

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 - 733, 文章 - 0, 评论 - 314, 阅读 - 179万

导航

博客园
首页
新随笔
联系
订阅 
管理

公告

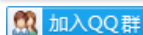


渡我不渡她 -

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9个月
粉丝：623
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

Android(22)
Android 开发(8)
C# 开发(4)
CH395Q学习开发(17)
CH579M学习开发(6)
ESP32学习开发(8)
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)

102-CH579M学习开发-基本外设-串口

<p> <iframe name="ifd" src="https://mnifdv.cn/resource/cnblogs/LearnCH579M" frameborder="0" scrolling="auto" width="100%" height="1500"> </iframe> </p>

单片机CH579M(带蓝牙和以太网口的ARM M0内核的单片机)学习开发

替代STM32,替代串口转以太网DTU

开发板链接:<https://item.taobao.com/item.htm?ft=t&id=648634562877>

芯片购买链

接:<https://item.taobao.com/item.htm?ft=t&id=649533679749>

开发板原理

图:<https://mnifdv.cn/resource/cnblogs/LearnCH>

资料源码下载链

接:<https://github.com/yangfengwu45/LearnCH5>

■ [学习Android](#)

教程中搭配的Android , C#等教程如上 , 各个教程正在整理。

■ [001-硬件使用说明,下载和运行第一个程序](#)

■ [002-官方资料学习说明,开发板蓝牙\(蓝牙定位\),网口通信测试](#)

■ [003-新建工程说明](#)

■ [100-基本外设-GPIO输入输出](#)

■ [101-基本外设-定时器](#)

■ [102-基本外设-串口](#)

-
-
-
-

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(58)
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(10)
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:C#开发: 通信篇-TCP客
户端
感谢分享，直接就用上了
--Zfen
2. Re:03-STM32+Air724UG
远程升级篇OTA(阿里云物联
网平台)-STM32+Air724UG
使用阿里云物联网平台OTA
远程更新STM32程序
楼主，单片机和Air724模块
之间是通过AT指令通讯的
吗？
--a314825348

阅读排行榜

1. ESP8266使用详解(AT,LUA,
SDK)(172533)
2. 1-安装MQTT服务器(Windo
ws)并连接测试(98112)
3. ESP8266刷AT固件与node
mcu固件(64347)
4. 用ESP8266+android,制作
自己的WIFI小车(ESP8266篇)
(63642)

说明

单片机共有4个串口

6.3 寄存器描述	24
6.4 功能描述及配置	28
6.4.1 RTC计数器初始 化	28
6.4.2 RTC时钟源切换 为LSE晶振	28
6.4.3 RTC定时功能	28
6.4.4 RTC触发功能	28
6.4.5 用HSE校准内部 32K时钟LSI	28
第7章 通用I/O和复用功能	30
7.1 GPIO简介	30
7.2 外部中断/唤醒	30
7.3 GPIO的复用与重映 射	30
7.3.1 复用功能	31
7.3.2 功能引脚重映 射	32

ET+/ET-	连接到内部以太网收发器	浮空输入
ER+/ER-	连接到内部以太网收发器	浮空输入

7.3.2 功能引脚重映射

为了使外设功能的同时利用率达到最优，可以通过设置 R16_PIN_ALTERNATE 功能引脚重映射寄存器把一些功能引脚重新映射到其他引脚上。

表 7-7 复用功能重映射引脚

外设功能引脚	默认所在的 GPIO 引脚	重映射到的 GPIO 引脚
SPI0	PA[12]/PA[13]/PA[14]/PA[15]	PB[12]/PB[13]/PB[14]/PB[15]
RXD3/TXD3	PA[4]/PA[5]	PB[20]/PB[21]
RXD2/TXD2	PA[6]/PA[7]	PB[22]/PB[23]
RXD1/TXD1	PA[8]/PA[9]	PB[8]/PB[9]
RXD0/TXD0	PB[4]/PB[7]	PA[15]/PA[14]
TMR3/PWM3/CAP3	PA[2]	PB[18]
TMR2/PWM2/CAP2	PA[11]	PB[11]
TMR1/PWM1/CAP1	PA[10]	PB[10]
TMR0/PWM0/CAP0	PA[3]	PB[19]

7.4 寄存器描述

表 7-8 GPIO 相关寄存器列表

名称	访问地址	描述	复位值
R16_PIN_ALTERNATE	0x40001018	功能引脚重映射寄存器	0x0000

第 9 章 通用异步收发器 UART

9.1 UART 简介

芯片提供了 4 组全双工的异步串口，UART0/1/2/3。支持全双工和半双工串口通讯，其中 UART0 提供发送状态引脚用于切换 RS485，并且支持 MODEM 调制解调器信号 CTS、DSR、RI、DCD、DTR、RTS。

9.1.1 主要特性

- 兼容 16C550 异步串口并且有所增强。
- 支持 5、6、7 或者 8 个数据位以及 1 或者 2 个停止位。
- 支持奇、偶、无校验、空白 0、标志 1 等校验方式。
- 可编程通讯波特率，最高达 5Mbps 波特率。
- 内置 8 个字节的 FIFO 先进先出缓冲器，支持 4 个 FIFO 触发级。
- UART0 支持 MODEM 调制解调器信号 CTS、DSR、RI、DCD、DTR、RTS。
- UART0 支持硬件流控制信号 CTS 和 RTS 自动握手和自动传输速率控制，兼容 TL16C550C。
- 支持串口帧错误检测、支持 Break 线路间隔检测。
- 支持全双工和半双工串口通讯，UART0 提供发送状态引脚用于切换 RS485。

9.2 寄存器描述

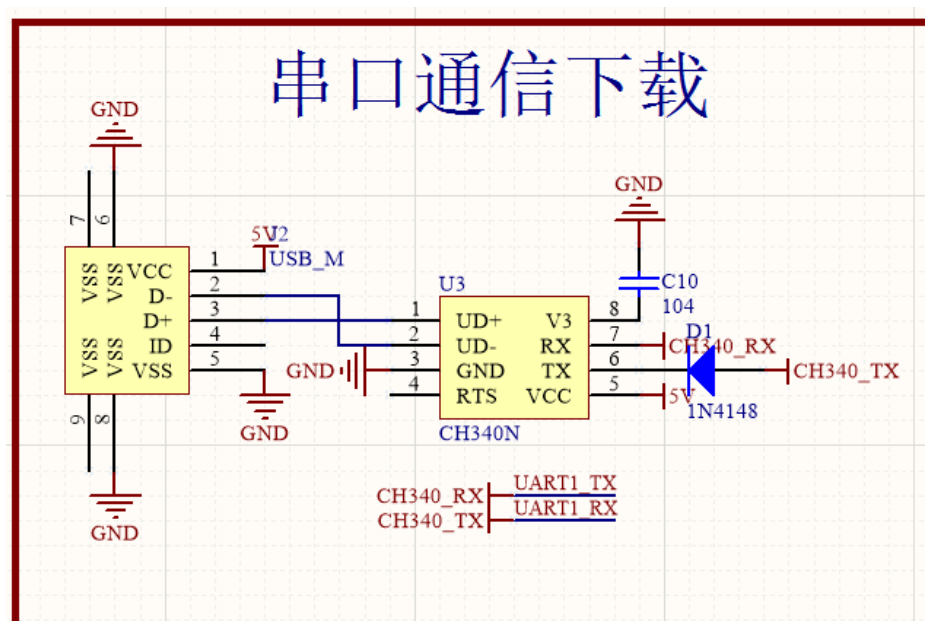
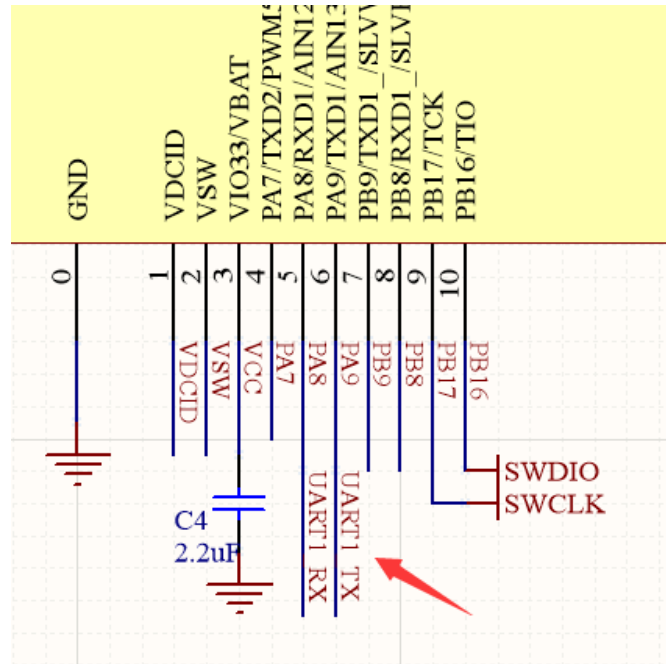
表 9-1 UART0 相关寄存器列表

5. 有人WIFI模块使用详解(38377)
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)(35802)
7. 关于TCP和MQTT之间的转换(32923)
8. C#中public与private与static(31962)
9. android 之TCP客户端编程(31710)
10. android服务端+eps8266+单片机+路由器之远程控制系統(31263)

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)

开发板上连接的是串口1



在中断函数里面接收数据,并返回接收的数据

```

CH5795FR.h  CH57x_uart.h  CH57x_timer0.c  main.c  CH57x_uart1.c  CH57x_sys.c  CH57x_gpio.h  CH57
1 #include "CH57x_common.h"
2
3 __align(4) uint8 RxBuffer[ MAX_PACKET_SIZE ]; // IN, must even address
4 __align(4) uint8 TxBuffer[ MAX_PACKET_SIZE ]; // OUT, must even address
5
6
7 int main()
8 {
9     /*配置串口1*/
10    GPIOA_SetBits(GPIO_Pin_9);
11    GPIOA_ModeCfg(GPIO_Pin_8, GPIO_ModeIN_PU); // RXD-配置上拉输入
12    GPIOA_ModeCfg(GPIO_Pin_9, GPIO_ModeOut_PP_5mA); // TXD-配置推挽输出, 注意先让Io口输出高电平
13    UART1_DefInit();
14    /*使能接收中断,接收错误中断*/
15    UART1_INTCfg( ENABLE, RB_IER_RECV_RDY|RB_IER_LINE_STAT );
16    // NVIC_SetPriority(TMR0_IRQn,18); //设置中断优先级(也可以不设置,默认14)
17    NVIC_EnableIRQ( UART1_IRQn );
18
19    while(1)
20    {
21    }
22 }
23 /*串口中断*/
24 void UART1_IRQHandler(void)
25 {
26     char data;
27     switch( UART1_GetITFlag() )
28     {
29         case UART_II_LINE_STAT: // 线路状态错误
30             UART1_GetLinSTA();
31             break;
32         case UART_II_RECV_RDY: // 数据达到设置触发点
33             data = UART1_RecvByte();
34             UART1_SendByte(data);
35             break;
36         case UART_II_RECV_TOUT: // 接收超时, 暂时一帧数据接收完成
37             data = UART1_RecvByte();
38             UART1_SendByte(data);
39             UART1_SendByte('b'); //打印一个字符,说明进入了此函数
40             break;
41         case UART_II_THR_EMPTY: // 发送缓存区空, 可继续发送
42             break;
43         case UART_II_MODEM_CHG: // 只支持串口0
44             break;
45         default:
46             break;
47     }
48 }

```



```

int main()
{
    /*配置串口1*/
    GPIOA_SetBits(GPIO_Pin_9);
    GPIOA_ModeCfg(GPIO_Pin_8, GPIO_ModeIN_PU); // RXD-配置上拉输入
    GPIOA_ModeCfg(GPIO_Pin_9, GPIO_ModeOut_PP_5mA); // TXD-配置推挽输出, 注意先让Io口输出高电平
    UART1_DefInit();
    /*使能接收中断,接收错误中断*/
    UART1_INTCfg( ENABLE, RB_IER_RECV_RDY|RB_IER_LINE_STAT );
    // NVIC_SetPriority(TMR0_IRQn,18); //设置中断优先级(也可以不设置,默认14)
    NVIC_EnableIRQ( UART1_IRQn );

    while(1)
    {
    }
}

/*串口中断*/
void UART1_IRQHandler(void)
{
    char data;
    switch( UART1_GetITFlag() )
    {
        case UART_II_LINE_STAT: // 线路状态错误
            UART1_GetLinSTA();
            break;
        case UART_II_RECV_RDY: // 数据达到设置触发点
            data = UART1_RecvByte();
            UART1_SendByte(data);
            break;
        case UART_II_RECV_TOUT: // 接收超时, 暂时一帧数据接收完成
            data = UART1_RecvByte();
            UART1_SendByte(data);
            UART1_SendByte('b'); //打印一个字符,说明进入了此函数
            break;
        case UART_II_THR_EMPTY: // 发送缓存区空, 可继续发送
            break;
        case UART_II_MODEM_CHG: // 只支持串口0
            break;
        default:
            break;
    }
}

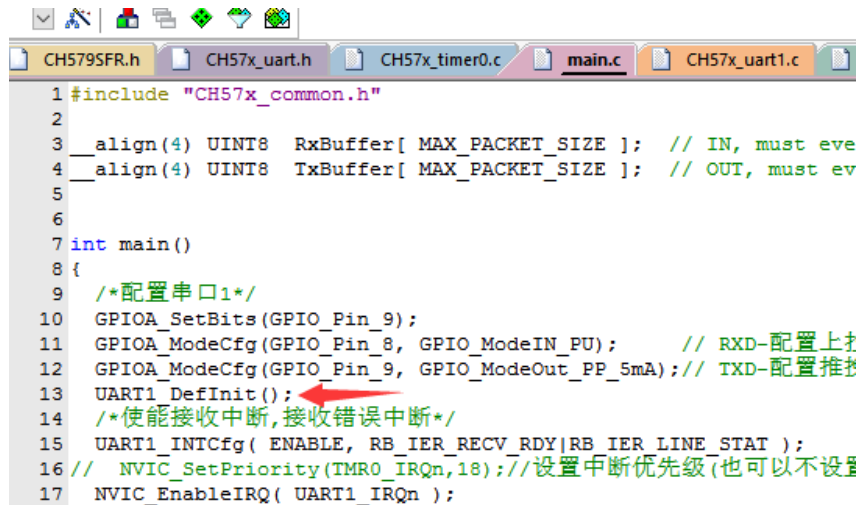
```

```

        data = UART1_RecvByte();
        UART1_SendByte(data);
        UART1_SendByte('b');//'òÔ;Ô»,ö×Ô·û,ÈµÃ÷¼øÈèÁÈ´È°-Èý
        break;
    case UART_II_THR_EMPTY:        // ·¿Èí»°´æÇø;Ô£¬¿È¼ÏÐø·¿Èí
        break;
    case UART_II_MODEM_CHG:        // Ô»Ô$³Ô´@¿Ú0
        break;
    default:
        break;
}
}

```

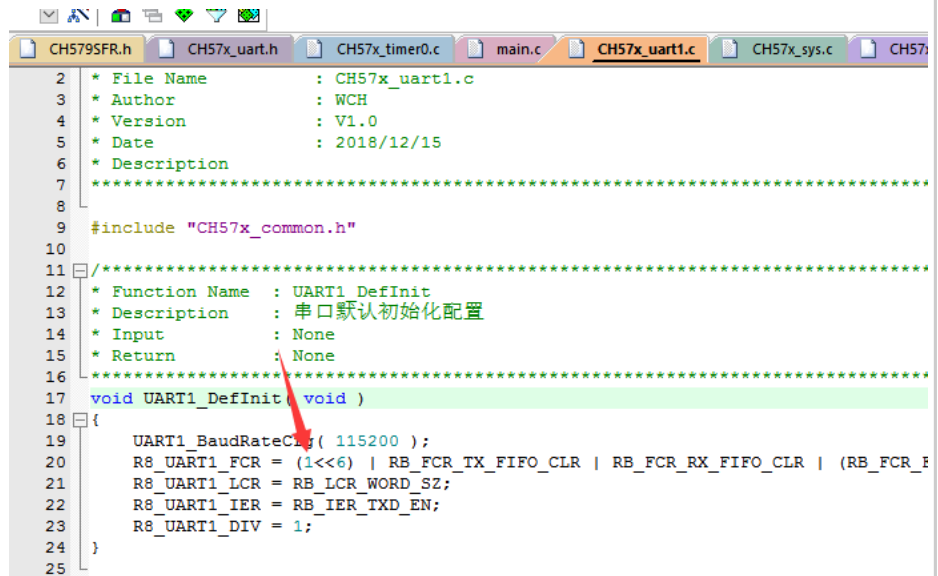
注意:需要修改下官方的UART1_DefInit();函数



```

1 #include "CH57x_common.h"
2
3 __align(4) UINT8  RxBuffer[ MAX_PACKET_SIZE ]; // IN, must eve
4 __align(4) UINT8  TxBuffer[ MAX_PACKET_SIZE ]; // OUT, must ev
5
6
7 int main()
8 {
9     /*配置串口1*/
10    GPIOA_SetBits(GPIO_Pin_9);
11    GPIOA_ModeCfg(GPIO_Pin_8, GPIO_ModeIN_PU); // RXD-配置上拉
12    GPIOA_ModeCfg(GPIO_Pin_9, GPIO_ModeOut_PP_5mA); // TXD-配置推挽
13    UART1_DefInit();
14    /*使能接收中断,接收错误中断*/
15    UART1_INTCfg( ENABLE, RB_IER_RECV_RDY|RB_IER_LINE_STAT );
16    // NVIC_SetPriority(TMR0_IRQn,18); //设置中断优先级(也可以不设置)
17    NVIC_EnableIRQ( UART1_IRQn );

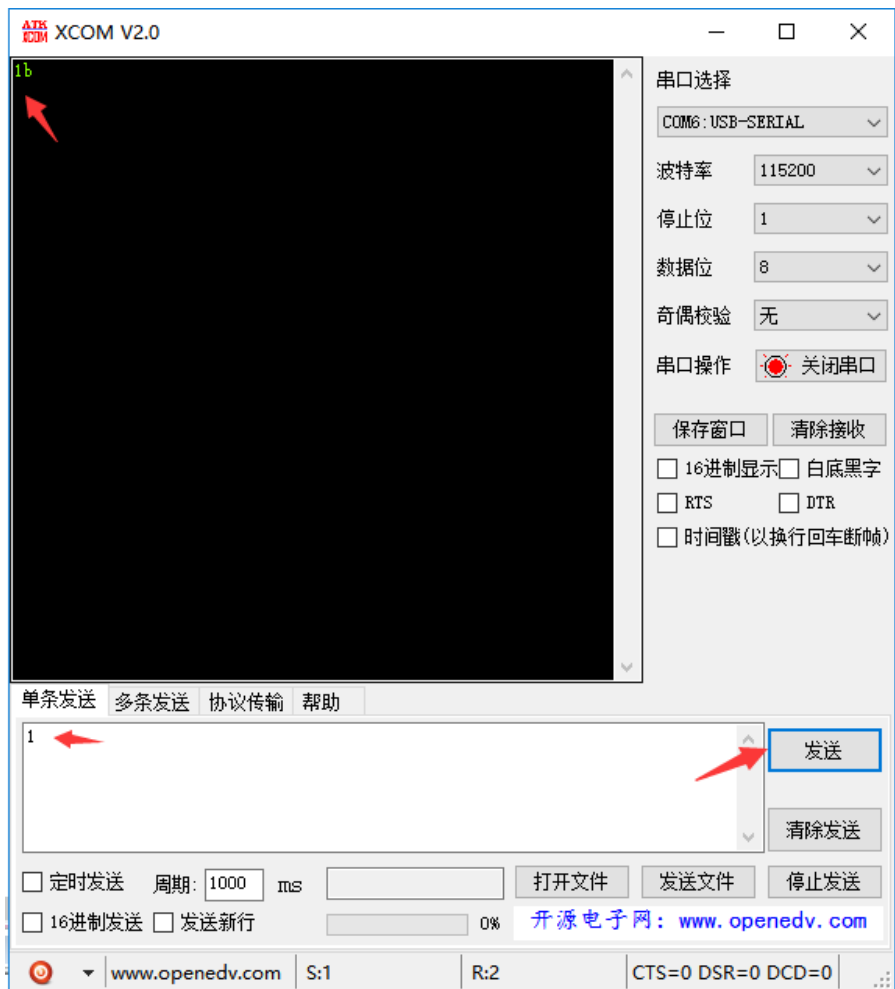
```

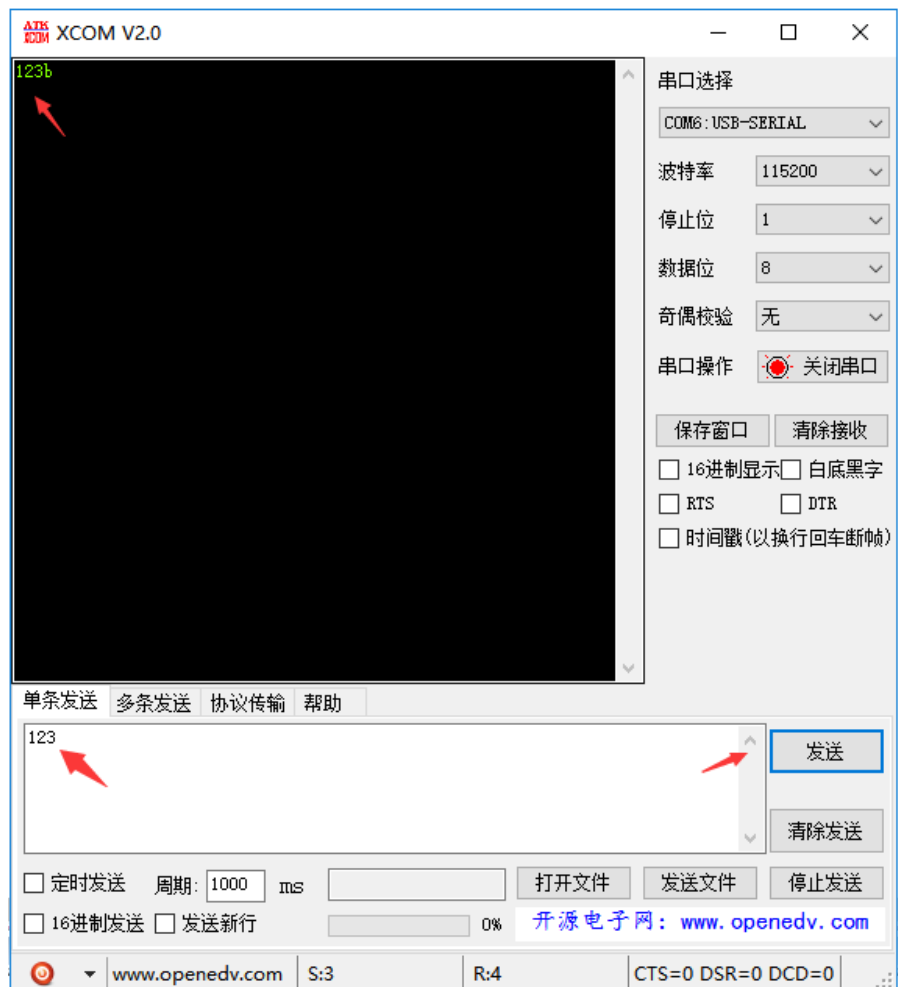


```

2  * File Name      : CH57x_uart1.c
3  * Author         : WCH
4  * Version        : V1.0
5  * Date           : 2018/12/15
6  * Description
7  *****
8
9  #include "CH57x_common.h"
10
11  /******
12  * Function Name  : UART1_DefInit
13  * Description    : 串口默认初始化配置
14  * Input         : None
15  * Return        : None
16  *****
17  void UART1_DefInit( void )
18  {
19      UART1_BaudRateCfg( 115200 );
20      R8_UART1_FCR = (1<<6) | RB_FCR_TX_FIFO_CLR | RB_FCR_RX_FIFO_CLR | (RB_FCR_F
21      R8_UART1_LCR = RB_LCR_WORD_SZ;
22      R8_UART1_IER = RB_IER_TXD_EN;
23      R8_UART1_DIV = 1;
24  }
25

```





注意:接收数据的最后一个字节是在空闲中断事件里面读取

```
22 }
23 /*串口中断*/
24 void UART1_IRQHandler(void)
25 {
26     char data;
27     switch( UART1_GetITFlag() )
28     {
29         case UART_II_LINE_STAT:           // 线路状态错误
30             UART1_GetLinSTA();
31             break;
32         case UART_II_RECV_RDY:           // 数据达到设置触发点
33             data = UART1_RecvByte();
34             UART1_SendByte(data);
35             break;
36         case UART_II_RECV_TOUT:           // 接收超时，暂时一顿数据接收完成
37             data = UART1_RecvByte();
38             UART1_SendByte(data);
39             UART1_SendByte('b');//打印一个字符,说明进入了此函数
40             break;
41         case UART_II_THR_EMPTY:           // 发送缓存区空，可继续发送
42             break;
43         case UART_II_MODEM_CHG:           // 只支持串口0
44             break;
45         default:
46             break;
47     }
48 }
```

空闲中断默认的时间

9.3.3 串口接收

串口接收数据可用中断 UART_II_RECV_RDY 是指接收 FIFO 中的已有数据字节数已经到或超过由 FCR 寄存器的 RB_FCR_FIFO_TRIG 设置选择的 FIFO 触发点。当从 RBR 读取数据使 FIFO 字数低于 FIFO 触发点时，该中断被清除。

串口接收数据超时中断 UART_II_RECV_TOUT 是指接收 FIFO 中至少有一个字节的数据，并且从上一次串口接收到数据和从上一次被系统取走数据开始，已经等待了相当于接收 4 个数据的时间。当再次接收到一个新的数据后，该中断被清除，或者当单片读取一次 RBR 寄存器后，该中断也能被清除。当接收 FIFO 全空时，LSR 寄存器的 RB_LSR_DATA_RDY 位为 0；当接收 FIFO 中有数据时，RB_LSR_DATA_RDY 位为 1 有效。

在中断触发方式下，当收到串口接收数据超时的中断后，可以读取 R8_UARTx_RFC 寄存器查询当前 FIFO 中剩余数据计数，直接读取全部数据，或者不断查询 LSR 寄存器的 RB_LSR_DATA_RDY，如果此位有效则读数据，直到此位无效。当收到串口接收数据可用的中断后，可以先从 RBR 寄存器一次性读取 RB_FCR_FIFO_TRIG 设定字节个数的数据，或者也可以根据 RB_LSR_DATA_RDY 位和 R8_UARTx_RFC 寄存器读取当前 FIFO 中所有数据。

在查询方式下，可以根据 LSR 寄存器的 RB_LSR_DATA_RDY 位判断接收 FIFO 是否为空，或读取 R8_UARTx_RFC 寄存器获取当前 FIFO 中数据计数，来获取串口接收的所有数据。

使用中断发送数据

```
CH579SFR.h CH57x_uart.h CH57x_timer0.c main.c CH57x_uart1.c CH57x_sys.c CH57x_gpio.h
4 __align(4) UINT8 TxBuffer[ MAX_PACKET_SIZE ]; // OUT, must even address
5
6 char send_len = 0;
7 char send_buff[11]="987654321\n\r";
8 int main()
9 {
10  /*配置串口1*/
11  GPIOA_SetBits(GPIO_Pin_9);
12  GPIOA_ModeCfg(GPIO_Pin_8, GPIO_ModeIN_FU); // RXD-配置上拉输入
13  GPIOA_ModeCfg(GPIO_Pin_9, GPIO_ModeOut_PP_5mA); // TXD-配置推挽输出，注意先让IO口输出高电平
14  UART1_DefInit();
15  /*使能接收中断,接收错误中断*/
16  UART1_INTCfg( ENABLE, RB_IER_RECV_RDY|RB_IER_LINE_STAT );
17  // NVIC_SetPriority(TMR0_IRQn,18); //设置中断优先级(也可以不设置,默认14)
18  NVIC_EnableIRQ( UART1_IRQn );
19
20  while(1)
21  {
22      send_len = 11;
23      UART1_INTCfg( ENABLE, RB_IER_THR_EMPTY ); //使能中断发送
24      DelayMs(1000);
25  }
26
27  /*串口中断*/
28  void UART1_IRQHandler(void)
29  {
30      char data;
31      switch( UART1_GetITFlag() )
32      {
33          case UART_II_LINE_STAT: // 线路状态错误
34              UART1_GetLinSTA();
35              break;
36          case UART_II_RECV_RDY: // 数据达到设置触发点
37              data = UART1_RecvByte();
38              break;
39          case UART_II_RECV_TOUT: // 接收超时，暂时一帧数据接收完成
40              data = UART1_RecvByte();
41              break;
42          case UART_II_THR_EMPTY: // 发送缓存区空，可继续发送
43              if(send_len>0)
44              {
45                  send_len--;
46                  UART1_SendByte(send_buff[send_len]);
47              }
48              else
49              {
50                  UART1_INTCfg( DISABLE, RB_IER_THR_EMPTY ); //关闭中断发送
51              }
52              break;
53      }
54  }
```



```
char send_len = 0;
char send_buff[11]="987654321\n\r";
int main()
{
    /*ÃÄÃ'©Ü1*/
}
```



```

GPIOA_SetBits(GPIO_Pin_9);
GPIOA_ModeCfg(GPIO_Pin_8, GPIO_ModeIN_PU); // RXD-ÄäÖÄÉíÄÊaÊë
GPIOA_ModeCfg(GPIO_Pin_9, GPIO_ModeOut_PP_5mA); // TXD-ÄäÖÄíÊíÊä³ô£¬×çÖäí
UART1_DefInit();
/*Ê¹ÄÜ»ÖÖÐ¶ü,»ÖÖÖ´íóÖÐ¶ü*/
UART1_INTCfg( ENABLE, RB_IER_RECV_RDY|RB_IER_LINE_STAT );
// NVIC_SetPriority(TMR0_IRQn,18); //ÊÖÖÖ¶üíÖÄíÊ¶¶(Ö²çÊÖÖ²»ÊÖÖÄ,Ä-Êí14)
NVIC_EnableIRQ( UART1_IRQn );

while(1)
{
    send_len = 11;
    UART1_INTCfg( ENABLE, RB_IER_THR_EMPTY ); //Ê¹ÄÜÖÐ¶ü·çÊí
    DelayMs(1000);
}
}
/*´ÖçÖÖÐ¶ü*/
void UART1_IRQHandler(void)
{
    char data;
    switch( UART1_GetITFlag() )
    {
        case UART_II_LINE_STAT: // íÊÄ·×´í¬´íó
            UART1_GetLinSTA();
            break;

        case UART_II_RECV_RDY: // ÊýÝ´íµÊÖÄ´ý·çµä
            data = UART1_RecvByte();
            break;

        case UART_II_RECV_TOUT: // »ÖÖÖ³-Ê±£¬ÖýÊ±ÖÖÖ;ÊýÝ»ÖÖÖíÊ³Ê
            data = UART1_RecvByte();
            break;

        case UART_II_THR_EMPTY: // ·çÊí»°´æÇøÖ£¬çÊíÐø·çÊí
            if(send_len>0)
            {
                send_len--;
                UART1_SendByte(send_buff[send_len]);
            }
            else
            {
                UART1_INTCfg( DISABLE, RB_IER_THR_EMPTY ); //¹Ø±ÖÖÐ¶ü·çÊí
            }

            break;

        case UART_II_MODEM_CHG: // Ö»ÖÖ³Ö´ÖçÖÖ
            break;

        default:
            break;
    }
}

```

分类: [CH579M学习开发](#)

好文要顶

关注我

收藏该文



杨奉武

关注 - 1

粉丝 - 623

0

0

« 上一篇: [101-CH579M学习开发-基本外设-定时器](#)


[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】百度智能云特惠：新用户首购云服务器低至0.7折，个人企业同享

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

【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】投资训练营：一杯咖啡的价格，教你学会投资，增加被动收入

【推荐】加州大学伯克利分校高管教育：大数据与数学科学-在线课程

【推荐】和开发者在一起：华为开发者社区，入驻博客园科技品牌专区

编辑推荐：

- [AspNetCore & MassTransit Courier 实现分布式事务](#)
- [传统.NET 4.x应用容器化体验（1）](#)
- [EF Core3.1 CodeFirst 动态自动添加表和字段的描述信息](#)
- [探索互斥锁 Mutex 实现原理](#)
- [我用段子讲.NET之依赖注入其二](#)



最新新闻：

- [人类评估已不是NLG的最佳标准，华盛顿大学提出新观点遭网友质疑](#)
- [一行命令装下所有「炼丹」工具及依赖项，就靠这个免费软件源了](#)
- [腾讯联合国家天文台启动探星计划，用AI寻找脉冲星线索](#)
- [北上广的年轻人，越来越“不要脸”](#)
- [淘特为何能出淘？](#)
- » [更多新闻...](#)



单片机,物联网,上位机,...

扫一扫二维码，入群聊。