

# 串口通信

---

## 知识学习

---

什么是通信协议

为什么要有通信协议

串口通信的物理层

串口通信的协议层

解释重要概念：数据帧（一个通信数据），校验位（异步串行通信中判断每一位数据位传送完成的数  
据），波特率（每秒钟传送二进制数码的位数，以bit/s（bps）为单位）

串口有哪几种中断？哪些事件可以触发串口中断？

一. 计算机通信的基本概念

分为串行通信与并行通信。

串行通信：数据逐位传输 并行通信：多维数据同时传输

并行通信示意图：

串行通信示意图：

异步串行通信的数据格式：

异步串行通信通信速率：

波特率：每秒钟传送二进制数码的位数，以bit/s（bps）为单位。

常用bps：9600,19200,38400,57600和115200。

每一位的持续时间为1/bps

两个关键点：

字符格式：字符中数据的传输形式。

传输时低位在前，高位在后。

波特率：决定字符中每一位数据的持续时间。

数据接收过程：

本质为数据接收。

时钟周期？

比特周期的采样点？

数据位周期的采样点？

串口通信的数据传输方向：

串口通信是全双工。

三种常用的错误校验方式：

串口收发单元的功能框图：

数据寄存器DR：

双缓存：可以同时写入新的数据或者读取已接收的数据，提高数据的传输效率。

通信状态标志位：

TXE：TDE寄存器内容发送到发送移位寄存器时置1

TC：数据发送完成时置1

RXNE：接收数据寄存器不为空的标志。移位寄存器的内容传送到接收数据寄存器RDR时为1。

轮询方式下可以直接检测标志位

中断方式下需要在中断服务程序中通过检测不同的中断标志位来判断终端类型，以执行后续任务。

STM32F411芯片中串口通信（UART）的引脚：

USB转TTL串口连接

从数据手册上查到连接方式：

模块的RX连接板子的TX（PA2）。

模块的TX连接板子的RX（PA3）

HAL库外设初始化设计思想：

串口数据类型：

串口的数据类型定义：

串口初始化数据类型：

WordLength的取值范围：

StopBits取值范围：

Parity取值范围

Mode取值范围

HwFlowCtr取值范围

硬件流控：控制数据传输的进程，防止数据丢失，主要是在双方传输速度不匹配的时候使用。

OverSampling取值范围

外设初始化设置：

串口初始化过程：

串口初始化函数

与MCU相关的初始化函数

轮询方式的串口通信：

轮询方式的接口函数：

轮询方式的初始化函数：

# 实验

将核心板通过USB转串口模块连接上电脑（若板载USB-TTL CH340模块，直接用数据线连接即可；若无板载，需自行购买并连接）通过以下形式实现：电脑给单片机发送定长数据，单片机接收后以原样传回阻塞式收发串口接收中断

## 1. CubeMX配置

Categories

A->Z

Connectivity

I2C1

I2C2

I2C3

SDIO

SPI1

SPI2

SPI3

SPI4

SPI5

USART1

**USART2**

USART16

USB\_OTG\_FS

USART2 Mode and Configuration

**串口外设配置**

Mode

Mode Asynchronous

Hardware Flow Control (RS232) Disable

Configuration

Reset Configuration

DMA Settings

User Constants

GPIO Settings

NVIC Settings

Parameter Settings

Basic Parameters

Baud Rate 115200 Bits/s

Word Length 8 Bits (including P

Parity None

Stop Bits 1

Advanced Parameters

Data Direction Receive and Trans

Over Sampling 16 Samples

异步模式，无硬件流控

设置通信参数：  
波特率**115200**  
8位数据位  
无奇偶校验  
1位停止位  
使能接收和发送  
16倍过采样

## 2. 程序编写

1. /\* USER CODE BEGIN PM \*/

2. **uint8\_t RecBuf[10];**

3. /\* USER CODE END PM \*/

4. **UART\_HandleTypeDef huart2;**

用户定义

CubeMX生成

**用户变量定义**

// 接收缓冲区定义

// 串口2句柄定义

1. while (1)

2. {

3. /\* USER CODE BEGIN 3 \*/

4. // 接收5个字符完成

5. if( HAL\_UART\_Receive(&huart2,RecBuf,5,100) == HAL\_OK )

6. {

7. HAL\_UART\_Transmit(&huart2,RecBuf,5,100); // 把接收的字符原样发回

8. }



9. }


10. /\* USER CODE END 3 \*/

步骤六：程序编写

**用户应用代码**

## 串口调试助手

串口号:  COM15 

波特率:  115200

数据位: 8

校验位: None

停止位: One

关闭串口


接收区设置

☐ 接收并保存到文件

☐ 十六进制显示


☐ 暂停接收显示

☐ 自动断帧 ? 20


☐ 接收脚本  Add Timesta

保存数据 清空数据

发送区设置

☐ 发送文件  扩展命令


☐ 十六进制发送

☐ 发送脚本  ADD8

☐ 定时发送 1.0 秒

☐ DTR ☐ RTS

换行符 \r\n (CRLF)

☒ 显示发送字符串 


1 设置串口通信参数

2 设置接收区和发送区

UESTC

输入5个字符

发送: 10 接收: 10

 发送按钮

» UESTC

« UESTC

» UEST

« UESTC

右箭头表示发送字符

左箭头表示接收字符