

# OpenHarmony E53模块开发-智慧路灯



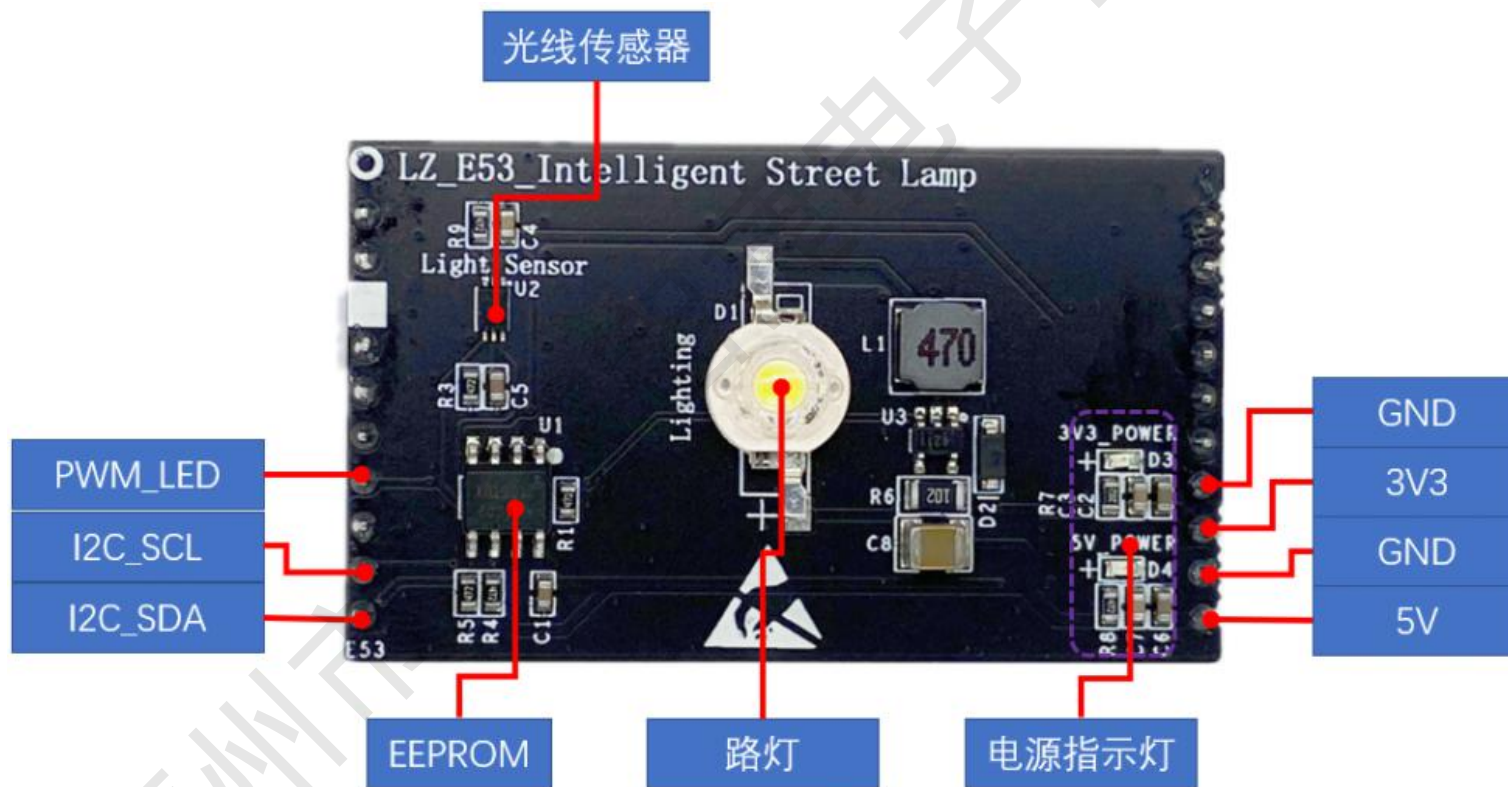
# 目 录

## CONTENTS

- [ 01 ] 硬件设计
- [ 02 ] 工作原理
- [ 03 ] API分析
- [ 04 ] 实例分析

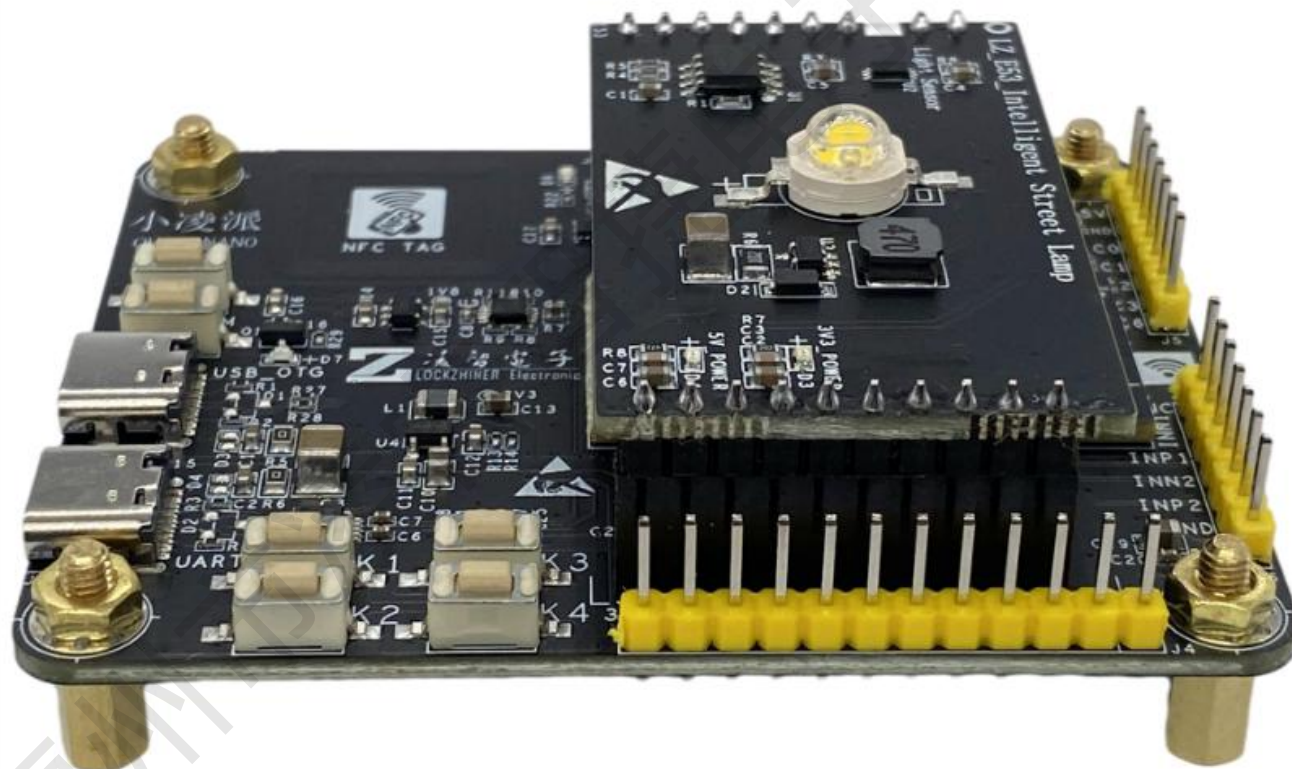
# 01

## 硬件设计



# 01

## 硬件设计



## 02

## 工作原理

## ● 指令集合结构

指令	功能代码	注释
断电	0000_0000	无激活状态。
通电	0000_0001	等待测量指令。
重置	0000_0111	重置数字寄存器值, 重置指令在断电模式下不起作用。
连续 H 分辨率模式	0001_0000	在 11x 分辨率下开始测量。 测量时间一般为 120ms。
连续 H 分辨率模式 2	0001_0001	在 0.51x 分辨率下开始测量。 测量时间一般为 120ms。
连续 L 分辨率模式	0001_0011	在 411x 分辨率下开始测量。 测量时间一般为 16ms。

## 02

## 工作原理

首先，我们通过i2c总线下发“通电”指令给光线传感器，让光线传感器处于准备状态。“通电”指令就是0x01。

其次，通过i2c总线下发“连续H分辨率模式”给光线传感器，让光线传感器处于11x分辨率下开始测量，测量时间一般为120ms。

再次，通过i2c总线直接读取光线传感器芯片2个数值。这2个数据就是当前光线传感器的光线强度。

最后，根据读取到的光线强度，判断路灯是否要打开。

## 03

# API设计

智慧路灯接口的头文件

```
/vendor/lockzhiner/rk2206/samples/c3_e53_intelligent_street_lamp/include/e53_intelligent_street_lamp.h
```

OpenHarmony E53模块开发中，智慧路灯模块的接口主要分为几大类：

- (1) 初始化智慧路灯模块；
- (2) 控制光线传感器；
- (3) 控制路灯。

## 03

# API设计

```
void void e53_isl_init();
```

该函数主要功能是智慧路灯模块初始化，包括初始化I2C0、初始化路灯控制GPIO以及初始化光线传感器。

无返回值。

```
void start_bh1750();
```

该函数主要功能是智慧路灯模块使能光线传感器。即通过i2c总线下发连续H分辨率模式指令集。

无返回值。



## 03

# API设计

```
float e53_isl_read_data();
```

该函数主要功能是智慧路灯模块通过i2c总线读取光线传感器数据，得到亮度值。  
返回值是光线传感器返回的亮度值。

```
void isl_light_set_status(SWITCH_STATUS_ENUM status);
```

该函数主要功能是智慧路灯模块控制路灯开关。

■ 参数status：控制路灯亮灭。

无返回值。

## 04

# 实例分析

### 1、打开sdk下面路径的文件

```
vendor/lockzhiner/rk2206/samples/c3_e53_intelligent_street_lamp/e53_intelligent_street_lamp_example.c
```

### 2、创建任务

在e53\_isl\_example函数中，创建的一个任务e53\_isl\_thread。

```
task.pfnTaskEntry = (TSK_ENTRY_FUNC)e53_isl_thread;
```

```
task.uwStackSize = 10240;
```

```
task.pcName = "e53_isl_thread";
```

```
task.usTaskPrio = 24;
```

```
ret = LOS_TaskCreate(&thread_id, &task);
```

## 04

## 实例分析

在e53\_isl\_thread函数中，每2s读取一次传感器数据并打印亮度数据。当亮度小于20时，打开路灯，否则关闭路灯。

```
void e53_isl_thread()
{
    float lum = 0;

    e53_isl_init();

    while (1)
    {
        lum = e53_isl_read_data();

        printf("luminance value is %.2f\n", lum);
```

```
        if (lum < 20)
        {
            isl_light_set_status(ON);
            printf("light on\n");
        }
        else
        {
            isl_light_set_status(OFF);
            printf("light off\n");
        }

        LOS_Msleep(2000);
    }
}
```

### 3、修改编译脚本

修改 `vendor/lockzhiner/rk2206/sample` 路径下 `BUILD.gn` 文件，指定 `c3_e53_intelligent_street_lamp` 参与编译。

```
"/c3_e53_intelligent_street_lamp:e53_isl_example",
```

修改 `device/lockzhiner/rk2206/sdk_liteos` 路径下 `Makefile` 文件，添加 `-le53_isl_example` 参与编译。

```
hardware_LIBS = -lhal_iohardware -lhardware -le53_isl_example
```

### 4、编译固件

```
hb set -root .
```

```
hb set
```

```
hb build -f
```



## 5、烧写固件

## 6、通过串口查看结果

运行结果

```
luminance value is 45.83
```

```
light off
```

```
luminance value is 4.17
```

```
light on
```

```
.....
```

# 谢谢聆听

单击此处添加副标题内容