单线半双工通信简单例程（STM32与电脑通信）

单线半双工通信的官方资料

单线半双方模式通过设置**USART_CR3**寄存器的**HDSEL**位选择。在这个模式里，下面的位必须保持清零状态：
● **USART_CR2**寄存器的**LINEN**和**CLKEN**位
● **USART_CR3**寄存器的**SCEN**和**IREN**位
USART可以配置成遵循单线半双工协议。在单线半双工模式下，TX和RX引脚在芯片内部互连。使用控制位"HALF DUPLEX SEL"(USART_CR3中的HDSEL位)选择半双工和全双工通信。
当HDSEL为'1'时
● RX不再被使用
● 当没有数据传输时，TX总是被释放。因此，它在空闲状态的或接收状态时表现为一个标准I/O口。这就意味该I/O在不被USART驱动时，必须配置成悬空输入(或开漏的输出高)。
除此以外，通信与正常USART模式类似。由软件来管理线上的冲突(例如通过使用一个中央仲裁器)。特别的是，发送从不会被硬件所阻碍。当TE位被设置时，只要数据一写到数据寄存器上，发送就继续。

关于上文提到的寄存器及对应的位

1.首先是CR3：

25.6.6 控制寄存器 3(USART_CR3)

地址偏移：0x14
复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

	注：UART4和UART5上不存在这一位。
位5	**SCEN**: 智能卡模式使能 (Smartcard mode enable) 该位用来使能智能卡模式 0：禁止智能卡模式； 1：使能智能卡模式。 注：UART4和UART5上不存在这一位。
位4	**NACK**：智能卡NACK使能 (Smartcard NACK enable) 0：校验错误出现时，不发送NACK； 1：校验错误出现时，发送NACK。 注：UART4和UART5上不存在这一位。
位3	**HDSEL**：半双工选择 (Half-duplex selection) 选择单线半双工模式 0：不选择半双工模式； 1：选择半双工模式。
位2	**IRLP**：红外低功耗 (IrDA low-power) 该位用来选择普通模式还是低功耗红外模式 0：通常模式； 1：低功耗模式。
位1	**IREN**：红外模式使能 (IrDA mode enable) 该位由软件设置或清除。 0：不使能红外模式； 1：使能红外模式。
位0	**EIE**：错误中断使能 (Error interrupt enable) 在多缓冲区通信模式下，当有帧错误、过载或者噪声错误时(USART_SR中的FE=1，或者ORE=1，或者NE=1)产生中断。 0：禁止中断； 1：只要USART_CR3中的DMAR=1，并且USART_SR中的FE=1，或者ORE=1，或者NE=1，则产生中断

对应代码：USART1->CR3|=1<<3;

USART1->CR3&=~(1<<1);
USART1->CR3&=~(1<<5);
2.然后是CR2:

25.6.5 控制寄存器 2(USART_CR2)

地址偏移：0x10

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	保留	LBDIE	LBDL	保留	ADD[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw		rw	rw	rw	rw

位31:15 保留位，硬件强制为0。

位14	**LINEN**：LIN模式使能 (LIN mode enable) 该位由软件设置或清除。 0：禁止LIN模式； 1：使能LIN模式。 在LIN模式下，可以用USART_CR1寄存器中的SBK位发送LIN同步断开符(低13位)，以及检测LIN同步断开符。
位13:12	**STOP**：停止位 (STOP bits) 这2位用来设置停止位的位数 00：1个停止位； 01：0.5个停止位； 10：2个停止位； 11：1.5个停止位； 注：UART4和UART5不能用0.5停止位和1.5停止位。
位11	**CLKEN**：时钟使能 (Clock enable) 该位用来使能CK引脚 0：禁止CK引脚； 1：使能CK引脚。 注：UART4和UART5上不存在这一位。

对应代码：
USART1->CR2&=~（1<<14）;
USART1->CR2&=~(1<<11);

修改代码（以实验4 串口实验为模板）

uart_init()部分修改如下：
1.去掉PA10的GPIO_Init部分
2.添加几个寄存器相关代码

```
08    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 3;     //子优先级3
09    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;      //IRQ通道使能
10    NVIC_Init(&NVIC_InitStructure); //根据指定的参数初始化VIC寄存器
11
12     //USART 初始化设置
13
14    USART_InitStructure.USART_BaudRate = bound;//串口波特率
15    USART_InitStructure.USART_WordLength = USART_WordLength_8b;//字长为8位数据格式
16    USART_InitStructure.USART_StopBits = USART_StopBits_1;//一个停止位
17    USART_InitStructure.USART_Parity = USART_Parity_No;//无奇偶校验位
18    USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None
19    USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx; //收发模式
20
21    USART_Init(USART1, &USART_InitStructure); //初始化串口1
22    USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);//开启串口接受中断
23    USART_Cmd(USART1, ENABLE);                    //使能串口1
24
25    USART1->CR2&=~(1<<11);
26    USART1->CR2&=~(1<<14);
27    USART1->CR3|=1<<3;
28    USART1->CR3&=~(1<<1);
29    USART1->CR3&=~(1<<5);
30
31  }
32
```

中断部分我添加了“如果单片机收到数据，则灯亮”的代码。
主函数我改成了按键发送：

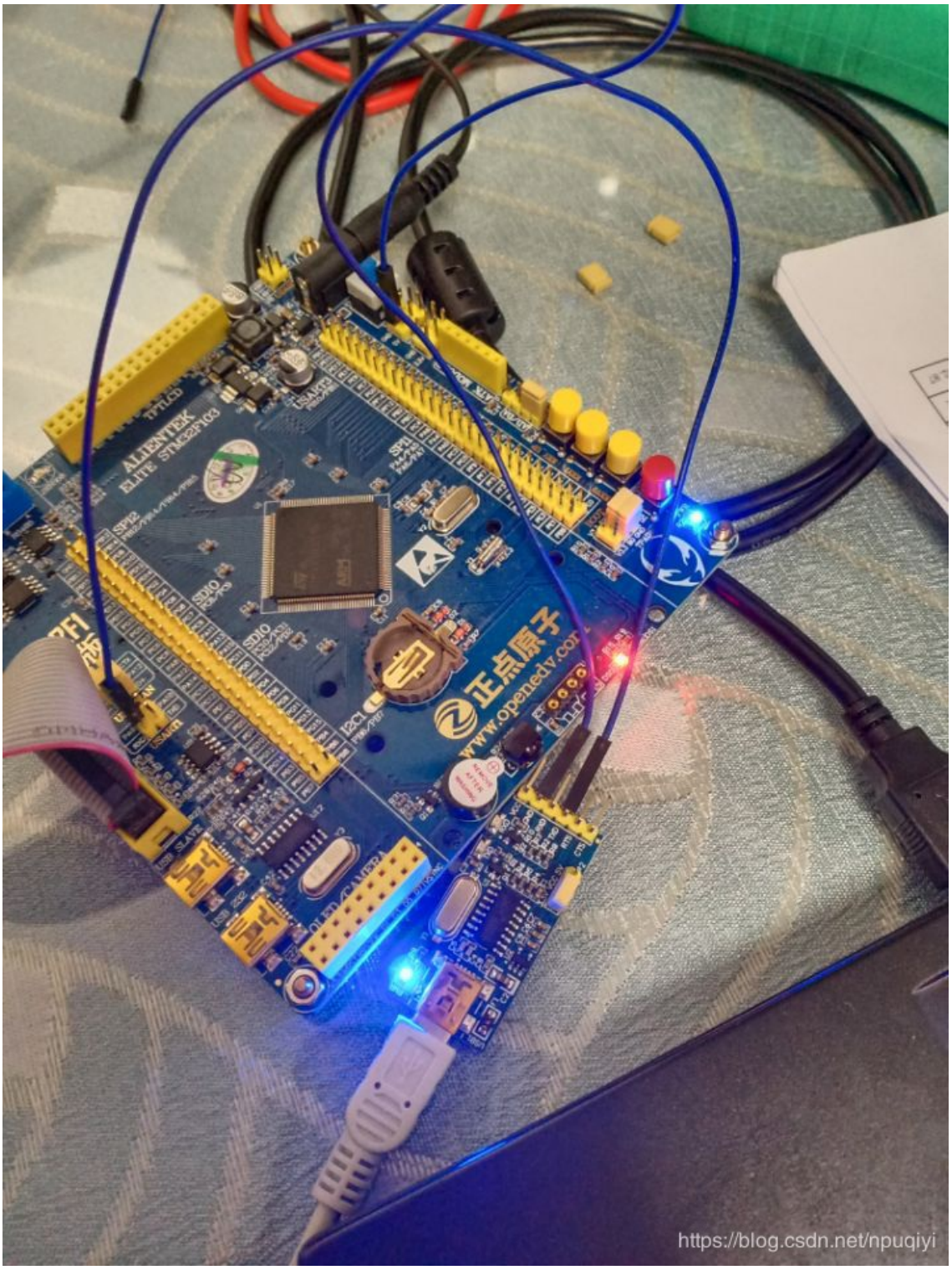
```
int main(void)
{
 u16 t;
 u8 key,test[4]={'T','E','S','T'};
 u16 len;
 u16 times=0;
 delay_init();          //延时函数初始化
 NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2); //设置NVIC中断分组2:2位抢占优先级，2位响应优先级
 uart_init(115200);     //串口初始化为115200
 LED_Init();            //LED端口初始化
 KEY_Init();            //初始化与按键连接的硬件接口
 while(1)
 {
   key=KEY_Scan(0);
   if(key==KEY0_PRES)
   {
     for(t=0;t<4;t++)
     {
       USART_SendData(USART1, test[t]);//向串口1发送数据
       while(USART_GetFlagStatus(USART1,USART_FLAG_TC)!=SET);//等待发送结束
     }

   }else
   {
     times++;
     if(times%30==0)LED0=!LED0;//闪烁LED,提示系统正在运行.
     delay_ms(10);
   }
 }
}
```


注意事项

因为设置为单线半双工通信后，原来的TXD和RXD现在是连接在一起的，所以用TXD发送数据时，同样可以触发中断，（也就是串口发送的数据串口自身会接收到），所以需要软件设置好相关内容。

硬件连接

最后验证时，按KEY0发送数据，PA9接U转串的RXD，或者直接用板载的USB接口，接收数据时，PA9接U转串的TXD。


ALIENTEK
ELITE STM32F103
正点原子
www.openedv.com
https://blog.csdn.net/npuqiyi

作者：NPU_QY