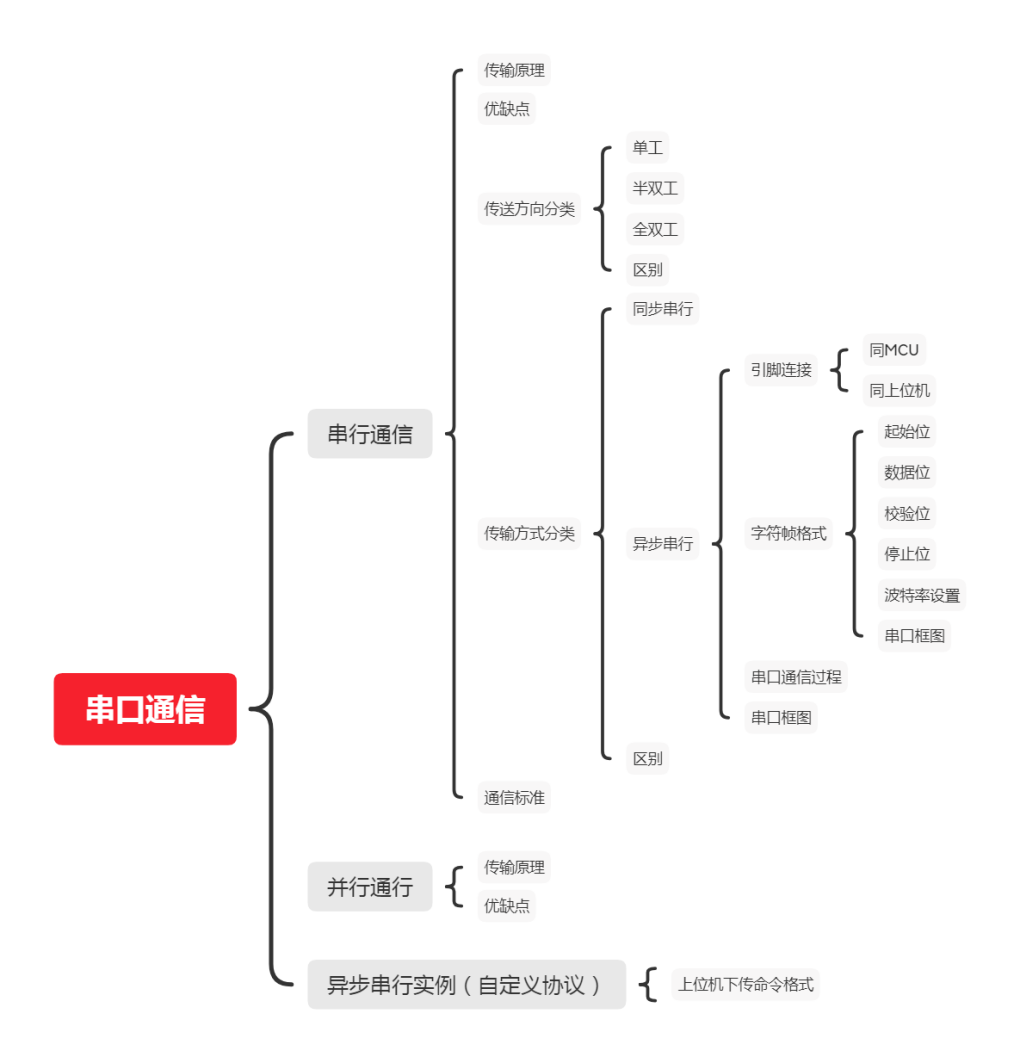
**串口通信**

在设计实现某种功能的系统时，常需要不同设备之间的通信。通信主要有串行和并行两种方式，应用比较广泛的UART(Universial Asynchonous Receiver Transmitter)通用异步收发器,SPI(Serial Peripheral Interface)串行外设接口，I2C(Inter Integrated Circuit)集成电路总线都属于这两种方式之一。下面就以应用比较广泛的串行通信作以介绍，文末附有自定义通信协议的代码。下图是本文整体框架。



目录

[1 串行通信 - 3 -](#_Toc15603)

[1.1 原理与优缺点 - 3 -](#_Toc895)

[1.2 分类 - 3 -](#_Toc21029)

[1.2.1 按通信方向 - 3 -](#_Toc14083)

[1.2.2 按通信方式 - 3 -](#_Toc12006)

[1.3 异步串行引脚连接 - 4 -](#_Toc18682)

[1.3.1 串口外设之间 - 4 -](#_Toc22007)

[1.3.2 ARM与PC之间 - 4 -](#_Toc15368)

[1.4 字符帧格式 - 5 -](#_Toc18090)

[1.5 串口通信过程 - 5 -](#_Toc1135)

[1.6 串口框图 - 6 -](#_Toc13714)

[2 并行通信 - 6 -](#_Toc6275)

[3 异步串行实例（自定义通信协议） - 6 -](#_Toc11153)

[3.1 项目需求 - 7 -](#_Toc8698)

[3.2 项目框架 - 7 -](#_Toc18036)

[3.3 上位机下传处理器帧格式 - 7 -](#_Toc16016)

[3.4 STM32F4代码设计 - 8 -](#_Toc28560)

1. 串行通信
   1. 原理与优缺点

数据按位依次传输，使用少数几条[通信线路](https://baike.baidu.com/item/%E9%80%9A%E4%BF%A1%E7%BA%BF%E8%B7%AF/1527630" \t "https://baike.baidu.com/item/%E4%B8%B2%E8%A1%8C%E9%80%9A%E4%BF%A1/_blank)就可以完成系统间交换信息，适合于主机与主机，主机与外设之间的远距离通信。优点是占用引脚资源少，缺点是速度相比并行要慢。

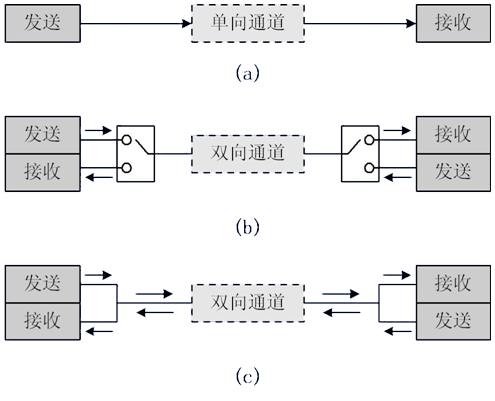
* 1. 分类
     1. 按通信方向

按通信方向分为单工、半双工、全双工三种。

a、单工：数据传输只支持数据在一个方向上传输。

b、半双工：允许数据在两个方向上传输。但是，在某一时刻，只允许数据在一个方向上传输，它实际上是一种切换方向的单工通信；它不需要独立的接收端和发送端，两者可以合并一起使用一个端口。

c、全双工：允许数据同时在两个方向上传输。因此，全双工通信是两个单工通信方式的结合，需要独立的接收端和发送端。



* + 1. 按通信方式

按通信方式分为同步串行和异步串行。

同步串行：同步通信时，通信双方共用一个时钟，这是同步通信区分于异步通信的最显著的特点。同步通信中，数据开始传送前用同步字符来指示（常约定1～2 个），并由时钟来实现发送端和接收端的同步，即检测到规定的同步字符后，下面就连续按顺序传送数据，直到一块数据传送完毕。同步传送时，字符之间没有间隙，也不要起始位和停止位，仅在数据开始时用同步字符SYNC来指示。例如SPI，I2C通信接口。

异步串行：无时钟信号的驱动，收发双方约定数据帧的格式。每一帧由起始位、数据位、奇偶校验位和停止位组成。传送过程字符之间可以有空隙的存在。例如UART,单总线。

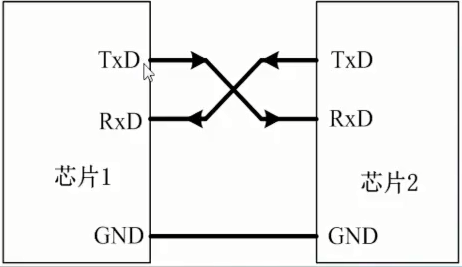


* 1. 异步串行引脚连接

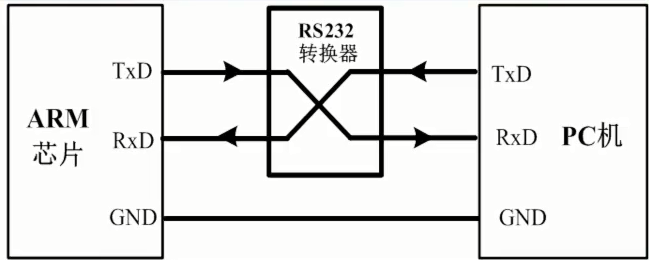
RXD：数据输入引脚，数据接收。

TXD：数据发送引脚，数据发送。

* + 1. 串口外设之间



* + 1. ARM与PC之间



当ARM与电脑USB口连接传输数据时，需要一个USB转串口的装置，因为电脑无法识别TTL电平，所以需要连接一个RS232转换器将TTL电平转换成232电平。常采用的是MAX232或国产的CH340芯片。

ARM的电平标准（TTL电平）：对于TTL电平，其输入门闩值分别是0.8V和2.0V, 即输入高电平范围是 2.0--5V, 低电平范围是0--0.8V，对于LVTTL，高低电平范围分别是 2.4--5V，0-- 0.4V，即LOW VOLTAGE TTL，是低电平标准的TTL。

RS232的电平标准：+15 ~ +13 V表示0，-15 ~ 13V表示1。

更多RS232电平连接方式及知识可参考：

<https://www.21ic.com/jichuzhishi/datasheet/RS232/jiekou/187973.html>

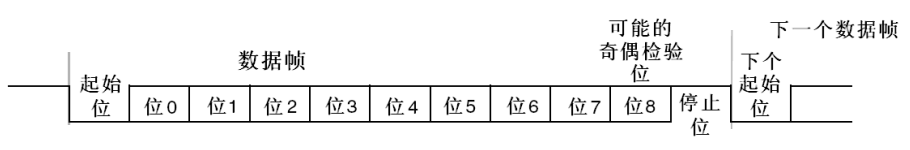
* 1. 字符帧格式

起始位

数据位

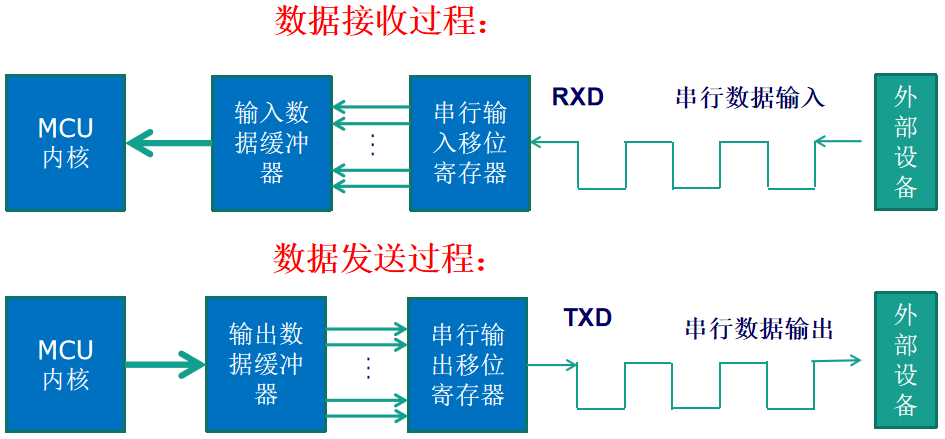
奇偶校验位：奇检验就是保证8位数据位中1的个数为基数个，如果1是基数个则检验位就为0，反之为1；偶检验就是保证8位数据位中1的个数为偶数个，如果1是偶数个则检验位就为0，反之为1；

停止位



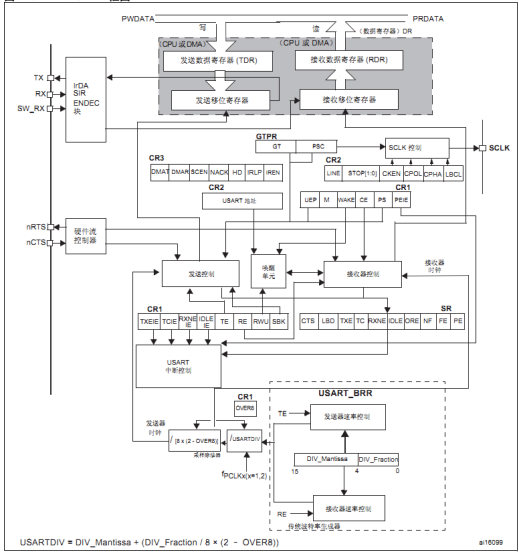
波特率：波特率是指数据传送时，每秒传送数据二进制代码的位数，它的单位是位/秒（bit/s）。在异步串行通信中，接收设备和发送设备保持相同的传送波特率，并以每个字符数据的起始位与发送设备保持同步。起始位。数据位。奇偶位和停止位的约定，在同一次传送过程中必须保持一致，这样才能成功的传送数据。

* 1. 串口通信过程



* 1. 串口框图

详细图片可查看STM32F4数据手册26.3节。



1. 并行通信
   1. 传输原理与优缺点

数据各位同时传送，快速设备之间采用并行通信，例如CPU 与存储设备、存储器与存储器、主机与打印机等都采用并行通信。并行通信，有多少位数据就必须有多少根数据线。优点是传输速度快，效率高，多用在对实时性要求高的场合。缺点是长距离通信时抗干扰能力差，占用引脚资源多。并行通信应用较少，在此不多做介绍。

1. 异步串行实例（自定义通信协议）

说明：本实例适合于具体项目中需要自定义帧头、帧尾等数据传输格式的应用场合。比较基础的串口设置步骤例如使能时钟，初始化参数等在此不多做介绍，具体可参考原子哥的STM32F4 开发指南(库函数版) ->5.3.3节和第九章《串口通信实验》。

更多串口接收数据的方式还可以参考：

<http://www.openedv.com/forum.php?mod=viewthread&tid=67915&highlight=%B4%AE%BF%DA%BD%D3%CA%D5%CA%FD%BE%DD>

由于涉及具体项目，故不会透露太多细节，只对通信协议进行介绍，方便大家移植。进入正题。

* 1. 项目需求

系统分为三个模块，处理器（STM32F4）、FPGA、模拟电路。具体的系统功能由模拟电路实现。处理器负责接收来自上位机的指令，并控制FPGA输出命令信息和通信信息。

* 1. 项目框架

上位机与ARM通过板载的蓝牙通信，采用了HC-05蓝牙转串口模块，模块的引脚RXD与TXD分别与ARM的串口收发端相连接，这样即可实现上位机与ARM的通信。

ARM收到正确的上位机指令后，将指令或数据按照通信协议发送给FPGA，由FPGA控制模拟电路完成系统功能。

* 1. 上位机下传处理器帧格式

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 说明：上位机给处理器的数据格式  通信格式（16进制）：C1 D2（2字节） + 命令号（1字节）+ 数据长度（2字节）+ 数据（n字节） + 校验和（2位）+ E3 F4（2字节） | | | | | |
| 开始标识 | 命令号 | 数据长度 | 数据 | 校验和 | 结束标识 |
| C1 D2 | 0x01 | 0(0x0000) | 0 | 0x0194 | E3 F4 |
| 0x02 | 30(0x001E) | 30个0xFB | ...... |
| 0x03 | 1(0x0001) | 0x9B | 0x0232 |
| 0x04 | 2(0x0002) | 0x8C,0x9E | 0x02C3 |
| 0x05 | 153(0x0099) | 153个0x3F | ...... |
| 0x06 | 251(0x00FB) | 251个0x1A | ...... |

说明：如果是自定义帧头的话，尽量选择55，AA之类不容易在数据部分出现的码，或者是用两个字节充当帧头。

校验和为开始标识+命令号+数据长度+数据（16进制求和）。表中省略号的部分

……当然是懒得算啦，一个一个输入也太傻了。分享一个计算校验和的网站：<https://www.23bei.com/tool-205.html>

当然帧头/尾、数据长度、数据、校验和这些你可以任意安排，只要上位机按照这种数据格式发送数据，作为处理器端按这种数据格式接收数据就行。这就是双方的通信协议。

数据长度与校验和都是高位在前，低位在后发送。例如命令号为01，没有数据的上位机指令为：C1D20100000194E3F4

* 1. STM32F4代码设计

当上位机指令有错误时，ARM会返回对应的错误号码，上位机发送指令正确，则会返回上位机发送的数据（只是帧格式中的数据，不包含命令号等其他的）。发送上位机指令用的是串口助手，串口助手设置见下图，串口助手的参数一定要跟ARM程序中设置的一样，就像两个人能正常交流那就得用约定好的语言规则。

不同指令运行结果见下图：

