



# HiSpark-WiFi-IoT 交通灯功能编程指南

文档版本 00B01

发布日期 2020/8/3

**版权所有 © 上海海思技术有限公司。保留一切权利。**

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## **商标声明**



**HISILICON**、海思和其他海思商标均为上海海思技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## **注意**

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## **上海海思技术有限公司**

地址：上海市青浦区金泽镇（西岑）水秀路 318 号 101 室 邮编：201718

网址：<http://www.hisilicon.com>



# 前言

## 概述

本文档主要介绍基于海思 WiFi 芯片 Hi3861 开发的 HiSpark-WiFi-IoT 套件演示指导书。

## 产品版本

与本文档相对应的主芯片版本如下。

产品名称	产品版本
Hi3861	V100R001C00SPC021

## 读者对象

本文档（本指南）主要适用于以下工程师：

- 软件开发工程师
- 硬件开发工程师

## 修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2020-08-3	00B01	第一次临时版本发布。





# 目 录

---

前 言.....	i
<b>1 WIFI-IoT 开发板交通灯功能实现 .....</b>	<b>4</b>
1.1 交通灯硬件准备.....	4
1.2 交通灯功能硬件介绍.....	5
1.3 交通灯功能软件实现.....	5



# 1 WIFI-IoT 开发板交通灯功能实现

## 1.1 交通灯硬件准备

WIFI-IOT 开发板套件，名称 HiSpark，以下简称 HiSpark 开发板套件。

图 1.1-1 主板

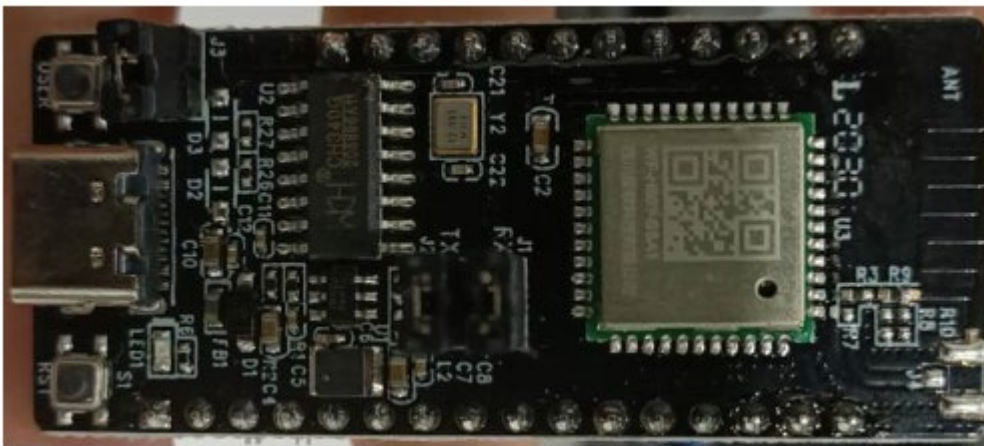
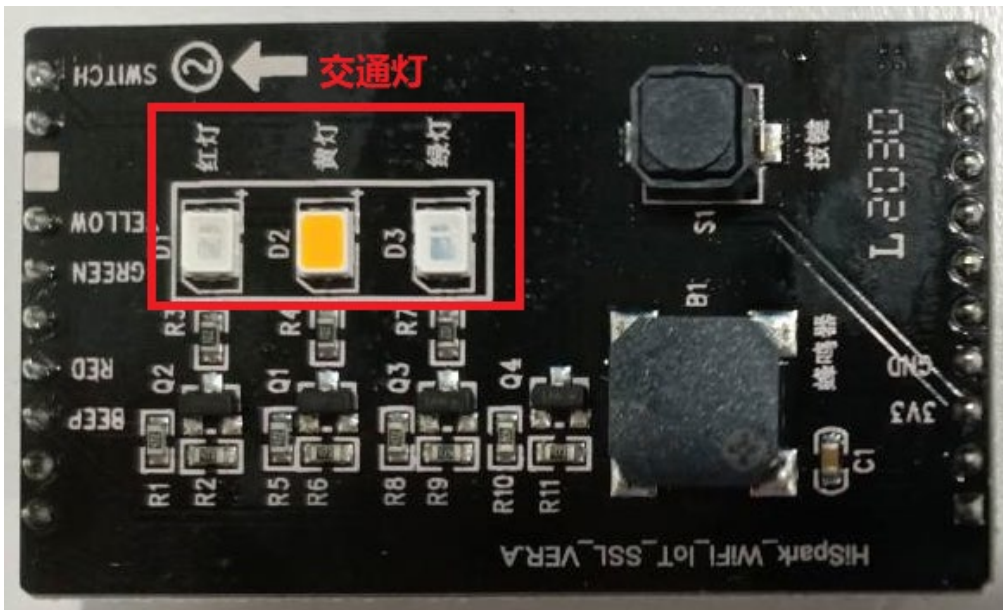


图 1.1-2 交通灯板





## 1.2 交通灯功能硬件介绍

交通灯灯为 IO 控制引脚，当 IO 口为高电平时，对应灯点亮；当 IO 口为低电平时，对应灯熄灭。

交通灯中，各色灯对应的 GPIO 口：

RED	——	GPIO_10
GREEN	——	GPIO_11
YELLOW	——	GPIO_12

## 1.3 交通灯功能软件实现

1. LiteOS 打开 WIFI-IOT 的 SDK 后，首先初始化 PWM 配置

PWM 初始化需要调用三个函数。分别为：

```
hi_u32 hi_io_set_func(hi_io_name id, hi_u8 val);  
hi_u32 hi_pwm_init(hi_pwm_port port);  
hi_u32 hi_pwm_set_clock(hi_pwm_clk_source clk_type);
```

以下对这三个函数的详细介绍：

- hi\_u32 hi\_io\_set\_func(hi\_io\_name id, hi\_u8 val);

此函数为 IO 端口复用功能配置函数。

**hi\_io\_name id:** gpio 硬件管脚编号，对应的是硬件管脚上的 GPIO pin 脚，枚举类型如下：

```
typedef enum {  
    HI_IO_NAME_GPIO_0,    /**< GPIO0 */  
    HI_IO_NAME_GPIO_1,    /**< GPIO1 */  
    HI_IO_NAME_GPIO_2,    /**< GPIO2 */  
    HI_IO_NAME_GPIO_3,    /**< GPIO3 */  
    HI_IO_NAME_GPIO_4,    /**< GPIO4 */  
    HI_IO_NAME_GPIO_5,    /**< GPIO5 */  
    HI_IO_NAME_GPIO_6,    /**< GPIO6 */  
    HI_IO_NAME_GPIO_7,    /**< GPIO7 */  
    HI_IO_NAME_GPIO_8,    /**< GPIO8 */  
    HI_IO_NAME_GPIO_9,    /**< GPIO9 */  
    HI_IO_NAME_GPIO_10,   /**< GPIO10 */  
    HI_IO_NAME_GPIO_11,   /**< GPIO11 */  
    HI_IO_NAME_GPIO_12,   /**< GPIO12 */  
    HI_IO_NAME_GPIO_13,   /**< GPIO13 */
```



```
HI_IO_NAME_GPIO_14,    /**< GPIO14 */
HI_IO_NAME_SFC_CSN,    /**< SFC_CSN */
HI_IO_NAME_SFC_IO1,    /**< SFC_IO1 */
HI_IO_NAME_SFC_IO2,    /**< SFC_IO2 */
HI_IO_NAME_SFC_IO0,    /**< SFC_IO0 */
HI_IO_NAME_SFC_CLK,    /**< SFC_CLK */
HI_IO_NAME_SFC_IO3,    /**< SFC_IO3 */
HI_IO_NAME_MAX,
} hi_io_name;
```

**hi\_u8 val** : 对应 GPIO 引脚功能，如果该引脚复用，可以配置为其他的复用功能，如：UART/PWM/I2C/SPI/SDIO 等功能；

**例如：**需要选取 gpio\_10 的 PWM 复用功能，找到 hi\_io\_func\_gpio\_10 枚举定义：

```
typedef enum {
    HI_IO_FUNC_GPIO_10_GPIO,
    HI_IO_FUNC_GPIO_10_I2C0_SDA,
    HI_IO_FUNC_GPIO_10_UART2_CTS_N,
    HI_IO_FUNC_GPIO_10_SDIO_D3,
    HI_IO_FUNC_GPIO_10_SPI0_CK,
    HI_IO_FUNC_GPIO_10_PWM1_OUT,
    HI_IO_FUNC_GPIO_10_I2S0_TX = 7,
} hi_io_func_gpio_10;
```

找出 PWM 复用功能对应参数为 HI\_IO\_FUNC\_GPIO\_10\_PWM1\_OUT，对应的函数实现为 hi\_io\_set\_func (HI\_IO\_NAME\_GPIO\_10, HI\_IO\_FUNC\_GPIO\_10\_PWM1\_OUT)；功能为将 GPIO10 配置为 PWM 复用。

- hi\_u32 hi\_pwm\_init(hi\_pwm\_port port);

此函数功能为初始化 PWM。

**hi\_pwm\_port port:** 对应 PWM 端口，取值范围为 hi\_pwm\_port 枚举值，枚举类型如下：

```
typedef enum {
    HI_PWM_PORT_PWM0 = 0,
    HI_PWM_PORT_PWM1 = 1,
    HI_PWM_PORT_PWM2 = 2,
    HI_PWM_PORT_PWM3 = 3,
    HI_PWM_PORT_PWM4 = 4,
    HI_PWM_PORT_PWM5 = 5,
```





```
HI_PWM_PORT_MAX
```

```
} hi_pwm_port;
```

不同 IO 口对应的 PWM 端口不同，需选择对应端口。

**例如：**GPIO\_10 的 PWM 端口为 PWM1，对应的实现函数为 hi\_pwm\_init (HI\_PWM\_PORT\_PWM1)；

- hi\_u32 hi\_pwm\_set\_clock(hi\_pwm\_clk\_source clk\_type);

此函数功能为设置 PWM 的时钟。

**hi\_pwm\_clk\_source clk\_type:** 时钟源类型，取值范围为 hi\_pwm\_clk\_source 枚举值，枚举类型如下：

```
typedef enum {  
/*160M 工作时钟*/  
    PWM_CLK_160M,  
/*24M 或 40M 晶体时钟*/  
    PWM_CLK_XTAL,  
/*最大值，不可使用*/  
    PWM_CLK_MAX  
} hi_pwm_clk_source;
```

**例如：**选择工作时钟为 PWM 的时钟源，对应的实现函数为 hi\_pwm\_set\_clock(PWM\_CLK\_160M);

## 2. 输出 PWM 信号

PWM 输出函数：hi\_u32 hi\_pwm\_start(hi\_pwm\_port port, hi\_u16 duty, hi\_u16 freq);

此函数为启动 PWM 信号输出。其中，各参数的含义为：

**hi\_pwm\_port port:** PWM 端口号

**hi\_u16 duty:** 占空比计数值（此值可自己设置）

**hi\_u16 freq:** 分频倍数（此值可自己设置）

信号占空比为： $duty/freq$ ，占空比越大，灯越亮。

PWM 频率为： $时钟源频率/freq$ 。

**例如：**调节 GPIO\_10 红灯的亮度。时钟源频率为 160M，分频系数为 40000，PWM 频率为  $160M/40000 = 4000$ ，代码实现如下：

```
hi_pwm_start(HI_PWM_PORT_PWM1, 1,40000) ;//不亮
```

```
hi_pwm_start(HI_PWM_PORT_PWM1, 20000,40000) ;//中等亮度
```

```
hi_pwm_start(HI_PWM_PORT_PWM1, 40000,40000) ;//最亮
```