CAN－bus 规范 V2.0 版本

引言

随着串行通讯进入更多应用领域，因此，在一些应用里，需要对通讯功能的报文识别位提出分配标准化的要求。原先的地址范围由 11 个识别位定义，如果地址范围扩大，则这些应用就可以更好地由 CAN 来实现。

因此引入了第二种报文格式（‘扩展格式’）的概念，其定义的地址范围更宽，由 29 位定义。系统设计者将从考虑定义良好的结构命名方案中得到解放。有的用户不需要由扩展格式提供的识别符范围，可以继续沿用常规的 11 位识别符范围（‘标准格式’），在这种情况下，可以采用市场上可用的 CAN 仪器，或使用兼容这两种模式的新控制器类仪器。

为了区别标准格式和扩展格式，按 CAN 1.2 规范定义，使用了 CAN 报文格式的第一个保留位。因为 CAN1.2 定义的信息格式相当于标准格式，因此仍然是有效的。此外，由于扩展格式已经定义，因此网络中会共存标准格式和扩展格式的报文。

这本 CAN 规范技术规范由两部分组成：

• A 部分：CAN 的报文格式说明（按 CAN1.2 规范定义）。

• B 部分：标准格式和扩展格式的说明。

为了兼容 CAN2.0，要求 CAN 的仪器应兼容 A 部分或 B 部分。

注意：

只要没有用到扩展格式，那么，根据 A 部分或 CAN 旧版本设计的仪器可以和根据 B 部分设计的仪器相互间进行通讯。

A 部分

目录

1．介绍

控制器局域网（CAN）为串行通讯协议，能有效地支持具有很高安全等级的分布实时控制。CAN 的应用范围很广，从高速的网络到低价位的多路接线都可以使用 CAN。在汽车电子行业里，使用 CAN 连接发动机控制单元、传感器、防刹车系统、等等，其传输速度可达 1 Mbit/s。同时，可以将 CAN 安装在卡车本体的电子控制系统里，诸如车灯组、电气车窗等等，用以代替接线配线装置。

这本技术规范的目的是为了在任何两个 CAN 仪器之间建立兼容性。可是，兼容性有不同的方面，比如电气特性和数据转换的解释。为了达到设计透明度以及实现柔韧性，CAN 被细分为以下不同的层次：

• CAN 对象层（the object layer）

• CAN 传输层（the transfer layer）

• 物理层（the phyical 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：

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

在这里，定义对象处理较为灵活。传输层的作用主要是传送规则，也就是控制帧结构、执行仲裁、错误检测、出错标定、故障界定。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。位定时的一些普通功能也可以看作是传输层的一部分。理所当然，传输层的修改是受到限制的。

物理层的作用是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。

这本技术规范的目的是定义传输层，并定义 CAN 协议于周围各层当中所发挥的作用（所具有的意义）。

2．基本概念

CAN 具有以下的属性：

• 报文的优先权

• 保证延迟时间

• 设置灵活

• 时间同步的多点接收

• 系统宽数据的连贯性

• 多主机

• 错误检测和标定

• 只要总线一处于空闲，就自动将破坏的报文重新传输

• 将节点的暂时性错误和永久性错误区分开来，并且可以自动关闭错误的节点

CAN 节点的层结构（Layered Structure od a CAN node）

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

- 物理层定义实际信号的传输方法。本技术规范没有定义物理层，以便允许根据它们的应用，对发送媒体和信号电平进行优化。
- 传输层是 CAN 协议的核心。它把接收到的报文提供给对象层，以及接收来自对象层的报文。传输层负责位定时及同步、报文分帧、仲裁、应答、错误检测和标定、故障界定。
- 对象层的功能是报文滤波以及状态和报文的处理。

这本技术规范的目的是为了定义传输层及定义 CAN 协议在周围各层中所发挥的作用(所具有的意义)。

报文（Messages）

总线上的信息以不同的固定报文格式发送，但长度受限（见第 3 节的报文传输）。当总线空闲时任何连接的单元都可以开始发送新的报文。

信息路由（Information Routing）

在 CAN 系统里，节点不使用任何关于系统配置的信息（比如，站地址）。以下是几个重要的概念。

- 系统灵活性：不需要改变任何节点的应用层及相关的软件或硬件，就可以在 CAN 网络中直接添加节点。
- 报文路由：报文的内容由识别符命名。识别符不指出报文的目的地，但解释数据的含义。因此，网络上所有的节点可以通过报文滤波确定是否应对该数据做出反应。
- 多播：由于引入了报文滤波的概念，任何数目的节点都可以接收报文，并同时对此报文做出反应。
- 数据连贯性：在 CAN 网络内，可以确保报文同时被所有的节点接收（或同时不被接收）。因此，系统的数据连贯性是通过多播和错误处理的原理实现的。

位速率（Bit rate）：

不同的系统，CAN 的速度不同。可是，在一给定的系统里，位速率是唯一的，并且是固定的。

优先权（Priorities）：

在总线访问期间，识别符定义一静态的报文优先权。

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。数据帧和相应的远程帧是由相同的识别符（IDENTIFIER）命名的。

多主机（Multimaster）：

总线空闲时，任何单元都可以开始传送报文。具有较高优先权报文的单元可以获得总线访问权。

仲裁（Arbitration）：

只要总线空闲，任何单元都可以开始发送报文。如果 2 个或 2 个以上的单元同时开始传送报文，那么就会有总线访问冲突。通过使用识别符的位形式仲裁可以解决这个冲突。仲裁的机制确保信息和时间均不会损失。当具有相同识别符的数据帧和远程帧同时初始化时，数据帧优先于远程帧。仲裁期间，每一个发送器都对发送位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”电平而监控视到一“显性”电平（见总线值），那么该单元就失去了仲裁，必须退出发送状态。

安全性（Safety）：

为了获得最安全的数据发送，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。

错误检测（Error Detection）：

为了检测错误，必须采取以下措施：

- 监视（发送器对发送位的电平与被监控的总线电平进行比较）
- 循环冗余检查
- 位填充
- 报文格式检查

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

错误检测的机制要具有以下的属性：

- 检测到所有的全局错误
- 检测到发送器所有的局部错误
- 可以检测到一报文里多达 5 个任意分布的错误
- 检测到一报文里长度低于 15（位）的突发性错误
- 检测到一报文里任一奇数个的错误

对于没有被检测到的错误报文，其残余的错误可能性概率低于：报文错误率* $4.7 * 10^{-11}$。

错误标定和恢复时间（Error Sinalling and Recovery Time）：

任何检测到错误的节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。如果不再出现新错误的话，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。

故障界定（Fault Confinement）：

CAN 节点能够把永久故障和短暂扰动区分开来。永久故障的节点会被关闭。

连接（Connections）：

CAN 串行通讯链路是可以连接许多单元的总线。理论上，可连接无数多的单元。但由于实际上受延迟时间以及/或者总线线路上电气负载的影响，连接单元的数量是有限的。

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。通过此通道可以获得数据的再同步信息。要使此通道实现通讯，有许多的方法可以采用，如使用单芯线（加上接地）、2 条差分线、光缆等等。这本技术规范不限制这些实现方法的使用，即未定义物理层。

总线值（Bus value）：

总线可以具有两种互补的逻辑值之一：“显性”或“隐性”。 “显性”位和“隐性”位同时传送时，总线的结果值为“显性”。比如，在执行总线的“线与”时，逻辑 0 代表“显性”等级，逻辑 1 代表“隐性”等级。本技术规范不给出表示这些逻辑电平的物理状态（比如，电压、光）。

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。

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。唤醒时，虽然传输层要等待一段时间使系统振荡器稳定，然后还要等待一段时间直到与总线活动同步（通过检查 11 个连续的“隐性”的位），但在总线驱动器被重新设置为“总线在线”之前，内部运行已重新开始。为了唤醒系统上正处于睡眠模式的其他节点，可以使用一特殊的唤醒报文，此报文具有专门的、最低等级的识别符。（rrr rrrd rrrr; r = ‘’隐性’ d = ‘显性’）

3．报文传输

3.1 帧类型

报文传输由以下 4 个不同的帧类型所表示和控制：

- 数据帧：数据帧携带数据从发送器至接收器。
- 远程帧：总线单元发出远程帧，请求发送具有同一识别符的数据帧。
- 错误帧：任何单元检测到一总线错误就发出错误帧。
- 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。

数据帧（或远程帧）通过帧间空间与前述的各帧分开。

3.1.1 数据帧

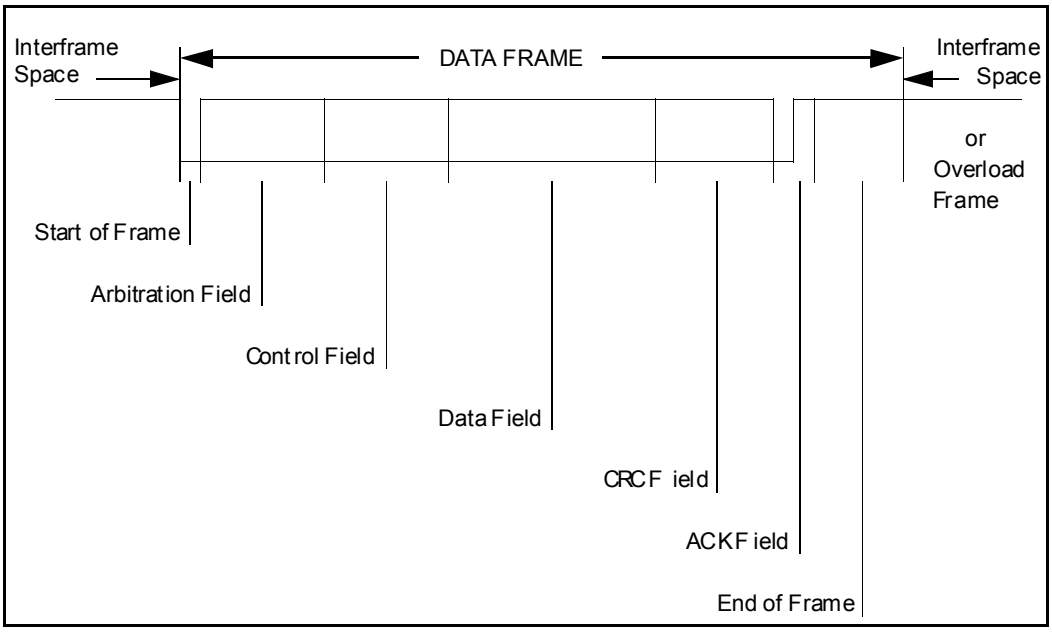
数据帧由 7 个不同的位场组成：

帧起始、仲裁场、控制场、数据场、CRC 场、应答场、帧结尾。数据场的长度可以为 0。

帧起始

它标志数据帧和远程帧的起始，由一个单独的“显性”位组成。

只在总线空闲（参见“总线空闲”）时，才允许站开始发送（信号）。所有的站必须同步于首先开始发送信息的站的帧起始前沿（参见“硬同步”）。

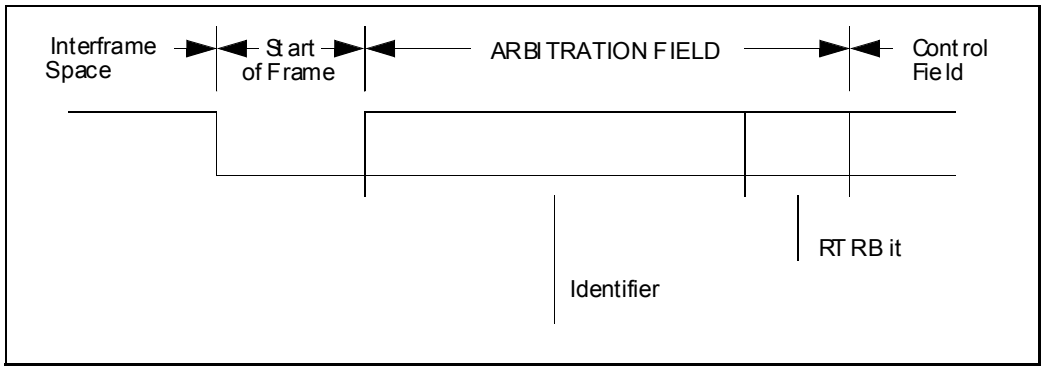


仲裁场

仲裁场包括识别符和远程发送请求位（RTR）。

识别符：识别符的长度为 11 位。这些位的发送顺序是从 ID-10 到 ID-0。最低位是 ID-0。最高的 7 位（ID-10 到 ID-4）必须不能全是“隐性”。

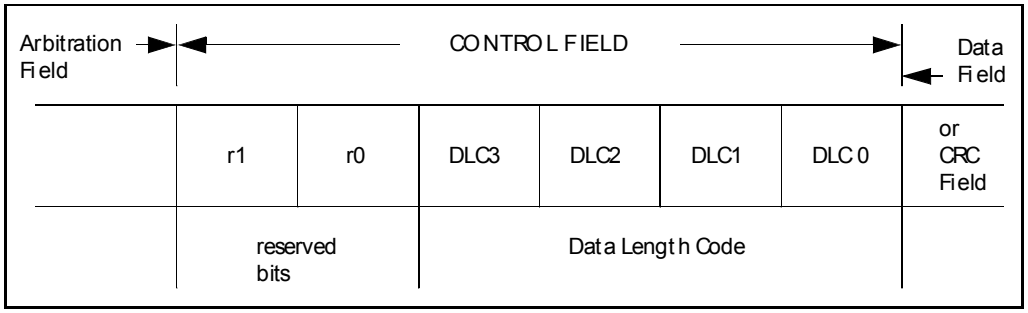
RTR 位：该位在数据帧里必须为“显性”，而在远程帧里必须为“隐性”。



控制场

控制场由 6 个位组成，包括数据长度代码和两个将来作为扩展用的保留位。所发送的保留位必须为“显性”。接收器接收所有由“显性”和“隐性”组合在一起的位。

数据长度代码：数据长度代码指示了数据场中字节数量。数据长度代码为 4 个位，在控制场里被发送。



数据长度代码中数据字节数的编码（DATA LENGTH CODE）：

缩写： d—“显性”

r—“隐性”

Number of Data Bytes	Data Length Code			
	DLC3	DLC2	DLC1	DLC0
0	d	d	d	d
1	d	d	d	r
2	d	d	r	d
3	d	d	r	r
4	d	r	d	d
5	d	r	d	r
6	d	r	r	d
7	d	r	r	r
8	r	d	d	d

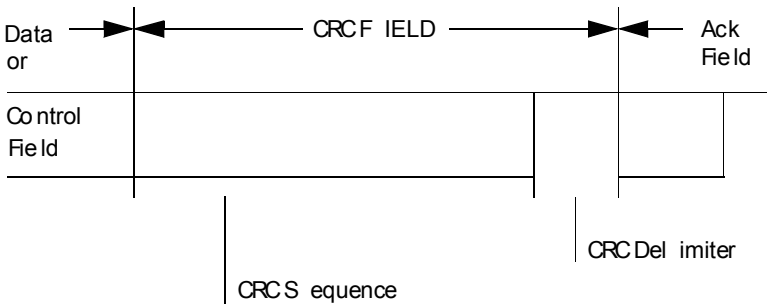
数据帧：允许的数据字节数：{0,1,....,7,8}。其他的数值不允许使用。

数据场

数据场由数据帧中的发送数据组成。它可以为 0～8 个字节，每字节包含了 8 个位，首先发送 MSB。

CRC 场

CRC 场包括 CRC 序列（CRC SEQUENCE），其后是 CRC 界定符（CRC DELIMITER）。



CRC 序列：由循环冗余码求得的帧检查序列最适用于位数低于 127 位〈BCH 码〉的帧。为进行 CRC 计算，被除的多项式系数由无填充位流给定，组成这些位流的成分是：帧起始、仲裁场、控制场、数据场（假如有），而 15 个最低位的系数是 0。将此多项式被下面的多项式发生器除（其系数以 2 为模）：

$$X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1$$

这个多项式除法的余数就是发送到总线上的 CRC 序列（CRC SEQUENCE）。为了实现这个功能，可以使用 15 位的位移寄存器 CRC_RG（14:0）。如果用 NXTBIT 标记指示位流的下一位，它由从帧的起始到数据场末尾都由无填充的位序列给定。

CRC 序列（CRC SEQUENCE）的计算如下：

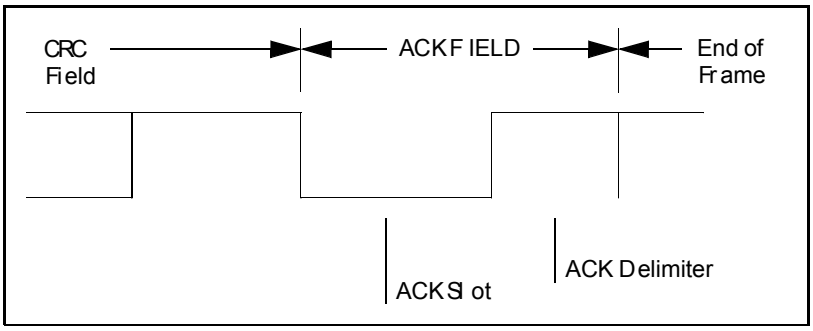
```
CRC_RG = 0;                                    // 初始化移位寄存器
REPEAT;
```

```
        CRCNXT = NXTBIT EXOR CRC_RG（14）;
        CRC_RG（14:1） = CRC_RG（13:0）;              // 寄存器左移 1 位
        CRC_RG（0） = 0;
        IF CRCNXT THEN
        CRC_RG（14:0） = CRC_RG（14:0） EXOR （4599hex）;
        ENDIF
UNTIL （CRC 序列开始或存在一个错误条件）
```

在传送/接收数据场的最后一位以后，CRC_RG 包含有 CRC 序列。CRC 序列之后是 **CRC 界定符**，它包含一个单独的“隐性”位 。

应答场

应答场长度为 2 个位，包含应答间隙（ACK SLOT）和应答界定符（ACK DELIMITER）。在应答场里，发送站发送两个“隐性”位。当接收器正确地接收到有效的报文，接收器就会在应答间隙（ACK SLOT）期间（发送 ACK 信号）向发送器发送一“显性”的位以示应答。



应答间隙：所有接收到匹配 CRC 序列（CRC SEQUENCE）的站会在应答间隙（ACK SLOT）期间用一“显性”的位写入发送器的“隐性”位来作出回答。

ACK 界定符：ACK 界定符是 ACK 场的第二个位，并且是一个必须为“隐性”的位。因此，应答间隙（ACK SLOT）被两个“隐性”的位所包围，也就是 CRC 界定符（CRC DELIMITER）和 ACK 界定符（ACK DELIMITER）。

帧结尾

每一个数据帧和远程帧均由一标志序列界定。这个标志序列由 7 个“隐性”位组成。

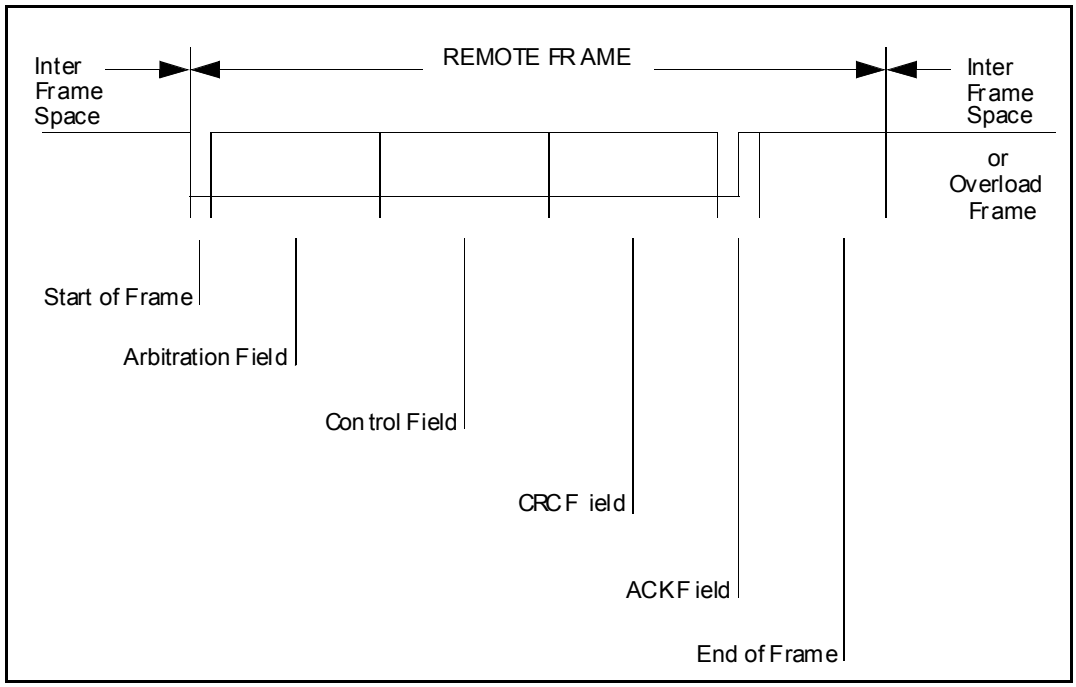
3.1.2 远程帧

通过发送远程帧，作为某数据接收器的站通过其资源节点对不同的数据传送进行初始化设置。

远程帧由 6 个不同的位场组成：

帧起始、仲裁场、控制场、CRC 场、应答场、帧末尾。

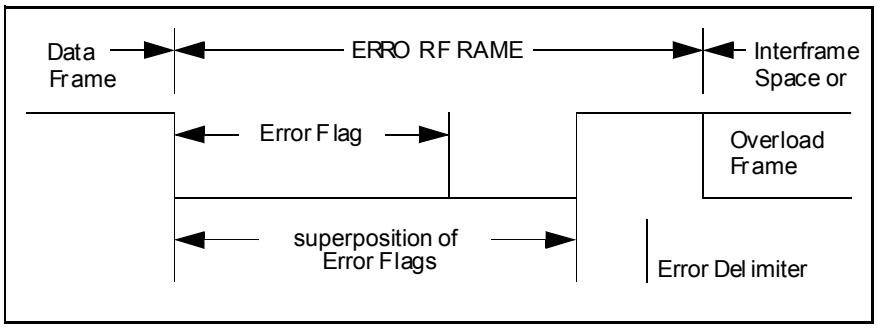
与数据帧相反，远程帧的 RTR 位是“隐性”的。它没有数据场，数据长度代码的数值是不受制约的（可以标注为容许范围里 0...8 的任何数值）。此数值是相应于数据帧的数据长度代码。



RTR 位的极性表示了所发送的帧是一数据帧（RTR 位“显性”）还是一远程帧（RTR“隐性”）。

3.1.3 错误帧

错误帧由两个不同的场组成。第一个场用作为不同站提供的错误标志（ERROR FLAG）的叠加。第二个场是错误界定符。



为了能正确地终止错误帧，一“错误被动”的节点要求总线至少有长度为 3 个位时间的总线空闲（如果“错误被动”的接收器有本地错误的话）。因此，总线的载荷不应为 100%。

有两种形式的错误标志，主动错误标志（Active error flag）和被动错误标志（Passive error flag）。主动错误标志由 6 个连续的“显性”位组成。被动错误标志由 6 个连续的“隐性”的位组成，除非被其他节点的“显性”位重写。

检测到错误条件的“错误主动”的站通过发送主动错误标志，以指示错误。错误标志的形式破坏了从帧起始到 CRC 界定符的位填充规则（参见“编码”），或者破坏了应答场或帧末尾场的固定形式。所有其他的站由此检测到错误条件并与此同时开始发送错误标志。因此，“显性”位（此“显性”位可以在总线上监视）的序列导致一个结果，这个结果就是把各个单独站发送的不同的错误标志叠加在一起。这个顺序的总长度最小为 6 个位，最大为 12 个位。

检测到错误条件的“错误被动”的站试图通过发送被动错误标志，以指示错误。“错误被动”的站等待 6 个相同极性的连续位（这 6 个位处于被动错误标志的开始）。当这 6 个相同的位被检测到时，被动错误标志的发送就完成了。

错误界定符包括 8 个“隐性”的位。

错误标志传送了以后，每一站就发送“隐性”的位并一直监视总线直到检测出一个“隐性”的位为止。然后就开始发送 7 位以上的“隐性”位。

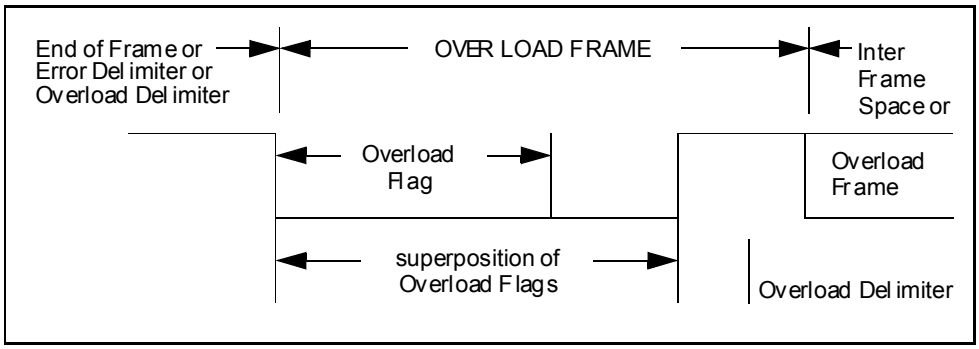
3.1.4 过载帧

过载帧包括两个位场：过载标志和过载界定符。

有两种过载条件都会导致过载标志的传送：

1. 接收器的内部条件（此接收器对于下一数据帧或远程帧需要有一延时）。
2. 间歇场期间检测到一“显性”位。

由过载条件 1 而引发的过载帧只允许起始于所期望的间歇场的第一个位时间开始。而由过载条件 2 引发的过载帧应起始于所检测到“显性”位之后的位。



通常为了延时下一个数据帧或远程帧，两个过载帧都会产生。

过载标志

过载标志由 6 个“显性”的位组成。过载标志的所有形式和主动错误标志的一样。

过载标志的形式破坏了间歇场的固定形式。因此，所有其他的站都检测到一过载条件并与此同时发出过载标志。（万一有的节点在间歇的第 3 个位期间于本地检测到“显性”位，则其他的节点将不能正确地解释过载标志，而是将这 6 个“显性”位中的第一个位解释为帧的起始。这第 6 个“显性”的位破坏了产生错误条件的位填充的规则。）

过载界定符

过载界定符包括 8 个“隐性”的位。

过载界定符的形式和错误界定符的形式一样。过载标志被传送后，站就一直监视总线直到检测到一个从“显性”位到“隐性”位的发送（过渡形式）。此时，总线上的每一个站完成了过载标志的发送，并开始同时发送 7 个以上的“隐性”位。

3.1.5 帧间空间（INTERFRAME SPACING）

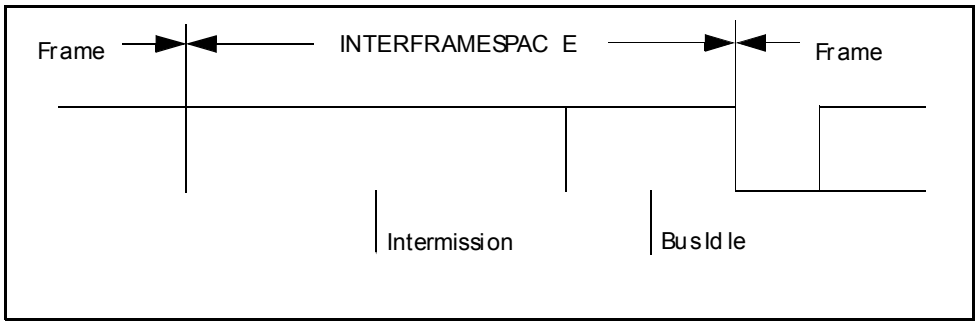
数据帧（或远程帧）与其前面帧的隔离是通过帧间空间实现的，无论其前面的帧为何类型（数据帧、远程帧、错误帧、过载帧）。所不同的是，过载帧与错误帧之前没有帧间空间，多个过载帧之间也不是由帧间空间隔离的。

帧间空间

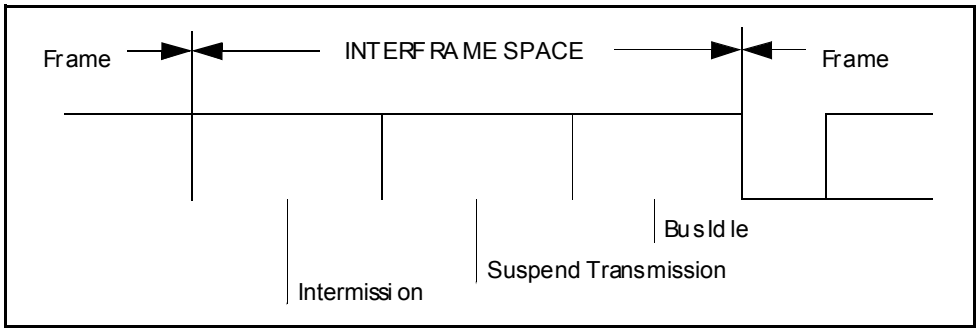
帧间空间包括间歇场、总线空闲的位场。如果“错误被动”的站已作为前一报文的发送器时，则其帧空间除了间歇、总线空闲外，还包括称作挂起传送（SUSPEND TRANSMISSION）的位场。

对于不是“错误被动” 的站，或者此站已作为前一报文的接收器，其帧间空间如下图所示：

:



对于已作为前一报文发送器的“错误被动”的站，其帧间空间如下图所示：



间歇

间歇包括 3 个“隐性”的位。

间歇期间，所有的站均不允许传送数据帧或远程帧，唯一要做的是标示一个过载条件。

总线空闲

总线空闲的（时间）长度是任意的。只要总线被认定为空闲，任何等待发送信息的站就会访问总线。在发送其他信息期间，有报文被挂起，对于这样的报文，其传送起始于间歇之后的第一个位。

总线上检测到的“显性”的位可被解释为帧的起始。

挂起传送

“错误被动”的站发送报文后，站就在下一报文开始传送之前或总线空闲之前发出 8 个“隐性”的位跟随在间歇的后面。如果与此同时另一站开始发送报文（由另一站引起），则此站就作为这个报文的接收器。

3.2 发送器/接收器的定义

发送器（TRANSMITTER）

产生报文的单元被称之为报文的“发送器”。此单元保持作为报文发送器直到总线出现空闲或此单元失去仲裁（ARBITRATION）为止。

接收器（RECEIVER）

如果有一单元不作为报文的发送器并且总线也不空闲，则这一单元就被称之为报文的“接收器”。

4．报文校验

校验报文是否有效的时间点，发送器与接收器各不相同。

发送器：

如果直到帧的末尾位均没有错误，则此报文对于发送器有效。如果报文破损，则报文会根据优先权自动重发。为了能够和其他信息竞争总线，重新传输必须在总线空闲时启动。

接收器：

如果直到一最后的位（除了帧末尾位）均没有错误，则报文对于接收器有效。

5．编码

位流编码：

帧的部分，诸如帧起始、仲裁场、控制场、数据场以及 CRC 序列，均通过位填充的方法编码。无论何时，发送器只要检测到位流里有 5 个连续识别值的位，便自动在位流里插入一补码位。

数据帧或远程帧（CRC 界定符、应答场和帧末尾）的剩余位场形式相同，不填充。错误帧和过载帧的形式也相同，但并不通过位填充的方法进行编码。

其报文里的位流根据“不返回到零”（NRZ）之方法来编码。这就是说，在整个位时间里，位电平要么为“显性”，要么为“隐性”。

6．错误处理

6.1 错误检测

有以下 5 种不同的错误类型（这 5 种错误不会相互排斥）

• 位错误

站单元在发送位的同时也对总线进行监视。如果所发送的位值与所监视的位值不相符合，则在此位时间里检测到一个位错误（BIT ERROR）。但是在仲裁场（ARBITRATION FIELD）的填充位流期间或 ACK 间隙（ACK SLOT）发送一“隐性”位的情况是例外的—— 此时，当监视到一“显性”位时，不会发出位错误（BIT ERROR）。当发送器发送一个被动错误标志但检测到“显性”位时，也不视为位错误。

• 填充错误

如果在使用位填充法进行编码的信息中，出现了第 6 个连续相同的位电平时，将检测到一个填充错误。

• CRC 错误

CRC 序列包括发送器的 CRC 计算结果。接收器计算 CRC 的方法与发送器相同。如果计算结果与接收到 CRC 序列的结果不相符，则检测到一个 CRC 错误（CRC ERROR）。

• 形式错误

当一个固定形式的位场含有 1 个或多个非法位，则检测到一个形式错误（FORM ERROR）。

• 应答错误

只要在 ACK 间隙（ACK SLOT）期间所监视的位不为“显性”，则发送器会检测到一个应答错误（ACKNOWLEDGMENT ERROR）。

6.2 错误标定

检测到错误条件的站通过发送错误标志指示错误。对于“错误主动”的节点，错误信息为“主动错误标志”，对于“错误被动”的节点，错误信息为“被动错误标志”。站检测到无论是位错误、填充错误、形式错误，还是应答错误，这个站会在下一位时发出错误标志信息。

只要检测到的错误的条件是 CRC 错误，错误标志的发送开始于 ACK 界定符之后的位（其他的错误条件除外）。

7．故障界定

至于故障界定，单元的状态可能为以下三种之一：

- “错误主动”

- “错误被动”
- “总线关闭”

“错误主动”的单元可以正常地参与总线通讯并在错误被检测到时发出主动错误标志。

“错误被动”的单元不允许发送主动错误标志。“错误被动”的单元参与总线通讯而且在错误被检测到时只发出被动错误标志。而且，发送以后，“错误被动”单元将在预设下一个发送之前处于等待状态。（见“挂起发送”）

“总线关闭”的单元不允许在总线上有任何的影响（比如，关闭输出驱动器）。

在每一总线单元里实现两种计数以便故障界定：

- 发送错误计数
- 接收错误计数

这些计数按以下规则改变（注意：在给定的报文发送期间，可能要用到的规则不只一个）：

1. 当接收器检测到一个错误，接收错误计数就加 1。在发送主动错误标志或过载标志期间所检测到的错误为位错误时，接收错误计数器值不加 1。

2. 当错误标志发送以后,接收器检测到的第一个位为“显性”时，接收错误计数值加 8。

3. 当发送器发送一错误标志时，发送错误计数器值加 8。

例外情况 1：

发送器为“错误被动”，并检测到一应答错误（注：此应答错误由检测不到一“显性”应答 以及当发送被动错误标志时检测不到一“显性”位而引起）。

例外情况 2：

发送器因为填充错误而发送错误标志（注：此填充错误发生于仲裁期间。引起填充错误是由于：填充位〈填充位〉位于 RTR 位之前，并已作为“隐性”发送，但是却被监视为“显性”）。

例外情况 1 和例外情况 2 时，发送错误计数器值不改变。

4. 发送主动错误标志或过载标志时，如果发送器检测到位错误，则发送错误计数器值加 8。

5. 当发送主动错误标志或过载标志时，如果接受器检测到位错误（位错误），则接收错误计数器值加 8。

6. 在发送主动错误标志、被动错误标志或过载标志以后，任何节点最多容许 7 个连续的 “显性”位。以下的情况，每一发送器将它们的发送错误计数值加 8，及每一接收器的接收错误计数值加 8：

当检测到第 14 个连续的“显性”位后；

在检测到第 8 个跟随着被动错误标志的连续的“显性”位以后；

在每一附加的 8 个连续“显性”位顺序之后。

7. 报文成功传送后（得到应答及直到帧末尾结束没有错误），发送错误计数器值减 1，除非已经是 0。

8. 如果接收错误计数值介于 1 和 127 之间，在成功地接收到报文后（直到 ACK 间隙接收没有错误，及成功地发送了应答位），接收错误计数器值减 1。如果接收错误计数器值是 0，则它保持 0，如果大于 127，则它会设一值介于 119 到 127 之间。

9. 当发送错误计数器值等于或超过 128 时，或当接收错误计数器值等于或超过 128 时，节点为“错误被动”。让节点成为“错误被动”的错误条件致使节点发出主动错误标志。

10.当发送错误计数器值大于或等于 256 时，节点为“总线关闭”。

11. 当发送错误计数器值和接收错误计数器值都小于或等于 127 时，“错误被动”的节点重新变为“错误主动”。

12. 在总线监视到 128 次出现 11 个连续“隐性”位之后，“总线关闭”的节点可以变成“错误主动”（不再是“总线关闭”），它的错误计数值也被设置为 0。

备注：一个大约大于 96 的错误计数值显示总线被严重干扰。最好能够采取措施测试这个条件。

备注：起动/睡眠：如果起动期间内只有 1 个节点在线，以及如果这个节点发送一些报文，则将不会有应答，如此检测到错误并重复报文。由于此原因，节点会变为“错误被动”，而不是“总线关闭”。

8．位定时要求

标称位速率

标称位率为一理想的发送器在没有重新同步的情况下每秒发送的位数量。

标称位时间

标称位时间 = 1 /标称位速率

可以把标称位时间划分成了几个不重叠时间的片段，它们是：

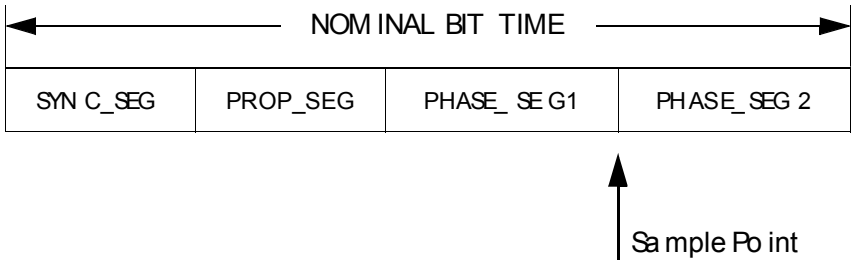
同步段（SYNC_SEG）

传播时间段（PROP_SEG）

相位缓冲段 1（PHASE_SEG1）

相位缓冲段 2（PHASE_SEG2）

位时间如下图所示：



同步段（SYNC SEG）

位时间的同步段用于同步总线上不同的节点。这一段内要有一个跳变沿。

传播段（PROP SEG）

传播段用于补偿网络内的物理延时时间。

它是总线上输入比较器延时和输出驱动器延时总和的两倍。

相位缓冲段 1、相位缓冲段 2（PHASE SEG1、PHASE SEG2）：

相位缓冲段用于补偿边沿阶段的错误。这两个段可以通过重新同步加长或缩短。

采样点（SAMPLE POINT）：

采样点是读总线电平并解释各位的值的一个时间点。采集点位于相位缓冲段 1（PHASE_SEG1）之后。

信息处理时间（INFORMATION PROCESS TIME）

信息处理时间是一个以采样点作为起始的时间段。采集点用于计算后续位的位电平。

时间份额（TIME QUANTUM）

时间份额是派生于振荡器周期的固定时间单元。存在有一个可编程的预比例因子，其整体数值范围为 1—32 的整数，以最小时间份额为起点，时间份额的长度为：

时间份额（TIME QUANTUM）＝ m * 最小时间份额（MINIMUM TIME QUANTUM）

（m 为预比例因子）

时间段的长度（Length of Time Segments）

同步段（SYNC_SEG）为 1 个时间份额； 传播段（PROP_SEG）的长度可设置为 1，2，…，8 个时间份额；缓冲段 1 （PHASE_SEG1）的长度可设置为 1，2，…，8 个时间份额；相位缓冲段 2（PHASE_SEG2）的长度为阶段缓冲段 1（ PHASE_SEG1）和信息处理时间（INFORMATION PROCESSING TIME）之间的最大值； 信息处理时间少于或等于 2 个时间份额。

一个位时间总的的时间份额值可以设置在 8—25 的范围。

同步（SYHCHRONIZATION）

硬同步（HARD SYHCHRONIZATION）：

硬同步后，内部的位时间从同步段重新开始。因此，硬同步强迫由于硬同步引起的沿处于重新开始的位时间同步段之内。

重新同步跳转宽度（RESYHCHRONIZATION JUMP WIDTH）

重新同步的结果，使相位缓冲段 1 增长，或使相位缓冲段 2 缩短。相位缓冲段加长或缩短的数量有一个上限，此上限由重新同步跳转宽度给定。重新同步跳转宽度应设置于 1 和最小值之间（此最小值为 4，PHASE_SEG1）。

时钟信息可以从一位值转换到另一位值的跳变中得到。后续位有固定的最大数值，其数值相同。这个属性提供了总线单元在帧期间重新和位流同步的可能性。（这里有一个属性，即：只有后续位的一固定最大值才具有相同的数值。这个属性使总线单元在帧期间重新同步于位流成为可能。可用于重新同步的两个过渡过程之间的最大长度为29个位时间。）

一个沿的相位误差

一个沿的相位误差由相关于同步段的沿的位置给出，以时间份额量度。相位误差定义如下：

- e = 0 如果沿处于同步段里（SYNC_SEG）.
- e > 0 如果沿位于采集点（SAMPLE POINT）之前.
- e < 0 如果沿处于前一个位的采集点（SAMPLE POINT）之后.

重新同步

当引起重新同步沿的相位误差的幅值小于或等于重新同步跳转宽度的设定值时，重新同步和硬件同步的作用相同。当相位错误的量级大于重新同步跳转宽度时：

- 如果相位误差为正，则相位缓冲段 1 被增长。增长的范围为与重新同步跳转宽度相等的值。
- 如果相位误差为负，则相位缓冲段 2 被缩短。缩短的范围为与重新同步跳转宽度相等的值。

同步的原则

硬同步和重新同步都是同步的两种形式，遵循以下规则：

1. 在一个位时间里只允许一个同步。
2. 仅当采集点之前探测到的值与紧跟沿之后的总线值不相符合时，才把沿用作于同步。
3. 总线空闲期间，有一“隐性”转变到“显性”的沿，无论何时，硬同步都会被执行。
4. 如果仅仅是将“隐性”转化为“显性”的沿用作于重新同步使用，则其他符合规则 1 和规则 2 的所有从“隐性”转化为“显性”的沿可以用作为重新同步。有一例外情况，即，当发送一显性位的节点不执行重新同步而导致一“隐性”转化为“显性” 沿，此沿具有正的相位误差，不能作为重新同步使用。

9 增加 CAN 振荡器容差

这章介绍 CAN 协议的向上兼容的修改，就如同第 1 到 8 章介绍的那样。

9.1 协议修改

为了把振荡器最大容差从目前的 0.5%增加到 1.5%，有必要作以下修改以便向上兼容现有的 CAN 规范：

1） 如果 CAN 节点在间歇的第三位采集到一显性位，则此位被解释为帧的起始位。

2） 如果 CAN 节点有一信息等待发送并且节点在间歇的第三位采集到一显性位，则此位被解释为帧的起始位，并从下一个位开始发送具有识别符作为首位的报文，而不是首先发送帧的起始位或成为一接收器。

3） 如果节点在错误界定符或过载界定符的第八个位采集到一显性位，则在下一位开始发送一过载帧（而不是错误帧）。错误计数器值不会增加。

4） 仅为隐性转换到显性的沿才会用于重新同步。为符合现有的规范，以下的规定仍然有效。

5） 在硬同步时，所有 CAN 控制器同步于帧起始位。

6） 直到遇上三个隐性的间歇位，CAN 才发送帧起始位。

7） 这个修改允许振荡器最大为 1.58%的容差，并在总线速度达到 125 KB/秒时使用一陶瓷谐振器。为了满足 CAN 协议的整个总线速度范围，仍然需要一晶振。只要符合以下的要求，就可以保持现有协议及增强型协议的兼容性：

8） 同一个网络里的控制器为现有 CAN 协议及增强型 CAN 协议时，所有的控制器必须使用晶振。

9） 具有最高振荡准确度要求的芯片，决定了其他节点的振荡准确度。只有在所有的节点使用增强型的 CAN 协议时才能使用陶瓷谐振器。

B 部分

目录

1. 简介

控制器局域网 CAN 为串行通讯协议，能有效地支持具有很高安全等级的分布实时控制。CAN 的应用范围很广，从高速的网络到低价位的多路接线都可以使用 CAN。在汽车电子行业里，使用 CAN 连接发动机控制单元、传感器、防刹车系统、等等，其传输速度可达 1 Mbit/s。同时，可以将 CAN 安装在卡车本体的电子控制系统里，诸如车灯组、电气车窗等等，用以代替接线配线装置。

这本技术规范的目的是为了在任何两个 CAN 仪器之间建立兼容性。可是，兼容性有不同的方面，比如电气特性和数据转换的解释。为了达到设计透明度以及实现灵活性，根据 ISO/OSI 参考模型，CAN 被细分为以下不同的层次：

- 数据链路层
  - 逻辑链路控制子层 (LLC)
  - 媒体访问控制子层（MAC）
- 物理层

注：在前版本的 CAN 规范中，数据链路层的 LLC 子层和 MAC 子层的服务及功能分别被解释为“对象层”和“传输层”。逻辑链路控制子层 (LLC)的作用范围如下：

- 为远程数据请求以及数据传输提供服务。
- 确定由实际要使用的 LLC 子层接收哪一个报文。
- 为恢复管理和过载通知提供手段。

在这里，定义对象处理较为自由。MAC 子层的作用主要是传送规则，也就是控制帧结构、执行仲裁、错误检测、出错标定、故障界定。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 MAC 子层里确定。位定时的一些普通功能也可以看作是 MAC 子层的一部分。理所当然，MAC 子层的修改是受到限制的。

物理层的作用是在不同节点之间根据所有的电气属性进行位的实际传输。同一网络的物理层对于所有的节点当然是相同的。尽管如此，在选择物理层方面还是很自由的。

这本技术规范的目的是定义数据链路层中 MAC 子层和一小部分 LLC 子层，以及定义 CAN 协议于周围各层当中所发挥的作用（所具有的意义）。

2 基本概念

CAN 具有以下的属性：

- 报文的优先权
- 保证延迟时间
- 设置灵活
- 时间同步的多点接收
- 系统内数据的连贯性
- 多主机
- 错误检测和错误标定
- 只要总线一处于空闲，就自动将破坏的报文重新传输
- 将节点的暂时性错误和永久性错误区分开来，并且可以自动关闭由 OSI 参考模型分层 CAN 结构的错误的节点。

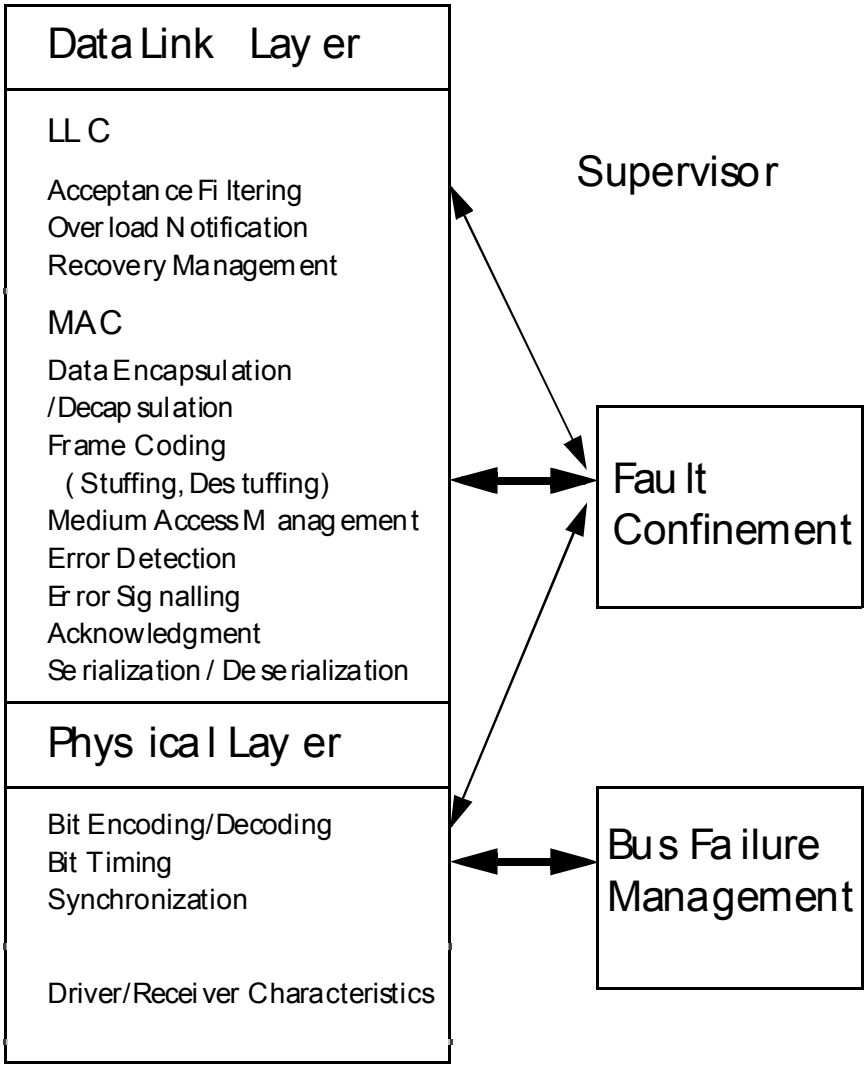
依据 ISO/OSI 参考模型的层结构

- 物理层定义信号是如何实际地传输的，因此涉及到位时间、位编码、同步的解释。本技术规范没有定义物理层的驱动器/接收器特性，以便允许根据它们的应用，对发送媒体和信号电平进行优化。

- MAC 子层是 CAN 协议的核心。它把接收到的报文提供给 LLC 子层，并接收来自 LLC 子层的报文。MAC 子层负责报文分帧、仲裁、应答、错误检测和标定。MAC 子层也被称作故障界定的管理实体监管。

此故障界定为自检机制，以便把永久故障和短时扰动区别开来。

• LLC 子层涉及报文滤波、过载通知、以及恢复管理。



LLC = Logical Link Control
MAC = Medium Access Control

此技术规范的目的是为了定义数据链路层及定义 CAN 协议在周围各层中所发挥的作用(具有的意义)。

报文（Messages）

总线上的报文以不同的固定报文格式发送，但长度受限（见第 3 节的报文传输）。当总线空闲时任何连接的单元都可以开始发送新的报文。

信息路由（Information Routing）

在 CAN 系统里，CAN 的节点不使用任何关于系统配置的报文（比如，站地址）。以下是几个重要的概念：

- 系统灵活性：不需要应用层以及任何节点软件和硬件的任何改变，可以在 CAN 网络中直接添加节点。

- 报文路由：报文的内容由识别符命名。识别符不指出报文的目的地，但解释数据的含义。因此，网络上所有的节点可以通过报文滤波确定是否应对该数据做出反应。
- 多播：由于引入了报文滤波的概念，任何节点都可以接收报文，并与此同时对此报文做出反应。
- 数据连贯性：应确保报文在 CAN 网络里同时被所有的节点接收（或同时不被接收）。因此，系统的数据连贯性是通过多播和错误处理的原理实现的。

位速率（Bit rate）：

不同的系统，CAN 的速度不同。可是，在一个给定的系统里，位速率是唯一的，并且是固定的。

优先权（Priorities）：

在总线访问期间，识别符定义一个静态的报文优先权。

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。数据帧和相应的远程帧是由相同的识别符命名的。

多主机（Multimaster）：

总线空闲时，任何单元都可以开始传送报文。具有较高优先权报文的单元可以获得总线访问权。

仲裁（Arbitration）：

只要总线空闲，任何单元都可以开始发送报文。如果 2 个或 2 个以上的单元同时开始传送报文，那么就会有总线访问冲突。通过使用了识别符的逐位仲裁可以解决这个冲突。仲裁的机制确保了报文和时间均不损失。当具有相同识别符的数据帧和远程帧同时初始化时，数据帧优先于远程帧。仲裁期间，每一个发送器都对发送位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”电平而监视的是一“显性”电平（见总线值），那么单元就失去了仲裁，必须退出发送状态。

安全性（Safety）：

为了获得最安全的数据发送，CAN 的每一个节点均采取了强有力的措施以便于错误检测、错误标定及错误自检。

错误检测：

要进行检测错误，必须采取以下措施：

- 监视（发送器对发送位的电平与被监控的总线电平进行比较）
- 循环冗余检查
- 位填充
- 报文格式检查

错误检测的执行：

错误检测的机制要具有以下的属性：

- 检测到所有的全局错误
- 检测到发送器所有的局部错误
- 可以检测到报文里多达 5 个任意分布的错误
- 检测到报文里长度低于 15（位）的突发性错误
- 检测到报文里任一奇数个的错误

对于没有被检测到的错误报文，其剩余的错误可能性概率低于：报文错误率* $4.7 * 10^{-11}$。

错误标定和恢复时间：

任何检测到错误的节点会标志出损坏的报文。此报文会失效并将自动地开始重新传送。如果不再出现错误的话，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 31 个位的时间。

故障界定（Fault Confinement）：

CAN 节点能够把永久故障和短暂扰动区别开来。故障的节点会被关闭。

连接（Connections）：

CAN 串行通讯链路是可以连接许多单元的总线。理论上，可连接无数多的单元。但由于实际上受延迟时间以及/或者总线线路上电气负载的影响，连接单元的数量是有限的。

单通道（Single Channel）：

总线包括有一单独的通道。通过此通道可以获得数据的再同步报文。要使此通道实现通讯，有许多的方法可以采用，如使用单芯线（加上接地）、2 条差分线、光缆等等。这本技术规范不限制这些实现方法的使用。

总线值（Bus values）：

总线有一个补充的逻辑值：“显性”或“隐性”。“显性”位和“隐性”位同时传送时，总线的结果值为“显性”。比如，在总线的“写-与”执行时，逻辑 0 代表“显性”等级，逻辑 1 代表“隐性”等级。本技术规范不包括表示逻辑等级的物理状态（比如，电压、灯光）。

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。

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。唤醒时，虽然 MAC 子层要等待一段时间使振荡器稳定，然后还要等待一段时间直到与总线活动同步（通过检查 11 个连续的“隐性”的位），但在总线驱动器被重新设置为“总线在线”之前，内部运行已重新开始。

振荡器容差（Oscillator Tolerance）：

位定时要求允许凭经验地把陶瓷谐振器使用在传输率高达 125kbit/s 的应用里。有关更多准确的评估，请参考：

Dais, S; Chapman, M;

“Impact of Bit Representation on Transport Capacity and Clock Accuracy in Serial Data Streams”, SAE Technical Paper Series 890532, Multiplexing in Automobiles SP-773 March 1989

为了满足 CAN 协议的整个总线速度范围，需要使用晶振。

3．报文传输

3.1 帧格式

有两种不同的帧格式，不同之处为识别符场的长度不同：具有 11 位识别符的帧称之为标准帧。而含有 29 位识别符的帧为扩展帧。

3.2 帧类型

报文传输由以下 4 个不同的帧类型所表示和控制：

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，请求发送具有同一识别符的数据帧。

错误帧：任何单元检测到总线错误就发出错误帧。

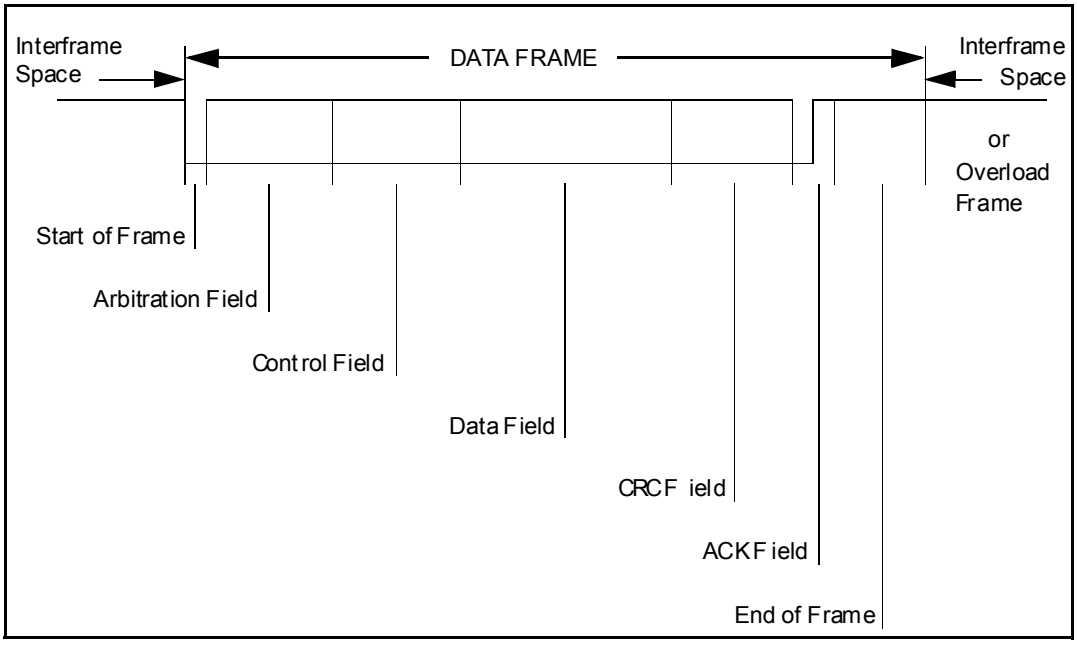
过载帧：过载帧用以在先行的和后续的数据帧（或远程帧）之间提供一附加的延时。

数据帧和远程帧可以使用标准帧及扩展帧两种格式。它们用一个帧间空间与前面的帧分隔。

3.2.1 数据帧（Data Frame）

数据帧由 7 个不同的位场组成：

帧起始（Stsrt of Frame）、仲裁场（Arbitration Frame）、控制场（Control Frame）、数据场（Data Frame）、CRC 场（CRC Frame）、应答场（ACK Frame）、帧结尾（End of Frame）。数据场的长度可以为 0。



帧起始（标准格式和扩展格式）

帧起始（SOF）标志数据帧和远程帧的起始，仅由一个“显性”位组成。

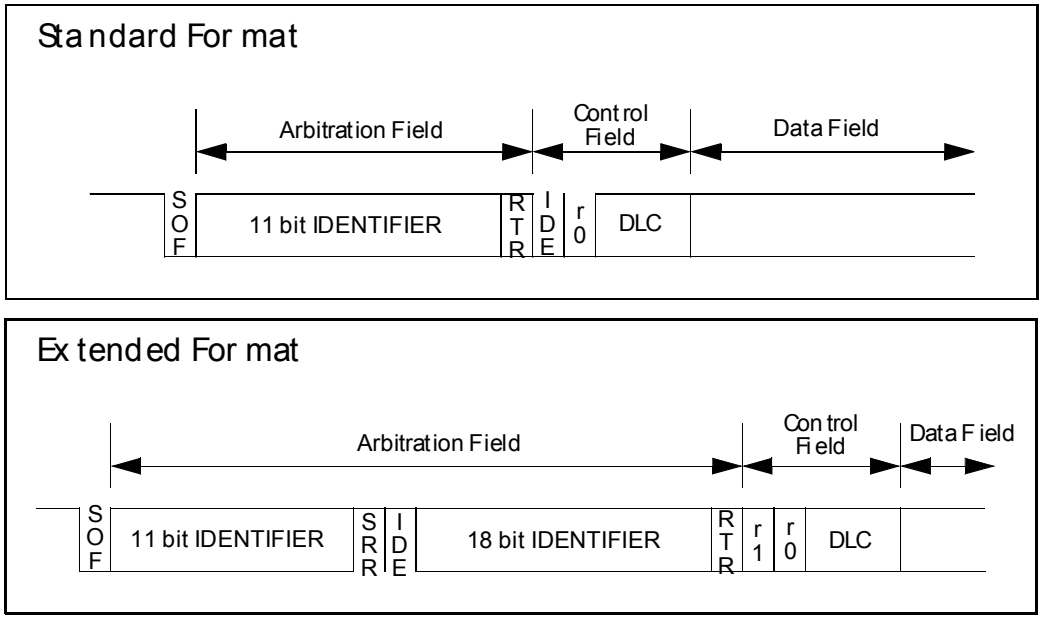
只在总线空闲（参见“总线空闲”）时才允许站开始发送（信号）。所有的站必须同步于首先开始发送报文的站的帧起始前沿（参见“硬同步”）。

仲裁场

标准格式帧与扩展格式帧的仲裁场格式不同。

- 标准格式里，仲裁场由 11 位识别符和 RTR 位组成。识别符位由 ID-28…ID-18。
- 扩展格式里，仲裁场包括 29 位识别符、SRR 位、IDE 位、RTR 位。其识别符由 ID-28... ID-0。

为了区别标准格式和扩展格式，前版本 CAN 规范 1.0-1.2 的保留位 r1 现表示为 IDE Bit。



识别符

识别符—**标准格式**

识别符的长度为 11 位，相当于扩展格式的基本 ID（Base ID）。这些位按 ID-28 到 ID-18 的顺序发送。最低位是 ID-18。7 个最高位（ID-28 - ID-22）必须不能全是“隐性”。

识别符—**扩展格式**

和标准格式形成对比，扩展格式由 29 位组成。其格式包含两个部分：11 位**基本 ID**、18 位**扩展 ID。**

基本 ID：基本 ID 包括 11 位。它按 ID-28 到 ID-18 的顺序发送。它相当于标准识别符的格式。基本 ID 定义扩展帧的基本优先权。

扩展 ID：扩展 ID 包括 18 位。它按 ID-17 到 ID-0 顺序发送。

标准帧里，识别符其后是 RTR 位。

RTR 位（标准格式以及扩展格式）

RTR 的全称为“远程发送请求位（Remote Transmission Request BIT）”。

RTR 位在数据帧里必须为“显性”，而在远程帧里必须为“隐性”

扩展格式里，基本 ID 首先发送，其次是 IDE 位和 SRR 位。扩展 ID 的发送位于 SRR 位之后。

SRR 位（扩展格式）

SRR 的全称是“替代远程请求位（Substitute Remote Request BIT）”。

SRR 是一隐性位。它在扩展格式的标准帧 RTR 位位置，因此代替标准帧的 RTR 位。

因此，标准帧与扩展帧的冲突是通过标准帧优先于扩展帧这一途径得以解决的，扩展帧的基本 ID（参见以下的“扩展识别符”）如同标准帧的识别符。

IDE 位（扩展格式）

IDE 的全称是“识别符扩展位（Identifier Extension Bit）”

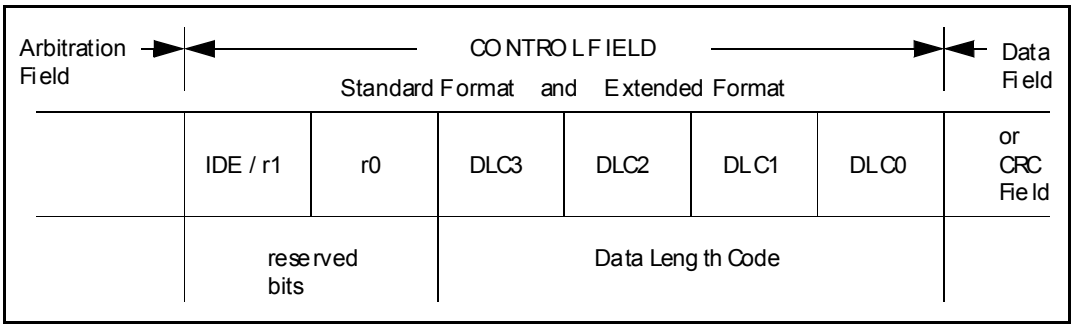
IDE 位属于：

- 扩展格式的仲裁场
- 标准格式的控制场

标准格式里的 IDE 位为“显性”，而扩展格式里的 IDE 位为“隐性”。

控制场（标准格式以及扩展格式）

控制场由 6 个位组成。标准格式的控制场格式和扩展格式的不同。标准格式里的帧包括数据长度代码、IDE 位（为显性位，见上文）、及保留位 r0。扩展格式里的帧包括数据长度代码和两个保留位：r1 和 r0。其保留位必须发送为显性，但是接收器认可“显性”和“隐性”位的组合。



数据长度代码（标准格式以及扩展格式）

数据长度代码指示了数据场里的字节数量。数据长度代码为 4 个位，它在控制场里发送。

数据长度代码中数据字节数的编码

缩写： d—“显性”

r—“隐性”

Number of Data Bytes	Data Length Code			
	DLC3	DLC2	DLC1	DLC0
0	d	d	d	d
1	d	d	d	r
2	d	d	r	d
3	d	d	r	r
4	d	r	d	d
5	d	r	d	r
6	d	r	r	d
7	d	r	r	r
8	r	d	d	d

数据帧：允许的数据字节数：{0,1,....,7,8}。

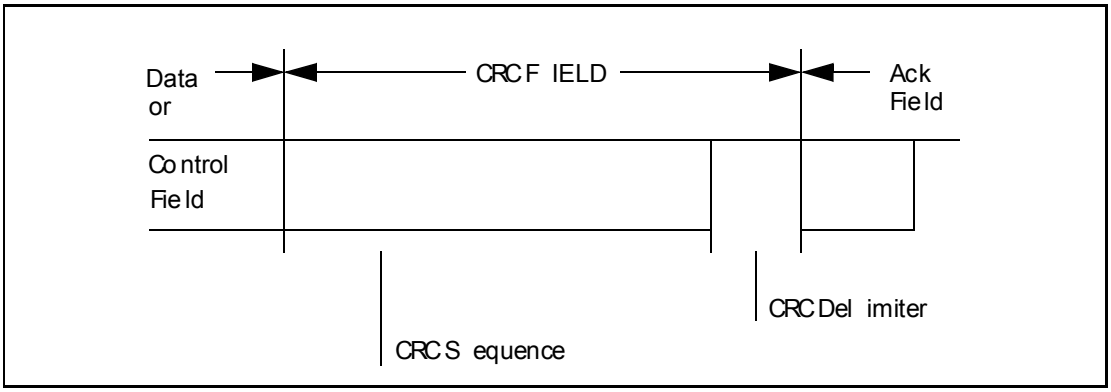
其他的数值不允许使用。

数据场（标准格式以及扩展格式）

数据场由数据帧里的发送数据组成。它可以为 0～8 个字节，每字节包含了 8 个位，首先发送 MSB。

CRC 场（标准格式以及扩展格式）

CRC 场包括 CRC 序列（CRC SEQUENCE），其后是 CRC 界定符（CRC DELIMITER）。



CRC 序列（标准格式以及扩展格式）

由循环冗余码求得的帧检查序列最适用于位数低于 127 位〈BCH 码〉的帧。

为进行 CRC 计算，被除的多项式系数由无填充位流给定，组成这些位流的成分是：帧起始、仲裁场、控制场、数据场（假如有），而 15 个最低位的系数是 0。将此多项式被下面的多项式发生器除（其系数以 2 为模）：

$$X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1$$

这个多项式除法的余数就是发送到总线上的 CRC SEQUENCE（CRC 序列）。为了实现这个功能，可以使用 15 位的位移寄存器—CRC_RG(14:0)。如果 NXTBIT 指示位流的下一位，那么从帧的起始到数据场末尾都由没有填充的位顺序给定。CRC 序列（CRC SEQUENCE）的计算如下：

```
CRC_RG = 0;                                          // 初始化移位寄存器
REPEAT
    CRCNXT = NXTBIT EXOR CRC_RG(14);
    CRC_RG(14:1) = CRC_RG(13:0);                     // 寄存器左移一位
    CRC_RG(0) = 0;
    IF CRCNXT THEN
    CRC_RG(14:0) = CRC_RG(14:0) EXOR (4599hex);
    ENDIF
UNTIL (CRC 序列起始或有一错误条件)
```

在传送/接收数据场的最后一位以后，CRC_RG 包含有 CRC 顺序。

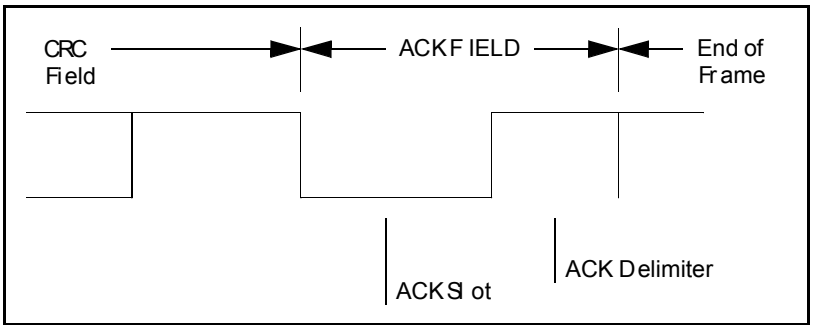
CRC 界定符（标准格式以及扩展格式）

CRC 序列之后是 CRC 界定符，它包含一个单独的“隐性”位 。

应答场（标准格式以及扩展格式）

应答场长度为 2 个位，包含应答间隙（ACK SLOT）和应答界定符（ACK DELIMITER）。在 ACK 场（应答场）里，发送站发送两个“隐性”位。

当接收器正确地接收到有效的报文，接收器就会在应答间隙（ACK SLOT）期间（发送 ACK 信号）向发送器发送一“显性”位以示应答。



应答间隙

所有接收到匹配 CRC 序列（CRC SEQUENCE）的站会在应答间隙（ACK SLOT）期间用一“显性”的位写入发送器的“隐性”位来作出回答。

应答界定符

应答界定符是应答场的第二个位，并且是一个必须为“隐性”的位。因此，应答间隙（ACK SLOT）被两个“隐性”的位所包围，也就是 CRC 界定符（CRC DELIMITER）和应答界定符（ACK DELIMITER）。

帧结尾（标准格式以及扩展格式）

每一个数据帧和远程帧均由一标志序列定界。这个标志序列由 7 个“隐性”的位组成。

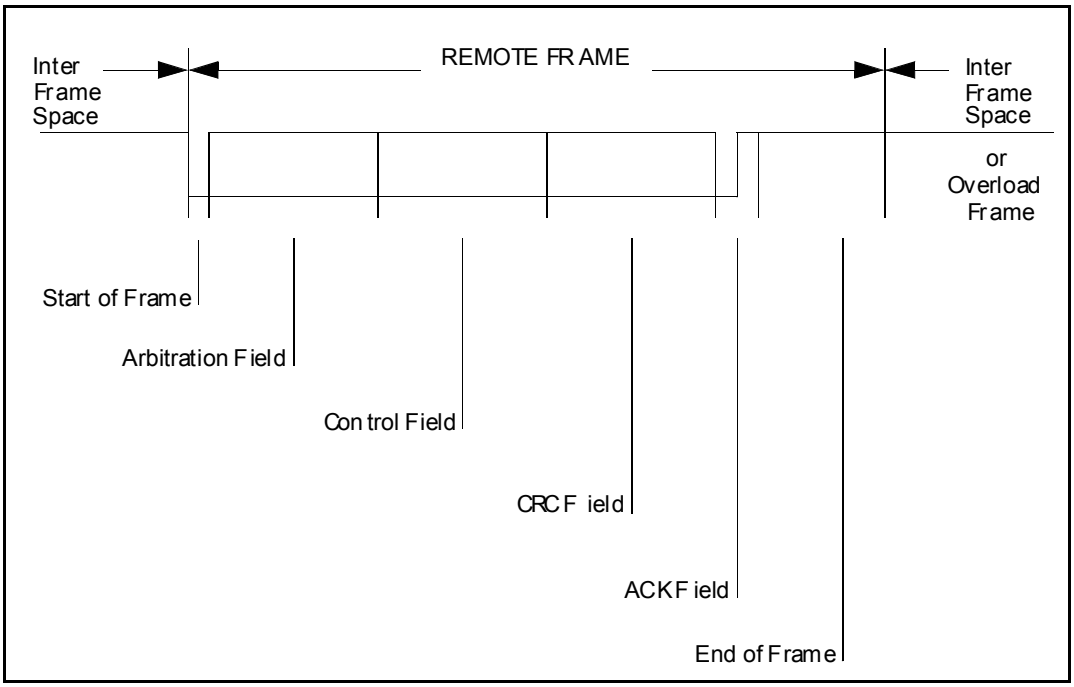
3.2.2 远程帧

通过发送远程帧，作为某数据接收器的站可以初始化通过其资源节点传送不同的数据。

远程帧也有标准格式和扩展格式，而且都由 6 个不同的位场组成：

帧起始、仲裁场、控制场、CRC 场、应答场、帧结尾。

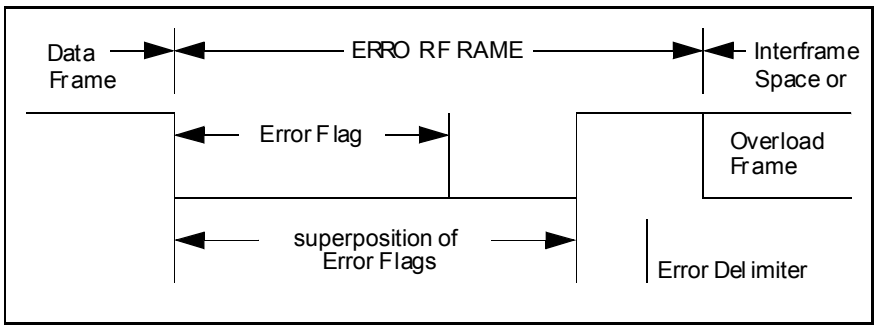
与数据帧相反，远程帧的 RTR 位是“隐性”的。它没有数据场，数据长度代码的数值是不受制约的（可以标注为容许范围里 0...8 的任何数值）。此数值是相应于数据帧的数据长度代码。



RTR 位的极性表示了所发送的帧是一数据帧（RTR 位“显性”）还是一远程帧（RTR“隐性”）。

3.2.3 错误帧

错误帧由两个不同的场组成。第一个场用是不同站提供的错误标志（ERROR FLAG）的叠加。第二个场是错误界定符。



为了能正确地终止错误帧，一“错误被动”的节点要求总线至少有长度为 3 个位时间的总线空闲（如果“错误被动”的接收器有局部错误的话）。因此，总线的载荷不应为 100%。

错误标志

有两种形式的错误标志：主动的错误标志和被动的错误标志。

1. 主动的错误标志由 6 个连续的“显性”位组成。
2. 被动的错误标志由 6 个连续的“隐性”的位组成，除非被其他节点的“显性”位重写。

检测到错误条件的“错误激活”的站通过发送主动错误标志指示错误。错误标志的形式破坏了从帧起始到 CRC 界定符的位填充的规则（参见“编码”），或者破坏了 ACK 场或帧结尾场的固定形式。所有其他的站由此检测到错误条件并与此同时开始发送错误标志。因此，“显性”位（此“显性”位可以在总线上监视）的序列导致一个结果，这个结果就是把个别站发送的不同的错误标志叠加在一起。这个序列的总长度最小为 6 个位，最大为 12 个位。

检测到错误条件的“错误被动”的站试图通过发送被动错误标志指示错误。“错误被动”的站等待 6 个相同极性的连续位（这 6 个位处于被动错误标志的开始）。当这 6 个相同的位被检测到时，被动错误标志的发送就完成了。

错误界定符

错误界定符包括 8 个“隐性”的位。

错误标志传送了以后，每一站就发送“隐性”的位并一直监视总线直到检测出一个“隐性”的位为止。然后就开始发送其余 7 个“隐性”位。

3.2.4 过载帧

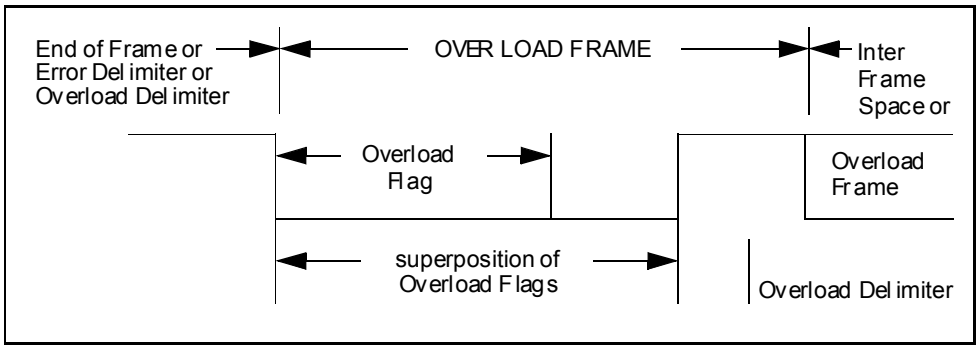
过载帧包括两个位场：过载标志和过载界定符。

有三种过载的情况，这三种情况都会引发过载标志的传送：

1. 接收器的内部情况（此接收器对于下一数据帧或远程帧需要有一延时）。
2. 在间歇的第一和第二字节检测到一个“显性”位。
3. 如果 CAN 节点在错误界定符或过载界定符的第 8 位（最后一位）采样到一个显性位，节点会发送一个过载帧（不是错误帧）。错误计数器不会增加。

根据过载情况 1 而引发的过载帧只允许起始于所期望的间歇的第一个位时间，而根据情况 2 和情况 3 引发的过载帧应起始于所检测到“显性”位之后的位。

通常为了延时下一个数据帧或远程帧，两种过载帧均可产生。



过载标志（Overload Flag）

过载标志由 6 个“显性”的位组成。过载标志的所有形式和主动错误标志的一样。

过载标志的形式破坏了间歇场的固定形式。因此，所有其他的站都检测到过载条件并与此同时发出过载标志。如果有的节点在间歇的第 3 个位期间检测到“显性”位，则这个位将解释为帧的起始。

备注：

基于 CAN1.0 和 CAN1.1 版本的控制器对第 3 个位有另一解释，如下：

有的节点在间歇的第 3 个位期间于本地检测到一“显性”位，则其他的节点将不能正确地解释过载标志，而是将这 6 个“显性”位中的第一个位解释为帧的起始。这第 6 个“显性”的位破坏了产生错误条件的位填充的规则

过载界定符（Overload Delimeter）

过载界定符包括 8 个“隐性”的位。

过载界定符的形式和错误界定符的形式一样。过载标志被传送后，站就一直监视总线直到检测到一个从“显性”位到“隐性”位的跳变。此时，总线上的每一个站完成了过载标志的发送，并开始同时发送其余 7 个“隐性”位。

3.2.5 帧间空间

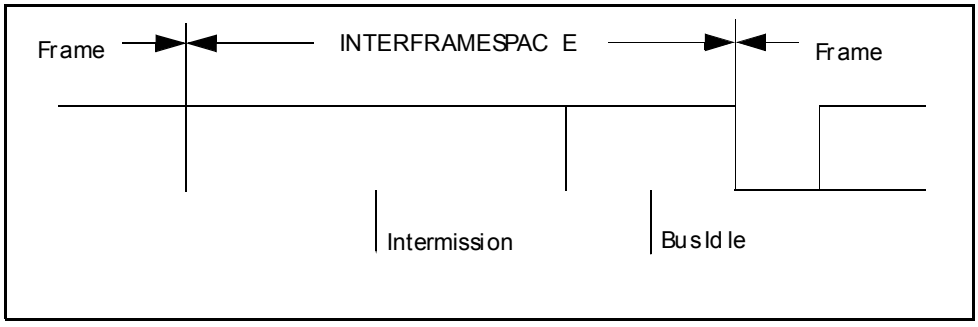
数据帧（或远程帧）与先行帧的隔离是通过帧间空间实现的，无论此先行帧类型如何（数据帧、远程帧、错误帧、过载帧）。所不同的是，过载帧与错误帧之前没有帧间空间，多个过载帧之间也不是由帧间空间隔离的。

帧间空间（Interframe Space）

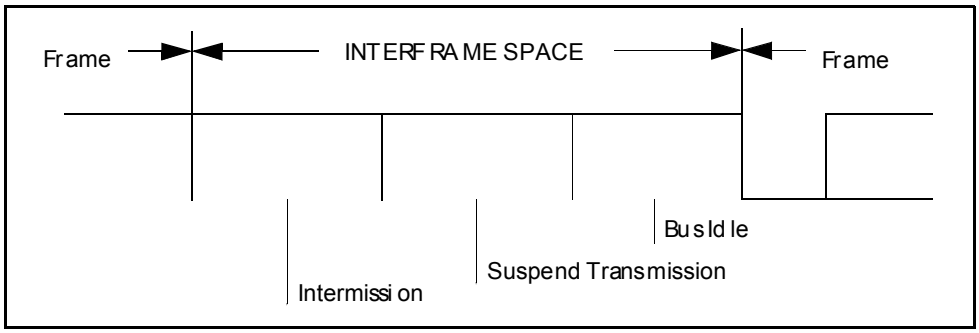
帧间空间包括间歇、总线空闲的位场。如果“错误被动”的站已作为前一报文的发送器时，则其帧空间除了间歇、总线空闲外，还包括称作挂起传送（SUSPEND TRANSMISSION）的位场。

对于已作为前一报文发送器的“错误被动”的站，其帧间空间如下图所示：

:



对于已作为前一报文发送器的“错误主动”的站，其帧间空间如下图所示：



间歇（Intermission）

间歇包括 3 个“隐性”的位。

间歇期间，所有的站均不允许传送数据帧或远程帧，唯一要做的是标示一个过载条件。

备注：

如果 CAN 节点有一报文等待发送并且节点在间歇的第三位采集到一显性位，则此位被解释为帧的起始位，并从下一个位开始发送报文的识别符首位，而不用首先发送帧的起始位或成为一接收器。

总线空闲（Bus IDLE）

总线空闲的时间是任意的。只要总线被认定为空闲，任何等待发送报文的站就会访问总线。在发送其他报文期间，有报文被挂起，对于这样的报文，其传送起始于间歇之后的第一个位。

总线上检测到的“显性”的位可被解释为帧的起始。

挂起传送（Suspend Transmission）

“错误被动”的站发送报文后，站就在下一报文开始传送之前或总线空闲之前发出 8 个“隐性”的位跟随在间歇的后面。如果与此同时另一站开始发送报文（由另一站引起），则此站就作为这个报文的接收器。

3.3 关于帧格式的符合性

标准格式相当于在 CAN1.2 规范中描述的数据/远程帧。而扩展格式是 CAN 协议的一新特色。为了使控制器的设计相对地简单，不要求扩展格式的仪器达到它的满扩展（比如，在扩展格式里发送报文或接收来自于报文的数据 ）。但是，仪器必须无条件地支持标准格式。

如果新的控制器至少具有以下属性（这些属性与 3.1 和 3.2 定义的帧格式有关），则被认为是符合 CAN 规范：

- 每一新的控制器支持标准格式
- 每一新的控制器可以接收扩展格式的报文。这需要扩展格式不因其格式而被破坏。可是，不要求新的控制器非得支持扩展格式。

3.4 发送器/接收器的定义

发送器（Transmitter）

产生报文的单元被称之为报文的“发送器”。此单元保持作为报文发送器直到总线出现空闲或此单元失去仲裁（ARBITRATION）为止。

接收器（Receiver）

如果有一单元不作为报文的发送器并且总线也不空闲，则这一单元就被称之为报文的“接收器”。

4．报文滤波

报文滤波取决于整个识别符。允许在报文滤波中将任何的识别符位设置为“不考虑”的可选屏蔽寄存器，可以选择多组的识别符，使之被映射到隶属的接收缓冲器里。

如果使用屏蔽寄存器，它的每一个位必须是可编程的，即，他们能够被允许或禁止报文滤波。屏蔽寄存器的长度可以包含整个识别符，也可以包含部分的识别符。

5 报文校验

校验报文有效的时间点，发送器与接收器各不相同。

发送器（Transmitter）

如果直到帧的末尾位均没有错误，则此报文对于发送器有效。如果报文破损，则报文会根据优先权自动重发。为了能够和其他报文竞争总线，重新传输必须在总线空闲时启动。

接收器（Receiver）

如果直到一最后的位（除了帧末尾位）均没有错误，则报文对于接收器有效。帧末尾最后的位被置于“不重要”状态，如果是一个“显性”电平也不会引起格式错误（参见 7.1 章节）。

6．编码

位流编码（Bit Stream Coding）

帧的部分，诸如帧起始、仲裁场、控制场、数据场以及 CRC 序列，均通过位填充的方法编码。无论何时，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一补充位。

数据帧或远程帧（CRC 界定符、应答场和帧结尾）的剩余位场形式固定，不填充。错误帧和过载帧的形式也固定，但并不通过位填充的方法进行编码。

其报文里的位流根据“不返回到零”（NRZ）之方法来编码。这就是说，在整个位时间里，位的电平要么为“显性”，要么为“隐性”。

7．错误处理

7.1 错误检测

有以下 5 种不同的错误类型（这 5 种错误不会相互排斥）

• 位错误（Bit Error）

单元在发送位的同时也对总线进行监视。如果所发送的位值与所监视的位值不相符合，则在此位时间里检测到一个位错误。但是在仲裁场（ARBITRATION FIELD）的填充位流期间或应答间隙（ACK SLOT）发送一“隐性”位的情况是例外的—— 此时，当监视到一“显性”位时，不会发出位错误。当发送器发送一个被动错误标志但检测到“显性”位时，也不视为位错误。

• 填充错误（Struff Error）

如果在使用位填充法进行编码的信息中，出现了第 6 个连续相同的位电平时，将检测到一个填充错误。

• CRC 错误（CRC Error）

CRC 序列包括发送器的 CRC 计算结果。接收器计算 CRC 的方法与发送器相同。如果计算结果与接收到 CRC 序列的结果不相符，则检测到一个 CRC 错误。

• 形式错误（Form Error）

当一个固定形式的位场含有 1 个或多个非法位，则检测到一个形式错误。(备注：接收器的帧末尾最后一位期间的显性位不被当作帧错误)

• 应答错误（Acknowledgment Error）

只要在应答间隙（ACK SLOT）期间所监视的位不为“显性”，则发送器会检测到一个应答错误。

7.2 错误标定

检测到错误条件的站通过发送错误标志指示错误。对于“错误主动”的节点，错误信息为“主动错误标志”，对于“错误被动”的节点，错误信息为“被动错误标志”。站检测到无论是位错误、填充错误、形式错误，还是应答错误，这个站会在下一位时发出错误标志信息。

只要检测到的错误的条件是 CRC 错误，错误标志的发送开始于 ACK 界定符之后的位（其他的错误条件除外）。

8．故障界定

至于故障界定，单元的状态可能为以下三种之一：

- ‘错误主动’
- ‘错误被动’
- ‘总线关闭’

“错误主动”的单元可以正常地参与总线通讯并在错误被检测到时发出主动错误标志。

“错误被动”的单元不允许发送主动错误标志。“错误被动”的单元参与总线通讯，在错误被检测到时只发出被动错误标志。而且，发送以后，“错误被动”单元将在初始化下一个发送之前处于等待状态。（见“挂起发送”）

“总线关闭”的单元不允许在总线上有任何的影响（比如，关闭输出驱动器）。

在每一总线单元里使用两种计数以便故障界定：

1) 发送错误计数

2) 接收错误计数

这些计数按以下规则改变（注意，在给定的报文发送期间，可能要用到的规则不只一个）：

1. 当接收器检测到一个错误，接收错误计数就加 1。在发送主动错误标志或过载标志期间所检测到的错误为位错误时，接收错误计数器值不加 1。

2. 当错误标志发送以后，接收器检测到的第一个位为“显性”时，接收错误计数值加 8。

3. 当发送器发送一错误标志时，发送错误计数器值加 8

例外情况 1：

发送器为“错误被动”，并检测到一应答错误（注：此应答错误由检测不到一“显性”ACK 以及当发送被动错误标志时检测不到一“显性”位而引起）。

例外情况 2：

发送器因为填充错误而发送错误标志（注：此填充错误发生于仲裁期间。引起填充错误是由于：填充位〈填充位〉位于 RTR 位之前，并已作为“隐性”发送，但是却被监视为“显性”）。

例外情况 1 和例外情况 2 时，发送错误计数器值不改变。

4. 发送主动错误标志或过载标志时，如果发送器检测到位错误，则发送错误计数器值加 8。

5. 当发送主动错误标志或过载标志时，如果接受器检测到位错误（位错误），则接收错误计数器值加 8。

6. 在发送主动错误标志、被动错误标志或过载标志以后，任何节点最多容许 7 个连续的 “显性”位。以下的情况，每一发送器将它们的发送错误计数值加 8，及每一接收器的接收错误计数值加 8：

当检测到第 14 个连续的“显性”位后；

在检测到第 8 个跟随着被动错误标志的连续的“显性”位以后；

在每一附加的 8 个连续“显性”位顺序之后。

7. 报文成功传送后（得到 ACK 及直到帧末尾结束没有错误），发送错误计数器值减 1，除非已经是 0。

8. 如果接收错误计数值介于 1 和 127 之间，在成功地接收到报文后（直到应答间隙接收没有错误，及

成功地发送了 ACK 位），接收错误计数器值减 1。如果接收错误计数器值是 0，则它保持 0，如果大于 127，则它会设置一个介于 119 到 127 之间值。

9. 当发送错误计数器值等于或超过 128 时，或当接收错误计数器值等于或超过 128 时，节点为“错误被动”。让节点成为“错误被动”的错误条件致使节点发出主动错误标志。

10.当发送错误计数器值大于或等于 256 时，节点为“总线关闭”。

11. 当发送错误计数器值和接收错误计数器值都小于或等于 127 时，“错误被动”的节点重新变为“错误主动”。

12. 在总线监视到 128 次出现 11 个连续“隐性”位之后，“总线关闭”的节点可以变成“错误主动”（不再是“总线关闭”），它的错误计数值也被设置为 0。

备注：

一个大约大于 96 的错误计数值显示总线被严重干扰。最好能够预先采取措施测试这个条件。

备注：

起动/睡眠：如果起动期间内只有 1 个节点在线，以及如果这个节点发送一些报文，则将不会有应答，并检测到错误和重复报文。由此，节点会变为“错误被动”，而不是“总线关闭”。

9 振荡器容差

由于给定的最大的振荡器容差为 1.58%，因此凭经验可将陶瓷谐振器使用在传输率高达 125 kbit/s 的应用里。有关更多准确的评估，请参考：

Dais, S; Chapman, M;
“Impact of Bit Representation on Transport Capacity and Clock Accuracy in Serial Data Streams”,
SAE Technical Paper Series 890532, Multiplexing in Automobiles SP-773 March 1989

为了满足 CAN 协议的整个总线速度范围，需要使用晶振。

具有最高振荡准确度要求的芯片，决定了其他节点的振荡准确度。

备注：

使用这个版本 CAN 规约的控制器以及使用前版本 V1.0 和 V1.1 控制器，当它们在一个网络中共同作用时，所有的控制器必须配备石英晶振。这就是说，陶瓷谐振器只能用于所有节点都为 CAN 协议规范 V1.2 或更晚版本的网络。

10．位定时要求

标称位速率

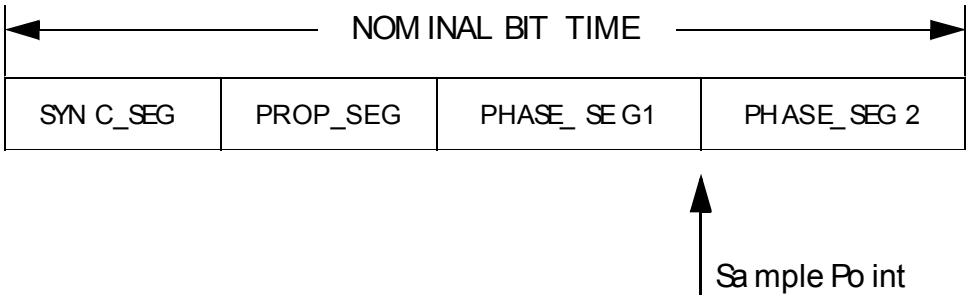
标称位率为一理想的发送器在没有重新同步的情况下每秒发送的位数量。

标称位时间

标称位时间 = 1 /标称位速率

可以把标称位时间划分成了几个不重叠时间的片段，它们是：

- 同步段(SYNC_SEG)
- 传播时间段(PROP_SEG)
- 相位缓冲段 1(PHASE_SEG1)
- 相位缓冲段 2(PHASE_SEG2)



同步段（SYNC SEG）

位时间的同步段用于同步总线上不同的节点。这一段内要有一个跳变沿。

传播段（PROP SEG）

传播段用于补偿网络内的物理延时时间。它是总线上输入比较器延时和输出驱动器延时总和的两倍。

相位缓冲段 1、相位缓冲段 2（PHASE SEG1、PHASE SEG2）

相位缓冲段用于补偿边沿阶段的误差。这两个段可以通过重新同步加长或缩短。

采样点（SAMPLE POINT）

采样点是读总线电平并解释各位的值的一个时间点。采样点位于相位缓冲段 1（PHASE_SEG1）之后。

信息处理时间（INFORMATION PROCESSING TIME）

信息处理时间是一个以采样点作为起始的时间段。采样点用于计算后续位的位电平。

时间份额（TIME QUANTUM）

时间份额是派生于振荡器周期的固定时间单元。存在有一个可编程的预比例因子，其整体数值范围为 1—32 的整数，以最小时间份额为起点，时间份额的长度为：

时间份额（TIME QUANTUM）＝ m * 最小时间份额（MINIMUM TIME QUANTUM）

（m 为预比例因子）

时间段的长度（Length of Time Segments）

同步段（SYNC_SEG）为 1 个时间份额； 传播段（PROP_SEG）的长度可设置为 1，2，…，8 个时间份额；缓冲段 1 （PHASE_SEG1）的长度可设置为 1，2，…，8 个时间份额；相位缓冲段 2（PHASE_SEG2）的长度为阶段缓冲段 1（ PHASE_SEG1）和信息处理时间（INFORMATION PROCESSING TIME）之间的最大值； 信息处理时间少于或等于 2 个时间份额。

一个位时间总的的时间份额值可以设置在 8—25 的范围。

备注：

人们通常不想在控制单元的现场 CPU 和它的通讯器件里使用不同的振荡器。因此，CAN 器件的的振荡频率趋向于现场 CPU 的振荡频率，而且取决于控制单元的需求。为了得到所需的比特率，位定时的可设置性是有必要的。另一方面，由于这些器件允许选择外部的振荡器以便于被调整到合适的比特率，因此，对于这些部件，可配置性不是必要的。

但是，应该将所有节点的采样点选择于共有的位置。为此，SLIO 器件必须兼容以下的位时间定义。

硬同步（HARD SYHCHRONIZATION）：

硬同步后，内部的位时间从同步段重新开始。因此，硬同步强迫由于硬同步引起的沿处于重新开始的位时间同步段之内。

重新同步跳转宽度（RESYHCHRONIZATION JUMP WIDTH）

重新同步的结果使相位缓冲段 1 增长，或使相位缓冲段 2 缩短。相位缓冲段加长或缩短的数量有一个上限，此上限由重新同步跳转宽度给定。重新同步跳转宽度应设置于 1 和最小值之间（此最小值为 4，PHASE_SEG1）

可以从一位值转换到另一位值的过渡过程得到时钟信息。这里有一个属性，即：只有后续位的一固定最大数值才具有相同的数值。这个属性使总线单元在帧期间重新同步于位流成为可能。可用于重新同步的两个过渡过程之间的最大的长度为 29 个位时间。

一个沿的相位误差（PHASE ERROR of an edge）

一个沿的相位误差由相关于同步段的沿的位置给出，以时间额度量度。相位误差定义如下：

- e = 0 如果沿处于同步段里（SYNC_SEG）。.
- e > 0 如果沿位于采集点（SAMPLE POINT）之前。.
- e < 0 如果沿处于前一个位的采集点（SAMPLE POINT）之后。

重新同步（RESYHCHRONIZATION）

当引起重新同步沿的相位误差的幅值小于或等于重新同步跳转宽度的设定值时，重新同步和硬件同步的作用相同。当相位错误的量级大于重新同步跳转宽度时：

- 如果相位误差为正，则相位缓冲段 1 被增长。增长的范围为与重新同步跳转宽度相等的值。
- 如果相位误差为负，则相位缓冲段 2 被缩短。缩短的范围为与重新同步跳转宽度相等的值。

同步的原则（SYHCHRONIZATION RULES）

硬同步和重新同步都是同步的两种形式，遵循以下规则：

1. 在一个位时间里只允许一个同步。
2. 仅当采集点之前探测到的值与紧跟沿之后的总线值不相符合时，才把沿用作于同步。
3. 总线空闲期间，有一“隐性”转变到“显性”的沿，无论何时，硬同步都会被执行。
4. 符合规则 1 和规则 2 的所有从“隐性”转化为“显性”的沿可以用作为重新同步。有一例外情况，即，当发送一显性位的节点不执行重新同步而导致一“隐性”转化为“显性”沿，此沿具有正的相位误差，不能用作于重新同步。

区别

这本技术规范的 B 部分已经包括了 CAN1.2 的修改。以下各个内容替换的地方用星号标注。

B part

关于故障界定第 6 个规则的更改

B part

根据 ISO/OSI 的参考模型，通过不同层来说明 CAN 的层结构

B part

包含在“振荡器容差”里的注释

B part

识别符的位数已修改

B part

根据振荡器容差，介绍了第三种产生过载帧的条件。

B part

增加了一个注释。因为有关间歇最后一个位的解释已被修改。

B part

有一注释的介绍，因为关于帧的起始有另一解释。

B part

3.3 章节“关于帧格式的符合性”

B part

第 4 章，最近介绍了“报文滤波”

B part

关于根据振荡器容差修改的协议兼容性

B part

关于没有本地 CPU 的 ECUs 实现的位定时注释