



## RDA5981 Flash 分区手册及 OTA 接口

版本号	描述	时间
Version1	初始版本	2016.12.24
Version2	增加 bootloader，增加 1/2/4 MB flash 支持	2017.04.10
Version2.1	用户数据区可以设置大小	2017.04.11
Version2.2	修改启动分区位置的书写错误，增加读取用户数据区长度接口	2017.04.20
Version2.3	开放 flash 读写接口	2017.06.29

更多信息请访问 RDA IOT 论坛:

<http://bbs.rdamicro.com>

# 1. Flash 分区

Flash 分为四部分：

a. 启动分区。

大小固定：4KB；位置固定：0x18000000-0x18000fff。该分区为 RDA 芯片内部 bootrom 程序专用区域，用户不能操作该区域。

b. 系统数据区。

大小固定：4KB；用户可配置该区域起始地址。该区域存放 MAC 地址、已保存路由器 SSID/Passwd 等信息，用户只能通过 RDA 接口进行操作。

```
/*
 * return 0:ok, else:error.
 */
int rda5981_flash_read_mac_addr(unsigned char *mac_addr);
int rda5981_flash_write_mac_addr(unsigned char *mac_addr);
```

c. 用户数据区。

用户可配置该区域起始地址和大小。该区域存放用户私有数据，可以通过 RDA 接口进行读写。

```
/*
 * read 3rd parter data length from flash
 * return user data length
 */
int rda5981_flash_read_3rdparter_data_length();

/*
 * read 3rd parter data from flash
 * @buf, buf to store user data
 * @buf_len, length of buf
 * return user data length
 */
int rda5981_flash_read_3rdparter_data(unsigned char *buf, unsigned int buf_len);

/*
 * write 3rd parter data from flash
 * @buf, data to write
 * @buf_len, length of buf. max:4k
 * return 0:ok, else:fail
 */
int rda5981_flash_write_3rdparter_data(const unsigned char *buf, unsigned int buf_len);
```

d. 代码区。

Flash 剩余空间均可作为代码区，用 RDA5981 下载工具进行烧写。

## 2. 修改分区位置

RDA 提供的库默认使用 1MB Flash，系统数据放在 0x180fb000-0x180fbfff，用户数据放在 0x180fc000-0x180fcfff。

用户也可以在 main() 函数初始位置设置 Flash 大小，以及系统数据区起始位置、用户数据区起始位置、用户数据区大小。**三个值都要保证 4K 对齐。**

```
/*
 * set flash size
 * @size, 1MB:0x100000, 2MB:0x200000, 4MB:0x400000. default size: 1MB
 * return 0:ok, else:fail
 */
int rda5981_set_flash_size(const unsigned int size);

/*
 * set userdata location on flash
 * @sys_data_addr, data to save system parameter, user can not operate this area directly.
 * size:4KB. default location:0x180fb000
 * @user_data_addr, data to save user data. user can save own data in this area by
@rda5981_flash_read_3rdpartter_data
 * and @rda5981_flash_write_3rdpartter_data
 * default location:0x180fc000
 * @user_data_len, user data length, default:4KB
 * return 0:ok, else:fail
 */
int rda5981_set_user_data_addr(const unsigned int sys_data_addr,
    const unsigned int user_data_addr, const unsigned int user_data_len);
```

## 3. 芯片启动流程

RDA5981 芯片启动后，会自动跳转到 0x18001000 地址执行，用户可以在该位置放置自己的可执行文件，或者放置 bootloader 进行二次跳转。

**通过 bootloader，客户可以实现自定义分区大小以及位置，以便实现定制化方案。**RDA 提供 bootloader 供客户参考设计。

## 4. Flash 读写接口

```
/*
 * function: erase flash
```

```
* @addr: mast be 4k alignment
* @len: must be 4k alignment. (package 64KB erase and 4KB erase for different condition automatically)
* return: 0:success, else:fail
*/
```

```
int rda5981_erase_flash(unsigned int addr, unsigned int len);
```

```
/*
* function: write flash
* @addr: mast be 256 alignment
* @buf: data to be written, best be 4 alignment
* @len: buffer len, mast be 4 alignment
* return: 0:success, else:fail
*/
```

```
int rda5981_write_flash(unsigned int addr, char *buf, unsigned int len);
```

```
/*
* function: read flash to @buf
* @addr: best be 4 alignment
* @buf: best be 4 alignment
* @len: buffer len
* return: 0:success, else:fail
*/
```

```
int rda5981_read_flash(unsigned int addr, char *buf, unsigned int len);
```

## 5. OTA 升级接口

```
/*
* function: start to wirte a partition. this func will erase given flash region
* @addr: partition start address, must be 4k alignment
* @img_len: length of image getted from OTA server, must be 4k alignment
* return: 0:success, else:fail
*/
```

```
int rda5981_write_partition_start(unsigned int addr, unsigned int img_len);
```

```
/*
* function: write image to flash, without erase.
* the write region must be inside of the area given by func rda5981_write_partition_start
* the write region must be in order, otherwise the end function will return crc error.
* the maximum length could be write once time is 0x1000
* @offset: offset from image inital position, must be 1k alignment
* @buf: data to be written
```

```
* @len: buffer len, max #0x1000, must be 1k alignment
* return: 0:success, else:fail
*/
int rda5981_write_partition(unsigned int offset, const unsigned char *buf, unsigned int len);

/*
* function: end of writing partition
* return: 0:crc32 check success, else:fail
*/
int rda5981_write_partition_end();
```

## 6. 其他

RDA MCU WIFI BBS 网址:

<http://bbs.rdamicro.com/forum.php?mod=forumdisplay&fid=36>