

# KF8F3132——Usart 模块

## 引言

本应用笔记提供了 KF8F3132—Usart 模块相关的配置信息以及如何能够快速的理解并上手使用该模块的一些配置方式。

本应用笔记须与 KF8F3132 数据手册结合使用。

## 寄存器

寄存器使用说明：

OSCCTL： 系统控制寄存器

寄存器OSCCTL：系统频率控制寄存器(地址:2FH)

	bit7						bit0
复位值 0011 ----	CLKOE	IRCS2	IRCS1	IRCS0	-	-	-
	R/W	R/W	R/W	R/W	U	U	U

图注： R = 可读      W = 可写      P = 可编程      U = 未使用  
 - = 读为0      x = 状态未知

TR1： P1 口方向控制寄存器（输入/输出）

TR1： P1口方向控制寄存器(地址: 27H)

	bit7						bit0
复位值 --11 1111	-	-	TR15	TR14	TR13	TR12	TR11
	U	U	R/W	R/W	R/W	R/W	R/W

RSCTL： 接收状态和控制寄存器

寄存器11.3: RSCTL: 接收状态和控制寄存器(地址:120H)

	bit7						bit0
复位值 0000 000x	SPEN	RX9	SRXEN	CRXEN	ADREN	FRER	OVFER
	R/W	R/W	R/W	R/W	R/W	R	R

BRCTL： 波特率控制寄存器

寄存器11.1: BRCTL: 波特率控制寄存器(地址:123H)

	bit7						bit0
复位值 0100 0000	ABRDOV F	RCIDLF	SCKPS	BRG16	-	-	WUEN
	R/W	R	R/W	R/W	R/W	R/W	R/W

TSCTL： 发送状态和控制寄存器

寄存器11.2: TSCTL: 发送状态和控制寄存器(地址:124H)

	bit7						bit0
复位值 0000 0010	CSRS	TX9	TXEN	SYNC	SENDB	HBRG	TXSRS
	R/W	R/W	R/W	R/W	R/W	R/W	R

EUBRGL： Usart 波特率数据寄存器低字节

EUBRGH： Usart 波特率数据寄存器高字节

TXSDR： Usart 发送数据寄存器

RXSDR: Usart 接收数据寄存器

位操作使用说明:

8 位单片机支持对寄存器的位进行直接的操作，因此在使用的过程中不仅可以通过给寄存器赋值来达到想要的配置，同时还可以直接对位进行操作来达到需要的配置。

以下是对程序中使用到的位进行说明:

UARTPIN: Usart 模块功能引脚切换位

RCIF: Usart 接收中断标志位

RCIE: Usart 接收中断使能位

PUIE: 外设中断使能位

AIE: 总中断使能位

TXSRS: 发送移位寄存器状态位

OVFER: 溢出错误位

CRXEN: 连续接收使能位

开发环境：chipon IDE

功能简述：使用 Usart 模块实现多字节连续的收发实验。电脑端的串口助手发送 16 进制的 1H，开发板返回“HELLO!”。若串口助手发送十六进制的 2H，开发板返回“ByeBye!”。

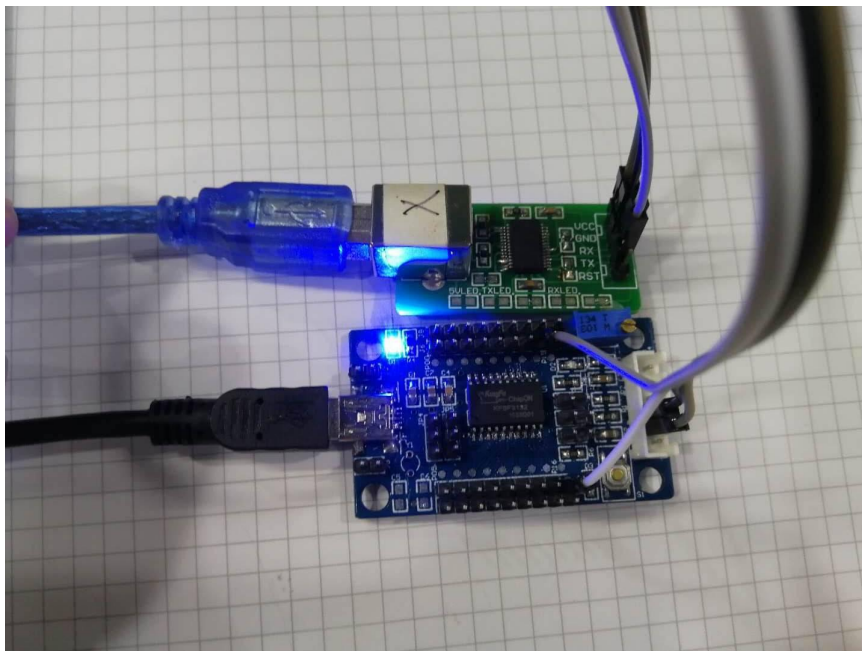
硬件说明：实验时，需要用 USB—USART 模块连接 KF8F3132 开发板和 PC 端的串口助手。串口助手设置为：波特率 9600、8 位数据位、1 位停止位、无奇偶校验位。

USB—USART 模块的 TX——开发板的 P14

USB—USART 模块的 RX——开发板的 P16

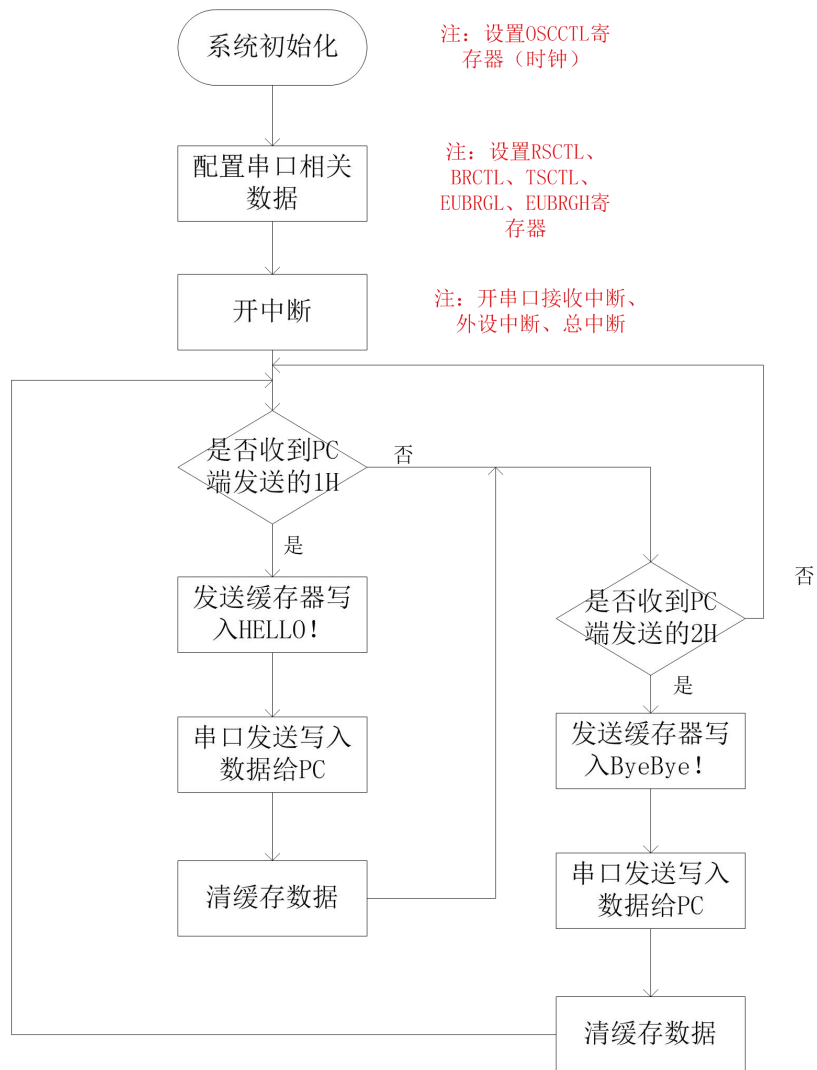
USB—USART 模块的 GND——开发板的 GND

USB—USART 模块的 5V——开发板的 VCC





## Usart 样例流程图:



注:

串口相关寄存器具体的配置可以看程序部分, 有详细的注释。PC 端串口助手的数据设置可查看截图中的具体设置。因串口属于外设模块所以要使能外设中断。

测试板判断是否收到 PC 端的发送数据是通过中断方式来判断的, 当接受到数据时产生接收中断, 读取接收到的数据后做判断。

样例程序如下：

时钟、IO 口及 Usart 模块初始化部分：

```
void Init_mcu()
{
    OSCCTL=0x70;//16M系统时钟

    TR1 =0x10;

    UARTPIN =0;//RX—P14,TX—P16

    RSCTL =0x90;//使能串口模块、使能接收器
    BRCTL =0x40;//接收器空闲
    TSCTL =0x22;//使能发送功能、移位寄存器空
    EUBRGL= 0x19;//波特率寄存器设置为25，当前波特率是9600，波特率为SCLK/
    【m*（EUBRGH: EUBRGL+1）】，m的设置方法参考手册
    EUBRGH =0;

    RCIF =0;
    RCIE =1;//打开接收中断

    PUIE=1;
    AIE=1;//使能全局变量
}
```

Usa

rt 多字节发送函数：

```
void Uart_send(uchar k)
{
    uchar i=0;
    for(i=0;i<k;i++)
    {
        TXSDR=BUF1[i];
        while(TXSRS==0);
    }
}
```

发送缓存写 HELLO:

```
void Write_hello()
{
    uchar i=0;
    for(i=0;i<10;i++)
    {
        BUF1[i]=0;
    }

    BUF1[0]='H';
    BUF1[1]='E';
    BUF1[2]='L';
    BUF1[3]='L';
    BUF1[4]='O';
    BUF1[5]='!';
    BUF1[6]='\r';
}
```

## 发送缓存写 ByeBye:

```
void Write_Bye()
{
    uchar i=0;
    for(i=0;i<10;i++)
    {
        BUF1[i]=0;
    }

    BUF1[0]='B';
    BUF1[1]='y';
    BUF1[2]='e';
    BUF1[3]='B';
    BUF1[4]='y';
    BUF1[5]='e';
    BUF1[6]='!';
    BUF1[7]='\n';
}
```

## 中断函数:

```
void int_fun0() __interrupt (0)
{
    if(RCIF)
    {
        if(OVFER==1)
        {
            CRXEN=0;    // 清溢出错误
            CRXEN=1;    // 重新使能接收
        }
        Rev_Temp=RXSDR; // 清零 RXIF
    }
}
```

## 主函数:

```
void main()
{
    Init_mcu();

    while(1)
    {
        if(Rev_Temp==1)//收到十六进制的1H时，返回“HELLO!”
        {
            Write_hello();
            Uart_send(7);
            Rev_Temp=0;
        }

        if(Rev_Temp==2)//收到十六进制的1H时，返回“ByeBye!”
        {
            Write_Bye();
            Uart_send(8);
            Rev_Temp=0;
        }
    }
}
```



## 模块使用注意事项：

1、USART 模块可通过 PINSET 寄存器的 UARTPIN 选择相关功能引脚位置。

当 UARTPIN=0 时，RX/DT 在 P1.4 脚，TX/CK 在 P1.6 脚；

当 UARTPIN=1 时，RX/DT 在 P0.0 脚，TX/CK 在 P0.1 脚。

注：14 或 16 引脚使用 USART 功能时 UARTPIN 必须配置为 1。

2、波特率发生器（BRG）是 8 位或者 16 位定时器，可以自行选择工作模式。默认情况下，BRG 工作在 8 位模式。

3、波特率发生器的时钟是系统主时钟 SCLK，波特率计算公式位：

$$\text{目标波特率} = \frac{SCLK}{m \times ([EUBRGH : EUBRGL] + 1)}$$

其中 m 为倍频器选择，下面是 m 选择表：

配置位			BRG/USART 模式	倍频器 m
SYNC	BRG16	HBRG		
0	0	0	8 位/异步	64
0	0	1	8 位/异步	16
0	1	0	16 位/异步	
0	1	1	16 位/异步	4
1	0	x	8 位/同步	
1	1	x	16 位/同步	

注：x 是无关位。

4、在帧格式中，一个字符由 4 部分组成：起始位，数据位，奇偶校验位和停止位。