

# STC15W4K32S4单片机 串口通信

## 串行口

STC15W4K32S4系列单片机有4个串行口，串行口1有4种工作方式，其中两种方式波特率可变，两种波特率固定。串行口2，3，4只有2种工作方式，均为波特率可变。

### UART1

#### 寄存器

串行口控制寄存器(Serial Control Register)(可位寻址)

| SFR name | Address | bit  | B7     | B6  | B5  | B4  | B3  | B2  | B1 | B0 |
|----------|---------|------|--------|-----|-----|-----|-----|-----|----|----|
| SCON     | 98H     | name | SM0/FE | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |

- SM0/FE:当PCON中的SMOD0=0时，该位和SM1共同指定串行通信的工作方式

#### SM0 SM1 工作方式 功能说明

| SM0 | SM1 | 工作方式 | 功能说明         |
|-----|-----|------|--------------|
| 0   | 0   | 方式0  | 同步移位串行方式     |
| 0   | 1   | 方式1  | 8位UART,波特率可变 |
| 1   | 0   | 方式2  | 9位UART       |
| 1   | 1   | 方式3  | 9位UART,波特率可变 |

- SM2:方式2或方式3多机通信控制位 使用方式0，方式1时，该位应为0。
- REN:允许禁止串行控制接收位
- TB8:方式2或方式3中要发送的第九位数据 使用方式0，方式1时，该位不用
- RB8:在方式2或方式3中为要接收的第九位数据 使用方式0，方式1时，该位不用
- TI:发送中断请求标志位，方式0中发送数据第八位结束后，内部硬件自动置位，向cpu请求中断，响应中断后需软件清零。其他工作方式则在停止位开始发送时硬件置位，响应中断后需软件清零。
- RI:接收中断请求标志位，方式0中接收数据第八位结束后，内部硬件自动置位，向cpu请求中断，响应中断后需软件清零。其他工作方式则在接收到停止位的中间时刻硬件置位，向cpu请求中断，响应中断后需软件清零。

电源控制寄存器(Power Control Register)(不可位寻址)

| SFR name | Address | bit  | B7   | B6    | B5   | B4  | B3  | B2  | B1 | B0  |
|----------|---------|------|------|-------|------|-----|-----|-----|----|-----|
| PCON     | 87H     | name | SMOD | SMOD0 | LVDF | POF | GF1 | GF0 | PD | IDL |

- SMOD:波特率选择位，SMOD=1时；则使串行通信方式1,2,3的波特率加倍。
- SMOD0:帧检测有效控制位，SMOD0=1时，SCON寄存器的SM0/FE位用于FE(帧错误检测)功能
- 其他与UART1无关

串行口数据缓冲寄存器(serial data buffer)

- 两个数据缓冲器读SBUF 和 写SBUF共用一个地址

辅助寄存器(Auxiliary Register) (不可位寻址)

#### 定时器2配置

| SFR name | Address | bit  | B7    | B6    | B5        | B4  | B3      | B2    | B1     | B0    |
|----------|---------|------|-------|-------|-----------|-----|---------|-------|--------|-------|
| AUXR     | 8EH     | name | T0x12 | T1x12 | UART_M0x6 | TR2 | T2_C/~T | T2x12 | EXTRAM | S1ST2 |

- 使用定时器1,2波特率发生器时需对该寄存器进行配置
- T0x12: 定时器0速度控制位 AUXR &= ~(1<<7);/12分频
- T1x12: 定时器0速度控制位 AUXR &= ~(1<<7);/12分频

- T2R:定时器2允许控制位
- T2\_C/~T:控制定时器2用做定时器或计数器
- T2x12: 定时器2速度控制位  $AUXR \&= \sim(1 \ll 2); // 12 \text{分频}$   $AUXR |= (1 \ll 2); // 1 \text{分频}$
- EXTRAM: 内部/外部RAM存取控制位 0允许, 1禁止
- **S1S2:UART1 选择T1,T2作为波特率发生器控制位 0:T1; 1:T2,此时T1释放, 可以作为独立定时器使用。UART1默认选择T2作为波特率发生器**

## 定时器2的寄存器T2H, T2L

- 用于保存重装时间常数。

## 从机地址控制寄存器SADEN和SADDR

- ???

## 中断允许寄存器(可位寻址)

| SFR name | Address | bit  | B7 | B6   | B5   | B4 | B3  | B2  | B1  | B0  |
|----------|---------|------|----|------|------|----|-----|-----|-----|-----|
| IE       | A8H     | name | EA | ELVD | EADC | ES | ET1 | EX1 | ET0 | EX0 |

- EA:总中断控制允许位
- ES:串行口中断控制允许位

## 中断优先级控制寄存器 (可位寻址)

| SFR name | Address | bit  | B7   | B6   | B5   | B4        | B3  | B2  | B1  | B0  |
|----------|---------|------|------|------|------|-----------|-----|-----|-----|-----|
| IP       | B8H     | name | PPCA | PLVD | PADC | <b>PS</b> | PT1 | PX1 | PT0 | PX0 |

- PS:串行口中断优先级控制位 PS = 1;//最高优先级 (优先级1)

## 辅助寄存器AUXR1(P\_SW1)(不可位寻址)

### UART1输出引脚配置

| Mnemonic    | Address | name                 | B7           | B6           | B5     | B4     | B3     | B2     | B1 | B0  |
|-------------|---------|----------------------|--------------|--------------|--------|--------|--------|--------|----|-----|
| AUXR1 P_SW1 | A2H     | Auxiliary register 1 | <b>S1_S1</b> | <b>S1_S0</b> | CCP_S1 | CCP_S0 | SPI_S1 | SPI_S0 | 0  | DPS |

Reset Value : 0000 0000

\*S1\_S1与S1\_S0选择串行口输出引脚。

| S1_S1 | S0_S0 | 串行口在P1~P3之间来回切换                                 |
|-------|-------|---|
| 0     | 0     | UART1在[P30/RxD][P31/TxD]                        |
| 0     | 1     | UART1在[P36/RxD_2][P37/TxD_2]                    |
| 1     | 0     | UART1在[P16/RxD_3/XTAL2][P17/TxD_3/XTAL1]需使用内部时钟 |
| 1     | 1     | 无效、   |

## 时钟分频寄存器CLK\_DIV

### 中继广播方式设置

| Mnemonic        | Address | name | B7      | B6      | B5   | B4           | B3      | B2     | B1     | B0    |
|-----------------|---------|------|---------|---------|------|--------------|---------|--------|--------|-------|
| CLK_DIV (PCON2) | 97H     | —    | MCKO_S1 | MCKO_S0 | ADRJ | <b>Tx_RX</b> | MCLKO_2 | CLK_S2 | CLK_S1 | CLKS0 |

\*TxRx:中继广播方式设置位,为1时TxD管脚的对外输出实时反映Rx端口输入的电平状态。

## UART1 工作方式1配置

```
#include<STC15Fxxxx.h>
unsigned char  uart1busy = 0;
unsigned char read_count = 0;
unsigned char send_count = 0;
char read[20];
char send[20] = {"mou i chi do"};
```

```
void Uart1_Init()
{
//串口配置
PS = 1; //高优先级中断
```

```

SCON = (SCON & 0x3f) | (1<<6) ;//工作方式1

//定时器2配置
AUXR &= ~(1<<4); //关闭T2;
AUXR |= 0X01; //选择T2位UART1的波特率发生器
AUXR &= ~(1<<3); //T2工作在定时模式
AUXR |= (1<<2); //1T不分频
TH2 = 0xFE>>8; // 1T 波特率=(SYSclk / (65536 - [RL_TH,RL_TL]))/4), 12分频 /12
TL2 = 0XE0; //11.0592M晶振 波特率
IE2 &= ~(1<<2); //禁止T2中断
AUXR |= (1<<4); //启动T2
//串口配置
P_SW1 = (P_SW1 & 0x3f) | (0x40 & 0xc0); //shiyong使用P36 P37引脚
CLK_DIV &= ~(1<<4); //串口1为正常工作模式, 不中继
ES = 1; //串口1中断允许
EA = 1; //总中断允许
}
void Uart1_sendchar(unsigned char usdata)
{
    while(uart1busy);
    uart1busy = 1;
    SBUF = usdata;
}
void Uart1_sendstring(unsigned char *usdata,unsigned int slen)
{
    unsigned int len;
    for(len=0;len<slen;len++)
    {
        Uart1_sendchar(usdata[len]);
        uart1busy = 0;
    }
}

void main(void)
{
    EA = 1;
    Uart1_Init();
    Uart1_sendstring(send,20);
    while(1);
}

//中断函数
void Uart1_int() interrupt 4
{
    if(RI)
    {
        RI = 0;
        read[read_count ++]= SBUF;
        if(read_count >= 20)
            read_count = 0;
    }
    if(TI)
    {
        TI = 0;
        uart1busy = 0;
    }
}

```

## UART2

- 对于15系列单片机而言，串行口2只能使用定时器2作为波特率发生器，不能选择其他定时器作为波特率发生器

### 寄存器

#### 串行口2的控制寄存器S2CON(不可位寻址)

- 只有2种工作方式，由 S2COM 的 bit7 S2SM0 控制: =0,8位UART波特率可变。=1,9位uart 波特率可变。
- bit6保留，该位复位后为1
- bit5 S2SM2
- bit4 S2REN
- bit3 S2TB8
- bit2 S2RB8
- bit1 S2TI
- bit0 S2RI

#### 串行口2的数据缓冲寄存器S2BUF(不可位寻址)

## 定时器2配置相关寄存器

### 中断允许寄存器2 IE2(不可位寻址)

- bit0 ES2

EA = 1;

### 外围设备功能切换控制寄存器2(P\_SW2)(不可位寻址)

#### UART2输出引脚配置

- bit0 S2\_S=0:[P1.0/RxD2,P1.1/TXD2];=1:[P4.6/RxD2\_2,P4.7/TXD2\_2]

配置和使用同UART1,波特率计算公式也一样。

## UART3

串行口3默认选择定时器2为波特率发生器，也可以选择定时器3作为波特率发生器

### 寄存器

#### 串行口3的控制寄存器S3CON(不可位寻址)

- 只有2种工作方式，由 S3COM 的 bit7 S3SM0 控制: =0,8位UART波特率可变。=1,9位uart 波特率可变。
- bit6 S3ST3 串行口3波特率发生器选择位 = 0:选择定时器二作为其波特率发生器； = 1:选择定时器3作为波特率发生器
- bit5 S3SM2
- bit4 S3REN
- bit3 S3TB8
- bit2 S3RB8
- bit1 S3TI
- bit0 S3RI

#### 串行口3的数据缓冲寄存器S3BUF(不可位寻址)

#### T4T3M(不可位寻址)

- bit3 T3R 定时器3运行控制位
- bit2 T3\_C/~T 定时器3用作定时器/计数器控制位
- bit1 T3x12 定时器3是否分频控制位 =0:12分频；

### TH3,TL3

#### IE2(不可位寻址)

- bit5 ET3 定时器3的中断允许位
- bit3 ES3 串行口3中断允许位

IE: EA = 1;

#### PSW\_2(不可位寻址)

#### UART3输出引脚配置

- bit1 S3\_S=0:[P0.0/RxD3,P0.1/TxD3] =1:[P5.0/RxD3\_2,P5.1/TxD3\_2]

## UART4

串行口4默认选择定时器2为波特率发生器，也可以选择定时器4作为波特率发生器

### 寄存器

#### 串行口4的控制寄存器S4CON(不可位寻址)

- 只有2种工作方式，由 S4COM 的 bit7 S3SM0 控制: =0,8位UART波特率可变。=1,9位uart 波特率可变。
- bit6 S4ST3 串行口3波特率发生器选择位 = 0:选择定时器二作为其波特率发生器； = 1:选择定时器4作为波特率发生器
- bit5 S4SM2
- bit4 S4REN
- bit3 S4TB8
- bit2 S4RB8

- bit1 S4TI
- bit0 S4RI

串行口4的数据缓冲寄存器S4BUF(不可位寻址)

T4T3M(不可位寻址)

- bit7 T4R 定时器4运行控制位
- bit6 T4\_C/~T 定时器4用作定时器/计数器控制位
- bit5 T4x12 定时器4是否分频控制位 =0:12分频;

TH4,TL4

IE2(不可位寻址)

- bit6 定时器4的中断允许位
- bit4 ES4 串行口4中断允许位

IE: EA = 1;

PSW\_2(不可位寻址)

UART3输出引脚配置

- bit2 S3\_S =0:[P0.2/RxD4,P0.3/TxD4] =1:[P5.2/RxD4\_2,P5.3/TxD4\_2]