

优秀不够，你是否无可替代

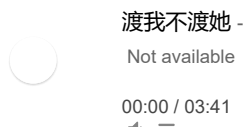
知识从未如此性感。烂程序员关心的是代码,好程序员关心的是数据结构和它们之间的关系 --QQ群: 607064330 --本人
QQ:946029359 --淘宝 <https://shop411638453.taobao.com/>

随笔 - 746, 文章 - 0, 评论 - 315, 阅读 - 182万

导航

博客园
首页
新随笔
联系
订阅 
管理

公告

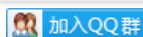


渡我不渡她 -

Not available

00:00 / 03:41

- 1 渡我不渡她
- 2 小镇姑娘
- 3 PDD洪荒之力



昵称：杨奉武
园龄：5年10个月
粉丝：633
关注：1

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

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(13)
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 LUA开发基础入门篇
备份(22)

103-ESP32_SDK开发-硬件定时器timer

<p><iframe name="ifd" src="https://mnifdv.cn/resource/cnblogs/LearnESP32" frameborder="0" scrolling="auto" width="100%" height="1500"></iframe></p>

开源ESP32开发(源码见资料源码)

测试板链接:[ESP32测试板链接](#)

资料源码:<https://github.com/yangfengwu45/learn-esp32.git>

【点击加入乐鑫WiFi模组开发交流群】(群号
822685419)https://jq.qq.com/?_wv=1027&k=fXgd3UOo

python虚拟机: [python-3.8.4-amd64.exe](#)

ESP-IDF工具安装器: [esp-idf-tools-setup-2.3.exe](#)

- 基础开源教程:ESP32开发(arduino)
 - 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。

- [000-ESP32开发板使用说明](#)
- [ESP32_SDK开发](#)
- [001-开发环境搭建\(Windows+VSCode\)](#)
- [002-测试网络摄像头\(OV2640\),实现远程视频监控\(花生壳http映射\)](#)
- [003-学习ESP32资料说明](#)
- [004-新建工程模板和创建新的文件](#)
- [005-新建工程补充-通过官方示例创建工程](#)
- [006-关于操作系统-任务,任务堆栈空间,任务的挂起,恢复,删除](#)
- [007-使用缓存管理传递数据](#)
- 基本外设-----
- [101-ESP32管脚说明](#)
- [102-GPIO](#)
- [103-硬件定时器timer](#)

ESP8266 SDK开发(32)
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)
软件安装使用: OpenResty(6)
更多

最新评论

1. Re:单片机模块化程序: 看看是不是你想要的按键处理视频不见了
--伊森亨特
2. Re:C#开发: 通信篇-TCP客户端
感谢分享，直接就用上了
--Zfen

阅读排行榜

1. ESP8266使用详解(AT,LUA,SDK)(172759)
2. 1-安装MQTT服务器(Windows),并连接测试(98869)
3. ESP8266刷AT固件与node mcu固件(64670)
4. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(64224)
5. 有人WIFI模块使用详解(38510)

说明

硬件定时器有两组,0和1. 然后每一组都有两个定时器,0和1.

所以共有四个定时器

什么是定时器?定时器具体是怎么到了时间进去中断的?

定时器里面是啥?就是个计数器.定时器的时钟,就是计数器的时钟.假设计数器的时钟是1Hz

假设设置了计数器计数到1的时候就进入中断,那么就是每隔1S进入中断了.

假设计数器的时钟是80MHz,我想每隔1ms进入一次中断,我应该设置计数值是多少呢???

计数器每记一次需要 $1/80000000$ 秒 也就是 $1/80000$ 毫秒

要记到80000次才到1ms

想定时多少毫秒,设置初值可以写成 $X*(80000000/1000)$ X就是要定时的ms数

假设时钟分频了8,那么现在就是10MHz

我想每隔1ms进入一次中断,我应该设置计数值是多少呢???

计数器记一次是 $1/100000000$ 秒 也就是 $1/10000$ 毫秒

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)(35927)
7. 关于TCP和MQTT之间的转换(33211)
8. C#中public与private与static(32422)
9. android之TCP客户端编程(31921)
10. android服务端+esp8266+单片机+路由器之远程控制系统(31321)

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)

需要多少个 1/10000 毫秒 才能到1ms呢? 10000次

想定时多少毫秒,设置初值可以写成 $X \times (80000000 / \text{分频系数} / 1000)$ X就是要定时的ms数

想定时多少微秒,设置初值可以写成 $X \times (80000000 / \text{分频系数})$ X就是要定时的us数

使用定时器分组0的0号定时器产生1S中断

```
55 void app_main(void)
56 {
57     gpio_init();//初始化gpio
58
59     /*设置定时器初始化参数*/
60     timer_config_t config = {
61         .divider = 8, //分频系数[2-65535]
62         .counter_dir = TIMER_COUNT_UP, //计数方式是向上计数
63         .counter_en = TIMER_PAUSE, //调用timer_init函数以后不启动计数,调用timer_start时才开始计数
64         .alarm_en = TIMER_ALARM_EN, //到达计数值启动报警(计数值溢出,进入中断)
65         .auto_reload = 1, //自动重新装载初值
66     };
67     /*初始化定时器;TIMER_GROUP_0(定时器分组0); TIMER_0(分组0中的0号定时器)*/
68     timer_init(TIMER_GROUP_0, TIMER_0, &config);
69
70     /*设置定时器预装值,0*/
71     timer_set_counter_value(TIMER_GROUP_0, TIMER_0, 0x00000000ULL);
72     /*设置报警阈值*/ // 1000[定时1000ms]*(TIMER_BASE_CLK[定时器时钟]/8[分频系数])/1000[想延时ms级别所以除以1000],
73     timer_set_alarm_value(TIMER_GROUP_0, TIMER_0, 1000*(TIMER_BASE_CLK/8/1000));
74     /*使能定时器中断*/
75     timer_enable_intr(TIMER_GROUP_0, TIMER_0);
76     /*注册定时器中断函数*/
77     timer_isr_register(TIMER_GROUP_0, TIMER_0,
78         timer_group0_isr, //定时器回调函数
79         (void *) TIMER_0, //传递给定时器回调函数的参数
80         ESP_INTR_FLAG_IRAM, //把中断放到 IRAM 中
81         NULL //调用成功以后返回中断函数的地址,一般用不到
82     );
83     /*启动定时器*/
84     timer_start(TIMER_GROUP_0, TIMER_0);
85 }
86
87
```

```
/*定时器中断函数
因为设置中断的时候,中断设置的 ESP_INTR_FLAG_IRAM 所以中断函数需要加 IRAM_ATTR
*/
void IRAM_ATTR timer_group0_isr(void *para)
{
    //获取定时器分组0中的哪一个定时器产生了中断
    uint32_t timer_intr = timer_group_get_intr_status_in_isr(TIMER_GROUP_0);
    if (timer_intr & TIMER_INTR_T0) { //定时器0分组的0号定时器产生中断
        /*清除中断*/
        timer_group_clr_intr_status_in_isr(TIMER_GROUP_0, TIMER_0);
        /*重新使能定时器中断*/
        timer_group_enable_alarm_in_isr(TIMER_GROUP_0, TIMER_0);
    }

    /*设置gpio输出高低电平*/
    gpio_set_level(gpio_pin, 1-gpio_get_level(gpio_pin));
}
```



```
#include <stdio.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/queue.h"
```

```

#include "driver/gpio.h"
#include "driver/timer.h"

#define gpio_pin 25

/*定时器中断函数
因为设置中断的时候,中断设置的 ESP_INTR_FLAG_IRAM 所以中断函数需要加 IRAM_ATTR
*/
void IRAM_ATTR timer_group0_isr(void *para)
{
    //获取定时器分组0中的哪一个定时器产生了中断
    uint32_t timer_intr = timer_group_get_intr_status_in_isr(TIMER_GROUP_0);
    if (timer_intr & TIMER_INTR_T0) { //定时器0分组的0号定时器产生中断
        /*清除中断*/
        timer_group_clr_intr_status_in_isr(TIMER_GROUP_0, TIMER_0);
        /*重新使能定时器中断*/
        timer_group_enable_alarm_in_isr(TIMER_GROUP_0, TIMER_0);
    }

    /*设置gpio输出高低电平*/
    gpio_set_level(gpio_pin, 1-gpio_get_level(gpio_pin));
}

void gpio_init(void) {
    gpio_config_t io_conf;
    //禁止中断
    io_conf.intr_type = GPIO_PIN_INTR_DISABLE;
    //输入输出模式
    io_conf.mode = GPIO_MODE_INPUT_OUTPUT;
    //配置要设置的引脚
    io_conf.pin_bit_mask = (unsigned long long)1<<gpio_pin;
    //禁止下拉
    io_conf.pull_down_en = 0;
    //禁止上拉
    io_conf.pull_up_en = 0;
    //配置gpio(不设置上下拉默认输出低电平)
    gpio_config(&io_conf);
}

void app_main(void)
{
    gpio_init();//初始化gpio

    /*设置定时器初始化参数*/
    timer_config_t config = {
        .divider = 8, //分频系数[2-65535]
        .counter_dir = TIMER_COUNT_UP, //计数方式是向上计数
        .counter_en = TIMER_PAUSE, //调用timer_init函数以后不启动计数,调用timer_start
        .alarm_en = TIMER_ALARM_EN, //到达计数值启动报警(计数值溢出,进入中断)
        .auto_reload = 1, //自动重新装载预装值
    };
    /*初始化定时器;TIMER_GROUP_0(定时器分组0); TIMER_0(分组0中的0号定时器)*/
    timer_init(TIMER_GROUP_0, TIMER_0, &config);

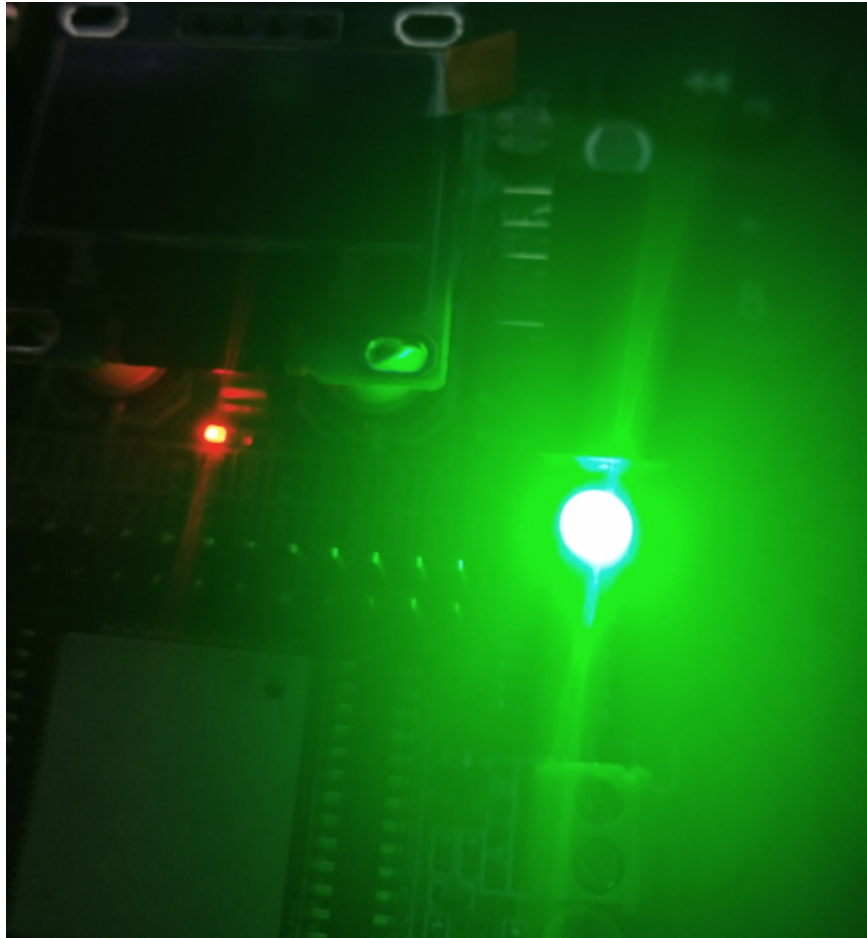
    /*设置定时器预装值,0*/
    timer_set_counter_value(TIMER_GROUP_0, TIMER_0, 0x00000000ULL);
    /*设置报警阈值*/ // 1000[定时1000ms]*(TIMER_BASE_CLK[定时器时钟]/8[分频系数])/10
    timer_set_alarm_value(TIMER_GROUP_0, TIMER_0, 1000*(TIMER_BASE_CLK/8/1000
    /*使能定时器中断*/
    timer_enable_intr(TIMER_GROUP_0, TIMER_0);
    /*注册定时器中断函数*/

```

```
timer_isr_register(TIMER_GROUP_0, TIMER_0,
    timer_group0_isr, //定时器回调函数
    (void *) TIMER_0, //传递给定时器回调函数的参数
    ESP_INTR_FLAG_IRAM, //把中断放到 IRAM 中
    NULL //调用成功以后返回中断函数的地址,一般用不到
);
/*启动定时器*/
timer_start(TIMER_GROUP_0, TIMER_0);
}
```



下载进去之后,会看到led每隔1S闪耀



如果想让定时器是一次性的,可以把重新使能屏蔽

```

/*定时器中断函数
因为设置中断的时候,中断设置的 ESP_INTR_FLAG_IRAM 所以中断函数需要加 IRAM_ATTR
*/
void IRAM_ATTR timer_group0_isr(void *para)
{
    //获取定时器分组0中的哪一个定时器产生了中断
    uint32_t timer_intr = timer_group_get_intr_status_in_isr(TIMER_GROUP_0);
    if (timer_intr & TIMER_INTR_T0) { //定时器0分组的0号定时器产生中断
        /*清除中断*/
        timer_group_clr_intr_status_in_isr(TIMER_GROUP_0, TIMER_0);
        /*重新使能定时器中断*/
        timer_group_enable_alarm_in_isr(TIMER_GROUP_0, TIMER_0);
    }

    /*设置gpio输出高低电平*/
    gpio_set_level(gpio_pin, 1-gpio_get_level(gpio_pin));
}

```

如果想使用分组1

TIMER_GROUP_1

分类: [ESP32学习开发](#)

好文要顶

关注我

收藏该文



杨奉武

关注 - 1

粉丝 - 633

0

0

« 上一篇: [007-ESP32_SDK开发-使用缓存管理传递数据](#)

posted on 2021-08-07 00:29 杨奉武 阅读(0) 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!

【推荐】在强大的算力算法加持下，国双上榜 “中国大数据企业50强”

编辑推荐：

- 源码 | “@Value 注入失败”引发的一系列骚操作
- Redis挂了，流量把数据库也打挂了，怎么办？
- 五个 .NET 性能小贴士
- Web动画 | 科技感十足的暗黑字符雨动画
- 聊聊【向上管理】中的“尺度”



最新新闻：

- 字节跳动还有下一个奇迹吗？
 - 锂资源不够用，“钠”就取而代之？
 - 重识拼多多：当低垂的果实被摘完
 - 海信最后的倔强，激光电视最终难逃“过渡产品”的命运？
 - 没有视频，不能点赞，这个产品却想用抖音的方式让你上瘾
- » 更多新闻...

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。