

# R16

## 修改系统配置使用书



# 目 录

<b>修改系统配置使用书.....</b>	<b>1</b>
<b>1. 概述.....</b>	<b>1</b>
1.1. 编写目的.....	1
1.2. 适用范围.....	1
1.3. 相关人员.....	1
1.4. 实现的功能.....	1
<b>2. 修改系统配置.....</b>	<b>2</b>
2.1. 修改配置的命令介绍.....	2
2.2. 用户按键说明.....	2
2.3. 修改命令示例.....	2
<b>3. 保存系统配置.....</b>	<b>4</b>
<b>4. 目前修改 SYS_CONFIG 功能不支持的修改.....</b>	<b>5</b>
<b>5. DEBUG 模式.....</b>	<b>6</b>
5.1. Debug_mode 介绍.....	6
5.2. 操作步骤说明.....	6
5.3. 在 android 阶段往 misc 分区写入 debug_mode.....	6

## 1. 概述

### 1.1. 编写目的

目前在 uboot 中没有提供修改 sys\_config.fex 文件的接口，在平时的开发过程中，有时候只需要修改某一个配置，但是往往是需要在自己电脑修改完之后重新打包下载，浪费了很多时间，现在在 uboot 中提供了两个命令可供用户调用的接口，用户只需输入相应的命令，即可对 sys\_config.fex 中的信息进行修改并且重新下载到 flash 中。

### 1.2. 适用范围

不区分平台，可以在公司芯片的任一平台使用；

### 1.3. 相关人员

对于平常的开发工作中，需要经常修改系统配置的开发人员，测试人员。

### 1.4. 实现的功能

- 通过在 uboot shell 输入命令：setcfg mainkey 进行相关主键次键的设置，savecfg 将会对之前修改的内容进行保存。
- 在 uboot 阶段进入 debug 模式，即通过在 uboot 特定的操作，让其打印切换到 card0，以及将系统的打印等级 loglevel = 8 。

## 2. 修改系统配置

### 2.1. 修改配置的命令介绍

目前快速修改 sys\_config.fex 的命令有：

**setcfg mainkey subkey or setcfg mainkey**

将修改完之后的 sys\_config.fex 进行保存的命令有：

**savecfg**

注：setcfg 只能在当前系统有效，如果没有执行 savecfg，下次启动系统，还是按照原先的系统配置运行。savecfg 这条命令，只有在有修改系统配置的前提下，执行才有效，没有修改的前提，是空操作。

### 2.2. 用户按键说明

Ctrl+c：假设在修改的中途按下，直接退出 sys\_config 修改命令，表示当前修改的主键不会保存进内存，但是之前操作的修改主键就已经被保存进当前环境。

Enter：当没有用户键入值时，默认使用当前值，即跳过不修改；

另外也会对用户键入的值进行合法性检测。当需要输入字母时，除了字母，其他一概不接受修改，并且会提醒用户使用合法性键值。

当输入阿拉伯数字的时候，支持十六进制跟十进制输入，十六进制输入的时候有特定格式：0xXXXX，十进制就没有这个要求。

### 2.3. 修改命令示例

修改系统配置信息，支持修改一下内容：

#### 1: gpio 配置；

Sys\_config.fex 中 gpio 口的表现形式示例：

**twi\_sda = port:PH15<2><default><default><default>**

示例：

```
sunxi#setcfg twi_para twi_sda
--twi_para--                //主键名
twi_sda :                    //次键名
=====
PORT : H?a                  //表示使用 PH gpio，用户修改为 a 端口，字母输入大小写都支持
PORT_NUM :15?11             //端口号由 15 改为 11
MUL_SEL :2?2                //选择功能端口
PULL :-1?1                  //default 默认为-1，需要修改的时候可以键入
DRV_LEVEL :-1?1             //驱动能力
DATA :-1?1                  //输出数据端口
=====
```

#### 2: 字符串配置；

Sys\_config.fex 中字符串的配置示例：

**pa\_bias = "axp809:dc1:3000"**

示例：

```
sunxi#setcfg gpio_bias pa_bias //输入命令
--gpio_bias-- //主键名
pa_bias : axp809:dc1:3000 //次键名以及次键名的内容
?axp806:dc2:3000 //用户输入的字符串
```

注：用户输入的字符串有以下约定，

1：用户输出字符串的长度不可大于原来字符串的长度 $((length > 2) \ll 2 + 1) * 4$ （四个字节对齐）  
当超过原来字符串长度的时候，将不会修改原来字符串的内容

### 3：数字配置；

Sys\_config.fex 中数值的配置示例：

logical\_start = 40960

示例：

```
sunxi#setcfg card_boot logical_start //输入命令
--card_boot-- //主键名
logical_start : 40960?40960 //次键名，当前值以及用户输入要修改的值
```

### 3. 保存系统配置

在串口输入端数据 **savecfg** 之后，假设用户之前有对配置进行修改，那么就会将配置信息重新下载进 flash 中。

```
sunxi#savecfg
uboot size = 0xb4000
storage type = 0
uboot:physical version: 0 0 20140325 1840
NAND_PIORequest, nand_index: 0x0
Reg 0x06000848: 0x22222222
Reg 0x0600084c: 0x22222222
Reg 0x06000850: 0x7222
Reg 0x06000864: 0x5140
Reg 0x06000868: 0x14
Reg 0x06000b08: 0xa
nand : get id_number_ctl fail, 1
uboot:nand info: eb94dead fffff74 318c 30708 4
burn uboot normal mode!
pages_per_block: 0x100
pages_per_copy: 0x2d
uboot 2
fill uboot block with dummy data
uboot 3
fill uboot block with dummy data
uboot 4
fill uboot block with dummy data
uboot 5
fill uboot block with dummy data
uboot 6
fill uboot block with dummy data
verify uboot blk 2
verify uboot blk 3
verify uboot blk 4
verify uboot blk 5
verify uboot blk 6
uboot nand_request_dma: current platform do not support general dma!
```

#### 4. 目前修改 **sys\_config** 功能不支持的修改

- 1: 不支持增加或者减少 `sys_config.fex` 中的主键个数;
- 2: 假设次键的属性为字符串, 当修改的字符串长度大于原来字符串长度时, 不支持修改;
- 3: 不支持修改次键的属性, 坚持原来的属性。
- 4: 不支持在 `boot0` 已经读取/设置的参数, 如 `darm` 参数的配置。



## 5. debug 模式

### 5.1. Debug\_mode 介绍

我们给客户的固件，一般都是统一从串口打印的，在这种情况下，当机子出现问题时，需要给小机引出串口线，过程较繁琐，浪费时间。现在提供一种人为操作的方法，可以实现 uboot 阶段将串口打印的切换到 card0 打印上。

在这个模式下，会有几个改变：

- 会将系统的打印等级改为 8，
- jtag 打开，便于开发人员调试。

上述改变只在当前开机系统下有效。

上述方法是在小机无法进入系统，可以通过在 uboot 阶段将打印切到 card0 打印。对其他系统应用无任何影响。

### 5.2. 操作步骤说明

修改 sys\_config.fex 文件：

```
[force_uart_para]
force_uart_port    = 0           //卡 0 对应的 uart 口
force_uart_tx      = port:PF02<3><1><default><default>
force_uart_rx      = port:PF04<3><1><default><default> //将卡 0 配置成打印模式的配置
注：不同平台上卡 0 打印的配置也不一样，需要使用前确认下是否跟对应平台的卡 0 配置相同，详细信息请查看相应平台的 spec
```

需要的工具：

能工作的电脑主机，usb 线

操作步骤：

- 在小机关机的情况下，按住非 power 按键的同时，按住 power 按键让小机开机，同时非 poweron 按键不松开；
- 将小机通过 usb 线连接，插拔电脑主机次数超过 3 次，如果在连接卡 0 的串口屏幕上能够看到打印信息，就说明成功切换到该模式下了；

注：如果想进入 debug\_mode，开机之后就不要按住 power 按键；

### 5.3. 在 android 阶段往 misc 分区写入 debug\_mode

另外，提供另外一种方法进入 debug 模式，前提是要能进入安卓系统。

操作步骤：

- 在小机进入安卓，在 pc 运行 command 命令，输入命令 “adb reboot debug\_mode”，重启之后就能够从卡 0 看到打印信息。