# USART——串口通讯

## USART简介

通用同步异步收发器(Universal Synchronous Asynchronous Receiver and Transmitter)是一 个串行通信设备，可以灵活地与外部设备进行全双工数据交换。有别于USART还有一个UART(Universal Asynchronous Receiver and Transmitter)，它是在 USART 基础上裁剪掉了同步通信功能，只有异步通信。简单区分同步和异步就是看通信时需不需要对外提供时钟输出，**我们平时用的串口通信基本都是 UART**。串行通信一般是以帧格式传输数据，即是一帧一帧的传输，每帧包含有起始信号、数据信息、停止信息，可能还有校验信息。USART 就是对这些传输参数有具体规定，当然也不是只有唯一一个参数值，很多参数值都可以自定义设置，只是增强它的兼容性。

关于USART的详细介绍网上有很多资料，此处不再详细讲解。

串口通讯（上）——基础概念https://zhuanlan.zhihu.com/p/24858971

串口通讯（中）——USART https://zhuanlan.zhihu.com/p/26129454

串口通讯（下）——简单的USART数据收发实验https://zhuanlan.zhihu.com/p/26315125

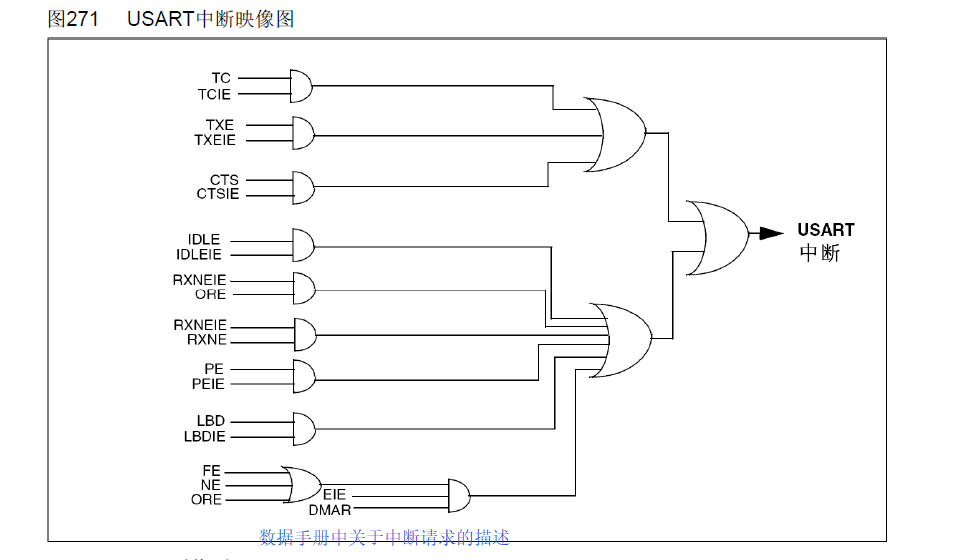
**USART 在 STM32 应用最多莫过于“打印”（可以理解为c中的printf）**程序信息一般在硬件设计时都会预留一个USART通信接口连接电脑，用于在调试程序是可以把一些调试信息“打印”在电脑端的 串口调试助手工具上，从而了解程序运行是否正确、如果出错哪具体哪里出错等等。

实际使用中我们要将单片机的发送端接到串口的接收端，单片机接收端同理。（TX：发送数据输出引脚。 RX：接收数据输入引脚。）为了使发送方与接收方有共同的参考电压，**需要将两者的gnd（地线）接到一起**（否则接收到的数据会乱码）。因此在使用中**最少应同时接三根线**。

## USART中断简介

USART有多个中断请求源，如下表所示：

|  |  |  |
| --- | --- | --- |
| **中断事件** | **事件标志** | **使能控制位** |
| 发送寄存器空 | TXE | TXEIE |
| CTS标志位 | CTS | CTSIE |
| 发送完成 | TC | TCIE |
| 准备好读取接收到的数据 | RXNE | RXENE |
| 检测到上溢错误 | ORE |
| 检测到空闲线路 | IDLE | IDLEIE |
| 奇偶检验错误 | PE | PEIE |
| 断路标志 | LBD | LBDIE |
| 多缓冲通信中的噪声标志、上溢错误和帧错误 | NF/ORE/FE | EIE |



其中事件标志位在事件发生后由硬件自动置位，使能位由软件控制向USART\_CRx(x=1,2,3)寄存器写入相应的值来控制是否有效。（具体请查阅数据手册）

由上图可知不同的中断请求源都会指向同一个中断UASRTx（x=1,2,3）中断，所以如果使能了多个USART中断，在进入中断后，通常使用USART\_GetITStatus()函数来判断具体发生了那一个中断。

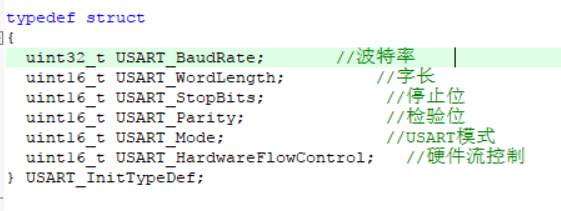
## USART初始化结构体介绍

标准库函数对每个外设都建立了一个初始化结构体，比如 USART\_InitTypeDef，结构体成员用于设置外设工作参数，并由外设初始化配置函数，比如USART\_Init()调用，这些设定参数将会设置外设相应的寄存器，达到配置外设工作环境的目的。初始化结构体和初始化库函数配合使用是标准库精髓所在，理解了初始化结构体每个 成员意义基本上就可以对该外设运用自如了。初始化结构体定义在 stm32f10x\_usart.h 文件 中，初始化库函数定义在 stm32f10x\_usart.c 文件中，编程时我们可以结合这两个文件内注释使用。

在使用USART前需要先将GPIO初始化为对应的复用模式，开启GPIO时钟，在数据手册中描述如下：



在配置完GPIO模式后还需配置USART的工作模式，同时按照USART所挂载总线开启外设时钟（时钟共开启了两次）（USART1为APB2，其余为APB1，详情参见数据手册第一章系统结构框图）



1) USART\_BaudRate：波特率设置。一般设置为 2400、9600、19200、115200。标准库函数会根据设定值计算得到 USARTDIV值，从而设置 USART\_BRR寄存器值。

2) USART\_WordLength：数据帧字长，可选 8位或 9位。它设定 USART\_CR1寄存器的 M 位的值。如果没有使能奇偶校验控制，一般使用 8 数据位；如果使能了奇偶校验则一般设置为 9 数据位。

3) USART\_StopBits：停止位设置，可选 0.5 个、1 个、1.5 个和 2 个停止位，它设定USART\_CR2 寄存器的 STOP[1:0]位的值，一般我们选择 1 个停止位。

4) USART\_Parity ： 奇 偶 校 验 控 制 选 择 ， 可 选 USART\_Parity\_No( 无校验 ) 、USART\_Parity\_Even( 偶校验 ) 以 及 USART\_Parity\_Odd( 奇 校 验 ) ， 它 设 定USART\_CR1 寄存器的 PCE 位和 PS 位的值。

5) USART\_Mode：USART模式选择，有 USART\_Mode\_Rx和 USART\_Mode\_Tx，允许使用逻辑或‘|’运算选择两个，它设定 USART\_CR1 寄存器的 RE 位和 TE 位。

6) USART\_HardwareFlowControl：硬件流控制选择，只有在硬件流控制模式才有效，可选有⑴使能 RTS、⑵使能 CTS、⑶同时使能 RTS 和 CTS、⑷不使能硬件流。

软件设计

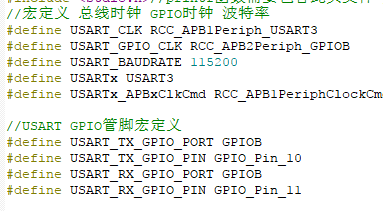
这里只讲解部分核心代码，完整的内容请参考例程。

### 编程要点

1. 使能RX和TX引脚GPIO时钟和USART时钟
2. 初始化GPIO，初始化USART
3. 使能USART
4. 实现接收与发送

### 代码分析（USART（中断））

**GPIO与USART宏定义**



**嵌套中断控制器NVIC配置**



**USART初始化**



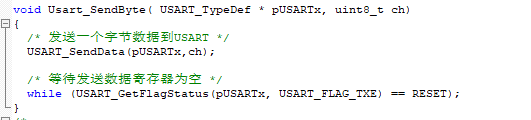
使用GPIO\_InitTypeDef和USART\_InitTypeDef结构体定义了一个GPIO初始化结构体和一个USART初始化结构体。

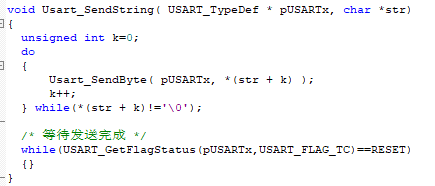
调用RCC\_APB2PerphClockCmd函数开启GPIO端口时钟（使用GPIO前必须开启对应时钟）。使用RCC\_APB1PerphClockCmd开启USART3时钟。

接下来就是GPIO管脚模式的设置和USART的配置：TX管脚设置为复用推挽输出，RX管脚设置为浮空输入；波特率设置为115200，字长为8，1个停止位，没有检验位，不使用硬件流控制，收发一体工作模式，然后调用USART初始化函数完成配置。

程序用到了USART接收中断，所以需要配置NVIC，这里调用NVIC\_Configuration函数完成配置。配置完之后调用USART\_ITConfig函数使能USART接收中断。

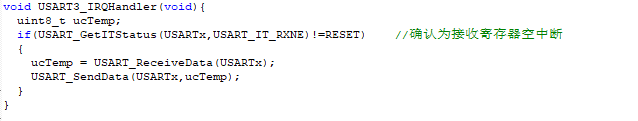
最后调用USART\_Cmd函数使能USART，这个函数最终配置的是USART\_CR1的UE位，具体作用是开启USART的工作时钟。





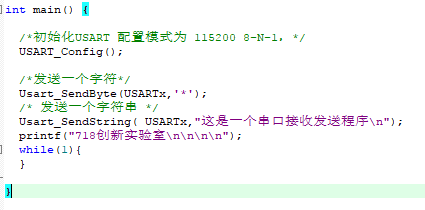
Usart\_SendByte 函数用来在指定 USART 发送一个 ASCLL 码值字符，它有两个形参， 第一个为 USART，第二个为待发送的字符。它是通过调用库函数 USART\_SendData 来实现 的，并且增加了等待发送完成功能。通过使用 USART\_GetFlagStatus 函数来获取 USART 事 件标志来实现发送完成功能等待，它接收两个参数，一个是 USART，一个是事件标志。这 里我们循环检测发送数据寄存器为空这个标志，当跳出 while循环时说明发送数据寄存器为 空这个事实。 Usart\_SendString 函数用来发送一个字符串，它实际是调用 Usart\_SendByte 函数发送每 个字符，直到遇到空字符才停止发送。最后使用循环检测发送完成的事件标志TC来实现保 证数据发送完成后才退出函数。

**中断服务函数**



使能了 USART 接收中断以后，当 USART 有接收到数据就会执行 USART\_IRQHandler 函数。USART\_GetITStatus 函数与 USART\_GetFlagStatus 函数类似用来 获取标志位状态，但 USART\_GetITStatus 函数是专门用来获取中断事件标志的，并返回该标志位状态。使用 if语句来判断是否是真的产生 USART数据接收这个中断事件，如果是真的就使用 USART 数据读取函数 USART\_ReceiveData 读取数据到指定存储区。然后再调用 USART 数据发送函数 USART\_SendData 把数据又发送给源设备，即 PC 端的串口调试助手。

**主函数**



首先我们需要调用 USART\_Config 函数完成 USART 初始化配置，包括 GPIO 配置， USART 配置，接收中断使能等等信息。 接下来就可以调用字符发送函数把数据发送给串口调试助手了。 最后主函数什么都不做，只是静静地等待USART接收中断的产生，并在中断服务函数 把数据回传。

## 下载验证

保证开发板相关硬件连接正确，用 串口线或USB转TTL转接板连接stm32与电脑（gnd相连，rxd接txd，txd接rxd），在电脑端打开串口调试助手并配置好相关参数：115200 8-N-1，把编译好的程序下载到开发板，此时串口调试助手即可收到开发板发过来的数据。我们在串口调试助手发送区域输入任意字符，点击发送按钮，马上在串口调试助手接收区即可看到相同的字符。