

# Rockchip RKDevInfoWriteTool使用指南

---

文件标识: RK-YH-YF-275

发布版本: V1.0.6

日期: 2020-08-21

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 © 2020 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)

---

## 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
V1.0.4	lsh	2020-06-08	
V1.0.5	lsh	2020-08-21	
V1.0.6	lsh	2021-04-23	

## 目录

### Rockchip RKDevInfoWriteTool使用指南

- 一、简介
- 二、写号配置
  - 2.1 主界面
  - 2.2 写号配置
    - 通用设置
    - 写入项配置
    - 手动模式
    - 自增模式
    - 文件模式
  - 2.3 注意事项
    - 2.3.1 SN配置
    - 2.3.2 MAC配置
- 三、插件
- 四、二次开发需求
- 五、问题解析
  - 1 提示 获取XXX失败
  - 2 提示写入失败
  - 3 MAC写入后不能生效问题

## 一、简介

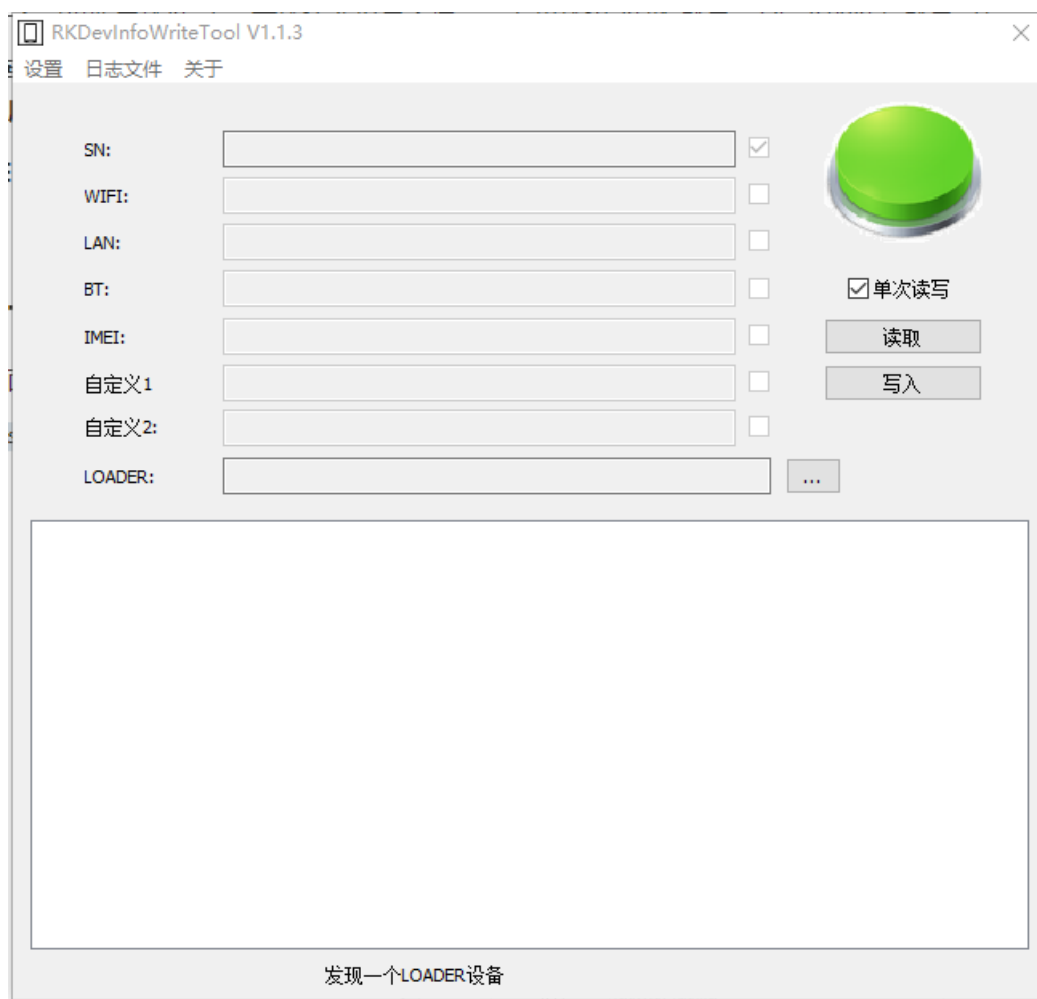
RKDevInfoWriteTool用于向VendorStorage（勾选兼容模式的话，写在IDB分区）分区写入用户定义数据，如机器的SN、Wi-Fi、IMEI、用户自定义等数据，这些数据在设备恢复出厂设置后，不会丢失。

### 两种设备模式：

- 1、MASKROM模式：短接FLASH CLK引脚，这个时候，SOC无法从可用的FLASH加载固件，会切换到USB下载状态（MASKROM模式）。这个时候，是需要用户选择正确的MiniLoaderAll.bin代码，烧写进去，MiniLoaderAll.bin初始化USB之后，才能写号。
- 2、LOADER模式：按住“音量+”开机或者通过adb reboot loader命令，uboot在检测相应条件之后，切换到Loader模式。在该模式下面写号，设备必须是已经有烧写过固件的（至少MiniLoaderAll.bin、parameter、uboot.img）。

## 二、写号配置

## 2.1 主界面



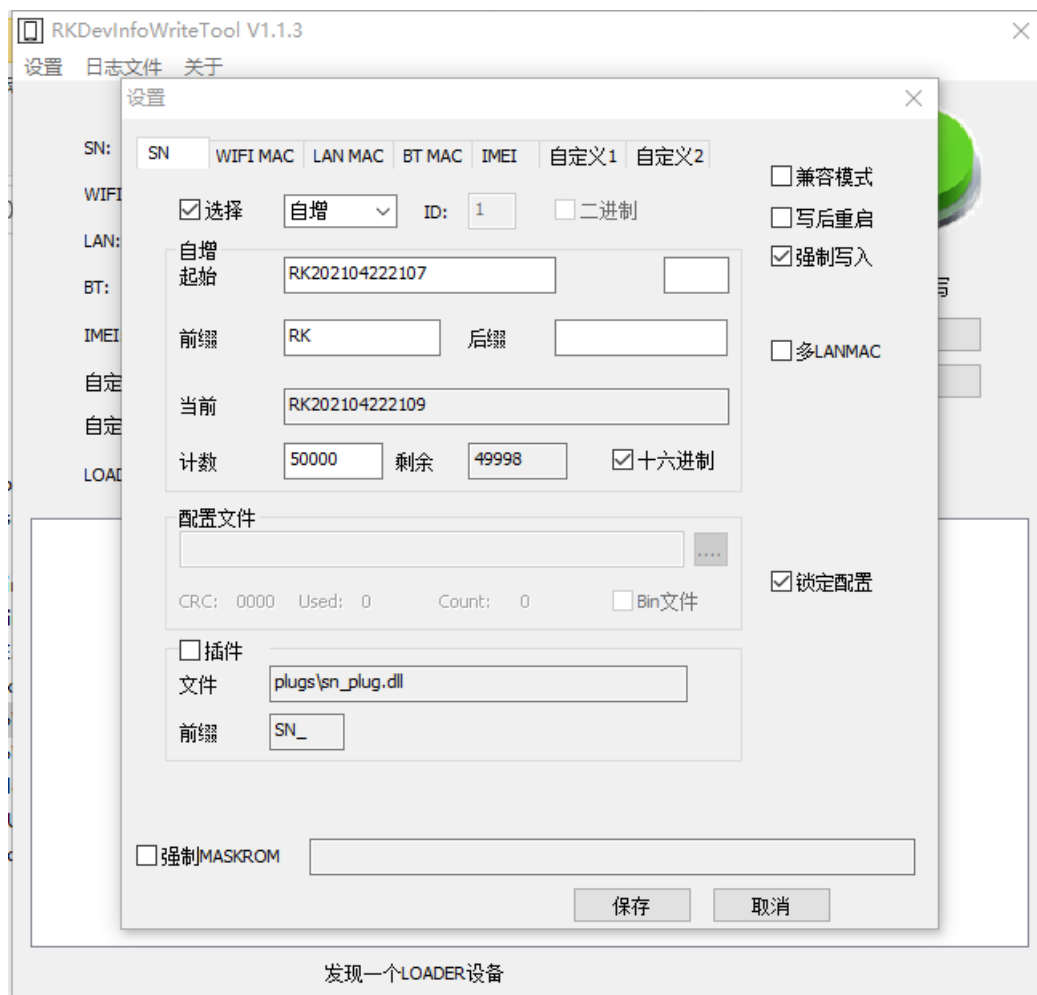
**1 单次读写：**勾选“单次读写”，每次接入设备之后，需要再次按“读取”或者“写入”才能读号或者写号；

**2 连续读写：**不勾选“单次读写”，接入新设备自动触发读写操作。写号时候，在手动输入模式下面，全部项都输完之后，敲回车键，也会触发写入操作，**这个操作，方便用户使用扫描枪写号**，注意要设置扫描枪末尾自动回车功能；

**3 LOADER：**选择MiniLoader文件，设备在“MASKROM模式”的话，必须选择正确的MiniLoader文件。

## 2.2 写号配置

单击主界面菜单栏“设置”进入写号工具配置界面：



## 通用设置

- 1、兼容模式：选择“兼容模式”下面写号，在这种模式下面，可以选择“强制MASKROM”强制机器到maskrom下面写号，或者不选择“强制MASKROM”，那么机器需要先切换到loader模式写号（适用于在空片情况下使用）；
- 2、写后重启：写完后重启机器；
- 3、强制写入：不勾选的话，写入前会检查有没有写过号，如果写过了，会跳过；如果勾选，不管设备有没有写过号，直接写入；

下面两项仅仅针对自定义可配置：

## 写入项配置

- 1、ID：SN、Wi-Fi、LAN MAC、BT MAC等数据项存储VendorStorage分区，每个数据项通过一个唯一的ID 进行检索。只有自定义项支持自定义ID，且不可与前面定义的重复；
- 2、二进制：选定输入的项是否是二进制格式。如“11223344”，如果是二进制格式，则实际写入到设备的是“0x11 0x22 0x33 0x44”四个字节；没有勾选二进制格式的话，则按文本方式写入，实际写入的是“11223344”字符串；

对于写入项的获取，可以在设置里面，通过下拉框选择三种模式：“手动”、“自增”、“文件”

## 手动模式

直接在主界面从键盘或者扫描枪输入要写入的数据项

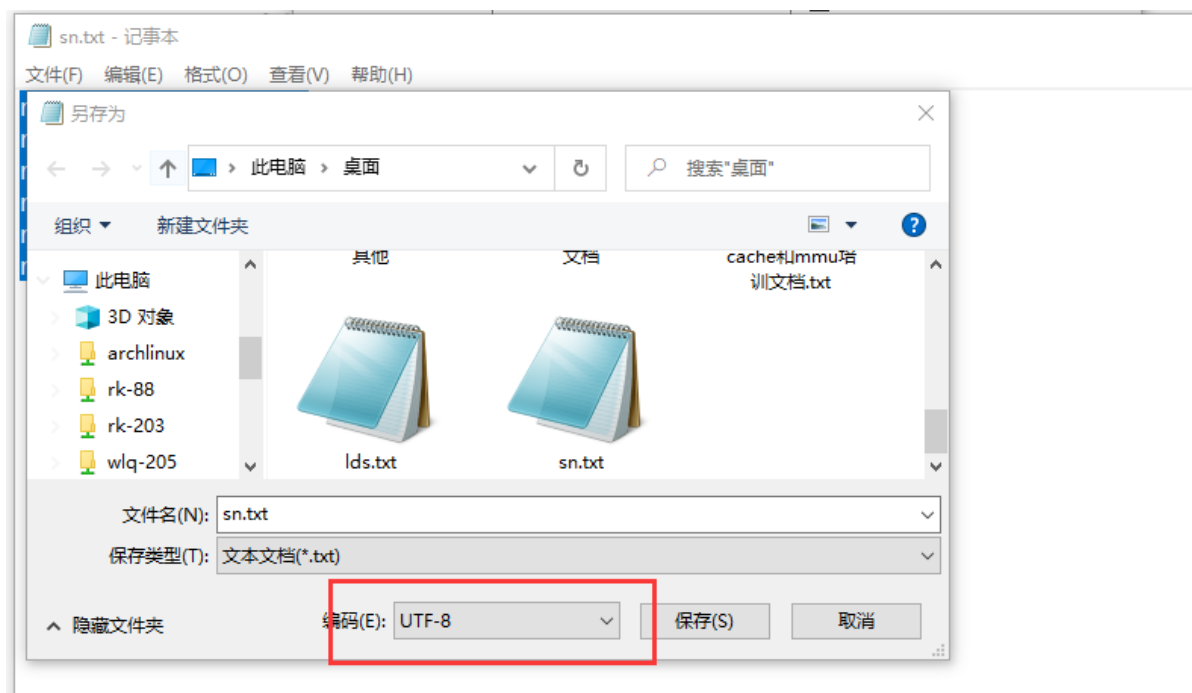
## 自增模式

在这个模式下，用户可以设置一个包含数字的字符串作为起始字符串，如“rockchip201810150029sn”，每次成功写完一个号之后，软件都会自动给数字部分自增1 (+1)。默认是按十进制自增的，如果勾选“十六进制”，那么是按十六进制自增。“前缀”，“后缀”是起始字符串的前面部分，和后面部分，用户可以通过设置“前缀”，“后缀”，让这部分内容不会自增。

## 文件模式

在这个模式下，写入项可以通过一个文本获取。

**文本格式：**如果没有勾选“bin文件”选项的话，那么文件内容必须是文本文件，软件每次从文件里面获取一行，然后写到设备。注意文本必须以UTF-8编码，比如在Windows下面的记事本里面，必须按下图选择格式保存：



SN号文件文本实例：

```
RK202104222107
RK202104222108
RK202104222109
RK202104222110
RK202104222111
RK202104222112
```

MAC号文件文本实例：

```
0a0b11223344
0a0b11223345
0a0b11223346
0a0b11223347
0a0b11223348
0a0b11223349
0a0b11223350
```

**BIN文件：**勾选“Bin文件”选项的话，那么软件每次写入会把整个文件内容写到设备里面。

## 2.3 注意事项

### 2.3.1 SN配置

SN默认都是要求符合GMS的规范，**要求长度不大于14且字母开头的数字字母组合**，如果没有按要求写入，重启设备之后，看到的SN不是自己写入的SN，而是系统随机生成的。对于这个问题的解决办法

1. 重新写入符合GMS的规范的SN
2. 对于不需要使用GMS服务的场景，可以从SDK里面去掉SN校验的功能：

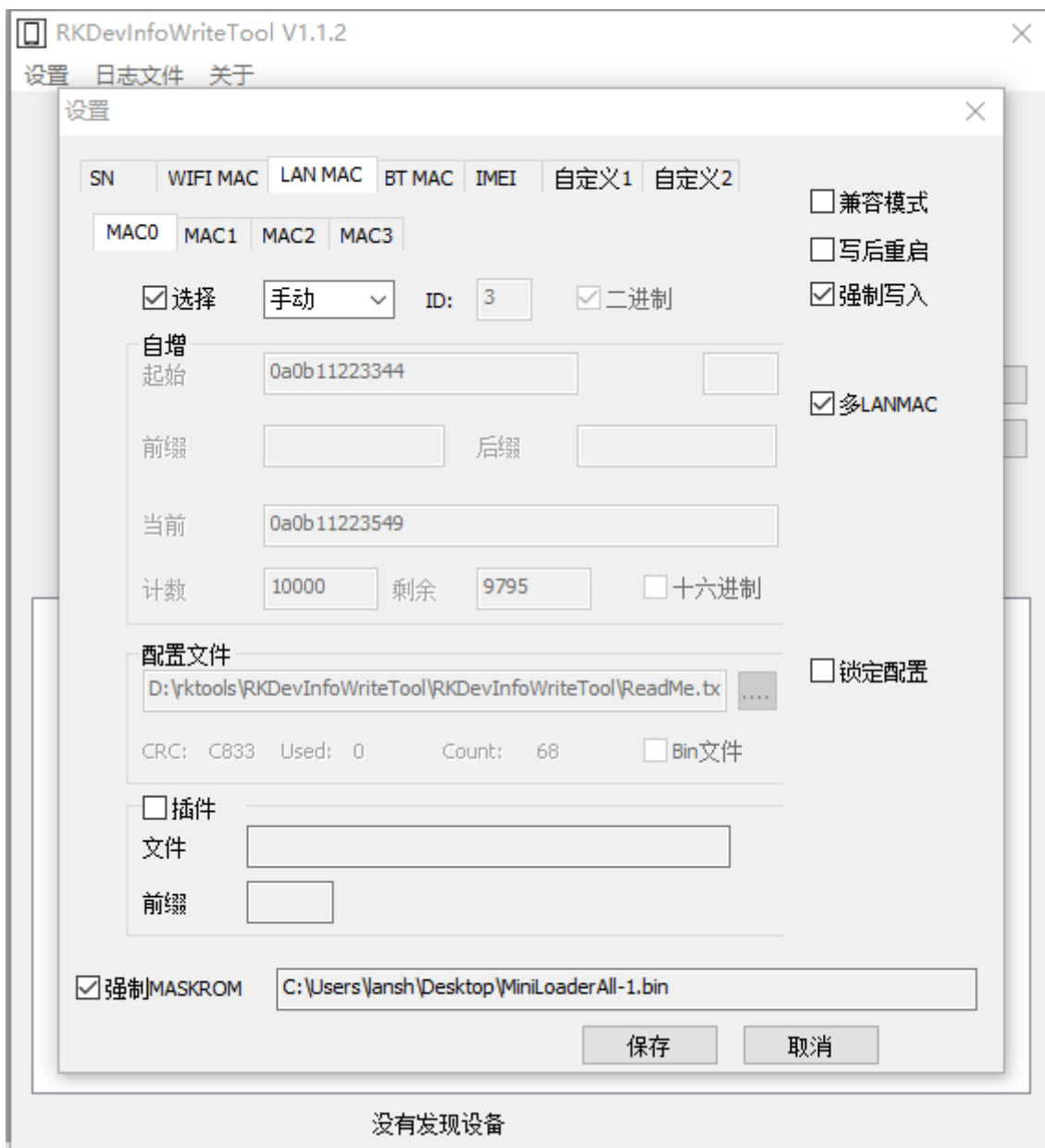
```
lsh@rk-intel-1:~/rk-sdk/android10/hardware/rockchip/drmsservice$ git diff
diff --git a/Android.mk b/Android.mk
index 2606cae..6f18054 100644
--- a/Android.mk
+++ b/Android.mk
@@ -22,7 +22,7 @@ LOCAL_CPPFLAGS := \

# API 28 -> Android 9.0
ifeq (1,$(strip $(shell expr $(PLATFORM_SDK_VERSION) \>= 29)))
-LOCAL_CFLAGS += -DENABLE_CMDLINE_VERIFY
+#LOCAL_CFLAGS += -DENABLE_CMDLINE_VERIFY
endif

include $(BUILD_EXECUTABLE)
```

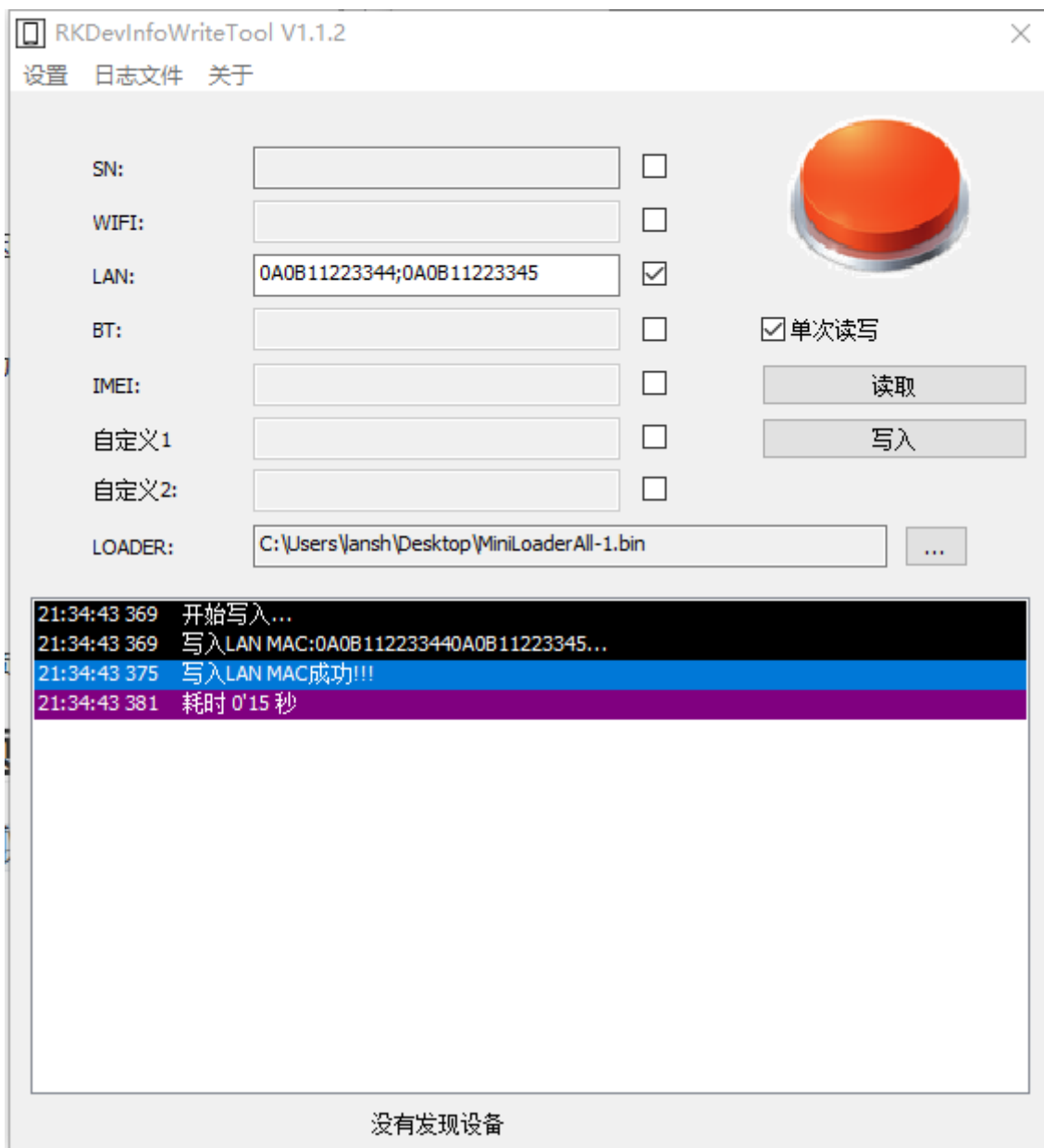
### 2.3.2 MAC配置

对于RK356X有双MAC的需求，而且后续可能考虑更多的MAC地址，因此工具添加了一个 **多LAN MAC** 的配置。选择 **多LAN MAC** 之后，对应界面如下：



在工具里面，因为界面的局限，对于每个LANMAC的模式不能独立配置，必须同为手动、自增、文件模式。因为同一个ID里面，有多个号要写入，所以配置方式不太一样：

1. 对于配置为手动写入，在输入框每个号须以;号分割：

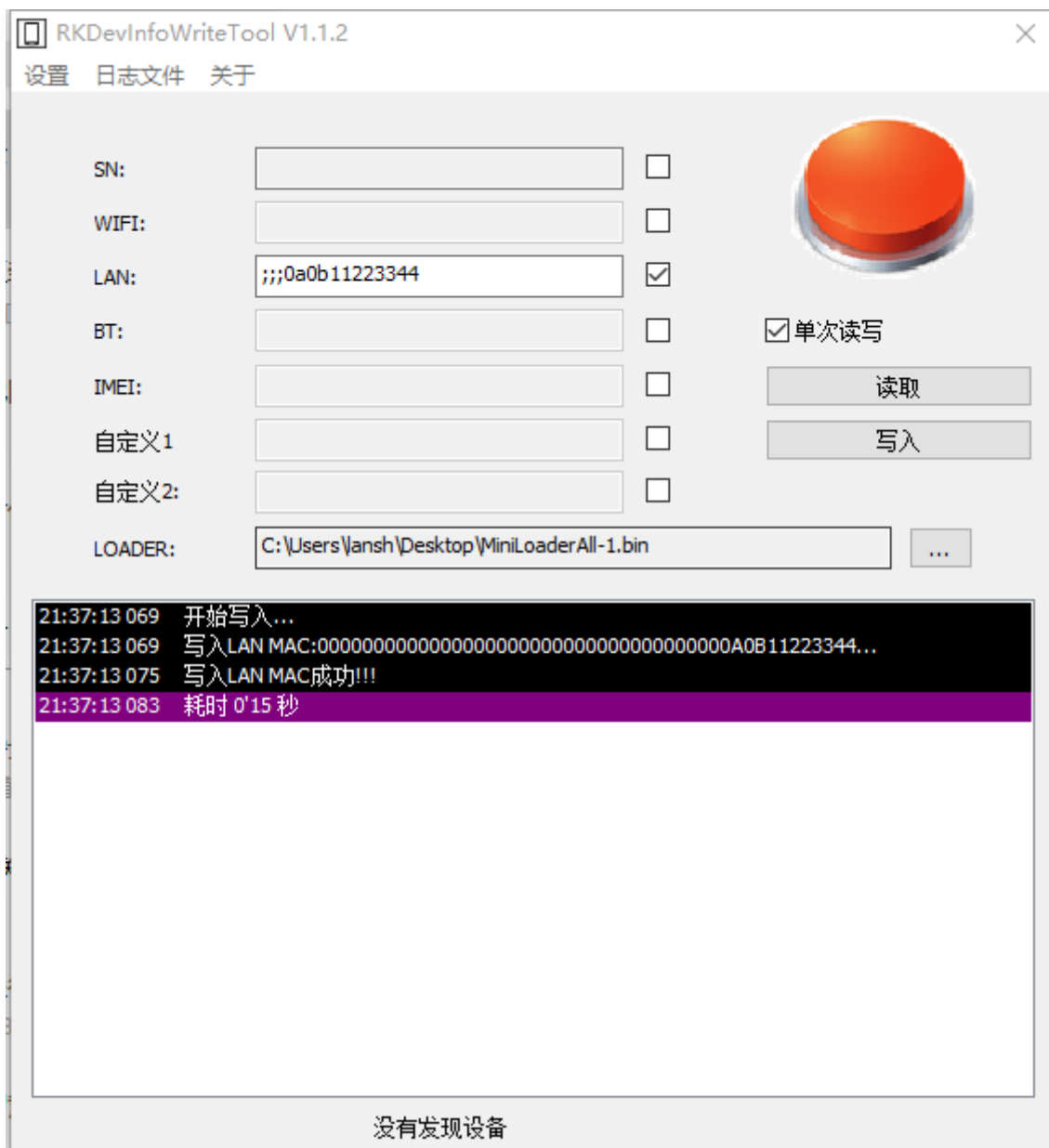


2. 对于自增模式，必须每个写入MAC都要配置好自增数据
3. 对于文件模式，必须每个写入MAC都要配置好文件数据

**注意:** 全部MAC都是保存在ID为3的存储项里面的，每个MAC是按6字节存储的。如：MAC0存储在第一个6字节，MAC1存储在第二个6字节....。驱动里面，读取MAC的时候，也是按6字节为单位获取的。顺序通过aliases——对应：ethernet0 对应MAC0， ethernet1 对应MAC1

```
/* dts */
/ {
    compatible = "rockchip,rk3568";
    ....
    aliases {
        csi2dphy0 = &csi2_dphy0;
        csi2dphy1 = &csi2_dphy1;
        csi2dphy2 = &csi2_dphy2;
        dsi0 = &dsi0;
        dsi1 = &dsi1;
        ethernet0 = &gmac0;
        ethernet1 = &gmac1;
    }
}
```





### 三、插件

该项用于写入项的一些预检查，如果带GMS包固件，对SN有14位长度限制，必须以字母开头等。可以通过勾选插件来确保。修改该项需要重启软件。

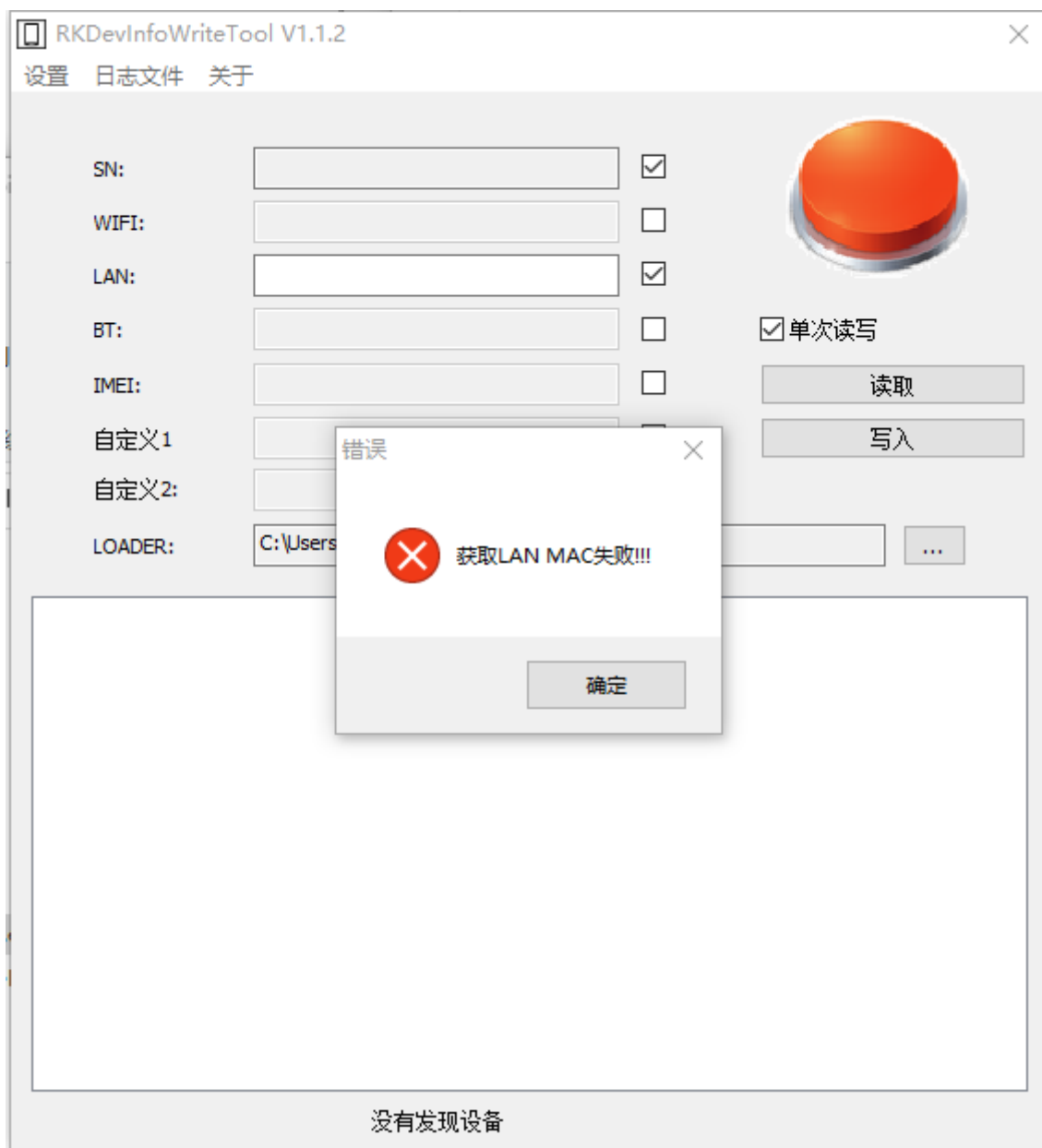
### 四、二次开发需求

如果以上配置不能满足需求，可以申请源码进行二次开发。申请方式，写邮件给业务，抄送[ish@rock-chips.com](mailto:ish@rock-chips.com)。

RK工具开发环境，目前统一为：VS2008 + VS2008SP1

### 五、问题解析

#### 1 提示 获取XXX失败

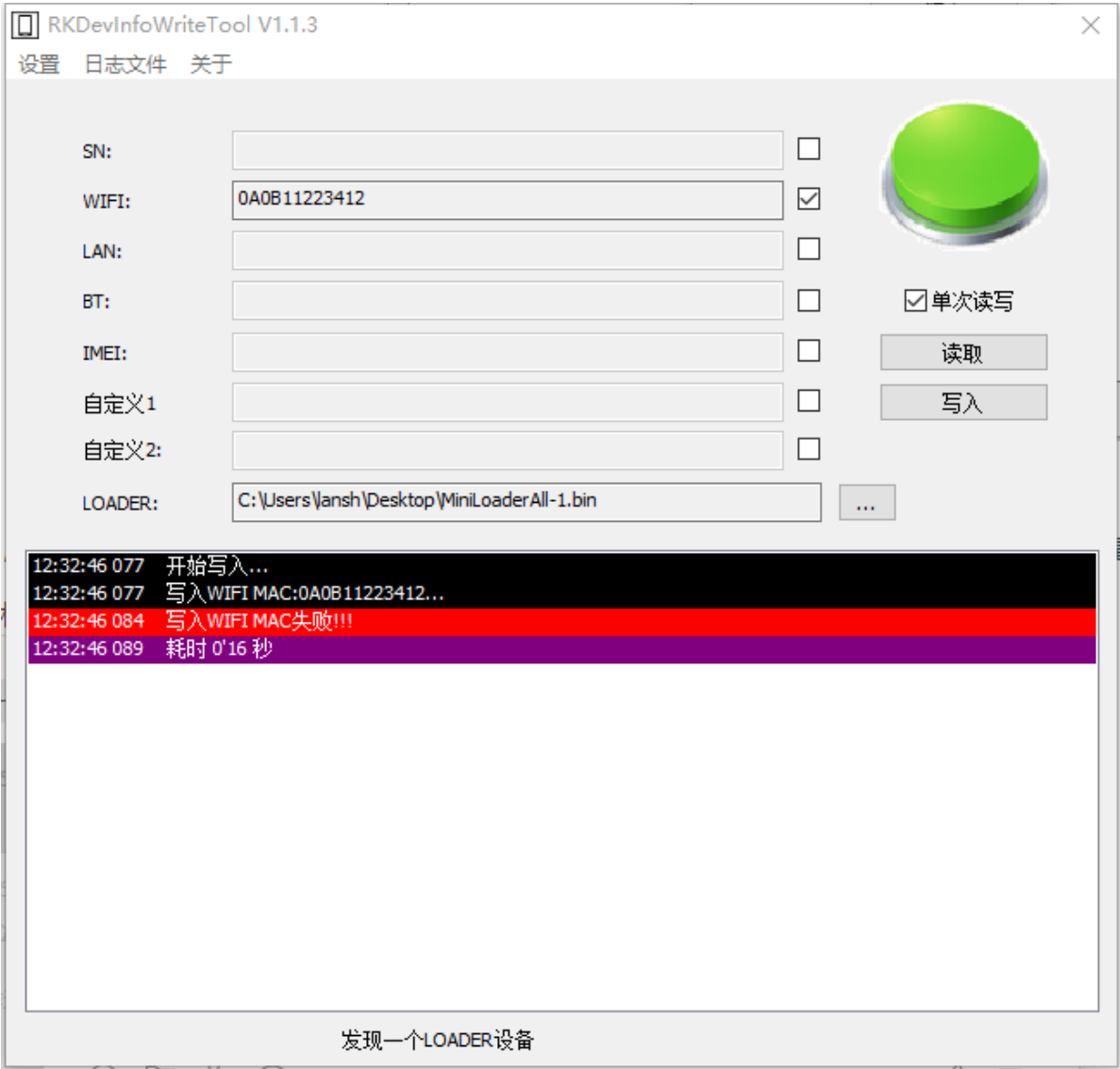


这个问题，是根据用户配置，获取写入数据时候获取数据失败，不是在写号的阶段失败的。检查下配置就好了。如

- 手动 模式，检查下是不是输入框没有输入

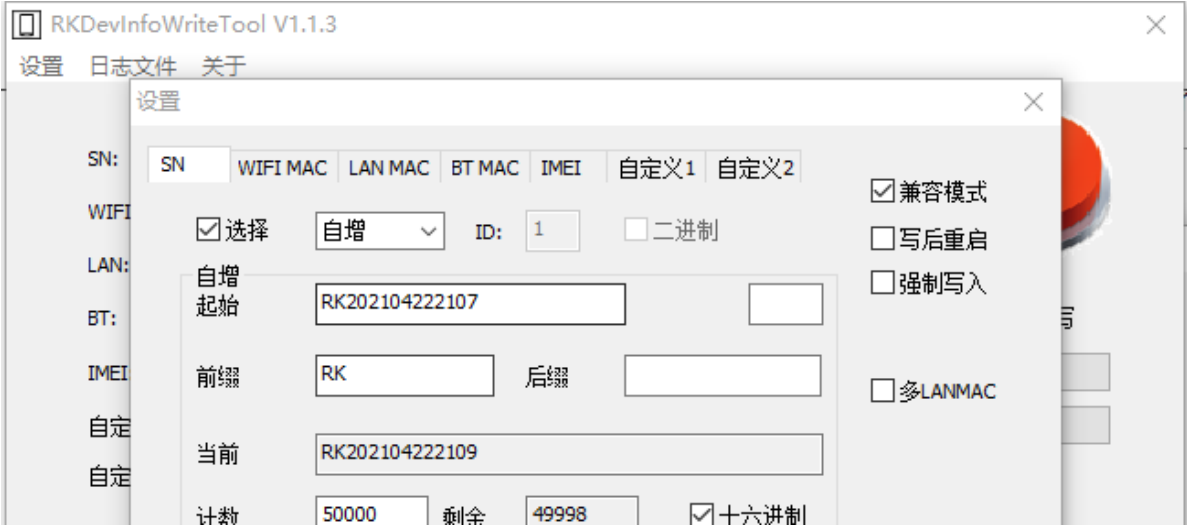
- 自增 模式，检查是不是配号用光了，或者起始字段没写
- 文件 模式，检查文件是不是被删了，或者是不是配号用光了

## 2 提示写入失败



因为UBOOT版本升级的改动，写号的命令有做过升级，一般默认工具会自动选择对应命令，如果写号一直提示失败的问题，有可能是选择失败，这个时候可以通过一下方法尝试：

1. 确认下是不是勾选了 兼容模式，如果有勾选，去掉试试。这个模式在Android7.1之后，基本不用了。



2. 修改配置文件，强制使用某个命令。配置文件是安装目录下面config.ini，在修改前，记得要先关闭软件：

```
[System]
ForceApi=0
ForceApiType=0
```

如上，默认 ForceApi=0，改为1， ForceApiType 的值可以是 0 - 2

ForceApiType 跟uboot对应一般如下：

- 0- uboot分支为next-dev
- 1 - uboot分支为rkdevelop
- 2 - 适用于Android5.1之前版本，写在IDB中

### 3 MAC写入后不能生效问题

工具往vendorstorage写入有效的MAC地址后，会由UBOOT读取，然后通过dtb传递给内核。如果写完设备的MAC地址显示不对，可以先在[rockchip\\_set\\_ethaddr](#)函数里面，把MAC地址打印出来，先做第一步判断，对应uboot代码

```
/* ./arch/arm/mach-rockchip/board.c */
static int rockchip_set_ethaddr(void)
/* ./common/fdt_support.c */
void fdt_fixup_ethernet(void *fdt)
```