

# OPL1000

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

## Sensor Reference Design Application Guide



OPULINKS

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

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

---

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

Date	Version	Contents Updated
2019-02-27	0.1	<ul style="list-style-type: none"><li>Initial Release</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	3
2.1. 项目构成	3
2.2. 工作原理	4
3. 运行 OPL 温度计应用	5
3.1. 生成 OPL1000 设备固件	5
3.1.1. 使用 ParamCfg 工具生成和下载固件	5
3.1.2. 使用 Keil C 和 Download 工具编译下载固件	7
3.2. OPL APP 完成蓝牙配网	7
3.3. 检查 OPL 设备工作状态	8
3.4. Ali APP 监测温度	9
4. OPL 温度计应用设计	10
4.1. 项目工程构成	10
4.2. 执行流程和模块说明	11
4.2.1. 执行流程	11
4.2.2. 主要 Task Handler	12
4.2.3. 云连接和数据传输	13
4.2.4. 温度计算	13
4.2.5. 工程整体内存分配	15
5. Ali 云 IOT 设备创建	18
5.1. 创建 IOT 设备	18
5.2. IOT 设备三元组	18
6. Ali 云 APP 生成	20
6.1. 界面设计	20
6.2. 账号配置及手机 APP 设置	21
6.3. APP 生成和安装	21

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 .....5

Figure 4:ParamCfg 更改三元组参数界面 .....6

Figure 5:手机 APP 蓝牙扫描列表.....7

Figure 6:手机 APP 端扫描 AP 界面 .....7

Figure 7:阿里云连接串口 log 状态信息 .....8

Figure 8:阿里云官网设备在线状态信息 .....8

Figure 9:Ali APP 温度检测界面 .....9

Figure 10:工程文件构成 ..... 10

Figure 11:固件执行流程图..... 11

Figure 12:MQTT 实现方式图..... 13

Figure 13:温度传感器电路图 ..... 13

Figure 14:获取首末地址 ..... 15

Figure 15:修改 scatter file..... 16

Figure 16:调整 HeapSize ..... 17

Figure 17:调整 Stack Size ..... 17

Figure 18:阿里云 APP 创建界面 ..... 18

Figure 19:阿里云设备三元组创建..... 19

Figure 20:阿里云 APP 界面设计 ..... 20

Figure 21:生成 Android APP ..... 21

LIST OF TABLES

Table 1: Sensor 项目文件夹和内容..... 10

# 1. 介绍

## 1.1. 文档应用范围

本文档介绍如何基于 OPL1000 开发传感器类型完整应用。内容包括固件设计，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](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

传感器项目主要由两部分组成：一是共有资料，二是传感器项目特有部分。

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。

传感器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 sensor 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；

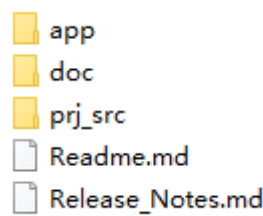
doc 目录下存放 应用指南文档，即本文档。

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



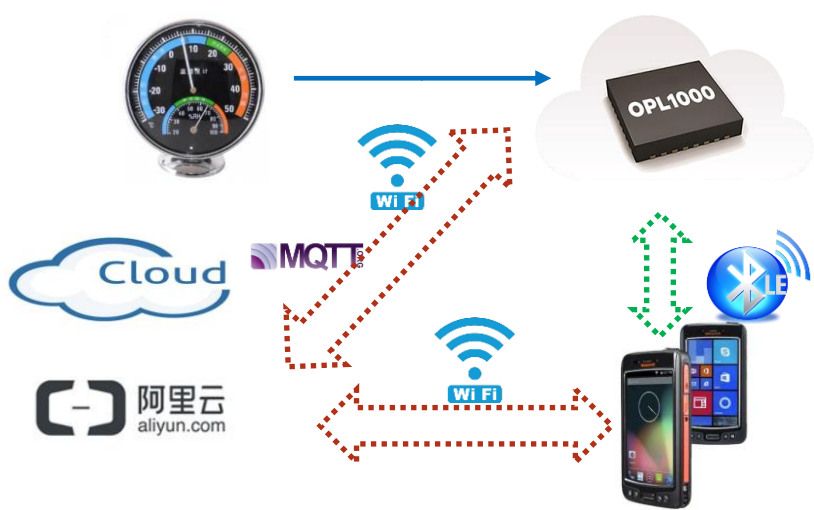
2.2. 工作原理

传感器项目主要部件：物联网模块 OPL1000，移动设备（安装 2 个 APP），云端（阿里云）和温度传感器。

它的工作过程为：

- 1. 温度传感器测量得到温度数据，OPL1000 查询传感器得到到原始信息数据，经过标定和补偿后得到温度信息数据。
- 2. 手机 APP 和 OPL1000 设备通信完成蓝牙配网操作，连接到无线 AP。
- 3. OPL1000 将温度信息打包成 MQTT 协议数据包，通过 AP 上报到阿里云。
- 4. 阿里云接收到温度信息后，实施更新下载到手机端。手机上的 APP 会根据温度变化实时的刷新温度数据。

Figure 2:工作原理图



## 3. 运行 OPL 温度计应用

运行 OPL1000 温度计应用需要以下步骤：

- 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）。
- b) 使用编译工具完成项目工程编译。
- c) 通过 download tool 下载 opl1000.bin 文件到 opl1000 模块。
- d) 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。
- e) OPL1000 的固件连接云端，周期性上报温度数据。此时打开 Opl\_sensor 手机 APP 能够看到温度数据在不断更新。

### 3.1. 生成 OPL1000 设备固件

编译 Sensor 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。找到自己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



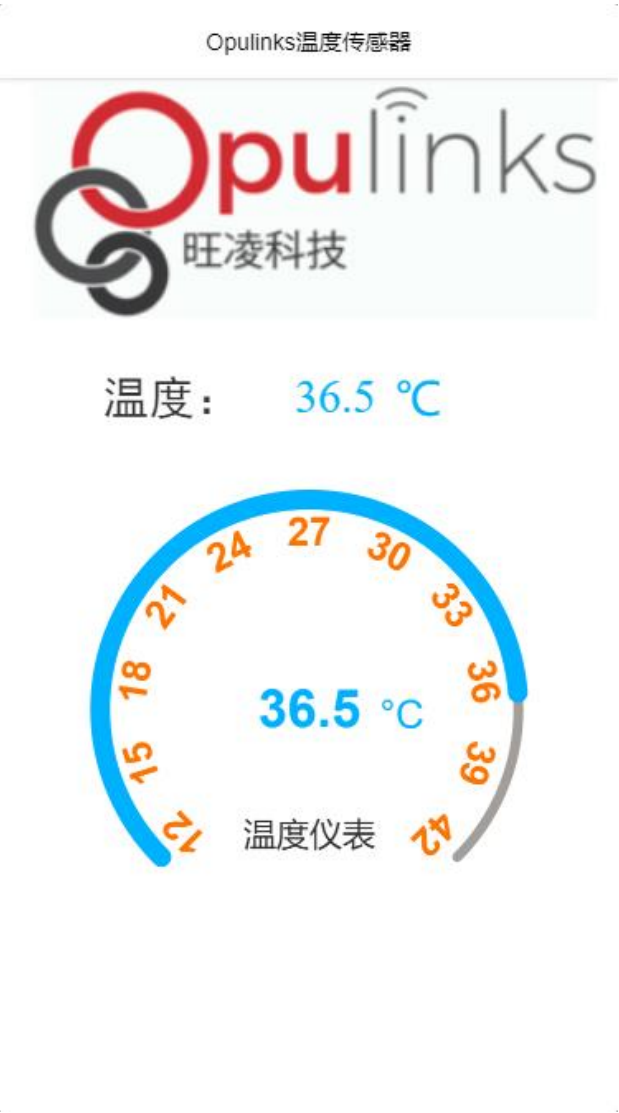
### 3.4. Ali APP 监测温度

下图是 Ali APP 温度监控界面，Ali APP 主要由四部分构成：导航栏，旺凌科技 LOGO，温度数字显示，温度仪表显示。

本参考设计采用实时刷新机制显示温度，当有温度变化时，数字显示及仪表显示会根据温度变化实时刷新，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 温度检测界面



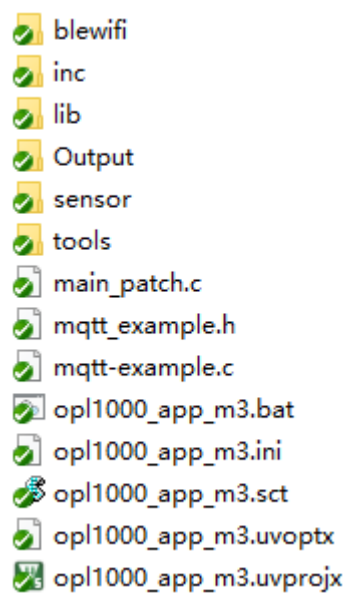
## 4. OPL 温度计应用设计

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，sensor 项目包含蓝牙配网，传感器，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

Sensor	此文件夹主要存放对温度传感器的数据处理，保存等操作的.h 及.c 文件，用户可以根据不同需求更改传感器文件
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. 执行流程和模块说明

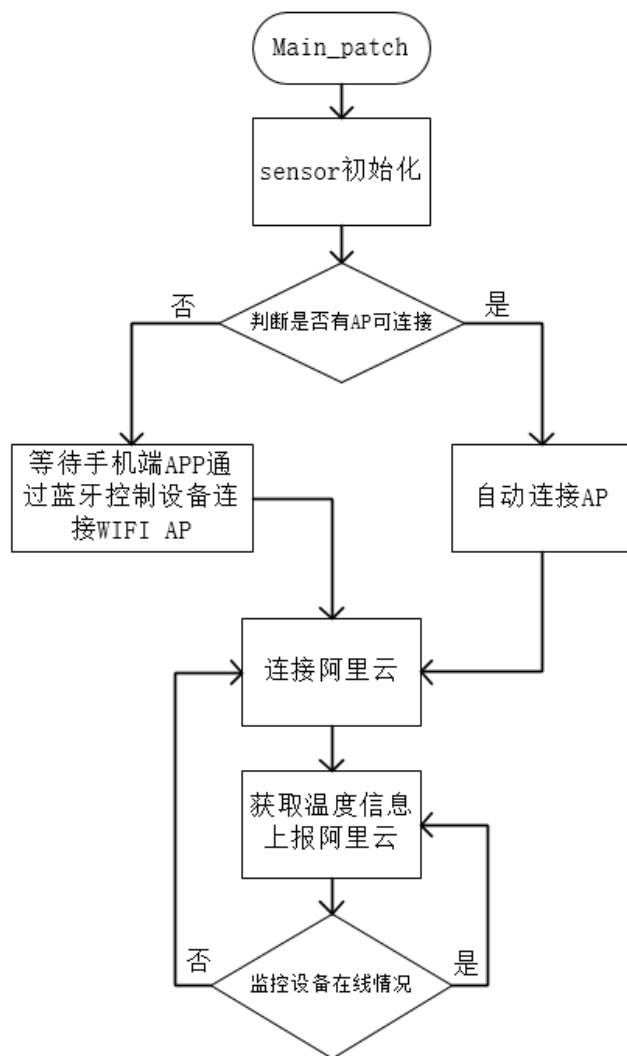
本章节介绍 OPL1000 固件处理流程。

4.2.1. 执行流程

主程序执行流程如 Figure 11 所示。在完成设备和传感器初始化操作后，设备将自动尝试连接阿里云。如果连接成功，则周期性测量温度信息，上报给云端。

Figure 11:固件执行流程图





#### 4.2.2. 主要 Task Handler

本工程项目主要用到几个特别的 Handler。

##### 1. BLE Handler

BLE Handle 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

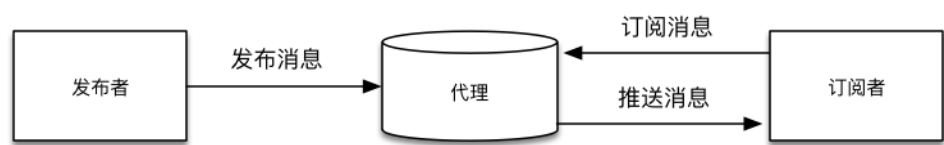
WIFI Handler 是 OPL1000 与 AP 建立连接后，连线及断线检查，断线后重连功能

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

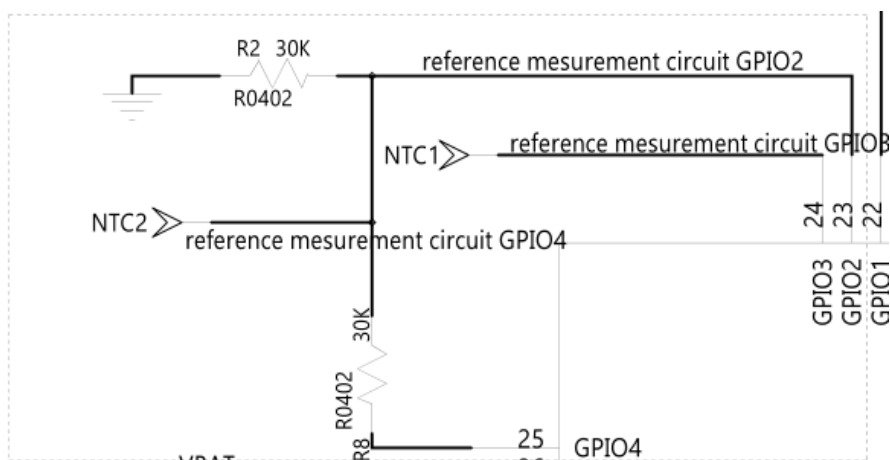
Topic，可以理解为消息的类型，订阅者订阅（Subscribe）后，就会收到该主题的消息内容（payload）

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. 温度计算

温度传感器的硬件电路图如 Figure 13 所示。其工作原理为：当外界温度变化时，传感器的阻值变化，引起分压电路电压变化。OPL1000 通过 AUX ADC 读入电压的模拟值，经过标定补偿后计算出阻值大小，然后采用区间差分法计算出相应温度。

Figure 13:温度传感器电路图



在本参考设计中将传感器接入 OPL1000 的三个引脚，GPIO2/3/4。GPIO2 为电压输入，GPIO3 和 GPIO4 接传感器两端信号。

计算过程分为 3 个步骤：

1 阻值校验，更改引脚三种不同的状态获取初始电压值。设置三个管脚工作模式为：

GPIO2 -> PIN\_TYPE\_AUX\_2，  
 GPIO3 -> PIN\_TYPE\_NONE，  
 GPIO4 -> PIN\_TYPE\_GPIO\_OUTPUT\_LOW  
 得到电压 U1。

2 设定 GPIO4 属性为高电平输出

GPIO2 -> PIN\_TYPE\_AUX\_2，  
 GPIO3 -> PIN\_TYPE\_NONE，  
 GPIO4 -> PIN\_TYPE\_GPIO\_OUTPUT\_HIGH  
 得到此时的电压 U2。

3 再次设定 GPIO4 属性为高电平输出

GPIO2 -> PIN\_TYPE\_AUX\_2，  
 GPIO3 -> PIN\_TYPE\_NONE，  
 GPIO4 -> PIN\_TYPE\_GPIO\_OUTPUT\_HIGH  
 得到此时的电压 U3。

按照如下公式计算真实的温度阻值： $Real = ((U2 - U1) * 30 * 2) / (U3 - U1) - 30$

由于  $offset = real - compute$  ,  $offset = new\ base - old\ base$  ,  $new\ base = old\ base + (real - compute)$  , 从而可以计算新的温度数值

$$New\ base = ((U2-U1) * 30 * 2) / (U3-U1) - 30 - compute + old\ base$$

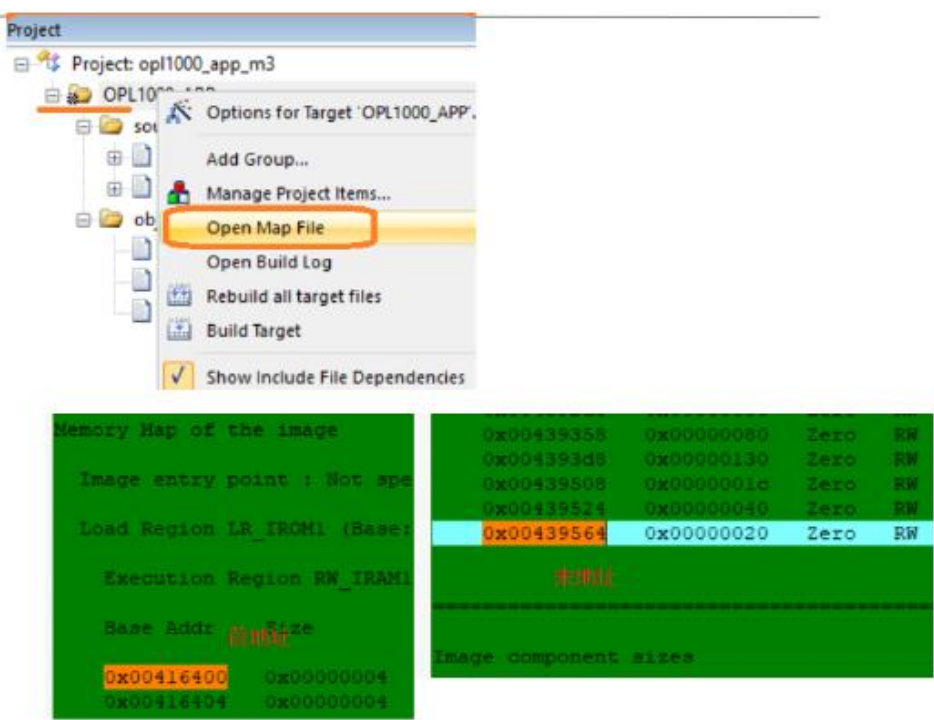
Old base 代表设置的基数电压 , compute 代表实时获取的数值。

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

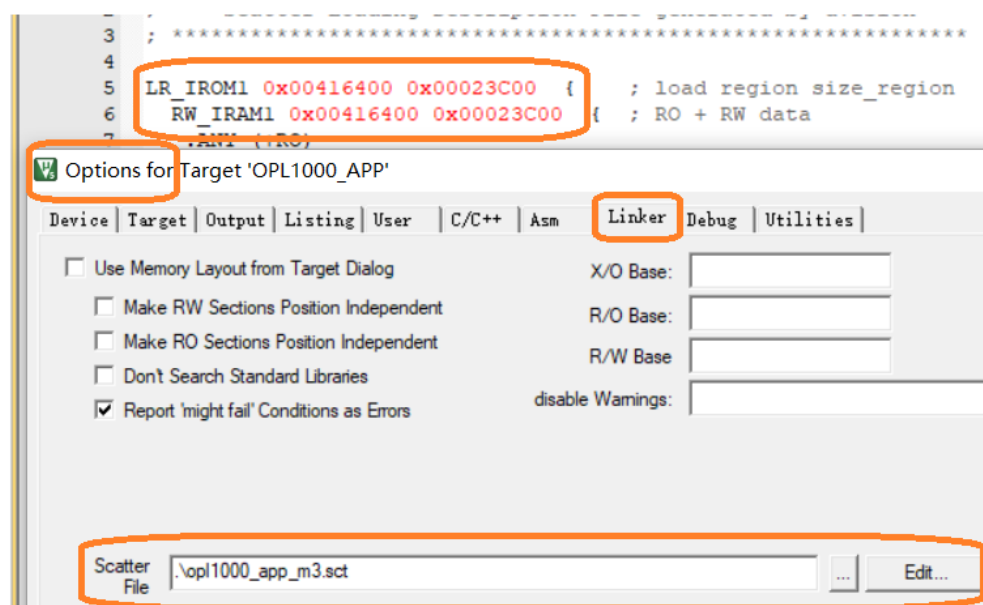


## 2. 配置 scatter file

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

```

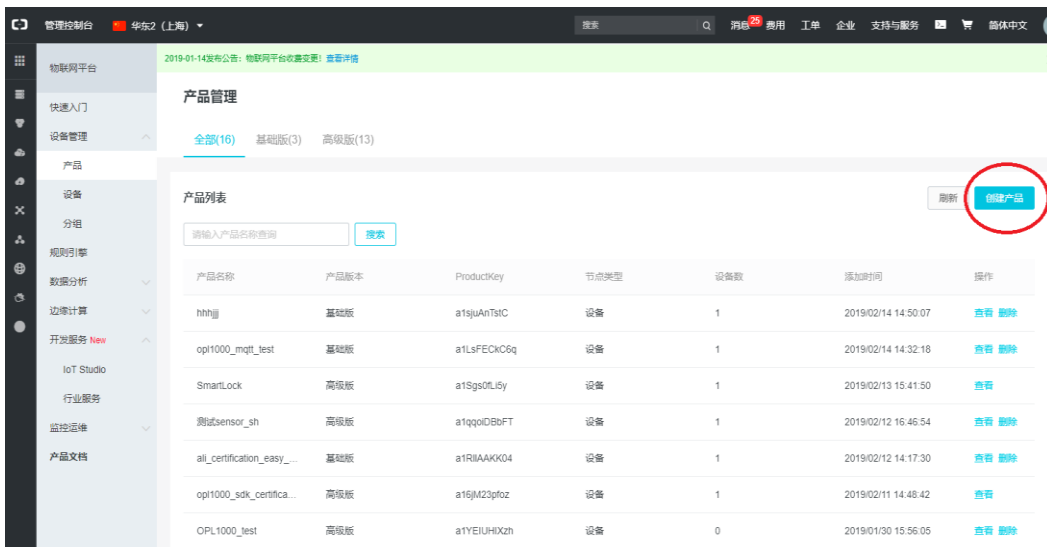
676     /* Wi-Fi operation start */
677     wifi_start();
678
679     /* Create task */
680     task_def.name = "lucy_app";
681     //task_def.stacksize = 512*3;
682     task_def.stacksize = 512*4;
683     task_def.tpriority = OS_TASK_PRIORITY_APP;
684     task_def.pthread = app_entry;
685     app_task_id = 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，例如温度计，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](mailto:sales@Opulinks.com)