

RDA5981 flash 简介

1、flash 分布

目前支持的 flash 大小包括 1MB、2MB 和 4MB，不使用 OTA 和使用 OTA 时的分布情况不同。

不使用 OTA 时的 flash 分布：

0 4K 1/2/4M

| bootrom data | firmware | sys data | 3rd partner data |

 4K n1 * 4K 4K n2 * 4K

使用 OTA 时的 flash 分布：

0 4K 12K 16K 1/2/4M

| bootrom data | bootloader | firmware info | firmware | upgrade | sys data | 3rd partner data |

 4K 8K 4K n1 * 4K n2 * 4K 4K n3 * 4K

flash 中可能存在如下分区，注意每个分区的大小都是 4KB 对齐的。

bootrom data: bootrom 所使用到的数据，用户不可读写。

firmware: firmware 代码区。

firmware info: firmware 的描述信息。

upgrade: upgrade 代码区。

bootloader: 执行升级操作的 bootloader。

sys data: 存放 MAC 地址、路由器的 SSID/PASSWORD、校准值等信息。

3rd partner data: 只用百度及其合作客户在使用，不建议其他客户使用。

OTA 相关问题请参考《ota porting guide》。

2、相关分区解释

sys data:

这个分区保存了我们的一些系统数据，比如 MAC 地址，路由器的 SSID/PASSWORD，校准值等信息。**这个分区其实就是我们常说的 user data**，读取和写入这里面的数据也是通过以下接口来实现。

在 rda5991h_wland.h 中定义了：

```
int rda5981_read_user_data(unsigned char *data, unsigned short len, unsigned int flag);
```

```
int rda5981_write_user_data(unsigned char *data, unsigned short len, unsigned int flag);
```

其中 flag 在 rda5981_flash.h 有定义。

用户基本不需要调用者两个接口，一般我们又封装了一层接口给用户，比如读写 IP、MAC 地址等接口。

在 rda5991h_wland.h 中还定义了：

```
int rda5981_read_user_data_regs(unsigned char *reg, unsigned char *value, unsigned int flag);
int rda5981_write_user_data_regs(unsigned char *reg, unsigned char *value, unsigned int flag);
```

这两个是读写校准值相关的接口，假如用户往 flash 中保存校准值，最终是通过调用这两个接口实现的。用户也不需要关心这两个接口，因为在 uarthut 这个 case 下都把这些东西做好了，可以参考 uarthut 这个 case。

3rd partner data:

这个分区只有百度及其合作客户在用，用来存放客户自己的 profile。

在 rda5991h_wland.h 中定义了 4 个相关接口。

```
int rda5981_flash_read_3rdpartner_data_length(void);
int rda5981_flash_read_3rdpartner_data(unsigned char *buf, unsigned int buf_len);
int rda5981_flash_write_3rdpartner_data(const unsigned char *buf, unsigned int buf_len);
int rda5981_flash_erase_3rdpartner_data(void);
```

其他不明白的客户完全不用关心它，这些接口对你来说是无用的，这块分区就不存在，和其他的地方没有什么不同，我们有别的通用接口可以操作它。

我们 flash 的起始地址是 0x18000000，默认大小是 1MB，sys data (user data) 的默认地址是 0x180fb000，3rd partner data 的默认地址是 0x180fc000，3rd partner data 的默认长度是 4KB。

如果要改变这些地址和大小，那么在主程序最开始处，可以通过调用 rda5991h_wland.h 中的如下接口更改：

```
int rda5981_set_flash_size(const unsigned int size);
```

size 是 flash 大小，1MB 就是 0x100000，2MB 就是 0x200000，4MB 就是 0x400000。

这个 flash 大小只是一个变量，在设置 sys data 和 3rd partner data 的地址时，软件会检查这两个地址会不会超过这个大小。所以，这个接口只有在同时满足如下两个条件是才会调用：

- 1、你使用的 flash 大小超过了 1MB，即你使用的芯片是 RDA5981AM 或者 RDA5981C。
- 2、你要改变 sys data 或者 3rd partner data 的存放地址。

```
int rda5981_set_user_data_addr(const unsigned int sys_data_addr,
    const unsigned int user_data_addr, const unsigned int user_data_len);
```

这里需要注意一点，sys_data_addr 就是我们说的 sys data(user data)的地址，默认是 0x180fb000。user_data_addr 实际上是 3rd partner data 的地址，默认是 0x180fc000。user_data_len 实际上是 3rd partner data 的长度，默认是 0x1000 即 4KB。

所以对于不使用 3rd partner data 的用户，把 user_data_addr 设置成和 sys_data_addr 一样，把 user_data_len 设置成 0 就行。

假如客户用的是 RDA5981C，flash 是 4MB，想把 sys data (user data) 放在最后 4KB 的位置，那么在程序最开始处调用一次：

```
rda5981_set_flash_size(0x400000);
rda5981_set_user_data_addr(0x183FF000, 0x183FF000, 0);
```

由于目前大部分客户都在使用 OTA 功能，建议客户自己划分好各个分区的地址，在程序的最开始处调用这两个接口。如果愿意使用我们的默认配置，就不需要调用这两个接口了。

3、通用 flash 操作接口

对于有自己的数据要存放在 flash 中的客户，可以使用以下三个通用 flash 接口，在 rda5991h_wland.h 中：

```
/*
 * 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 256 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);
```

这三个接口分别是 flash 的擦除、写入和读取接口，可以操作除了最前面 4KB (bootrom data) 外的所有 flash 地址。

在使用时注意以下几点：

- 1、注意传入参数的对齐方式，有 4 字节对齐，256 字节对齐和 4KB 对齐。
- 2、flash 的 sector size 是 4KB，flash 擦除的单位是 sector，擦除的长度要 4KB 对齐，且地址要 4KB 对齐。
- 3、flash 的 page size 是 256 字节，写入的单位是 page，写入的长度要 256 字节对齐，且地址要 256 字节对齐。
- 4、写入之前，要先擦除。
- 5、flash 的地址从 0x18000000 开始，假如你要操作第 8KB 处的 flash，传入的地址是 0x18002000。
- 6、操作 flash 之前，确保你要操作的地址没有保存 code 或者其他重要数据。

、

4、总结

1、只有百度及其合作用户需要关心 3rd partner data 这个分区，对其他用户来说，这个分区不存在。

2、通常说的 user data 其实是指 sys data，函数名中带 user_data 的函数操作的也是 sys data 这个分区。

3、只要当客户使用的芯片 flash 大于 1MB，即 RDA5981AM 和 RDA5981C，并且需要更改 sys data 或者 3rd partner data 的地址时，才需要在程序最开始处调用一次：

```
int rda5981_set_flash_size(const unsigned int size);
```

这个 size 只是影响 sys data 和 3rd partner data 地址的设置。

4、sys data 中保存的都是我们定义的一些数据，如果客户要在 flash 中保存自己的数据，使用第 3 章中的通用 flash 操作接口。

5、默认 sys data 的地址是 0x180fb000。