# 泰凌 B80 24G Lighting

# SDK 开发手册

AN-21113001-C1

Ver1.0.0

2021.11.30



## Keyword

B80, remote, control, 24GHz

## Brief

本文主要为泰凌微电子基于 B80 芯片的 24G lighting 的 SDK 开发手册。

**Published by**
**Telink Semiconductor**

**Bldg 3, 1500 Zuchongzhi Rd,**
**Zhangjiang Hi-Tech Park, Shanghai, China**



**Legal Disclaimer**

This document is provided as-is. Telink Semiconductor reserves the right to make improvements without further notice to this document or any products herein. This document may contain technical inaccuracies or typographical errors. Telink Semiconductor disclaims any and all liability for any errors, inaccuracies or incompleteness contained herein.



**Information**

For further information on the technology, product and business term, please contact Telink Semiconductor Company www.telink-semi.com

For sales or technical support, please send email to the address of:

telinksales@telink-semi.com

telinksupport@telink-semi.com

# 修订历史

| 版本 | 修改内容 |
| --- | --- |
| V1.0.0 | 初次发布 |

# Contents

# 1 SDK 概述

SDK 给用户提供基于 B80 开发的遥控灯，包含 remote-light，remote-RGB light，remote-beacon 这三种模式，用户可以在这些 demo code 基础上开发应用层代码和适配 pcb 板。

## 1.1 SDK 的文件架构

目前 sdk 的文件结构图为：

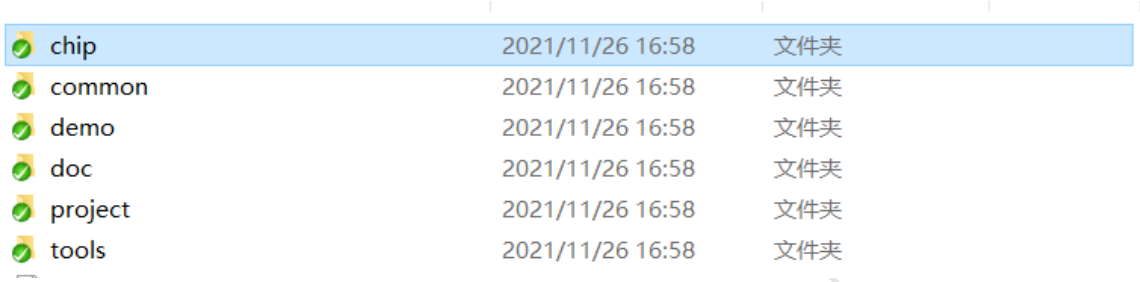


**Figure 1.1**: "sdk 的文件架构图"

### 1.1.1 chip

chip 里面主要包含 B80 芯片的驱动和启动代码，其中启动代码包含了三种模式，boot 文件夹中包含 cstartup_flash.S，cstartup_otp.S，cstartup_sram.S，分别对应宏 MCU_STARTUP_FLASH，MCU_STARTUP_OTP，MCU_STARTUP_SRAM 来控制，通过在 IDE 中配置对应的宏来选择不同的模式。

cstartup_flash.S ： flash 模式，烧录程序在 flash 中运行，开发板有外置 flash 可以选择这种模式。

cstartup_otp.S：otp 模式，烧录程序在 otp 中，由于 B80 最终是采用内置 otp 的方式，最终量产的产品采用 otp 的选项。

cstartup_sram.S：sram 模式，在没有外置 flash，需要做简单调试的时候可以采用这种模式进行调试，避免 otp 被写后无法改变。

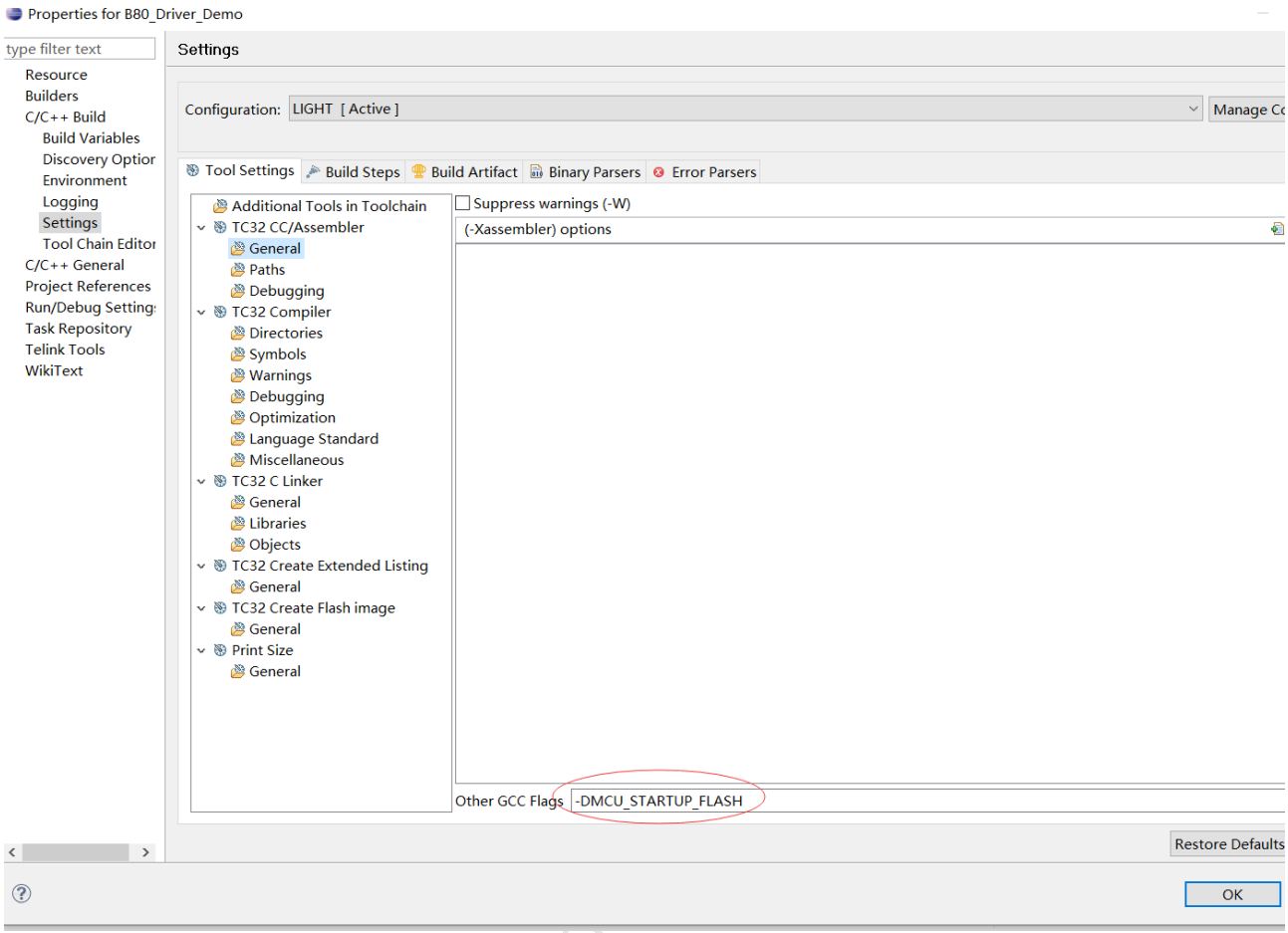


**Figure 1.2**: "编译模式的选择"

Driver 文件夹里面对应的是 B80 对应的驱动文件以及 rf，pm 对应的函数库。

### 1.1.2 common

div_mod.s 定义的是一些通用的接口，例如大数乘法之类的汇编接口。其余的函数是通用的类型定义以及类似于 memcpy，memcmp 之类的接口。

### 1.1.3 demo

demo 文件夹里面定义的是用户层的代码。

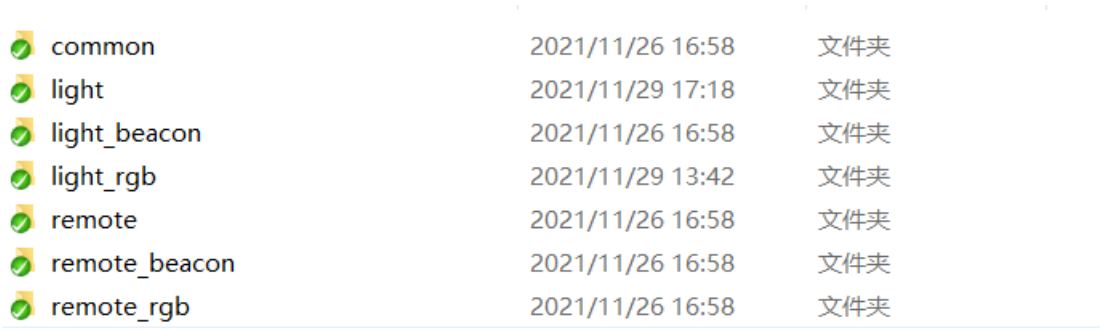


**Figure 1.3:** "demo 文件夹结构"

其中 common 文件夹中包含的是所有编译选项共用的文件夹选项。其余的文件夹分别对应不同的编译分支。

### 1.1.4 project

project 为工程文件夹，其中的.cproject,.project 文件为对应的工程文件，.boot.link 为链接文件，不建议客户修改，proj_lib 为库存放的路径，其中的文件夹为编译生成的文件，例如 LIGHT 文件夹里面会生成 LIGHT.bin 和 LIGHT.lst。

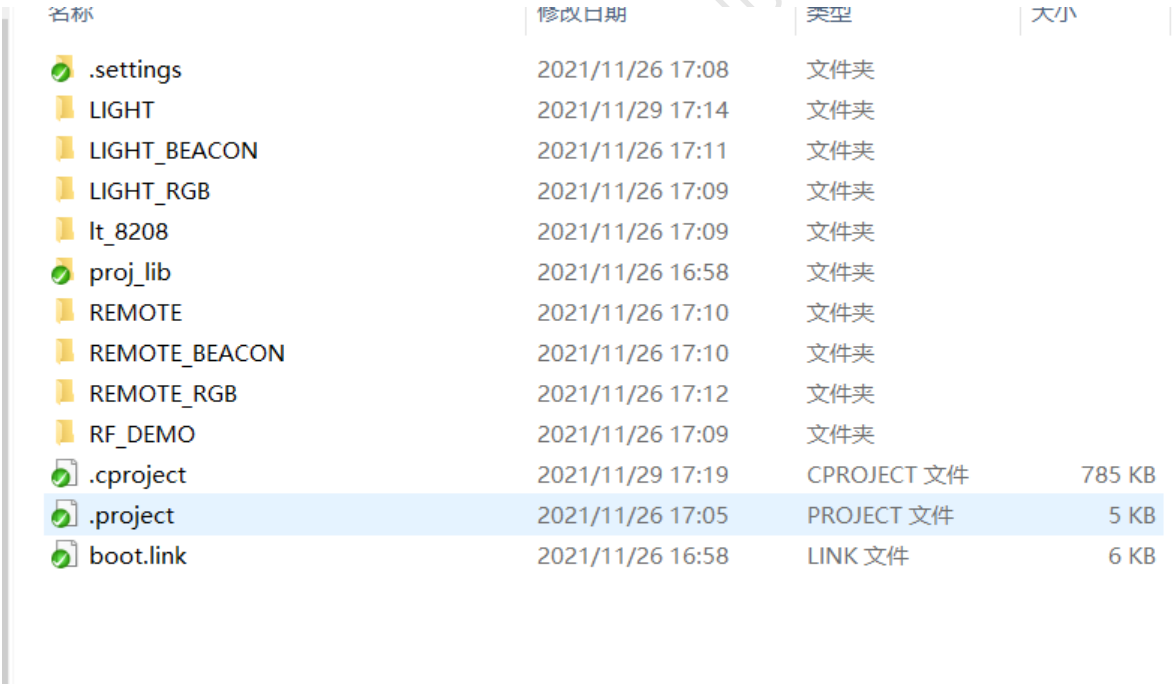


**Figure 1.4:** "project 文件夹结构"

### 1.1.5 **其余文件夹**

Doc 里面对应的是 release 版本信息，tool 里面对应的是一些打包的工具。

## 1.2 Demo project

B80 remote light 的 sdk 是给客户开发的编译分支，用户可以通过 demo 操作来观察直观的效果，也可以在 demo code 上面修改，完成自己的应用程序部分的开发。

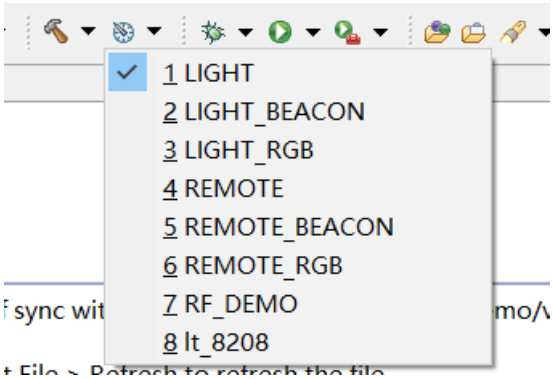


**Figure 1.5**: "编译选项列表"

## 1.3 SDK **收发包以及配对部分的处理**

以下讲述配对和收发包流程目前都基于 remote-light 的模式。

### 1.3.1 SDK **配对的流程介绍**

Tx 发射的参数：

发射频道：2401mhz，2424mhz，2451mhz，2476mhz，4 个信号上发送。

发射的 accesscode：{0x71,0x76,0x51,0x39,0x95}

发射功率为：11.46dbm

Sdk 的配对原理：

上电 5s 内，短按某组的开灯键一次。即可完成配对。配对成功 LED 闪灯 3 次。

上电 5s 内，短按某组的开灯键连按 5 次（间隔小于 500ms）。即可完成清除配对信息。清除配对信息成功 LED 闪灯 5 次。

上电 5s 内，收到非开灯键命令，退出配对模式，进入正常状态。

sdk 配对的流程图：

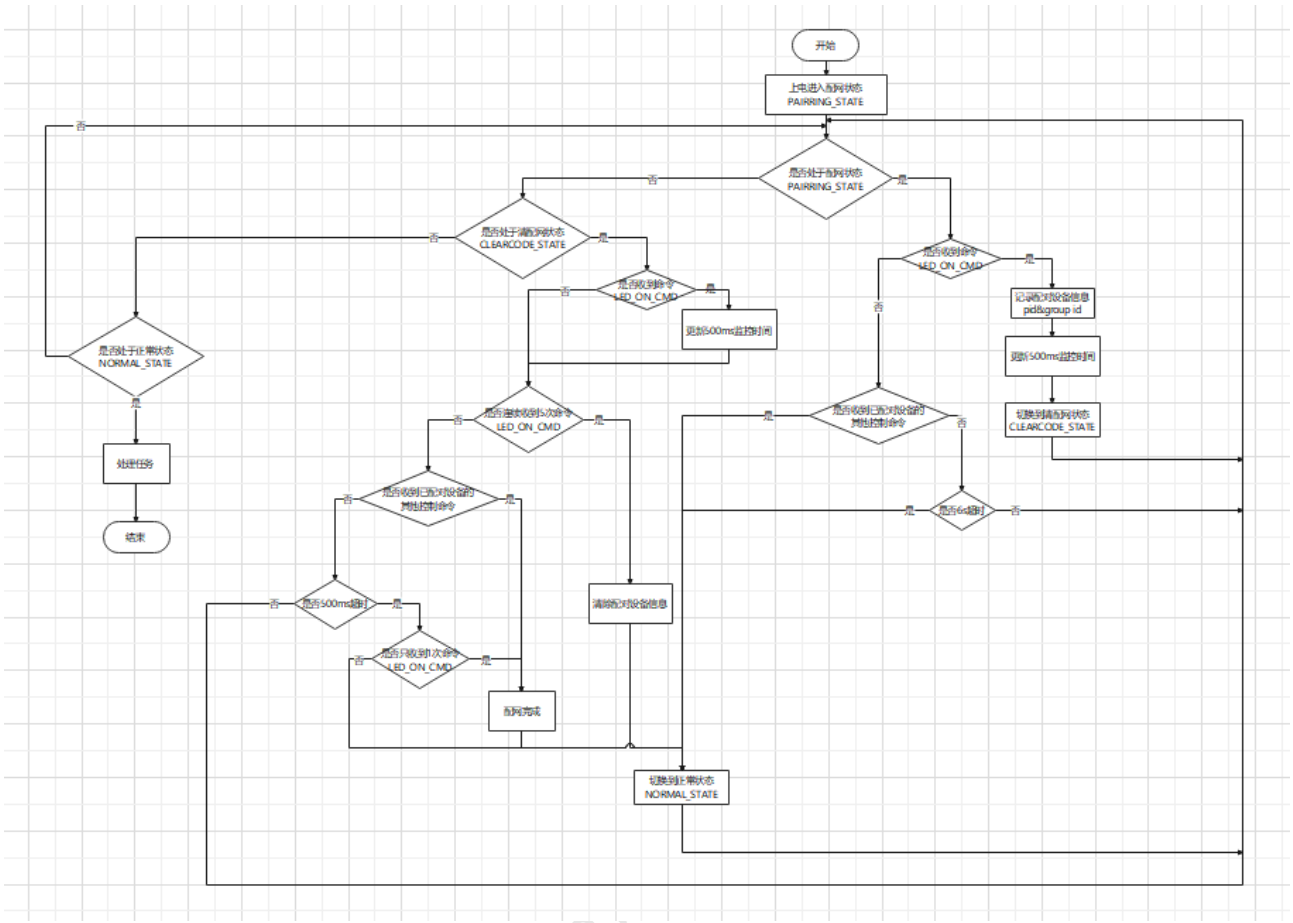


**Figure 1.6**: "light&remote 配对流程"

### 1.3.2 SDK **收发包的流程介绍**

以 remote 工程为例, 发包的函数说明:

```
typedef struct{
    unsigned int   dma_len;              // 0~3 DMA length
    unsigned char  rf_len;                  // 4 rf data length = 0x10
    unsigned char  rf_len1;
    unsigned short vid;                  // 5~6 vendor ID
    unsigned int   pid;                  // 7~10 product ID

    unsigned char  control_key;          // 11 function control key
    unsigned char  rf_seq_no;            // 12 rf sequence total number, save this value in 3.3v analog register.

    unsigned short button_keep_counter; // 13~14 sequence number in one certain channel.
    unsigned short control_key_value[3];    // 15, 16, 17, 18
    unsigned char  ttl;
}rf_packet_led_remote_t;//rf data packet from remoter end.
```

**Figure 1.7**: "命令包数据结构"

```
void package_data_init_func(void)
```

发包数据初始化。配置 VID 信息，重特定 otp 地址或者 remote 的 PID，从模拟寄存器获取保存的信息。

```
void package_data_set_newcmd(unsigned char key_value,unsigned char* para)
```

配置新命令。更新 rf_seq_no，控制码和控制码参数。

```
void package_data_send_func(void)
```

调用 rf 模块把命令发送出去。(remote 发送 TTL=5)（命令一般重复发送 15 次，除非中途被更新为新命令。）

```
void package_data_store_func(void)
```

保存命令码的 group&rf_seq_no 信息到模拟寄存器中。防止在深睡唤醒后丢失信息。

发包的逻辑流程图：

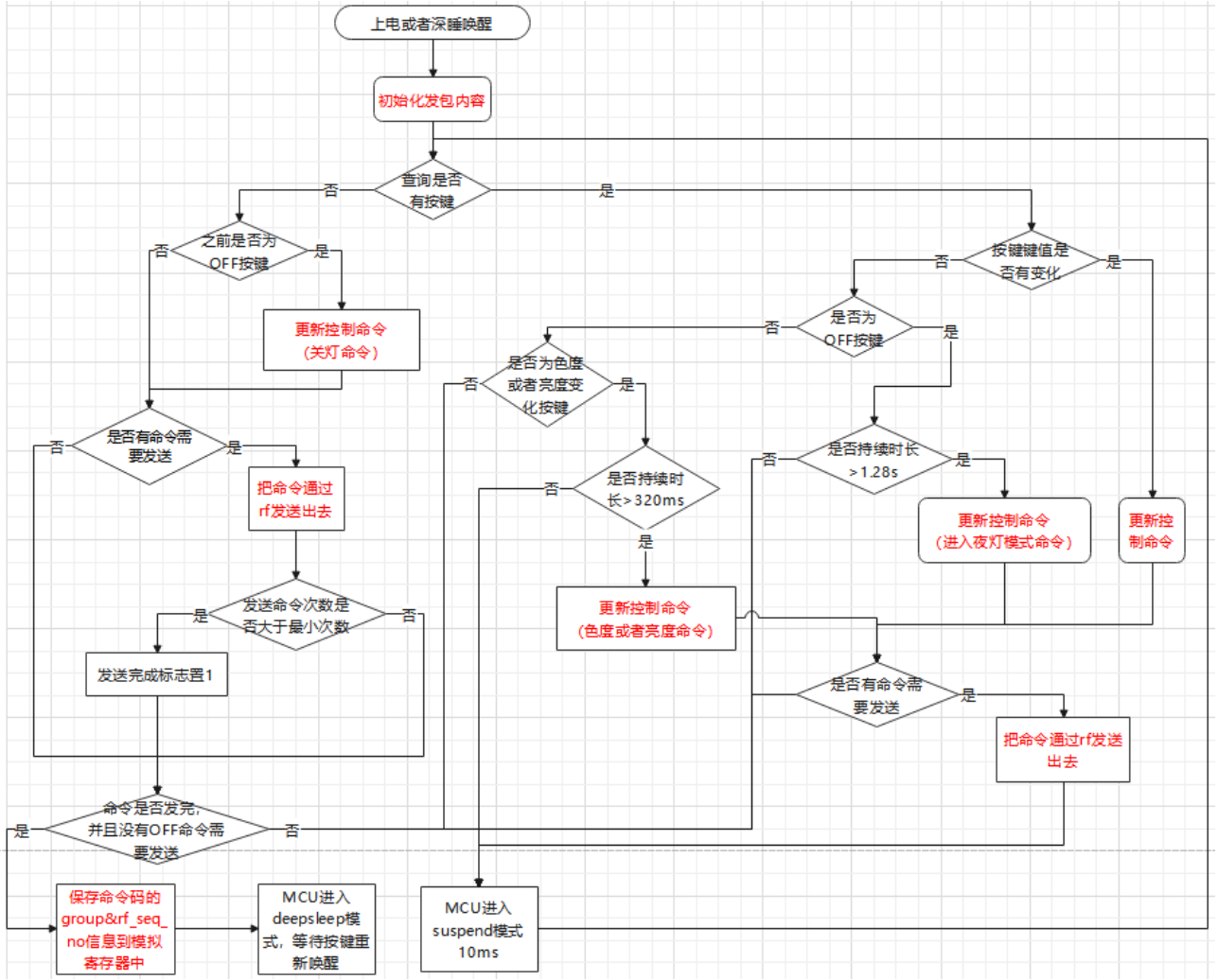


**Figure 1.8**: "remote 命令发包流程"

以 light 工程为例，，转发命令的函数说明：

Light 把收到的 remote 控制命令转发给其他 light 设备。

```
void rfc_send_relay_pkt(void)
```

转发收到的 remote 命令，ttl 减 1，防止无限转发。

发包的逻辑流程图：

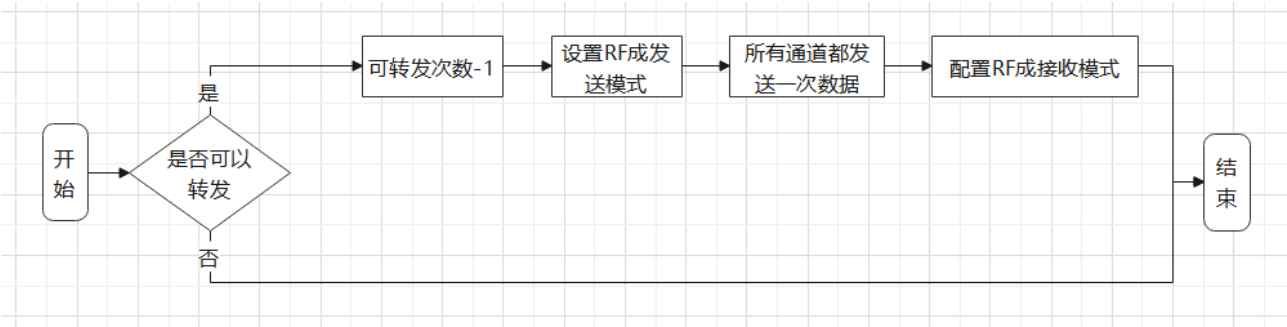


**Figure 1.9**: "light 命令转发流程"

收包的接口说明：

```
void sys_status_process(void)
```

中断函数 void light_irq_handler(void) 中判断接收的数据为配对的 remote 发出的新命令。然后在函数 void sys_status_process(void) 处理相应的控制命令。

收包的逻辑流程图：

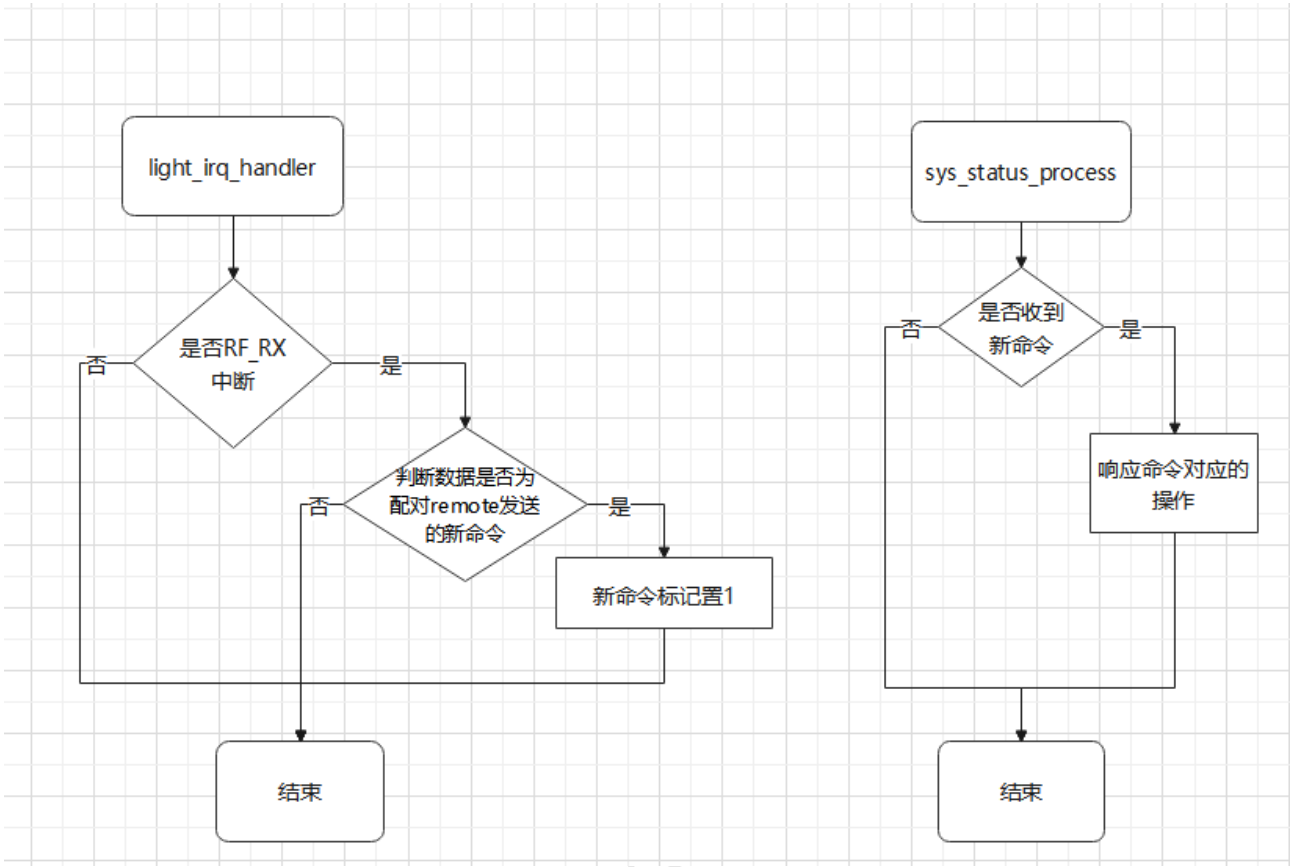


**Figure 1.10**: "light 命令响应流程"

## 14 remote 矩阵键盘扫描

按键矩阵行和列个数配置:

```
#define KB_COL_NUM   5
#define KB_ROW_NUM   3
```

按键行列的管脚配置:

```
gpio_column[KB_COL_NUM]={GPIO_PB0,GPIO_PB1,GPIO_PB4,GPIO_PB5,GPIO_PB6};//矩阵列的 IO
gpio_row[KB_ROW_NUM]   ={GPIO_PD3,GPIO_PD6,GPIO_PA0};//矩阵行的 IO
```

按键键值配置:

```
key_table[KB_ROW_NUM][KB_COL_NUM] //按键表格
```

按键扫描原理:

当一个按键两端分别接一个 IO 口，一个 IO 口置高电平另一个置低电平，当按下按键时高电平 IO 口电平被拉低，另一端还是为低电平，这时检测 IO 口值就是两个低电平。

## 1.5 EEPROM

Eeprom 目前是内置的 B80 芯片内部的，用于存储配对 pid 和 group 的信息，大小为 256bytes。

```
void e2prom_init();
```

配置 eeprom 的初始化，以及内部 eeprom 的管脚的初始化。

```
void e2prom_write (u8 adr, u8 *p, int len);
```

adr 为 eeprom 内部的地址，p 为写入 buffer 的头指针，len 为写入到 eeprom 的长度。

```
void e2prom_read (u8 adr, u8 *p, int len);
```

adr 为 eeprom 内部的地址，p 为读取 buffer 的头指针，len 为读取到 eeprom 的长度。

## 1.6 LED

亮度加减键和色度加减键调节 LED 变化原理：

light 目前预设 10 档变化，每档参数保存在数组 led_luminance_value 和 led_chroma_value。接收亮度或者色度变化命令后，跳入下一档参数作为 LED 的目标值。LED 渐变到目标值后，就不再变化。

LED 颜色渐变过程：

每隔 5ms 检查一次，当设置的目标亮度或者色度和当前值有差异的时候，单步调节当前的参数（亮度单步变化为 10，色度单步变化为 1，不足对齐目标值），一直到调节到目标值。完成颜色渐变的过程。

## 1.7 SDK **调试说明**

### 1.7.1 Tdebug **调试**

通过 wtcdb 工具，查看对应的变量，通过变量观察对应的变量来 debug。

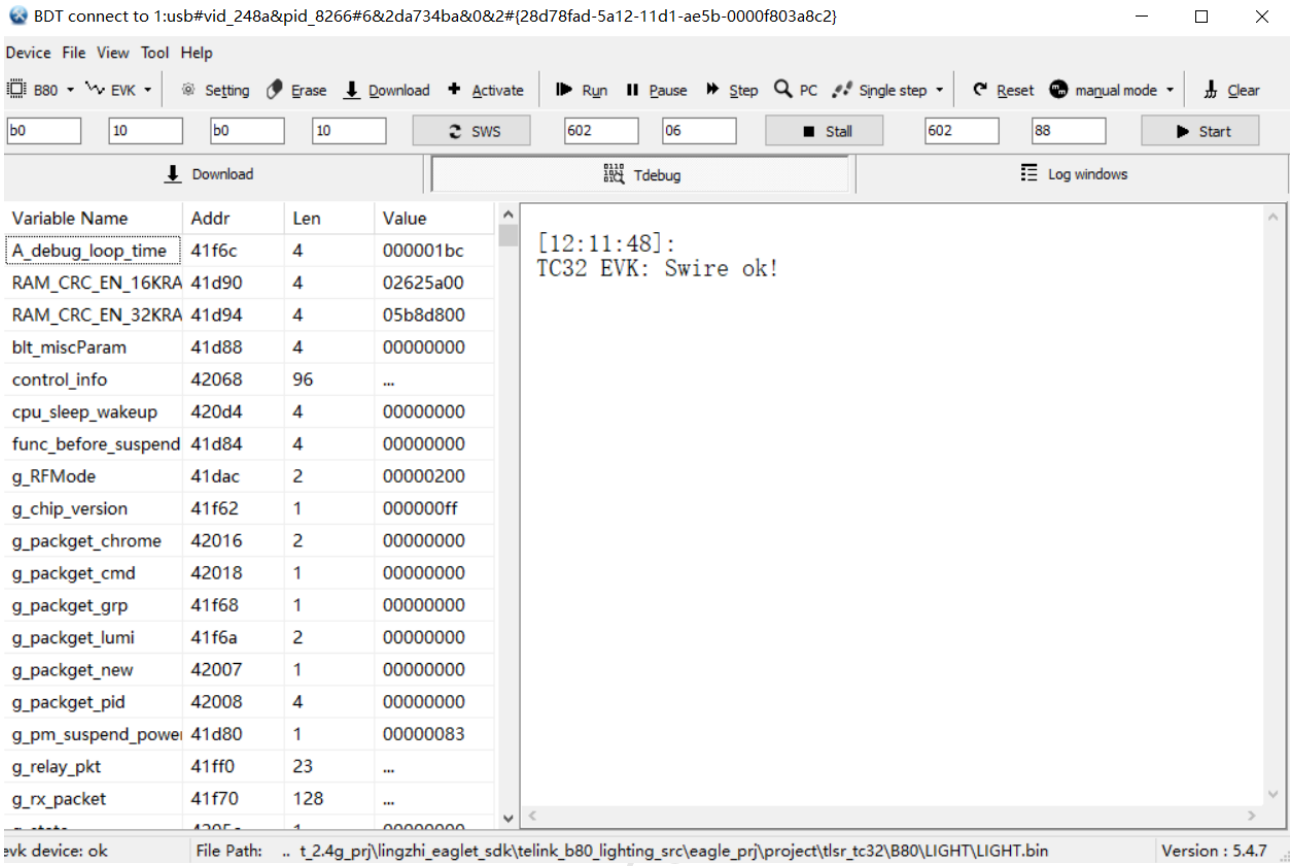


**Figure 1.11**: "wtcdb 调试"

### 1.7.2 **串口打印调试说明**

```
#define DEBUG_MODE 1
```

通过 DEBUG_MODE 来打开和关闭打印函数。

```
#define PRINT_BAUD_RATE                 115200 //1M baud rate,should Not bigger than 1Mb/s
#define DEBUG_INFO_TX_PIN               GPIO_PD3
```

通过 PRINT_BAUD_RATE 来配置串口的波特率，DEBUG_INFO_TX_PIN 来配置打印的接口。Printf 作为标准打印的接口，printhex 打印数组。

DEBUG_MODE =0 关闭打印模式，并且在编译的过程中，不会编入和打印相关的函数，客户在调试阶段可以打开，量产程序建议关闭能节约 ram 和 code。

# 2 SDK 的代码说明

分别介绍 remote-light，remote-rgb light，remote-beacon 这 3 种模式的详细介绍，以及以 light 工程为例，介绍初始化接口以及主循环里面的处理。

## 2.1 Remote

### 2.1.1 初始化函数接口介绍

```
led_gpio_init(LED1)
```

初始化 LED 管脚。每次只能初始化 1 个。

```
keyscan_gpio_init();
```

初始化按键矩阵管脚

```
package_data_init_func()
```

初始化发包数据。包括固化的 VID、从 OTP 中读取的 PID、从模拟寄存器恢复的 group&rf_seq_no.

```
rfc_init_func()
```

初始化 RF 模块。RF 配置成 tx_mode，用于发送命令。

### 2.1.2 Loop 接口介绍

```
unsigned char keyscan_scan_func(void)
```

扫描按键矩阵。返回按键键值。通过按键发送控制命令。(OFF 命令按键抬起才开始发送，OFF 长按发送夜灯模式命令，色温和亮度变化支持长按 >320ms 连续发送新命令，其他命令按键按下后发送命令)

## 2.2 Light

### 2.2.1 初始化函数接口介绍

```
rfc_init_func()
```

初始化 RF 模块。RF 配置成 Rx_mode。用于接收 remote 命令。

```
led_pwm_init_func()
```

把 LED 灯管脚配置成 PWM 功能。后续通过改变 PWM 波形，改变 LED 亮度和色度以及亮灭等状态。

```
led_init_func()
```

根据之前保存的参数，恢复 light 的亮灯状态。

```
sys_status_init()
```

初始化系统状态。默认配网状态。

### 2.2.2 Loop **接口介绍**

```
void sys_status_process(void)
```

处理 RF 接收到的控制命令，并触发对应的操作。

```
led_task_process_func()
```

处理 LED 闪烁。每隔 500ms，亮度由 500 变成 0，再由 0 变成 500。循环闪烁。

处理 LED 亮度和色度渐变。每隔 5ms，单步变化 LED 当前的亮度和色度参数，逐步向目标亮度和色度靠近，达到目标保存 LED 参数。

更改 LED 本地模式信息。如果非本地模式上电超过 500ms，则保存状态"下次上电不需要切换到本地模式"。下次重新上电不会进入本地模式。

```
time_event_process_func()
```

定时切换 RF 的接收通道。（默认 20ms 切换一次，4 通道循环切换）。

```
sys_status_check_func()
```

检查系统状态。配对模式超过 6s，切换到正常模式。CMD_ON 命令间隔超过 500ms, 退出清码状态。

## 2.3 LIGHT_BEACON

略。同 2.2。

## 24 REMOTE_BEACON

略。同 2.3。

## 2.5 LIGHT_RGB

```
rf_packget_pro_func()
```

处理 RF 接收到的控制命令，并触发对应的操作。

```
void led_task_process_func(void)
```

LED 闪烁。每隔 500ms，亮度由 500 变成 0，再由 0 变成 500。循环闪烁。

LED 亮度和色度渐变。每隔 5ms，单步变化 LED 当前的亮度和色度参数，逐步向目标亮度和色度靠近，达到目标保存 LED 参数。

LED 的 RGB 颜色渐变。包括流水灯功能和 RGB 显示。

## 2.6 REMOTE_RGB

略。同 2.3。

# 3 SDK 配置文件说明

## 3.1 射频参数和频点的配置

Accesscode：通过修改 RF_ACCESS_CODE_USE

频点：LIGHT&REMOTE 和 LIGHT RGB/REMOTE RGB 默认使用频点 {1,24,51,76}，修改变量 rf_channel[4] 可以改变频点。LIGHT BEACON&REMOTE BEACON 使用固定频点：{37,38,39}

发射功率：

```
#define RF_POWER                RF_POWER_P11p46dBm
```

发射的模式选择：

```
#define RF_MODE                 RF_PRIVATE_2M
```

## 3.2 IO 口的配置

```
#define LED_R                   GPIO_PB3//read
#define LED_G                   GPIO_PB4//green
#define LED_B                   GPIO_PB5//blue
#define LED_Y                   GPIO_PB6//yellow
#define LED_W                   GPIO_PD4//white
```

通过修改不同的 LED 的 gpio 的定义来修改不同的 LED 的配置。

## 3.3 修改主频

通过修改 app_config.h 文件中的宏定义 CLOCK_SYS_CLOCK_HZ，宏定义选项有 12M,16M,24M,32M,48M。

## 3.4 配置产品 VID

```
#define REMOTE_VID              0x5453
```

## 3.5 配置 REMOTE PID 的 OTP 地址

```
#define PID_ADDR                0x3fe0
```

## 3.6 配置命令发送次数

```
#define TTL_MAX                 5 //remote 命令可转发次数
#define NUM_SENDING_CMD_CTR     15//单个控制命令最少发送次数，除非中途切换新命令
#define NUM_SENDING_CMD_NONE    5//空命令重发次数
```

# 4 操作简介

## 4.1 REMOTE&LIGHT 和 REMOTE_BEACON&LIGHT_BEACON 按键功能介绍


**Figure 4.1**: "REMOTE&LIGHT 和 REMOTE_BEACON&LIGHT_BEACON 按键介绍"

**全开：**

短按或者长按，发出全开灯命令，把附近和 remote 配对的 light 全部开灯。

**全关：**

按下 <1.5s，按键抬起后发出全关灯命令，把附近和 remote 配对的 light 全部关灯。

按键 >1.5s，发送全部进入夜灯模式命令，把附近和 remote 配对的 light 全部进入夜灯模式。

**组 1 开：**

短按或者长按，发出组 1 开灯命令，把附近和组 1 配对的 light 全部开灯。

**组 1 关：**

按下 <1.5s，按键抬起后发出组 1 关灯命令，把附近和组 1 配对的 light 全部关灯。

按键 >1.5s，发送组 1 进入夜灯模式命令，把附近和组 1 配对的 light 全部进入夜灯模式。

**组 2 开/组 3 开/组 4 开/：**同理"组 1 开"。

**组 2 关/组 3 关/组 4 关/：**同理"组 1 关"。

**亮度加/亮度减：**

选择组别（按下组别开关键，例如"组 1 开"或者"组 1 关"），然后按下亮度加/亮度减键。

按键短按，单次发送单次命令。亮度单次变化。

按键长按，每个 320ms 左右，持续发送命令，亮度持续变化。

**色度加/色度减：**

选择组别（按下组别开关键，例如"组 1 开"或者"组 1 关"），然后按下色度加/色度减键。

按键短按，单次发送单次命令。色度单次变化。

按键长按，每个 320ms 左右，持续发送命令，色度持续变化。

## 4.2 REMOTE_RGB&LIGHT_RGB 按键功能介绍


**Figure 4.2:** "REMOTE_RGB&LIGHT_RGB 按键介绍"

**色温灯开：**

短按或者长按，发出开灯命令，把附近和 remote 配对的色温灯全部开灯。

**色温灯关：**

短按或者长按，发出关灯命令，把附近和 remote 配对的色温灯全部关灯。

**亮度加/亮度减/色度加/色度减：**

按键短按，发送单次命令，light led 单次变化。

按键长按，320ms 定时更新命令，light led 持续变化。

RGB 呼吸灯模式不响应色温和亮度调节命令。

**夜灯模式：**

短按或者长按，发出夜灯模式命令，把附近和 remote 配对的色温灯全部进入夜灯模式。

**RGB 呼吸灯模式：**

短按或者长按，发出夜灯模式命令，把附近和 remote 配对的色温灯全部进入 RGB 呼吸灯模式。

**对码：**

light 上电 <6s，remote 短按或者长按发出配对命令，light 收到命令，保留配对信息并闪灯 3 次。

light 上电 >6s，remote 短按或者长按发出配对命令,light 无任何动作。

**清码：**

light 上电 <6s，remote 短按或者长按发出清码命令，light 收到命令，清除配对信息并闪灯 5 次。

light 上电 >6s，remote 短按或者长按发出清码命令,light 无任何动作。

**RGB 设置：**

短按或者长按，发出 RGB 设置命令，把附近和 remote 配对的色温灯全部 RGB 灯亮 50%。