

OPL1000

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

Smart Lighting Reference Design Application Guide



OPULINKS

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

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

OPL1000-Reference-smart-light-application-guide-R01 | Version V01

Date	Version	Contents Updated
2019-02-27	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 SMART Lighting 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	8
3.2. OPL APP 完成蓝牙配网	8
3.3. 检查 OPL 设备工作状态	10
3.4. Ali Smart Lighting APP 管理	10
4. OPL Smart Lighting 应用设计	13
4.1. 项目工程构成	13
4.2. 执行流程和模块说明	14
4.2.1. 执行流程	14
4.2.2. 主要 Task Handler	15
4.2.3. 云连接和数据传输	15
5. Ali 云 IOT 设备创建	16
5.1. 创建 IOT 设备	16
5.2. IOT 设备三元组	16
6. Ali 云 APP 生成	18
6.1. 界面设计	18
6.2. 账号配置及手机 APP 设置	19
6.3. APP 生成和安装	19

LIST OF FIGURES

Figure 1:项目文件.....4

Figure 2:工作原理图.....4

Figure 3: IOT 设备三元组.....6

Figure 4 : 燈具 profile.....7

Figure 5: 軟件 Pack 操作图.....8

Figure 6:軟件 Download 操作图.....8

Figure 7:手机 APP 蓝牙扫描列表.....9

Figure 8:手机 APP 端扫描 AP 界面.....9

Figure 9: 阿里云连接串口 log 状态信息..... 10

Figure 10: 阿里云官网设备在线状态信息 10

Figure 11: 燈具之各燈泡開關界面..... 11

Figure 12: 燈泡亮度顯示界面 12

Figure 13:工程文件构成 13

Figure 14:固件执行流程图..... 14

Figure 15:MQTT 实现方式图..... 15

Figure 16:阿里云 APP 创建界面 16

Figure 17:阿里云设备三元组创建..... 17

Figure 18:阿里云 APP 界面设计 19

Figure 19:生成 Android APP 19

LIST OF TABLES

Table 1: Smart Lighting 项目文件夹和内容 13

1. 介绍

1.1. 文档应用范围

本文档介绍如何基于 OPL1000 开发智能照明类型完整应用。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

访问链接：<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

Copyright © 2017-2019, Opulinks. All Rights Reserved.
OPL1000-Reference-smart-light-application-guide-R01, Version V01

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

传感器项目主要由两部分组成：一是共有资料，二是智能照明项目特有部分。

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。

智能照明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

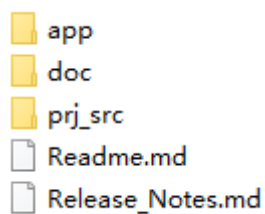
app 文件夹包含手机端阿里云 app 和 ParamCfg.ini 配置文件，阿里云 app 主要用于 smart lighting 事件的监控和管理，ParamCfg.ini 配置文件则用于部分头文件的更新，如云连接中所需三元组的配置更新；

doc 目录下存放 应用指南文档，即本文档。

prg_src 文件夹包含 smart lighting 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。

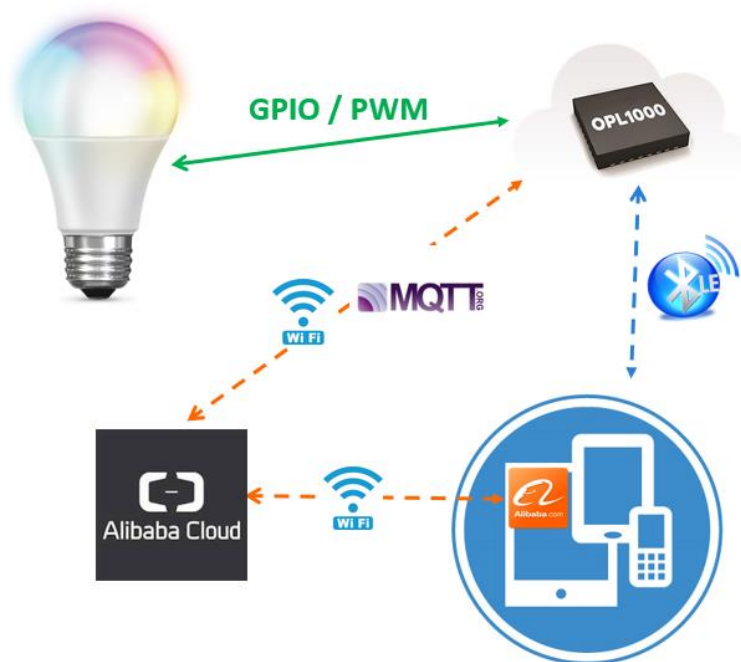
Figure 1:项目文件



2.2. 工作原理

Reference smart lighting 项目主要部件：物联网模块 OPL1000 · 移动设备（For install App）· 云端（阿里云）和 灯具固件。如详细介绍参考 chapter 4.2

Figure 2: 工作原理图



3. 运行 OPL SMART LIGHTING 应用

运行 OPL1000 智能照明应用需要以下步骤：

- 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。此时打开 Opl_Lighting 手机 APP 能够控制 Smart Lighting

3.1. 生成 OPL1000 设备固件

编译 Smart Lighting 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 (1) IOT 设备三元组 (2) 灯具 profile

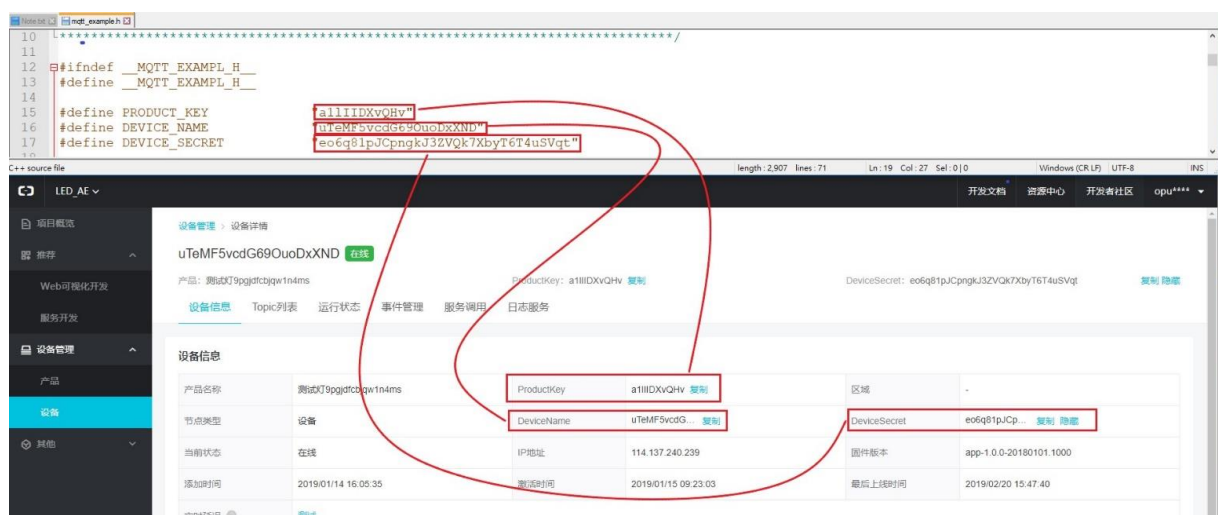
(1) 設置 IOT 设备三元组

File: prj_src\mqtt_example.h

修改下面三个参数:

1. Production Key
2. Device Name
3. Device Secret

Figure 3: IOT 设备三元组



(2) 灯具 profile

File: prj_src \mqtt_example.c

修改以下参数和 I/O 配置

- 1 Power switch
- 2 Brightness
- 3 I/O setting

Figure 4 : 灯具 profile

The figure illustrates the configuration of a light profile. It consists of two main parts: a code snippet and a web interface.

Code Snippet (mqtt_example.c):

```

26 #include "iot_export.h"
27 #include "hal_vic.h"
28 #include "hal_pwm.h"
29 #include "mqtt_example.h"
30
31 /* define light status */
32 #define NUMBER_OF_LIGHT 5
33 #define NUMBER_STATUS_OF_LIGHT 10
34 #define LENGTH_OF_LIGHT_STATUS 32
35 #define MAX_LIGHT_BRIGHTNESS 95
36 #define MIN_LIGHT_BRIGHTNESS 0
37 #define LIGHT_SWITCH_ON 1
38 #define LIGHT_SWITCH_OFF 0
39 #define DEFAULTE_PWM_FREQ 500
40
41 typedef struct
42 {
43     char Topic_of_Light_Switch[32]; // function defined in cloud server ex: RedLightSet
44     uint8_t Status_of_Light_Switch; // 0 : off 1: On
45     char Topic_of_Light_Brightness[32]; // function defined in cloud server ex: RedLightAdjustLevel
46     uint32_t Level_of_Light_Brightness; // 0-100
47     uint32_t IO_Control_Index; // ex : GPIO_IDX_23 or HAL_PWM_IDX_5
48 } T_Ref_Smart_Light_Profile;
49
50 T_Ref_Smart_Light_Profile Smart_Light_Profile_def[NUNBER_OF_LIGHT] =
51 {
52     {
53         "RedLightSet", LIGHT_SWITCH_OFF, "RedLightAdjustLevel", MIN_LIGHT_BRIGHTNESS, HAL_PWM_IDX_5,
54         "GreenLightSet", LIGHT_SWITCH_OFF, "GreenLightAdjustLevel", MIN_LIGHT_BRIGHTNESS, HAL_PWM_IDX_4,
55         "BlueLightSet", LIGHT_SWITCH_OFF, "BlueLightAdjustLevel", MIN_LIGHT_BRIGHTNESS, HAL_PWM_IDX_3,
56         "WhiteLightSet", LIGHT_SWITCH_OFF, "WhiteLightAdjustLevel", MIN_LIGHT_BRIGHTNESS, HAL_PWM_IDX_2,
57         "YellowLightSet", LIGHT_SWITCH_OFF, "YellowLightAdjustLevel", MIN_LIGHT_BRIGHTNESS, HAL_PWM_IDX_1,
58     }
59 };

```

Web Interface (LED_Alt):

The web interface shows a table of light profiles. The '自定义功能' (Custom Function) section is highlighted. The table lists various light profiles and their corresponding functions.

功能类型	功能名称	标识符	数据类型	数据定义
属性	所有亮度开关	LightSwitch	bool (布尔型)	布尔值: 关: 0; 开: 1;
事件	故障上报	Error	-	事件类型: 故障
自定义功能	自定义功能	自定义功能	自定义功能	自定义功能
属性	绿色亮度	GreenLightSet	bool (布尔型)	布尔值: 关: 0; 开: 1;
属性	蓝色亮度	BlueLightSet	bool (布尔型)	布尔值: 关: 0; 开: 1;
属性	白色亮度	WhiteLightSet	bool (布尔型)	布尔值: 关: 0; 开: 1;
属性	黄色亮度	YellowLightSet	bool (布尔型)	布尔值: 关: 0; 开: 1;
属性	红色亮度	RedLightAdjustLevel	int32 (整数型)	取值范围: 0 ~ 95
属性	绿色亮度	GreenLightAdjustLevel	int32 (整数型)	取值范围: 0 ~ 95
属性	蓝色亮度	BlueLightAdjustLevel	int32 (整数型)	取值范围: 0 ~ 95
属性	白色亮度	WhiteLightAdjustLevel	int32 (整数型)	取值范围: 0 ~ 95
属性	黄色亮度	YellowLightAdjustLevel	int32 (整数型)	取值范围: 0 ~ 95
属性	红色亮度	RedLightSet	bool (布尔型)	布尔值: 关: 0; 开: 1;

Red and green boxes and arrows indicate the mapping between the code and the UI:

- Red box: "RedLightSet", "GreenLightSet", "BlueLightSet", "WhiteLightSet", "YellowLightSet" in the code maps to the corresponding "LightSet" entries in the UI table.
- Green box: "RedLightAdjustLevel", "GreenLightAdjustLevel", "BlueLightAdjustLevel", "WhiteLightAdjustLevel", "YellowLightAdjustLevel" in the code maps to the corresponding "LightAdjustLevel" entries in the UI table.

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。

Figure 5: 软件 Pack 操作图

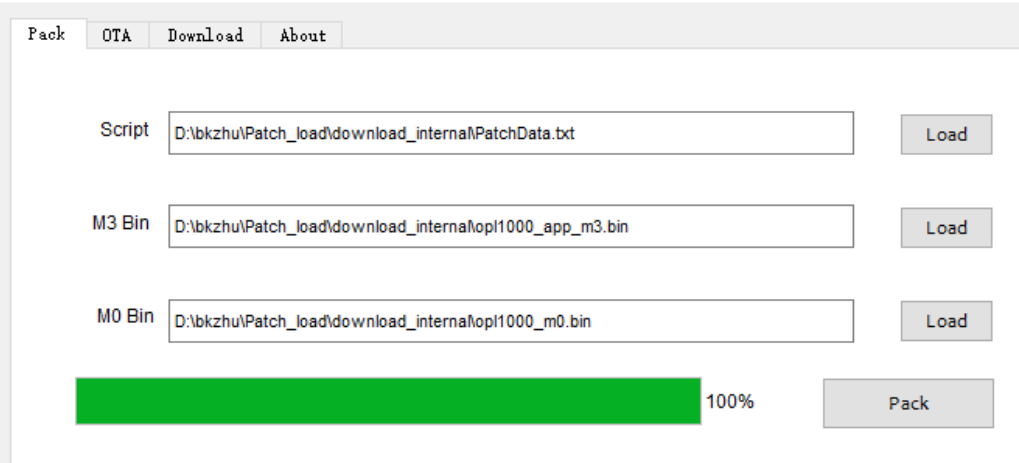
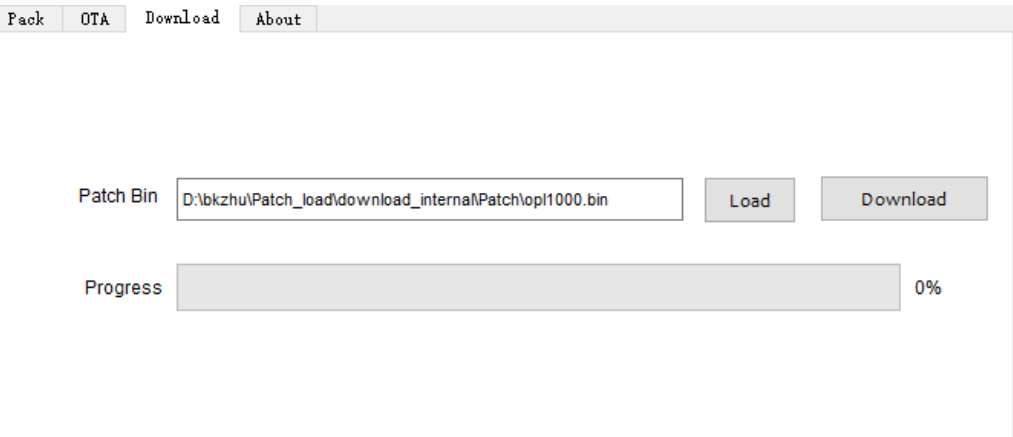


Figure 6: 软件 Download 操作图



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 7: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 8: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 9: 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 10: 阿里云官网设备在线状态信息

设备列表		设备管理					新增
设备列表		请输入 DeviceName					
		搜索					
<input type="checkbox"/>	DeviceName	产品	节点类型	状态	最后上线时间	操作	
<input type="checkbox"/>	uTeMF5vcdG89OuoDixND	奥松灯logidctjqv1n4ms	设备	在线	2019-02-27 16:47:53	激活凭证 查看 删除	

3.4. Ali Smart Lighting APP 管理

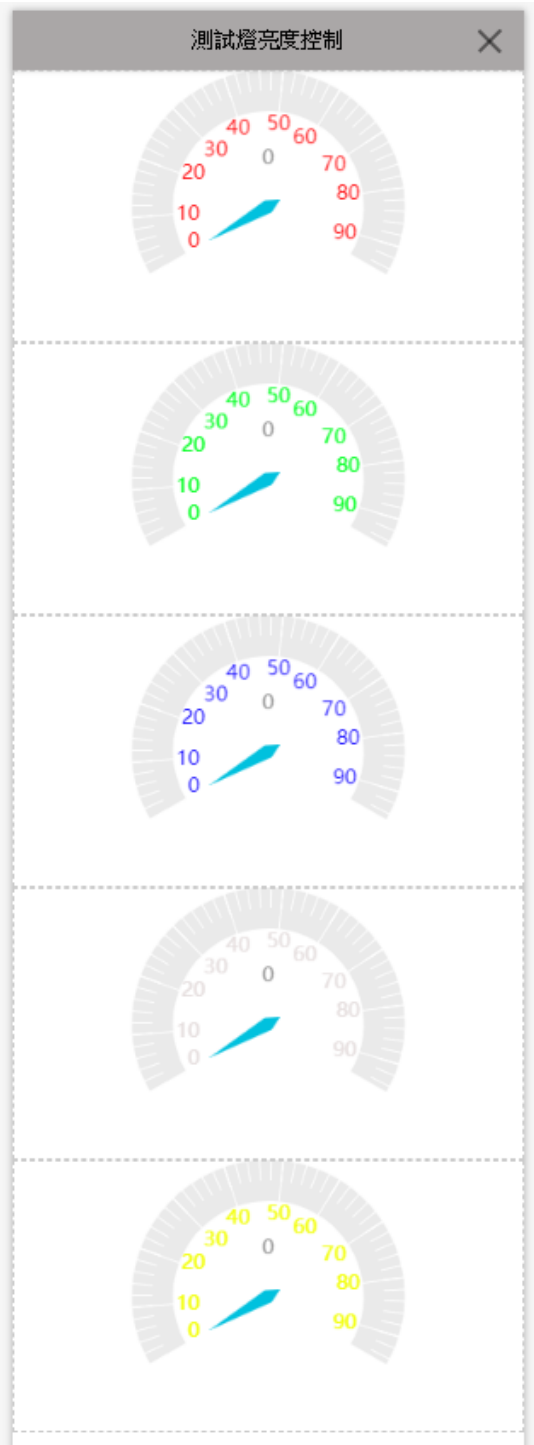
下图是 Ali APP 灯具控制监控界面，Ali APP 主要由四部分构成：(1)导航栏，(2)旺凌科技 LOGO，(3) 灯具之各灯泡开关，(4)各灯泡亮度显示。

Ali APP 设计过程参见第六章介绍。

Figure 11: 燈具之各燈泡開關界面



Figure 12: 燈泡亮度顯示界面



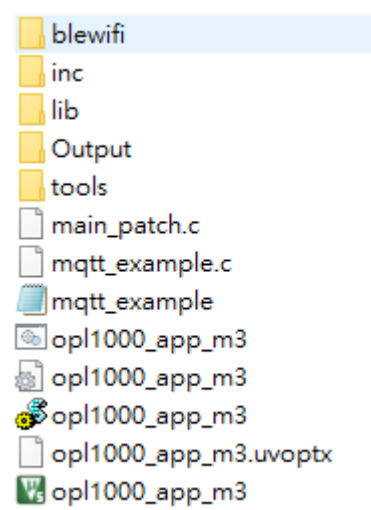
4. OPL SMART LIGHTING 应用设计

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

如错误!未找到引用源。所示，smart lighting 项目包含蓝牙配网，智能照明，MQTT 处理和库文件等目录。

Figure 13: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: Smart Lighting 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
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	燈具 profile 及 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。

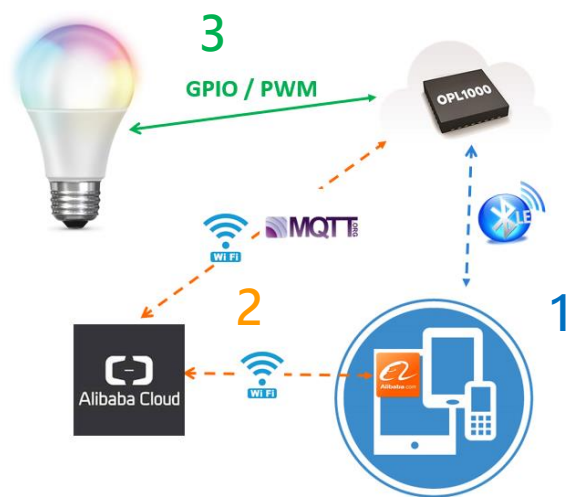
4.2. 执行流程和模块说明

本章节介绍 OPL1000 固件处理流程。

4.2.1. 执行流程

- 1. Use the tool on mobile device trigger OPL 1000 connect to Internet through BLE
- 2. ALI APK send the command for lighting to cloud then cloud server will send MQTT topic to OPL 1000. OPL 1000 can update lighting status to cloud server if lighting have change.
- 3. OPL 1000 will control lighting by GPIO or PWM.

Figure 14:固件执行流程图



4.2.2. 主要 Task Handler

{本章节说明几个主要 Task handler 完成的功能， TBD}

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 协议工作原理如错误!未找到引用源。所示。

Figure 15: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）

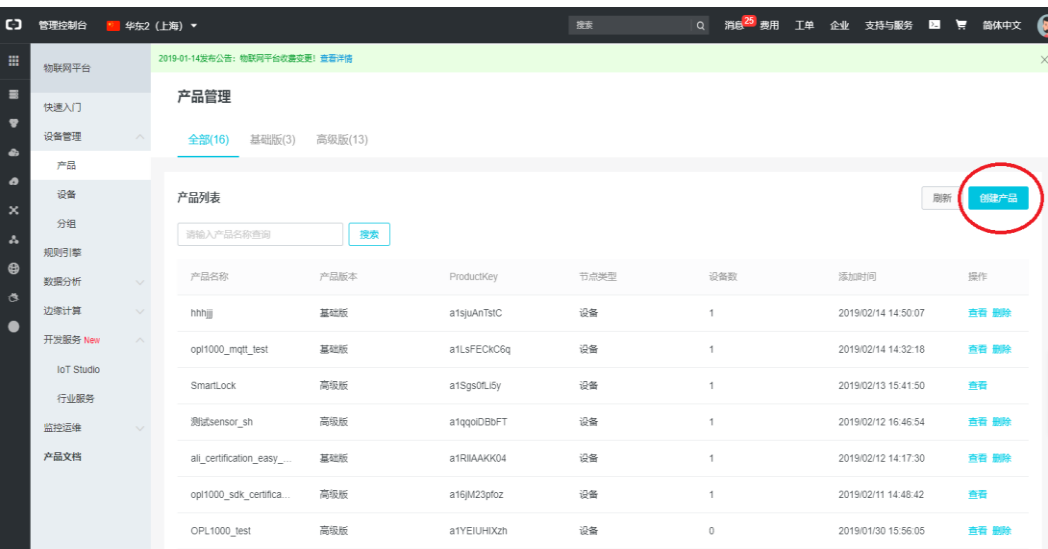
MQTT 会构建底层网络传输：它将建立客户端到服务器的连接，提供两者之间的一个有序的、无损的、基于字节流的双向传输，当应用数据通过 MQTT 网络发送时，MQTT 会把与之相关的服务质量（QoS）和主题名（Topic）相关连。

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 Smart Lighting，可以在这类产品中添加多个设备。下面以 Smart Lighting 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 16: 阿里云 APP 创建界面



创建智能照明设备过程：

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/电工照明/灯，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 17: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 18: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 19:生成 Android APP



CONTACT

sales@Opulinks.com