

由于 Socket CAN 涉及到 CAN 总线协议、套接字、Linux 网络设备驱动等。因此，为了能够全面地了解 Socket CAN 的原理。我们需要了解以下几个方面的知识点：

- ( 1 ) CAN 总线协议；
- ( 2 ) Socket 原理；
- ( 3 ) Linux 网络设备驱动；

当熟悉以下三个方面的知识点后，我们再去分析基于 Linux 的 Socket CAN 的驱动。这样的话理解起来更加容易、易懂。

- ( 4 ) Socket CAN 的驱动；

## 一、CAN 总线协议

由于 CAN 总线协议的内容太多，作为博文来说，不适宜很详细的讲解。需要深入了解的朋友们可以 Google 一下。以下只是作些简要的说明。

CAN 是 ControllerArea Network ( 控制器局域网 ) 的缩写。CAN 通信协议在 1986 年由德国电气商博世公司所开发，主要面向汽车的通信系统。现已是 ISO 国际化的串行通信协议。根据不同的距离、不同的网络，可配置不同的速度，最高速度为 1MBit/s。

CAN 被细分为三个层次：

- ( 1 ) CAN 对象层 ( the object layer ) ；
- ( 2 ) CAN 传输层 ( the transfer layer ) ；
- ( 3 ) CAN 物理层 ( the physical layer ) ；

对象层和传输层包括所有由 ISO/OSI 模型定义的数据链路层的服务和功能。

对象层的作用范围包括：

- ( 1 ) 查找被发送的报文。
- ( 2 ) 确定由实际要使用的传输层接收哪一个报文。
- ( 3 ) 为应用层相关硬件提供接口。

传输层的作用主要：

- ( 1 ) 传送规则，也就是控制帧结构、执行仲裁、错误检测、出错标定、故障界定。
- ( 2 ) 总线上什么时候开始发送新报文及什么时候开始接收报文，均在传输层里确定。
- ( 3 ) 位定时的一些普通功能也可以看作是传输层的一部分。
- ( 4 ) 传输层的修改是受到限制的。

物理层的作用：

在不同节点之间根据所有的电气属性进行位信息的实际传输。当然，同一网络内，物理层对于所有的节点必须是相同的。尽管如此，在选择物理层方面还是很自由的。

应用层
对象层
- 报文滤波
- 报文和状态的处理
- 传输层
- 故障界定
- 错误检测和标定
- 报文校验
- 应答
- 仲裁
- 报文分帧
- 传输速率和定时
物理层
- 信号电平和位表示
- 传输媒体

图 1 CAN 协议所对应的 ISO 模型

CAN 具有以下属性：

- ( 1 ) 报文 ( Messages ) ：简单来说就是具有固定格式的数据包。
- ( 2 ) 信息路由 ( Information Routing ) ：即，报文寻找结点的方式。
- ( 3 ) 位速率 ( Bit rate ) ：数据位的传输速度。
- ( 4 ) 优先权 ( Priorities ) ：即报文发送的优先权。
- ( 5 ) 远程数据请求 ( Remote Data Request ) ：通过发送远程帧，需要数据的节点可以请求另一节点发送相应的数据帧。
- ( 6 ) 多主机 ( Multimaster ) ：总线空闲时，任何结点都可以开始传送报文。

( 7 ) 仲裁 ( Arbitration ) : 当 2 个及以上的单元同时开始传送报文 , 那么就会有总线访问冲突。仲裁是确定哪个单元的具有发送优先权。

( 8 ) 安全性 ( Safety ) : CAN 的每一个节点均采取了强有力的措施以进行错误检测、错误标定及错误自检。

( 9 ) 错误检测 ( Error Detection ) : 包括监视、循环冗余检查、位填充、报文格式检查。

( 10 ) 错误检测的执行 ( Performance of Error Detection )

( 11 ) 错误标定和恢复时间 ( Error Sinalling and Recovery Time ) : 任何检测到错误的结点会标志出已损坏的报文。此报文会失效并将自动地开始重新传送。如果不再出现新的错误 , 从检测到错误到下一报文的传送开始为止 , 恢复时间最多为 29 个位的时间。

( 12 ) 故障界定 ( Fault Confinement ) : CAN 结点能够把永久故障和短暂扰动区分开来。永久故障的结点会被关闭。

( 13 ) 连接 ( Connections ) : CAN 串行通讯链路是可以连接许多结点的总线。理论上 , 可连接无数多的结点。但由于实际上受延迟时间或者总线线路上电气负载的影响 , 连接结点的数量是有限的。

( 14 ) 单通道 ( Single Channel ) : 总线是由单一进行双向位信号传送的通道组成。

( 15 ) 总线值 ( Bus value ) : 总线可以具有两种互补的逻辑值之一 : **“显性” ( 可表示为逻辑 0 ) 或 “隐性” ( 可表示为逻辑 1 )**。

( 16 ) 应答 ( Acknowledgment ) : 所有的接收器检查报文的连贯性。对于连贯的报文 , 接收器应答 ; 对于不连贯的报文 , 接收器作出标志。

( 17 ) 睡眠模式 / 唤醒 ( Sleep Mode / Wake-up ) : 为了减少系统电源的功率消耗 , 可以将 CAN 器件设为睡眠模式以便停止内部活动及断开与总线驱动器的连接。CAN 器件可由总线激活 , 或系统内部状态而被唤醒。

1、CAN 总线的报文格式

CAN 传输的报文，可分为五种类型：

- （ 1 ）数据帧：用于发送结点向接收结点传送数据的帧。
- （ 2 ）远程帧：总线结点发出远程帧，请求发送具有同一识别符的数据帧。
- （ 3 ）错误帧：任何结点检测到一总线错误就发出错误帧。
- （ 4 ）过载帧：过载帧用以在先行的和后续的数据帧（或远程帧）之间提供一附加的延时。
- （ 5 ）帧间隔：用于将数据帧及远程帧与前面的帧分离开来的帧。

数据帧由 7 个不同的位场组成：

帧起始	仲裁场	控制场	数据场	CRC 场	ACK 场	帧结束
-----	-----	-----	-----	-------	-------	-----

数据帧有标准格式和和远程格式，以下是其格式表示：

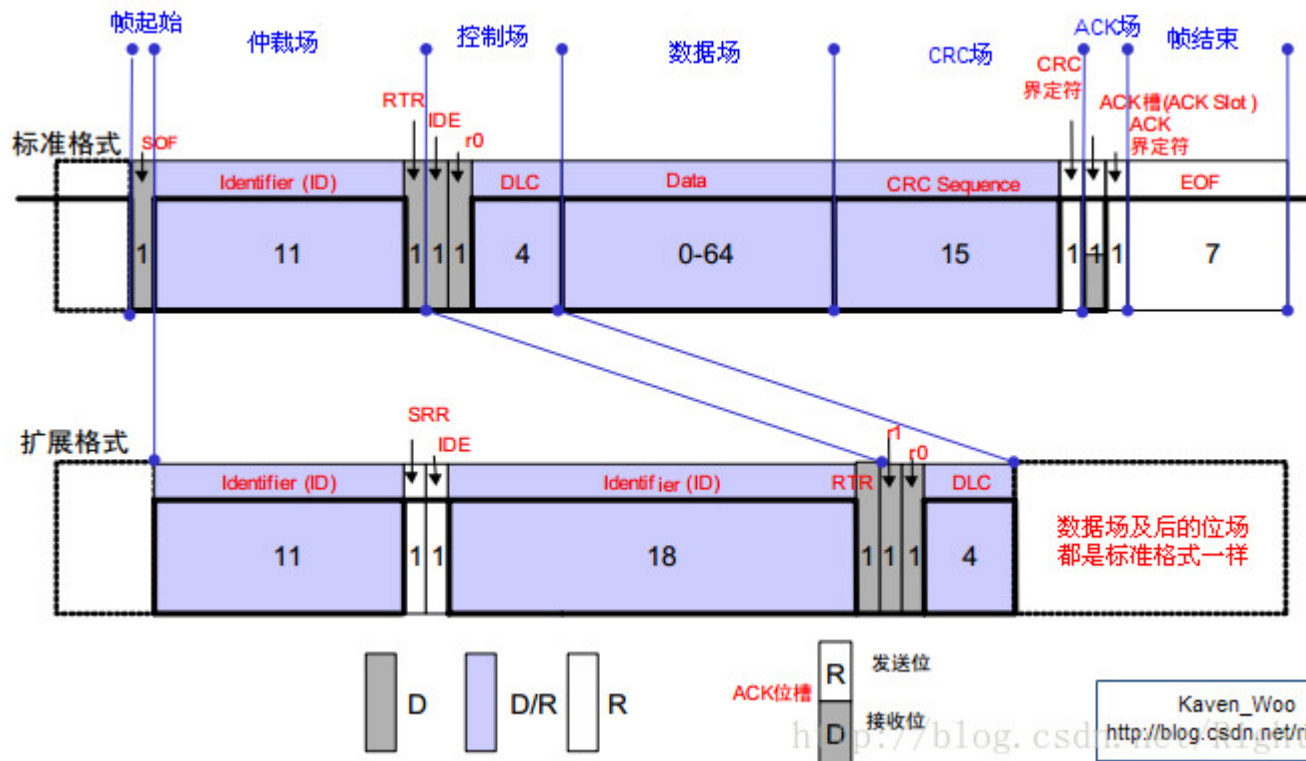


图 2 数据帧格式

远程帧由 6 个不同的位场组成：

帧起始	仲裁场	控制场	CRC 场	ACK 场	帧结束
-----	-----	-----	-------	-------	-----

远程帧没有数据帧的数据场，以下是其格式表示：

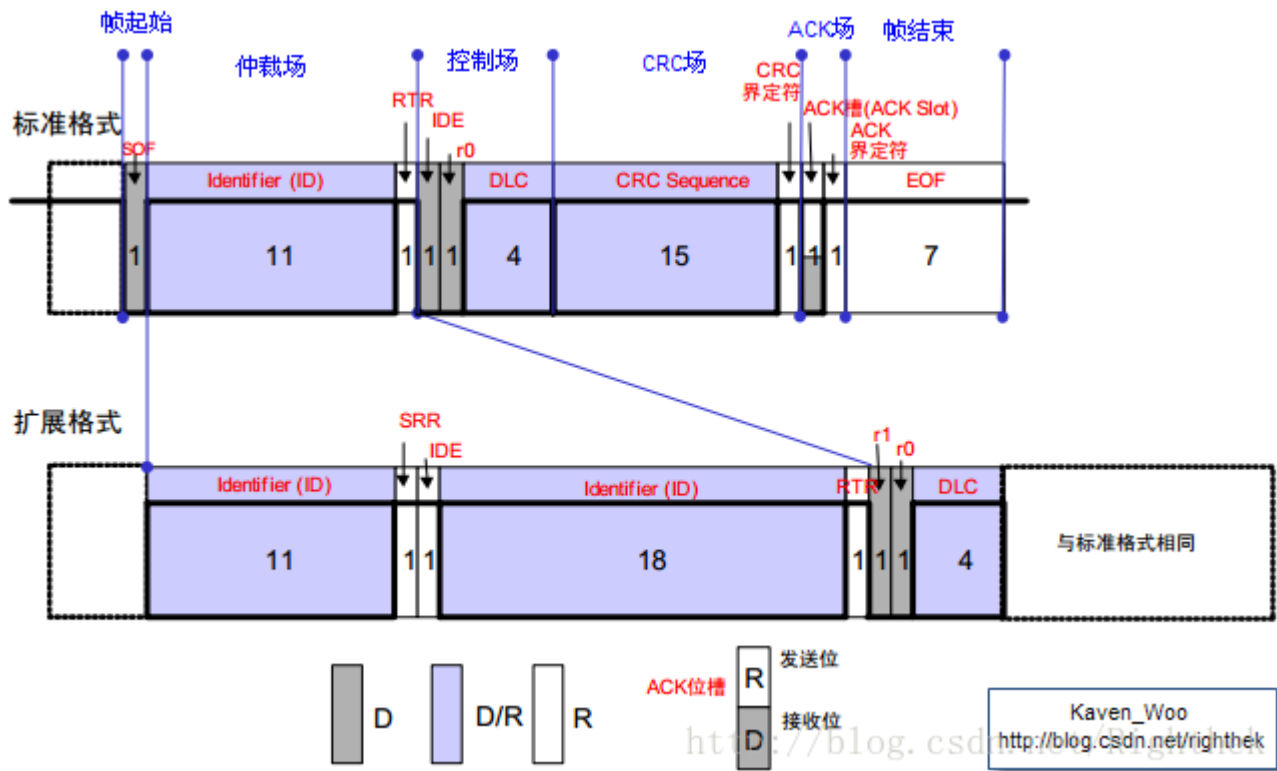


图 3 远程帧格式

错误帧用于在接收和发送消息时检测出错误，通知错误的帧。错误帧由错误标志和错误界定符构成。

错误标志包括主动错误标志和被动错误标志两种。

主动错误标志：6 个位的显性位，处于主动错误状态的单元检测出错误时输出的错误标志。

被动错误标志：6 个位的隐性位，处于被动错误状态的单元检测出错误时输出的错误标志。

错误界定符由 8 个位的隐性位构成。

错误帧格式如下表示：

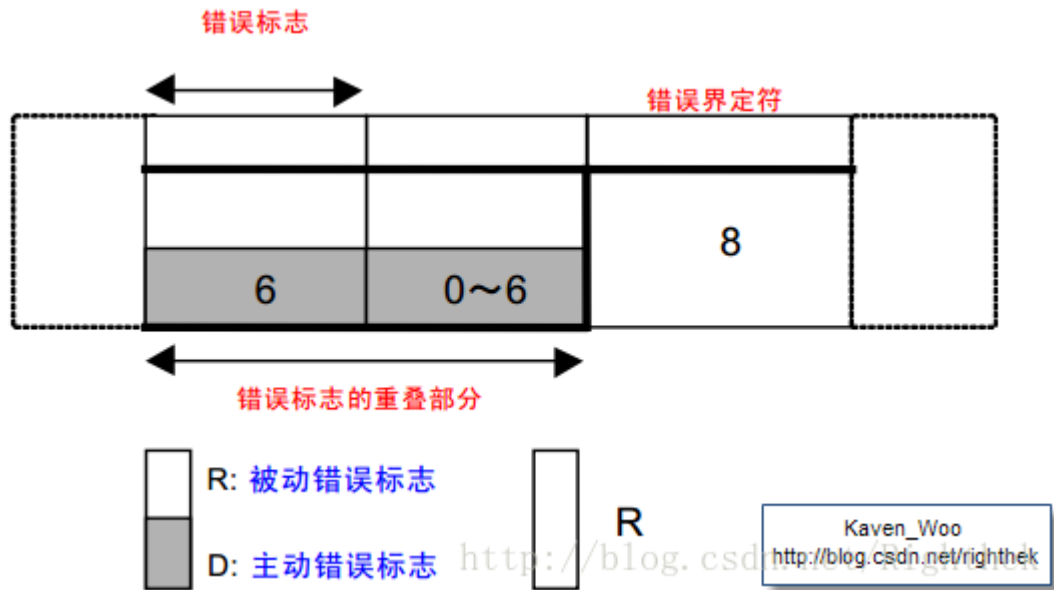


图4 错误帧格式

过载帧是用于接收单元通知其尚未完成接收准备的帧。过载帧由过载标志和过载界定符构成。过载帧格式如下表示：

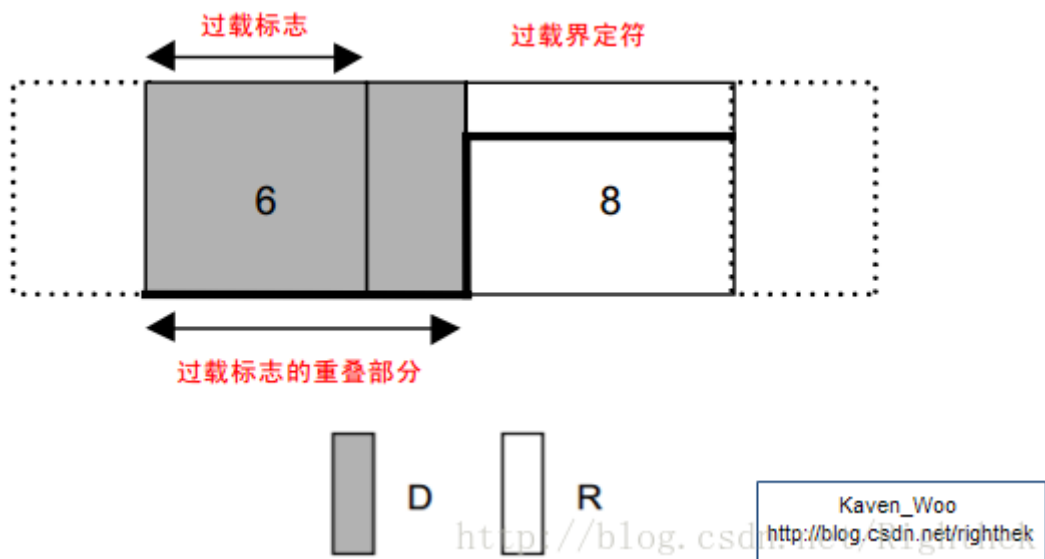


图5 过载帧格式



帧间隔是用于分隔数据帧和远程帧的帧。数据帧和远程帧可通过插入帧间隔将本帧与前面的任何帧（数据帧、远程帧、错误帧、过载帧）分开。过载帧和错误帧前不能插入帧间隔。帧间隔如下图所示：

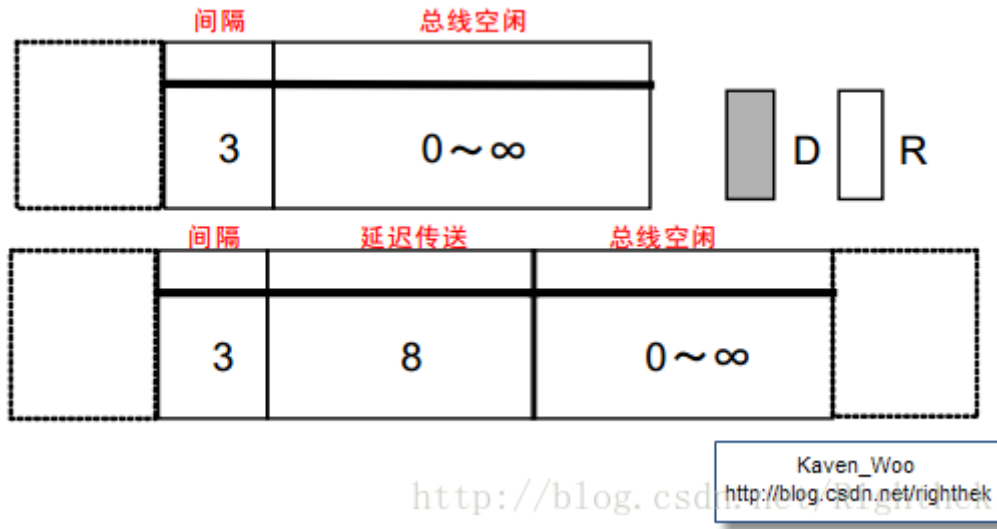


图 6 帧间隔格式

## 2、CAN 总线的仲裁方式

在总线空闲态，最先开始发送消息的单元获得发送权。多个单元同时开始发送时，各发送单元从仲裁段的第一位开始进行仲裁。连续输出显性电平最多的单元可继续发送。即逐位地对比各个结点发出的报文 ID。由于线与的关系，显示位“0”可以覆盖隐性位“1”，因此 ID 最小的节点赢得仲裁，总线上表现为该结点的报文，其他结点失去仲裁，退出发送，转为接收状态。

标准格式 ID 与具有相同 ID 的远程帧或者扩展格式的数据帧在总线上竞争时，标准格式的 RTR 位为显性位的具有优先权，可继续发送。

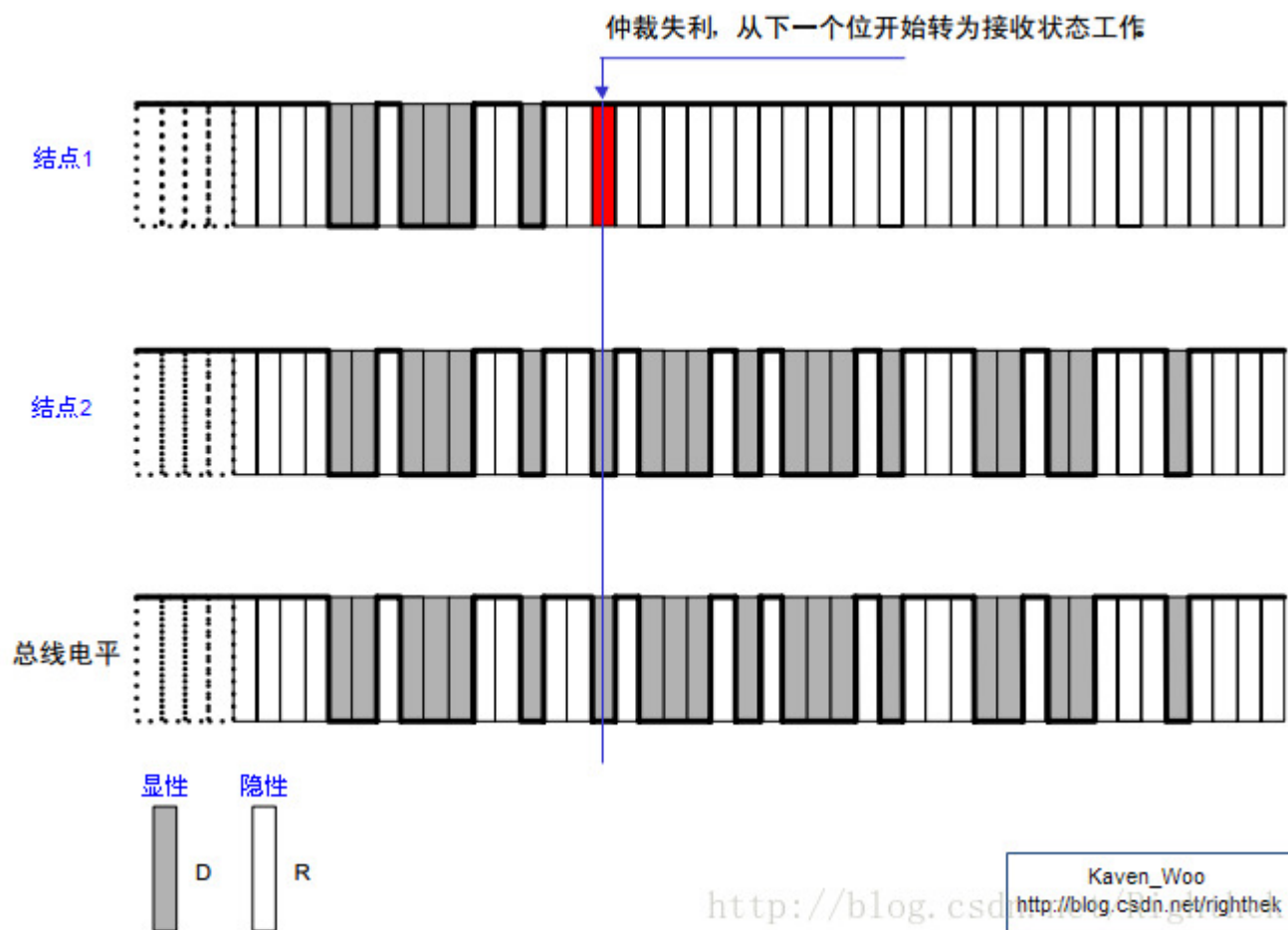


图 7 仲裁方式

### 3、位填充 ( BitStuffing )

位填充是为了防止突发错误而设定的功能。位填充的规则如下：

- ( 1 ) 5 位连续相同电平之后，必须填充一位反向位，即不允许有 6 个连续相同位；
- ( 2 ) SOF 之前为总线空闲状态，不需要同步，因此不需要位填充；
- ( 3 ) CRC 之后为固定格式，不允许填充；

(4) 由 CAN 控制器自动实现；

#### 4、CAN 的错误处理

CAN 控制器检测错误共有以下 5 种：

(1) 位填充错误；

在使用位填充的帧场内，结点如果检测到 6 个连续相同的位值，则产生位填充错误，在下一位开始时，该结点将发送一个错误帧。

(2) 位错误；

在发送期间，结点检测到总线的位值与自身发送的位值不一致时，则产生位错误，在下一位开始时，该结点将发送一个错误帧。

(3) CRC 错误；

接收结点计算的 CRC 码与数据帧本身自带的 CRC 码不一致，接收结点将丢弃该帧，并在 ACK 界定符之后发送一个错误帧。

(4) 应答错误；

发送结点在 ACK Slot 位会发送隐性位，同时监听总线是否为显性位，如果是显性位，则表明至少一个节点正确收到该帧；如果是隐性位，将产生 ACK 错误，发送结点发送一个错误帧。

(5) 格式错误；

发送结点在 (CRC 界定符、ACK 界定符、帧结束 EOF) 固定格式的位置检测到显性位时，将发生格式错误，并发送一个错误帧。

#### 5、CAN 总线同步

CAN 总线的通信方式为 NRZ 方式。各个位的开关或者结尾都没有附加同步信号。发送结点以与位时序同步的方式开始发送数据。另外，接收结点根据总线上电平的变化进行同步并进行接收工作。

但是，发送结点和接收结点存在的时钟频率误差及传输路径上的（电缆、驱动器等）相位延迟会引进同步偏差。因此接收结点需要通过同步的方式调整时序进行接收。

同步的作用是尽量使本地位时序与总结信号的位时序一致（本地同步段与总结信号边沿同步）。只有接收结点需要同步；同步只会发生在隐性到显性电平的跳沿。

同步的方式为硬件同步和再同步。