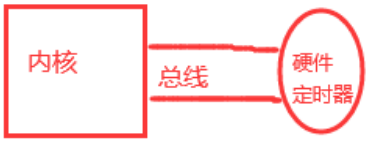
-
-
-

说明

系统滴答定时器是系统内核(RISC-V内核)自带的定时器.

有人会问:那后面学到的那些通用定时器,高级定时器呢?

那些都是外设,然后依靠总线(导线)和它通信控制其实现具体功能.



又比如说:单片机有串口功能,实际上就是内核+硬件串口的组合, 内核通过总线控制硬件串口通信.

随便看一个工程

1,官方给了一个不使用中断操作滴答定时器的例子

具体呢不做讨论了,直接使用就可以.

(对于初学者不必深究,即使深究最终也只是配置寄存器而已, 先学会用,然后使用遇到问题时再解决)

2. Re:ESP8266 SDK开发: 综合篇-C#上位机串口通信控制ESP8266
认真拜阅读您的程序 收获很大 ;但我发现一处问题 就是您在串口接收函数中 没有将空闲标志清零 导致程序最终并没有通过空闲中断来处理 而是每隔10ms处理一次

--觉代疯骚

debug.c

```
 */
void Delay_Init(void)
{
    p_us = SystemCoreClock / 8000000;
    p_ms = (uint16_t)p_us * 1000;
}

/*********************************************************************
 * @fn      Delay_Us
 *
 * @brief   Microsecond Delay Time.
 *
 * @param   n - Microsecond number.
 *
 * @return  None
 */
void Delay_Us(uint32_t n)
{
    uint32_t i;

    SysTick->SR &= ~(1 << 0);
    i = (uint32_t)n * p_us;

    SysTick->CMP = i;
    SysTick->CTLR |= (1 << 4) | (1 << 5) | (1 << 0);

    while((SysTick->SR & (1 << 0)) != (1 << 0))
        ;
    SysTick->CTLR &= ~(1 << 0);
}

/*********************************************************************
 * @fn      Delay_Ms
 *
 * @brief   Millisecond Delay Time.
 *
 * @param   n - Millisecond number.
 *
 * @return  None
 */
void Delay_Ms(uint32_t n)
{
    uint32_t i;

    SysTick->SR &= ~(1 << 0);
    i = (uint32_t)n * p_ms;
```

现在看中断方式

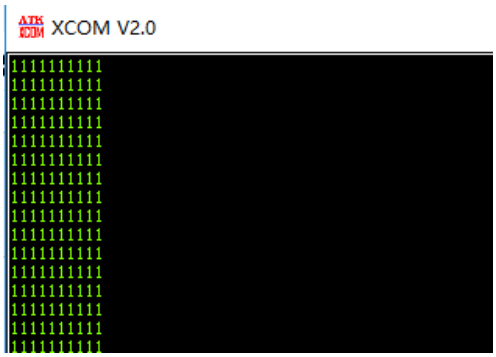
main.c

```c

#include "debug.h"
#include "ch32v30x.h"

volatile uint16_t SysTickCnt=0;

void SysTick_init(void)
{
    /*配置中断优先级*/
    NVIC_InitTypeDef NVIC_InitStructure = {0};
    NVIC_InitStructure.NVIC_IRQChannel = SysTicK_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;//抢占式优先级
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;//响应式优先级
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;//使能
    NVIC_Init(&NVIC_InitStructure);

    /*配置定时器*/
    SysTick->CTLR= 0;
    SysTick->SR  = 0;
    SysTick->CNT = 0;
    SysTick->CMP = SystemCoreClock/1000;//后面的1000代表1000HZ(那就是1ms进一次中断)
    SysTick->CTLR= 0xf;
}

__attribute__((interrupt("WCH-Interrupt-fast")))
void SysTick_Handler(void)
{
    SysTick->SR=0;//清除中断
    SysTickCnt++;
}

int main(void)
{
    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);//设置优先级分组为2
    USART_Printf_Init(115200);
    SysTick_init();
    while(1)
    {
        if (SysTickCnt>=1000) {
            SysTickCnt=0;
            printf("1111111111\r\n");
        }
    }
}

```


```c
#include "debug.h"
#include "ch32v30x.h"

volatile uint16_t SysTickCnt=0;

void SysTick_init(void)
{
    /*配置中断优先级*/
    NVIC_InitTypeDef NVIC_InitStructure = {0};
    NVIC_InitStructure.NVIC_IRQChannel = SysTicK_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;//抢占式优
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;//响应式优先级
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;//使能
    NVIC_Init(&NVIC_InitStructure);

    /*配置定时器*/
    SysTick->CTLR= 0;
    SysTick->SR  = 0;
    SysTick->CNT = 0;
    SysTick->CMP = SystemCoreClock/1000;//后面的1000代表1000HZ(那就是1ms
    SysTick->CTLR= 0xf;
}

__attribute__((interrupt("WCH-Interrupt-fast")))
void SysTick_Handler(void)
{
    SysTick->SR=0;//清除中断
    SysTickCnt++;
}

int main(void)
```

```
{
    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);//设置优先级分组为2
    USART_Printf_Init(115200);
    SysTick_init();
    while(1)
    {
        if (SysTickCnt>=1000) {
            SysTickCnt=0;
            printf("1111111111\r\n");
        }
    }
}
```

下载测试会看到串口每隔1S打印一次数据

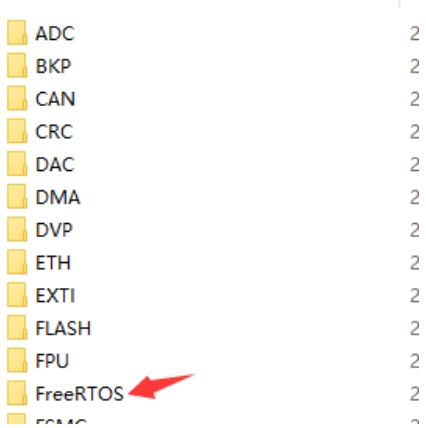


上面就是说完了，我说一下我是如何写的上面的程序

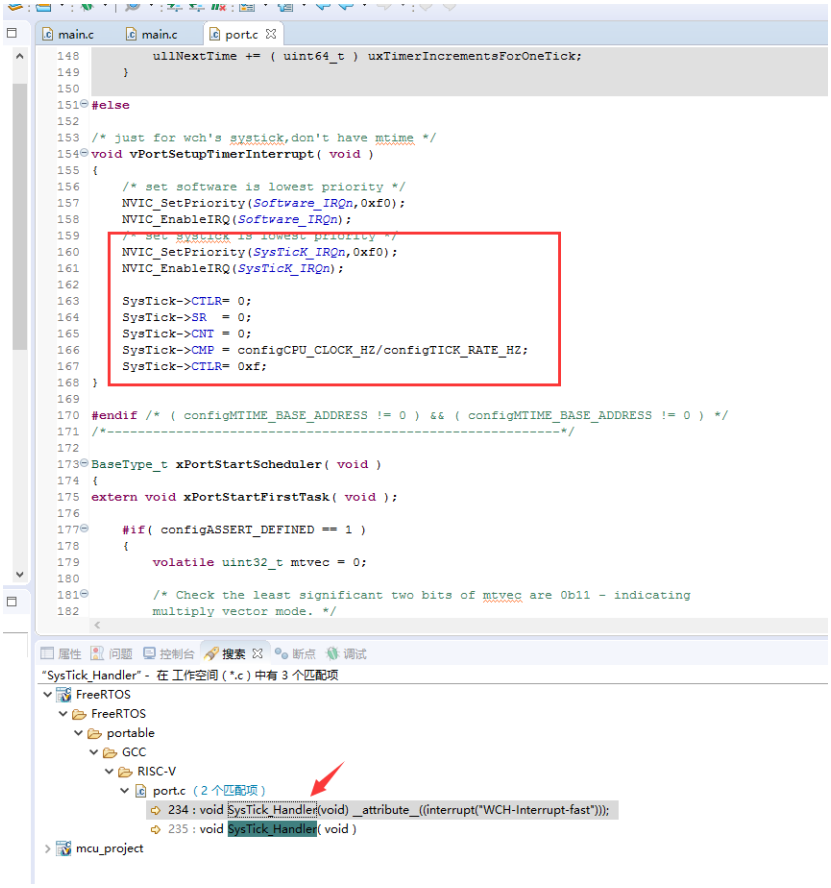
其实没有例程....

但是呢,官方提供了一个FreeRTOS(操作系统)的例子,

大家伙要知道,所有的操作系统都要依靠定时器来切换任务,一般都是使用系统滴答定时器



我就全局搜索 SysTick_Handler, 然后翻了翻就找到了



分类: CH32V307(WCH单片机)学习开发

杨奉武
关注 - 1
粉丝 - 779

0 0

« 上一篇： 移植升级底层包,把自己的用户程序增加上OTA功能

posted on 2022-05-18 23:49 杨奉武 阅读(0) 评论(0) 编辑 收藏 举报

自动补全

提交评论 退出

[Ctrl+Enter快捷键提交]

【推荐】大学生技术公益开发训练营，让你的应用为公益发光发热

【推荐】阿里云数据库训练营，云数据库 MySQL 从入门到高阶

编辑推荐：

· 闲置树莓派：种朵花然后做延时摄影吧
· 理解 ASP.NET Core - 发送 Http 请求(HttpClient)
· 探索 ABP 基础架构
· 万字长文深度剖析 RocketMQ 设计原理
· 戏说领域驱动设计（廿六）——再谈事务

最新新闻：

· ACL 2022奖项公布！达摩院、华为等机构获奖，陈丹琦、杨笛一亦榜上有名
· 斗鱼发布2022年Q1财报：总营收17.96亿元，持续加大内容生态投入
· 用了NASA太空服技术的运动衣，知冷又知热
· Google 新园区正式开放，100%「纯电动」
· 如何编写数学上完美的软件
» 更多新闻...

历史上的今天：

2021-05-18 04-STM32+Air724UG(4G模组)远程升级篇OTA(自建物联网平台)-STM3...
2017-05-18 STM32采集电阻触摸贴膜

单片机,物联网,上位机,…

扫一扫二维码，加入群聊。