HiSpark-WiFi-IoT OLED 功能实现编程指南

文档版本　00B01

发布日期　2020/8/3

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**

1 WIFI-IoT 开发板 OLED 功能实现 **4**

1.1 OLED 硬件准备 4

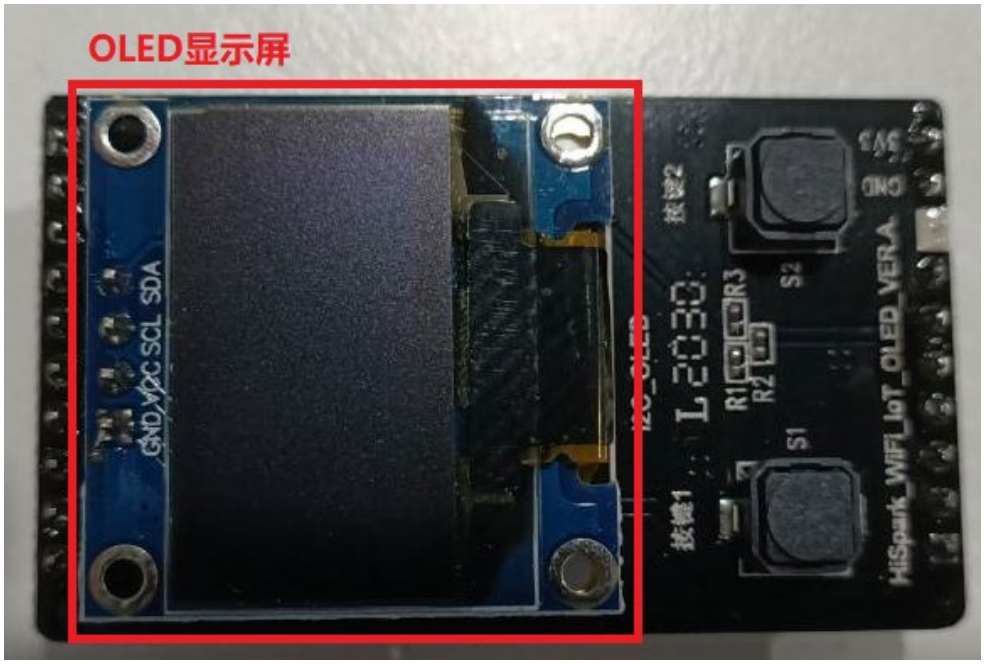
1.2 OLED 硬件介绍 4

1.3 OLED 功能软件实现 4

1 WIFI-IoT 开发板 OLED 功能实现

1.1 OLED 硬件准备

图 1.1-1



1.2 OLED 硬件介绍

OLED 屏幕使用 I2C 接口为 I2C0，对应管脚连接说明：

SDA ----- GPIO_13

SCL ------ GPIO_14

1.3 OLED 功能软件实现

OLED 屏幕通过 I2C 协议进行屏幕显示配置。

1. 首先初始化 I2C 配置，需要调用两个函数，分别是：

hi_u32 hi_i2c_init(hi_i2c_idx id, hi_u32 baudrate);

hi_u32 hi_i2c_set_baudrate(hi_i2c_idx id, hi_u32 baudrate);

以下对这两个函数进行详细介绍：

- hi_u32 hi_i2c_init(hi_i2c_idx id, hi_u32 baudrate);

此函数为 I2C 初始化函数。

hi_i2c_idx id：I2C 硬件设备选择，取值范围为 hi_i2c_idx 枚举类型，如下：

typedef enum {

HI_I2C_IDX_0,

HI_I2C_IDX_1,

} hi_i2c_idx;

hi_u32 baudrate：I2C 通信波特率。

例如：OLED 屏幕与板子通过 I2C0 连接，通讯波特率为 400K，故使用 hi_i2c_init(HI_I2C_IDX_0, 400000);对其进行初始化。

- hi_u32 hi_i2c_set_baudrate(hi_i2c_idx id, hi_u32 baudrate);

此函数功能为配置相对应的 I2C 通讯波特率。

hi_i2c_idx id 为 I2C 硬件设备选择，**hi_u32 baudrate** 所需配置波特率。

例如：对 OLED 屏幕配置 I2C0 为 400K 波特率，其函数实现为 hi_i2c_set_baudrate (HI_I2C_IDX_0, 400000);

2．初始化 OLED 屏幕。直接调用 hi_u32 oled_init(hi_void)函数对 OLED 屏幕初始化。

初始化成功后可调用 oled 函数对屏幕进行显示配置。

主要是要函数有：

hi_void oled_fill_screen(hi_u8 fii_data);

功能为全屏填充数据 fii_data，函数实现如下：

图 1.3-1

```
hi_void oled_fill_screen(hi_u8 fii_data)
{
    hi_u8 m =0;
    hi_u8 n =0;

    for (m=0;m<8;m++) {
        write_cmd(0xb0+m);
        write_cmd(0x00);
        write_cmd(0x10);

        for (n=0;n<128;n++) {
            write_data(fii_data);
        }
    }
}
```

hi_void oled_position_clean_screen(hi_u8 fill_data, hi_u8 line, hi_u8 pos, hi_u8 len);

功能为对 line 行 pos 位置起的 len 个数据填充为 fill_data。函数实现如下：

图 1.3-2

```
hi_void oled_position_clean_screen(hi_u8 fill_data, hi_u8 line, hi_u8 pos, hi_u8 len)
{
    hi_u8 m =line;
    hi_u8 n =0;

    write_cmd(0xb0+m);
    write_cmd(0x00);
    write_cmd(0x10);

    for (n=pos;n<len;n++) {
        write_data(fill_data);
    }
}
```

hi_void oled_show_str(hi_u8 x, hi_u8 y, hi_u8 *chr, hi_u8 char_size);

功能为在 x 行 y 位置的数据写入 chr 字符串，写入长度为 char_size。函数实现如下：

图 1.3-3

```
hi_void oled_show_str(hi_u8 x, hi_u8 y, hi_u8 *chr, hi_u8 char_size)
{
    hi_u8 j=0;

    if (chr == NULL) {
        printf("param is NULL,Please check!!!\r\n");
        return;
    }

    while (chr[j] != '\0') {
        oled_show_char(x, y, chr[j], char_size);
        x += 8;
        if (x>120) {
            x = 0;
            y += 2;
        }
        j++;
    }
}
```