



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 - 781, 文章 - 0, 评论 - 325, 阅读 - 193万

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

渡我不渡她

小镇姑娘

3 PDD洪荒之力

⚠ 加入QQ群

昵称: 杨奉武 园龄:6年 粉丝: 671 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(5)

Android(22)

Android 开发(8)

C# 开发(4)

CH395Q学习开发(17)

CH579M物联网开发(12)

CH579M学习开发(8)

ESP32学习开发(20)

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)

8-HC32F460(华大单片机)-串口(定时器空闲检测)

<iframe name="ifd" src="https://mnifdv.cn/resource/cnblogs/LearnHC32F460"</p> frameborder="0" scrolling="auto" width="100%" height="1500"></iframe>

HC32F460(华大单片机)学习开发

开发板原理

图:https://mnifdv.cn/resource/cnblogs/LearnHC32F460/HC32F460.PDF

资料源码下载链

接:https://github.com/yangfengwu45/learnHC32F460.git

- 1-硬件使用说明
- 2-工程模板使用说明
- 3-GPIO输出高低电平
- 4-GPIO引脚电平检测
- 5-串口(基本使用)
- 6-时钟树
- 7-定时器Timer0
- 8-串口(定时器空闲检测)

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(华大单片机)学习开发(8)

NB-IOT Air302 AT指令和LUA 脚本语言开发(27)

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(24)

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(12)

感想(6)

更多

阅读排行榜

- 1. ESP8266使用详解(AT,LUA, SDK)(173618)
- 2. 1-安装MQTT服务器(Windo ws),并连接测试(102510)
- 3. 用ESP8266+android,制作 自己的WIFI小车(ESP8266篇) (66289)
- 4. ESP8266刷AT固件与node mcu固件(66032)
- 5. 有人WIFI模块使用详解(390 45)
- 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)(36536)
- 7. 关于TCP和MQTT之间的转 换(34519)
- 8. C#中public与private与stat ic(34460)
- 9. android 之TCP客户端编程 (32524)
- 10. android客服端+eps8266 +单片机+路由器之远程控制系统(31528)

串口空闲中断

串口发送数据的时候都是一条一条发送的.

单片机判断接收完一条数据一般做法是:

在接收数据的过程中,如果超过一段时间没有新的数据过来,就认为 前面接收完了一条数据.

然后再去处理数据.

关于此单片机的空闲中断

感觉这单片机的功能制作是总结了项目经验之后做的.

这个单片机使用的是单独的定时器去配置超时时间,

定时器0的第一路的A通道作为串口1的串口接收超时定时器

定时器0的第一路的B通道作为串口2的串口接收超时定时器

定时器0的第二路的A通道作为串口3的串口接收超时定时器

定时器0的第二路的B通道作为串口4的串口接收超时定时器

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)

最新评论

1. Re:用ESP8266+android, 制作自己的WIFI小车 (Android 软件) 百度盘都失效了

--ghggaojian

2. Re:201-STM32+Air724UG基本控制 篇(阿里云物联网平台)-设备 使用物模型Topic上报温湿度

你好,有源码吗?

--zsw1997



26.4.1.6 UART 接收 TIMEOUT 功能

UART 接收数据停止位被检测时 TIMEOUT 计数器启动, 经过设定的 TIMEOUT 时间(设定单位为接收位数)后未检测到下一帧接收数据时, 发生 TIMEOUT, 如果此时 CRI.RE=1,则 TIMEOUT 状态位 USARTn_SR.RTOF 置位,如果此时 USARTn_CRI.RE=0,则等待 USARTn_CRI.RE=1后 TIMEOUT 状态位 USARTn SR.RTOF置位。

TIMEOUT 计数器采用 Timer0 模块的计数器, 具体对应关系如下:

USART1: Timer0 Unit1 A 通道 USART2: Timer0 Unit1 B 通道 USART3: Timer0 Unit2 A 通道 USART4: Timer0 Unit2 B 通道

TIMEOUT 功能 Timer() 比较计数器值设定

Timer0 为 16 位计数器,计数时钟最大可以选择 1024 分频,TMR0_CMPAR 值设 定计算公式如下:

$$CMPA < B > R = \frac{RTB}{2^{CKDIVA < B>}} - 4 \text{ (计数时钟不分频)}$$

$$CMPA < B > R = \frac{RTB}{2^{CKDIVA < B>}} - 2 \text{ (计数时钟 2 分頻)}$$

$$CMPA < B > R = \frac{RTB}{2^{CKDIVA < B>}} - 1 \text{ (计数时钟 4 分頻及以上)}$$

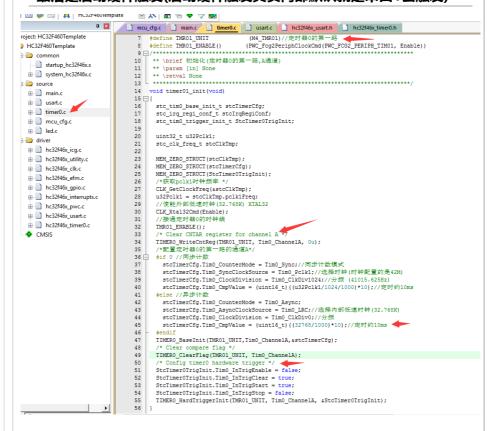
直接看程序

1.我使用的是串口1. 首先看定时器配置

配置定时器0的第一路的通道A

我设置的定时时间是10ms左右, 那么串口空闲接收超时时间就是10ms

最后是启动硬件触发(启动硬件触发其实内部默认就是串口1去触发)



2. 串口配置

串口和先前的例程没有大的区别,但是要注意需要让其输出时钟信号 □ □ ▼ | ♥ | ▼ | ▼ | ▼ | ▼ | ■ | ■ | ■ | ■ | ₩ ₩ HC32F460Template mcu_cfg.c main.c timer0.c usart.c hc32f46x_usart.h hc32f46x_timer0.h 'roject: HC32F460Template /* Enable peripheral clock *//*打开时钟*/ HC32F460Template PWC_Fcg1PeriphClockCmd(PWC_FCG1_PERIPH_USART1 | PWC_FCG1_PERIPH_USART2 | PWC_FCG1_PERIPH_USART3 | PWC_FCG1_PERIPH_USART4, Enable); startup hc32f46x.s system_hc32f46x.c /* Initialize USART TO */ /*配置相应的TO作为由口的TX.RX引脚* PORT_SetFunc(USART1_TX_PORT, USART1_TX_PIN, Func_Usart1_Rx, Disable); PORT_SetFunc(USART1_TX_PORT, USART1_TX_PIN, Func_Usart1_Tx, Disable); ∃ 🍅 source main.c ⊕ usart.c Const sto_usert_uart_init_t stofinitGr = { UsartIntClkCkOutput,//使用內部时幹源,輸出通信的时幹信号 UsartClkDiy_1,//时幹不分頻 timer0.c 101 mcu_cfg.c 102 ⊕ led.c UsartDataBits8.//一个字节数据用8位数据位表示 driver UsartDataBits8,//一个手节数据用8位数据位表示 UsartDataLsbFirst,//朱徐新低位 UsartDasStopBit,//停止位1位 UsartParityNone,//无奇偶校验 UsartSamleBit8,//每次传输8位(1字节),也可以传输 UsartSamleBit16(16位,2字节) UsartStartBitFallEdge,// UsartStartBitFallEdge,// hc32f46x icq.c hc32f46x_utility.c ⊕ hc32f46x_clk.c hc32f46x_efm.c 110 hc32f46x_gpio.c 111 hc32f46x_interrupts.c 112 /* Initialize UART *//*初始化串口配置*/ enRet = USART_UART_Init(USART1_CH, &stcInitCfg); ⊕ hc32f46x_pwc.c he22f46v usart s 然后呢加上串口空闲中断 mcu_cfg.c main.c timer0.c <u>usart.c</u> hc32f46x_usart.h hc32f46x_timer0.h UsartStartBitFallEdge,// UsartRtsEnable,//使能RIS (串口开始传输前让RIS输出一个高脉冲信号) HC32F460Template 110 i 🗀 🗀 common 112 startup hc32f46x.s /* Initialize UART *//*初始化串口配置*. system_hc32f46x.c /- Introduce Our *//*2019年代中国地域*/ enRet = OR)While (1); if (enRet != OR)While (1); /* Set baudrate *//*设置串口波特率*/ main.c usart.c timer0.c /* Set USARI RX IRQ *//*设置串口接收中断*. ⊕ 📗 mcu_cfg.c /* Set USART RX IRQ *//*设置中二接级中部/ suclrageqicGg.enRngn = USARTI_RI IRQ;//设置中断向量 suclrageqicGg.pfnclalback = AUBartIRXirqCallback;//设置中断回调函数 suclrageqicGg.enIntSre = USARTI_RI_NUM;//中断名称(申口接收中断) enIrqReqistration(sstcIrqReqicGg); NVIC_SetPriority(stcIrqReqicGg.enIRQn,) USARTI_RI_IRQ_PRIORITY);//设置中断优先级 NVIC_ClearPendingTRQ(suclrageqicGg.enIRQn); NVIC_EnableIRQ(stcIrqReqicGg.enIRQn); led.c driver hc32f46x_icg.c hc32f46x_utility.c hc32f46x_clk.c hc32f46x_efm.c /* Set USART RX error IRQ *//*设置中口接收错误中断*/ stoIrqRegicfg.enIRQn = USARTi_EI_IRQ://设置中断向量 stoIrqRegicfg.enIRQn = USARTi_EI_IRQ://设置中断向量 stoIrqRegicfg.enInCsr = USARTi_EI_INDM://中断名称(申口I接收错误中断) enIrqRegistration(astoIrqRegicfg): NVIC_SetPriority(stoIrqRegicfg.enIRQn, USARTi_EI_IRQ_PRIORITY);//设置中断优先级 NVIC_ClearPendingIRQ(stoIrqRegicfg.enIRQn); NVIC_EnableIRQ(stoIrqRegicfg.enIRQn); hc32f46x_gpio.c hc32f46x_interrupts.c hc32f46x_pwc.c hc32f46x_usart.c · CMSIS /* Set USART RX timeout error IRQ (幸口接收空用中断)*/ stcirgRegiCfg.enIRQn = USART1_RIO_IRQ;//中断向量 stcirgRegiCfg.pfnCallbeck = 4USART1_RIO_IRQ;//中断向量 stcirgRegiCfg.enIntSro = USART1_RIO_IRW;//中断各称(串口1空闭中断) enIrqRegistration(satcirgRegiCfg); NVIC_SetPriority(stcirgRegiCfg.enIRQn, USART1_RIO_IRQ_FRIORITY); NVIC_ClearPendingIRQ(stcirgRegiCfg.enIRQn); NVIC_EnableIRQ(stcirgRegiCfg.enIRQn); /*Enable RX 66 RX interupt function 66 UsartIx*/ USART_FuncCnd(USART1_CH, UsartRx, Enable)://使能接收 USART_FuncCnd(USART1_CH, UsartRxInt, Enable)://使能接收中断 USART_FuncCnd(M4 USART1_UsartIx, Enable)://使能接坡 //打开操收額时和接收額时中断 USART_FuncCnd(USART1_CH, UsartTimeOut, Enable); USART_FuncCnd(USART1_CH, UsartTimeOutInt, Enable);

我在中断里面是设置led翻转

```
の 図 mcu_cfg.c main.c timef0.c usart.c hc32446x_usart.h hc32446x_timer0.h late 31 /* 車口空闲中断代先級*/ 32 edefine USARTI RTO_TRQ_PRIORITY (DDL_IRQ_PRIORITY_DEFAULT) 33 /* 車口空用中断向量*/ 24 Machine TROPTS TOO TRO (TOP 100 3 TROP)
 Project: HC32F460Template
HC32F460Template
 e Common
                                           34 #define USART1_RTO_IRQ
                                                                                        (Int003_IRQn)
      startup_hc32f46x.s
     system_hc32f46x.c

☐ ☐ source
   main.c
                                           38 /**
39 ** \prief USART RX irq callback function.//申口接收中断函数
40 ** \param [in] None
41 ** \retval None
42 **
43 static void UsartlRxIrqCallback(void)
44 {
5     uint16_t m_u16RxData;
6     m_u16RxData = USART_RecData(USARTI_CH);//获取申口接收的数据
47
    timer0.c
    mcu_cfg.c
     🖃 🧀 driver
    hc32f46x_icg.c
     ⊕ hc32f46x_utility.c
                                                USART_SendData(USART1_CH, m_u16RxData);
     ⊕ ll hc32f46x_clk.c
     hc32f46x_efm.c
     ⊕ hc32f46x_gpio.c
    hc32f46x_interrupts.c
                                            53 ** \brief USART timeout irq callback function.
54 ** \param [in] None
55 ** \retval None
56 ***
    ⊕ hc32f46x_pwc.c
     hc32f46x_usart.c
     hc32f46x timer0.c
                                                {
gpio_set_get(1,-1);
TIMERO_Cmd(M4_IMRO1, TimO_ChannelA, Disable);
USARI_ClearStatus(USART1_CH, UsartRxTimeOut);
                                            66 ** \brief USART RX error irq callback function.(串口接收错误中断处理函数)
```

测试

把这节程序下载到开发板,然后使用调试助手给单片机串口1发送数据

会观察到,每次发送完数据,开发板上的小灯就会翻转

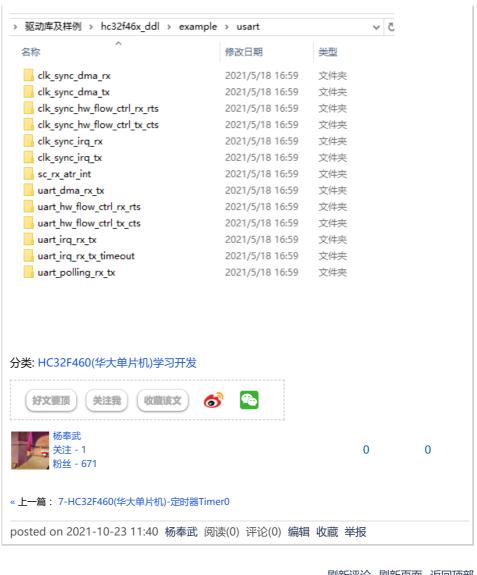


提示

定时器0以后就不要使用了哈....

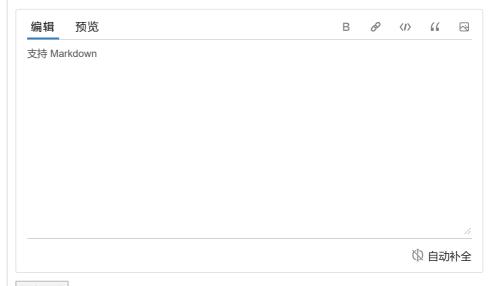
用别的定时器吧

其它串口例程参考官方例程(后面不一定写串口中断发送和DMA的例程)



刷新评论 刷新页面 返回顶部

发表评论



提交评论 退出

[Ctrl+Enter快捷键提交]

【推荐】并行超算云面向博客园粉丝推出"免费算力限时申领"特别活动

【推荐】跨平台组态\工控\仿真\CAD 50万行C++源码全开放免费下载!

【推荐】华为全面助力青少年编程教育普及,花瓣少儿编程正式上线



编辑推荐:

- ·消息队列那么多,为什么建议深入了解下RabbitMQ?
- ·技术管理进阶——管人还是管事?
- · 以终为始: 如何让你的开发符合预期
- ·五个维度打造研发管理体系
- ·不会SQL也能做数据分析?浅谈语义解析领域的机会与挑战

最新新闻:

- ·T-Mobile将于3月关闭旗下Sprint的CDMA 3G网络 (2021-10-23 09:17)
- ·NASA"毅力号"拍摄三角洲及山丘图像以了解杰泽罗陨石坑过去的历史(2021-10-23 09:11)
- ·马斯克调侃库克:快来看苹果抛光布(2021-10-23 09:06)
- ·苹果更新应用商店规则:允许开发者使用替代支付方式(2021-10-2309:00)
- ·大脑并行处理语音和其它声音(2021-10-2222:41)
- » 更多新闻...

Powered by: 博客园 Copyright © 2021 杨奉武 Powered by .NET 6 on Kubernetes







单片机,物联网,上位机,… 扫一扫二维码,加入群聊。