物理层通信

|  |  |
| --- | --- |
| 编 制 人： | Lizy |
| 发布日期： | 2023.3.4 |
| 页 数： | 17 |

更改记录

|  |  |
| --- | --- |
| **发布日期** | **更改描述** |
| 20230304 | 第一版发布 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 通信基本概念



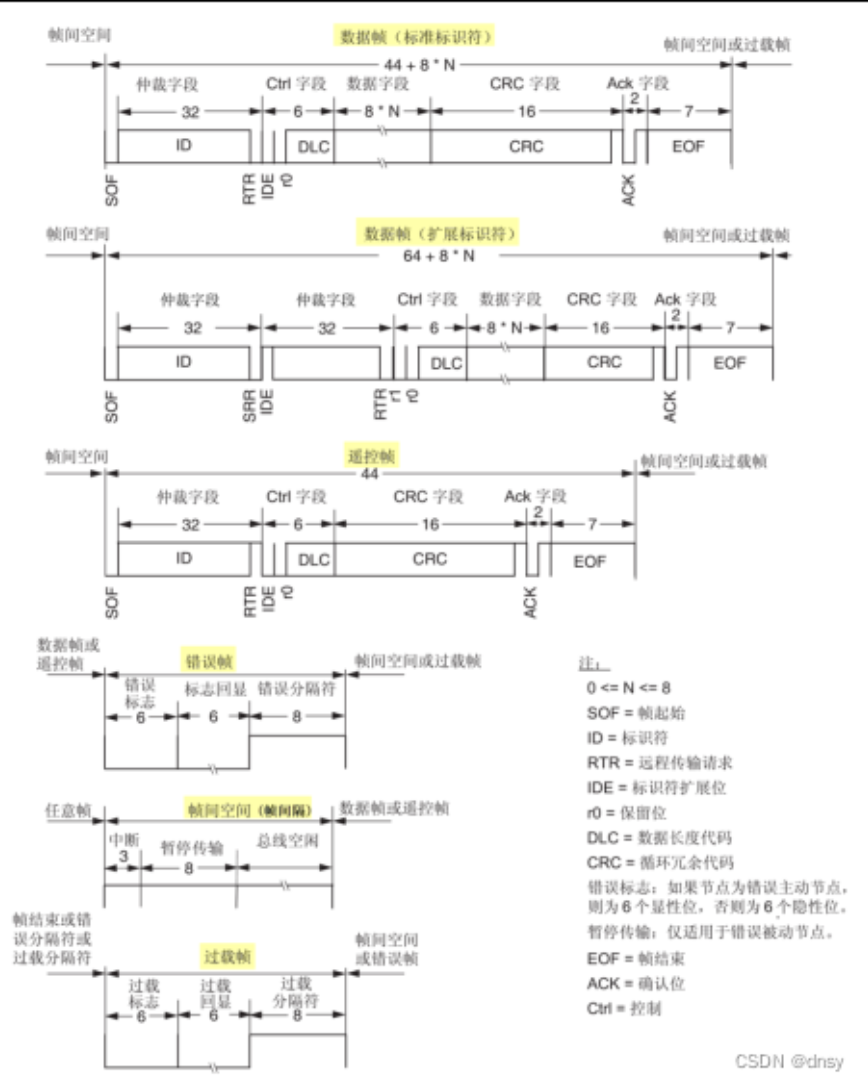
# 基本名词

|  |  |
| --- | --- |
| **名词** | **解释** |
| SCI | 串行通信的总称，可以包括串行同步通信SPI，也可以包括串行异步通信Uart, 区别于并行通信，表示一个字节一个字节的依次传输 |
| Uart | 通用异步收发器，是串行通信的一种通信协议，规定了串行通信的波特率、起始/停止位、数据位、校验位等格式，以及各种异步握手信号（0，1信号）；  通信双方只要采用相同的帧格式和波特率，就能在未共享时钟信号的情况下，仅用仅用两根信号线（Rx 和Tx）就可以完成通信过程 |
| RS232  RS485  RS422 | 物理接口电气标准，规定传输介质（电缆）、收发电平等。都是SCI，通信协议相同（都是Uart，只有波特率、起始/停止位、数据位、校验位等参数），仅传输介质和收发电平不同。  TTL型传输的uart，高电平最高为5V，低电平最低为0V，传输距离比较近；  RS232传输的uart，高电平最高为25V，低电平最低为-25V，传输距离中等；  RS485传输的uart，高电平最高为12V，低电平最低为-7V，传输距离最远； |
| 异步通信 | 通信中2个字符（8位（数据位构成一个字符））之间的时间是不固定的，而在一个字符内各位的时间间隔是固定的；  异步通信是按字符传输的，接收设备在收到起始信号之后只要在一个字符的传输时间内能和发送设备保持同步就能正确接收。下一个字符起始位的到来又使同步重新校准（依靠检测起始位来实现发送与接收方的时钟自同步的） |

# 通信协议

| **通信协议** | **基本介绍** | **数据格式** |
| --- | --- | --- |
| Uart | 1. 串行数据总线，用于**串行异步**通信； 2. 该总线双向通信，可以实现全双工传输和接收，（TX、RX）； 3. 串行通信即将传输的数据帧的一位一位的传输； | 带有格式信息的数据字符叫帧，一帧包含1位启动位、1~8位数据、1或2位停止位、可选择的奇偶校验位 |
| CAN | 1. 用于实时应用的的**串行通讯协议**总线，它可以使用**双绞线**来传输信号，是世界应用最广的**现场总线**之一； 2. **半双工**； 3. 采用数据位仲裁方式代替传统的地址轮询方式，因此可以支持多主多从的工作方式 4. CAN控制器根据两根总线上的**电位差**来判断总线电平。总线电平分为显性电平（逻辑0）和隐性电平（逻辑1）。显性电平对应逻辑电平为0，CAN-H与CAN-L之间的差为2.5V左右；隐性电平对应逻辑电平为1，CAN-H与CAN-L之间的差为0V左右；在总线上**显性电平具有优先权**，只要有一个单元输出显性电平，总线上即为显性电平。(CAN\_High、CAN\_Low) 5. 一个邮箱数据最大为8个字节（64位），分高、低2个邮箱   为了更有效的控制通信，CAN一共规定了5种类型的帧（报文）    数据帧：标准数据帧和扩展数据帧 | **1.发送过程：**    CAN控制器将CPU传来的信号转为逻辑电平（0或1），CAN发送器接收到逻辑电平后，再将其转换为差分信号输出到CAN总线上；  **2.接收过程：**    CAN接收器将CAN\_H和CAN\_L线上传来的差分电平转换成逻辑电平输出给CAN控制器，CAN控制器再把逻辑电平转成相应信号给CPU   1. CAN的报文种类及结构   当使用CAN协议进行通讯时，需要对数据、操作命令(如读/写)以及同步信号进行打包，打包后的这些内容称为报文。  **（1）数据帧：**以一个显性位（逻辑0）开始，以连续7个隐形位（逻辑1）结束。在他们之间，分别有**仲裁段**、**控制段**、**数据段**、**CRC段**和**ACK段**。  **1）帧起始：**SOF段，1位，显性位（逻辑0），它用于通知各个节点将有数据传输，其他节点通过帧起始信号的电平跳变沿来进行硬同步；  **2）仲裁段：**包括2部分：本数据帧的**ID信息（标识符）**和远程发送请求位**RTR；**  **ID标识符：**  当同时有两个报文被发送时，总线会根据**仲裁段的ID信息决定**哪个数据能被传输；  数据帧具有标准格式和扩展格式两种，区别在ID标识符的长度上，标准格式的ID为**11位**，扩展格式的ID为**29位**；  CAN协议中，ID起着重要作用，它决定数据帧发送的优先级，也决定着其他节点是否接收这个数据帧。  **报文的优先级**，是通过对ID的仲裁来确定的。根据前面对物理层的分析我们知道如果总线上同时出现显性电平和隐性电平，总线的状态会被置为显性电平，CAN正是利用这个特性进行仲裁。    CAN总线上数据是以广播的形式发送的，所有连接在CAN总线的节点都会收到其他节点所发送的有效数据，CAN控制器具有**ID过滤报文**的功能，只接受与自己匹配的某些报文。  **远程发送请求位RTR：**  **1位，**用来区分数据帧（显性位）和遥控帧（远程帧，隐形位）；  **3）控制段：**共6bit，分别是**扩展标识符IDE**（1位）、保留位R0（1位）、**数据长度编码位**DLC（4位）；  **4）数据段：**数据段为数据帧的核心内容，它是节点要发送的原始信息，由**0~8个字节**组成，**MSB**先行；  **5）循环校验段CRC**：包括15bit的CRC序列和1bit的界定符DEL（表示循环校验序列结束）  **CRC序列**  一旦接收节点算出的CRC码跟接收到的CRC码不同，则它会向发送节点反馈出错信息，利用**错误帧**请求它重新发送。CRC部分的计算一般由CAN控制器硬件完成，出错时的处理则由软件控制最大重发数。在CRC校验码之后，有一个CRC界定符，  **CRC界定符DEL**  为隐性位，主要作用是把CRC校验码与后面的ACK段间隔起来，表示循环校验序列结束；  **6）确认段ACK**：包括1bit的ACK槽位和1bit的ACK界定符位  在ACK槽位中，发送节点发送的是隐性位，而接收节点则在这一位中发送显性位以示应答。在ACK槽和帧结束之间由ACK界定符间隔开；  **7）帧结束EOF**：由发送节点发送的7个隐性位表示结束。  **（2）远程帧：**通过发送远程帧，作为数据接收器的节点可以发起各自数据源的数据传送清求，即向数据发送器请求发送具有相同id的数据帧；  选程帧包括帧开始、 仲裁段、控制段、CRC段、ACK段和帧结束。远程帧没有数据段；  **（3）错误帧**：有两种错误标志:主动(正）错误标志和被动（负）错误标志。活动错误标志：由6个连续的显示位组成。 |
| SPI | 1. **全双工**短距离单主设备通信协议，与Uart不同，他是**串行同步通信**协议； 2. 连接方式之一：主从连接。 3. 支持一主多从。在SPI总线上，某时刻可以出现多个从机，但只能存在一个主机，主机通过片选信号来确定要通信的从机，这就要求从机的MISO口具有三态性，使得该口在器件未被选中时表现为高阻态 4. 可以同时发送和接收数据。发送的数据先到发送数据缓冲区，再到移位寄存器中按位传送，数据都是以字节为单位，高位优先。 5. SPI通信传输过程其实就是主、从设备数据交换的一个过程，MOSI 和 MISO在一个时钟周期内都会进行数据位的发送和接收，如果不需要接收的数据，直接忽略就可以了，不做处理。 6. 四种工作模式。时钟极性（CKP/CPOL）和时钟相位（CKE/CPHA）共同决定SPI读取数据的模式。   **时钟极性**：设为“0”表示空闲状态时钟为低电平，  设为“1”表示空闲状态时钟为高电平；  **时钟相位**：设为"0"表示在时钟信号的第一个跳变沿采样；  设为"1"表示在时钟信号的第二个跳变沿采样；  **模式0 (CPOL = 0,CPHA = 0)：**  在空闲状态下，SCLK处于低电平，在第1个时钟边沿（跳变）进行数据采样，采样完在下一个跳变沿发送数据。  **模式1 (CPOL = 0,CPHA = 1)**  在空闲状态下，SCLK处于低电平，在第2个时钟边沿进行数据采样，第一个时钟沿进行发送数据  **模式2 (CPOL = 1,CPHA = 0)**  在空闲状态下，SCLK处于高电平，在第1个时钟边沿进行数据采样，采样完在下一个跳变沿发送数据。  **模式3 (CPOL = 1,CPHA = 1)**  在空闲状态下，SCLK处于高电平，在第2个时钟边沿进行数据采样，第一个时钟沿进行发送数据。 | **1.接口：四根线**  SCL：串行时钟，用来同步数据传输，由主机输出；  MISO：主入从出；  MOSI：主出从入  CS/SS：片选信号，低电平有效，由主机输出  **2.传输方式**    一个SPI时钟周期内，会完成：  主机通过MOSI线发送一位数据，从机通过该线读取这一位数据；  从机通过MISO线发送一位数据，主机通过该线读取这一位数据。  这是通过移位寄存器来实现的。主机和从机各有一个移位寄存器，且二者连接成环。随着时钟脉冲，数据按照从高到低的方式依次移出主机寄存器和从机寄存器，并且依次移入从机寄存器和主机寄存器。当寄存器中的内容全部移出时，相当于完成两个寄存器内容的交换。  **3.工作模式：** |
| I2C | 1. 两条线连接的**半双工**协议（数据线：SDA、时钟线：SCL）； 2. **串行同步通信；** 3. 总是要通过电阻上拉至输入电压，工作于三种速度：高速、快速、慢速； 4. 每个连接到总线上的器件都有**唯一的地址**，任何器件既可以作为主机也可以作为从机，但同一时刻只允许有一个主机 5. 传输方式：   1)如果主机要发送数据给从器件，则主机首先寻找从器件，然后主动发送数据至从器件，最后由主机终止数据传输；  2) 如果主机要接收从器件的数据，首先由主机寻找从器件，然后主机接收从器件发送的数据，最后由主机终止接收过程。这种情况下，主机负责产生定时时钟和终止数据传送； | **1.数据格式：**  每个字节必须是8位，最后一位为响应位；  每次传输可以发送的字节数量不受限制；  最开始发送的是从机地址    **2.应答响应：**  **地址匹配**完成就会得到响应信号（ACK），说明从设备存在；  **开始信号**：主机让时钟SCL保持高电平，数据SDA由高电平跳到低电平；  **停止信号**：主机让时钟SCL保持高电平，数据SDA由低电平跳到高电平；  **响应信号**（ACK）：接收器在接收到8位数据后，在第9个时钟周期拉低SDA为低电平；  SDA上传输的数据必须在SCL为高电平的期间保持稳定，外部设备会在SCL为高电平时读取数据；  SDA的数据只能在SCL为低电平期间发生变化。 |
| Arinc429 | 1. 数据总线，民用飞机机载总线规范； 2. 是一种**串行标准**，为面向接口型的**单向广播式**传输总线。该总线上只允许有一个发送器，但可以有多个接收器（最多为20个）。 3. 以**差动输出**的对称(平衡)方式工作。 4. 采用**双绞屏蔽线异步**传输数据。 5. 总线用**标志码字**来区分设备和信号名称 6. 速率为12. 5 kbps 或100 kbps，通常以脉冲形式发送。 | **1.数据：**  ARINC-429通信采用带有**奇偶校验的32位信息字**，采用双极性归零码的三态调制编码方式，调制信号由“高”、“零”、“低”三种电平状态。  接收器输入端（考虑到干扰情况下）：  高电平为+6.5V~+13V；  零电平为- 2.5V~ + 2.5V；  低电平为-6.5V~-13V。  数据传输中，发送端以足够高的速率传输数据，按开环方式控制传输，不需要接收器通知发送器已接收到信息。  **2.总线规范**  每个字格式由**32位**组成。  在连续传输的字与字之间至少有4个位的时间间隔。  对于每一个字的同步，可通过检测每个字第一位的跃变来实现。  在发送每组数据后有四位零周期，它是隔离符号，以便于发送下一组数据    **数据传输顺序**： 7,6,5,4,3,2,1,0, 8,9,10,11,12……29,30,31  数据字主要包括五部分:  **（1）标号( LABEL)** 占8位（0bit-7bit）  信息标识符，标号用**八进制**表示。  它标记出包括在这个传送字内的信息的类型，也就是传送的代码的意义是什么。  如传送的是VHF信息，则标号为八进制数030；  若是DME数据，则标号为八进制数201等  **标号位发送数据顺序与正常顺序相反, 先发送高字位再发送低字位**。  **（2）源/ 目的标识( SDI)**  占2位（8bit-9bit）  它指示信息的来源或信息的终端。  当需要将一些专用字传输到一个多系统的特定系统时，就可以用来识别字的目的地址。也可以根据字内容来判明一个多系统的原系统。  **（3）数据组(DATA)**  占19位（10bit-28bit）  根据字的类型可确定为是18还是19位。  它所代表的是所确定的特定数据。如标号为030，则10到28位为频率数据，使用的是**BCD编码数据格式**。  **（4）符号/状态位(SSM)**  占2位（29bit-30bit）  根据字的类型号可确定为28到29到31。  它指出数据的特性，如方向、符号等（南，北，正，负）。    **（5）奇偶校验位(P)**  占1位（31bit）  用于检查发送的数据是否有效。 |
|  |  |  |

CAN 的帧格式：



# 常见通信方式

| **通信**  **类别** | **通信**  **协议** | **接口/传输方式** | **传输距离** | **传输速度** | **优点** | **缺点** | **备注** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| RS232 | Uart  串行异步通信 | 三根线，TX、RX、GND，**全双工** | 最大50m | 20Kbps | 全双工效率高 | 传输距离短，抗干扰差容易产生共模干扰 | 1.一对一通信  2.不可以与TTL兼容 |
| RS485 | Uart  串行异步通信 | 差分信号，+-两根线，**半双工** | 比RS232长，最大1200m  (9600bps) | 最高传输速度10Mbps,但是速度越高，传输距离越短，所以往往设为9600bps | 传输距离长，多节点，差分信号抗干扰强，**工业应用首选** | 半双工效率低 | 1.RS485牺牲了232全双工的效率来达到远距离传输；  2.可以一对一，也可以一对多  3.可以与TTL兼容 |
| RS422 | Uart  串行异步通信 | **2个485**，4根线，**全双工** |  |  | 全双工效率高，传输距离长 | 线太多，问答式（一问一答）通信体现不出优势 | 既像232一样全双工，又像485有较远的传输距离 |
| CAN | CAN  串行异步通信 | 由控制器和收发器组成，  双绞线，差分信号CAN\_High、CAN\_Low，**半双工** | 通信距离远  最远10km（速率低于5kbps）,  最短小于40m(通信速率1Mbps) | 通信速度快，  最高1Mbps，通常500kbps、250kbps | 多节点，通信距离远，通信速度快，具有错误检测、错误通知、错误恢复功能，抗干扰性强 | 1.半双工效率低；  2.长数据传输时数据负载偏低（以扩展帧为例，其29位ID和64位的数据，导致数据承载只有60%）；  3. 长数据（大于8字节）传输分帧丢帧，会导致整包传输失败。 | 多主控制：在总线空闲时，所有单元都可以发送信息，而两个以上单元同时发送信息时，根据标识符（ID）决定优先级，优先级高的单元可以继续发送信息，其他单元停止发送转为接受工作 |
| SPI | SPI  **串行同步**通信 | 4根线，有一个时钟信号，**全双工** | 可靠传输距离一般为1-3米， | 速度比I2C快；  SPI是一种事实标准，由Motorola开发，并没有一个官方标准。已知的有的器件SPI已达到**50Mbps** | 全双工；  简单的软件配置；  数据传输速率较高；  数据传输大小可任意配置 | 占用主机较多的口线（每个从机需要一根片选线）；  只支持单主机；  没有应答机制确认是否接收到数据；  传输距离短 | 驱动能力有限，一般不超过10个 |
| Arinc429 | Arinc429  **串行异步**通信 | **差分信号，全双工** | 传输距离较远，最远几百米 | 速率为12. 5 kbps 或100 kbps， | 结构简单、性能稳定、抗干扰性强；  最大的优势是可靠性高，这是由于非集中控制、传输可靠、错误隔离性好 |  |  |
| I2C | I2C  **串行同步通信** | 一根时钟线和一根数据，半双工 | 近距离，总线仲裁机制 | 没有严格的波特率要求；  传输速度四种模式：  标准模式：100 Kbps  快速模式：400 Kbps  高速模式：3.4 Mbps  超快速模式：5 Mbps | 所有组件之间都存在简单的主/从关系，连接到总线的每个设备均可通过唯一地址进行软件寻址；  I2C是真正的多主设备总线，可提供仲裁和冲突检测；  传输速度分为四种模式： | 半双工效率低；  距离短，适合电路板上的通讯 | 1.I2C具有冲突检测机制和仲裁机制的真正意义上的**多主机总线**，它能在多个主机同时请求控制总线时利用仲裁机制避免数据冲突并保护数据。  2. 最大主设备数：无限制；  最大从机数：理论上是127。 |

总结：

**速度上：SPI > I2C≈CAN > UART**

**I2C总线速度一般普遍慢于SPI**，I2C一般的速度是100kbs、400kbps和1Mbps，而SPI的速度可以在几Mbps，也可以10+Mbps。UART和SPI、I2C不同，他是异步传输的，一般来说传输速度比较慢，传统的传输速度一般在115200bps或以下。

CAN和I2C看不同场景，差不多。

**距离上：RS485 > RS232 > CAN > SPI > I2C**

RS485比RS232长，最大1200m(9600bps)，RS232是50m，CAN的距离一般为40m以内（提高传输速度），SPI为1-3米，I2C一般用作板件通讯，如E2PROM、传感器等。

# 补充

## RS232、RS485

这两个属于七层网络模型中的**物理层**，物理层定义了电平模式及两端的物理接口要求等

## CAN、Ethernet

这两个不是单纯的属于某一层，而是对其中的**物理层、链路层**进行了定义。比如在链路中分别叫CAN帧、以太网帧。具体在物理层和链路层中定义自行查阅。

## CAN Open、EtherCat

这两个是属于**应用层**的协议，只不过指定了自己要用的物理层及链路层，分别使用CAN及Ethernet定义的物理层及链路层。

## Modbus

Modbus也是**应用层**协议，只不过它不对下层进行指定，既可以使用RS232、RS484作为物理层，也可以用Ethernet定义的物理层、链路层。其中使用RS232、RS485时需要用URAT作为链路层对物理层进行指挥。使用Ethernet时则配合TCP、IP分别作为传输层和网络层，当然了，其实TCP/IP本身就是选用Ethernet作为链路层及物理层的。