

带时间触发的 CAN 网络设计

饶运涛

(东华理工大学电子与机械工程学院, 抚州 344000)

摘 要:为解决 CAN 总线系统中事件触发报文发送时, 固定优先级仲裁机制所引起的发送时间不确定, 低优先级报文可能被延误的问题, 根据时间触发 CAN 的基本原理和结构特征, 结合通用 CAN 控制器 SJA1000, 举例介绍如何组建一个 TTCAN 实验系统的基本步骤和算法。该设计可以确保实时和安全性要求较高的自控系统的正常运行。

关键词:带时间触发的 CAN; 时间基准; 时间主机; 基本循环

Design of Time Triggered CAN Network

RAO Yun-tao

(College of Electronic & Mechanical Engineering, East China Institute of Technology, Fuzhou 344000)

【Abstract】 With the intention of solving a problem that a message with low priority may be delayed over in CAN system sometimes, its cause is the principle for event triggering communication with fixed arbitration priority of message on CAN bus. It is introduced by an example that how to design a Time Triggered CAN(TTCAN) basic system and an arithmetic, according to TTCAN principium. Such design can ensure normal operation in autocontrol systems which demands more instant ability and higher security, but it sacrifices the flexibility during configuring or altering a CAN system.

【Key words】 Time Triggered CAN(TTCAN); time base; time master; base cycle

1 概述^[1]

现有的 CAN 国际标准 ISO 11898 规定了 CAN 网络的物理层和数据链路层的协议, 其中的介质访问控制子层(MAC)采用位仲裁的方式来解决节点之间访问总线时的冲突。也就是说, 在一个多主系统中, 任何节点在确认总线空闲时都可以随机向总线发送数据, 如果多点冲突, 则由各自报文的标识符优先级来裁定哪个节点赢得总线使用权, 这称为“事件触发”方式(每个节点发送报文是由它本身要求的事件引起)。由此可以看到, 只有优先级最高的报文发送时间有最好的确定性; 在总线传输的高峰时期, 报文优先级越低, 其发送时间就越不能确定。这种方式在节点访问的时间规律性很强、实时要求严格的控制系统中就会出问题。

解决上述问题的一个方案是“时间触发 CAN”(Time Triggered CAN, TTCAN)模式。其基本原理是: 把时间划分成一个个间隙(窗口), 每个节点在规定间隙内发送数据; 然而, TTCAN 并不改变现有 CAN 的物理层和数据链路层的协议。

2 TTCAN系统的结构特征^[1-2]

由 TTCAN 的基本原理可以想到, 它应当是一个主从结构的系统。这是因为把时间划分成一个个定时的间隙, 必定要有一个节点来定时发布时间基准(time bases)信号, 其他节点都必须把本地的时钟(Local Time)与该时间基准同步, 同时要知道, 自己是在什么时刻开始的时间窗口内发送报文。那么, 时间基准信号的周期如何确定? 每个时间间隙的长度是多少? 如何分配? 还有, 一旦主节点停止工作了, 该如何应对? 等等。这些问题都需要事先经过周密的策划, 一经确定后, 就不能随意更改。

3 时间主机与循环周期^[1-2]

在 TTCAN 网络中, 节点同步的时间基准是所谓的参照

报文, 它由一个特定的节点定期发送, 这就是时间主机。参照报文是一个 CAN 数据帧, 其特征在于它的标识符。有效的参照报文同时被所有节点识别。每一个有效参照报文启动了一个新的基本循环并且引起了每一个节点的循环时间复位。在 TTCAN 中规定了时间间隔度量 NTU(网络时间单元), 每一个 NTU 使节点的计数器加 1。一个参照报文启动一个循环周期, 如图 1 所示。

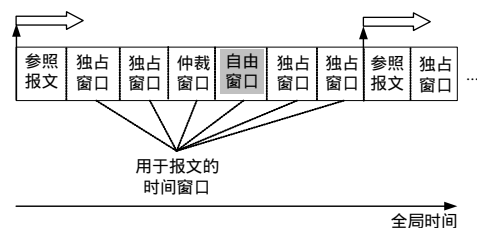


图 1 一个参照报文启动一次 TTCAN 基本循环示意图

一个基本循环内包含的各时间窗口可以大小不同, 它提供传输不同报文所需要的时间。这里有 3 种类型的时间窗口:

(1)独占时间窗口。它是专门为定期报文保留的。在独占时间窗口中, 窗口的开始点决定了一个节点预先确定的报文的发送时刻。如果一个系统有完整的规划, 同时离线设计工具也分析了这个通信模式, 那么就不会发生冲突; 即使发生冲突状态, CAN 协议的特性(位仲裁, 只有当总线空闲时才发送)也能起到作用。这个系统的工程人员必须离线决定哪个报文在哪个独占时间窗口一定发送。为了给系统设计者提供较高的灵活性, 在基本循环中一个独占窗口可重复多次。独

作者简介:饶运涛(1964 -), 男, 教授, 主研方向: 嵌入式系统, 现场总线

收稿日期:2007-12-25 **E-mail:** ryt_163@163.com

占时间窗口不允许 CAN 报文的自动重复传输,这里保证了在独占时间窗口的报文不被其他总线运作延时。

(2)仲裁时间窗口。用于自发报文的时间窗口称之为仲裁时间窗口。在这样的窗口内,多个报文通过 CAN 的无破坏仲裁机制竞争总线;位仲裁决定了哪个节点的哪个报文能成功上总线。在设计时允许在一个仲裁窗口安排多个报文。所以,应用程序在运行时可以决定,是否使用仲裁窗口来发送一个报文以及哪一个报文将在某个仲裁窗口发送。仲裁时间窗口也同样不允许 CAN 报文的自动重发。

(3)自由时间窗口。在设计期间也可以为将来的网络扩展保留一些自由时间窗口。当新的节点通信需要更大的空间或现有节点的带宽需要扩展时,它们可以变成仲裁或独占时间窗口。

4 TTCAN中节点的特定知识^[2]

在 TTCAN 中一个节点的网络控制器为了运行,它只要知道那些为了本节点发送报文和检查被接收的报文是否按时到达而定义的时间间隙所需要的时间标记就足够。如图 2 所示,控制器(节点 4)在独占窗口 2 和窗口 6 发送报文 C,在仲裁窗口 3 发送自发报文 F。这个控制器只对独占窗口 1 接收报文 A 感兴趣。与其他严格的时间触发通信系统相比,在 TTCAN 中节点特定的知识量是最少的。

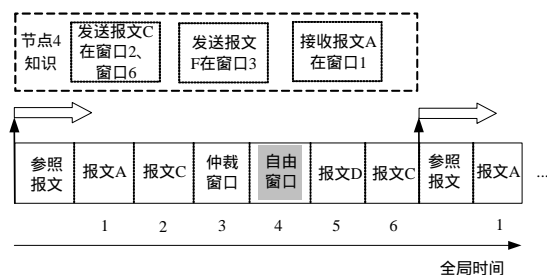


图 2 TTCAN 通信-TTCAN 控制器的本地信息

5 系统矩阵^[2]

实际应用程序往往包含不同周期的许多控制循环和任务。对于它们的信息,都需要有各自的发送模式。单一的基本循环不能提供足够的灵活性来满足这种要求。TTCAN 规范允许使用多个基本循环组建系统工程人员所要求的通信矩阵或系统矩阵。几个基本循环连接起来组成一个矩阵循环。在矩阵中多种模式都有可能包含,如图 3 所示。

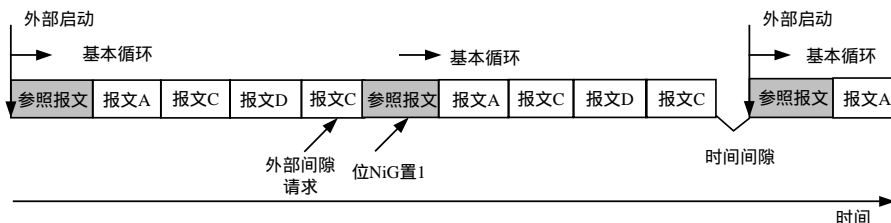


图 4 时间间隙和事件同步的基本循环机制

TTCAN 还要考虑其他一些问题,如全局时间与时间漂移的补偿、时间触发器的类型和构成、NTU 的精度设置等。

8 一个简单TTCAN基本循环的实现^[2-3]

由上述可以看出,TTCAN 是在未变的标准 CAN 协议之上的一个高层协议,它同步一个网络中的所有 CAN 节点的通信时间表并且提供了一个全局系统时间。当节点被同步时,报文都可以在各自独占的时间窗口发送出去,而没有与其他报文对总线的竞争。这样就预先知道通信的确切时间,可以

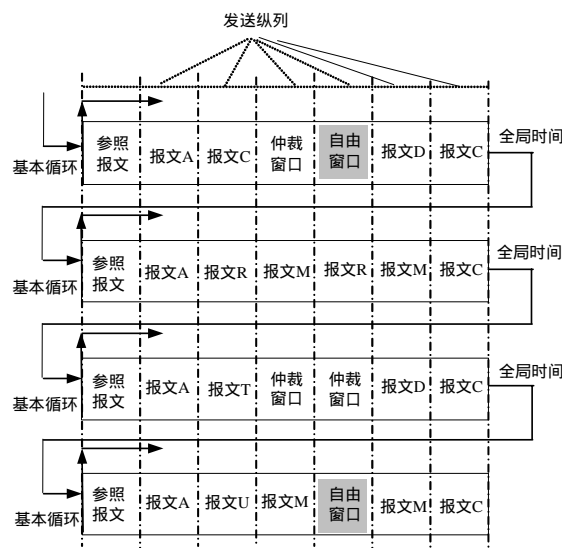


图 3 TTCAN 系统中的矩阵循环

6 时间主机的故障误差机制^[2]

由于在 TTCAN 模式中时间主机起着至关重要的作用,因此必须建立一套故障-误差机制。通过预先定义某 TTCAN 控制器作为备用的时间主机。它有对应的标识符。在 TTCAN 控制器复位后,备用时间主机检测在总线上是否已经有通信,是否已经有参照报文发送;如果没有,这个备用时间主机就发送带有它标识符的参照报文,然而不论什么时候,只要接收到优先级更高的参照报文,这个备用时间主机就停止发送参照报文,并且同步于高优先级时间主机给定的基本循环。

7 事件同步的循环时间^[2]

对于某些应用,多数时候一个严格周期安排是有必要的,但是有时要求能够中断这个时间表,又可通过应用程序异步重新启动它。TTCAN 就有这样的机制。在一个应用程序运行期间,主机想要中断这个周期安排表,设计者使能这个特征,时间主机在下一个参照报文中使“下一个是间隙位”(NiG)置 1,在这个基本循环结束后就会出现一个间隙。这样其他节点在这个基本循环结束后将不会把这个缺失的参照报文解释为一个错误,特别是备份的时间主机不会发送参照报文。如果这个时间主机的应用程序想要重启周期安排表,那么它开始发送同步于一个相应事件的参照报文,如图 4 所示。

避免仲裁的时间损失。现有的 CAN 控制器可以接收 TTCAN 网络的每一个报文,TTCAN 控制器可以运行在现有的 CAN 网络中。为了把一个现有的 CAN 控制器提升到时间触发运行,所要求的最少附加硬件是一个局部时间基准和捕获这个时间基准的机制,这个捕获由总线传输的信息来触发。基于已存在于某些 CAN 控制器中的硬件,就可能在软件上实现 TTCAN 级别 1 的 TTCAN 控制器的功能。TTCAN 级别 2 的 TTCAN 控制器的功能提供了全部的 TTCAN 特征,诸如全局

时间、时间标记中断和时间基准同步等必须在硅片中实现。级别 2 的时间精度比级别 1 高,而且消耗 CPU 的资源少。

这里以 SJA1000 构成的 CAN 节点为例,说明实现 TTCAN 级别 1 的基本步骤:

(1)总线上的节点数 $N=64$ (考虑到总线收发器的负载能力),假设波特率为 80 Kb/s。

(2)设置一个节点用作主机,任务是发布全局时间参照报文,另一个节点作为备用主机,当主机故障时,由它发布全局时间参照报文。

(3)各节点工作在扩展模式下,采用双滤波配置。其中,第 2 道滤波器根据各节点 ID 设置自己的 ACR2 和 ACR3,设 AMR2 和 AMR3 为全 0(对 ID 的每一位都验收);而各节点的第 1 道滤波器 ACR0 和 ACR1 设置为全 1,AMR0 和 AMR1 组成的 16 位中除最低位为 1 外其余全为 0,这是因为要使主机全局时间标记的报文 ID 优先级最高(全 1),备用主机 ID 次之(最低位为 0,其余为 1);保证它不因总线位仲裁而延时。由此也可以看到,全局时间报文可以同时被所有节点接收到,保证了各节点局部时间的同步。

(4)根据系统功能要求制定主机定时周期,即基本循环周期。例如,现共有 9 个工作节点,每个占用总线时间(间隙)为 1.5 ms(假设都一样)。当然也可以根据需要配置仲裁窗口和自由窗口(如图 1、图 3 所示),每个也占用时间 1.5 ms。整个时序安排如图 5 和表 1 所示,一个基本循环周期是 18 ms。

参 照	报 文 1	报 文 2	自 由	报 文 3	报 文 4	仲 裁	报 文 5	报 文 6	报 文 7	自 由	报 文 8	报 文 9
--------	-------------	-------------	--------	-------------	-------------	--------	-------------	-------------	-------------	--------	-------------	-------------

图 5 该 TTCAN 系统报文时序安排

表 1 该 TTCAN 系统中各节点参数设置

节点号	主机	备主机	节点 1	节点 2	节点 3	...	节点 6	节点 7	节点 8	节点 9
定时/ms	18.0	18.0	0.0	1.5	4.5	...	10.5	12.0	15.0	16.5
ACR0,1	FFFFH	FFFFH	FFFFH	FFFFH	FFFFH	...	FFFFH	FFFFH	FFFFH	FFFFH
AMR0,1	0001H	0000H	0001H	0001H	0001H	...	0001H	0001H	0001H	0001H
ACR2,3	本身 ID	本身 ID	本身 ID	本身 ID	本身 ID	...	本身 ID	本身 ID	本身 ID	本身 ID
AMR2,3	0000H	0000H	0000H	0000H	0000H	...	0000H	0000H	0000H	0000H

(上接第 209 页)

这把“密钥”才能正确还原;而对于非法访问者,虽然可以进行解码,但是由于缺少这把“密钥”,因此其解码的图像只是杂乱的数据,这使增加了混沌置乱预处理的分布式系统具有一定的安全特性。

6 结束语

为了解决分布式视频编码技术中的突发差错,提出了交织预处理和混沌置乱预处理 2 种解决方法。对于不同的视频系列,最佳的交织器大小很难确定,所以,交织预处理技术不能有效解决突发差错,而混沌置乱预处理通过混沌映射置乱图像,使得差错均匀地分布在置乱后的图像中。再结合传统的分布式视频编码技术就能有效地实现视频编码。实验结果表明,混沌置乱预处理能够提高分布式视频编码的性能,方法实现复杂度低,且可以应用于某些有安全要求的领域。

参考文献

[1] Rane S, Aaron A, Girod B. Systematic Lossy Forward Error

(5)根据系统各节点工作时序,分配好各节点的时间窗口顺序和长度,并且各节点在软件中只要“记住”自己的起始时刻和窗口长度(如图 2、图 5 所示)。在本例中,每当接到时间参照报文,节点 6 就把本身的定时器初值设置为 10.5 ms,使能该中断;当定时中断发生时,就可以发送自身的报文,关闭该中断。这里节点 1 的窗口开始于接到参考报文时刻。

(6)备用主机的设置:当备用主机不能按时接收到主机发送的时间标记报文时,就认定主机出故障,并且替代主机发送该类报文,其 ID 如第(3)步所设,优先级仅低于主机。一旦主机恢复工作,由于它的 ID 优先级最高,因此备用机在接收到后,就停止替代工作。

(7)为使问题简化,这里暂不考虑时间漂移补偿和 NTU 等问题。

9 结束语

尽管 TTCAN 还在完善中,一些厂商已陆续推出带 TTCAN 功能的芯片,例如 ATMEL 的带 CAN 控制器的单片机 AT89C51CC0X, MICROCHIP 出品的独立 CAN 控制器 MCP2515 等,它们都有支持 TTCAN 的相应硬件单元,这使得设计该类系统简单,时间精度也得以提高。当然,可以看到组建 TTCAN 系统比较复杂,而且也失去 CAN 系统原有的机动灵活性。近几年,在汽车控制系统中的 CAN 正在向 TTCAN 发展,笔者的实验系统设计也在不断完善中。可以预见,新兴的 TTCAN 将使 CAN 总线的生命力更强,使用的范围更广。

参考文献

[1] Leen G, Heffernan D. TTCAN: A New Time-triggered Controller Area Network[J]. Microprocessors and Microsystems, 2002, 26(2): 77-94.
[2] Hartwich F, Müller B, Führer T, et al. CAN Network with Time Triggered Communication[C]//Proc. of the 7th International CAN Conference. Amsterdam, Netherlands: [s. n.], 2000.
[3] 饶运涛, 邹继军, 郑勇芸. 现场总线 CAN 原理与应用技术[M]. 北京: 北京航空航天大学出版社, 2002.

Protection for Error Resilient Digital Video Broadcasting[C]//Proc. of SPIE Visual Communications and Image Processing Conf.. San Jose, CA, USA: [s. n.], 2004.

[2] Pradhan S S, Ramchandran K. Distributed Source Coding Using Syndromes: Design and Construction[J]. IEEE Transactions on Information Theory, 2003, 49(3): 626-643.

[3] Aaron A, Rane S, Zhang Rui, et al. Wyner-Ziv Coding for Video: Applications to Compression and Error Resilience[C]//Proc. of IEEE Data Compression Conference. Utah, USA: [s. n.], 2003: 93-102.

[4] Stankovic V, Liveris A D, Xiong Zixiang, et al. Design of Slepian-Wolf Codes by Channel Code Partitioning[C]//Proc. of IEEE Data Compression Conf.. Utah, USA: [s. n.], 2004: 302-311.

[5] Girod B, Aaron A M, Rane S, et al. Distributed Video Coding[J]. Proceedings of the IEEE, 2005, 93(1): 71-83.