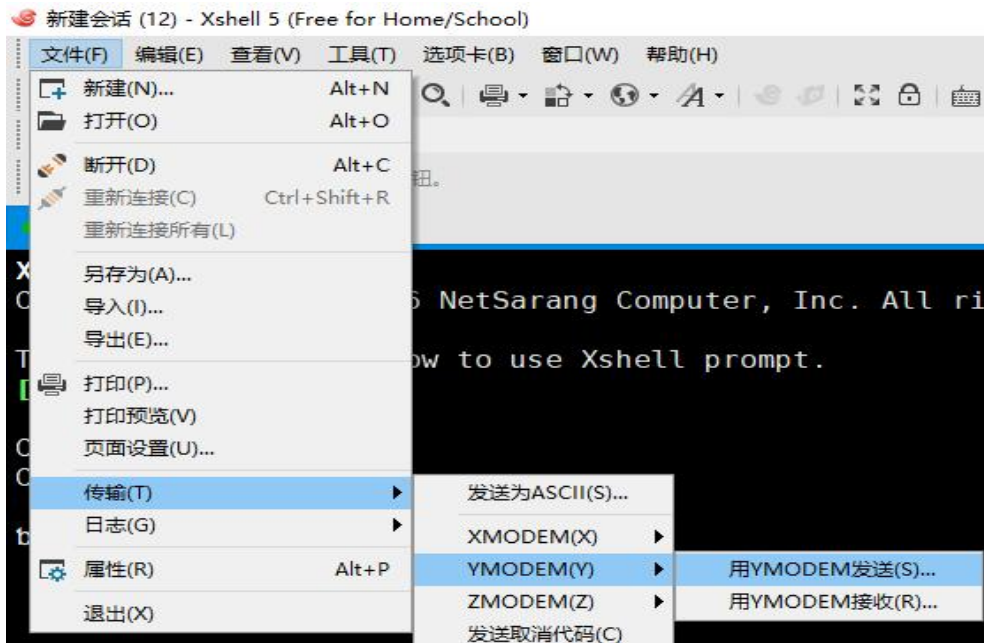
首先进入串口:右键---->传输--->选择 YMODEM-->用 YMODEM 发送--->选择传输的 uboot, BOOT\_SEL1 配置为 1, BOOT\_SEL0 配置为 0 进入 UART 启动, 传输结束, 进入 uboot 命令行: 先擦除 ddr: `mw.b 0x80600000 0xff 0x1000000`

然后输入: `loady 0x80600000`, 然后右键---->传输--->选择 YMODEM-->用 YMODEM 发送--->选择传输文件, 这里的 YMODEM 也可以换成 ZMODEM, ZMODEM 的速度比 YMODEM 更快; 最后将固件写入 flash:

```
sf probe;sf erase 0x0 0x1000000;sf write 0x80600000 0x0 0x1000000
```

```
Reserving 431k for U-Boot at: 83f94000
Reserving 32800k for malloc() at: 81f8c000
Reserving 32 Bytes for Board Info at: 81f8bfe0
Reserving 124 Bytes for Global Data at: 81f8bf64
Reserving 128k for boot params() at: 81f6bf64
Stack Pointer at: 81f6bf48
Now running in RAM - U-Boot at: 83f94000
MMC: MSC: 0
Data transfer: serial
Net: Jz4775-9161
Hit any key to stop autoboot: 0
isvp_t41#
isvp_t41#
isvp_t41#
isvp_t41# Loady 0x80600000
## Ready for binary (ymodem) download to 0x80600000 at 115200 bps...
C
```

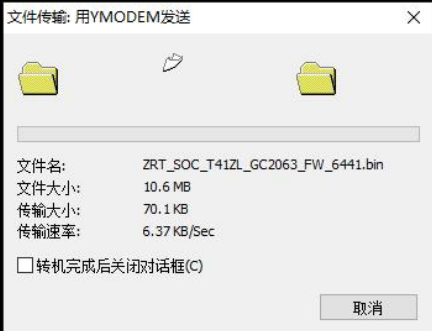
图 10 串口烧录



```

DDR clk rate 650000000
board_init_r
image entry point: 0x80100000
Board: ISVP (Ingenic XBurst T41 SoC)
DRAM: ddr size is 0x4000000
64 MiB
Top of RAM usable for U-Boot at: 84000000
Reserving 431k for U-Boot at: 83f94000
Reserving 32800k for malloc() at: 81f8c000
Reserving 32 Bytes for Board Info at: 81f8bfe0
Reserving 124 Bytes for Global Data at: 81f8bf64
Reserving 128k for boot params() at: 81f6bf64
Stack Pointer at: 81f6bf48
Now running in RAM - U-Boot at: 83f94000
MMC: MSC: 0
Data transfer: serial
Net: Jz4775-9161
Hit any key to stop autoboot: 0
isvp_t41#
isvp_t41#
isvp_t41#
isvp_t41# loady 0x80600000
## Ready for binary (ymodem) download to 0x80600000 at 115200 bps...
CCCCCCCC[

```



```

isvp_t41# loady 0x80600000
## Ready for binary (ymodem) download to 0x80600000 at 115200 bps...
CCCxyzModem - CRC mode, 87361(SH)/0(STX)/0(CAN) packets, 4 retries

## Total Size      = 0x00aaa000 = 11182080 Bytes
isvp_t41# sf0 probe;sf0 erase 0x0 0x1000000;sf0 write 0x80600000 0x0 0x1000000
isvp_t41#
isvp_t41#
isvp_t41# sf probe;sf erase 0x0 0x1000000;sf write 0x80600000 0x0 0x1000000
SF: Detected GD25Q128C, manufacturer id: c8
--->probe spend 4 ms
SF: 16777216 bytes @ 0x0 Erased: OK
--->erase spend 10853 ms
SF: 16777216 bytes @ 0x0 Written: OK
--->write spend 17979 ms
isvp_t41#
isvp_t41# reset


```

## 2) minicom 操作:

首先设置 ymodem,如下图, 进入串口, 快捷键: **Ctrl+a z**, 选择发送文件, 选择 ymodem, 选择要发送的 uboot (注意: uboot 中不要有“printf”), 此时置 boot\_sel1 为 1, boot\_sel0 为 0, 完成后会进入 uboot 命令行, 输入: **loady 0x80600000**, 快捷键: **Ctrl+a z**, 选择发送文件完成后输入: **sf probe;sf erase 0x0 0x1000000;sf write 0x80600000 0x0 0x1000000**



```
sdrain init start
DDR clk rate 650000000
board_init_r          +-[Upload]--+
image entry point: 0x80100000 | zmodem |
Board: ISVP (Ingenic XBurst T | ymodem |
DRAM: ddr size is 0x4000000 | xmodem |
64 MiB | kermit |
Top of RAM usable for U-Boot | ascii |
Reserving 431k for U-Boot at: +-----+
Reserving 32800k for malloc() at: 81f8c000
Reserving 32 Bytes for Board Info at: 81f8bfe0
Reserving 124 Bytes for Global Data at: 81f8bf64
Reserving 128k for boot params() at: 81f6bf64
Stack Pointer at: 81f6bf48
Now running in RAM - U-Boot at: 83f94000
MMC:  MSC: 0
Data transfer:  serial
Net:  Jz4775-9161
Hit any key to stop autoboot:  0
isvp_t41# loady 0x80600000
## Ready for binary (ymodem) download to 0x80600000 at 115200 bps...
C78(SOH)/0(STX)/0(CAN) packets, 15 retries
## Total Size      = 0x00be3000 = 12464128 Bytes
```



```
[Bootrom]total_length: 14970
[Bootrom]store_addr: 0x80004a80
[Bootrom]start_retries: 4
[Bootrom]-----[ymodem upload - Press CTRL-C to quit]-----+
-----|Sending: u-boot-with-spl.bin
Board: IS|Ymodem sectors/kbytes sent: 186/23k
DRAM: 12|
Top of RA|
Reserving|
Reserving|
Reserving|
Reserving+-----+
Reserving 128k for boot params() at: 85f6ff64
Stack Pointer at: 85f6ff48
```

```
U-Boot SPL 2013.07-00004-g9aa625e26 (Sep 07 2022 - 14:25:35)
Timer init

apll_freq+-----[ymodem upload - Press CTRL-C to quit]-----+
mpll_freq|Sending: ZRT_SOC_T41ZL_GC2063_FW.bin
vppll_freq|Ymodem sectors/kbytes sent: 9316/1164k
|
|ccclk = 12|
|l2clk = 6|
|h0clk = 2|
|h2clk = 2|
|pclk = 13+-----+
cpu clk source: apll
ddr clk source: mpll
sdram init start
DDR clk rate 650000000
board_init_r
image entry point: 0x80100000
Board: ISVP (Ingenic XBurst T41 SoC)
DRAM: ddr size is 0x4000000
64 MiB
Top of RAM usable for U-Boot at: 84000000
Reserving 431k for U-Boot at: 83f94000
Reserving 32800k for malloc() at: 81f8c000
Reserving 32 Bytes for Board Info at: 81f8bfe0
Reserving 124 Bytes for Global Data at: 81f8bf64
Reserving 128k for boot params() at: 81f6bf64
Stack Pointer at: 81f6bf48
Now running in RAM - U-Boot at: 83f94000
MMC: MSC: 0
Data transfer: serial
Net: Jz4775-9161
Hit any key to stop autoboot: 0
isvp_t41# loady 0x80600000
## Ready for binary (ymodem) download to 0x80600000 at 115200 bps...
C
```

```
Hit any key to stop autoboot: 0
isvp_t41# loady 0x80600000
## Ready for binary (ymodem) download to 0x80600000 at 115200 bps...
C78(SOH)/0(STX)/0(CAN) packets, 15 retries
## Total Size = 0x00be3000 = 12464128 Bytes
isvp_t41#
isvp_t41# sf probe;sf erase 0x0 0x1000000;sf write 0x80600000 0x0 0x1000000
SF: Detected GD25Q128C, manufacturer id: c8
--->probe spend 5 ms
SF: 16777216 bytes @ 0x0 Erased: OK
--->erase spend 15492 ms
SF: 16777216 bytes @ 0x0 Written: OK
--->write spend 19226 ms
isvp_t41#
isvp_t41# reset
```

