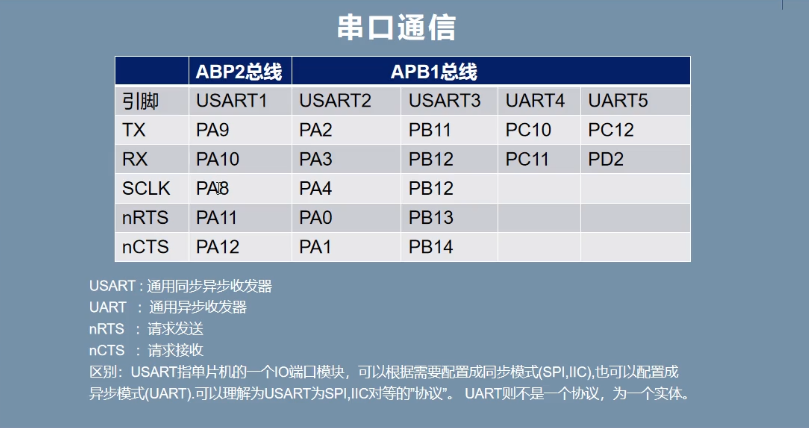
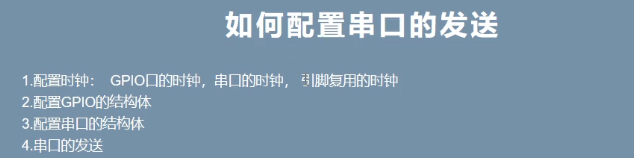
***页数347***











#include "usartset.h"

#include "stm32f10x.h"

void usartconfig(void){

    RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOA, ENABLE);

    RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_USART1, ENABLE);

    RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_AFIO, ENABLE);

*//tx*

    GPIO\_InitTypeDef GPIO\_StructInit;

    GPIO\_StructInit.GPIO\_Mode = GPIO\_Mode\_AF\_PP;

    GPIO\_StructInit.GPIO\_Pin = GPIO\_Pin\_9;

    GPIO\_StructInit.GPIO\_Speed = GPIO\_Speed\_10MHz;

    GPIO\_Init(GPIOA, &GPIO\_StructInit);

*//rx*

    GPIO\_StructInit.GPIO\_Mode = GPIO\_Mode\_IN\_FLOATING;

    GPIO\_StructInit.GPIO\_Pin = GPIO\_Pin\_10;

    GPIO\_Init(GPIOA, &GPIO\_StructInit);

    USART\_DeInit(USART1);

    USART\_InitTypeDef USART\_StructInit;

    USART\_StructInit.USART\_BaudRate = 115200;

    USART\_StructInit.USART\_HardwareFlowControl = USART\_HardwareFlowControl\_None;

    USART\_StructInit.USART\_Mode = USART\_Mode\_Rx  | USART\_Mode\_Tx;

    USART\_StructInit.USART\_Parity = USART\_Parity\_No;

    USART\_StructInit.USART\_StopBits = USART\_StopBits\_1;

    USART\_StructInit.USART\_WordLength = USART\_WordLength\_8b;

    USART\_Init(USART1, &USART\_StructInit);

    USART\_ClearFlag(USART1, USART\_IT\_TXE);

    USART\_ITConfig(USART1, USART\_IT\_RXNE, ENABLE);

    USART\_Cmd(USART1, ENABLE);

    NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_1);

    NVIC\_InitTypeDef NVIC\_InitStruct;

    NVIC\_InitStruct.NVIC\_IRQChannel = EXTI0\_IRQn;

    NVIC\_InitStruct.NVIC\_IRQChannelCmd = ENABLE;

    NVIC\_InitStruct.NVIC\_IRQChannelPreemptionPriority = 0;

    NVIC\_InitStruct.NVIC\_IRQChannelSubPriority = 1;

    NVIC\_Init(&NVIC\_InitStruct);

}

void send\_data(USART\_TypeDef \**USARTx*, uint16\_t *data*){

    USART\_SendData(*USARTx*, *data*);

    while (USART\_GetFlagStatus(*USARTx*,USART\_FLAG\_TXE) ==RESET);

}

void send\_array(uint8\_t \**array*, uint8\_t *len*){

    uint16\_t i;

    for (i = 1; i < *len*;i++){

        send\_data(USART1, *array*[i]);

    }

}

    void USART1\_IRQHandler(void)

{

    int temp;

    if (USART\_GetITStatus(USART1, USART\_IT\_RXNE) != RESET)

    {

        temp = USART\_ReceiveData(USART1);

        if (temp == 1)

        {

            send\_data(USART1, 0x11);

        }

   }

}

**Table 706. 函数 USART\_DeInit**

函数名 USART\_DeInit

函数原形 void USART\_DeInit(USART\_TypeDef\* USARTx)

功能描述 将外设 USARTx 寄存器重设为缺省值

输入参数 USARTx：x 可以是 1，2 或者 3，来选择 USART 外设

输出参数 无

返回值 无

先决条件 无

被调用函数

RCC\_APB2PeriphResetCmd()

RCC\_APB1PeriphResetCmd()

**#include "stm32f4xx.h"**

**void USART\_Configuration(void) {**

**// 配置USART1外设的相关寄存器**

**// ...**

**}**

**void Reconfigure\_USART(void) {**

**// 先将USART1寄存器重设为缺省值**

**USART\_DeInit(USART1);**

**// 然后重新配置和初始化USART1外设**

**USART\_Configuration();**

**}**

**int main() {**

**// 初始化系统和GPIO等**

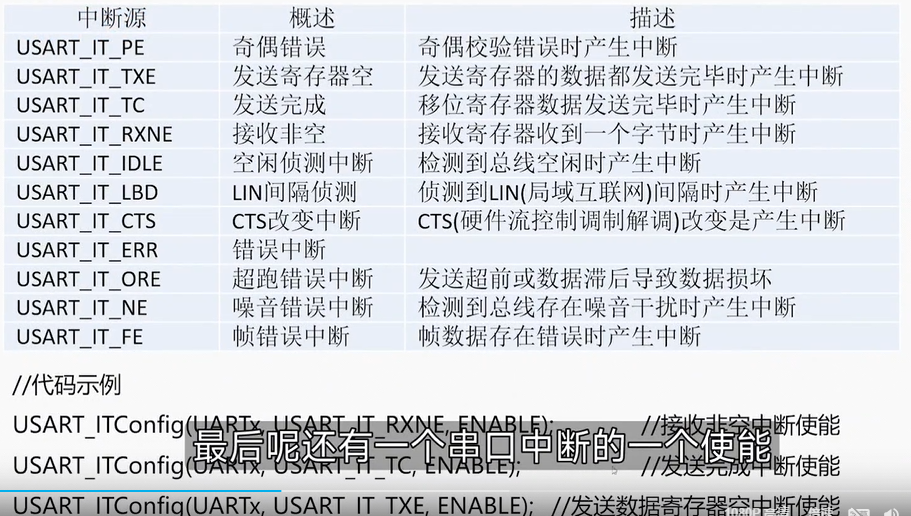
**Reconfigure\_USART(); // 重新配置和初始化USART1**

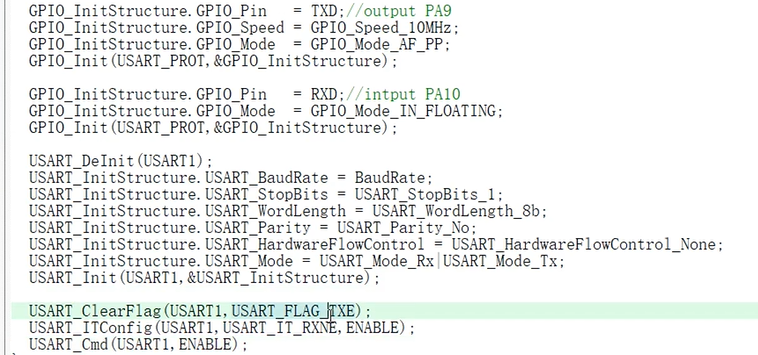
**while (1) {**

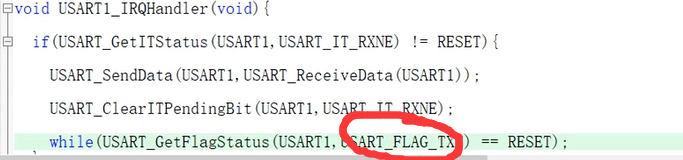
**// 执行其他操作**

**}**

**}**

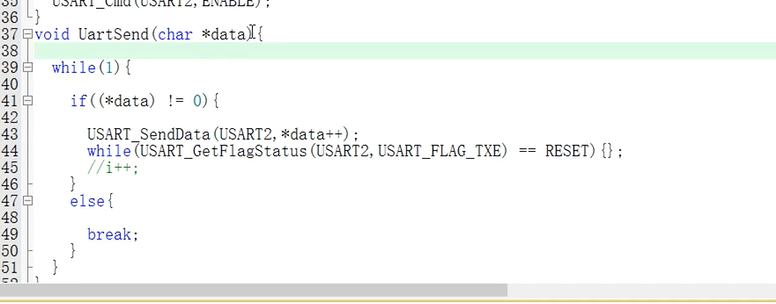


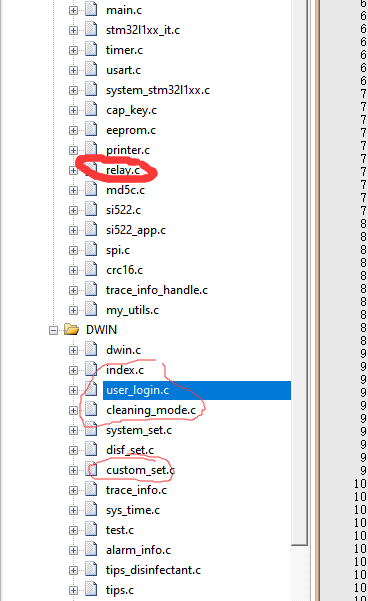


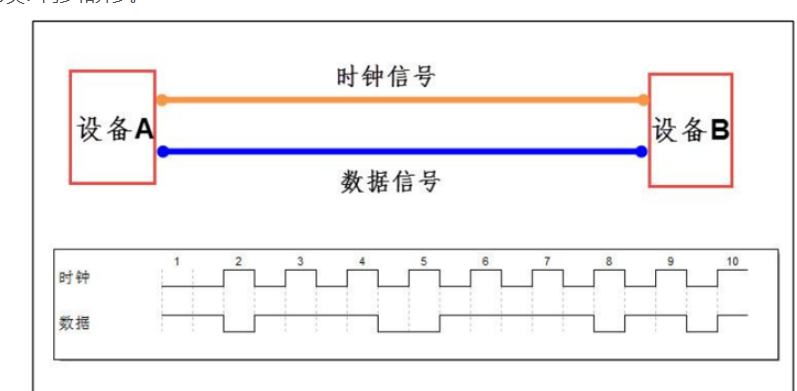


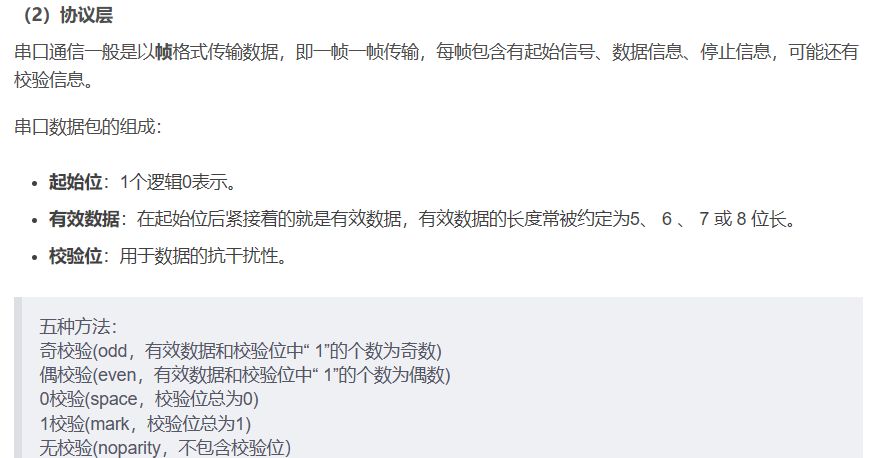
用TC不稳定

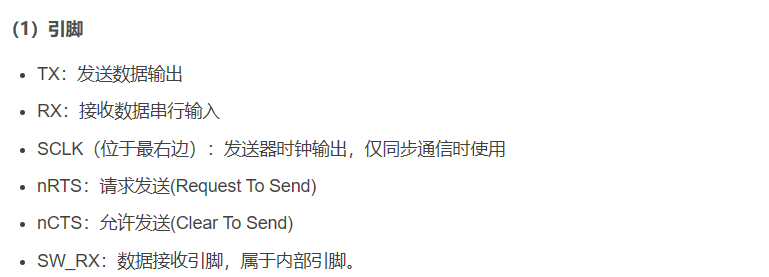
当usart\_getflagstatus()不能等于0退出











void USART\_Config(void)

{

GPIO\_InitTypeDef GPIO\_InitStructure;

USART\_InitTypeDef USART\_InitStructure;

// 1. 开启USART时钟和对应的引脚时钟

// USART1

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOA, ENABLE);

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_USART1, ENABLE);

// 2. 配置USART引脚

// USART1-TX发送 GPIOA.9 复用推挽模式

GPIO\_InitStructure.GPIO\_Pin = USART1\_TX\_GPIO\_PIN;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_AF\_PP;

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

GPIO\_Init(USART1\_TX\_GPIO\_PORT, &GPIO\_InitStructure);

// USART1-RX接收 GPIOA.10 浮空输入模式

GPIO\_InitStructure.GPIO\_Pin = USART1\_RX\_GPIO\_PIN;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_IN\_FLOATING;

GPIO\_Init(USART1\_RX\_GPIO\_PORT, &GPIO\_InitStructure);

// 3. 配置USART1串口参数

USART\_InitStructure.USART\_BaudRate = USART\_BAUDRATE; // 3.1 配置波特率

USART\_InitStructure.USART\_WordLength = USART\_WordLength\_8b; // 3.2 配置数据字长

USART\_InitStructure.USART\_StopBits = USART\_StopBits\_1; // 3.3 配置停止位

USART\_InitStructure.USART\_Parity = USART\_Parity\_No; // 3.4 配置校验位

USART\_InitStructure.USART\_Mode = USART\_Mode\_Tx | USART\_Mode\_Rx; // 3.5 配置工作模式

USART\_InitStructure.USART\_HardwareFlowControl = USART\_HardwareFlowControl\_None; // 3.6 配置硬件控制流

USART\_Init(USART1, &USART\_InitStructure); // 4. 初始化串口

USART\_ITConfig(USART1, USART\_IT\_RXNE, ENABLE);//USART中断使能

USART\_Cmd(USART1, ENABLE); // 5. 使能串口

}