

TTL协议是芯片直接的协议，逻辑1时高电平5V，逻辑0时低电平0V
RS232是用于设备之间通讯的串行协议，逻辑1时低电平-15V，逻辑0时高电平15V

关于STM32编程

- 首先新建一个bsp文件夹（板级支持包），在里面新建bsp.c和bsp.h两个文件
 - 逻辑是在.h头文件里面进行引脚和寄存器的宏定义，在bsp.c文件里面编写函数，再将.c文件里面的函数声明在.h头文件里面，在main.c主文件中包含bsp.h，就可以直接调用其中的函数
- 然后打开keil工程，把新建的文件添加到工程里面（添加bsp.c文件，在c文件里面包含bsp.h头文件，引用的时候使用 引号）
- 然后在 魔术棒-C/C++ 选项里添加头文件所在的文件夹，程序才能找到这个头文件
- 然后在bsp.h文件里面写 条件编译 语句：

```
#ifndef __BSP_USART_H
#define __BSP_USART_H

#include "stm32f10x.h"
#include <stdio.h> // 这两个肯定要有

// 在这里面写头文件内容

#endif /* __BSP_USART_H */
```

- 关于串口编程，需要以下流程：
 - 1.初始化串口需要的GPIO
 - 引脚一般选择复用推挽模式
 - 使用结构体初始化
 - 在串口编程中，需要将TX配置成复用推挽模式，将RX配置成浮空输入模式
 - 2.初始化串口，USART_InitTypeDef
 - 3.中断配置（接收中断，中断优先级）
 - 4.使能串口
 - 5.编写发送和接收函数
 - 6.编写中断服务函数
- 首先在头文件里面搞一些宏定义，把一些GPIO引脚或者寄存器定义成新的名称，方便移植
- 在bsp.c文件中进行初始化配置函数编写的时候顺序为：
 - 1.定义结构体变量，如 GPIO_InitTypeDef GPIO_InitStructure
 - 2.开时钟
 - 3.使用结构体配置GPIO的模式
 - 4.如果是串口编程的话接下来使用结构体配置串口的工作模式
- STM32的串口一次只能发8位数据，如果位数更多需要分批发送
 - 发送一个字节检查TXE标志位，发送一连串数据检测TC标志位
- 发送字符串可以使用 do while 方式，判断字符串的最后一位为'\0'
- 可以通过修改底层的 fputc 和 fgetc 函数（把里面的端口改成串口）来直接使用printf和scanf函数实现串口的发送和接收功能
 - 在使用getchar等用法的时候需要将串口配置中关于中断的配置注释掉