



# HiSpark-WiFi-IoT 人体红外传感器编程指南

文档版本 00B01

发布日期 2020/8/6

**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
2020-08-6	00B01	第一次临时版本发布。





# 目 录

前 言.....	i
<b>1 WIFI-IoT 开发板-人体红外传感器功能实现.....</b>	<b>4</b>
1.1 WIFI-IOT 开发板套件——人体红外传感器工作原理.....	4
1.2 人体红外传感器功能实现逻辑.....	5



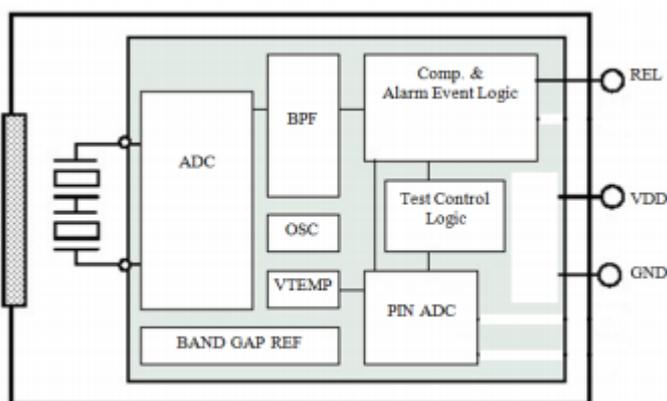
# 1 WIFI-IoT 开发板-人体红外传感器功能实现

## 1.1 WIFI-IOT 开发板套件——人体红外传感器工作原理

1. AS312 人体红外传感器是将数字智能控制电路与人体探测敏感源都集中成电磁屏蔽罩内的热释电红外传感器。

人体探测源将感应到的人体移动信号通过甚高阻抗差分输入电路耦合到数字智能集成电路芯片上，数字智能集成电路将信号转化成 15 位 ADC 数字信号，当 PIR 信号超过选定的数字阈值时就会有延时的 REL 电平输出，所有的信号处理都在芯片上完成。

图 1.1-1 AS312 红外传感器内部结构图



2. AS312 特性：

- 1) 数字信号处理；
- 2) 低电压、低功耗，启动后瞬间工作；
- 3) 电源抑制比高，抗射频干扰；
- 4) 特别内置红外传感器的二阶巴特沃带通滤波器，屏蔽其他频率输入干扰；
- 5) 稳定性好、有效抑制重复晃动动作。



## 1.2 人体红外传感器功能实现逻辑

1. AS312 可以通过:

- 1) 采集 ADC 电压, 根据阈值做业务处理;
- 2) 也可直接判断接在 AS312 上的 GPIO 的电平高/低, 判断是否有人在移动。

2. HiSpark 开发板代码采用的是直接读取 GPIO 口上的电平变化, 判断是否有人在移动。

AS312 人体红外传感器接在 HiSpark 的 GPIO7 端口。

图 1.2-1 软件实现逻辑

```
/*
mode:5.Human detect
模式5: 人体红外检测模式(设置一个好看的灯高度)
有人靠近灯亮, 无人在检测区域灯灭
*/
hi_void human_detect_sample(hi_void)
{
    hi_u8 current_mode = g_current_mode;

    printf("human_detect_sample start\n");

    hi_io_set_func(HI_IO_NAME_GPIO_7, HI_IO_FUNC_GPIO_7_GPIO); // 由gpio8 改为 gpio7, 老硬件用的是gpio8,新硬件用的是gpio7
    hi_gpio_set_dir(HI_GPIO_IDX_7, HI_GPIO_DIR_IN);
    hi_gpio_value gpio_val = HI_GPIO_VALUE1;

    while (1) {
        /*use adc channal_0*/
        // g_someone_walking_flag = get_human_status();
        /*use gpio7*/
        hi_gpio_get_input_val(HI_GPIO_IDX_7, &gpio_val); // 由gpio8 改为 gpio7, 老硬件用的是gpio8,新硬件用的是gpio7

        if (gpio_val == HI_GPIO_VALUE1) {
            g_someone_walking_flag = OLED_FALG_ON;
        } else {
            g_someone_walking_flag = OLED_FALG_OFF;
        }

        if (g_someone_walking_flag == OLED_FALG_ON) {

            hi_pwm_start(HI_PWM_PORT_PWM1, PWM_SMALL_DUTY, PWM_MIDDLE_DUTY);
            hi_pwm_start(HI_PWM_PORT_PWM2, PWM_SMALL_DUTY, PWM_MIDDLE_DUTY);
            hi_pwm_start(HI_PWM_PORT_PWM3, PWM_SMALL_DUTY, PWM_MIDDLE_DUTY);
        } else if (g_someone_walking_flag == OLED_FALG_OFF) {

            hi_pwm_start(HI_PWM_PORT_PWM1, PWM_LOW_DUTY, PWM_MIDDLE_DUTY);
            hi_pwm_start(HI_PWM_PORT_PWM2, PWM_LOW_DUTY, PWM_MIDDLE_DUTY);
            hi_pwm_start(HI_PWM_PORT_PWM3, PWM_LOW_DUTY, PWM_MIDDLE_DUTY);
        }

        if (current_mode != g_current_mode) {
            hi_pwm_start(HI_PWM_PORT_PWM1, PWM_LOW_DUTY, PWM_MIDDLE_DUTY);
            hi_pwm_start(HI_PWM_PORT_PWM2, PWM_LOW_DUTY, PWM_MIDDLE_DUTY);
            hi_pwm_start(HI_PWM_PORT_PWM3, PWM_LOW_DUTY, PWM_MIDDLE_DUTY);
            break;
        }
        hi_sleep(SLEEP_1_MS);
    }
}
```

读取GPIO7端口的电平  
判断是否有人在移动