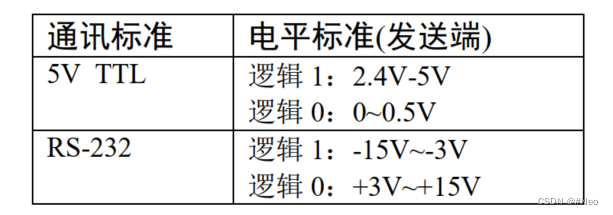
## USART串口通讯

**传输模式**

全双工

**物理层**

一般采用Vcc，RXD，TXD，GND四条线



**协议层**

一帧数据通常是一个字符，包含起始位，数据位，校验位，停止位

起始位：由一个逻辑0表示

停止位：由0.5/1/1.5/2个逻辑1表示 //需要约定好

数据位：一般约定为5-8位长

校验 ：可选位

奇校验，偶校验，0校验，1校验或者无校验

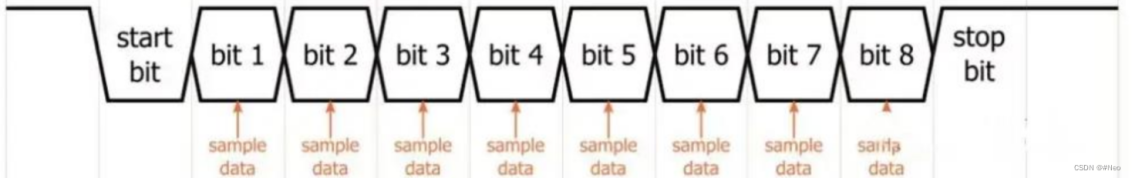
**波特率**

代替时钟信号，码元长度

**异步串行通信数据接收过程**

本质是一种数据采集，假设接收端采样时钟为波特率的16倍

1. 起始位下降沿启动接收
2. 接收端等待8个时钟周期，以建立一个接近比特周期中间的采样点
3. 接收端等待16个时钟周期，进入第一个数据位周期的中点
4. 采样第一个数据位并记录在接收寄存器中
5. 等待16个时钟周期，采集第二个数据位
6. 由停止位上升沿使数据线空闲



**USART与UART**

USART：通用同步异步收发器

UART ：在USART基础上裁剪掉了同步通信功能

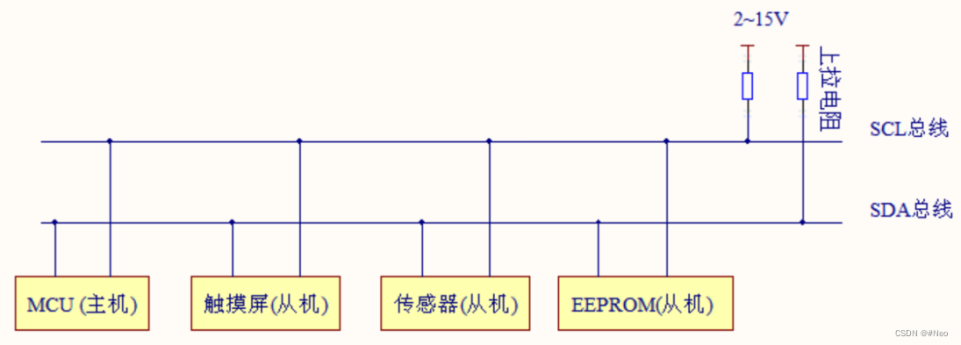
区分同步异步：是否需要对外提供时钟输出

## IIC

**传输模式**

同步半双工

**物理层**



1. 支持多设备总线，多个主机从机
2. 一条双向数据线SDA，一条时钟线SCL
3. 设备具有独立地址
4. 总线通过上拉电阻接电源：设备空闲输出高阻态，所有设备空闲时由 上拉电阻把总线拉成高电平
5. 利用仲裁方式确定哪个主机使用总线，以防数据冲突
6. 三种传输模式 标准100kbit/s、400kbit/s、1Mbit/s（大部分不支 持）
7. 相同总线上设备数量受总线最大电容400pF限制

**协议层**

数据由起始信号，停止信号和应答信号组成

传输单位为字节，传输字节数不做限制

起始信号：

//软件模拟 IIC 起始信号

void IIC\_Start(void)

{

SDA\_OUT(); //sda 线输出

IIC\_SDA=1;

IIC\_SCL=1;

delay\_us(4);

IIC\_SDA=0;//START:when CLK is high,DATA change form high to low

delay\_us(4);

IIC\_SCL=0;//钳住 I2C 总线，准备发送或接收数据

}

停止信号：

//产生 IIC 停止信号

void IIC\_Stop(void)

{

SDA\_OUT();//sda 线输出

IIC\_SCL=0;

IIC\_SDA=0;//STOP:when CLK is high DATA change form low to high

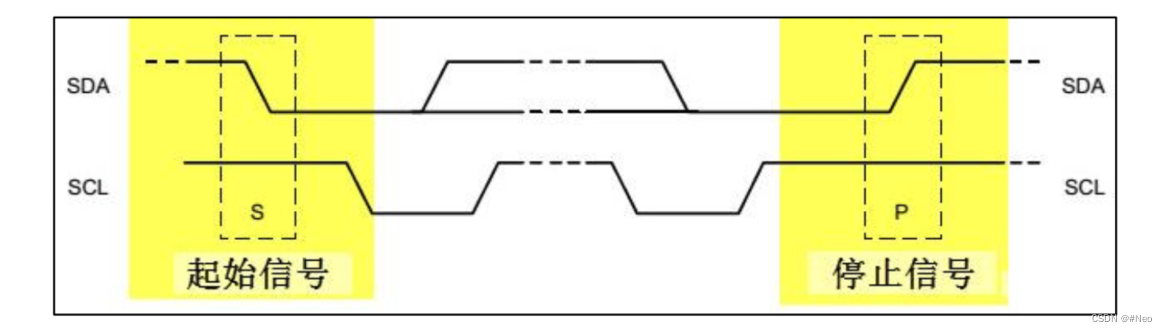
delay\_us(4);

IIC\_SCL=1;

IIC\_SDA=1;//发送 I2C 总线结束信号

delay\_us(4);

}



有效数据

SDA在SCL的每个时钟周期进行一位传输，且SCL为高电平时SDA 为有效数据;在SCL为低电平时，SDA进行电平切换

地址及数据方向

地址位数一般为7位或者10位，前者更为广泛

主机发起通讯时，SDA会发送设备地址且设备地址后会跟一数据位 表示读/写

0：主机向从机写

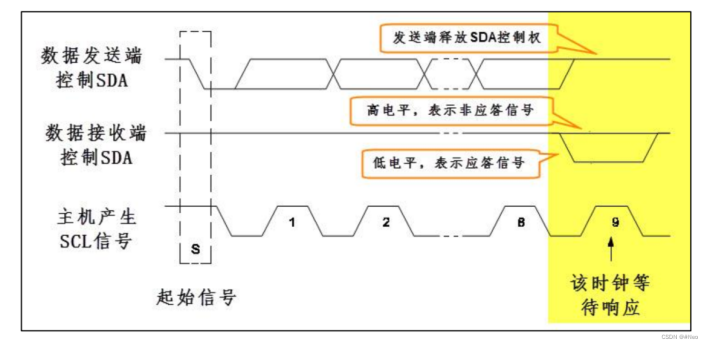
1：主机从从机读，主机释放对SDA的控制

响应信号

包括ACK和NACK两种信号

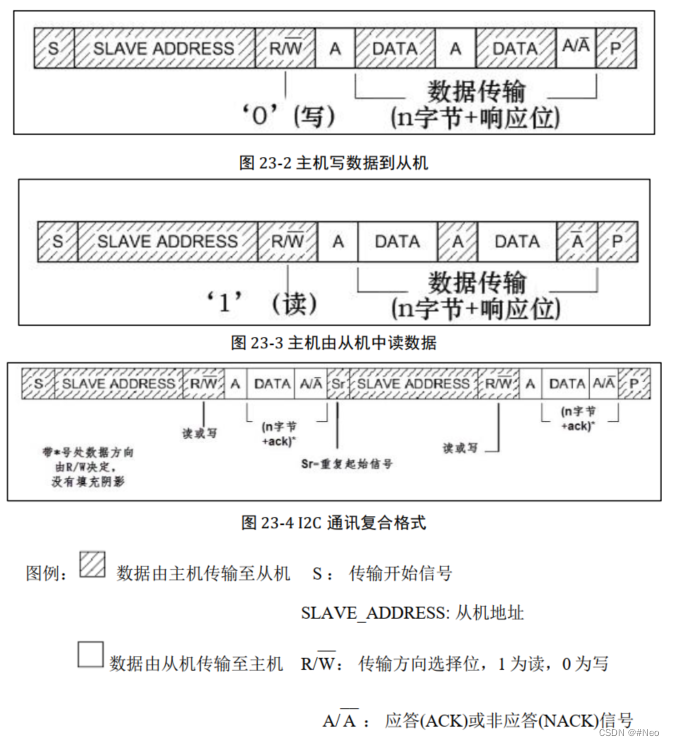
需要对方继续发送数据则发送ACK信号，否则为NACK

接收到NACK信号后会发送一个停止信号



第九个时钟周期，发送端释放SDA，等待接收端拉高释放NACK信号/拉低释放ACK信号

**IIC读写**



## SPI

**传输模式**

同步全双工

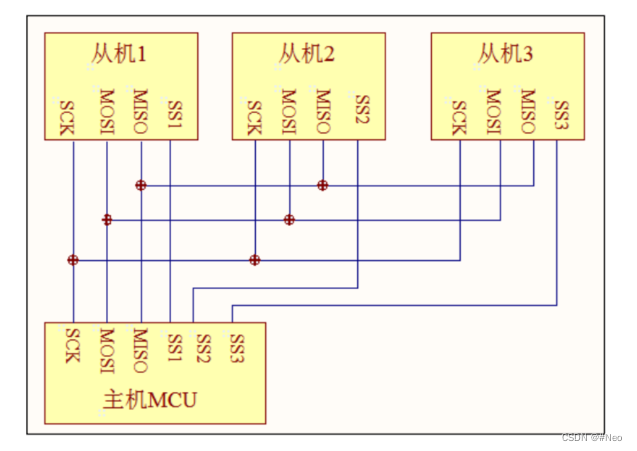
**物理层**

SS:片选线，也叫NSS、CS。当选中时，将片选线电平拉低，即为开始， 拉高为结束

SCK：时钟线，用于数据同步。决定通讯速率。STM32的最大时钟频率为 fpclk/2,最终通信速率取决于低速设备

MOSI：主机输出从机接收线

MISO：主机接收从机输出线



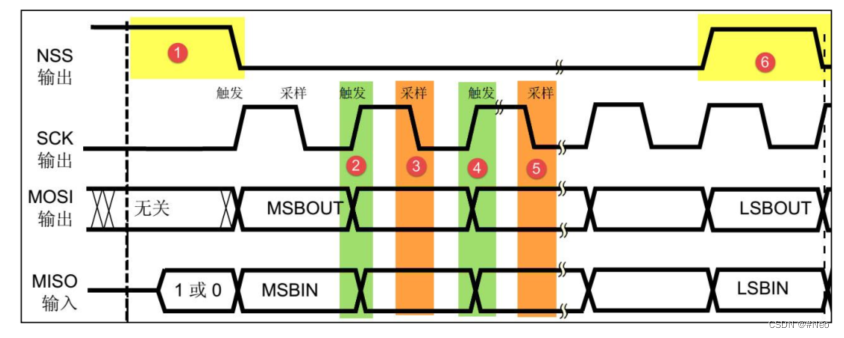
**协议层**

每次以8位或者16位为数据单位传输，单位数不受限

起始信号：片选线拉低

停止信号：片选线拉高

有效数据：数据位在SCK的下降沿被采集，在SCK的上升沿变化，其他 时刻数据无效



**通讯模式**

一共有四种模式，区别为空闲时刻SCK的状态以及采样时刻不同。

时钟极性CPOL

设备处于空闲状态SCK的电平信号，若CPOL=0，则SCK空闲时为0， 反之为1

时钟相位CPHA

当CPHA为0，数据在SCK的奇数边沿被采集，反之为偶数边沿采集



## RS-485

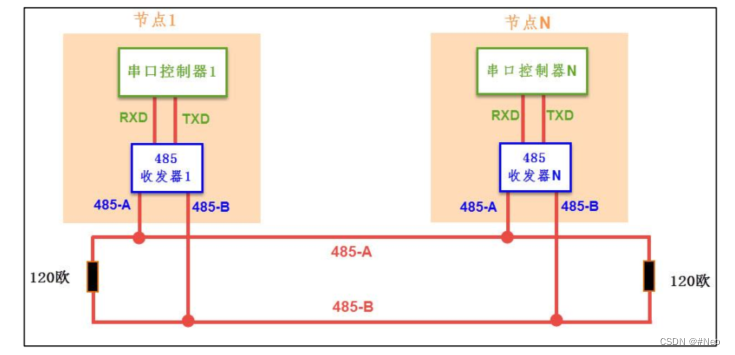
**传输模式**

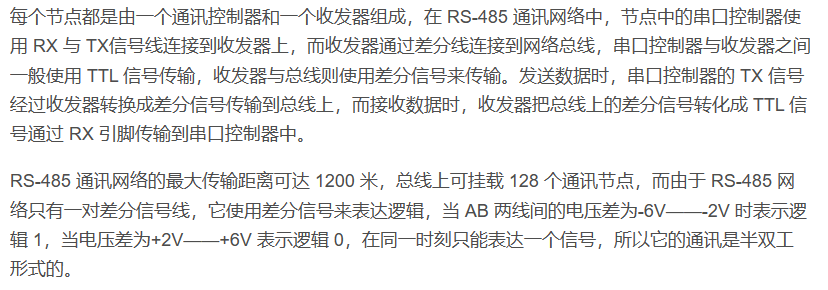
异步半双工，由RS232改进而来

**物理层**

将原有信号改变为差分信号。

每个节点由控制器和收发器组成。





## CAN

**传输模式**

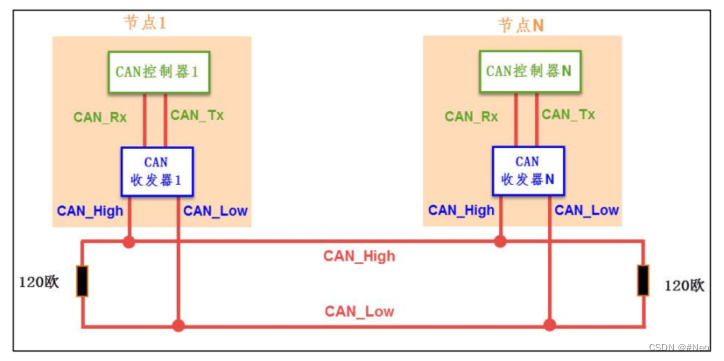
异步通讯

**物理层**

只有CAN\_High和CAN\_Low两条线（H和L），构成一组差分信号线

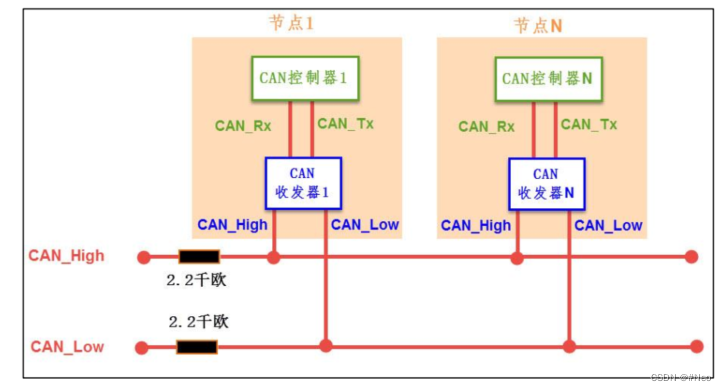
闭环总线网络：

总线最大长度40M，最高速度1Mbps，要求总线两端有 一个120欧 姆的电阻，结构形似RS485



开环总线网络：

最大传输距离1km，最高通信速率125kbps，两根总线独立且要求串 联2.2k欧姆电阻



通讯节点：

通讯节点无地址，通过对数据内容编码实现区分，理论上节点数只 与负载有关。

节点由一个控制器和一个收发器组成，控制器与收发器之间通过Tx 与Rx线连接，收发器与总线通过High和Low连接。前者是普通的 TTL信号，后者是差分信号

**协议层**

波特率：

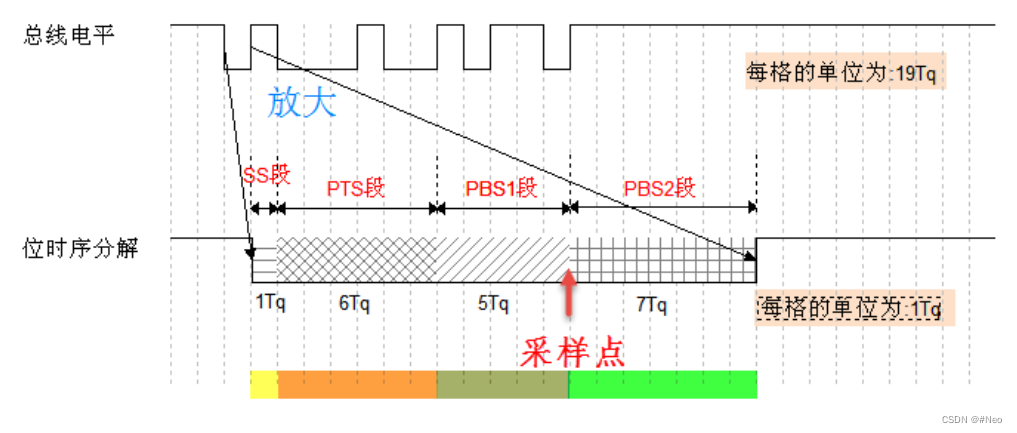
CAN属于异步通讯没有时钟线，所以需要约定波特率

位同步：

一种抗干扰的方式

一个数据位被分解成SS/PTS/PBS1/PBS2四段，其最小时间单位为Tq， 一般一个数据位有8-25Tq，各个节点只需要约定1Tq的大小长度以 及一位数据有多少Tq即可确定通讯波特率。

例:1Tq = 1us,1数据19Tq，波特率为 1 \* 10^6 / 19 == 52631



SS：同步段，当数据信号的跳变沿在SS段之内时，节点与总线才是 时序同步的。电平才为有效电平，大小为固定1Tq

PTS：传播时间段，用于补偿网络的物理延时，一般是输入比较器和 输出比较器延时总和的两倍

PBS1：相位缓冲段，补充边沿阶段误差，可在重新同步时加长

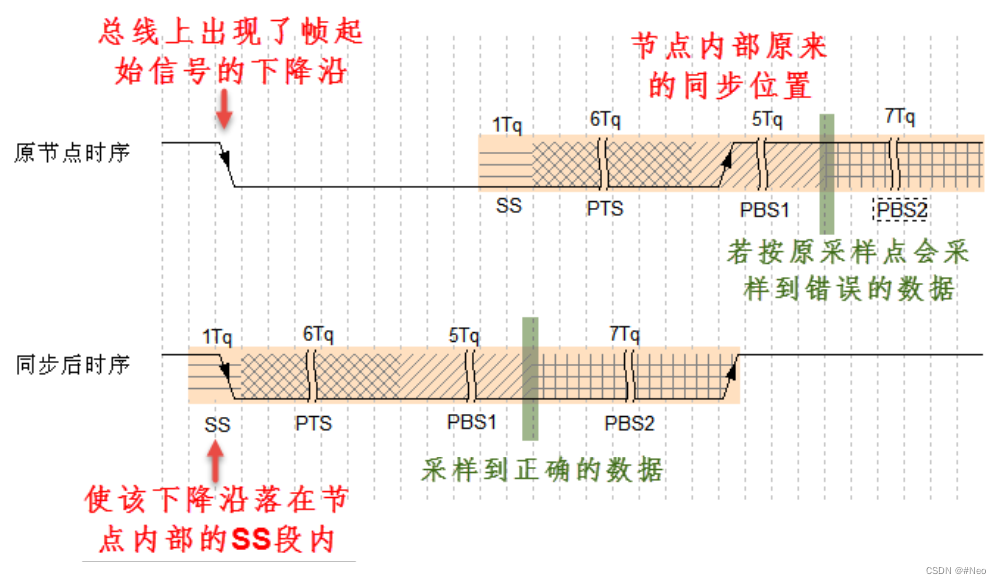
PBS2：同上

硬同步

节点发送数据时，会发送一个起始帧信号，该信号是一个下降沿。

当总线出现帧信号，某节点检测起始信号不在节点SS段范围内， 则判断为不同步。

此时则会把自己位时序的SS段平移至总线出现下降沿的部分来 获得同步。

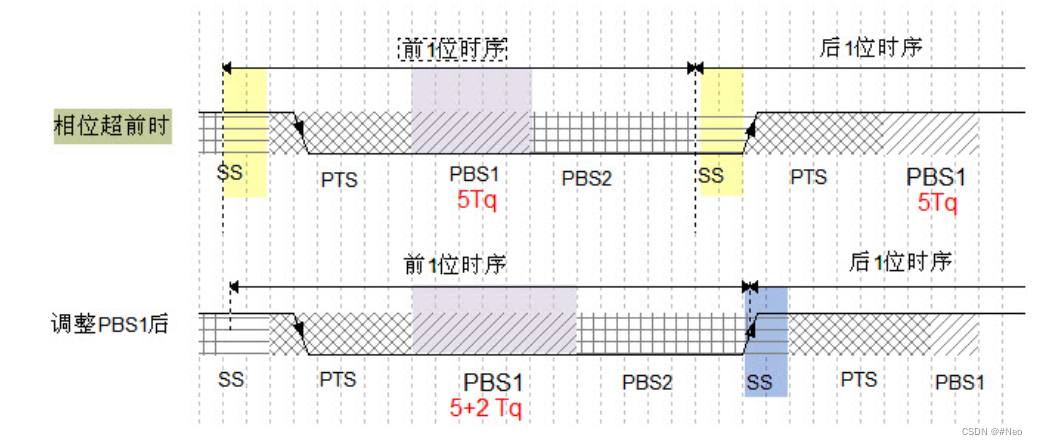


重新同步

在一帧很长的数据内，存在相位偏移是=时，往往用重新同步方 式。利用普通数据位的下降沿来同步。

超前：

节点从总线的边沿跳变中，检测到内部时序比总线时序超前 nTq，则会在下一个位序的PBS1中增加2Tq



滞后：

类似的，检测到节点内部时序比总线时序滞后nTq，则会在 控制器的前一位时序中，减少2Tq。

增加或减少的时间称为“重新同步补充宽度SJW”，一般会限制 该值的大小，也可通过多次增减进行调整

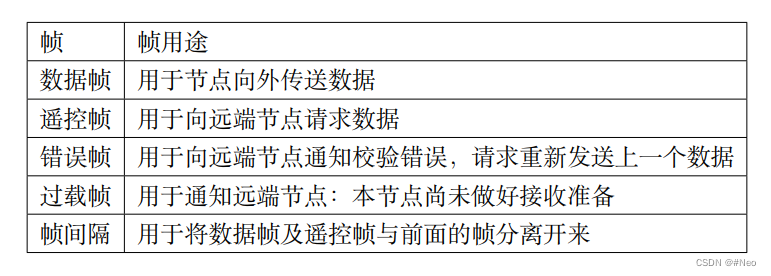
**CAN报文**

差分信号一次只能表达一个信号。

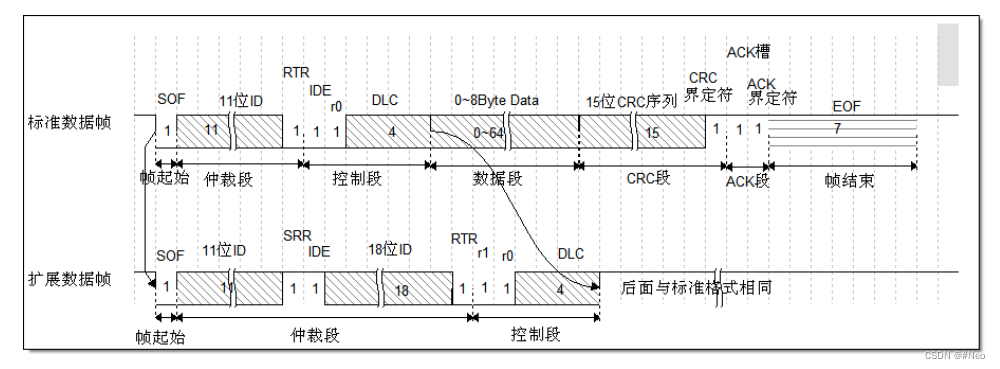
对数据进行打包，即给原始数据段前加上传输起始标签、片选标签。控制标 签等等。在数据段尾加上CRC校验标签、应答/结束标签等。

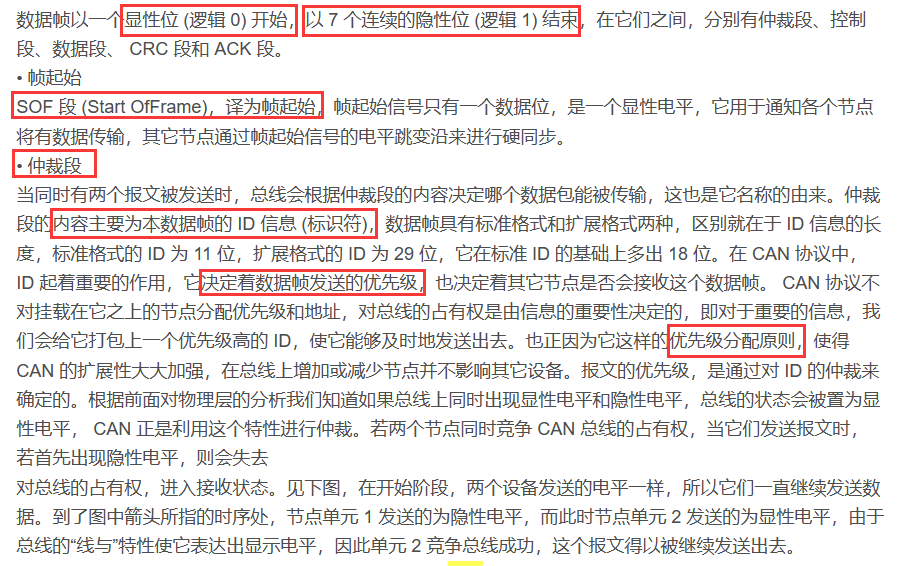
对于一个报文，只需要按照特定格式解读，即可还原数据类型。

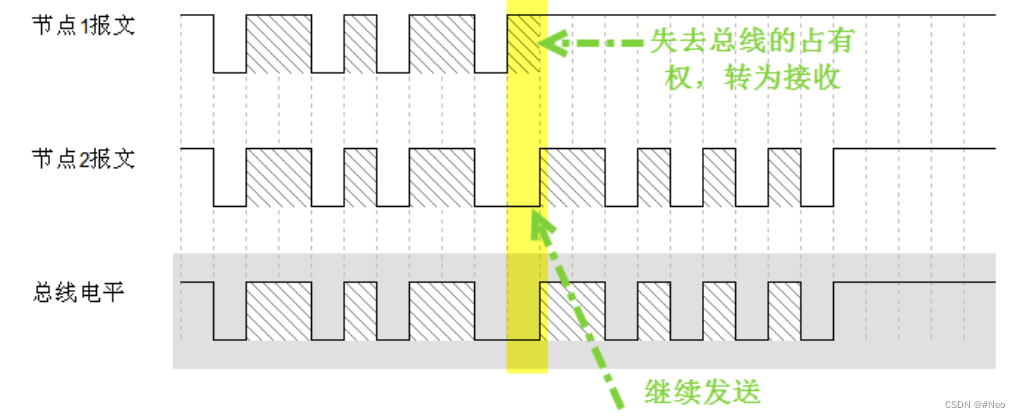
不同报文，是不同的帧。

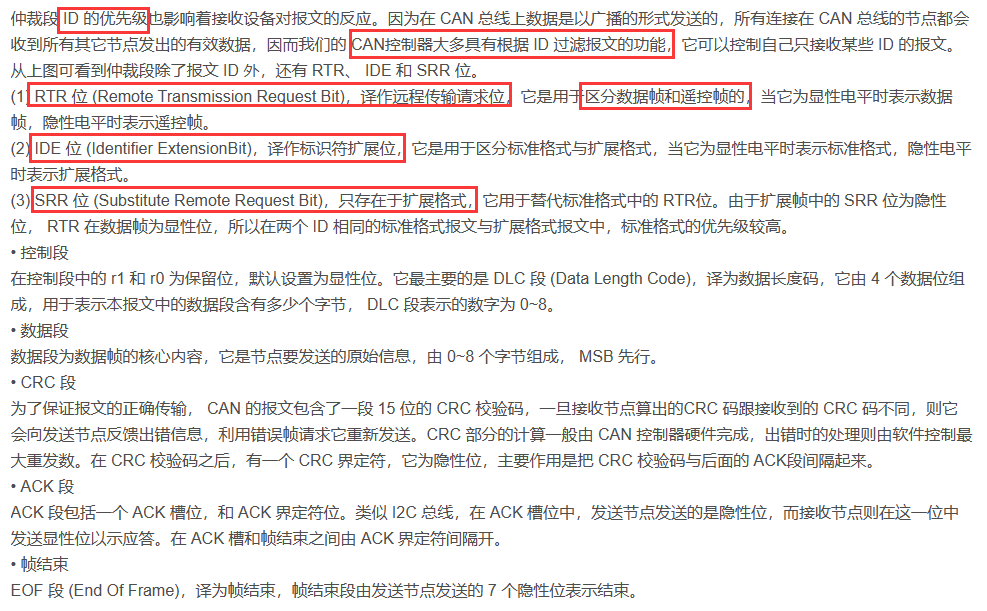


**数据帧结构**









## MQTT

[【MQTT协议详解】MQTT协议-CSDN博客](https://blog.csdn.net/CM_STC89C52/article/details/122712188?ops_request_misc=%7B%22request%5Fid%22%3A%22171349591616800211514927%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171349591616800211514927&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-122712188-null-null.142^v100^pc_search_result_base5&utm_term=MQTT&spm=1018.2226.3001.4187)

## Modbus

[详解Modbus通信协议---清晰易懂-CSDN博客](https://blog.csdn.net/as480133937/article/details/123197782?ops_request_misc=%7B%22request%5Fid%22%3A%22171350295016800227448572%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171350295016800227448572&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-123197782-null-null.142^v100^pc_search_result_base5&utm_term=Modbus&spm=1018.2226.3001.4187)

## SDIO

**传输模式**

**物理层**

**协议层**