

控制器局域网（CAN）——未来的工业微处理器串行通讯？

目录

CAN 介绍	2
低成本的远程 IO.....	2
两种 CAN	2
报文冲突的处理.....	3
交互的通讯	3
物理层	3
特定工业领域的定制 CAN.....	4
用 CAN 作为解决方案的情况	4
CAN 总线协议	4
CAN 控制器.....	5
CAN 报文帧类型	5
总线仲裁.....	6
CAN 错误检测	7
CAN 支持工具	8

CAN 介绍

控制器局域网（CAN）是由 ISO 定义的串行通讯总线，它最初出现在 80 年代末的汽车工业里。它的基本设计规范要求有高的位速率，高抗电磁干扰性，而且能够检测出产生的任何错误。由于 CAN 串行通讯总线具有这些特性，它很自然地在汽车、制造业以及航空工业中受到广泛应用。

CAN 通讯协议描述了在设备之间信息如何传递。它对层的定义与开放系统互连模型（OSI）一致。每一层与另一设备上相同的那一层通讯。实际的通讯是发生在每一设备上相邻的两层，而设备只通过模型物理层的物理介质互连。CAN 的结构定义了模型的最下面的两层：数据链路层和物理层。应用层通过不同的新兴协议层（专门用于特殊的工业领域加上由个别 CAN 用户定义的任何合适的方案）和物理层连接。

可能最好的 CAN 协议工业标准也许就是 Allen-Bradley 的 DeviceNet，它是为 PLC 和智能传感器设计的。国际上广泛应用的 CAN 协议工业标准还有 CANopen 和 SDS。

物理介质包括带有特定终端的双绞线。在 BasicCAN 的规范中，它的传输速率能达到 250KBaud，而 PeliCAN 能达到 1Mbaud。

物理层和数据链路层对于系统设计者来说是透明的，并包含在所有执行 CAN 协议的部件中。这些部件有：带有集成 CAN 接口的微控制器，例如与 8051 兼容的 Philips P97C591 处理器和 16 位的 Philips XA。82C200、SJA1000 是一个独立的 CAN 控制器，它可以和很多微控制器直接接口。而与物理介质连接可以用分离部件或者 82C250、TJA1050 集成电路或相似的方法实现。SIEMENS、NEC 和 INTEL 也提供独立的 CAN 控制器。

低成本的远程 IO

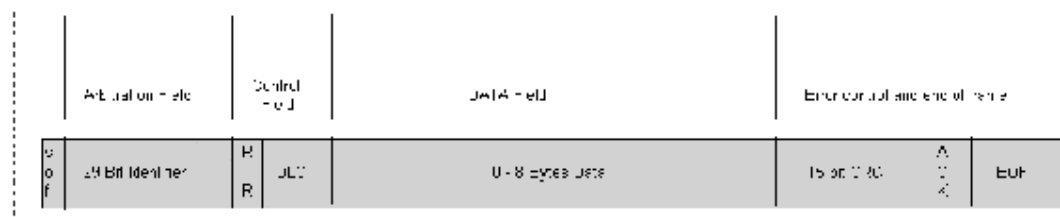
传统上，CAN 是一个基于微控制器的器件互相连接的网络，这意味着其每个节点的成本并不是特别的低。最有趣的发展成果就是“SLIO”模块这个概念。它是一个单芯片，能够在 CAN 网络里充当一个沉默的输入输出网关，并把报文转化为实际的数字 IO 信号。它能读 IO 引脚并把数据当作报文传输，它还能使用集成 A/D 转换器来生成报文并传输（引进）到网络上。这些装置非常便宜并且对于驱动远程感应器、执行器或采集数字和模拟数据都非常理想。它们能够被看作中央微控制器的远程附件。现今来说，只能使用 BasicCAN SLIO，但毫无疑问 Philips 和其他 PeliCAN 厂家将生产出相应的 PeliCAN 的设备。

两种 CAN

CAN 以两种形式存在：一种是具有 11 位 ID 标识符的 BasicCAN，另一种是带有扩展成 29 位 ID 标识符的高级形式 PeliCAN。内部的验收滤波器、屏蔽滤波器可通过标识符 ID 来接收需要的报文，屏蔽不相关的报文，即只向 CPU 提交合适的报文。Philips、Intel、Siemens 均支持 BasicCAN 和 PeliCAN。同时，PeliCAN 协议允许两段长度的标识符：A 部分使用 11 位报文标识符，能够识别出 2032 个不同的标识符（保留十六位），此部分兼容 BasicCAN；而 PeliCAN（B 部分）有 29 位，能够产生 536870912 个不同的标识符。

PART A 的设备只能用标准的 CAN 协议发送和接收。如果在 29 位 ID 的扩展 CAN 系统中使用，设备会发生错误并破坏网络。SIEMENS 81C90 和 81C91 是相类似的 PART A（11 位 ID）设备，但可在扩展 CAN 上使用而不会引起总线错误。原因很简单，因为它们忽略了扩展 CAN 帧，而是被认为是 PART B 设备。PART B 的设备（PeliCAN）则可由设定，工作于其中一种模式下。

数据链路层定义了报文传输的格式和定时协议。这里有两种描述符，都能达到 8 字节的数据。描述符是非常重要的，因为它们定义了报文的优先权以及报文传输的类型。



标识符场有 11 位，并用来识别报文以及决定总线访问的优先权。标识符的二进制值越小，其优先权就越高。报文的优先权分配是 CAN 总线的特征之一，特别在一个强大的实时控制环境里，这一点尤其吸引人。标识符场中所有的位都能够决定报文的优先权。这意味着，在一个 11 位系统里面，报文的优先权编号（号码）可以从 16（高优先权）到 2032（低优先权）。CAN 规范保证了优先权与延时是相关联的。

报文冲突的处理

如前所说，CAN 的一个基本特征是越低的报文编号（号码），其优先权越高——一个标识符如果全部是零的话，那么它具有最高优先权。因而，两个节点同时开始传送报文，第一个发送零而另一个发送 1 的话，那么发送零的那个就能够先分配到 CPU 并完成报文发送。这种“非破坏性的逐位”仲裁与每一个节点都可以监控自己发送的能力联合使用。因而如果一个发送器“A”被另一个发送器“B”发送有更高优先权的报文支配（overrule），回读的报文和“A”企图发送的报文不一致，“A”将暂停发送。一旦总线空闲，另一个请求就会马上发送。这是第一层的部分功能，它完全包含在 CAN 控制器里。因此它对 CAN 用户来说是透明的。

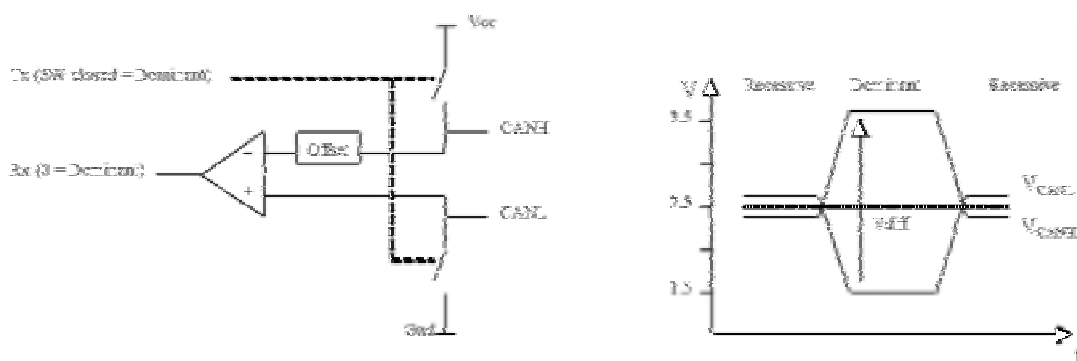
交互的通讯

它可以发送一个数据请求到特定的地址，远程发送请求（RTR）位定义了报文发送数据请求还是发送实际数据。数据长度代码告诉接收器报文里面包含了多少个数据字节。如果是数据请求，则没有数据字节随传，那么数据长度代码就不会与数据字节数有直接关系。

在 CAN 总线中最大的节点数典型是 32 个但是最终的数量是由物理层的特性决定。在总线为 250KB 传送速率的 CAN 系统中，根据每个报文的字节数，系统每秒钟大约能发送 2000 个到 5000 个报文。

物理层

CAN 能够使用很多物理介质例如双绞线，光纤等。只要物理驱动器是在“开集电极”而且每个节点都能够监听到它自己以及其他所有节点，那么 CAN 就能够工作。最常用的就是双绞线。一个信号能够使用差分电压传送，CAN 驱动器能够因此而避免噪声和容错。这两条信号线被称为“CAN_H”和“CAN_L”，静态时是 2.5V。用 CAN_H 比 CAN_L 高表示的逻辑“0”被称为“显形”位，而用 CAN_L 比 CAN_H 高表示的逻辑“1”叫做“隐性”位。



差分的电压令 CAN 网络即使在一条信号线断开或者在噪声极大的环境中也能够工作。只需要一对双绞线，差分的 CAN 输入就能够很有效地抵偿噪音，只需要该噪音是在通常的波段里。

CAN 线收发器提供便宜的接口，把 5V 的逻辑电平转换成 CAN 要求的对称（线）电平。

特定工业领域的定制 CAN

报文号码的分配（与优先权有关）可以由个别的用户字节确定，但要注意某些工业集团互相统一了某些报文的意义以及实际使用的协议。例如，发动机驱动器的制造商可能会决定报文 0x10 在 CAN 网络中是任何线圈的电流反馈信号，而 0x11 是转速计速度。而 0x10 比 0x11 先有一个零，和电流值相关的报文将使和转速计读数相关的报文无效（不能传输）。

在 DeviceNet 中，不同的制造商生产的多种 PLC 能够相连在一起。外围设备如压力和温度传感器都能添加到 CAN 网络里。由于预先定义好传感器产生报文，PLCs 将知道哪些报文总是和温度有关，而不需要考虑它是哪个厂家的产品。

用 CAN 作为解决方案的情况

CAN 在微控制器之间需要互相通信或微控制器和远程的外围器件要互相通信的情况下是一个理想的解决方法。在它的原始应用环境——车中，CAN 最初用于关键任务的实时监控系统，例如引擎管理系统和变速箱控制交换信息。而在这里 CAN 的短报文和有保证的报文延迟时间允许每一个网络的端口都能用当前的数据工作，甚至数据的改变时间在上百个微秒时标的情况下也可以使用。这些系统都利用 CAN 控制器的 PeliCAN 将不需要的报文滤出以减少主 CPU 的负载。但是，低成本的独立 PeliCAN 设备就允许不是实时的任务例如门系统（带有窗的升降，镜子控制等）逐步成为 CAN 网络的一部分。实际上，传统的线束 CAN 网络在一些情况下，甚至在普通的器件上（譬如刹车灯和指示灯只是附加的节点），代替了两线的 CAN 网络。

同时，11 个标识符的 BasicCAN 已经越来越广泛地被应用到工业控制领域的一般网络中。PHILIPS 努力发展并正在推广的 PeliCAN 允许微控制器和外围器件之间以 1M 波特的速度进行简单通讯。实际上，即使是便宜的 SLIO 设备也有高达 16 个 IO 引脚，加上原来的 IO 引脚能够分配 6 个 10 位的 A/D 或 D/A 信道。SLIOs 有由外部电阻设定的唯一的标识符。因而它们能够识别给它们的报文以及在已接收的报文基础上生成报文。

工业应用如果使用有 PeliCAN 的 P87C591（它们也生产 CAN 控制器，作为普通微控制器的附件）微控制器，它就能从 PeliCAN 的 1M 波特中获益。但是，PeliCAN 的原则是，它为微处理单元之间高速数据的交换保留而不是和低速的 IO 端口级通讯。

CAN 总线协议

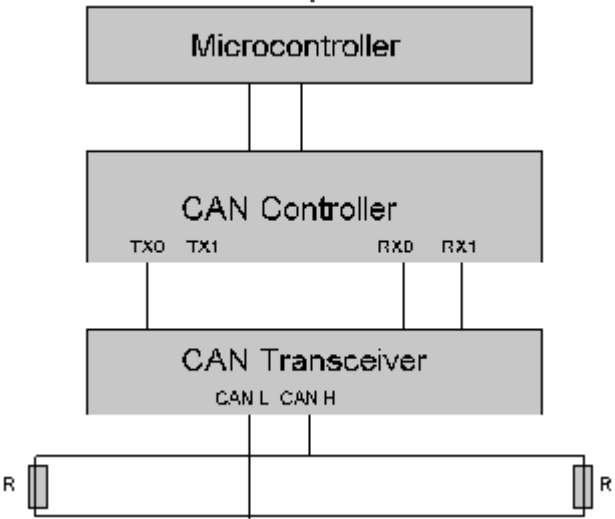
CAN 总线被设计为具有最大传输速率 1Mbit/sec 的多主结构。CAN 不像传统的网络，它不会点到点地传送报文，例如从 A 节点到 B 节点。在 CAN 报文中，标识符是给予数据而不是节点；报文在网络中广播，任何对报文有兴趣的节点都能够接收这个数据。例如在车上，一个节点可能传送了车轮的速度。这个数据可能会同时被 ABS 单元、设备组（INSTRUMENT CLUSTER）以及引擎管理系统所接收，而这些部件会知道这个信息是从哪里发出。

在一个典型的嵌入式系统中，传送少量的重要数据是必要的，这些数据必须在一个实时的时间帧内到达，这与非实时的网络譬如在 PC LAN 中传输一个文件不同。一般来说，CAN 报文包允许传输最高 8 个字节的数据，而总线仲裁的方法保证了一个重要的数据能够在一个保证的时隙内到达。

CAN 是特别设计用于电噪声很大的环境，这个环境中的报文最容易被丢失或破坏。CAN 协议提供了五种错误检测和修正的方法，因此如果报文被破坏，它能够检测出来而且网络中的所有节点会忽略这个报文；该报文会一直被传送到被准确接收为止。

从一个设计者的观点来看，CAN 协议主要的吸引之处在于整个协议都是由一块芯片以及程序来运行，它看起来就像一个增强的多信道 UART。所有的错误检测传送和接收技术都能够用 CAN 控制器芯片的硬件透明地执行。

CAN 控制器

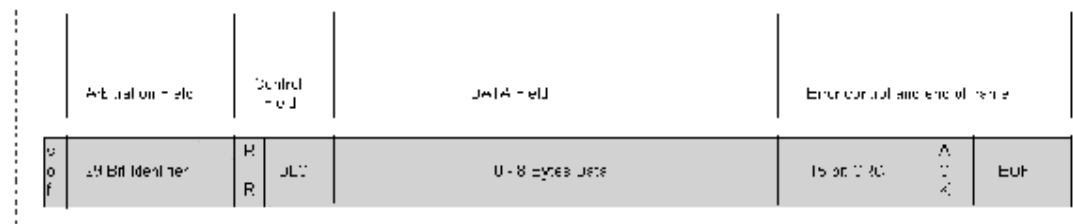


典型的 CAN 节点都带有一个微控制器，这个微控制器中专用的 CAN 控制器存储器映射到它的外部地址总线。一些控制器被设计成可以作为并行设备和地址总线接口或通过控制器的 UART 的串行接口或者甚至是一对端口引脚和地址总线接口。

CAN 控制器要通过线驱动器和总线接口。总线典型是双绞线，两端都接上一个 120 Ω 的电阻。CAN 有发送和接收两端，因此它能够同时读写总线。这个功能对于错误检测与总线仲裁都很重要。CAN 收发器和物理层并不由 CAN 规范来定义，但是只要它符合上述的要求，就可以使用例如 RS485 驱动器和 OP AMPS（运算放大器）等工具。PHILIPS, SILICONIX 和 TEXAS 等都供应专用的 CAN 驱动芯片，这些 CAN 收发器有 8 引脚表面装配和 DIP 封装的两种形式。

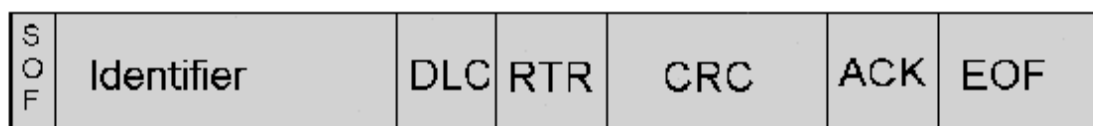
CAN 报文帧类型

在 CAN 总线中，有四个通讯对象：带有应用数据的报文帧，向网络请求数据的远程帧，能够报告每个节点错误的出错帧，如果节点的接收器电路尚未准备好就会延迟传送的过载帧。



报文帧在标识符前面具有一个“帧起始（START OF FRAME）”场。标识符可以有 29 位（扩展 CAN 或者 CAN 2.0B）或者 11 位（CAN 2.0A）。当节点接收到一个报文，与所使用的 CAN 结构有关的标识符，就会与验收滤波器或者用户在 CAN 控制器定义的标识符列表相比较。如果标识符通过了验收过滤器或者与列表的标识符匹配，它就会被节点接收并产生 CPU 中断。因而相同的报文能够被设计者要求的节点所接收。标识符同样可以用于仲裁场中，如果标识符的编号越低，那么报文的优先权就越高。标识符后面是控制场，控制场包含数据长度场（其中的 DLC 规定了数据场中字节的数量），RTR 场将在稍后讨论。剩下的场是由 CAN 控制器产生，包含了 15 位循环冗余检查（对前面的场进行计算）和一个应答隙（ACK 场）。

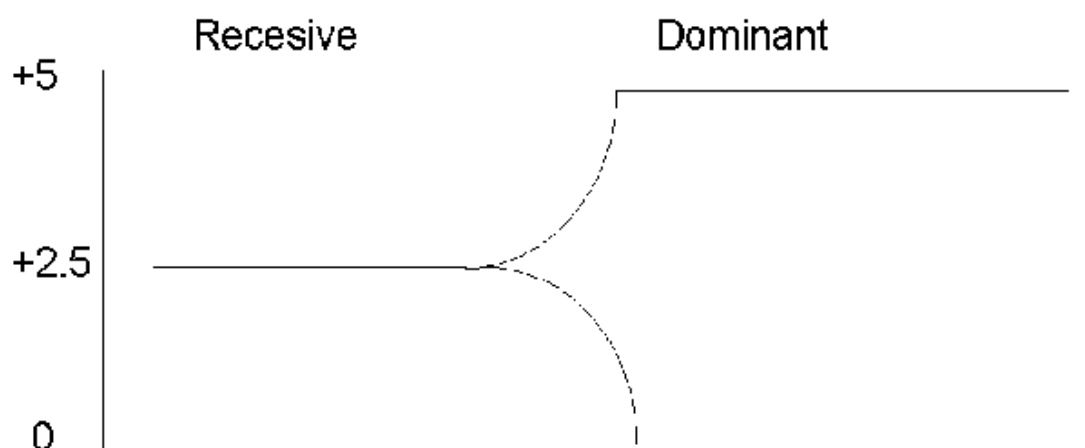
应答隙是报文包中的一个间隙，它允许一个已经正确接收报文的节点插入一个应答。这等于告诉了发送器报文已经被正确接收。如果发送节点没有得到应答，它会继续重新发送报文直到得到应答为止。



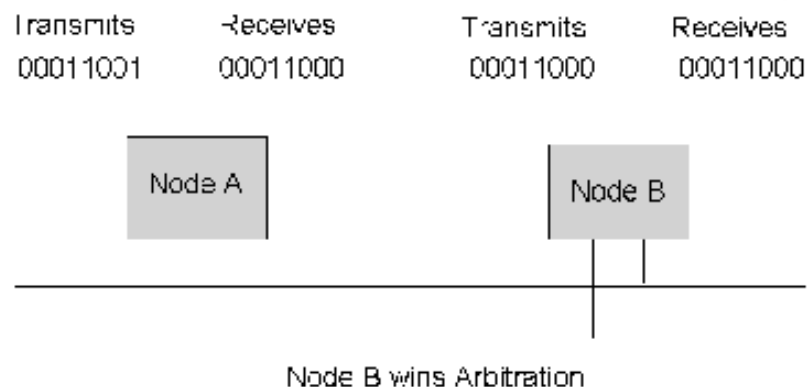
远程请求帧是报文帧的简短形式。它用于向网络请求数据。传送数据请求的标识符时，远程传输请求位（RTR）要置位。与标识符相配的节点这时就会将请求的数据用报文帧传送。有一点要注意的是，DLC在远程帧中并不是为零，而是被设定成期望接收到的数据字节的数量。一般来说，远程帧是用在低总线负荷的网络中，这种情况下主机向网络查询数据。这就简化了正个系统的设计并避免了总线冲突。

以上的帧都服从位填充规则。这个规则中，如果有超过五个连续的 1 或 0，将插入一个极性相反位以防止纯 DC（直流）电平。这种错误违背规则。我们将在介绍总线仲裁后再介绍错误处理。

总线仲裁



CAN 总线物理层通常是双绞线。当逻辑“1”被写进总线里，两条线的电平是 2.5V，并被定义为“隐性”位。当逻辑“0”被写进总线里，一条线被上拉到为 5V（CAN 高）另一条线被下拉到地（CAN 低）。这叫做“显形”位。但是如果显形位和隐性位都被不同的节点同时写进总线里，总线显示“显形”位，所以显性位覆盖了隐性位，这些都是 CAN 网络冲突检测的基础。



当节点开始传送它们各自的报文时，标识符的每一个位都被写到总线里，而且能够被每一个节点读回。如果一个节点写进了隐性位而读回显性位，它会知道另一个较低标识符号码的（高优先权）节点正在访问总线。它会停止传送报文并继续接收更重要的报文。这种技术叫做“非破坏性逐位”仲裁。它保证了如果发送一个高优先权报文，它将会赢得仲裁并能够在保证的时间限度里到达它的目的节点，这些都由系统设计者来控制。必须注意：如果两个节点用同样的标识符发出数据，两个都会得到总线的仲裁并开始传输数据。在这一点，数据会不同并产生错误。两个节点都会后退并且重新开始整个传输过程；在最坏的情况下，它会锁死网络！

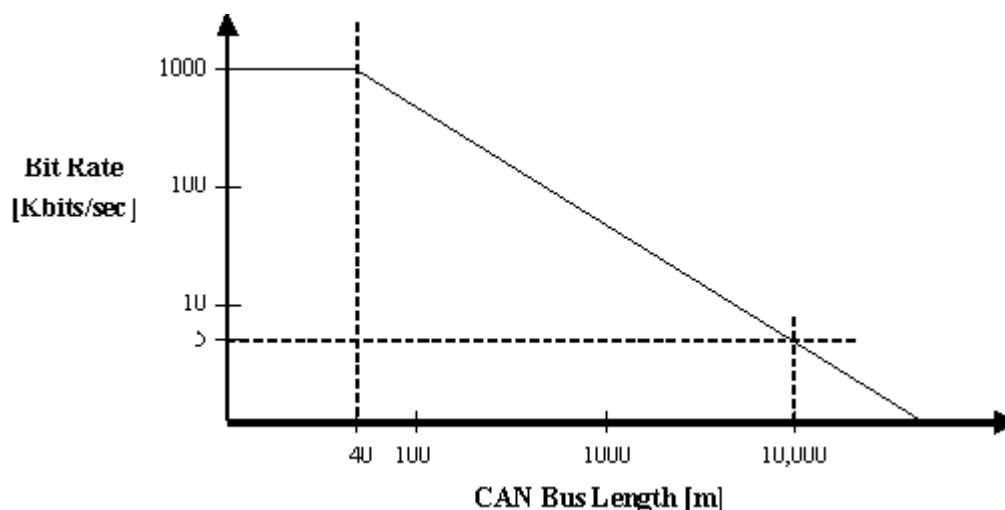
CAN 错误检测

CAN 协议有五种错误检测的方法；三个是报文级的而两个是位级的。如果一个报文出错，那么错误检测的任何一个方法使节点不接收这个报文并产生一个出错帧，使所有的帧都忽略它并使发送节点重新发送这个报文。

在报文级检查中，有 CRC 检查和应答隙。CRC 检查是一个 15 位 CRC，它计算描述符场和数据字节的 CRC。应答场有两位，包括一个应答位和一个应答界定符。这个发送器将会把一个隐性位放在应答场。任何一个正确接收报文的节点在应答场写一个显性位。如果发送器在应答场没有读回一个显形位，它将产生一个出错帧并重新传送报文。最后，在报文级还有一个形式检查。它检查那些总是隐性位的报文场。如果检测到显形位，就会产生错误。它检查：帧起始、帧结束、应答界定符以及 CRC 界定符位。

在位级检查中，每一个位都由发送器监控。如果一个位被写进总线但读到的是它的反，错误就会产生。只有标识符场用于仲裁和应答隙是除外的，它要求显性位覆盖隐性位。最后的一种错误检测方法是通过位填充规则。当一个报文没有被填充，即如果在逻辑电平相同的连续 5 位后，下一位不是前面的反，则产生一个错误。

活动错误帧包括六个显形位，它们违背了位填充规则，所有的 CAN 节点都认为它是一个错误并产生自己的错误帧，所以错误帧的长度可以在 6 位和 12 位之间。错误帧后是 8 位隐性位界定符场，而总线在重发被破坏的报文前是空闲的。要注意报文在被成功接收之前仍要争取仲裁。



CAN 最初是设计用于总线长度大约为 5—10 米的汽车系统。但是自从 CAN 发现它要在很多要求增长总线长度的系统中使用后，其最大的传送范围不断地增大。标准的驱动器能够驱动长度高达 1 公里的总线。上述的图表描述了理想的期望传输速率和传输长度的关系。

CAN 支持工具

虽然 CAN 相对比较年轻，但是它已经被大量的开发工具支持。这些工具从简单开发板到大规模的 CAN 分析仪都有。位于德国曼兹的 PHYTEC 公司在 P87C591 和 Philips XA 结构的基础上生产出一系列配备 CAN 的微控制器板。它包含了微控制器和外围 CAN 控制器（Philips P87C591）或者单一的 CAN 微控制器（SJA1000，P82C200）。这些板能够使用 C 语言或者汇编语言来编程，并且能使用先进的调试器譬如 Hitex 的 HiTOP。PHYTEC 也提供小型的 SLIO 模块。

位于慕尼黑的 IXXAT 公司生产系列的 CAN 工具和接口。它们包括基于普通 PC 卡或者小型 PCMCIA 模块的 CAN-PCMCIA 和 VME 插槽，这些对于 CAN 的调试非常有用。

由于如此紧密地与微控制器相联系，现存的工具例如“在电路仿真”提供非常有用的功能，譬如无论是在片或者不在片的状态下，都能实时监控 CAN 控制器的输入和输出数据。