



# ESP8266 IR红外例程及使用说明

**Version 0.2**

**Espressif Systems IOT Team**

**Copyright (c) 2015**

## 免责声明和版权公告

本文中的信息，包括供参考的URL地址，如有变更，恕不另行通知。

文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

Wi-Fi联盟成员标志归Wi-Fi联盟所有。

文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权归© 2015 乐鑫信息科技（上海）有限公司所有。保留所有权利。



# Table of Contents

1.	红外发送与接收使用说明 .....	4
1.1.	红外发送使用方法 .....	4
1.2.	红外接收使用方法 .....	4
2.	使用详细过程 .....	4
2.1.	timer_frc2_dsr_init() .....	5
2.2.	ir_rx_init() .....	5
2.3.	ir_timer() .....	5
2.4.	test_ir_nec_tx() .....	5
2.5.	红外接收中断 .....	6
2.6.	ir_intr_test() .....	6



## 1. 红外发送与接收使用说明

本文档以32bit NEC发送与接收协议为例，实现红外遥控功能。

接收：

红外接收功能主要通过GPIO的边沿中断完成。通过读取系统counter时间，将两次时间相减可以得到波形持续时间。

需要注意，红外接收通过GPIO 中断实现，同时，系统只能注册一个IO中断处理程序，如果有其他IO口也需要中断的话，需要将这些中断在同一个处理程序中处理(判断中断源并相应处理)

发送：

以I2S的BCK或者WS脚产生38KHz载波用于发送。通过系统FRC2的DSR TIMER产生并驱动红外发送状态机。

### 1.1. 红外发送使用方法

(1) 初始化FRC2：

RTOS SDK:

```
timer_frc2_dsr_init();
```

NONE OS SDK:

```
system_timer_reinit();
```

(2) 设置标准格式的数据：

```
set_tx_data(u8 tx_addr,u8 tx_cmd,u8 tx_rep)
```

或自定义数据：set\_tx\_data\_custom(u8 tx\_addr,u8 tx\_addr2,u8 tx\_cmd,u8 tx\_cmd2,u8 tx\_rep)

(3) 开始发送数据：ir\_tx\_test();

### 1.2. 红外接收使用方法

(1) 初始化红外接收GPIO端口和GPIO中断：ir\_rx\_init();

(2) 当触发中断时，进入中断服务程序interrupt\_serv();在收到32位正确数据后，可以使用这些数据。

## 2. 使用详细过程

该例程使用以GPIO14作为红外接收，接收GPIO13红外发送的数据。

从void user\_init(void)开始：



### 2.1. timer\_frc2\_dsr\_init()

timer\_frc2\_dsr\_init(); //FRC2初始化，详见FRC\_TIMER说明

### 2.2. ir\_rx\_init()

ir\_rx\_init(); //红外接收初始化

```
GPIO_ConfigTypeDef gpio_in_cfg; //定义一个gpio输入结构体
gpio_in_cfg.GPIO_IntrType = GPIO_PIN_INTR_NEGEDGE; //GPIO下降沿触发中断
gpio_in_cfg.GPIO_Mode = GPIO_Mode_Input; //GPIO输入模式
gpio_in_cfg.GPIO_Pin = GPIO_Pin_14; //IR引脚为GPIO14
gpio_in_cfg.GPIO_Pullup = GPIO_PullUp_DIS; //无上拉电阻
gpio_config(&gpio_in_cfg); //配置GPIO寄存器
GPIO_REG_WRITE(GPIO_STATUS_W1TC_ADDRESS,
    BIT(IR_GPIO_IN_NUM)); //清IR_GPIO_IN_NUM的中断
gpio_intr_handler_register(interrupt_serv); //指定GPIO中断服务程序
xt_isr_unmask(1 << ETS_GPIO_INUM); //开GPIO中断
```

### 2.3. ir\_timer()

```
ir_timer();
os_timer_disarm(&ir_timer);
os_timer_setfn(&ir_timer, test_ir_nec_tx, NULL);
os_timer_arm_ms(&ir_timer, 2000, 1);
//起ir_timer,用于定时调用test_ir_nec_tx函数，实现定时发送
```

### 2.4. test\_ir\_nec\_tx()

test\_ir\_nec\_tx(); //测试红外nec发送

(1) 设置要发送的数据：

```
set_tx_data(u8 tx_addr, u8 tx_cmd, u8 tx_rep); //标准的要发送的数据，发送处理中，将addr的反码，
cmd反码也一并发送
set_tx_data_custom(u8 tx_addr, u8 tx_addr2, u8 tx_cmd, u8 tx_cmd2, u8 tx_rep); //自定义发送的数据，
tx_addr, tx_addr2, tx_cmd, tx_cmd2构成一个32bit的数据
```

(2) ir\_tx\_test(); //红外发送，分为四步完成：

**step1) case IR\_TX\_IDLE:**



```
gen_i2s_clk(); //产生38KHz载波信号
os_timer_arm_us(&ir_tx_timer, IR_NEC_HEADER_HIGH_US, 0); //9ms的载波
```

**step2) case IR\_TX\_HEADER:**

```
ir_tx_carrier_clr(); //关闭载波
os_timer_arm_us(&ir_tx_timer, IR_NEC_HEADER_LOW_US, 0); //4.5ms的低电平
```

**step3) case IR\_TX\_DATA:** //发送数据，发送地址码和地址反码，命令码和命令反码

ir\_tx\_standard = 1时，使用标准格式的数据，addr1=addr2,cmd1=~cmd2;  
ir\_tx\_standard = 0时，使用自定义的数据，可获得4个数据addr,addr2,cmd,cmd2

**step4) case IR\_TX\_REP :** //发送重复码

## 2.5. 红外接收中断

**step1) ir\_rx\_enable();**

```
gpio_pin_intr_state_set(GPIO_ID_PIN(IR_GPIO_IN_NUM),
GPIO_PIN_INTR_NEGEDGE); //设置红外输入引脚为下降沿触发中断
```

**step2) interrupt\_serv();**

```
gpio_status = GPIO_REG_READ(GPIO_STATUS_ADDRESS);
if( (gpio_status>>IR_GPIO_IN_NUM)& BIT0 ){ //获取IR_GPIO_IN_NUM状态
    ir_intr_test(); //若红外接收pin脚触发中断，进入处理函数
}
```

## 2.6. ir\_intr\_test()

ir\_intr\_test(); //红外接收中断处理，分为三步完成：

**step1) case IR\_NEC\_STATE\_IDLE:** //获取引导码，包括9ms低电平，4.5ms高电平

**step2) case IR\_NEC\_STATE\_CMD:** //获取地址码和地址反码，命令码和命令反码，当接收到32位数据时，校验cmd1与cmd2的反码是否相同，相同则认为收到数据正确，否则重新接收数据。

```
if(cnt==IR_NEC_BIT_NUM*4){ //收到32位数据
    if(nec_code_check(ir_cmd)){ //校验cmd1与cmd2的反码是否相同
        printf(" addr : %02x,\n",ir_cmd&0xff); //获取第一个字节
        printf("~addr : %02x,\n",(ir_cmd>>8)&0xff); //获取第二个字节
        printf(" cmd : %02x,\n",(ir_cmd>>16)&0xff); //获取第三个字节
```



```
printf("~cmd%02x,\n",(ir_cmd>>24)&0xff); //获取第四个字节

ir_state=IR_NEC_STATE_REPEAT; //下次进入接收重复码状态
rep_flg=0;
}else{
    ir_state=IR_NEC_STATE_IDLE; //下次进入接收引导码状态
    cnt=0;
    ir_cmd=0;
}
}
```

**Step3) case IR\_NEC\_STATE\_REPEAT:** //获取重复码