# CAN总线学习

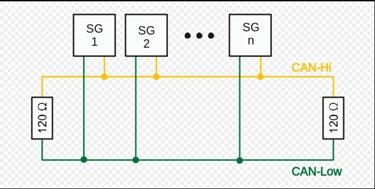
## 基本概念

CAN 是**Controller Area Network** 的缩写（以下称为CAN）即**控制器局域网络**，是ISO国际标准化的串行通信协议。

## CAN总线的物理结构与特性

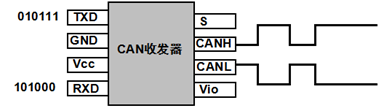
### CAN总线网路

**CAN总线网络主要挂在CAN\_H和CAN\_L**，各个节点通过这两条线**实现信号**的**串行差分传输**，为了避免信号的反射和干扰，还需要在CAN\_H和CAN\_L之间接上120欧姆的终端电阻，但是为什么是120欧姆呢？那是因为电缆的特性阻抗为120欧。



### CAN收发器

CAN收发器的**作用**是**负责逻辑电平**和**信号电平之间的转换**。



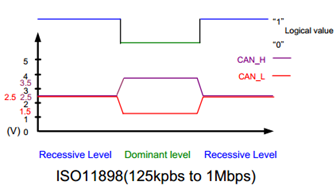
即从CAN控制芯片**输出逻辑电平到CAN收发器**，然后经过CAN收发器**内部转换**将逻辑电平转换为**差分信号输出**到**CAN总线**上，CAN总线上的节点都可以决定自己是否需要总线上的数据。具体的管教定义如下：



### 信号表示

CAN总线采用**不归零码位填充**技术，也就是说CAN总线上的信号有**两种不同的信号状态**，分别是**显性**的（**Dominant**）**逻辑0**和**隐形**的（**recessive**）**逻辑1**，**信号每一次传输完后不需要返回到逻辑0（显性）的电平。**

**位填充规则：发送器只要检测到位流里有5个连续相同值的位，便自动在位流里插入补充位。**



可以看到上图中的当第一段为隐性(recessive)，CAN\_H和CAN\_L电平几乎一样，也就是说CAN\_H和CAN\_L电平很接近甚至相等的时候，总线表现隐性的,而两线点位差较大时表现为显性的，按照定义的：

**CAN\_H-CAN\_L < 0.5V 时候为隐性的**，逻辑信号表现为**"逻辑1"**- 高电平。

**CAN\_H-CAN\_L > 0.9V 时候为显性的**，逻辑信号表现为**"逻辑0"**- 低电平。

即：

|  |  |  |
| --- | --- | --- |
| **状态** | **逻辑信号** | **电压范围** |
| 显性Dominant | 0 | CAN\_H-CAN\_L > 0.9 |
| 隐性recessive | 1 | CAN\_H-CAN\_L < 0.5V |

### 通讯速率与通讯距离

|  |  |  |  |
| --- | --- | --- | --- |
| **类型** | **标准** | **最高速率** | **描述** |
| 高速CAN | CAN/ISO 11839-2 | 1Mbit/秒 | 最通用的CAN总线类型 |
| 低速CAN | ISO/ISO 11839-3 | 125Kbit/秒 | 容错，在一条总线短路的时候仍然能工作 |
| 单线CAN | SAE J2411 | 50Kbit/秒 | 高速模式可达到100Kbit/s主要用在汽车上，例如通用公司 |

CAN总线上**任意两个节点**的**最大传输距离**与其**位速率**有关，如下表：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 位速率/kbps | 1000 | 500 | 250 | 125 | 100 | 50 | 20 | 10 | 5 |
| 最大距离/m | 40 | 130 | 270 | 530 | 620 | 1300 | 3300 | 6700 | 10000 |

这里的**最大通信距离指的是同一条总线上两个节点之间的距离**。可以看到速率越低通讯距离就越远，也就是说**CAN总线的通讯距离和波特率成反比**。在位速率为5千比特位每秒的时候达到最大的传输距离10公里。其中一般的工程中比较常用的为500K每秒的通讯速率。这个速率在实际测试的时候也是非常可靠的。

## CAN总线协议

### CAN总线协议

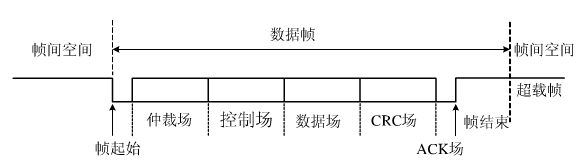
CAN 总线是一个广播类型的总线，所以任何在总线上的节点都可以监听总线上传输的数据。也就是说总线上的传输不是点到点的，而是**一点对多点的传输**，这里多点的意思是总线上所有的节点。但是总线上的节点如何知道那些数据是传送给自己的呢？CAN总线的硬件芯片提供了一种叫做**本地过滤的功能**，通过这种本地过滤的功能可以过滤掉一些和自己无关的数据，而保留一些和自己有关的信息。

### CAN总线消息机制

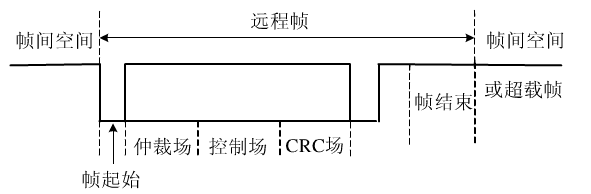
CAN标准定义了**四种消息类型**，每条消息用一种叫做**比特位仲裁**(Arbitration)机制来控制进入CAN总线，并且每条消息都标记了优先权。另外CAN标准还定义了一系列的错误处理机制。

CAN报文的四种消息类型：

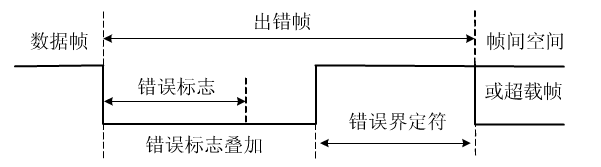
**数据帧**：数据帧将数据从发送器传输到接收器。



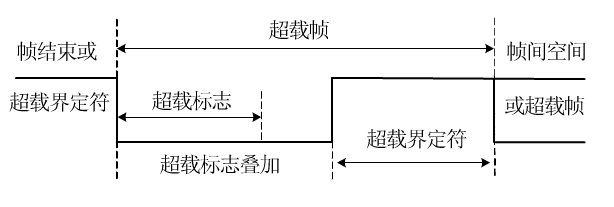
**远程帧**：总线单元发出远程帧，请求发送具有同一标识符的数据帧。



**错误帧**：任何单元检测到总线错误就发出错误帧。

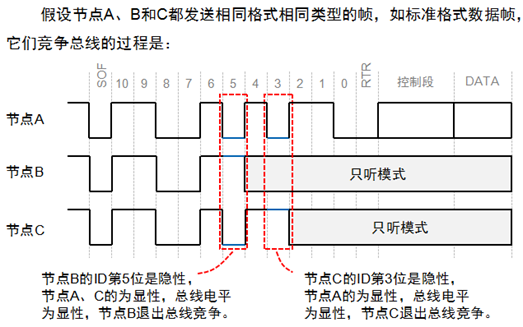


**过载帧**：过载帧用在相邻数据帧或远程帧之间的提供附加的延时。



### 仲裁

只要总线空闲，总线上任何节点都可以发送报文，如果有两个或两个以上的节点开始传送报文，那么就会存在**总线访问冲突**的可能。但是**CAN使用了标识符的逐位仲裁方法可以解决这个问题**。



在仲裁期间，每一个发送器都对**发送的电平与被监控的总线电平**进行**比较**。如果电平相同，则这个单元可以继续发送。如果发送的是一"隐性"电平而监视到的是一"显性"电平，那么这个节点失去了仲裁，必须退出发送状态。如果出现不匹配的位不是在仲裁期间则产生**错误事件**。

**帧ID越小，优先级越高。**由于数据帧的RTR位为显性电平，远程帧为隐性电平，所以帧格式和帧ID相同的情况下，**数据帧优先于远程帧**；由于标准帧的IDE位为显性电平，扩展帧的IDE位为隐形电平，对于前11位ID相同的标准帧和扩展帧，**标准帧优先级比扩展帧高。**