

优秀不够，你是否无可替代

知识从未如此性感。烂程序员关心的是代码,好程序员关心的是数据结构和它们之间的关系 --QQ群: 607064330 --本人
QQ:946029359 --淘宝 <https://shop411638453.taobao.com/>

随笔 - 751, 文章 - 0, 评论 - 317, 阅读 - 183万

导航

博客园
首页
新随笔
联系
订阅 8ML
管理


公告

渡我不渡她 -
Not available
00:00 / 03:41

1 渡我不渡她

2 小镇姑娘

3 PDD洪荒之力

 加入QQ群

昵称：杨奉武
园龄：5年10个月
粉丝：637
关注：1

搜索

 找找看
 谷歌搜索

我的标签

8266(88)
MQTT(50)
GPRS(33)
SDK(29)
Air202(28)
云服务器(21)
ESP8266(21)
Lua(18)
小程序(17)
STM32(16)
更多

随笔分类

Air724UG学习开发(2)
Android(22)
Android 开发(8)
C# 开发(4)
CH395Q学习开发(17)
CH579M学习开发(7)
ESP32学习开发(15)
ESP8266 AT指令开发(基于STC89C52单片机)(3)
ESP8266 AT指令开发(基于STM32)(1)
ESP8266 AT指令开发基础入门篇备份(12)
ESP8266 LUA脚本语言开发(13)

ESP8266 SDK开发: 外设篇-GPIO输出高低电平

ESP8266:SDK开发(源码见资料源码)

开发板购买链接:[开发板购买链接](#)

资料源码:<https://github.com/yangfengwu45/learn-esp8266-sdk.git>

开发软

件:https://mnifdv.cn/resource/cnblogs/Learn8266ForSDK/AiThinkerIDE_V0.5

点击加入群聊【ESP8266开发交流群】：

- 基础开源教程:ESP8266:LUA脚本开发
- 基础开源教程:ESP8266 AT指令开发(基于51单片机)
- 基础开源教程:Android学习开发
- 基础开源教程:C#学习开发
- 基础开源教程:微信小程序开发入门篇
- 需要搭配的Android, C#等基础教程如上, 各个教程正在整理。
- 1.01-准备工作-硬件说明
- 1.02-整体运行测试-APP使用SmartConfig配网绑定ESP8266,并通过MQTT远程通信控制,采集DHT11温湿度数据
- 2.01 开发环境搭建(RTOS 2.2.0)(建议只参考这篇文章搭建即可,教程以NONOS版本为主!)
- 2.01 开发环境搭建(NONOS 2.2.0).
- 2.02-外设篇-GPIO输出高低电平
- 2.03-外设篇-GPIO输入检测
- 2.04-外设篇-GPIO中断检测
- 2.05-外设篇-定时器,延时
- 2.05-外设篇-系统任务(消息队列,通知)
- 2.06-外设篇-串口
- 2.07-外设篇-PWM,呼吸灯(RTOS 2.2.0).
- 2.08-外设篇-SPI(RTOS 2.2.0).
- 2.09-外设篇-温湿度传感器-DHT11
- 2.11-外设篇-时钟芯片DS1302使用和拓展知识time.h的使用
- 2.12-外设篇-内存分布说明及Flash读写
- 3.02-网络篇-8266TCP服务器(espconn实现).(NONOS 2.2.0).
- 3.03-网络篇-8266连接路由器
- 3.04-网络篇-TCP客户端(espconn).(NONOS 2.2.0).
- 3.10-网络篇-UDP通信 - 微信小程序篇-微信小程序通过UDP实现和ESP8266局域网通信控制
- 4.01-自建MQTT服务器篇-安装MQTT服务器,ESP8266连接MQTT服务器实现通信控制
- 4.02-自建MQTT服务器篇-ESP8266配网 SmartConfig
- 4.03-自建MQTT服务器篇-APP使用SmartConfig配网绑定ESP8266,并通过MQTT远程通信控制
- 4.05-自建MQTT服务器篇-编写微信小程序连接MQTT服务器程序
- 4.10 阿里云物联网平台篇-测试MQTT调试助手和ESP8266连接阿里云物联网平台
- 4.11-阿里云物联网平台篇-ESP8266连接阿里云物联网平台使用自定义Topic实

ESP8266 Lua开发基础入门篇
备份(22)
ESP8266 SDK开发(33)
ESP8266 SDK开发基础入门篇
备份(30)
GPRS Air202 Lua开发(11)
HC32F460(华大) +
BC260Y(NB-IOT) 物联网开发
(5)
NB-IOT Air302 AT指令和Lua
脚本语言开发(25)
PLC(三菱PLC)基础入门篇(2)
STM32+Air724UG(4G模组)
物联网开发(43)
STM32+BC26/260Y物联网开
发(37)
STM32+CH395Q(以太网)物
联网开发(21)
STM32+ESP8266(ZLESP8266/
物联网开发(1)
STM32+ESP8266+AIR202/30:
远程升级方案(16)
STM32+ESP8266+AIR202/30:
终端管理方案(6)
STM32+ESP8266+Air302物
联网开发(64)
STM32+W5500+AIR202/302
基本控制方案(25)
STM32+W5500+AIR202/302
远程升级方案(6)
UCOSii操作系统(1)
W5500 学习开发(8)
编程语言C#(11)
编程语言Lua脚本语言基础入
门篇(6)
编程语言Python(1)
单片机(LPC1778)LPC1778(2)
单片机(MSP430)开发基础入门
篇(4)
单片机(STC89C51)单片机开发
板学习入门篇(3)
单片机(STM32)基础入门篇(3)
单片机(STM32)综合应用系列
(16)
电路模块使用说明(11)
感想(6)
软件安装使用: MQTT(8)
更多

最新评论

1. Re:(一)Lua脚本语言入门
楼主可以分享一下这本电子
书吗?
--戢思
2. Re:学习C语言-学习指针
学到了学到了, 很清晰的思
路, 给博主赞赞赞
--*夏日么么茶

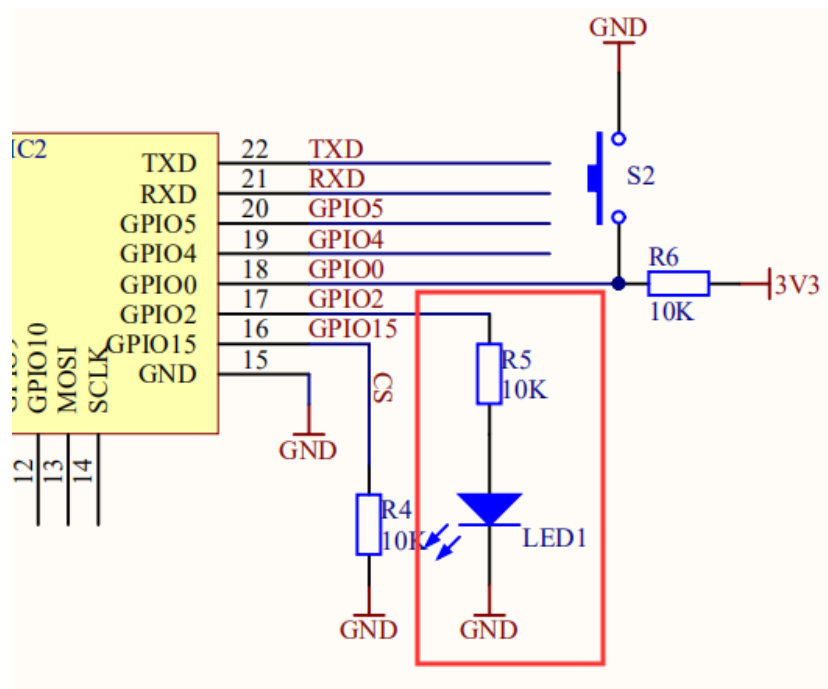
阅读排行榜

1. ESP8266使用详解(AT,LUA,
SDK)(172847)
2. 1-安装MQTT服务器(Windo
ws),并连接测试(99168)
3. ESP8266刷AT固件与node
mcu固件(64823)
4. 用ESP8266+android,制作
自己的WIFI小车(ESP8266篇)
(64354)
5. 有人WIFI模块使用详解(385
49)

- [5.11-阿里云物联网平台篇-ESP8266连接阿里云物联网平台使用物理模型Topic实现自定义数据的上报和数据下发](#)
- [4.12-阿里云物联网平台篇-ESP8266连接阿里云物联网平台使用物理模型Topic实现温湿度数据显示](#)
- [4.13-阿里云物联网平台篇-阿里云物联网平台加入规则引擎\(云产品流转\),让MQTT设备之间实现通信](#)
- [4.14-阿里云物联网平台篇-Android和ESP8266连接阿里云物联网平台,并通过云平台实现远程温湿度采集和继电器控制](#)
-
-
- [6.01-综合实战篇-C#上位机串口通信控制ESP8266\(RTOS 2.2.0\)](#)
- [6.02-综合实战篇-8266TCP服务器\(LWIP,RAW模式,PCB控制块实现\)\(RTOS 2.2.0\)与C#TCP客户端实现无线网络通信控制](#)
- [6.03-综合实战篇-8266TCP服务器\(espconn实现\)\(NONOS 2.2.0\)与Android TCP客户端实现无线网络通信控制](#)
-
- [9.01-常见问题及程序BUG修复](#)

开发板原理图

一,LED(GPIO2)



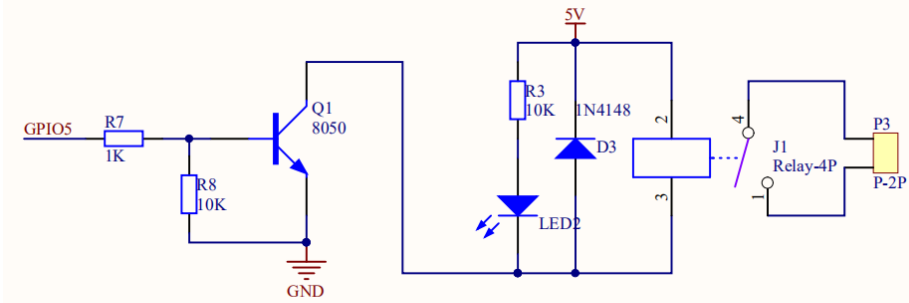
二,继电器(GPIO5)

6. (一)基于阿里云的MQTT远
程控制(Android 连接MQTT服
务器,ESP8266连接MQTT服
务器实现远程通信控制----简单
的连接通信)(35982)
7. 关于TCP和MQTT之间的转
换(33340)
8. C#中public与private与stat
ic(32611)
9. android 之TCP客户端编程
(31968)
10. android服务端+eps8266
+单片机+路由器之远程控制系
统(31338)

推荐排行榜

1. C#委托+回调详解(9)
2. 用ESP8266+android,制作
自己的WIFI小车(ESP8266篇)
(8)
3. 用ESP8266+android,制作
自己的WIFI小车(Android 软
件)(6)
4. ESP8266使用详解(AT,LUA,
SDK)(6)
5. 关于TCP和MQTT之间的转
换(5)

继电器(GPIO5)



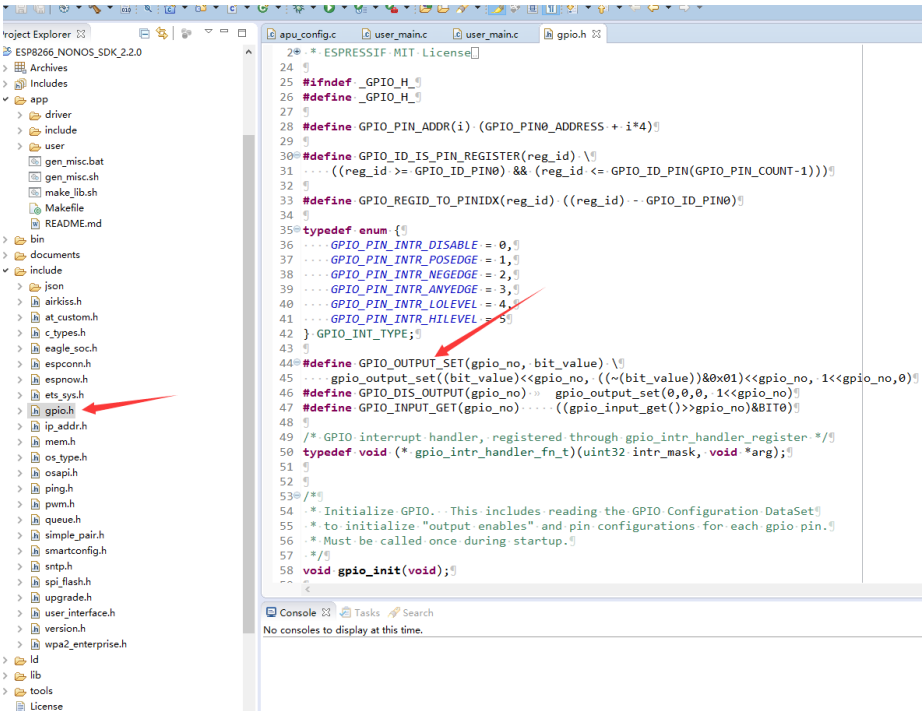
XD
XD

NONOS(2.2.0)

NONOS(2.2.0)

NONOS(2.2.0)

1.设置GPIO输出高低电平

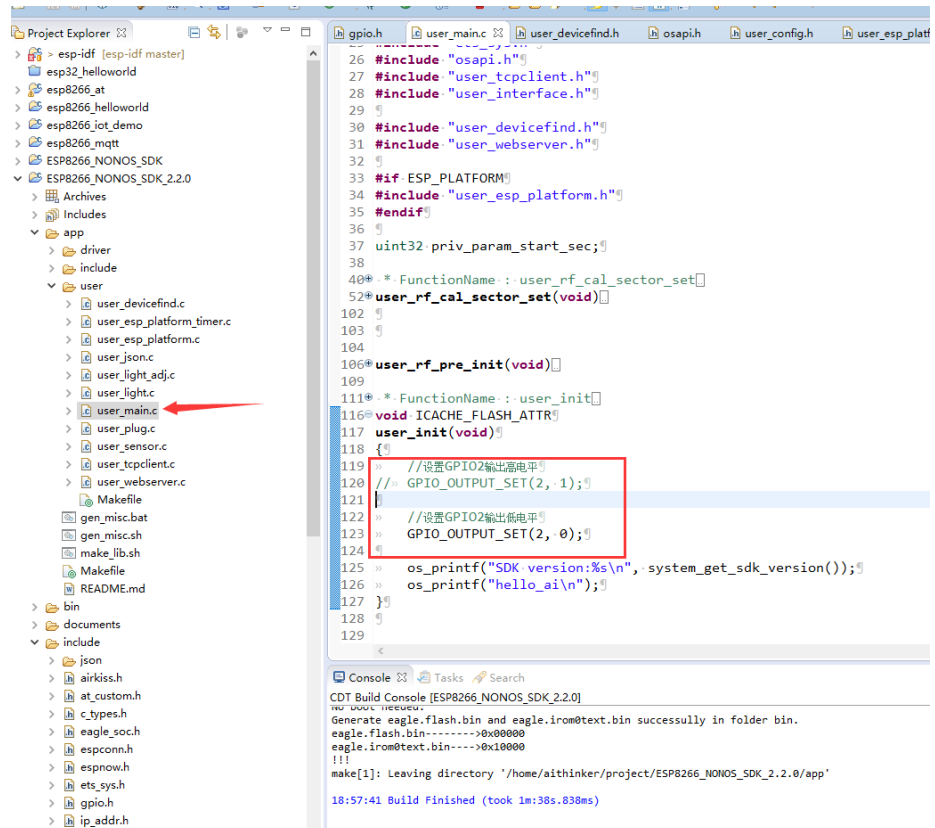


设置GPIO2输出高电平

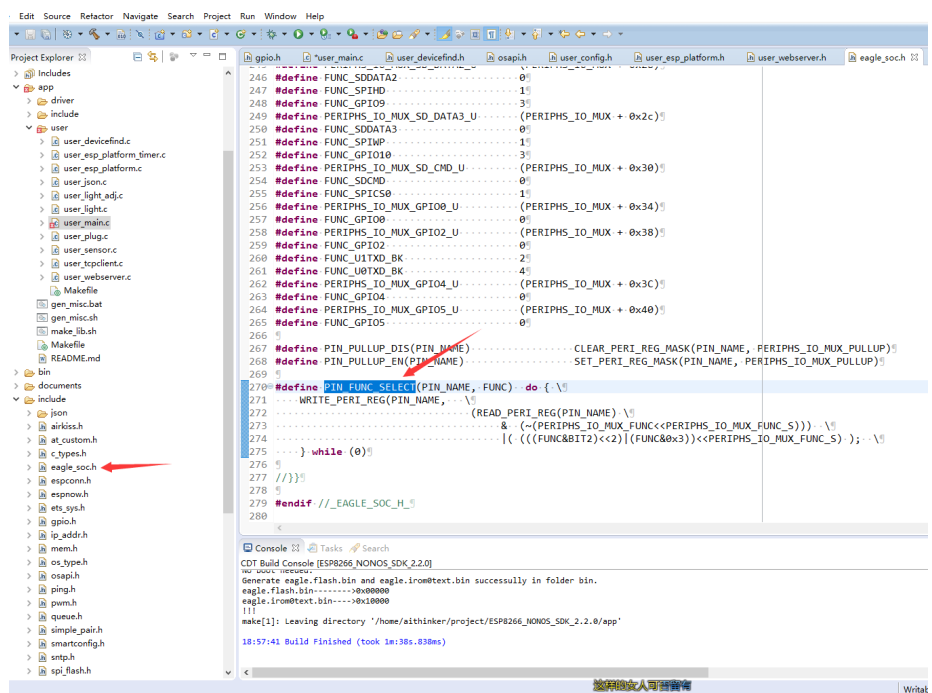
```
GPIO_OUTPUT_SET(2, 1);
```

设置GPIO2输出低电平

```
GPIO_OUTPUT_SET(2, 0);
```



不过正常流程是先设置这个引脚的功能



PIN_FUNC_SELECT(IO口的寄存器地址, 功能)

举例1:配置 GPIO2作为普通IO

PIN_FUNC_SELECT(PERIPHERS_IO_MUX_GPIO2_U
, FUNC_GPIO2);

举例2:配置 GPIO2作为串口功能

PIN_FUNC_SELECT(PERIPHERS_IO_MUX_GPIO2_U
, FUNC_UART1_TXD_BK);

说一下: FUNC_UART1_TXD_BK 这个我具体也不知道是什么意思,只知道可能是作为串口1的TX输出

举例3:配置 GPIO5作为普通IO功能

PIN_FUNC_SELECT(PERIPHERS_IO_MUX_GPIO5_U,
FUNC_GPIO5);

注: 这一个是默认的功能

```

110 #define PERIPHS_IO_MUX_SD_DATA3_U ..... (PERIPHS_IO_MUX + 0x2c) //该IO口的寄存器地址
111 #define FUNC_SDDATA3 ..... 0 //
112 #define FUNC_SPIWP ..... 1 //
113 #define FUNC_GPIO10 ..... 3 //
114 #define FUNC_HSPIWP ..... 4 //
115
116 #define PERIPHS_IO_MUX_SD_CMD_U ..... (PERIPHS_IO_MUX + 0x30) //该IO口的寄存器地址
117 #define FUNC_SDCMD ..... 0 //
118 #define FUNC_SPICS0 ..... 1 //
119 #define FUNC_GPIO11 ..... 3 //
120 #define U1RTS ..... 4 //
121 #define UART1_RTS ..... 4 //
122
123 #define PERIPHS_IO_MUX_GPIO0_U ..... (PERIPHS_IO_MUX + 0x34) //该IO口的寄存器地址
124 #define FUNC_GPIO0 ..... 0 //默认作为普通GPIO //
125 #define FUNC_SPICS2 ..... 1 //SPI的CS引脚
126 #define FUNC_CLK_OUT ..... 4 //时钟输出引脚
127
128 #define PERIPHS_IO_MUX_GPIO2_U ..... (PERIPHS_IO_MUX + 0x38) //该IO口的寄存器地址
129 #define FUNC_GPIO2 ..... 0 //默认作为普通GPIO //
130 #define FUNC_I2SO_WS ..... 1 //作为IIC的..不知道
131 #define FUNC_U1TXD_BK ..... 2 //作为串口1的输出吗??
132 #define FUNC_UART1_TXD_BK ..... 2 //作为串口1的输出
133 #define FUNC_U0TXD_BK ..... 4 //作为串口0的输出吗??
134 #define FUNC_UART0_TXD_BK ..... 4 //作为串口0的输出
135
136 #define PERIPHS_IO_MUX_GPIO4_U ..... (PERIPHS_IO_MUX + 0x3C) //该IO口的寄存器地址
137 #define FUNC_GPIO4 ..... 0 //默认作为普通GPIO //
138 #define FUNC_CLK_XTAL ..... 1 //晶振的时钟引脚
139
140 #define PERIPHS_IO_MUX_GPIO5_U ..... (PERIPHS_IO_MUX + 0x40) //该IO口的寄存器地址
141 #define FUNC_GPIO5 ..... 0 //默认作为普通GPIO //
142 #define FUNC_CLK_RTC ..... 1 //RTC的时钟引脚
143
144 #define PIN_PULLUP_DIS(PIN_NAME) ..... CLEAR_PERI_REG_MASK(PIN_NAME, PERIPHS_IO_MUX
145 #define PIN_PULLUP_EN(PIN_NAME) ..... SET_PERI_REG_MASK(PIN_NAME, PERIPHS_IO_MUX_P
146 //此函数用于配置引脚作为哪个功能
147 #define PIN_FUNC_SELECT(PIN_NAME, FUNC) do { \
148 ..... CLEAR_PERI_REG_MASK(PIN_NAME, (PERIPHS_IO_MUX_FUNC << PERIPHS_IO_MUX_FUNC_S))
149 ..... SET_PERI_REG_MASK(PIN_NAME, (((FUNC & BIT2) << 2) | (FUNC & 0x3)) << PERIPHS_
150 ..... } while (0)
151
152 #endif // PIN_MUX_H

```

注:在配置完PIN_FUNC_SELECT 以后,当前引脚为高阻态,既不是高电平也不是低电平.

测试(GPIO2输出高电平点亮LED)

```
#include "gpio.h"
```

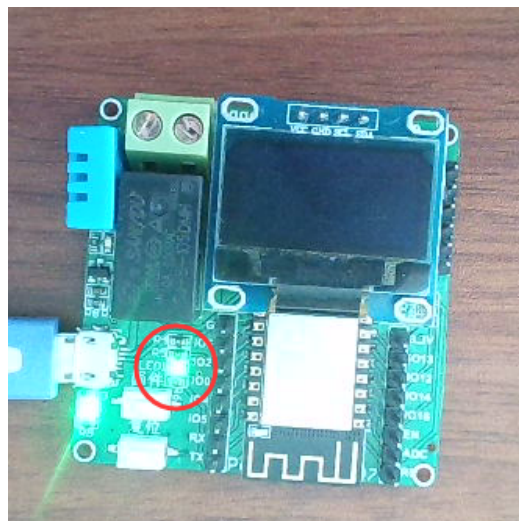
```
PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO2_U
, FUNC_GPIO2);
```

```
GPIO_OUTPUT_SET(2, 1); //设置GPIO2输出高电平
```

```

user_main.c  gpio.h  pin_mux_register.h  eagle_soc.h  gpio.c  gpio_register.h
20  /* ESPRESSIF MIT License
24
25  #include "esp_common.h"
26  #include "gpio.h"
27  /*****
28  * FunctionName: user_rf_cal_sector_set
29  * Description: SDK just reversed 4 sectors, used for rf init data and paramter
30  * ..... We add this function to force users to set rf cal sector, since
31  * ..... we don't know which sector is free in user's application.
32  * ..... sector map for last several sectors: ABCCC
33  * ..... A: rf cal
34  * ..... B: rf init data
35  * ..... C: sdk parameters
36  * Parameters: none
37  * Returns: rf cal sector
38  *****/
39  uint32 user_rf_cal_sector_set(void)
40  {
41  /*****
42  * FunctionName: user_init
43  * Description: entry of user application, init user function here
44  * Parameters: none
45  * Returns: none
46  *****/
47  void user_init(void)
48  {
49  uart_init_new();
50  printf("SDK version:%s\n", system_get_sdk_version());
51
52  printf("Ai-Thinker Technology Co.,Ltd.\r\n%s %s\r\n", __DATE__, __TIME__);
53
54  PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO2_U, FUNC_GPIO2);
55  GPIO_OUTPUT_SET(2, 1); //设置GPIO2输出高电平
56  }

```



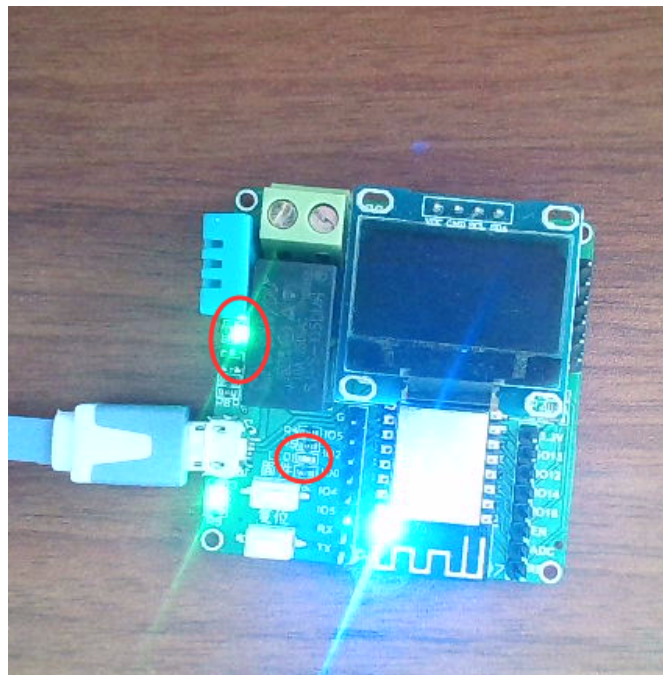
测试(GPIO2输出低电平,GPIO5输出高电平控制继电器吸合)

```
#include "gpio.h"
```

```
GPIO_OUTPUT_SET(2, 0); //设置GPIO2输出低电平
```


GPIO_OUTPUT_SET(5, 1); //设置GPIO5输出高电平

```
ject Run Window Help
user_main.c gpio.h pin_mux_register.h eagle_soc.h gpio.c gpio_register.h
24 * ESPRESSIF MIT License
25 #include "esp_common.h"
26 #include "gpio.h"
27
28 * FunctionName: user_rf_cal_sector_set
29 * Description: SDK just reversed 4 sectors, used for rf init data and paramters.
30 * We add this function to force users to set rf cal sector, since
31 * we don't know which sector is free in user's application.
32 * sector map for last several sectors: ABCCC
33 * A: rf cal
34 * B: rf init data
35 * C: sdk parameters
36 * Parameters: none
37 * Returns: rf cal sector
38
39 uint32 user_rf_cal_sector_set(void)
40
41 * FunctionName: user_init
42 * Description: entry of user application, init user function here
43 * Parameters: none
44 * Returns: none
45
46
47 void user_init(void)
48 {
49     uart_init_new();
50     printf("SDK version:%s\n", system_get_sdk_version());
51     printf("Ai-Thinker Technology Co., Ltd.\r\n%s %s\r\n", __DATE__, __TIME__);
52
53     PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO2_U, FUNC_GPIO2);
54     GPIO_OUTPUT_SET(2, 0); //设置GPIO2输出低电平
55
56     PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO5_U, FUNC_GPIO5);
57     GPIO_OUTPUT_SET(5, 1); //设置GPIO5输出高电平
58 }
59
60
61
62
```



如果控制有SPI功能的引脚,需要写上一句话,否则不能控制

WRITE_PERI_REG(PERIPHS_IO_MUX, 0x105);

```

/* Parameters: none
 * Returns: none
 */
void ICACHE_FLASH_ATTR
user_init(void)
{
    uart_init_2(BIT_RATE_115200,BIT_RATE_115200);

    PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO0_U, FUNC_GPIO0); //设置GPIO0普通模式
    GPIO_OUTPUT_SET(0, 1); //设置GPIO0输出高电平

    PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO2_U, FUNC_GPIO2); //设置GPIO2普通模式
    GPIO_OUTPUT_SET(2, 1); //设置GPIO2输出高电平

    WRITE_PERI_REG(PERIPHS_IO_MUX, 0x105);

    PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO4_U, FUNC_GPIO4); //设置GPIO4普通模式
    GPIO_OUTPUT_SET(4, 1);

    PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO5_U, FUNC_GPIO5); //设置GPIO5普通模式
    GPIO_OUTPUT_SET(5, 1);

    PIN_FUNC_SELECT(PERIPHS_IO_MUX_MTDI_U, FUNC_GPIO12); //设置GPIO12普通模式
    GPIO_OUTPUT_SET(12, 1);

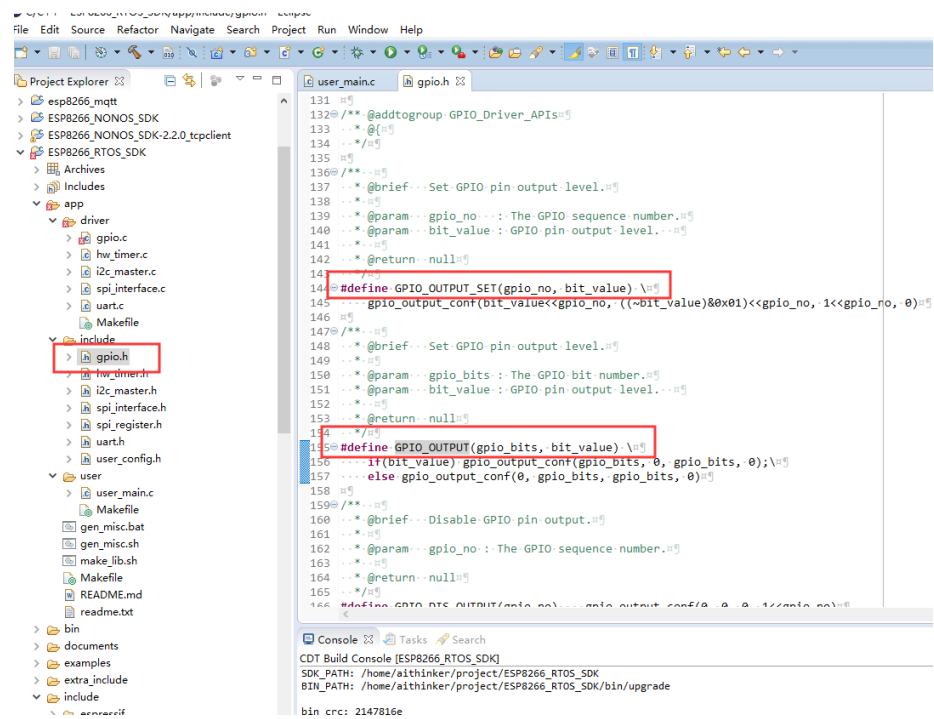
    PIN_FUNC_SELECT(PERIPHS_IO_MUX_MTCK_U, FUNC_GPIO13); //设置GPIO13普通模式
    GPIO_OUTPUT_SET(13, 1);
}
```

RTOS(2.2.0)

RTOS(2.2.0)

RTOS(2.2.0)

官方提供了两个函数



设置GPIO2输出高电平

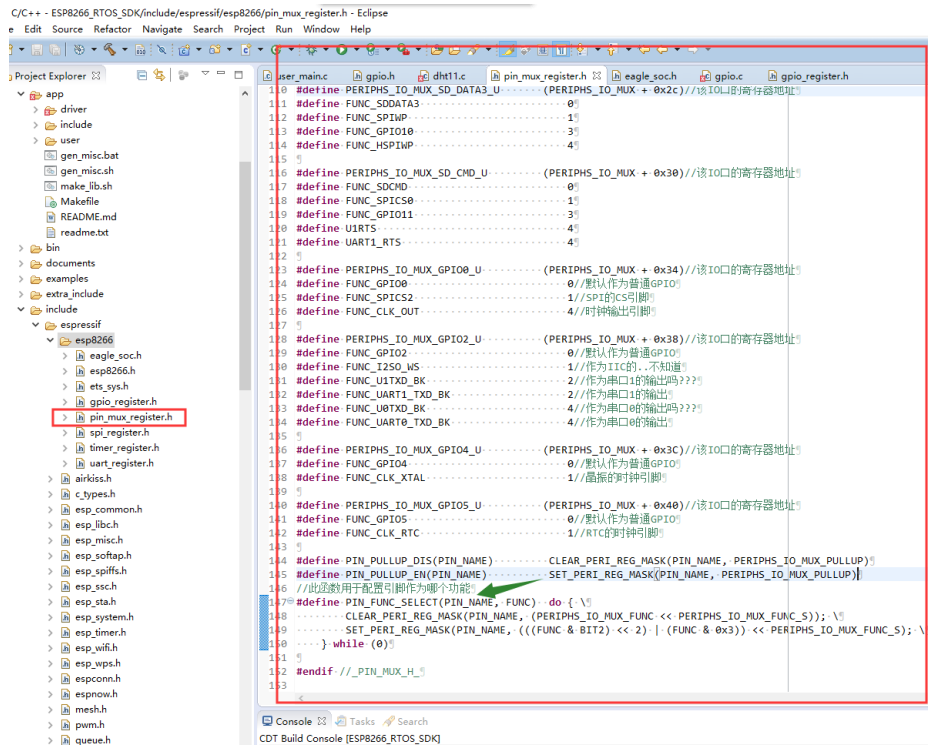
设置GPIO2输出低电平

GPIO_OUTPUT(gpio bits, bit value)

设置GPIO2和GPIO5输出低电平

GPIO_OUTPUT(BIT2 | BIT5, 0);

不过正常流程是先设置这个引脚的功能



PIN_FUNC_SELECT(IO口的寄存器地址，功能)

举例1:配置 GPIO2作为普通IO

```
PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO2_U, FUNC_GPIO2);
```

举例2:配置 GPIO2作为串口功能

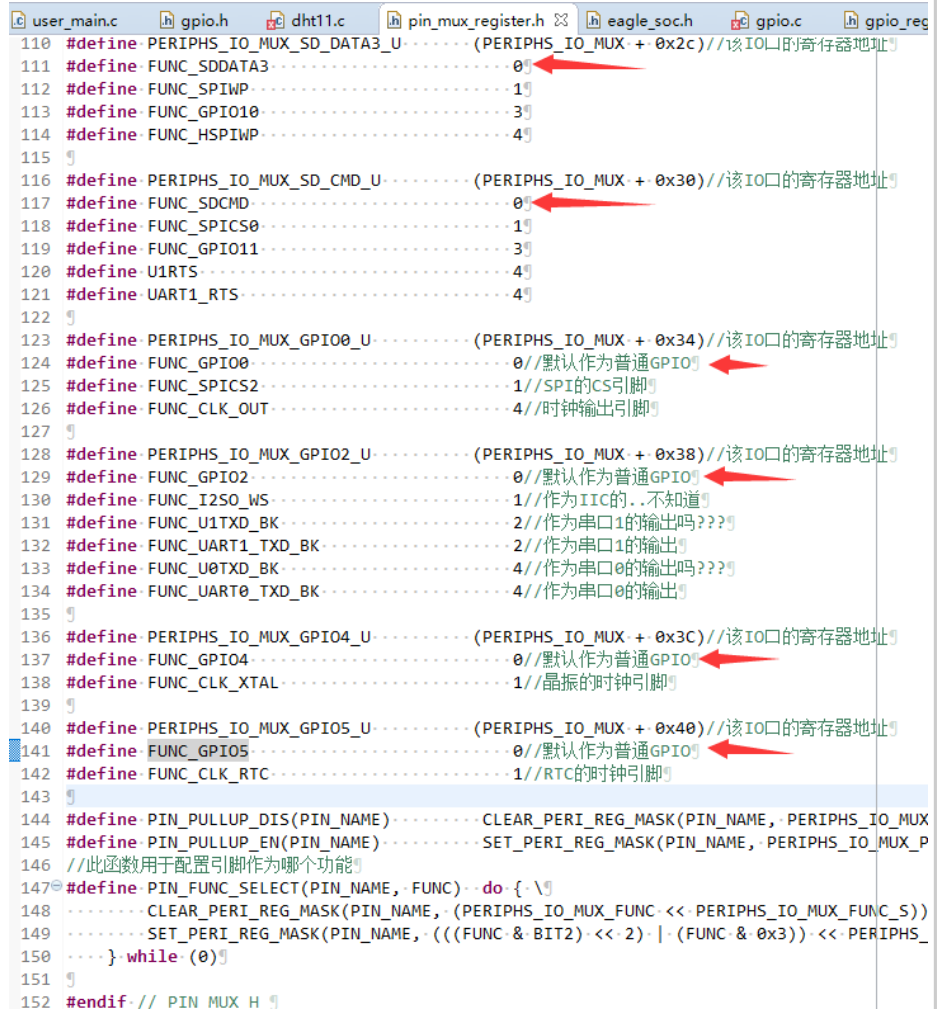
```
PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO2_U, FUNC_UART1_TXD_BK);
```

说一下: FUNC_UART1_TXD_BK 这个我具体也不知道是什么意思,只知道可能是作为串口1的TX输出

举例3:配置 GPIO5作为普通IO功能

```
PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO5_U,  
FUNC_GPIO5);
```

注: 这一个是默认的功能



```
110 #define PERIPHS_IO_MUX_SD_DATA3_U ..... (PERIPHS_IO_MUX + 0x2c) //该IO口的寄存器地址
111 #define FUNC_SDDATA3 ..... 0
112 #define FUNC_SDIWP ..... 1
113 #define FUNC_GPIO10 ..... 3
114 #define FUNC_HSPIWP ..... 4
115
116 #define PERIPHS_IO_MUX_SD_CMD_U ..... (PERIPHS_IO_MUX + 0x30) //该IO口的寄存器地址
117 #define FUNC_SDCMD ..... 0
118 #define FUNC_SPIC0 ..... 1
119 #define FUNC_GPIO11 ..... 3
120 #define U1RTS ..... 4
121 #define UART1_RTS ..... 4
122
123 #define PERIPHS_IO_MUX_GPIO0_U ..... (PERIPHS_IO_MUX + 0x34) //该IO口的寄存器地址
124 #define FUNC_GPIO0 ..... 0 //默认作为普通GPIO
125 #define FUNC_SPIC2 ..... 1 //SPI的CS引脚
126 #define FUNC_CLK_OUT ..... 4 //时钟输出引脚
127
128 #define PERIPHS_IO_MUX_GPIO2_U ..... (PERIPHS_IO_MUX + 0x38) //该IO口的寄存器地址
129 #define FUNC_GPIO2 ..... 0 //默认作为普通GPIO
130 #define FUNC_I2SO_WS ..... 1 //作为IIC的..不知道
131 #define FUNC_U1TXD_BK ..... 2 //作为串口1的输出吗???
132 #define FUNC_UART1_TXD_BK ..... 2 //作为串口1的输出
133 #define FUNC_U0TXD_BK ..... 4 //作为串口0的输出吗???
134 #define FUNC_UART0_TXD_BK ..... 4 //作为串口0的输出
135
136 #define PERIPHS_IO_MUX_GPIO4_U ..... (PERIPHS_IO_MUX + 0x3C) //该IO口的寄存器地址
137 #define FUNC_GPIO4 ..... 0 //默认作为普通GPIO
138 #define FUNC_CLK_XTAL ..... 1 //晶振的时钟引脚
139
140 #define PERIPHS_IO_MUX_GPIO5_U ..... (PERIPHS_IO_MUX + 0x40) //该IO口的寄存器地址
141 #define FUNC_GPIO5 ..... 0 //默认作为普通GPIO
142 #define FUNC_CLK_RTC ..... 1 //RTC的时钟引脚
143
144 #define PIN_PULLUP_DIS(PIN_NAME) ..... CLEAR_PERI_REG_MASK(PIN_NAME, PERIPHS_IO_MUX
145 #define PIN_PULLUP_EN(PIN_NAME) ..... SET_PERI_REG_MASK(PIN_NAME, PERIPHS_IO_MUX_P
146 //此函数用于配置引脚作为哪个功能
147 #define PIN_FUNC_SELECT(PIN_NAME, FUNC) do { \
148 ..... CLEAR_PERI_REG_MASK(PIN_NAME, (PERIPHS_IO_MUX_FUNC << PERIPHS_IO_MUX_FUNC_S))
149 ..... SET_PERI_REG_MASK(PIN_NAME, (((FUNC & BIT2) << 2) | (FUNC & 0x3)) << PERIPHS_
150 ..... } while (0)
151
152 #endif // PIN_MUX_H
```

注:在配置完PIN_FUNC_SELECT 以后,当前引脚为高阻态,既不是高电平也不是低电平.

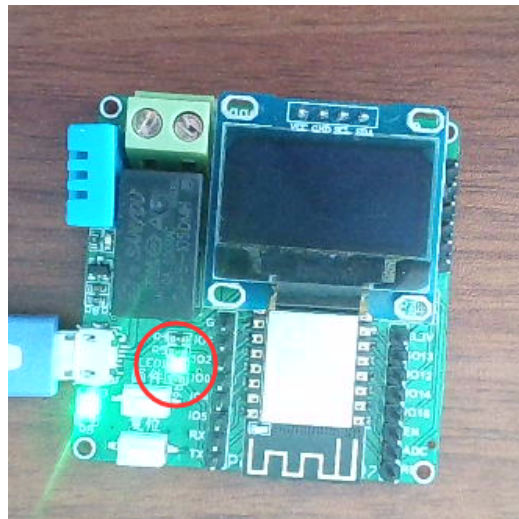
测试(GPIO2输出高电平点亮LED)

```
#include "gpio.h"
```

```
PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO2_U  
, FUNC_GPIO2);
```

GPIO_OUTPUT_SET(2, 1);//设置GPIO2输出高电平

```
user_main.c gpio.h pin_mux_register.h eagle_soc.h gpio.c gpio_register.h
24 /* * ESPRSSIF MIT License */
25 #include "esp_common.h"
26 #include "gpio.h"
27
28 /* * FunctionName: user_rf_cal_sector_set
29 * Description: SDK just reversed 4 sectors, used for rf init data and parameter
30 * ..... We add this function to force users to set rf cal sector, since
31 * ..... we don't know which sector is free in user's application.
32 * ..... sector map for last several sectors: ABCCC
33 * ..... A: rf cal
34 * ..... B: rf init data
35 * ..... C: sdk parameters
36 * Parameters: none
37 * Returns: rf cal sector
38 */
39 uint32 user_rf_cal_sector_set(void)
40 {
41 }
42
43 /* * FunctionName: user_init
44 * Description: entry of user application, init user function here
45 * Parameters: none
46 * Returns: none
47 */
48 void user_init(void)
49 {
50     uart_init_new();
51     printf("SDK version:%s\n", system_get_sdk_version());
52     printf("Ai-Thinker Technology Co., Ltd.\n%s %s\n", __DATE__, __TIME__);
53     PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO2_U, FUNC_GPIO2);
54     GPIO_OUTPUT_SET(2, 1);//设置GPIO2输出高电平
55 }
```



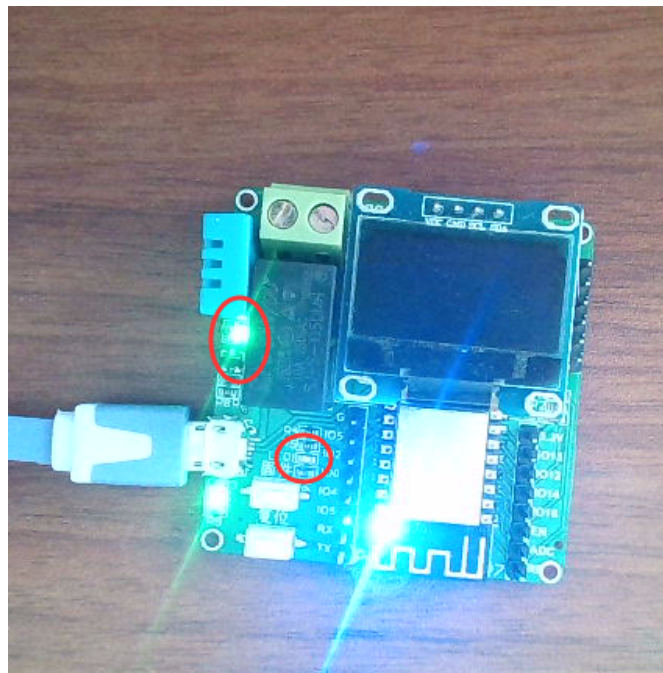
测试(GPIO2输出低电平,GPIO5输出高电平控制继电器吸合)

```
#include "gpio.h"
```

```
GPIO_OUTPUT_SET(2, 0);//设置GPIO2输出低电平
```

GPIO_OUTPUT_SET(5, 1); //设置GPIO5输出高电平

```
ject Run Window Help
user_main.c gpio.h pin_mux_register.h eagle_soc.h gpio.c gpio_register.h
24 * ESPRESSIF MIT License
25 #include "esp_common.h"
26 #include "gpio.h"
27
28 * FunctionName: user_rf_cal_sector_set
29 * Description: SDK just reversed 4 sectors, used for rf init data and paramters.
30 * We add this function to force users to set rf cal sector, since
31 * we don't know which sector is free in user's application.
32 * sector map for last several sectors: ABCCC
33 * A: rf cal
34 * B: rf init data
35 * C: sdk parameters
36 * Parameters: none
37 * Returns: rf cal sector
38
39 uint32 user_rf_cal_sector_set(void)
40
41 * FunctionName: user_init
42 * Description: entry of user application, init user function here
43 * Parameters: none
44 * Returns: none
45
46 void user_init(void)
47 {
48     uart_init_new();
49     printf("SDK version:%s\n", system_get_sdk_version());
50     printf("Ai-Thinker Technology Co., Ltd.\r\n%s %s\r\n", __DATE__, __TIME__);
51
52     PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO2_U, FUNC_GPIO2);
53     GPIO_OUTPUT_SET(2, 0); //设置GPIO2输出低电平
54
55     PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO5_U, FUNC_GPIO5);
56     GPIO_OUTPUT_SET(5, 1); //设置GPIO5输出高电平
57 }
58
59
60
61
62
```



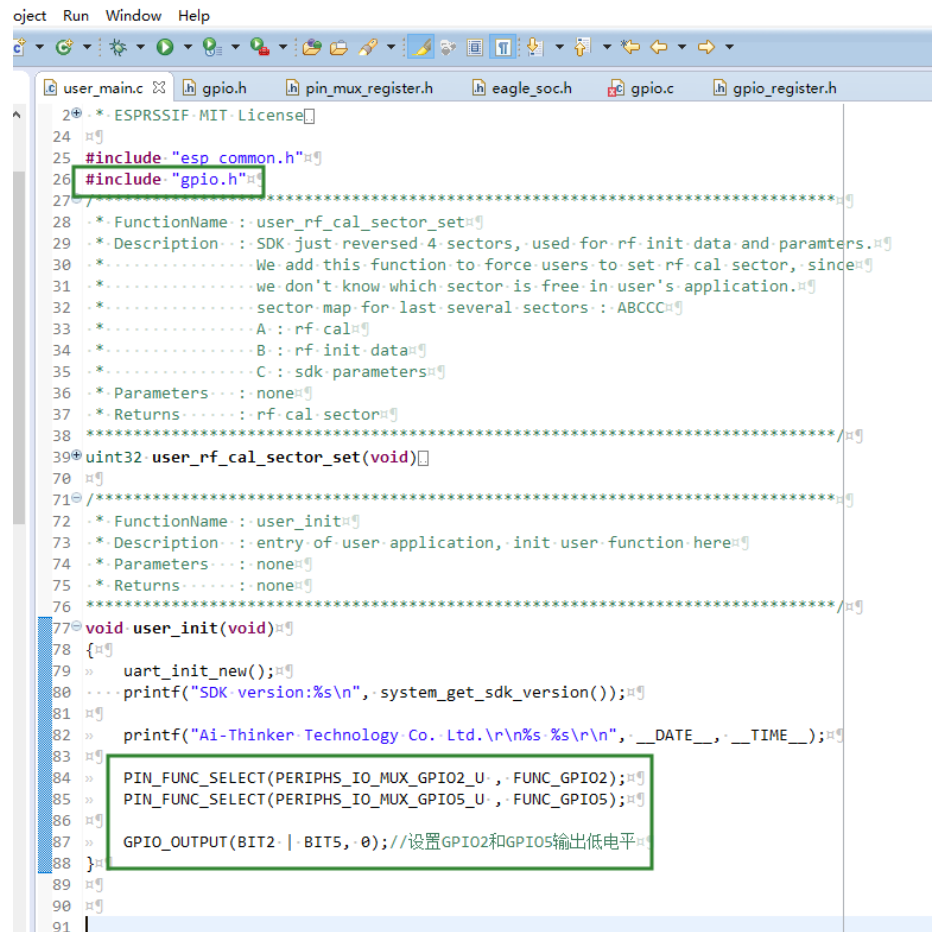
测试(GPIO2输出低电平,GPIO5输出低电平)

```
#include "gpio.h"
```

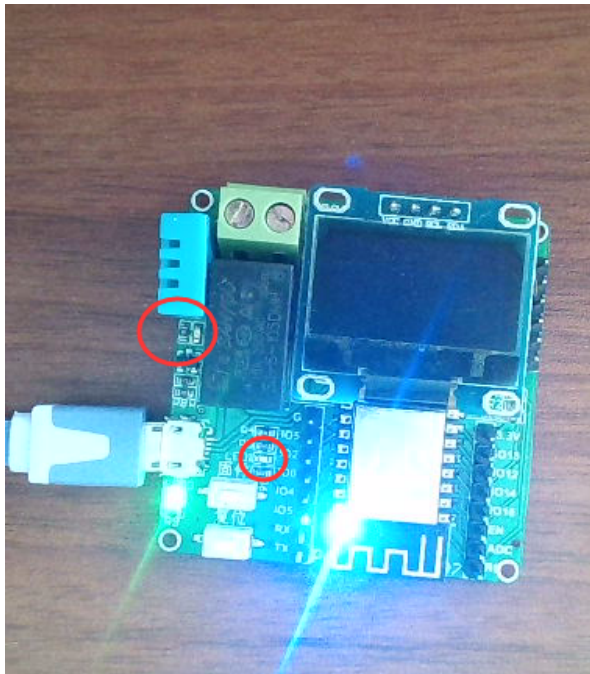
```
PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO2_U ,  
FUNC_GPIO2);
```

```
PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO5_U ,  
FUNC_GPIO5);
```

```
GPIO_OUTPUT(BIT2 | BIT5, 0); //设置GPIO2和GPIO5  
输出低电平
```



```
object Run Window Help
user_main.c gpio.h pin_mux_register.h eagle_soc.h gpio.c gpio_register.h
24 * ESPRESSIF MIT License
25 #include "esp_common.h"
26 #include "gpio.h"
27
28 /* FunctionName : user_rf_cal_sector_set
29 * Description : SDK just reversed 4 sectors, used for rf init data and paramters.
30 * ..... We add this function to force users to set rf cal sector, since
31 * ..... we don't know which sector is free in user's application.
32 * ..... sector map for last several sectors : ABCC
33 * ..... A : rf cal
34 * ..... B : rf init data
35 * ..... C : sdk parameters
36 * Parameters : none
37 * Returns : rf cal sector
38 */
39 uint32 user_rf_cal_sector_set(void)
40 {
41 }
42
43 /* FunctionName : user_init
44 * Description : entry of user application, init user function here
45 * Parameters : none
46 * Returns : none
47 */
48 void user_init(void)
49 {
50     uart_init_new();
51     printf("SDK version:%s\n", system_get_sdk_version());
52     printf("Ai-Thinker Technology Co., Ltd.\r\n%s %s\r\n", __DATE__, __TIME__);
53
54     PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO2_U, FUNC_GPIO2);
55     PIN_FUNC_SELECT(PERIPHS_IO_MUX_GPIO5_U, FUNC_GPIO5);
56
57     GPIO_OUTPUT(BIT2 | BIT5, 0); //设置GPIO2和GPIO5输出低电平
58 }
59
60
61
```



分类: [ESP8266 SDK开发](#)

好文要顶

关注我

收藏该文



杨奉武

关注 - 1

粉丝 - 637

1

0

« 上一篇: [W5500 学习开发: W5500学习说明](#)

» 下一篇: [ESP8266 SDK开发: 外设篇-GPIO输入检测](#)

posted on 2019-12-04 15:02 杨奉武 阅读(4157) 评论(0) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

发表评论

编辑 预览

B

支持 Markdown

自动补全

提交评论 退出

[Ctrl+Enter快捷键提交]

【推荐】百度智能云2021普惠上云节：新用户首购云服务器低至0.7折

【推荐】阿里云云大使特惠：新用户购ECS服务器1核2G最低价87元/年

【推荐】大型组态、工控、仿真、CAD\GIS 50万行VC++源码免费下载!

编辑推荐：

- C# 10 完整特性介绍
- 不是技术也能看懂云原生
- 记一次接口慢查排查
- 一个故事看懂HTTPS
- 人人都能看懂系列：分布式系统改造方案之数据篇



最新新闻：

- 你还抢购华为吗？门店可能没有存货了
- 字节新消费版图大起底：投资自营双管齐下
- 上市破发、资金受困，理想“勇争第一”空成口号
- 百度二季度财报点评：以更高维的ESG识别其价值
- 锂电专利战争：欧美、日韩围剿，中国换道超车
- » 更多新闻...

历史上的今天：

2016-12-04 Permission is only granted to system apps
2016-12-04 Duplicate id @+id/imageView, already defined earlier in this layout,android
2016-12-04 Unexpected namespace prefix "xmlns" found for tag Linear Layout

Powered by:

博客园

Copyright © 2021 杨奉武

Powered by .NET 5.0 on Kubernetes



单片机,物联网,上位机,...

扫一扫二维码, 入群聊。