

OPL1000

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

Scaler Reference Design Application Guide



OPULINKS

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

Copyright © 2017-2019, Opulinks. All Rights Reserved.

OPL1000-Reference-scaler-application-guide-R01 | Version V01

Date	Version	Contents Updated
2019-03-13	0.1	<ul style="list-style-type: none">Initial Release

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	3
2.1. 项目构成	3
2.2. 工作原理	4
3. 运行 OPL 智能秤项目	6
3.1. 生成 OPL1000 设备固件	6
3.1.1. 使用 ParamCfg 工具生成和下载固件	6
3.1.2. 使用 Keil C 和 Download 工具编译下载固件	8
3.2. OPL APP 完成蓝牙配网	8
3.3. 检查 OPL 设备工作状态	9
3.4. Ali APP 监测体重	10
4. OPL 体重计应用设计	11
4.1. 项目工程构成	11
4.2. 功能模块说明	12
4.2.1. OPL1000 固件	12
4.2.2. MEGA2560	14
4.2.3. 云连接和数据传输	14
4.2.4. 称重模块	15
4.2.5. 工程整体内存分配	16
5. Ali 云 IOT 设备创建	19
5.1. 创建 IOT 设备	19
5.2. IOT 设备三元组	19
6. Ali 云 APP 生成	21
6.1. 界面设计	21
6.2. 账号配置及手机 APP 设置	22
6.3. APP 生成和安装	22

LIST OF FIGURES

Figure 1:项目文件.....4

Figure 2:工作原理图4

Figure 3: 阿里云三元组参数配置6

Figure 4:ParamCfg 更改三元组参数界面7

Figure 5:手机 APP 蓝牙扫描列表.....8

Figure 6:手机 APP 端扫描 AP 界面8

Figure 7:阿里云连接串口 log 状态信息9

Figure 8:阿里云官网设备在线状态信息9

Figure 9:Ali APP 体重检测界面 10

Figure 10:工程文件构成 11

Figure 11:固件执行流程图..... 13

Figure 12:MQTT 实现方式图..... 14

Figure 13:智能秤电路图 15

Figure 14:获取首末地址 16

Figure 15:修改 scatter file..... 17

Figure 16:调整 HeapSize 17

Figure 17:调整 Stack Size 18

Figure 18:阿里云 APP 创建界面 19

Figure 19:阿里云设备三元组创建..... 20

Figure 20:阿里云 APP 界面设计 21

Figure 21:生成 Android APP 22

LIST OF TABLES

Table 1: Sensor 项目文件夹和内容..... 11

1. 介绍

1.1. 文档应用范围

本文档介绍如何基于 OPL1000 开发智能秤类型完整应用。内容包括固件设计，Arduino MEGA2560 模块设计，云端设备配置，手机 APP 设计以及操作过程。本参考设计基于 Ali 云的 MQTT 协议实现。

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
MQTT	Message Queuing Telemetry Transport 消息队列遥测传输协议
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/OPL1000-SDK/tree/master/Doc/OPL1000-patch-download-tool-user-guide.pdf>

[3] 通用参数配置工具使用指南 OPL1000-Parameter-Configure-Tool-guide.pdf

[4] Ali Cloud 移动端开发官方文档

访问连接：<https://linkdevelop.aliyun.com/studiomobile-doc#mobile-wswg-editor-summary.html>

[5] Ali Cloud 物联网产品及设备创建官方指导文档

访问连接：

https://help.aliyun.com/document_detail/73705.html?spm=a2c4g.11186623.4.2.4d74577a4OAW4r

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

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

2. 项目构成和工作原理

2.1. 项目构成

旺凌科技的智能秤项目主要由两部分组成：一是共有资料，二是智能秤项目特有部分。

共有资料对所有参考设计是共同的，它包括两部分：

第一部分是工具类资料，保存在 Tool 目录下，包括

- 1) 通用参数配置工具 ParamCfg.exe 软件。ParamCfg.exe 用于工程文件中工作参数的修改，具体为修改若干头文件中宏定义的值。具体使用参考“通用参数配置工具使用指南 OPL1000-Parameter-Configure-Tool-guide.pdf”文档。
- 2) 固件下载工具 download_RELEASE.exe，它用于 OPL1000 固件的生成和下载。
- 3) 蓝牙配网工具 opulinks_iot_app.apk，它是 Android 手机应用程序，用于 OPL1000 蓝牙配网（这两款软件均可以在 tools 目录下找到）；

第二部分是 OPL1000 M0 MCU 固件补丁和 Pack 脚本文件，保存在 FW_Binary 目录下。

智能秤项目特有部分主要由五类内容构成，包括

- 1) app
- 2) doc
- 3) prg_src
- 4) Readme.md
- 5) Release_Notes.md

app 文件夹包含手机端阿里云 app 和 ParamCfg.ini 配置文件，阿里云 app 主要用于 Scaler 事件的监控和管理，ParamCfg.ini 配置文件则用于部分头文件的更新，如云连接中所需三元组的配置更新；

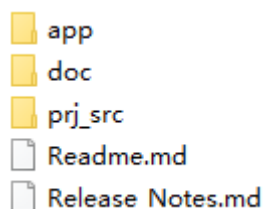
doc 目录下存放 应用指南文档，即本文档。

prg_src 文件夹包含 sensor 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，内容。

Release_Notes 描述本版本发布更新内容和注意事项。

Figure 1:项目文件



2.2. 工作原理

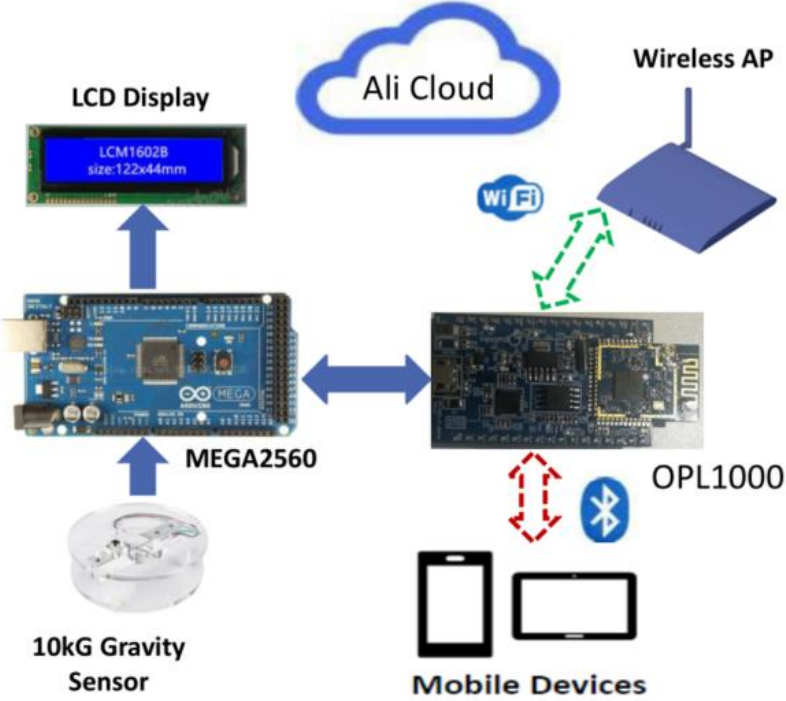
智能秤项目的主要部件有：物联网模块 OPL1000，移动设备（安装 2 个 APP），云端（阿里云）和 MEGA2560。如图 Figure 3 所示

旺凌科技的智能秤参考设计使用双芯片方案，其中 OPL1000 设备作为透传模块，负责和云端的连接；Arduino MEGA2560 作为外部 MCU，负责采集重力传感器数据，将数据发送给 OPL1000 设备，并显示重量数据和联网信息到液晶屏上。

称重之前 OPL1000 设备需要在移动设备 APP 控制下完成蓝牙配网操作。当有称重发生时，外部 MCU 将数据传递给 OPL1000，再通过 AP 发送到云端。LCD 上显示重量数据，手机 APP 上显示称重数据和时间。

用户按下压力传感器，模拟称重操作。Arduino MCU 采集到重量信息，显示到 LCD 上。该重量信息通过 AT 指令传递给 OPL1000 设备，OPL1000 将数据通过 MQTT 协议发送到阿里云。手机 APP 端同步显示重量数据和测量时间。

Figure 2:工作原理图



3. 运行 OPL 智能秤项目

运行 OPL1000 智能秤项目需要以下步骤：

- 更新工程配置文件，修改头文件中的宏定义参数（参考 3.1）。
- 使用编译工具完成项目工程编译。
- 通过 download tool 下载 opl1000.bin 文件到 opl1000 模块。
- 复位 opl1000 设备，打开 opulinks_iot_app 蓝牙配网 app。手机扫描 opl1000 蓝牙设备，绑定后扫描无线 AP，然后连接能够访问 Internet 的 AP。
- OPL1000 的固件连接云端，按需上传称重数据。此时打开 Opl_scaler 手机 APP 能够看到称重数据在不断更新。

3.1. 生成 OPL1000 设备固件

编译 Scaler 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。有两种方法可修改工作参数。

3.1.1. 使用 ParamCfg 工具生成和下载固件

使用 ParamCfg 工具可以一次性完成参数修改，固件生成和下载。包括修改工作参数，工程编译和固件下载。

ParamCfg 工具的具体使用请参考“通用参数配置工具使用指南 OPL1000-Parameter-Configure-Tool-guide.pdf”文档。本章节着重介绍关键参数的修改方法，以及如何执行工具完成修改、编译和下载操作。

关键参数包括 IOT 设备三元组，是否进入省电模式，设备名称等。如图 Figure 3

Figure 3: 阿里云三元组参数配置

```

[ali_product_key]
sub_dir = mqtt_example.h
typ = STRING
macro_name = PRODUCT_KEY
value = alqqoiDBbFT
length = 11
scope = 45
description = A macro to define ali cloud product key

[ali_device_name]
sub_dir = mqtt_example.h
typ = STRING
macro_name = DEVICE_NAME
value = sh_sensor02
length = 11
scope = 100
description = A macro to define ali cloud device name

[ali_device_secret]
sub_dir = mqtt_example.h
typ = STRING
macro_name = DEVICE_SECRET
value = QfnuQNldq0nrFr8NUy9zCI9dil55MIsU
length = 11
scope = 129
description = A macro to define ali cloud device secret

```

Figure 4:ParamCfg 更改三元组参数界面

```

===== General Parameter Config Tool v1.0 =====
smart_sleep_enable           : 1
ble_uuid_service             : 0xAAAA
ble_uuid_data_in             : 0xBBB0
ble_uuid_data_out            : 0xBBB1
ble_device_name_method       : 1
ble_device_name_prefix       : tmp_
ble_device_name_full         : tmpopl
ble_advertise_interval_min   : 0x640
ble_advertise_interval_max   : 0x640
wifi_connect_retry_times     : 10
wifi_connect_retry_idle      : 11
wifi_auto_connect_interval_init : 5000
wifi_auto_connect_interval_diff : 1000
wifi_auto_connect_interval_max : 30000
wifi_dtim_interval          : 30
ali_product_key              : a1qqoiDBbFT
ali_device_name              : sh_sensor01
ali_device_secret            : ZldAnUVK8MHVfJ2hXvkbLjf98IN8ndRr
===== End of General Parameter List =====

```

3.1.2. 使用 Keil C 和 Download 工具编译下载固件

使用 Keil C 手动更新参数配置需要分两步完成：

- 1 第一步使用 Keil C 开发工具打开头文件（blewifi_configuration.h, mqtt_example.h），修改 IOT 设备三元组，是否进入省电模式，设备名称等；
- 2 第二步使用 download 工具完成固件 Pack，下载操作，参考 “[2] Download 工具使用指南 OPL1000-patch-download-tool-user-guide.pdf”

3.2. OPL APP 完成蓝牙配网

首先确认需要连接设备的 MAC 地址，以及设备名称，设备名称信息可以参考 3.1 工程编译设备名称，在 APP 上选择正确的设备，然后点击 CONNECT。

Figure 5:手机 APP 蓝牙扫描列表



蓝牙设备绑定后，点击 Wifi Setup 扫描 AP，连接特定的 AP。

Figure 6:手机 APP 端扫描 AP 界面



3.3. 检查 OPL 设备工作状态

OPL1000 设备是否连接到阿里云可以有两种方式检查。

- 1 可以通过 OPL1000 设备的 UART 串口打印 log 信息来确认。出现下图信息表明阿里云连接成功。

Figure 7:阿里云连接串口 log 状态信息

```
[inf] HAL_TCP_Establish(104): success to establish tcp, fd=0
[inf] iotx_mc_connect(2502): mqtt connect success!
```

- 2 通过阿里云官网物联网接入模块中的“设备查看”检查设备在线状态。找到已连接的设备对应名称，当设备信息出现在线状态时，则判断阿里云连接成功

Figure 8:阿里云官网设备在线状态信息



3.4. Ali APP 监测体重

下图是 OPL1000 物联网智能称监控界面，它主要由六部分构成：导航栏，旺凌科技 LOGO，用户信息，重量数字显示，体重仪表显示，和电量显示。

本参考设计采用实时刷新机制显示体重，当有体重变化时，数字显示及仪表显示会根据体重变化实时刷新，仪表指针及仪表内数字显示会同时刷新。

Ali APP 设计过程参见第六章介绍。

Figure 9:Ali APP 体重检测界面



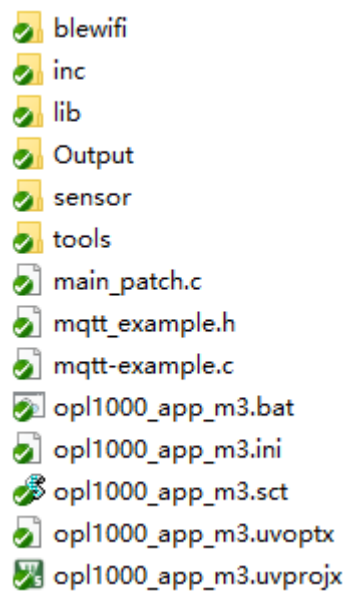
4. OPL 体重计应用设计

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

如 Figure 10 所示，scaler 项目包含蓝牙配网，称重计，MQTT 处理和库文件等目录。

Figure 10: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: Sensor 项目文件夹和内容

文件夹和文件	内容说明
belwifi	存放蓝牙配网功能相关的.c 及.h 文件
inc	存放调用 lib 库 API 函数对应的 include 头文件，包含 OPL WIFI，蓝牙协议栈头文件，第三方如 Lwip，Ali 云 SDK 库头文件。
lib	包含 OPL1000 最基本的库文件，一般不作更改
Output	主要存放编译时产生的相关文件其中包括编译成功后的 opl1000_app_m3.bin 文件

Scaler	此文件夹主要存放对温度智能秤的数据处理，保存等操作的.h 及.c 文件和 Arduino 的源文件，用户可以根据不同需求更改智能秤文件
Tools	存放生成 Bin 文件的转换工具，无需修改
Main_patch.c	Main_patch 是主文件
mqtt_example.c	MQTT 功能调用，完成与阿里云的连接和数据传输。其中阿里云三元组参数存放在 mqtt_example.h 文件中。用户可以根据自己创建的设备信息进行更改
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。如果需要调整内存空间，需要修改 opl1000_app_m3.sct 文件。

4.2. 功能模块说明

本节主要介绍 OPL1000 固件和 MEGA2560 的处理流程,及相关模块的介绍。关于 Ali 云端的配置和 APK 设计，请参考第五和第六章。

4.2.1. OPL1000 固件

OPL1000 设备作为透传模块，负责和云端的连接。它的功能如下：

- 与 OPL1000 通信完成蓝牙配网功能；
- 通过 AP 与 Ali 云端连接,并建立 MQTT 连接；
- 通过 UART 串口的 AT 命令获取称重信息，构建 MQTT 消息发布到云端。

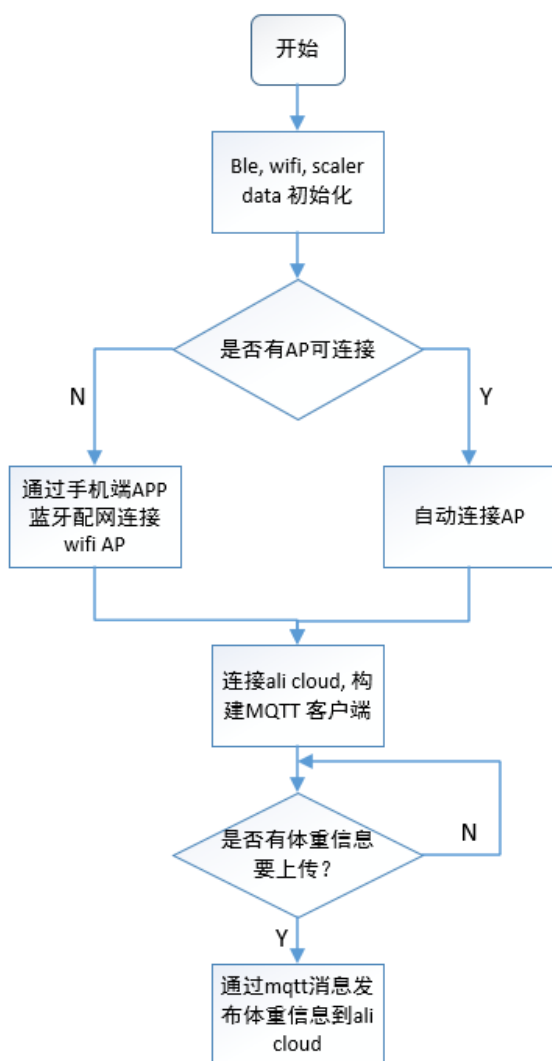
OPL1000 固件提供四个 at 命令。MEGA2560 通过这些命令和 OPL1000 交互数据和状态。四个 at 命令如下：

- at+inqap : 查询当前 OPL1000 是否已连接 ap。若是，返回 ‘ok’ ,否则返回 ‘error’ ；

- at+data : 发送称重数据到 OPL1000 。若是，返回 'ok' ,否则返回 'error' 。如，
at+data=1234 ;
- at+inqcloud: 查询 OPL1000 是否已将当前称重数据传送到 Ali 云。若是，返回 'ok' ,否则返回 'error' ;
- at+deepslp : 设置 OPL1000 进入 deep sleep 模式，待实现。

OPL1000 固件的主程序执行流程如 Figure 11 所示。在完成设备和智能秤初始化操作后，设备将自动尝试连接阿里云。如果连接成功并且有称重数据，则会构建 mqtt 消息上报给 Ali 云端。

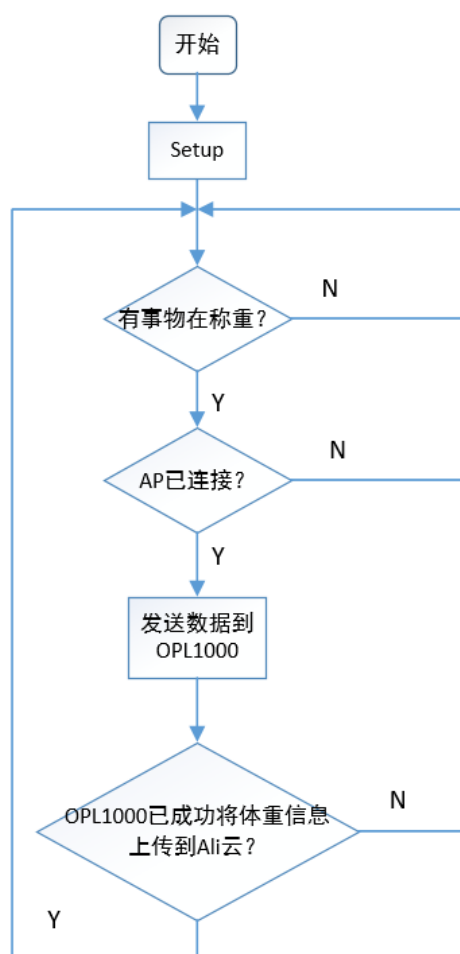
Figure 11:固件执行流程图



4.2.2. MEGA2560

Arduino MEGA2560 作为外部 MCU，负责采集重力传感器数据，将数据发送给 OPL1000 设备，并显示重量数据和联网信息到液晶屏上。主程序执行流程如 Figure 11 所示

Figure 12:固件执行流程图



4.2.3. 云连接和数据传输

OPL1000 与阿里云通过 TCP 协议连接，数据传输则采用的是 MQTT(v3.1)传输协议。

MQTT 协议工作原理如 Figure 12 所示。

Figure 12:MQTT 实现方式图



MQTT 协议中有三种身份：发布者（Publish）、代理（Broker）（服务器）、订阅者（Subscribe）。其中，消息的发布者和订阅者都是客户端，消息代理是服务器即阿里云，消息发布者可以同时是订阅者。

MQTT 传输的消息分为：主题（Topic）和负载（payload）两部分

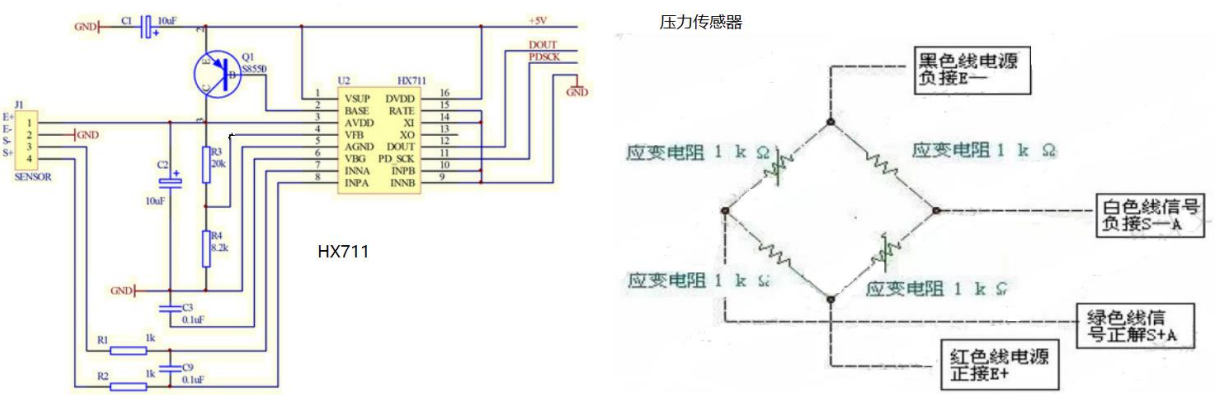
Topic，可以理解为消息的类型，订阅者订阅（Subscribe）后，就会收到该主题的消息内容（payload）

MQTT 会构建底层网络传输：它将建立客户端到服务器的连接，提供两者之间的一个有序的、无损的、基于字节流的双向传输，当应用数据通过 MQTT 网络发送时，MQTT 会把与之相关的服务质量（QoS）和主题名（Topic）相关连。

4.2.4. 称重模块

智能秤参考设计中的智能秤由 HX711 模块 加小量程称重压力传感器实现。其硬件电路图如 Figure 13 所示。

Figure 13:智能秤电路图



在本参考设计中将智能秤接入 Arduino MEGA2560 的四个引脚如下：

PWM D8 -> SCK

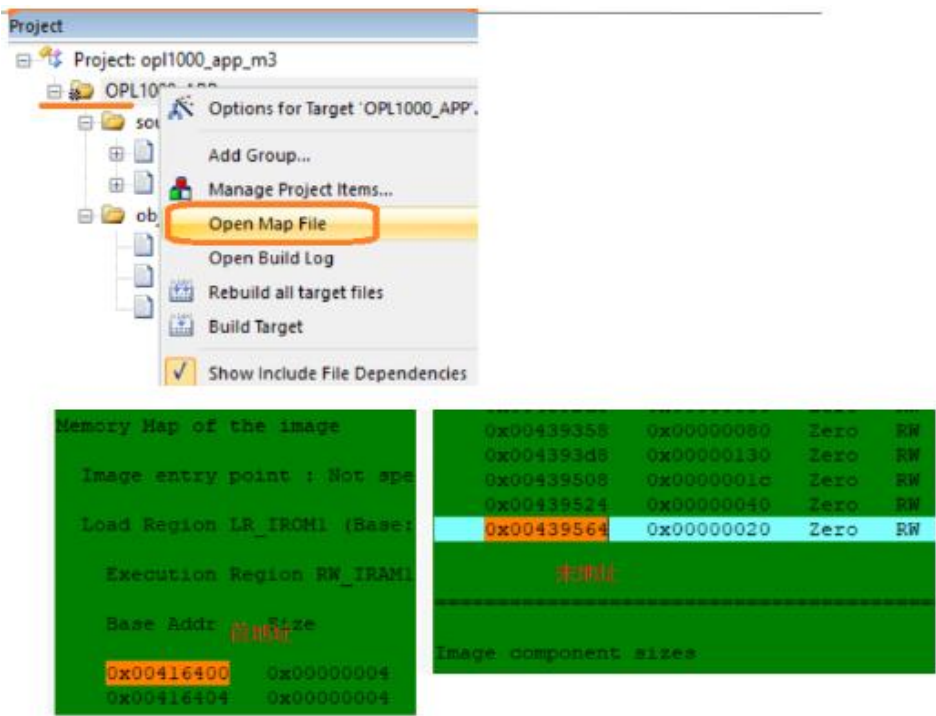
PWM D87 -> DT
VCC(5V) -> VCC(5V)
GND -> GND

4.2.5. 工程整体内存分配

此章节主要针对工程编译、下载以及运行内存不足时，作详细说明
内存分配主要有以下 4 个步骤：

- 1. 在 Memory Map 中获取首地址和末地址
 - 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。如图 Figure 14 所示。
 - 对末地址以 4k 为单位向上取整，为 0x0043a000.同时为 stack 留适量的空间

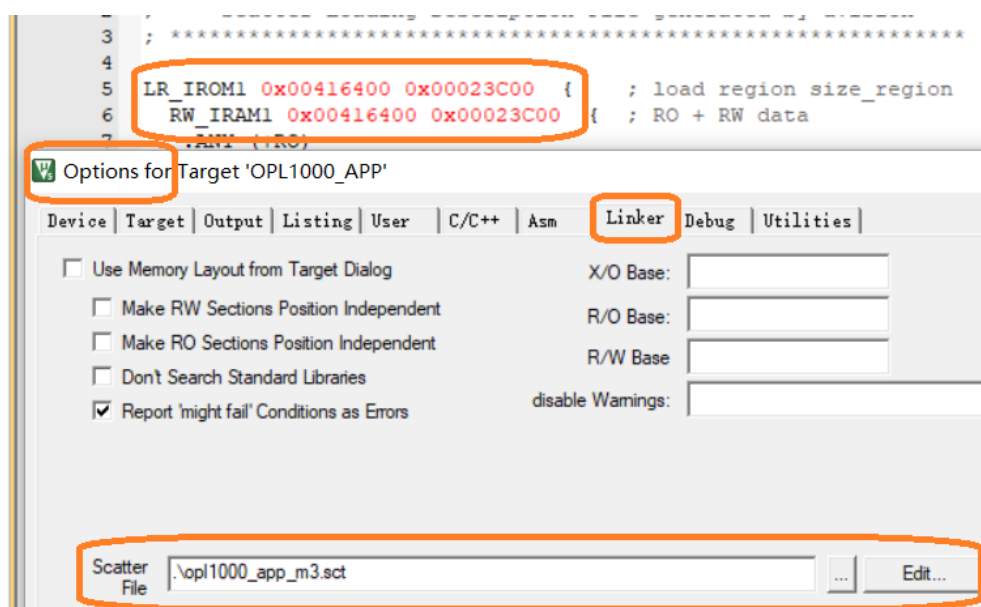
Figure 14:获取首末地址



- 2. 配置 scatter file
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,如图 Figure 15 所示
- 根据首末地址设置 LR_IROM1 region 的起始地址和大小,其中起始地址就是首地址,无需改动,而大小则可以改为末地址 - 首地址的值,如在该例中,大小 = $0x0043a000 - 0x00416400 = 0x00023C00$ 。
- 修改 RW_IRAM1 section 的内容,使它跟 region 的信息保持一致

Figure 15:修改 scatter file



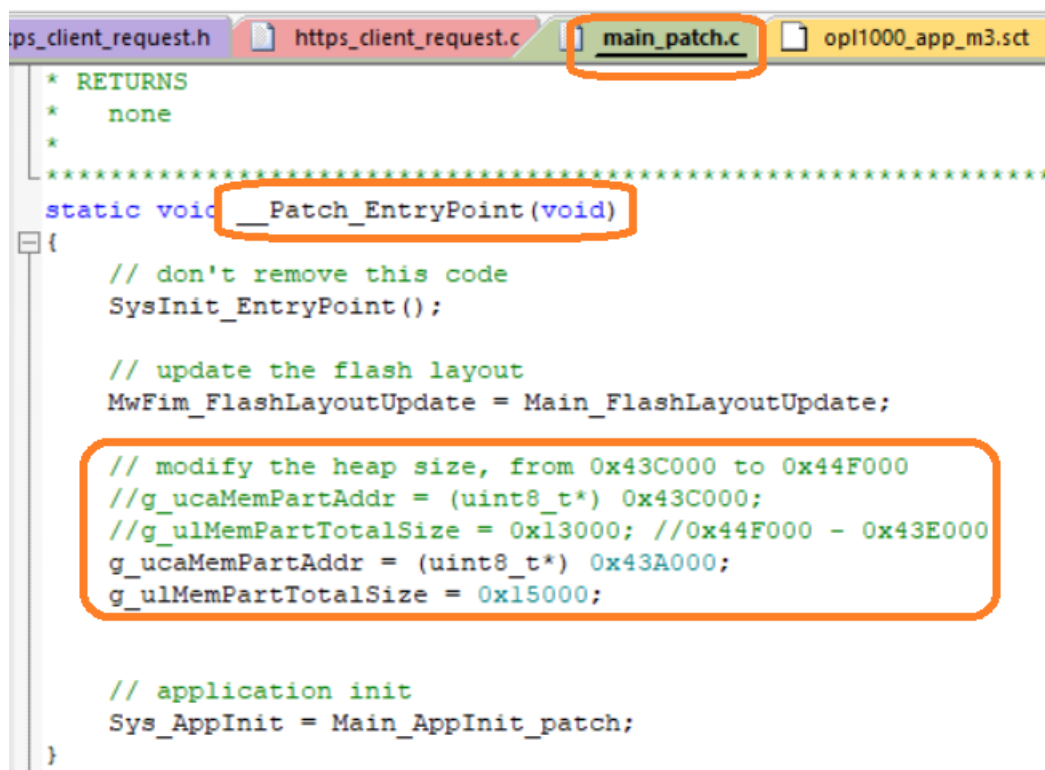
3. 配置 heap size

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,如图 Figure 16 所示。

- 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 = $0x44F000 - \text{末地址}$,对于有图中的例子,就是,大小 = $0x44F000 - 0x0043a000 = 0x15000$

Figure 16:调整 HeapSize



```

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;

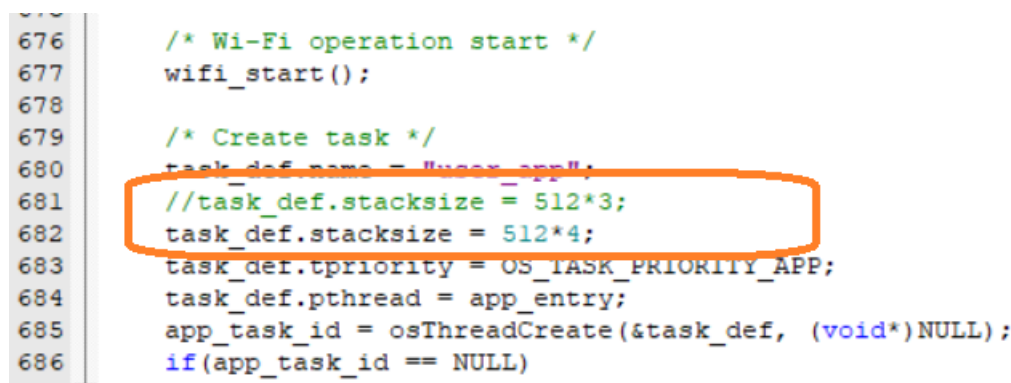
    // application init
    Sys_AppInit = Main_AppInit_patch;
}
    
```

4. 配置 stack size

按需设置 size 的大小，方法如下：

- Stack size 的设置通常都放在工程的 AppInit() 中进行，如图 Figure 17 所示。

Figure 17:调整 Stack Size



```

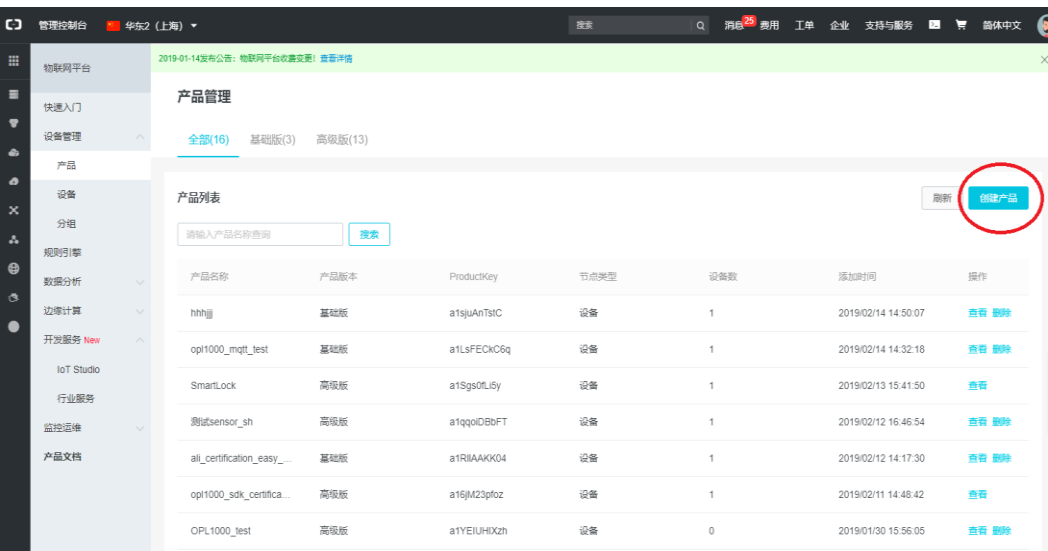
676     /* Wi-Fi operation start */
677     wifi_start();
678
679     /* Create task */
680     task_def.name = "lucy_app";
681     //task_def.stacksize = 512*3;
682     task_def.stacksize = 512*4;
683     task_def.tpriority = OS_TASK_PRIORITY_APP;
684     task_def.pthread = app_entry;
685     app_task_id = osThreadCreate(&task_def, (void*)NULL);
686     if(app_task_id == NULL)
    
```

5. ALI 云 IOT 设备创建

5.1. 创建 IOT 设备

产品创建是指创建同类设备，例如称重计，可以在这类产品中添加多个设备。下面以温度创建为例说明创建过程。更多功能请参考阿里云官方文档。

Figure 18: 阿里云 APP 创建界面



创建温度计设备过程：

1. 选择创建产品
2. 在弹出的对话框中，选择高级版，点击下一步
3. 填写产品名称，选在所属分类
这里选择智能生活/个护健康/体温计，其他默认选择，对于产品描述，用户可以根据需要自行填写，然后点击完成
4. 点击我们创建的产品操作下的查看，选择功能定义，根据需要添加相应功能及事件管理

5.2. IOT 设备三元组

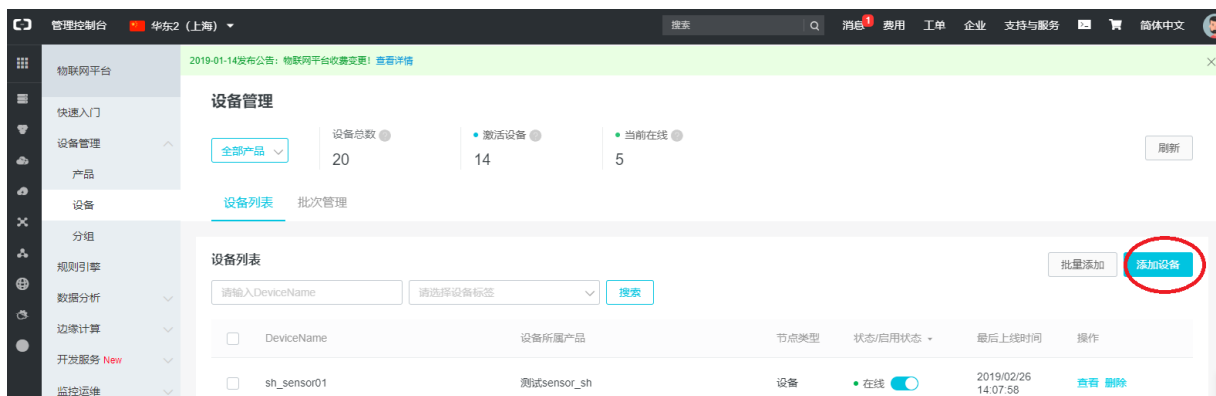
设备的添加将会是 OPL1000 连接阿里云的唯一标识，此时添加设备后会生成连接阿里云的三元组，包括：ProductKey, DeviceName, DeviceSecret，操作如下：

- ◆ 点击添加设备，在弹出的对话框下面产品栏选择 5.1 创建的产品，给需要添加的设备添加名称

点击下一步

◆ 此时会弹出三元组信息，记录相关信息，OPL1000 连接云端会使用，需要加入到头文件。

Figure 19: 阿里云设备三元组创建



6. ALI 云 APP 生成

Ali APP 生成主要在使用开发服务套件下面的 IOT Studio。主要包含以下步骤：

1. 新建项目，输入项目名称，根据需要输入描述内容，点击确定
2. 查看创建的项目
3. 点击移动应用

移动应用 ?

2

4. 点击新增可视化应用

新增可视化应用

5. 输入应用名称，点击完成

此时 APP 项目框架创建完毕，可以进入相应项目完成界面设计

6.1. 界面设计

在 APP 设计页面栏目下有一默认页面，用户可以根据需要点击页面右侧+，添加页面数量，此页面为 OPL1000 检测控制页面，即操作页面

为了页面整体布局的美观，用户可以以容器方式存放相关组件，然后将需要的组件拖拽到容器里面，使用容器的目的是为了组件能够摆放到我们想要的位置。

Figure 20:阿里云 APP 界面设计



6.2. 账号配置及手机 APP 设置

用户可以将自选的 APP 图标在设置界面上传并配置 APP 显示图标。在账号界面，用户可以手动添加能够登录 APP 的账号、用户名称及用户密码。

6.3. APP 生成和安装

Android APP 构建比较简单，点击页面右上角的构建，选择 Android 构建即可。如果仅仅自己使用不发布到应用市场，则选择“我只想自己使用”，输入安卓包文件名，或者随机生成，点击下一步，等待 APP 构建完成。有两种方式安装 APK：扫码安装或者下载到手机安装。

Figure 21:生成 Android APP



CONTACT

sales@Opulinks.com