

# OPL1000

ULTRA-LOW POWER 2.4GHZ WI-FI + BLUETOOTH SMART SOC

## Sensor Reference via Coolkit Cloud Application Guide



OPULINKS

---

<http://www.opulinks.com/>

Copyright © 2019, Opulinks. All Rights Reserved.

---

OPL1000-Reference-Sensor-Coolkit-cloud-guide | Version V04

Date	Version	Contents Updated
2019-11-08	0.1	<ul style="list-style-type: none"><li>Initial Release</li></ul>
2019-11-13	0.2	<ul style="list-style-type: none"><li>Add update Device ID and MAC address section.</li></ul>
2020-01-09	0.3	<ul style="list-style-type: none"><li>Add chapter 5 for application development.</li></ul>
2020-02-28	0.4	<ul style="list-style-type: none"><li>Update proj_src folder structure.</li></ul>

## TABLE OF CONTENTS

1. 介绍	1
1.1. 文档应用范围	1
1.2. 缩略语	1
1.3. 参考文献	1
2. 项目构成和工作原理	2
2.1. 项目构成	2
2.2. 工作原理	2
3. 运行酷宅门磁参考设计	4
3.1. 生成 OPL1000 设备固件	4
3.2. 更新设备 ID 和 MAC 地址	5
3.2.1. 接线方式	5
3.2.2. 确认 AT 能正常运行	5
3.2.3. 写入 Device ID 和 MAC 到 flash	6
3.3. 易微联 APP 完成蓝牙配网	8
3.4. APP 功能界面	11
4. 酷宅门磁 应用设计	12
4.1. 项目工程构成	12
4.2. 参数配置 blewifi_configuration.h 使用说明	13
4.3. 执行流程和模块说明	13
4.3.1. 执行流程	13
4.3.2. 主要 Task Handler	14
5. 酷宅门磁应用开发	15
5.1. 收发消息相关的更新	15
5.2. 外设控制	16
5.2.1. 外设状态的收集	16
5.2.2. 外设状态的上报	17
5.3. Heap size 的调整	18
5.4. 低功耗相关的注意事项	21

## LIST OF FIGURES

Figure 1:项目文件 .....	2
Figure 2:工作原理图 .....	3
Figure 3: CoolKit 與 AT UART 串口接线图 .....	5
Figure 4: 确认 AT 指令正常运行 .....	5
Figure 5: 写入 Device ID .....	6
Figure 6: 切换 AT UART 到 DBG UART .....	6
Figure 7: 写入 BLE MAC .....	7
Figure 8: 写入 Wi-Fi MAC .....	7
Figure 9: 切换 DBG UART 到 AT UART .....	7
Figure 10: 复位酷宅开发板 .....	7
Figure 11: 易微联 APP OPL1000 设备列表 .....	8
Figure 12:网络连接界面 .....	9
Figure 13: 易微联 APP 连云成功 .....	10
Figure 14:云端查看设备状态 .....	10
Figure 15:酷宅门磁功能界面 .....	11
Figure 16:工程文件构成 .....	12
Figure 17:固件执行流程图 .....	14
Figure 18: JSON 数据格式的定义 .....	15
Figure 19: 门磁状态的收集 .....	16
Figure 20: 门磁状态的上报 .....	17
Figure 21: 从 map 文件中获取首末地址 .....	19
Figure 22: 更新 SCT file 中的 size .....	19
Figure 23: 更新 heap size 相关的信息 .....	20
Figure 24: smart sleep 相关的宏定义 .....	21

# 1. 介绍

## 1.1. 文档应用范围

本文档介绍基于 OPL1000 A2 芯片的智能门磁控制参考设计。本设计中连接到酷宅云，使用 https 完成门磁的控制和状态更新。内容包括固件设计，云端设备配置以及操作过程。

## 1.2. 缩略语

Abbr.	Explanation
AP	Wireless Access Point 无线访问接入点
APP	APPLication 应用程序
APS	Application Sub-system 应用子系统，在本文中亦指 M3 MCU
Blewifi	BLE config WIFI 蓝牙配网应用
DevKit	Development Kit 开发工具板
OTA	Over-the-Air Technology 空间下载技术
TCP	Transmission Control Protocol 传输控制协议

## 1.3. 参考文献

[1] OPL1000 数据手册 OPL1000-DS-NonNDA.pdf

[2] Download 工具使用指南 OPL1000-patch-download-tool-user-guide.pdf

访问链接: <https://github.com/Opulinks-Tech/OPL1000A2-SDK/tree/master/Doc/OPL1000A2-patch-download-tool-user-guide.pdf>

[4] SDK 开发使用指南 OPL1000-SDK-Development-guide.pdf

访问连接: <https://github.com/Opulinks-Tech/OPL1000A2-SDK/blob/master/Doc/OPL1000-SDK-Development-guide.pdf>

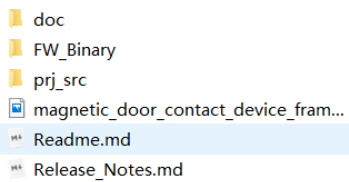
访问连接: <https://github.com/Opulinks-Tech/>

## 2. 项目构成和工作原理

### 2.1. 项目构成

酷宅门磁项目需要下载易微联 APP，用于 OPL1000 蓝牙配网以及设备的状态显示及操作（APP 软件可以在手机应用市场直接搜索‘易微联’）；

Figure 1:项目文件



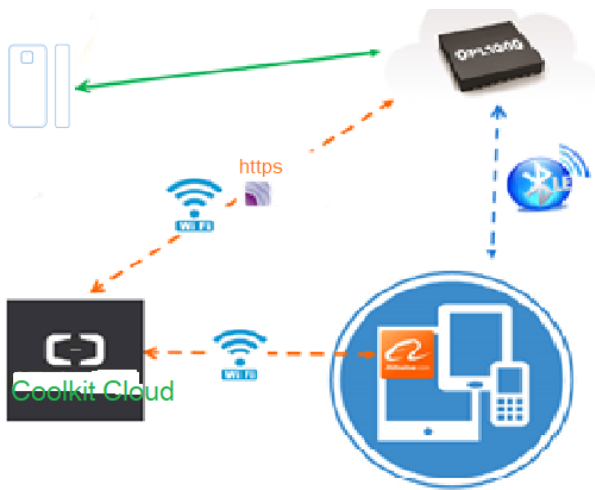
酷宅门磁项目主要由五类内容构成，说明如下

目录和文件	说明
doc	存放 应用指南文档，即本文档
FW_Binary	存放做好的 bin 文件和 Pack 脚本文件，可以直接下载使用
prg_src	包含酷宅门磁项目的库文件以及全部工程文件
Readme.md	说明本参考设计功能和内容
Release_Notes.md	描述本版本发布更新内容和注意事项

### 2.2. 工作原理

酷宅门磁参考设计主要部件：物联网模块 OPL1000，移动设备（APP），云端（Coolkit cloud）和门磁固件。

Figure 2:工作原理图



### 3. 运行酷宅门磁参考设计

运行 OPL1000 酷宅门磁应用需要以下步骤：

- a) 更新工程配置文件，修改头文件中的宏定义参数（参考 3.1 章节）。
- b) 使用编译工具完成项目工程编译，生成 M3 bin 文件（二进制固件文件）。
- c) 通过 download tool 打包 M3 bin 文件，生成完整固件 opl1000.bin，并下载到 opl1000 模块。
- d) 打开易微联 app，进行蓝牙扫描动作，扫描 opl1000 蓝牙设备，配置连接能够访问 Internet 的 AP。
- e) OPL1000 的固件连接云端。通过易微联 app 显示门磁的状态。

#### 3.1. 生成 OPL1000 设备固件

编译酷宅门磁项目工程文件可以生成 OPL1000 M3 固件。在编译之前用户可以根据需要自行修改参数及头文件。

使用 Keil C 手动更新参数配置需要分三步完成：

1. 第一步使用 Keil C 开发工具打开头文件（blewifi\_configuration.h），是否进入省电模式，设备名称等参数；酷宅云四元组由下面的四个宏定义确定(注意：因为目前酷宅没有提供申请设备 ID 的正式渠道，所以在编译时请保持下面的值不变。本文将在 3.4 节介绍如何更新设备 ID 的过程)：

```
#define APIKEY      "00000000-0000-0000-0000-000000000000"  
#define DEVICE_ID   "000000000000"  
#define CHIP_ID     "000000000000"  
#define MODEL_ID    "000-000-00"
```

2. 第二步编译工程，生成 M3 Bin 文件。
3. 第三步使用 download 工具完成固件 Pack，下载操作，请参考 [“Download 工具使用指南 OPL1000-patch-download-tool-user-guide.pdf”](#) 了解操作方法。



## 3.2. 更新设备 ID 和 MAC 地址

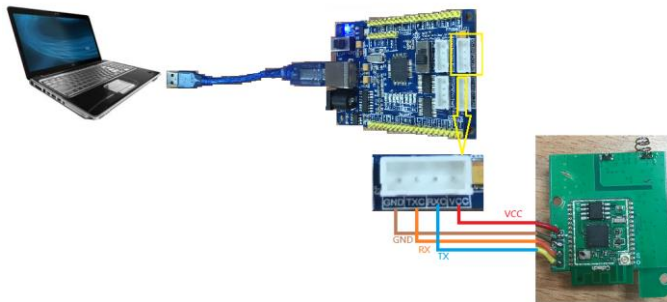
为了使门磁设备能够连上酷宅云并正常使用，用户需要在下载 3.1 节中生成的 bin 后，完成更新设备 ID 和 MAC 地址的过程如下：

### 3.2.1. 接线方式

CoolKit 板開放 AT UART 串口，即用來与 miniUSB 连接。

AT UART 串口接线如下：

Figure 3: CoolKit 與 AT UART 串口接线图



### 3.2.2. 确认 AT 能正常运行

在下载 3.1 节中生成的 bin 到 CoolKit 板后，打开 AT UART 串口后输入 'ATE1'，打开串口回显功能，然后输入 'AT' 命令，如果返回 'OK' 则说明 AT 能正常运行。

Figure 4: 确认 AT 指令正常运行

```
>AT
OK
```

### 3.2.3. 写入 Device ID 和 MAC 到 flash

写入 Device ID、Wi-Fi MAC 和 BLE MAC 到 FLASH 都以 AT 命令的方式实现。(注意：目前还没有官方的申请酷宅 Device ID、Wi-Fi MAC 和 BLE MAC 的渠道，需要找酷宅相应的 PM 或 opulinks Sales 申请)。

1. 使用 AT 指令，写入 Device ID。(以下指令为范例，请参考实际所需指令)。

**注意**，要在 5 秒内连续输入两行资料。

AT 指令：

```
at+writefim=0x01020001,0,156
```

```
34,36,34,62,39,31,65,66,2D,65,65,35,61,2D,34,34,30,33,2D,39,64,61,37,2D,36,32,37,38,65,30,30,6
6,36,39,65,63,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
0,00,00,00,00,31,30,30,30,39,65,36,35,66,65,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
0,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,64,30,3A,32,37,3A,30,31,3A,33,63,3A,63,39,3A,
34,63,00,00,00,4F,50,4C,2D,44,4D,41,2D,47,4C,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
```

Figure 5: 写入 Device ID

```
>at+writefim=0x01020001,0,156
OK
OK
>
```

第二行指令不会显示在终端上。出现两个“OK”代表成功。

2. 切换 AT UART 到 DBG UART。AT 指令：at+switchdbg

Figure 6: 切换 AT UART 到 DBG UART

```
>at+switchdbg
C
Switch: Dbg UART
>
```

3. 输入指令写入 BLE MAC. AT 指令 : lecfg bd\_addr 0xd027013cc94d

Figure 7: 写入 BLE MAC

```
>lecfg bd_addr 0xd027013cc94d
>
```

4. 输入指令写入 WiFi MAC. AT 指令 : wpa\_mac d0:27:01:3c:c9:4c 和 wpa\_dbg ib 0 1

Figure 8: 写入 Wi-Fi MAC

```
>wpa_mac d0:27:01:3c:c9:4c
>wpa_dbg ib 0 1
>
```

5. 切换 DBG UART 到 AT UART. AT 指令 : at+switchdbg

Figure 9: 切换 DBG UART 到 AT UART

```
>switchat
δ
Switch: AT UART
>
```

6. 输入指令复位酷宅开发板。指令 : at+rst

Figure 10: 复位酷宅开发板

```
>at+rst
OK
<CHECK>
SPI load patch, last index 0 result 2
BootMode 10
>
```

### 3.3. 易微联 APP 完成蓝牙配网

在更新好 Device ID 和 MAC 地址后, 在设备端按下按钮, 红色指示灯亮, 长按按钮直到红色指示灯快闪后即可松开按钮, 设备进入蓝牙配网状态 (长按时间约为 6 秒左右)。

然后在 APP 点击+添加设备, APP 会自动扫描附近的 OPL1000 蓝牙信息。然后点击+配置需要连接的 AP

Figure 11: 易微联 APP OPL1000 设备列表



点击需要配网的 OPL1000 设备右侧的 ">" 符号。选择 AP,并输入密码。点击确定后开始连接酷宅云。

Figure 12:网络连接界面



连云成功后，根据需要修改设备名称后，点击“完成添加”

Figure 13: 易微联 APP 连云成功



设备连接成功后，可以查看设备和门磁的状态如下：

Figure 14: 云端查看设备状态



### 3.4. APP 功能界面

下图是 APP 功能界面，APP 主要由三部分构成：电量显示，门磁状态显示和历史记录。

本参考设计通过打开或闭合控制门磁的状态。若门磁闭合，则状态为闭合状态；同时会新增加一条新的记录。

Figure 15:酷宅门磁功能界面



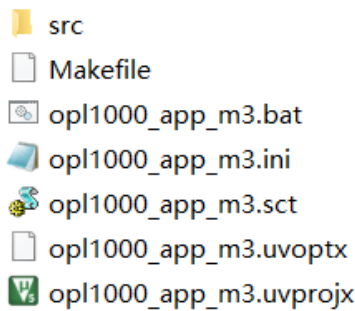
## 4. 酷宅门磁 应用设计

本章介绍设备端固件工作原理，以及如何进行功能扩展。

### 4.1. 项目工程构成

酷宅门磁项目包含蓝牙配网，酷宅门磁，和库文件等目录。

Figure 16:工程文件构成



各文件夹及文件构成如表。具体内容如 Table 1 所述。

Table 1: 酷宅门磁项目文件夹和内容

文件夹和文件	内容说明
src	存放蓝牙配网，数据收发相关.c 和.h 头文件，以及 main 文件
src/iot_data	收发 IOT 数据相关的代码
src/sensor	跟门磁传感器处理相关的代码
opl1000_app_m3.bat opl1000_app_m3.ini opl1000_app_m3.sct opl1000_app_m3.uvoptx opl1000_app_m3.uvprojx	编译工程文件。



4.2. 参数配置 blewifi\_configuration.h 使用说明

blewifi\_configuration.h 文件集中了需要配置的参数，用户可以根据实际应用更新参数配置。

blewifi\_configuration.h 文件定义了可配置参数的默认值。

Table 2 主要参数配置宏定义功能详细介绍

宏定义	说明
MW_FIM_VER11_PROJECT	Group11 的 FIM 版本信息，取值范围为 0x00-0xFF. <b>Notes:</b> 当该文件中的宏定义值有更新时，请务必更新一下这个值（只要跟原来的值不一样就好）。
MW_FIM_VER12_PROJECT	Group12 的 FIM 版本信息，取值范围为 0x00-0xFF. <b>Notes:</b> 当该文件中的宏定义值有更新时，请务必更新一下这个值（只要跟原来的值不一样就好）。
BLEWIFI_COM_POWER_SAVE_EN	是否 Enable smart sleep. 1: Enable. 0: Disable
APIKEY · DEVICE_ID · CHIP_ID · MODEL_ID	用于标志设备的元组。
BLEWIFI_COM_RF_POWER_SETTINGS	用于设置 RF 模式。具体取值请参考该文件的注释。

4.3. 执行流程和模块说明

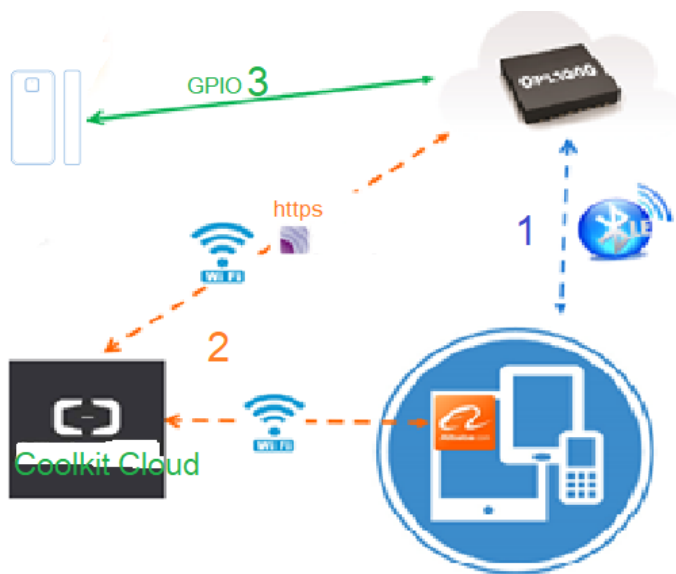
本章节介绍 OPL1000 固件处理流程。

4.3.1. 执行流程

主程序执行流程如下图所示。在完成设备和智能门磁初始化操作后，设备将自动尝试连接酷宅云。

如果连接成功，用户就可以在手机端通过易微联 APP 查看智能门磁的状态。

Figure 17:固件执行流程图



#### 4.3.2. 主要 Task Handler

本项目内部启动了两个任务处理器

##### 1. BLE Handler

BLE Handler 功能是等待手机端蓝牙与 OPL1000 的连接，此时 OPL1000 会持续发送 BLE 广播，直到蓝牙建立连接

##### 2. WIFI Handler

WIFI Handler 是 OPL1000 与 AP 建立连接后，连线及断线检查，断线后重连功能

## 5. 酷宅门磁应用开发

为方便用户在本例程的基础上做二次开发，本章将介绍一些基于酷宅门磁应用开发过程中需要改动的代码和常见问题的处理方法。

### 5.1. 收发消息相关的更新

用户做二次开发时，通常需要为设备规划好收发消息的数据包的格式（JSON），并对其做相应地修改。在本例程中，JSON 数据格式的定义在 sensor\_https.c 文件中实现如下：

Figure 18: JSON 数据格式的定义

```
#define CALC_AUTH_KEY_FORMAT "{\"apikey\": \"%s\", \"deviceid\": \"%s\", \"d_seq\": \"%u\"}"  
#define POST_DATA_DEVICE_ID FORMAT "{\"deviceid\": \"%s\", \"d_seq\": \"%u\", \"params\": {"  
#define POST_DATA_SWITCH_FORMAT "\"switch\": \"%s\", "  
#define POST_DATA_BATTERY_FORMAT "\"battery\": %s, "  
#define POST_DATA_FWVERSION FORMAT "\"fwVersion\": \"%s\", "  
#define POST_DATA_TYPE_FORMAT "\"type\": %d, "  
#define POST_DATA_CHIP_FORMAT "\"chipID\": \"%02x%02x%02x%02x%02x%02x\","  
#define POST_DATA_MACADDRESS_FORMAT "\"mac\": \"%02x%02x%02x%02x%02x%02x\","  
#define POST_DATA_RSSI_FORMAT "\"rssi\": %d}})"  
#define OTA_DATA_URL "%s?deviceid=%s&ts=%u&sign=%s"  
#define SHA256_FOR_OTA_FORMAT "%s%s"  
#define CKS_FW_VERSION_FORMAT "%d.%d.%d"  
  
#define POST_DATA_INFO1_FORMAT "\"deviceid\": \"%s\", \"time\": \"%u\", \"switch\": \"%s\", \"battery\": %s,"  
#define POST_DATA_INFO2_FORMAT "\"fwVersion\": \"%s\", \"type\": %d, \"rssi\": %d\""
```

```
#define BODY_FORMAT "POST %s %s\r\n%s%\r\n%s%\r\n%s%\r\n%s%d\r\n\r\n%s"
```

## 5.2. 外设控制

由于酷宅门磁应用只有上行数据的发送，所以本例程的外设控制只涉及外设状态的收集和上报两个方面。

### 5.2.1. 外设状态的收集

外设状态的收集指的是 OPL1000 通过 GPIO 口采集到各类传感器数据的过程。这个数据可以是灯的开或关的状态，物体的重量，和门磁的闭合状态等。这个过程会涉及 GPIO pin 的定义，API 接口函数的定义和调用位置等问题。

在本例程中，酷宅门磁的状态的收集用到的是 GPIO\_IDX\_05 pin(在 blewifi\_configuration.h 文件中定义)，由 Hal\_Vic\_GpioInput() 函数读取该引脚的电平，然后设置门磁的状态并把数据 push 到 ring buffer。

Figure 19: 门磁状态的收集

```
static void BleWifi_Ctrl_TaskEvtHandler_DoorDebounceTimeout(uint32_t evt_type, void *data, int len)
{
    BLEWIFI_INFO("BLEWIFI: MSG BLEWIFI_CTRL_MSG_DOOR_DEBOUNCETIMEOUT \r\n");
    unsigned int u32PinLevel = 0;

    // Get the status of GPIO (Low / High)
    u32PinLevel = Hal_Vic_GpioInput(MAGNETIC_IO_PORT);
    printf("MAG_IO_PORT pin level = %s\r\n", u32PinLevel ? "GPIO_LEVEL_HIGH:OPEN" : "GPIO_LEVEL_LOW:CLOSE");

    // When detect falling edge, then modify to raising edge.

    // Voltage Low / DoorStatusSet - True / Door Status - Close - switch on - type = 2
    // Voltage Hight / DoorStatusSet - False / Door Status - Open - switch off - type = 3
    if(GPIO_LEVEL_LOW == u32PinLevel)
    {
        /* Disable - Invert gpio interrupt signal */
        Hal_Vic_GpioIntInv(MAGNETIC_IO_PORT, 0);
        // Enable Interrupt
        Hal_Vic_GpioIntEn(MAGNETIC_IO_PORT, 1);

        if (BleWifi_Ctrl_EventStatusGet(BLEWIFI_CTRL_EVENT_BIT_DOOR) == false)
        {
            /* Send to IOT task to post data */
            BleWifi_Ctrl_EventStatusSet(BLEWIFI_CTRL_EVENT_BIT_DOOR, true); // true is Door Close
            Sensor_Data_Push(BleWifi_Ctrl_EventStatusGet(BLEWIFI_CTRL_EVENT_BIT_DOOR), BleWifi_Ctrl_EventStatusGet(
                Iot_Data_TxTask_MsgSend(IOT_DATA_TX_MSG_DATA_POST, NULL, 0);
            ));
        }
    }
    else
    {

```

### 5.2.2. 外设状态的上报

外设状态的上报指的是 OPL1000 在收集到外设相关的数据后，把它封装为 JSON 格式的数据包，并发布(publish)到酷宅云端的过程。

在本例程中，门磁状态的上报由 Sensor\_Https\_Post () 函数实现, 它存在于 sensor\_https.c 文件中，在 build 好数据后由 Sensor\_Https\_Post\_On\_Line () 调用实现数据上报。

Figure 20: 门磁状态的上报

```
int Sensor_Https_Post_On_Line(void)
{
    int len = 0;
    int PostResult = 0;
    int Count = 0;
    HttpPostData_t PostContentData;

    ...

    do
    {
        PostResult = Sensor_Https_Post(g_ubSendBuf, len);
        Count++;

        if (SENSOR_DATA_FAIL == PostResult)
            osDelay(100);

        // If device doesn't get IP, then break (no retry any more).
        if (false == BleWifi_Ctrl_EventStatusGet(BLEWIFI_CTRL_EVENT_BIT_GOT_IP))
        {
            printf("IP is gone , will not post count = %d\n", Count);
            break;
        }
    } while((SENSOR_DATA_OK != PostResult) && (Count < POST_FAILED_RETRY));
}
```

### 5.3. Heap size 的调整

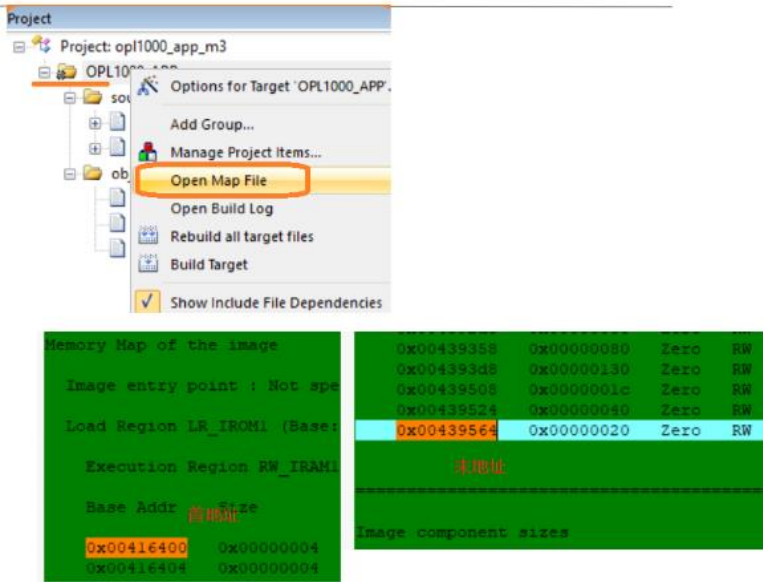
随着功能的不断加入，应用程序可能会碰到由于代码的不断增加，导致空间不够而引起的宕机问题.用户可以参考本节的内容，对 scatter file, heap size 做适当调整，尝试着解决内存不够问题。

( 1 ) 在调整 scatter file 之前，需要在 map 文件获取首末地址，方法如下：

- 用 keil 打开相应的工程
- 右键点击工程,选择'Open Map File'选项
- 在打开的.map 文件中获取 image 在 Memory Map 中的首地址和末地址。

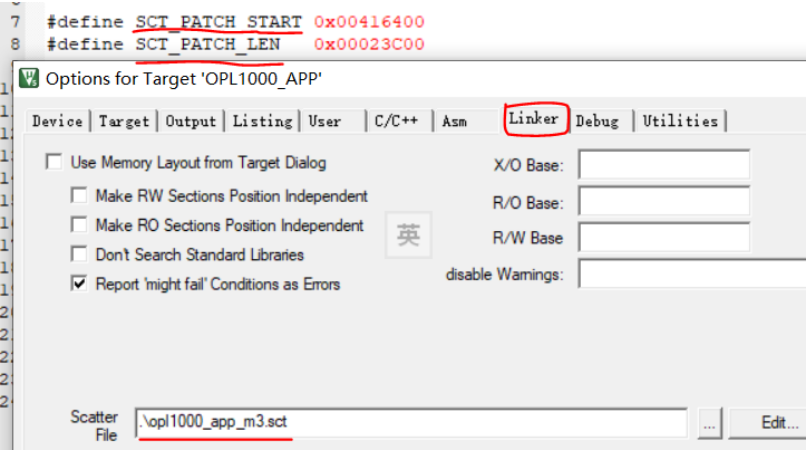
在该例子中，首地址是：0x00416400，末地址是：0x00439564。如下图所示。对末地址以 4k 或其它值(1k,512B 等)为单位向上取整，为 0x0043a000.

Figure 21: 从 map 文件中获取首末地址



- (2) 在 Memory Map 中获取首、末地址后，就可根据它们对 scatter file 做相应调整，方法如下：
- 右键点击工程，选择 'Options' 选项，选择 'Linker'，再点击 'Edit' scatter file，如下图所示
  - 保持 SCT\_PATCH\_START 的值不变，而 SCT\_PATCH\_LEN 则可以改为末地址 - 首地址的值，如在该例中，大小 = 0x0043a000 - 0x00416400 = 0x00023C00。

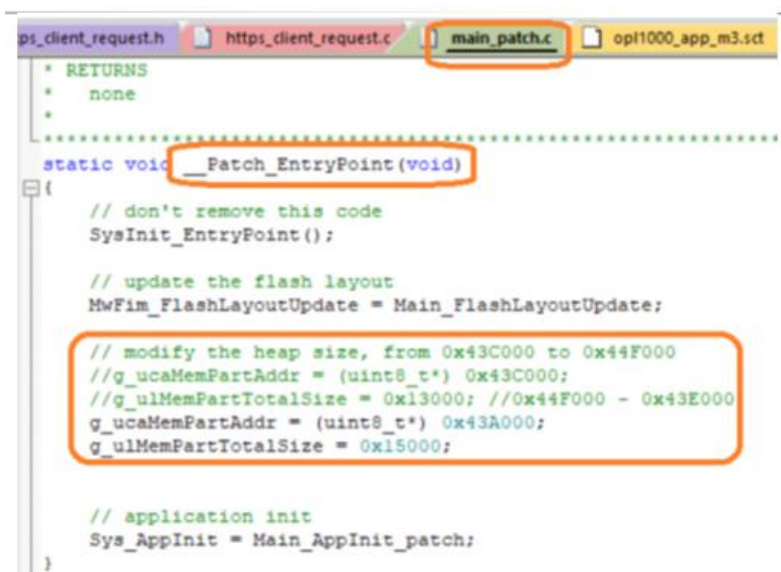
Figure 22: 更新 SCT file 中的 size



(3) 在修改 scatter file 文件后，就可根据它设置 Heap 的起始地址，并计算出它的大小，方法如下：

- Heap 的设置通常都放在工程的 main\_patch.c 文件的 \_\_Patch\_EntryPoint() 中进行；
- Heap 的起始地址可以设置为末地址，在该例中也就是的 0x0043a000。大小 = 0x44F000 - 末地址，对于下图中的例子，就是，大小 = 0x44F000 - 0x0043a000 = 0x15000

Figure 23: 更新 heap size 相关的信息



```
ps_client_request.h  https_client_request.c  main_patch.c  opl1000_app_m3.sct
* RETURNS
* none
*
*****
static void __Patch_EntryPoint(void)
{
    // don't remove this code
    SysInit_EntryPoint();

    // update the flash layout
    MwFim_FlashLayoutUpdate = Main_FlashLayoutUpdate;

    // modify the heap size, from 0x43C000 to 0x44F000
    //g_ucaMemPartAddr = (uint8_t*) 0x43C000;
    //g_ulMemPartTotalSize = 0x13000; //0x44F000 - 0x43E000
    g_ucaMemPartAddr = (uint8_t*) 0x43A000;
    g_ulMemPartTotalSize = 0x15000;

    // application init
    Sys_AppInit = Main_AppInit_patch;
}
```



### 5.4. 低功耗相关的注意事项

在 `blewifi_configuration.h` 文件中定义了三个跟低功耗有关的宏：`BLEWIFI_COM_POWER_SAVE_EN`，`BLEWIFI_WIFI_DTIM_INTERVAL`，和 `BLEWIFI_COM_RF_POWER_SETTINGS` 宏。

其中，`BLEWIFI_COM_POWER_SAVE_EN` 控制是否使能 smart sleep 模式，它的默认值为 1，即使能的。

Figure 24: smart sleep 相关的宏定义

```
blewifi_configuration.h
17 FIM version
18 */
19 #define MW_FIM_VER11_PROJECT      0x03    // 0x00 ~ 0xFF
20
21 /*
22 Smart sleep
23 */
24 #define BLEWIFI_COM_POWER_SAVE_EN  (1)    // 1: enable    0: disable
```

用户可根据需要自行调用以下接口函数(在 `ps_public.h` 文件中定义)：

```
void ps_smart_sleep(int enable);
```

`BLEWIFI_WIFI_DTIM_INTERVAL` 用于设置 DTIM 的默认值，该值越大，功耗越低，响应时间也相应加长。

用户可根据需要自行调用以下接口函数(在 `blewifi_wifi_api.h` 文件中定义)：

```
int BleWifi_Wifi_SetDTIM(uint32_t value);
```

`BLEWIFI_COM_RF_POWER_SETTINGS` 用于设置 BLE 和 WIFI 模块的功耗模式的默认值，默认值为 0x00，

用户可根据需要自行调用以下接口函数(在 `blewifi_common.h` 文件中定义)：

```
void BleWifi_RFPowerSetting(uint8_t level);
```

## CONTACT

[sales@Opulinks.com](mailto:sales@Opulinks.com)