



XRADIO FDCM Developer Guide

Revision 1.0

Nov 11, 2019

Declaration

THIS DOCUMENTATION IS THE ORIGINAL WORK AND COPYRIGHTED PROPERTY OF XRADIO TECHNOLOGY ("XRADIO"). REPRODUCTION IN WHOLE OR IN PART MUST OBTAIN THE WRITTEN APPROVAL OF XRADIO AND GIVE CLEAR ACKNOWLEDGEMENT TO THE COPYRIGHT OWNER.

THE PURCHASED PRODUCTS, SERVICES AND FEATURES ARE STIPULATED BY THE CONTRACT MADE BETWEEN XRADIO AND THE CUSTOMER. PLEASE READ THE TERMS AND CONDITIONS OF THE CONTRACT AND RELEVANT INSTRUCTIONS CAREFULLY BEFORE USING, AND FOLLOW THE INSTRUCTIONS IN THIS DOCUMENTATION STRICTLY. XRADIO ASSUMES NO RESPONSIBILITY FOR THE CONSEQUENCES OF IMPROPER USE (INCLUDING BUT NOT LIMITED TO OVERVOLTAGE, OVERCLOCK, OR EXCESSIVE TEMPERATURE).

THE INFORMATION FURNISHED BY XRADIO IS PROVIDED JUST AS A REFERENCE OR TYPICAL APPLICATIONS, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENT DO NOT CONSTITUTE A WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. XRADIO RESERVES THE RIGHT TO MAKE CHANGES IN CIRCUIT DESIGN AND/OR SPECIFICATIONS AT ANY TIME WITHOUT NOTICE.

NOR FOR ANY INFRINGEMENTS OF PATENTS OR OTHER RIGHTS OF THE THIRD PARTIES WHICH MAY RESULT FROM ITS USE. NO LICENSE IS GRANTED BY IMPLICATION OR OTHERWISE UNDER ANY PATENT OR PATENT RIGHTS OF XRADIO. THIRD PARTY LICENCES MAY BE REQUIRED TO IMPLEMENT THE SOLUTION/PRODUCT. CUSTOMERS SHALL BE SOLELY RESPONSIBLE TO OBTAIN ALL APPROPRIATELY REQUIRED THIRD PARTY LICENCES. XRADIO SHALL NOT BE LIABLE FOR ANY LICENCE FEE OR ROYALTY DUE IN RESPECT OF ANY REQUIRED THIRD PARTY LICENCE. XRADIO SHALL HAVE NO WARRANTY, INDEMNITY OR OTHER OBLIGATIONS WITH RESPECT TO MATTERS COVERED UNDER ANY REQUIRED THIRD PARTY LICENCE.

Revision History

Version	Date	Summary of Changes
1.0	2019-11-11	Initial Version

Table 1- 1 Revision History

Contents

Declaration.....	2
Revision History.....	3
Contents.....	4
Tables.....	5
Figures.....	6
1 概述.....	7
1.1 介绍.....	7
2 功能描述.....	8
2.1 模块结构.....	8
2.2 FDCM 管理的 Flash 区域.....	8
2.3 FDCM 读写数据原理.....	9
2.4 接口说明.....	9
3 使用说明.....	11
3.1 使用示例.....	11
3.2 注意事项.....	11
3.3 Q&A.....	11
3.3.1 FDCM 打开失败.....	11
3.3.2 FDCM 写入失败.....	12
3.3.3 FDCM 读取失败.....	12

Tables

表 2-1	FDCM 模块 api 说明.....	9
表 3-1	FDCM 打开失败原因分析.....	11
表 3-2	FDCM 写入失败原因分析.....	12
表 3-3	FDCM 读取失败原因分析.....	12

Figures

图 2-1 FDCM 与其他模块之间的关系.....	8
图 2-2 FDCM 管理的 Flash 区域内部结构示意图.....	8

1 概述

1.1 介绍

FDCM (Flash Data Chunk Management) 模块用于在指定的 Flash 区域对数据块进行管理。使用者调用该模块接口可方便地将数据写到 Flash 指定区域，并在需要使用数据时从 Flash 中读出。

FDCM 模块对数据块管理的主要特点为：对同一区域多次写入数据块后，从该区域读出的总是最后一次写入的数据块内容。适合用户在保存 SSID，PSK，以及其他的一些用户信息时使用，可以避免频繁地对 flash 进行擦除操作，延长 flash 使用寿命。

此文档主要介绍 FDCM 模块的基本原理和使用方法，并说明使用该模块的注意事项，以方便开发者正确使用该模块进行开发。

2 功能描述

2.1 模块结构

FDCM 模块与其他模块之间的关系如下图所示。Sysinfo, OTA 和 IMAGE 等模块中都有使用到 FDCM 模块接口，保存其对应的数据。其中，OTA 和 IMAGE 模块还会根据写在 FDCM 中的数据进行一些判断和配置。

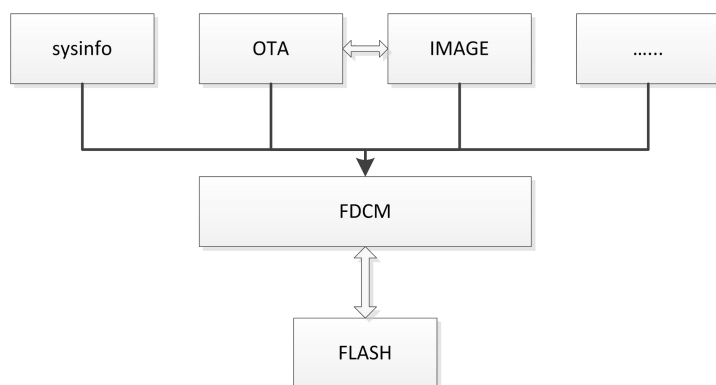


图 2-1 FDCM 与其他模块之间的关系

2.2 FDCM 管理的 Flash 区域

使用 FDCM 模块需要指定一段 Flash 区域来存储数据块，该区域必须与 Flash 可擦除的 Block 对齐。FDCM 模块管理的 Flash 区域内部结构如下图所示。

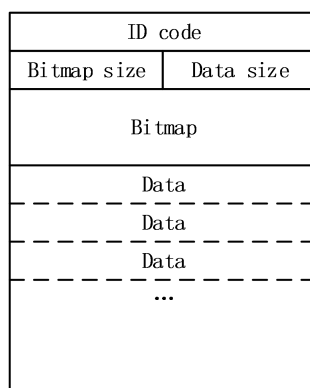


图 2-2 FDCM 管理的 Flash 区域内部结构示意图

这一段 Flash 区域内部可划分为三个部分：FDCM header 区域、Bitmap 区域和 Data 区域。FDCM header 区域包含 ID code、Bitmap size 和 Data size。其中，ID code 是一个特定值，作为 FDCM 管理的一段 Flash 区域的标志。Bitmap size 表示 Bitmap 区域的大小，在第一次写入数据块时根据数据块的大小和这一段 Flash 区域的大小计算得到。Bitmap 用于标记最新写入的数据块位于 Data 区域的具体位置。Data size 表示 Data 区域中一个

数据块的大小，在第一次写入数据块时确定，Data 区域中可存储多个大小相同的数据块。

2.3 FDCM 读写数据原理

第一次写入数据块时，FDCM 模块将构建上述内部结构。然后修改 Bitmap，并将数据块写在 Data 区域第一个数据块位置。之后每次写入相同大小数据块时，都修改一次 Bitmap，并将数据块写在前一次写入数据块的后一个位置。当 Data 区域写满后，将擦除这一段 Flash 区域，并重复第一次写入数据块时的操作。当某次写入的数据块大小与上一次写入的数据块大小不同（即写入数据块大小不等于 Flash 区域上的 Data size）时，将擦除这一段 Flash 区域，并重复第一次写入数据块时的操作。

用 FDCM 读数据块时，将先核对 ID code，然后根据 Bitmap size、Data size 以及 Bitmap 找到并读取出最后一次写入的数据块。由于在第一次写数据块时才构建区域的内部结构，所以在未写过数据块的区域进行读数据块操作将报错。

2.4 接口说明

下面对 FDCM 模块提供的接口进行简要说明。

表 2-1 FDCM 模块 api 说明

function	detail
fdcm_open();	<p>声明：fdcm_handle_t *fdcm_open(uint32_t flash, uint32_t addr, uint32_t size);</p> <p>目的：创建 fdcm 操作区域</p> <p>参数：flash: flash 设备号</p> <p>addr: flash 区域的起始地址</p> <p>size: flash 区域的大小</p> <p>返回值：对指定 Flash 区域进行读写操作的 handle</p>
fdcm_read();	<p>声明：uint32_t fdcm_read(fdcm_handle_t *hdl, void *data, uint16_t data_size);</p> <p>目的：读取 fdcm 区域数据块</p> <p>参数：hdl: fdcm 读写操作的 handle</p> <p>data: 待存放结果的 buffer</p> <p>size: 待读取的数据块长度</p> <p>返回值：读取的数据块长度</p>
fdcm_write();	<p>声明：uint32_t fdcm_write(fdcm_handle_t *hdl, const void *data, uint16_t data_size);</p> <p>目的：往 fdcm 区域写入数据块</p> <p>参数：hdl: fdcm 读写操作的 handle</p> <p>data: 待写入的数据块</p>

	<p>size: 待写入的数据块长度</p> <p>返回值: 写入的数据块长度</p>
<code>fdcm_close();</code>	<p>声明: <code>void fdcm_close(fdcm_handle_t *hdl);</code></p> <p>目的: 关闭 fdcm 区域, 释放该资源</p> <p>参数: hdl: fdcm 读写操作的 handle</p> <p>返回值: 无</p>

3 使用说明

3.1 使用示例

以对 Flash 0 起始地址为 0x00200000 大小为 4KB 的 flash 区域进行 FDCM 操作为例：

1. 获取对该 Flash 区域操作的 handle 指针。

```
fdcm_handle_t *hdl;
hdl = fdcm_open(0, 0x00200000, 4096);
```

2. 利用 handle 指针向该 Flash 区域写入数据块，假设 wr_data1, wr_data2 均为要写入的数据块。

```
fdcm_write(hdl, &wr_data1, sizeof(wr_data1));
fdcm_write(hdl, &wr_data2, sizeof(wr_data2));
```

3. 利用 handle 指针从该 Flash 区域读取数据块，将最后一次写入的数据块读取到 rd_data 中。

```
fdcm_read(hdl, &rd_data, sizeof(rd_data));
```

4. 不再用 FDCM 管理该 Flash 区域后，释放 handle。

```
fdcm_close(hdl);
```

此外，可参考 SDK 中提供的 FDCM 工程示例：

sdk-code/project/example/fdcm

3.2 注意事项

在使用 FDCM 模块时，有以下几点需要注意：

1. 用 FDCM 模块管理的 Flash 区域不要与其他模块（如 Image）使用的 Flash 区域产生冲突，避免相互影响。
2. 用 FDCM 模块管理的 Flash 区域必须与 Flash 可擦除的 block 对齐，即区域起始地址与 Flash 擦除 Block 的边缘对齐，并且区域大小为 Flash 可擦除最小 Block 大小的整数倍。
3. 先正确写入数据块后才能正确读出数据块，读数据块时的参数 data_size 应该与最后一次写数据块的参数 data_size 相同。
4. 连续多次写数据块允许每次写数据块的 data_size 不同，但根据 FDCM 读写数据原理，建议对同一 Flash 区域每次读写数据块的 data_size 尽可能一致，这样能减少对 Flash 擦除的次数。

3.3Q&A

3.3.1 FDCM 打开失败

表 3-1 FDCM 打开失败原因分析

可能原因	措施
Flash 地址参数没有与可擦除的 block 对齐	尝试传入 4k 对齐或者 64k 对齐的 flash 地址

3.3.2 FDCM 写入失败

表 3-2 FDCM 写入失败原因分析

可能原因	措施
没有先调用 fdcn_open(), 句柄为空	请先调用 fdcn_open(), 获取句柄后再进行写入操作
写入长度超出 open 时设定的 size 大小	检查写入长度和 open 时设定长度, 调整后再作尝试

3.3.3 FDCM 读取失败

表 3-3 FDCM 读取失败原因分析

可能原因	措施
该 flash 区域还没有写入过数据	先写入数据, 之后再进行读取操作
读取长度与保存的数据长度不匹配	确认之前保存的数据长度之后再尝试读取