

单位代码: 10359
学 号: 201011100574

分 类 号: TP393
密 级:



合肥工业大学
Hefei University of Technology

硕士学位论文

MASTER DEGREE THESIS

论文题目: 基于 AUTOSAR 标准的 CAN 网络管理研究

学位类别: 学 历 硕 士

学科专业: 计算机应用技术
(工程领域)

作者姓名: 于 萍

导师姓名: 张建军 教授

完成时间: 2013 年 04 月



基于 AUTOSAR 标准的 CAN 网络管理研究

Research of CAN network management based on AUTOSAR

作 者 姓 名 _____ 于 萍 _____

学 位 类 型 _____ 学 历 硕 士 _____

学 科、专 业 _____ 计算机应用技术 _____

研 究 方 向 _____ 计算机控制技术 _____

导 师 及 职 称 _____ 张建军 教授 _____

2013 年 4 月

合 肥 工 业 大 学

本论文经答辩委员会全体委员审查，确认符合合肥工业大学硕士学位论文质量要求。

答辩委员会签名：（工作单位、职称）

主 席： 陈孝宁 安徽大学 教授

委 员：
中科院合肥研究院 研究员 李阳
中科院电子学研究所 教授 李中
合肥工业大学 教授 曹路平
合肥工业大学 教授 王德
导 师： 张 迪 宇

基于 AUTOSAR 标准的 CAN 网络管理研究

摘 要

随着人们对车辆的功能需求和安全需求日益增多，越来越多的电子控制单元被引入到汽车中，引起了汽车控制方面的技术变革。将汽车作为一个整体的控制思想正引领者汽车向网络化方向发展。这不仅给汽车厂商带来挑战，同时也对网络通信的安全性和可靠性提出新的要求。汽车网络开放式架构标准 AUTOSAR 提供了统一的开发平台和规范标准，这不但降低了制造成本，而且对缩短我国同国外汽车电子技术水平的差距，提高自身的核心竞争力，加速其产业化具有积极的探索作用和促进意义。本文对国内外的大量研究资料和结果进行收集和分析，并在此基础上对 AUTOSAR 标准下的网络管理方法进行研究和改进，并基于 AUTOSAR 软件架构对 CAN 总线的网络管理进行实现。

论文首先介绍了汽车网络管理的背景和意义，分析国内外研究现状。然后研究了 OSEK 网络管理和 AUTOSAR 网络管理两种方法的基本工作原理，在此基础上，针对 AUTOSAR 网络管理功能在睡眠过程、错误检测等方面的不足，提出一种改进的 AUTOSAR 网络管理方法，并基于 AUTOSAR 软件规范，给出了 CAN 总线下该网络管理方法的实现。最后，通过 CANoe 软件对 CAN 网络管理方案进行仿真实验，验证了该方案的可行性。

关键词: CAN，网络管理，汽车总线，AUTOSAR

Research of CAN network management based on AUTOSAR

ABSTRACT

With the increasing functional requirements and security requirements in vehicles, more and more electronic control units are introduced into cars, which causes a technical change in vehicle control aspect. The idea of taking a car as a whole unit is leading a direction of networked. This is a challenge not only to the automobile manufacturers, but also make new demands on the safety and reliability of the network communication. The open architecture standard AUTOSAR automotive network protocol provides a unified development platform and standards, this will not only reduce the manufacturing costs, but also shorten the gap of automotive electronic technology between our country and foreign countries, improve our core competitiveness and accelerate the industrialization of it. In this thesis, a large number of research domestic and foreign are collected and analyzed, and on this basis, a network management method with AUTOSAR standard is researched and improved, then the improved method on CAN bus with AUTOSAR software architecture is implemented.

The thesis first describes the background and significance of the vehicle network management, analyzes the network management research status both at home and abroad. Then, analyzes the basic working principle of the OSEK network management and AUTOSAR network management, on this basis, directing at the insufficient of AUTOSAR network management function on sleeping and error detection, a method to improve the AUTOSAR network management is put forward in this thesis. On account of AUTOSAR software standard, provides the realization of network management method under CAN bus. At last, a simulation experiments about CAN bus network management solution is taken in CANoe, and the results verify that the solution is feasible.

Key words: CAN, network management, vehicle bus, AUTOSAR

致 谢

从 2010 年 9 月份踏入合肥工业大学的校园到今天,两年半的研究生学习即将画上一个句号。分布式控制研究所承载了我所有的美好回忆。值此论文完成之际,我首先感谢我的导师张建军教授。与张老师交流的过程中,张老师和蔼可亲、平易近人的性格给我留下非常深刻的印象,他那严谨的治学态度和认真负责的工作态度更是让我倍受鼓舞。张老师不但教会了我如何做好科研工作,更教会了我如何做人做事。

本项研究课题在选取和研究的过程中都得到了张本宏老师、王跃飞老师的悉心指导。两位老师帮我解答困惑,指点迷津的同时还帮我拓宽研究思路。他们认真的作风,求实的态度,踏实的精神,渊博的知识将影响和鞭策我一生。对两位老师的感激溢于言表。

感谢实验室的张利老师以及分布式控制研究所的各位老师。在实验室的日子里,老师们不仅在学业上对我给予精心指导,而且还在思想和生活上给我无微不至的关怀,在此我向各位老师致以崇高的敬意和衷心的感谢。

感谢我的同学田晓、侯亮、陆克华、陈牡丹、孙峰、杨永军、杨静荣、李亚娟等在我上研究生期间,对我的学习和生活上给予的关心、帮助和支持。

同时还要感谢我所有的亲人和朋友们,因为有你们的陪伴和鼓励,我才可以无所畏惧;因为有你们的信任和支持,我才可以勇往直前;因为有你们关心和照顾,我才可以幸福的生活。

最后,感谢所有帮助和支持我的人。

作者:于萍

2013 年 4 月

目录

第一章 绪论	1
1.1 课题研究背景及意义	1
1.2 国内外研究现状	3
1.2.1 汽车总线研究现状	3
1.2.2 网络管理研究现状	4
1.3 课题来源	5
1.4 研究内容和结构	5
第二章 车载网络技术基础	6
2.1 汽车网络简介	6
2.2 总线协议介绍	6
2.2.1 LIN 总线	6
2.2.2 CAN 总线	7
2.2.3 Flexray 总线	10
2.3 本章小结	13
第三章 汽车网络管理研究	14
3.1 OSEK/VDX 网络管理体系	14
3.1.1 OSEK/VDX 简述	14
3.1.2 直接网络管理	14
3.1.3 间接网络管理	17
3.2 AUTOSAR 网络管理体系	18
3.2.1 AUTOSAR 基础软件架构	18
3.2.2 AUTOSAR 网络管理	19
3.2.3 网络管理开发流程	20
3.3 一种改进的 AUTOSAR 网络管理方法	22
3.3.1 节点状态设计	22
3.3.2 状态转换设计	23
3.3.3 节点故障检测设计	25
3.3.4 协同睡眠设计	26
第四章 CAN 网络管理实现	28
4.1 总体设计	28
4.2 网络管理接口模块设计	29
4.3 网络管理模块设计	30
4.3.1 CAN 网络管理消息设计	30
4.3.2 错误处理	31

4.4 通信管理模块设计	32
4.4.1 设计目的	32
4.4.2 通信管理状态机设计	32
4.4.3 通信管理状态机转换	33
4.5 状态管理模块设计	34
4.6 CAN 接口和驱动设计	35
4.6.1 CAN 接口设计	35
4.6.2 CAN 驱动设计	36
4.7 本章小结	37
第五章 系统仿真与测试	38
5.1 仿真系统简介	38
5.1.1 CANoe 组成	38
5.1.2 CANoe 基本操作	39
5.2 仿真系统架构	40
5.3 仿真系统设计	41
5.3.1 数据库建立	41
5.3.2 软件设计	41
5.4 仿真过程及结果	42
5.4.1 节点故障检测	42
5.4.2 总线故障	43
5.4.3 协同睡眠	44
5.5 本章小结	44
第六章 总结与展望	45
6.1 工作总结	45
6.2 进一步工作	45
参考文献	47
攻读硕士学位期间发表的论文	50
读研期间参与的科研工作	50
特别声明	51

插图清单

图 2-1 CAN 数据帧结构	9
图 2-2 CAN 2.0 分层结构	10
图 2-3 Flexray 的通信时间周期	11
图 2-4 Flexray 数据帧格式	12
图 2-5 消息 ID	13
图 2-6 网络管理向量	13
图 3-1 OSEK 体系结构图	14
图 3-2 逻辑环结构	15
图 3-3 直接网络管理的逻辑状态转换	16
图 3-4 直接网络管理睡眠协商机制	17
图 3-5 间接 NM 状态图	18
图 3-6 AUTOSAR 基础软件结构	19
图 3-7 AUTOSAR 状态转换	20
图 3-8 系统设计和开发流程	21
图 3-9 状态转换设计	24
图 3-10 故障节点检测流程	25
图 3-11 协同睡眠过程	27
图 4-1 CAN 网络管理基本结构	28
图 4-2 网络管理接口	29
图 4-3 协同算法	30
图 4-4 通信状态机转换	33
图 4-5 状态管理自动机	35
图 5-1 CANoe 界面	39
图 5-2 仿真及其拓扑结构	40
图 5-3 CANdb++ Editor 工具	41
图 5-4 NMNormal 状态接收过程图	42
图 5-5 通信正常	43
图 5-6 节点故障通信	43
图 5-7 总线故障通信	43
图 5-8 节点同步睡眠	44

表格清单

表 3-1 基础软件层结构和功能	19
表 4-1 NM 协议数据单元设计	31
表 4-2 Control Bit Vector 结构设计	31
表 5-1 定时器设计	40

第一章 绪论

1.1 课题研究背景及意义

随着人们对汽车安全、节能和环保等性能要求的不断提高，汽车中采用的电子设备越来越多，且分布范围越来越广。从行驶、制动、转向系统到安全保证系统、仪表报警系统，从发动机系统到传动系统，电子设备渐渐形成了一个庞大复杂的控制系统^[1]。汽车总线技术是车上信息和控制系统的支撑技术，是现代汽车电子技术发展的必然趋势之一，它不仅解决了在汽车电子化过程中出现的线路复杂和线束增加等问题，而且还实现了控制系统间信息和资源的共享。随着时代的发展，汽车总线技术在现代汽车电子技术中的地位越来越重要。

汽车总线技术是指汽车内部采用总线进行控制的一种技术，它是一种串行的数据通讯协议，主要负责控制器件和与测试器件之间的数据交换。汽车内部的各个电子控制单元（ECU）通过相应的总线控制协议互连成一个汽车内部局域网。在该局域网中，节点根据自身的状态信息以及总线上的通信信息完成相应的控制功能和动作^[2]。如汽车上的灯光控制、车门控制、电机启停控制等。汽车总线采用非破坏性的总线性仲裁机制，具有数据帧结构短、通讯方式灵活等特点，满足了汽车在实时性和可靠性方面的要求。汽车总线是车载电子设备交换信息的载体，是汽车中的微控制器通讯设备，它将汽车电子控制器件组成一个网络。一个节点是由 MCU、接口电路、总线控制器、总线驱动器等设备构成的。

目前有众多的总线协议适用于汽车网络，它们都有各自的优点和不足。通常情况下，一种总线协议往往不能同时满足所有类型和档次汽车的需要。考虑到它们不同的数据传输能力、容错能力、稳定性和成本，在一辆汽车内部，往往采用多种不同的网络结构。因此可以说，汽车网络是一个分层式的复合网络结构。利用多种总线配合运行，能够制造出性能可靠、价格实惠的车型，有效提高汽车厂商的竞争力^[3]。

目前，绝大多数车用总线都被美国汽车工程师协会 SAE 下属的汽车网络委员会按照协议特性分为四类^[4]。

A 类总线的位传输速率一般小于 20KB/S，适应于面向传感器或执行器管理的低速网络。A 类总线以最有前途的 LIN（Local Interconnect Network）总线作为代表。LIN 总线协议是一种新型的低成本的开放式串行通讯协议，由摩托罗拉 Motorola 与奥迪 Audi 等知名企业联手推出的，在汽车内部的分布式电子单元控系统和传感器或执行器中的智能数字化通讯中得到应用^[5]。

B 类总线的位速度一般在 10~125KB/S 之间，面向中速网络，主要负责独立控制模块之间的信息共享。B 类总线以控制器局域网 CAN（Controller Area Network）最为著名。CAN 网络是由 BOSCH 公司开发的，服务于欧洲汽车市

场，主要负责汽车内部执行部件间的数据通讯。经过技术和功能的不断发展和完善，控制器局域网 CAN 的国际标准(ISO11898-1)在 1993 年正式被 ISO 命名为道路交通运输工具——数字信息交换高速通讯标准。B 类总线在车身的舒适性模块和显示仪表等设备中得到广泛使用。近几年，欧洲的一些车型中也开始广泛使用低速容错 CAN 标准 ISO 11519-2。CAN 是控制器局域网(Controller Area Network)的简称，它是汽车网络体系结构中最重要的一部分。与过去的汽车总线系统相比，CAN 具有传输速度快、容错能力强、性价比高等优点，一直受到广大汽车厂商的青睐^[6]。

C 类总线位速率多在 125KB/S ~ 1MB/S 之间，是一类面向闭环实时控制的多路传输高速网络，在实时性要求较高的通讯系统中得到大量使用，例如车上的动力传动系统。在欧洲，高速 CAN 即 ISO11898-1 被众多的汽车厂商作为 C 类总线使用。美国在建筑机械和课程、卡车、农业动力设备、拖车中则大量使用 SAEJ1939 通讯协议。

D 类总线的位速率一般在 2MB/S 以上，主要面向高速数据流传输的高性能网络以及多媒体设备中，如液晶显示设备和 CD 播放机等。Flexray 是新一代的总线通信标准，它适用于车载电子中的高速网络部分。Flexray 总线的数据传输速度可以达到 10Mbps，远超过 CAN 总线的传输速度。另外，它还支持总线型、星型、混合型等多种网络拓扑结构，并且能够进行双通道通信，可扩展性强，容错率也更加出色，可以满足高端汽车对实时性和安全性要求较高的需要。例如，应用在转向装置、自动变速装置、动力装置等^[7-9]。Flexray 总线价格昂贵，一般不会大规模使用。

随着汽车电子系统网络化进程的加快，许多新型的总线技术被研发出来，如故障诊断总线技术、安全总线技术、X-by-wire 总线技术等。由于各种原因这些新研发的总线未能被 SAE 收录，但是其重要性不言而喻，我们称之为专用总线^[10]。专用总线在汽车网络中也发挥重要作用，本文在此不做赘述。

汽车电子产业发展前景广阔。近二三十年来，随着电子信息技术的快速发展和汽车制造业的不断变革，汽车电子技术的应用和创新极大地推动了汽车工业的进步与发展，对提高汽车的动力性、经济性、安全性，改善汽车行驶稳定性、舒适性，降低汽车排放污染、燃料消耗起到了非常关键的作用，同时也使汽车具备了娱乐、办公和通信等丰富功能。国际汽车专家指出，近 10 年来汽车产业 70% 的创新来源于汽车电子技术及其产品的开发应用，汽车电子技术的应用水平已成为衡量汽车档次水平的主要标志，其应用程度的提高是汽车生产企业提高市场竞争力的重要手段^[11-12]。

汽车电子技术已成为现代汽车技术的核心技术，汽车电子产业发展水平对一个国家汽车工业的市场竞争力有着举足轻重的影响。现代汽车电子集电子技术、汽车技术、信息技术、计算机技术和网络技术等于一体，包括基础技术层、

电控系统层和人车环境交互层三个层面，经历了分立电子元器件控制、部件独立控制及智能化、网络化集成控制应用三个发展阶段。目前汽车电子产品可以分为电子控制系统和车载电子装置两大类，其中电子控制系统的性能直接决定着汽车整车的性能。

随着汽车电子技术的发展，越来越多的 ECU 控制单元被应用在汽车中，网络总线系统作为汽车控制网络平台，所有的 ECU 控制单元和车载电器都将逐步搭载到汽车网络平台上，以达到数据信息共享、实现全车智能化控制。在汽车朝着综合集成控制发展的趋势下，车载网络总线系统作为汽车全车控制网络及通信平台，对汽车全车通信、智能化控制及提升整车性能、安全性、操控性愈加至关重要^[13]。

1.2 国内外研究现状

1.2.1 汽车总线研究现状

上世纪 90 年代初，汽车中开始应用总线技术。如今，绝大多数的汽车控制都采用 CAN/LIN 总线控制技术。总线技术已经成为衡量现代汽车的重要标准。与此同时，总线零部件也成为汽车零部件发展的主流。我国汽车制造水平和零部件的发展水平与国外存在巨大差距：国内零部件配套体系尤其是智能零部件研发存在的不足，导致了国内自主研发的具有总线接口的零部件数量稀少^[14]。国内自主研发和生产了部分小批量的、使用 CAN/LIN 总线进行仪表控制、车灯控制、车门控制等功能的汽车，如北京市场上达到国 III 标准的扫雪车、公共汽车等。目前大批量生产的汽车中，具有 CAN/LIN 总线的零部件由国内研发的很少，几乎都是国外研发的。

总线技术并不是一项独立的技术，而是以总线零部件的智能化为基础的一种在电器之间进行信息交互的接口。我们从总线的应用概念到生产和维护等方面都还处于起步阶段，很多企业对总线的认识缺乏，只停留在节约电线和增加功能上，并没有上升到平台和电气标准的层次。我们在汽车零部件的智能化方面与国际先进水平的差距依然存在，而且相差很大，总线技术的发展受其制约。最近几年，我们在总线技术的研发方面与国际先进水平的差距正在缩小，但是对智能零部件的研发水平较低导致总线与零部件研发相分离，使得成功研发出来的总线产品没有优势，更谈不上产业化^[15]。

目前，我们的研发工作还处于向外国学习的阶段。由于国内在这方面的研究起步晚，研究的主力主要集中在高校和研究所。虽然目前对 CAN 和 FlexRay 取得了富有成效的研究，但这些研究目前没有投入到实际生产应用过程中，大多还停留在理论研究阶段。与此同时，很多国外文献对总线技术的研究只突出优点，对缺点避之不谈，没有全面的对总线技术进行论述，结果使相关的研发人员研发进度缓慢，突破技术封锁是我们面临的巨大挑战。

1.2.2 网络管理研究现状

汽车网络管理在汽车中的应用越来越普遍，目前主要有两种网络管理策略得到广泛的研究和使用，一个是 OSEK/VDX 规范的网络管理标准，另一个是在此基础上发展的开放式的软件架构平台，AUTOSAR 标准。

OSEK 标准是应用在静态实时操作系统上的标准，仅限于提供 API 接口和架构。OSEK 标准是由主要的汽车厂商、制造商、研究机构和开发商发起，通过静态配置本地节点的初始状态、网络参数初始值，利用网络管理消息的收发实现网络的动态监控。这样，就可以用一些简单的处理器代替其他更加昂贵的解决方案，控制任务执行^[16]。

OSEK 虽然提供了统一的架构和接口，但是并没有具体的方案来实施。它只提出了一个逻辑环模式和使用规则，但是并没有对网络上传输的各种网络管理消息的具体格式和传输规则进行规定，这在很大程度上造成了汽车电子制造商和企业之间在设计网络管理系统的方法上的不统一。

AUTOSAR 是在 2003 年由全球汽车零部件供应商、汽车制造商和其他半导体、电子、软件系统公司成立的致力于制定汽车电子软件标准的联盟。AUTOSAR 致力于为汽车行业开发一个标准化的、开放的软件架构标准，AUTOSAR 这个架构标准不仅有利于汽车电子软件的交互和更新，同时也为高效管理复杂的汽车电器元件提供了一个基础^[17]。

目前，AUTOSAR 的研发阵营中包含了众多的世界汽车厂商，如通用、宝马、福特、博世、雪铁龙、大众、戴姆勒等。截止 2010 年，AUTOSAR 已包含超过 160 个会员。除此之外，国内的各个科研院校和众多的汽车厂商也纷纷关注 AUTOSAR 标准下软件的设计、开发过程，极大的提高了国内汽车电子方面的研发效率和质量。

2004 年，浙江大学的 ESE 实验中心开始关注 AUTOSAR 标准，并率先加入到该组织。目前，ESE 实验中心已经成功研发出一套 ECU 开发工具链（SmartSAR Studio），该工具链是以符合 AUTOSAR 标准为基础。通过它可以进行 ECU 软件架构的设计、网络通讯系统的配置、基础软件核的配置以及仿真测试等，支持以软件为中心进行的快速由上到下的迭代开发模式。另外，ESE 实验中心已成功研发出符合 AUTOSAR 标准的基础软件模块，例如操作系统模块、通信模块等。2009 年，国内的整车厂技术研究院，例如一汽、长安等也开始使用 AUTOSAR 标准的工具进行 ECU 的设计和软件开发。

总体而言，国内对汽车网络的研究还相对滞后，我们必须密切关注世界汽车电子的发展趋势，借助汽车工业电子化的契机，实现民族汽车工业的大发展大繁荣。

1.3 课题来源

本课题来源于国家电子信息产业发展基金（工信部财函[2011]506 号）“汽车智能网络研发与产业化”和安徽省自然科学基金（1208085QF118）“基于时间隔离的汽车实时网络防危调度研究”。

1.4 研究内容和结构

本课题对车载网络中应用比较广泛的几种总线系统和网络管理标准进行研究分析，并在此基础上对目前广为使用的 CAN 总线系统进行研究。通过对 AUTOSAR 标准进行深入的分析和研究，对网络管理方面存在的不足进行改进，设计了一种改进的 AUTOSAR 车载网络管理办法，并给出该网络管理方法在 CAN 总线上的具体设计过程，最后利用汽车网络仿真工具 CANoe 对该方法进行仿真和验证。

本课题的主要结构内容如下：

第一章介绍了汽车总线的发展前景以及网络管理的相关研究背景，总结了汽车总线在车载网络中的主流地位，分析了汽车网络管理的发展现状，从而引出对 CAN 总线系统进行网络管理的必要性。

第二章梳理了汽车网络的基础知识，并对汽车总线中的两个网络管理协议标准 CAN 和 FlexRay 进行重点介绍，主要包括两种总线标准的特点，两种总线标准的数据帧结构以及数据帧的类型，并对两种标准的数据帧的传输特点进行比较。

第三章对 OSEK 和 AUTOSAR 两个网络管理标准进行研究。首先分析了 OSEK 网络管理的基本原理、特点。其次，对 AUTOSAR 网络管理的基本工作过程进行分析，比较两种网络管理方法的优缺点。最后，以 AUTOSAR 网络管理方法为基础，针对 AUTOSAR 网络管理存在的不足，提出一种改进的 AUTOSAR 车载网络管理方法。

第四章在前两章的基础上讨论了 CAN 总线网络管理的相关内容。根据 AUTOSAR 网络管理规范的基本架构，设计一种适用于 CAN 总线的网络管理方法，并对网络管理过程中的 NM 模块、NM 接口模块、状态管理模块、CAN 总线接口模块和驱动模块进行设计和实现。

第五章设计了 CAN 网络管理的仿真方案。利用 CANoe 软件建立一个实际节点和模拟节点混合的仿真系统来模拟实际运行状态，对 CAN 网络管理功能进行验证，并通过观察各种情况下系统的运行状况，验证本方案的可行性。

第六章对全文的工作进行总结，分析本文的缺陷和不足，提出下一步工作的方向。

第二章 车载网络技术基础

2.1 汽车网络简介

汽车网络减少了线束的连接、增加了汽车的有效空间、减少了汽车的载重，增加了汽车的安全性，是汽车发展的一个重要里程碑。现在的汽车除了满足用户对于交通工具的基本需求之外，还增加了各种车载应用领域的需求，如多媒体服务、助力系统、网络接入等^[18-20]。为实现用户所需的各种功能，大量的电控装置与车载电子设备，如传感器、ECU（电控单元）、GPS（全球定位系统）被广泛应用在汽车中。因此，电控装置与车载电子设备的数量和技术含量已经成为衡量汽车性能的一个重要标志。

最近几年，电子控制单元的应用越来越广泛。从发动机控制到传动系统控制都采用电子单元控制技术。随着车用电气设备的增多和舒适性的提高，汽车电气系统逐渐演变成一个复杂的庞大系统。伴随着 3G 新型电子通讯产品的普及和 ITS 的高速发展，人们在综合布线和信息共享交互方面对汽车的要求越来越高^[21-23]。

从布线的角度看，传统的电气系统之间联系很少，并且大多数都采用比较单一的、点对点的通信方式，这必然造成一个庞大的布线系统，导致材料成本过高且不利于工作效率的提高。据统计，一辆高档汽车中，电气节点大约有 1500 个，采用传统的布线方法，导线长度可达 2000 米。该数字以每十年增长 1 倍的速度在扩大，这就导致了有限的车身空间与庞大的线束之间的矛盾更加突出。随着车载电控设备的增多，车载电控装置之间以及不同控制系统之间共享数据，实时交互各种控制信息的需求也得不到满足，传统布线方法无论从工作效率还是材料成本方面都不能满足汽车未来的发展，阻碍了汽车电子化的进程。因此，将工业化互联网络应用到汽车领域的需求被提出来。互联网络的特点是支持自动控制，并且性能可靠，造价低廉。

从信息共享角度看，典型的控制单元有防抱死制动系统、电控传动系统、防滑控制系统、电控燃油喷射系统、空调系统、巡航系统和废气再循环控制。为了满足系统的实时性要求，需要共享车轮转速、发动机转速、油门踏板位置等汽车公共数据。各个单元实时性的要求各不相同，各个控制单元的控制周期和数据的更新速率也各不相同，从而要求数据交换具有较高的通信速率并且能够根据优先权进行竞争^[24]。

2.2 总线协议介绍

2.2.1 LIN 总线

LIN(Local Interconnect Network)总线协议是以确保安全性为目标的车载通信协议，由 Audi、BMW、Daimler、Volvo、VW 等几个主要的欧洲汽车厂商共

同提出来的，其传输速率可达为 420kbps。LIN 总线协议将串行通讯的概念标准化，使汽车厂商和零配件厂商能够以相对低廉的成本，实现并处理复杂的多任务系统。典型的 LIN 总线主要应用在汽车的联合装配单元，主要包括车门、雨滴传感器、座椅、照明灯、方向盘、空调、交流发电机等实时性要求不高的车载电控装置^[25]。

LIN 总线是一种用于汽车中的分布式电子系统控制，实现低速率、低成本的串行通信网络，主要是为现有的汽车网络（CAN 网络）提供辅助功能。譬如，制动装置之间的通讯和智能传感器间的通信，由于不需要 CAN 总线的高速带宽，使用 LIN 总线通信不仅能够满足网络的基本通信需求，而且降低了生产成本。LIN 网络采用的是一主多从的连接模式，汽车中分布式电子控制单元（ECU）的连接是通过一个没有固定基准时间的（节点同步）时钟线和一根额定电压为 12V 的信号线来实现的。

LIN 总线技术除了定义通信必须的基本协议和规范外，还对 LIN 总线的开发环境、应用软件接口和开发工具等进行了说明。LIN 总线的主要特性有：

(1)以低成本的 UART 接口为基础。UART 接口是目前几乎所有微处理器的必要组成部分，该接口通过几根信号线就可以满足 ISO9141 标准的通信要求；

(2)当需要在网络上增加新的节点时，不需要修改 LIN 网络中从节点的软件结构和硬件设计。

(3)LIN 总线的信号传输速率最高可以达到 20kb/s；

(4)LIN 总线没有消息仲裁机制，网络节点的连接采用一个主节点，多个从节点的工作模式；

目前，LIN 总线的标准化规范以及低廉的成本为低成本汽车软件开发提供了基础，赢得了市场空间，国际上主要的汽车制造商已经接纳 LIN 总线协议。

2.2.2 CAN 总线

控制器局域网（CAN）是一种串行总线通讯协议，能有效地支持分布式实时控制系统要求的突出的可靠性、实时性和灵活性。因此 CAN 总线的应用范围很广，从高速网络到低价位的多路接线都使用 CAN 总线。在汽车电子领域里，将传感器、发动机控制单元、防刹车系统等用 CAN 总线连接起来，其传输速度可以达到 1Mbit/s。世界上一些比较著名的汽车制造厂商的车内控制系统与各个检测单元和执行单元间的数据通信的保证也都通过 CAN 总线来实现，如 ROLLS-ROYCE(劳斯莱斯)、BENZ(奔驰)、PORSCHÉ(保时捷)、BMW(宝马)、JAGUAR(美洲豹)。同时，航海和传感器、自动控制领域、航空航天领域等领域也都逐渐开始使用 CAN 总线。CAN 总线已经被公认为是最有前途的现场总线之一，并且是到目前为止唯一一个具有国际标准的现场总线协议。

CAN 总线在数据通信方面具有较强的实时性、较高的安全性和可靠性、较

高的灵活性等突出特点。与一般总线相比，CAN 总线的主要特点可以归纳为以下几个方面：

(1)CAN 网络采用多主工作方式，网络中的所有节点地位相同。在任意时刻，网络上的任何一个节点都可以主动地向网络上发送信息。

(2)CAN 网络采用非破坏性的总线仲裁机制。通过为每个节点分配不同的 ID（标识符）来确定节点的优先级。同一时刻，当网络中的多个节点由于同时向总线发送消息而产生冲突的时候，ID 优先级较低的节点主动退出发送队列，而优先级较高的节点则继续发送。同时初始化具有相同标识符的远程帧和数据帧时，数据帧的优先级要高于远程帧^[26-30]。

(3)总线上传送的 CAN 消息以广播的形式在整个网络上传输。CAN 节点可以通过对报文 ID 进行过滤实现点对点、一点对多、全局广播等几种形式的接收。

(4)当 CAN 总线的通信速率在 5kbps 以下时，CAN 网络的直接通信距离最远可以达到 10km；当总线的通信速率达到 1Mbps 时，此时的最远通信距离为 40m。总线传输速率可以根据网络的规模进行配置。为了实现正常通信，同一个网络内的传输速率须一致。

(5)总线上的消息采用特定结构的数据帧进行传输，这些数据帧具有特定的格式，长度短，反应速度快。

(6)每帧数据均具有循环冗余校验和其他检错措施，可以检测出错误，并强制结束发送。

(7)总线能够区分节点故障的类型，并隔离永久故障的节点。

(8)CAN 总线可以同时连接多个电控单元。实际应用过程中 CAN 总线可连接的电控单元的个数受到传输延迟，网络负载，响应时间等各个因素的影响。

(9)CAN 器件可以设置为睡眠 / 唤醒模式，将 CAN 器件设为睡眠模式能够减少系统的电源消耗，停止器件内部的活动，断开与总线驱动器的连接。CAN 器件的唤醒模式可以通过总线激活和系统内部的状态唤醒两种方式。唤醒时，传输层需要等待一段时间，然后通过检查 11 个连续的“隐性”位判断是否已经与总线活动同步。

CAN 消息主要是通过数据帧、过载帧、错误帧、远程帧 4 种帧类型来进行消息传输的。

数据帧：负责将数据由发送器传输至接收器。

过载帧：过载帧主要是保证两个数据帧之间的间隔，在先行的数据帧（远程帧）和后续的数据帧（或远程帧）之间提供一个附加的延时。过载帧包括两个位场：过载标志、过载界定符。

错误帧：当任何一个单元检测到总线错误时，就会发出一个错误帧。一个错误帧包含两个不同的场。第一个场主要是由各个不同的站提供的错误标志

(**ERROR FLAG**) 的叠加组成。第二个场是错误界定符远程帧，由总线单元发出，请求数据接收器发送具有同一识别符的数据帧。

远程帧：发送节点请求网络中的其他节点发送具有同一标识符的数据帧，并初始化数据接收器，接收不同的数据。远程帧的 **RTR** 位是“隐性”的，用逻辑 1 表示，这与数据帧正好相反。由于远程帧没有数据场，所以数据长度代码是不受制约的。

无论前面的帧为上述四种帧类型的那种类型，远程帧或数据帧与上述的各种帧通过帧间空间进行分隔。而过载帧之间不是通过帧间空间进行隔离的，错误帧或过载帧的前面也没有使用帧间空间进行分隔。

CAN 总线数据帧的基本结构如图 2-1 所示。CAN 数据帧共包含 7 个不同的位场，帧起始、仲裁场、控制场、数据场、循环冗余校验码场、应答场、帧结尾。其中标识符可以分为 11 位的标准帧标示符和 29 位的扩展帧标示符，数据场的长度可以为 0，但是最多只能为 8 个字节。帧结构如图 2-1 所示。



图 2-1 CAN 数据帧结构

帧起始 (**SOF**)：总线空闲时允许该位发送。用一个单独的显性位来标识 CAN 数据帧或远程帧数据传输的开始。

仲裁场：仲裁场由帧标识符和远程发送请求位 **RTR** 构成。如前所述，标准帧的帧标识符为 11 位，扩展帧的帧标识符为 29 位。帧标识符的发送是从高位到低位依次发送。**RTR** 位主要用于区分是数据帧还是远程帧，在远程帧中 **RTR** 位呈隐性，在数据帧中 **RTR** 位为显性。

控制场：控制场是由 6 个位组成的，其中 4 个位表示数据长度代码，2 个保留位将来作为扩展使用。数据长度代码是指数据场中字节的数量。

数据场：一帧最多为 8 个字节，每个字节包含 8 个位，数据场中填充的是在控制场中指明的、用户需要的 0~8 个字节的数据，。

循环冗余校验码场：主要用于数据校验。由 15 位的循环冗余校验码序列和 1 位的循环冗余校验码界定符（隐性位）共同组成。

应答场 (**ACK**)：在应答场里，发送站发送两个“隐性”位，分别是应答间隙 (**ACK SLOT**) 和应答界定符 (**ACK DELIMITER**)。在应答间隙里，如果 CAN 节点正确接收了一帧数据，则向发送器发送一“显性”的位以示应答；如果没有正确接收到一帧数据，则不予应答。应答结束时应答界定符呈隐性电平。

结束：帧结束标志。

CAN 协议主要针对物理层和数据链路层进行定义和说明，其中数据链路层

在 CAN 协议中占主要部分。CAN2.0 协议规定的分层结构如图 2-2 所示。

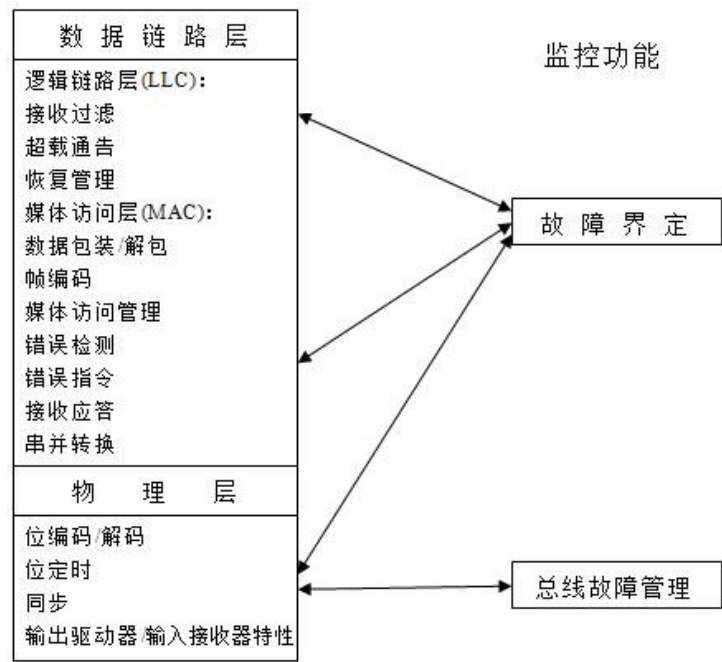


图 2-2 CAN 2.0 分层结构

物理层主要是根据电气属性在不同的节点间传输位信号。物理层又分为：物理信号层和物理介质附件层，物理信号层主要负责编码、解码和定时，物理介质附件层主要负责输入/输出等接收器特性。

数据链路层是 CAN 的核心部分。数据链路层又可以再划分为媒体访问控制层 MAC 和逻辑链路层 LLC^[31]。它们主要负责消息的接收、过滤、超载通告、数据包装和解包等，详见图 2-2。MAC 层是数据链路层最重要的组成部分，它主要负责网络上数据帧结构的指定、传递时序的安排、仲裁方法、应答信号和错误界定等的处理。

数据帧是 CAN 总线上数据传输的基本单位，根据节点发送消息的长度，将分别在总线上传输单帧和多帧。对于单独的一帧来说，根据仲裁域的长度，可以分为 11 位仲裁位的标准帧和 29 位仲裁位的扩展帧。

2.2.3 Flexray 总线

CAN 总线具有较高的数据传输速率、安全性和可靠性，能够满足人们对于汽车的驾乘舒适性和安全性的要求。但是，CAN 总线对于安全等级要求更高的系统，例如，制动传动系统，安全气囊控制等方面却难以满足人们的要求。Flexray 总线协议具有比较快速的实时响应能力、较高的可靠性和稳定性、10Mbps 的数据传输速率，能够满足较高安全等级系统对带宽和可靠性的要求，已经逐渐成为车载总线协议的新标准^[32]。Flexray 总线可以最大限度的满足车辆对信息传输速率、故障容错、确定性等方面的需求，开始受到越来越多的汽车制造商的重视。经过若干年的研究与应用，Flexray 带领汽车电子产业走向新的

目标。

Flexray 通信周期提供两种媒体时序接入方法，静态的时分多址时序接入（TDMA）和动态的柔性时分多址时序（FTDMA）接入^[33]。其中后者采用最小时间片（mini slot）的基本结构，Flexray 通信周期如图 2-3 所示。

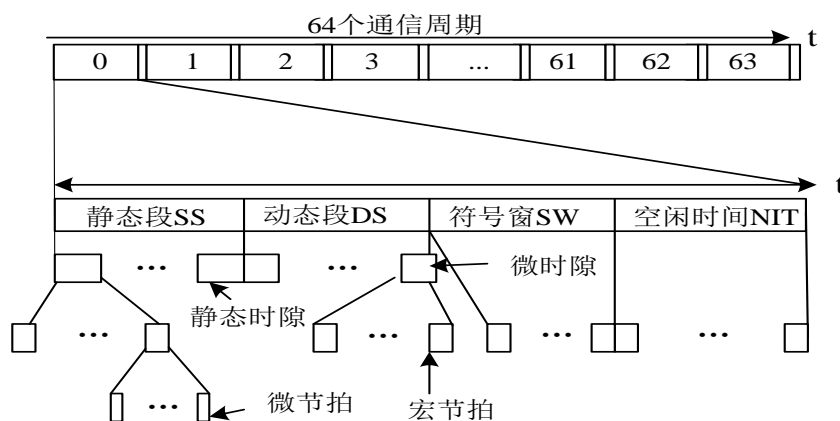


图 2-3 Flexray 的通信时间周期

FlexRay 协议包含 64 个通信周期，如图 2-3 所示，每个周期计数器都从 0 开始，到 63 结束。每一个通信周期都包含四个部分，分别是静态段 (Static Segment, SS)，动态段 (Dynamic Segment, DS)，符号窗口 (Symbol Window, SW) 和网络空闲时间 (Network Idle Time, NIT)。其中静态段使用的是时分多址访问 (Time Division Multiple Access, TDMA) 方法，是基于时间触发的，主要用于传输周期性的消息^[34]。而动态段采用的是柔性时分多址访问 (Flexible TDMA, FTDMA) 技术，主要是基于事件触发的，用于传输非周期消息，符号窗用于唤醒集群的信号表达，网络空闲时间主要用于时钟同步中纠正全局时间。

静态段是由若干个大小固定的时隙 (static slot) 组成，且所有的 static slot 长度相等，一个 static slot 是由若干个 macrostick 组成，一个 macrostick 又由若干个 microstick 组成。其中 microstick 是通信的最小时间单位。静态段的长度在每个周期里都相等，是根据需要传输的消息的最大长度作为标准，通过参数提前配置的，故周期性消息一般都放在静态段传输。在静态段里，static slot 从 1 开始编号，每经过一个 static slot 时间片，static slot 计数器就增加 1。每个节点检查需要发送的消息 ID 与计数器是否相同，如果相同，则传输消息帧，否则将等待下一个周期。静态段的组成如图 2-3 所示。时隙的长度在 FlexRay 协议的参数 gdstaticslot 中定义。

动态段由一些长度可变的时隙 (dynamic slot) 组成，而每个动态时隙又包括若干微时隙 (mini slot)，所有的 mini slot 长度相等。动态段的长度可以根据需要传输消息的大小发生改变，动态段的最大长度是传输全部消息所需的 mini slot 的总长度，当没有消息需要传输时，动态段的最短长度为 1 个 mini slot。

动态段内消息帧的发送时间也是由消息帧的 ID 与时间片计数器的值是否相等决定的。当消息 ID 匹配计数器的值，开始传输的消息会彻底发送完成，然后计数器才加 1，由此可知，当消息的长度大于剩余时间片长度时，即使等到 ID 匹配的时间片，消息也不能全部发送出去，这个时候就把该消息推迟到下一个周期进行发送。如果某一时刻总线上没有消息传输，动态时隙的长度就等于一个微时隙的长度。反之，如果在某个时隙总线上有消息在传输，该消息传输需要的若干微时隙的总长度就是该动态时隙的长度。

在一个时隙内，有且只有一个节点可以在总线上发送消息，而且当有消息要发送时，系统会为其随机分配一个帧标志。网络中的每个节点都有一个最小时间片计数器(Minislots counter)和一个时间片计数器(slot counter)。在每个通信周期开始时都初始化为 1，当要发送消息时，总线仲裁机制将帧标志(FrameID)与当前时隙计数器的值做比较。如果相等，则允许帧标志所对应的节点在这个时隙发送消息，如果不相等，则不允许在该时隙发送。静态时隙和微时隙均是由若干个宏节拍 Macrotick (MT) 组成，MT 的时间用 gdMacrotick 表示，宏节拍又由多个微节拍组成，其中微节拍是由晶振决定的。

符号窗口主要用于发送唤醒特征符。网络空闲时间用于实现分布式时钟同步和节点参数的初始化。

Flexray 的数据帧由帧头、有效数据和帧尾 3 个部分组成。Flexray 总线的数据帧格式见图 2-4 所示。

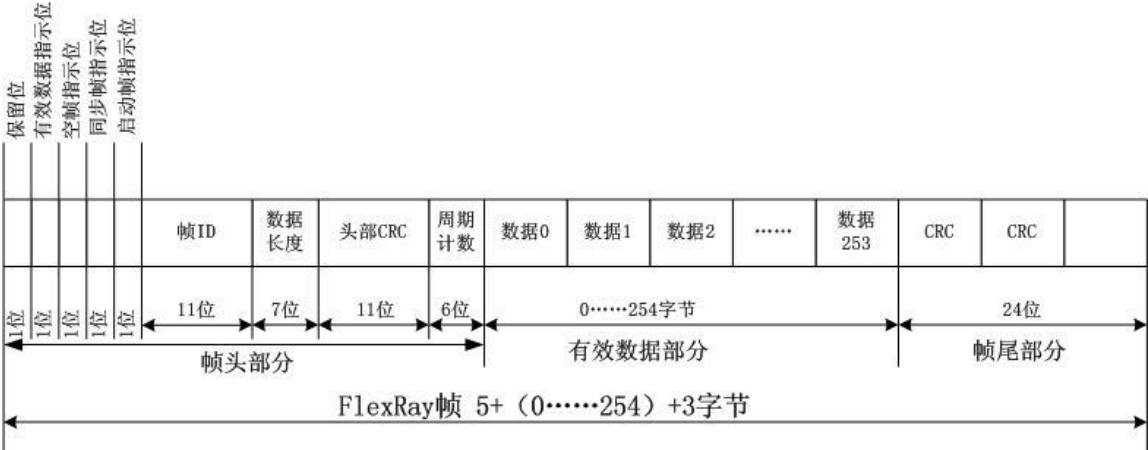


图 2-4 Flexray 数据帧格式

(1)帧头部分

帧头部分组成见图 2-4。其中保留位、同步帧指示位、数据指示位、启动帧指示位、空帧指示位各占 1 位，数据长度占 7 位、帧 ID 部分占 11 位、循环计数占 6 位、头部循环校验循环冗余校验码占 11 位。Flexray 数据帧的帧头部分共有 5 个字节（40bit）。

(2)有效数据部分

如图 2-4 所示，有效数据部分可以包括 0-254 个字节：从数据 0 到数据 253。

在 Flexray 数据帧的循环冗余校验码的校验过程中，当有效数据部分大于 248 个字节时，海明码距设为 4 个字节。当有效数据部分小于等于 248 个字节时，海明距离是有效数据部分的前 6 个字节。消息识别域即消息 ID 是用有效数据部分的前两个数据来表示的，这一特点是在动态时序部分传输的数据帧所特有，如图 2-5 所示。

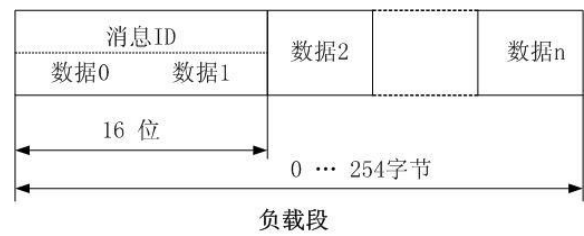


图 2-5 消息 ID

有效数据部分的前 13 个字节构成了网络管理向量（network management vector），其余字节用来传输数据。网络管理向量是静态时序传输的数据帧所特有，网络管理向量的基本结构如图 2-6 所示。



图 2-6 网络管理向量

(3)帧尾部分

24 位的循环冗余数据校验码构成了 Flexray 的帧尾部分。Flexray 的循环冗余校验码由帧头循环冗余校验码的计算值和数据帧的循环冗余校验码计算值两部分构成。

2.3 本章小结

本章首先描述了车载网络的基本概念、产生及其发展，然后主要介绍了目前比较常见的几种车载总线技术，LIN 总线、CAN 总线和 Flexray 总线的基本特点和消息帧格式，然后针对这 3 种总线的优缺点进行比较，并对目前主流的 CAN 总线和下一代总线标准 Flexray 总线在帧结构，通信周期等方面进行详细分析。

第三章 汽车网络管理研究

3.1 OSEK/VDX 网络管理体系

3.1.1 OSEK/VDX 简述

OSEK/VDX 网络管理的主要目的就是保证由各个电子控制单元组成的车载通讯网络的可靠性和安全性。目前，由不同的生产厂商提供的不同的电子控制单元被广泛应用到整车系统中，然后通过汽车总线连接成汽车网络。引入 OSEK/VDX 网络管理的目的是为了节省生产商的开发时间，减少不必要的变量，提高汽车网络的功能性保障，为各个 ECU 节点之间的网络互连提供一个规范化的网络连接标准^[35]。OSEK/VDX 网络管理规定了应用程序接口规范和网络管理的相关机制。

OSEK/VDX 网络管理提供了两种网络监控方法。一种是利用监控周期性信息的方法进行的间接监控，即间接网络管理 Indirect Network Management。另一种是采用逻辑环机制，设计专门的 NM 消息进行通信，从而监控整个网络状态，达到网络管理的目的，即直接网络管理 Direct Network Management。

OSEK/VDX 标准主要包含 4 个基本的模块，实际的操作系统 OSEK OS 模块、通讯子系统 OSEK COM 模块、网络管理系统 OSEK NM 模块、实现语言 OSEK OIL 模块。各个模块版本可以分别定义。OSEK/VDX 标准的体系结构如图 3-1 所示。

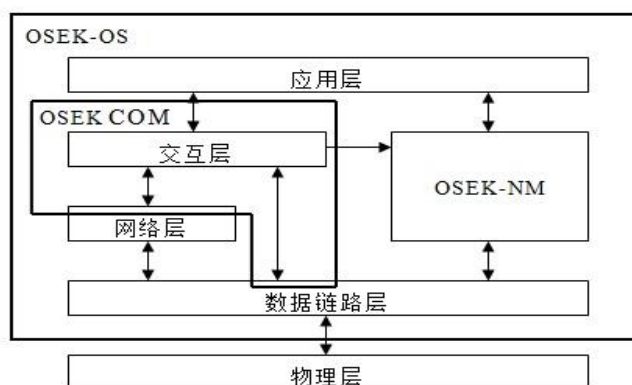


图 3-1 OSEK 体系结构图

为了满足汽车电子领域对系统可靠性和安全性等方面的需求，OSEK 组织制定了一个全面的、系统的操作系统规范，该规范极大的增强了设计师开发以及用户使用的便利性，降低了产品成本。

3.1.2 直接网络管理

直接网络管理是通过专门的 NM 消息来实现的。虽然这种方法给整个网络带来较高负载，但是，直接网络管理可以对网络信息进行精确的配置^[36]。直接网络管理功能分为四大部分，启动时的初始化、运行时的节点状态检测、睡眠

管理和相关的硬件操作。

在 OSEK NM 中，每个节点都是一个独立的逻辑个体，具有各种状态，并且能够传输消息。OSEK NM 中的每个节点都有一个独一无二的网络地址，也就是标识符。网络中的节点通过标识符发送特定的网络管理消息进行通信，主要的网络管理消息有 alive 消息，ring 消息，limphome 消息。

Alive 消息主要是通知网络上的其他节点，本地节点要上线，即加入到逻辑环中。当网络管理功能启动之后，经过一系列初始化操作，节点开始发送的第一个消息是不带睡眠标志的 alive 消息，当节点被跳过时，节点同样发送不带睡眠标志的 alive 消息，告诉网络中的节点自己被跳过，请求重新加入到逻辑环中。网络中的节点收到 alive 消息之后，经过判断，找到该节点在逻辑环中的位置。

Ring 消息的作用是负责逻辑环的正常通信。当网络中的所有节点成功建立逻辑环之后，节点就在逻辑环中传输 Ring 消息，Ring 消息的传输类似于令牌环机制，Ring 消息相当于令牌。

LimpHome 消息是节点在 LimpHome 模式下发送的周期性的网络管理消息，目的是为了保证网络中的其他节点能够监听到本地节点。当该节点能够成功接收到其他节点发送的网络管理消息时，说明节点已经恢复正常，它会发送 Alive 消息，重新加入逻辑环。

OSEK NM 规定逻辑环中的消息是由网络地址小的节点传送到网络地址大的节点，网络地址最大的节点再将消息传给网络地址最小的节点，从而首尾相接组成了一个环结构，称为逻辑环。CAN 网络中的消息是以广播的形式发送的，网络中的一个节点发送消息，网络中的其余节点都可以收到，收到消息的各个节点按照特定的后继检测算法，判断自己是不是发送方节点的逻辑后继，如果是，该节点就接收该消息，如果不是，则将消息抛弃，通过节点的消息过滤功能，实现消息按照逻辑环传输。逻辑环的消息传输结构如图 3-2 所示。

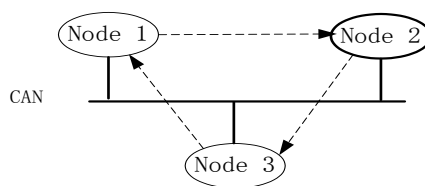


图 3-2 逻辑环结构

Node1 广播一个消息，此时 Node2 和 Node3 同时收到，但是 Node3 根据后继检测算法判断发现自己不是 Node1 的后继，于是放弃这个消息，Node2 通过后继检测算法判断之后，发现自己是该消息发送节点的后继节点，于是接收该消息，同理，Node3 接收到 Node2 的网络管理消息，Node1 接收到 Node3 的网络管理消息，从而组成一个逻辑环结构。

OSEK 直接网络管理定义了 NM On、NM Off 和 NM Shutdown 三个系统状态，而 NM On 状态又进一步划分为 NM Init、NM Awake、NM Bus Sleep3 个子

点都等待 CANNM_MSG_WAIT_BUSSLEEP_TIME 时间之后同时进入睡眠状态。睡眠协商的过程如图 3-4 所示。

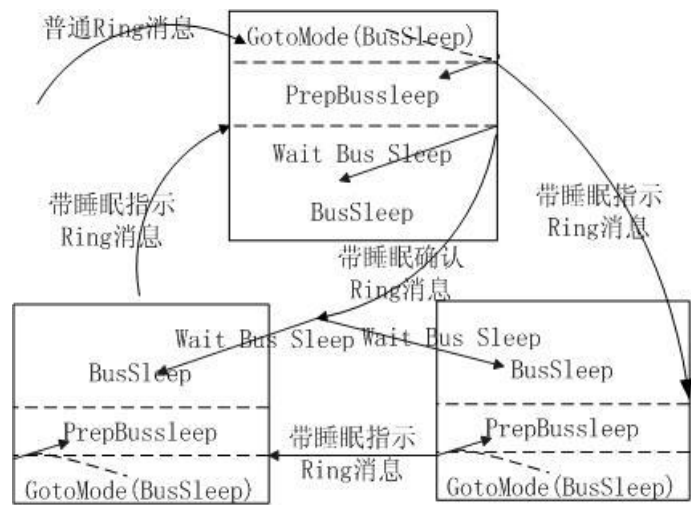


图 3-4 直接网络管理睡眠协商机制

应用程序通过网络管理模块的 GotoMode (BusSleep) 函数，设定睡眠操作码 sleep.ind 和 sleep.ack 控制节点进入睡眠状态。其中 sleep.ind 置 1 表示发送带有睡眠标志的网络管理消息，sleep.ack 置 1 表示发送带有睡眠确认的网络管理消息。

当应用程序调用 GotoMode (Awake) 或者 ECU 内部产生睡眠唤醒事件时，网络管理消息中的睡眠指示位被置 0，节点进入 NMReset 状态，重新启动网络管理并构建逻辑环结构。

在 LimpHome 状态下的节点接收到带有睡眠请求的网络管理消息，会将自己的睡眠标志置 1，等待 Wait_BusSleep_Time 时间之后进入睡眠状态。

3.1.3 间接网络管理

间接网络管理是通过监控网络管理消息来实现网络监控的。间接网络管理的实现很简单、负载也较低，但是容易受到应用消息的影响，相对直接网络管理来讲，很难提供网络配置的精确信息^[38-39]。

间接网络管理是通过监控应用消息来达到监控的目的，所以网络中的节点必须在正常操作模式下周期性地发送特定的消息。

间接网络管理提供两种消息定时的方法：全局定时器和独立定时器。

全局定时器是指总线上所有节点的周期性消息的发送均采用同一个定时器 TIMEOUT_TIME 。它具有占用资源少，实时性差等特点。但是，这种定时器不能保证系统的实时性。如果预定时间 TIMEOUT_TIME 之内没有收到其他节点的网络管理消息，则该节点认为网络上没有其他节点在线，如果在 TIMEOUT_TIME 超时之前收到来自其他节点的网络管理消息，则设置该消息的发送节点为在线状态。若节点超过预定时间 TIMEOUT_TIME 都没有收到自

已发送的网络管理消息，则设置本地节点为静音模式，反之设为非静音模式。

独立定时器是指为每个节点独立分配一个定时器 **Timer_x**，用于监控各个节点的周期性消息。每个节点可以根据自己的特点设置不同的定时时间，从而满足系统的实时性要求，但是独立定时器占用系统资源较多。如果节点 **n** 的定时器 **Timern** 超时，但是节点 **n** 并没有收到来自其他节点的网络管理消息，则设置其他节点为离线状态，反之，配置其他节点为在线状态。如果直到 **Timern** 超时，节点 **n** 都没有收到自己发送的网络管理消息，则设置自己为静音模式，反之设为非静音模式。

间接网络管理的状态转换如图 3-5 所示。间接网络管理主要包括 **NMOff** 和 **NMOn** 两种状态，其中 **NMOn** 状态又包括 **NM BusSleep** 和 **Awake** 两个子状态，**Awake** 又包括子状态 **NMNormal**、**NMWaitBusSleep** 和 **NMLimpHome**。**NM Awake** 是系统进入 **NM On** 之后的默认状态，而 **NM Normal** 又是进入 **Awake** 之后的默认状态。在间接网络管理中，只有当数据链路层出现重大错误时，节点才会进入 **NM LimpHome** 状态，这一转换条件与直接网络管理不同。当故障排除之后，节点将重新返回 **NM Normal** 状态。**NMWaitBusSleep** 是节点进入睡眠状态之前的一个等待状态。当计时器 **WAIT_BUSSLEEP_TIME** 超时时，所有节点进入睡眠状态。

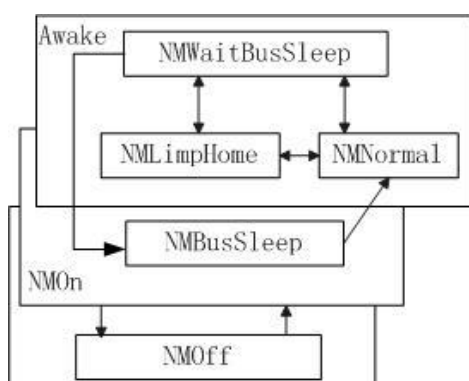


图 3-5 间接 NM 状态图

在间接网络管理中，节点睡眠不采用直接网络管理的协商机制，而是采用主节点控制的管理方法。当从节点需要睡眠时便向主节点发出睡眠请求，主节点询问其他从节点，若有从节点不同意睡眠，则网络维持在唤醒状态，若网络中节点都同意睡眠，则主节点发送睡眠命令，控制所有从节点进入等待睡眠状态，然后等待定时器 `Wait_BusSleep_Time` 超时之后，网络进入睡眠状态。

3.2 AUTOSAR 网络管理体系

3.2.1 AUTOSAR 基础软件架构

AUTOSAR 实际上是一套接口和软件模块的定义标准。车辆上的分布式系统也是以网络为基础的，这一复杂系统的运行最终要靠嵌入式软件来完成。

AUTOSAR 最重要的地方，便是为嵌入式软件提出了一个基本架构。对 AUTOSAR 表示认可的汽车制造商与日俱增，这便为系统设计、评估以及可行性研究等自动化软件的进一步发展搭建了舞台。虽然上述软件刚刚在汽车设计领域崭露头角，但它们最终将成为设计流程优化的基础，帮助设计师以前所未有的设计效率，推出效益可观的汽车产品^[40-42]。

汽车电子基础软件体系架构对不同的硬件提供对的应用软件支撑。AUTOSAR 基础软件的基本结构如图 3-6 所示。

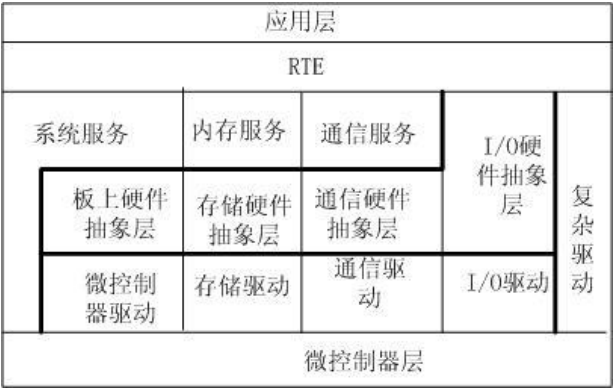


图 3-6 AUTOSAR 基础软件结构

运行时环境（RTE）充当 ECU 内部和 ECU 之间信息交换的通讯中心，不管是使用 ECU 间通讯通道（比如 CAN、LIN、FlexRay、MOST 等）还是在 ECU 内部通讯，通过提供相同的接口和服务，RTE 为 AUTOSAR 软件构件提供通讯抽象^[43]。

基础软件是标准化的软件层，它为 AUTOSAR 软件构件提供服务，是运行软件功能部件所必须的。基础软件位于 AUTOSAR 体系结构中运行时环境的下面，本身并不完成任何功能工作。汽车电子基础软件层是由 5 部分组成的。各部分及其功能如表 3-1 所示。

表 3-1 基础软件层结构和功能

微控制器抽象层	主要作用是避免从上层软件直接存取微控制器寄存器硬件操作，提供微控制器上的各种驱动
ECU 抽象层	主要是减弱上层软件与所有下层硬件的相关性，ECU 抽象为任何特定 ECU 的电气数值提供了软件接口
系统服务层	主要负责给应用软件提供相应的服务，包括网络服务、内存管理、网络通信、操作系统和诊断服务等
复杂驱动层	复杂驱动层主要负责给上层软件提供一种方法，可以直接对一些难以标准化或者实时性要求较高的芯片提供驱动

3.2.2 AUTOSAR 网络管理

网络管理模块位于通信服务（Communication Service）模块之中，其主要

功能是负责通信管理，防止因为节点故障而浪费通信资源，第二个功能是节点状态监控，通过保证每个节点最基本的管理帧发送，使每个节点能够被其它节点发现。AUTOSAR 网络管理的基本状态转换见图 3-7。

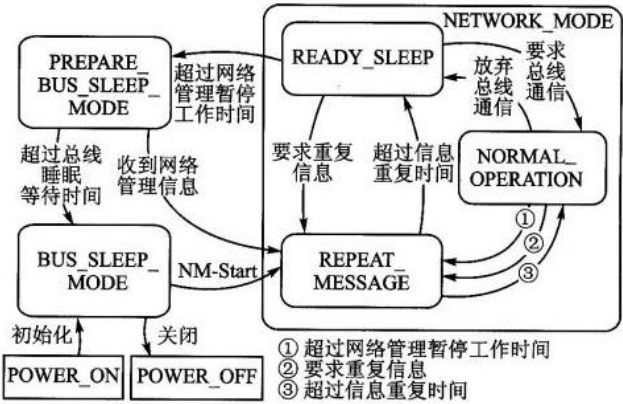


图 3-7 AUTOSAR 状态转换

AUTOSAR 网络管理采用分布式的直接网络管理方法，网络状态的变化是由周期性 NMPDU 的接收情况和节点请求网络的状态综合考虑的。当节点接收到一个 NMPDU 时，意味着该网络管理消息的发送节点意图保持网络的唤醒状态。如果某个节点停止发送 NMPDU，意味着该节点准备进入睡眠状态。但是，只要该节点接收到来自其它节点的 NMPDU，就重新启动等待定时器，推迟进入睡眠状态的时间。如果直到 CANNM_MSG_TIMEOUT_TIME 超时都没有收到其他节点的 NMPDU，意味着所有节点都想休眠，于是整个网络便进入总线睡眠状态。

3.2.3 网络管理开发流程

AUTOSAR 即汽车开放系统架构。汽车开放系统架构联盟是由全球汽车制造商、部件供应商及其他电子、半导体和软件系统公司联合建立，各成员保持开发合作伙伴关系。自 2003 年起，各伙伴公司携手合作，致力于为汽车工业开发一个开放的、标准化的软件架构。AUTOSAR 这个架构有利于车辆电子系统软件的交换与更新，并为高效管理愈来愈复杂的车辆电子、软件系统提供了一个基础。此外，AUTOSAR 在确保产品及服务质量的同时，提高了成本效率。

AUTOSAR 标准提供了一种标准的开发汽车电子软件的方法，并为软件架构中的各个模块定义了统一的接口和硬件平台。另外，AUTOSAR 标准定义了基础软件和嵌入式软件相关的开发、集成工具和一个标准化的软件一致性测试方法^[44]。AUTOSAR 组织的目的是减少各个汽车零部件厂商生产的不同产品的软件开发成本，增加产品内部以及不同产品之间软件的可移植性和复用性，增加产品的质量和可靠性。汽车电子软件的系统设计和开发流程见图为 3-8。

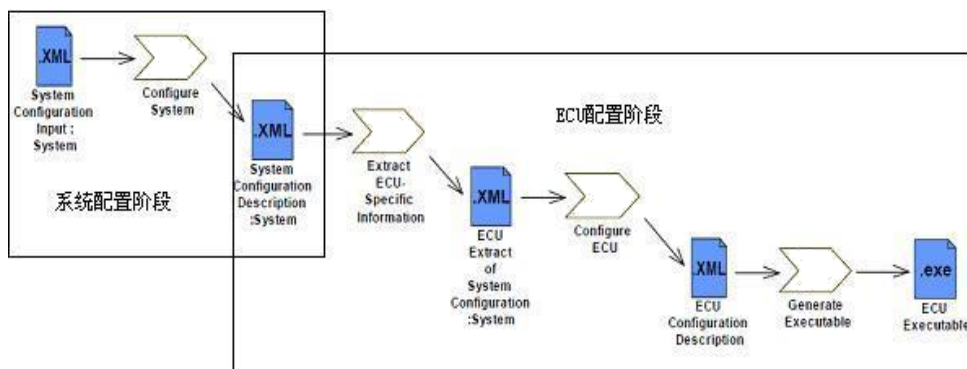


图 3-8 系统设计和开发流程

汽车电子软件的开发过程可划分为系统配置和 ECU 配置两个阶段。

第一阶段是系统配置阶段，属于系统级设计决策工作。主要编写系统配置输入文件和系统配置描述文件。系统配置输入文件是 XML 类型的文件，包括软件构件描述输入、ECU 资源描述输入、系统约束描述输入。软件构件描述输入定义了每个软件构件的接口，如形参的数据类型、端口、接口等。ECU 资源描述定义了每个 ECU 的资源需求，如处理器、外部设备、存储器、传感器和执行器等硬件资源。系统约束描述定义了总线信号、拓扑结构和软件构件的映射关系。系统配置描述文件是在系统配置输入文件基础上借助系统配置生成器生成的系统配置描述文件，是这一阶段的最终成果。该文件将包含所有的系统信息，包括将软件构件映射到相关的 ECU 上（这种映射需要考虑到构件的需要、构件的连接、资源需求以及约束条件，有时也需要考虑成本等方面的因素），以及通信矩阵（整车的网络结构、时序以及网络数据帧的内容）。

第二阶段是 ECU 的配置，这一阶段是第一阶段的延续，对每个 ECU 分别进行细化。首先从系统配置描述文件中提取出与各个 ECU 相关的系统配置描述信息，包含 ECU 通讯矩阵、拓扑结构、顶级功能组合（据此产生需映射到该 ECU 上的所有软件构件）等放在另外一个 XML 中。然后进入 ECU 配置的实际工作，负责给输入对象中添加具体应用所必需的信息，如任务调度、必要的 BSW 模块、BSW 配置信息、给任务分配可运行的实体等。这一步的结果被放在 ECU 配置描述文件中，它包含了具体 ECU 所需的所有信息。最后生成具体 ECU 的可执行程序，根据 ECU 配置描述文件中的配置信息构建完成 ECU 的基础软件设置和应用软件的集成，并生成 ECU 可执行的代码。在这个设计过程中，使用了虚拟功能总线的概念，即将 AUTOSAR 软件构件间的通信、软件构件与基础软件间的通信进行了抽象，同时使用了预先定义的标准接口。在系统实现过程中，虚拟功能总线所代表的功能最终以 RTE 的生成来体现。

通过 RTE 实现 AUTOSAR 软件构件（即应用程序）相互间的通信以及软件构件与基础软件之间的通信的前提是软件构件必须具有标准的 AUTOSAR 接口。目前，AUTOSAR3.1 版已定义了一些典型的汽车电子应用领域（动力，车

身/舒适和底盘)的标准接口。AUTOSAR 按照功能逻辑分别将这些领域的系统划分成若干个模块,这些模块可被视为一个软件构件或多个软件构件的组合,这些功能性的软件构件的接口被明确定义,所定义的接口的内容包括名称,含义,范围,数据类型,通信类型,单位等。应用软件开发者在软件构件的设计与开发时需要应用这些接口定义。

软件构件描述文件的生成,需要获取每个软件构件关于接口、行为、硬件接口、运行性能需求等方面的信息^[45]。软件构件描述文件本身将包含如下内容:一般特性,如名称、生产商等;通信属性,端口、接口;内部结构,子构件、连接关系;需要的硬件资源,处理时间、调度、内存等。

ECU 描述文件包含如下内容:一般特性,如名称、生产商等;温度(自身、环境);可用信号处理方法;可用编程能力;可用硬件,微控制器、内存、接口外设等;RTE 之下针对控制器的基础软件模块;从引脚到 ECU 抽象层的信号

系统约束描述文件包含如下内容:网络拓扑,总线、连接的 ECU,网关;通信;通信矩阵、网关表;软件构件的映射

以上描述的系统配置输入完成后,使用系统配置工具导出系统配置文件,决定哪个软件构件运行在哪个 ECU 上,生成 ECU 配置描述和系统内的通信矩阵。接下来就是 ECU 配置。将每个 ECU 配置信息从系统配置文件中提取出来,包含 ECU 通信矩阵、拓扑结构、顶级功能组合。将软件构件和基础软件的代码集成生成可执行的代码。在这整个过程中,需要进行多次的重复修改,以保证实现功能并且是最优的方式。

3.3 一种改进的 AUTOSAR 网络管理方法

AUTOSAR 网络管理存在如下不足:一是当节点处于 NORMAL 状态时,如果此时出现故障,则不会发送网络管理消息,其他节点却认为该节点准备睡眠;二是节点本身不能正常使用总线收发消息进行通信时,没有对应的处理措施。为此,对原有的网络管理方法进行改进,其基本思想主要包括两点:一是修改状态转换图,在原有的状态转换图的基础上,增加 LIMPHOME 状态,当节点不能正常使用总线时,则进入该状态;二是修改睡眠机制,在 NMPDU 中增加一个睡眠标志位,当节点准备睡眠时即进入 READY SLEEP 状态时,重复发送带有睡眠标志位的网络管理消息,其他节点在此基础上判断节点是处于故障状态还是准备睡眠状态。为此,对现有的 AUTOSAR 网络管理标准进行改进。

3.3.1 节点状态设计

在汽车网络管理中,根据节点在网络上的运行情况,将节点的状态划分为 PRE_BUS_SLEEP 模式、BUS_SLEEP 模式和 NETWORK 模式。PRE_BUS_SLEEP 模式是网络中的所有节点同意睡眠,进入预备睡眠状态。这是节点进入睡眠状态之前的一个瞬时过渡状态。BUS_SLEEP 模式是网络中的所有节点停止网络

通信，进入省电模式的状态，也是网络管理功能启动之后默认进入的一个状态。NETWORK 模式是网络中的节点进行正常的网络管理通信的一种状态，该模式又包含 REPEAT 状态、NORMAL 状态、READY SLEEP 状态 3 个子状态。

REPEAT 状态是节点从 BUS_SLEEP 状态或者 READY SLEEP 状态进入 NETWORK 状态之后的默认状态。节点会在该状态下停留一段时间，时间用参数 REPEAT_MESSAGE_TIME 表示。在该状态下，节点周期性的发送 Rpt 置 1 的网络管理消息，目的是为了更方便网络中的其他节点发现该节点，同时也让该节点在网络上停留一段时间。当 REPEAT_MESSAGE_TIME 超时，节点就离开 REPEAT 状态，如果此时的网络通信被请求，即通信层调用 Nm_NetworkMode()，节点便进入 NORMAL 状态，如果此时的网络通信没有被请求，节点进入 READY SLEEP 状态。

NORMAL 状态保证了当网络通信被请求时，网络中的任何一个节点都能够唤醒网络管理簇。在 NORMAL 状态下，节点发送正常的网络管理协议数据单元。

READY SLEEP 状态是网络中的节点已经没有通信需求了，于是网络中的节点在此状态下等待进入睡眠状态。这是网络中的节点同步的一个状态。在该状态下，当某个节点在总线上发送网络管理消息之后，网络中的节点都会收到该消息，当然，发送节点本身也会收到自己发送的网络管理消息，收到之后网络中的所有节点都重新启动定时器 NM_MSG_TIMEOUT_TIME，如果在 NM_MSG_TIMEOUT_TIME 超时之前没有收到网络管理消息，节点将同时进入到 PRE_BUS_SLEEP 状态。

3.3.2 状态转换设计

为了能对总线故障进行处理，引入 OSEK 网络管理对总线故障的处理，与 AUTOSAR 原有的状态转换图相比，增加了一个 LIMPHOME 状态。当节点在 NETWORK 模式下的发送错误或者接收错误的次数超过阈值，节点将进入 LIMPHOME 状态，并在该状态下每隔一段时间重复发送 LIMPHOME 消息。如果在指定时间内，该节点发送的消息都没有成功或者不能收到网络管理消息，则说明该节点不能正常的使用总线通信，节点将错误上报应用层。

节点在网络状态中都启动 NM_MSG_TIMEOUT_TIME 定时器，当节点在网络状态下收到网络管理消息或者成功发送网络管理消息时，就会重新启动该定时器，如果在 REPEAT 状态和 NORMAL 状态下 NM_MSG_TIMEOUT_TIME 定时器超时没有收到任何消息，则节点重新启动该定时器。只有在 READY SLEEP 状态下定时器 NM_MSG_TIMEOUT_TIME 超时，节点才会同步进入预备睡眠状态 PRE_BUS_SLEEP。当网络进入预备睡眠状态 PRE_BUS_SLEEP，等待 NM_MSG_WAIT_BUSSLEEP_TIME 时间后，如果没有通信需求，整个网络便

进入 BUS_SLEEP 状态，如果在等待 NM_MSG_WAIT_BUSSLEEP_TIME 超时的过程中收到消息，节点进入 REPEAT 状态。在准备睡眠状态和睡眠状态中，如果一般网络管理接口需要网络管理通信，可以通过调用相关的接口函数启动 CAN 网络管理进入 REPEAT 状态。

改进后的网络管理主要包含 5 个基本状态，它们依次是启动状态（POWER_ON）、关闭状态（POWER_OFF）、睡眠状态（BUS_SLEEP）、预备睡眠状态（PRE_BUS_SLEEP）、网络状态（NETWORK）。其中网络状态又细分为消息重复状态（REPEAT）、正常工作状态（NORMAL）、准备睡眠状态（READY_SLEEP）、跛脚状态（LIMPHOME）。各个状态之间的转换关系如图 3-9 所示。

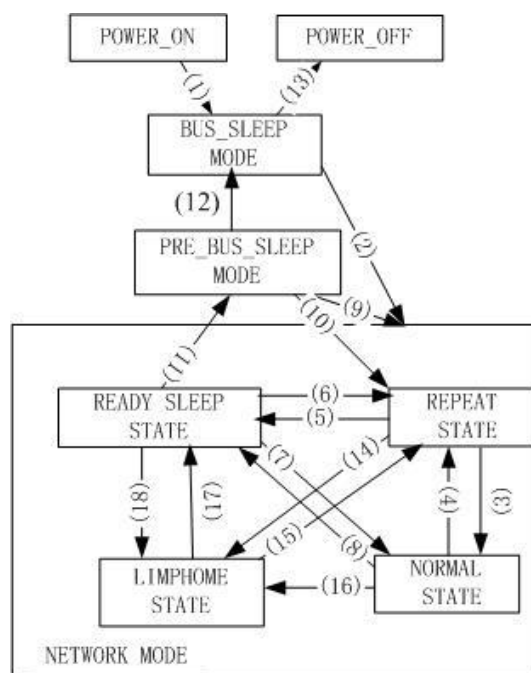


图 3-9 状态转换设计

状态转换的具体条件如下：(1) 网络管理初始化后进入 BUS_SLEEP 模式；(2) 节点通过 Nm_NetworkMode() 请求网络通信；(3) REPEAT 状态定时时间到、节点需要网络通信且节点能够正确收发网络管理消息；(4) 收到 REPEAT 状态的网络管理消息或者节点自身请求进入 REPEAT 状态；(5) REPEAT 状态定时时间到且节点自身不需要网络通信；(6) 收到 REPEAT 状态的网络管理消息或者节点自身请求进入 REPEAT 状态；(7) 节点需要通信，请求总线；(8) 节点不需要通信，释放总线；(9) 节点收到网络管理消息；(10) 节点自身请求网络通信；(11) 网络管理定时器超时；(12) 预休眠定时器超时；(13) 系统断电。(14) 在 REPEAT 状态定时时间内，节点发送错误达到或者接收错误超过阈值；(15) 节点能够正确的发送网络管理消息并且能够收到不带睡眠标志的网络管理消息；(16) 节点发送错误达到或者接收错误超过阈值；(17) 节点收到带有睡眠标志的网络管理消息；(18) 节点之前处于 LIMPHOME 状态，并且收到不带睡眠标志的网络管理消息；

眠的网络管理消息。

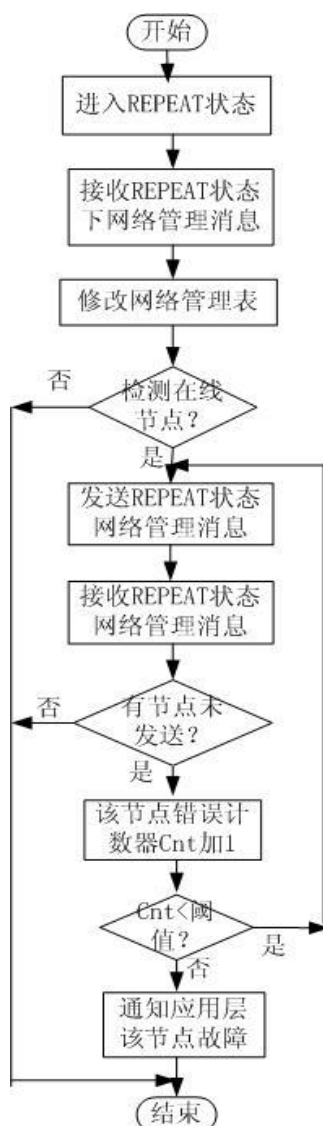


图 3-10 故障节点检测流程

3.3.3 节点故障检测设计

故障节点检测是网络管理的另一功能，通过修改网络的睡眠机制和在网络管理接口模块中配置一张动态的网络管理表来实现故障节点的检测功能。网络管理表包括三个字段，分别是节点 ID、未发送数据间隔和故障计数器 Cnt。

在 NM 启动之初，每个节点都默认进入 REPEAT 状态，并且各个节点按照一定的顺序在 REPEAT 状态下发送网络管理消息。其他节点将接收到的 NM 消息上报 NM 接口模块，并将收到的 NMPDU 的源地址存储在网络管理表节点 ID 列中，重复 ID 不记录该表。当收到任何一个节点的网络管理消息时，则将该节点对应的“未发送数据间隔”和“故障计数器 Cnt”清 0。当某个节点在未发送数据间隔内不发送网络管理消息，则由 AUTOSAR 网络管理机制可知，该节点可能存在故障。如果有节点需要对其进行检测，则发送 REPEAT 状态下的网络管

理消息，正常的节点收到该消息后均在下一个发送周期发送 REPEAT 状态下的网络管理消息，如果该节点不发送，则将故障计数器 Cnt 加 1。重复上述过程，当故障计数器 Cnt 达到阈值时，则认为该节点故障，并将其上报给应用层。故障节点检测过程见图 3-10。

3.3.4 协同睡眠设计

与改进前的 AUTOSAR 网络一样，改进后的网络管理也采用睡眠协商的方法。所不同的是，当网络中的节点不需要总线通信而进入 READY SLEEP 状态后，在网络管理定时器超时之前，继续重复发送带有睡眠标志位的网络管理消息。网络中不能正常发送消息的节点收到该睡眠消息时，设置自身也进入 READY SLEEP 状态，其他节点收到该消息后，除登记该节点准备睡眠的信息外，不对网络状态和网络管理定时器做任何修改。

网络中带有睡眠标志的网络管理消息对在线节点得网络状态没有任何影响，所以当网络中的节点都发送带有睡眠标志的网络管理消息时，节点能同步进入到睡眠模式。由于故障节点不能发送网络管理报文，采用这种协商睡眠的方法不但能够保证网络中的节点进入同步睡眠模式，同时也能够对网络中的故障节点进行判断。在协同睡眠的过程中，网络管理模块与其他各模块的通信关系如图 3-11 所示。

网络管理中的睡眠是由 ComM 层和网络管理层的通信情况共同决定的，如果 ComM 请求释放总线，但是总线上有消息传输，总线将不会被释放。只有总线上没有消息传输并且通信管理 ComM 模块释放总线，网络中的节点才会进入睡眠状态。

通信管理模块（ComM）通知网络管理模块（Nm）不需要总线通信了，请求释放总线。网络管理模块也通过一个 Nm_RemoteSleepIndication 函数检测到网络中的其它节点已经准备好进入 BUS_SLEEP 状态，此时网络管理模块启动一个关闭定时器。当定时器超时之后，网络管理接口模块（NM 接口）发送一个总线同步消息给网络管理模块（Nm），同步完成之后，网络管理接口模块通知网络管理模块释放总线，网络管理模块进入 READY SLEEP 状态。然后，网络在 READY SLEEP 状态下等待 NM_MSG_TIMEOUT_TIME 定时器，当定时器 NNM_MSG_TIMEOUT_TIME 超时之后都没有收到网络管理消息，网络管理模块通知网络管理模块接口模块，网络进入 PRE_BUS_SLEEP 状态。网络管理接口模块将网络的状态通知给 ComM 模块，为了使整个体系中各个模块的状态一致，ComM 模块将网络的状态通知给状态管理模块 STATE Manager（SM）。当网络在 WAIT_BUS_SLEEP 状态没有收到通信需求，等待定时器 WAIT_BUSSLEEP_TIME 超时之后，网络管理模块通知网络管理接口模块，网络进入 BUS_SLEEP 状态。同样的，网络管理接口模块再通知 ComM，网络进

入睡眠状态。ComM 再将网络的状态变化通知状态管理模块。

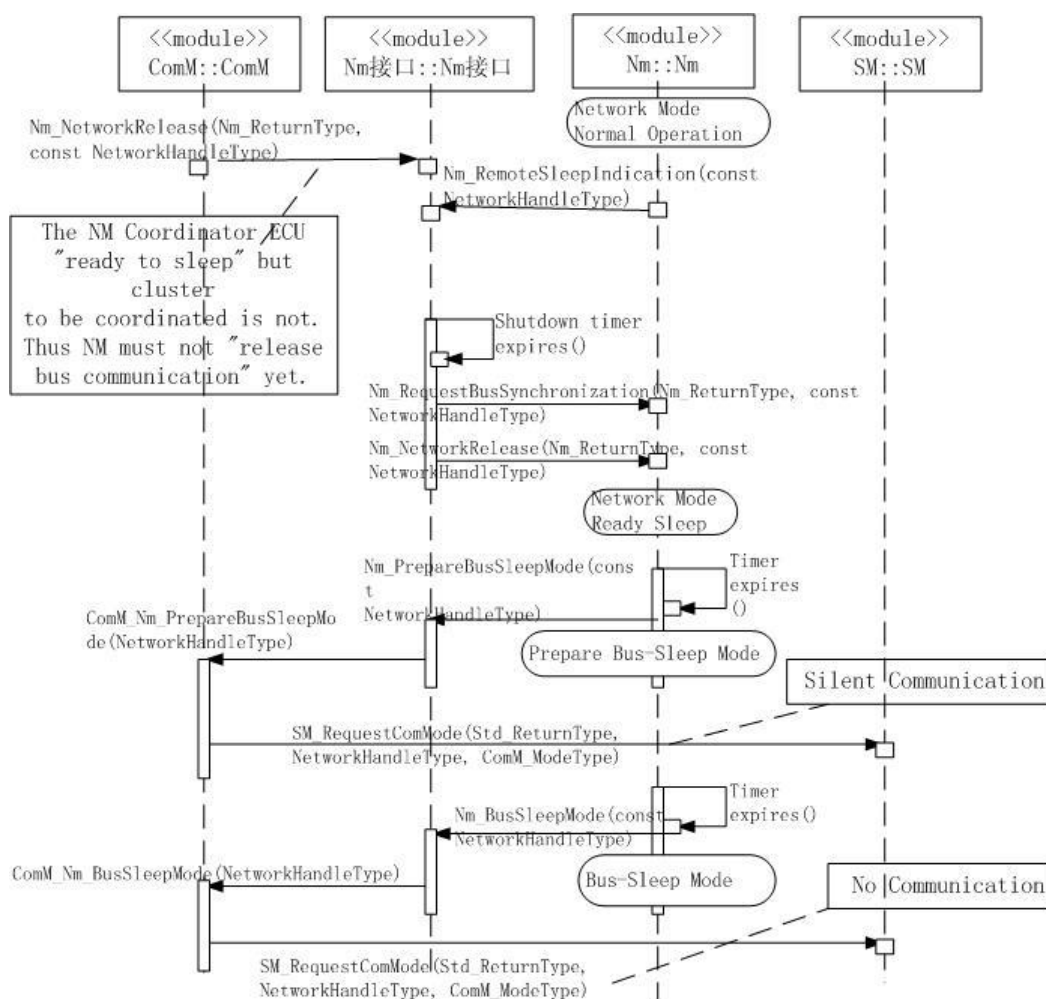


图 3-11 协同睡眠过程

第四章 CAN 网络管理实现

4.1 总体设计

AUTOSAR CAN 网络管理功能的实现是由通信管理 ComM (ComM, Communication Manager)、CAN 状态管理 (CANSIM, CAN State Manager)、网络管理 (NM) 接口、CAN 网络管理 (CAN NM, CAN Network Management)、CAN 接口 (CAN Interface)、CAN 驱动(CAN Driver)六个模块共同作用的。CAN 网络管理的基本结构如图 4-1 虚线框所示。

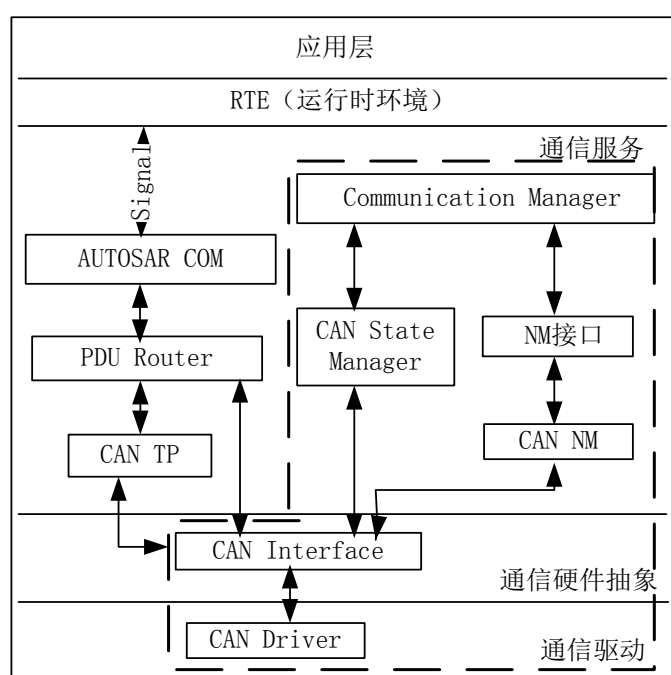


图 4-1 CAN 网络管理基本结构

ComM 模块封装了控制底层的通信服务^[46]。通信管理模块从通信请求者那里收集总线通信访问请求，并将这些请求进行协调。通信管理模块的主要目的是：

- (1) 简化用户对于通信栈的使用。这包含一个简化的网络管理处理；
- (2) 在一个 ECU 的多个独立软件组件之间，协调通信栈的可用性(允许发送和接收信号)；
- (3) 提供一个 API 禁止发送信号，从而阻止 ECU 积极唤醒通信总线；
- (4) 通过为每个通道设置一个状态机控制一个 ECU 的多个通信总线通道；
- (5) 提供一种可能性迫使 ECU 在 No Communication 模式下保持总线处于唤醒状态；
- (6) 通过为被请求的通信模式分配所有必须的资源，简化资源管理。

CANSIM 模块是通信服务层的一个成员，只能用于 CAN 通信系统。CANSIM 的主要任务是通过 CAN Interface 模块控制底层的一个或者多个 CAN 控制器和

CAN 收发器。

CAN NM 模块是一个只能应用于 CAN 总线，且独立于硬件的协议，是 NM 接口和 CAN Interface 之间的适配器。CAN NM 模块的核心功能是负责协调网络的正常操作模式和睡眠操作模式之间的转换。除此之外，它还提供检测在线节点服务，检测网络中的其它节点是否准备好睡眠等服务。

NM 接口模块是 ComM 模块和 CAN NM 模块之间的适配器，这是 NM 接口模块的基本功能。另外，NM 接口模块还提供可选的网络管理协同功能，主要是处理连接多个网络的 ECU 节点实现网络的同步睡眠的功能。

CAN 接口模块是与 CAN 驱动模块进行交互的接口，主要负责对网络管理消息的发送、接收过程进行控制。

CAN 驱动模块是 CAN 控制器与物理总线之间的接口，用于实现对硬件设备的访问，可以提供对总线的差动发送和接收功能。

4.2 网络管理接口模块设计

网络管理接口模块是位于 CAN 总线网络管理模块与 ComM 模块之间的一个独立于具体总线的适配层，它是应用层访问网络管理功能的唯一的途径。它提供接口给 ComM 层，接口中包含足以完成必要的网络管理功能所需的信息，但是不包含底层总线的具体信息。网络管理接口模块为上层应用服务提供网络管理的统一接口，网络管理接口模块可以同时接入多个总线的网络管理模块，见图 4-2。

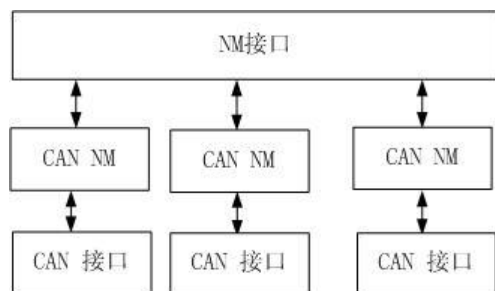


图 4-2 网络管理接口

网络管理接口模块包括两部分：第一部分是与 CAN 总线网络管理模块结合起来，保证程序正常运行所需要的基本功能，这些基本的功能位于每个 ECU 节点中；第二部分是在多个总线系统构成的网络中，NM 协调不同总线网络同步进入关闭状态。

网络管理接口通过使用协同算法来实现网络中单个节点、多个节点和所有节点的协同关闭功能，协同算法根据不同的配置来实现不同级别的协同。使用协同算法的节点通常被认为是 NM 协调器。网络管理接口通过配置 NmCoordinatorSupportEnabled 参数来定义协同算法是否需要 NM 协调器功能。协同算法的具体方法见图 4-3。

网络管理协同算法将不断的检测一个网络中的所有当前处于唤醒状态的节点是否准备进入睡眠状态。这个协同算法只在协同网络准备进入睡眠状态时才会启动，已经进入 `BUS_SLEEP` 状态的节点虽然能被该算法检测到，但是不被协同睡眠。检测是通过 `main` 函数中的网络管理的所有的应用程序接口调用。

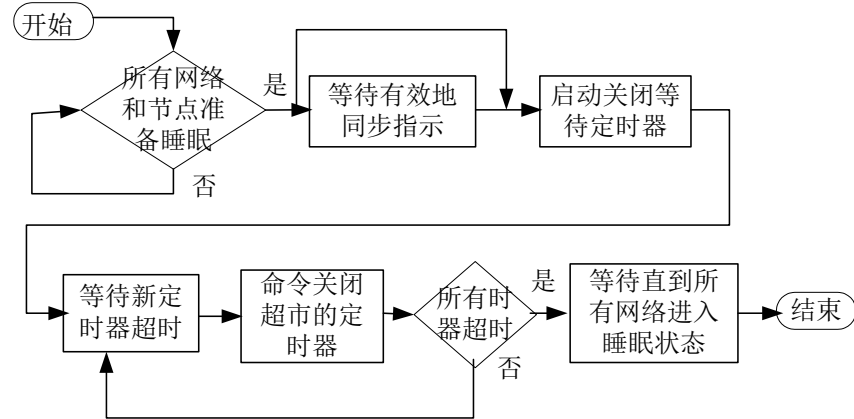


图 4-3 协同算法

关于网络的同步主要有：同步命令、同步启动和同步网络睡眠。

同步命令是通过参数 `NmSynchronizingNetwork` 实现的，主要同步不同网络的时间点，这有助于以最快的速度关闭所有网络。它的负面作用是，网络不能同时进入 `BUS_SLEEP` 模式。例如：当有多个 `CAN` 网络互联，其中一个网络的某个节点利用 `CANNm_NetworkRequest` 请求网络，此时多个互联的网络也将保持唤醒状态。但是，当所有节点都准备睡眠，不同的网络是否同时进入 `BUS_SLEEP` 状态已经无关紧要了。因为同步命令没有考虑到网络的周期性行为。同步启动是同步命令的一个延伸，考虑到网络释放只能在特定的时间点进行，同步启动命令进行同步关闭时，考虑到网络的周期性行为，它会等待一个合适的时间进行网络同步。

同步睡眠主要是同步不同的网络在同一个时刻进入 `BUS_SLEEP` 模式。通过接收到同步网络发送的标志，然后延迟特定时间释放网络，这样就能保证所有从 `NETWORK` 模式或者 `PRE_BUS_SLEEP` 模式转入 `BUS_SLEEP` 模式的网络能够尽可能的同步。

4.3 网络管理模块设计

4.3.1 CAN 网络管理消息设计

`CAN` 帧支持 8 个字节的内容，两个字节用于网络管理模块，其余的 6 个字节提供给系统用户，根据不同的网络布线特点进行网络管理扩展算法的实现。该系统中 `NMPDU` 的设计如表 1 所示。

表 4-1 NM 协议数据单元设计

Byte7- Byte2	Byte1	Byte0
User data	Control Bit Vector	Source Node Identifier

Source Node Identifier (Srd): 每一个发送方都有一个唯一的网络标识符, 这个网络标识符是在网络设计过程中静态配置的, 用于标识发送消息的节点。

User data: 用户自定义数据。

Control Bit Vector (CBV) 是表示是否需要检测在线节点, 以及发送消息的类型, CBV 具体设计如表 2 所示。

表 4-2 Control Bit Vector 结构设计

Bit7- Bit3	Bit 2	Bit1-Bit0
0	RdySleep	Rpt

Rpt (2 位): 11 表示网络请求故障检测, 01 表示传输普通的网络管理消息, 10 表示节点发送的是 LIMPHOME 消息。

Rdysleep (1 位): 该位主要用于标识节点是否需要总线通信, 即是否要进入睡眠状态。该位置 1 表示节点要进入睡眠状态, 该位置 0 表示节点需要总线通信。

4.3.2 错误处理

消息在总线上进行传输时, 产生错误是不可避免的, 当传输过程中有错误发生时, 通过错误处理机制发现错误并进行相应的处理。错误处理机制的基本原理是: 如果一个网络管理协议数据单元 (NMPDU) 传送成功, CAN Interface 模块将返回一个确认标志, 根据配置 CANNM 模块将对这个确认进行评估。错误处理的功能就是监视这些确认标志, 同时, 如果在一个特定的时间内没有收到确认标志, 错误处理模块将该消息通知上层模块。错误处理模块的功能是通过在每个发送的网络管理协议数据单元 (NMPDU) 之后启动一个定时器实现的。如果在定时器超时之前没有收到确认标志, 网络管理回调函数将被调用。当总线上有网络管理消息传输时, 启动网络管理消息发送定时器 CANNM_MSG_TIMEOUT_TIME。当 CAN Interface 调用网络管理发送确认函数 CanNm_TxConfirmation 时, 定时器 CANNM_MSG_TIMEOUT_TIME 将被禁止。当定时器 CANNM_MSG_TIMEOUT_TIME 超时, CAN 网络管理模块将调用一次网络管理发送超时函数 Nm_TxTimeoutException。

当 CAN 网络管理在 PASSIVE 模式下被启动即 NM_PASSIVE_MODE 为 true 或者 CAN 网络管理的立即确认功能被配置即 CANNM_IMMEDIATE_TXCONF 被设置为 true, CAN Interface 模块将不能返回消息发送成功的确认标志, 此时, CAN 网络管理的传输错误处理功能将无效。

4.4 通信管理模块设计

4.4.1 设计目的

通信管理模块（ComM）是基础软件（BSW）的一个组件，作用是将控制底层的通信服务进行封装。ComM 模块控制和通信相关的基本软件模块。通信管理模块从通信请求器那里收集总线的通信请求，并协调这些总线通信请求。

通信管理模块中的用户可以是一个运行的实体，一个软件组件，一个基本软件管理等，它的主要目的是：

(1)简化用户对总线通信栈的使用。这包括一个简化的网络管理处理；

(2)由于用户对于具体的硬件设备不了解，仅仅通过请求“通信模式”进行消息的发送，ComM 模块负责相应的通道的开关，即对于一个 ECU 上的多个独立的软件组件，负责协调总线通信栈的可用性（允许发送和接收信号）；

(3)提供一个 API 禁止发送信号，从而保证 ECU 不能唤醒总线。因为 ComM 可以控制不同的总线类型，对于 CAN 总线，任何一个 CAN 消息都能够唤醒总线，但是对于 FLEXRAY，只有通过 WAKE_UP 模式才能够唤醒总线，因此应该严格控制总线被唤醒；

(4)通过为每一个通道设置一个状态机，从而控制一个 ECU 的多个通信总线通道。ComM 通过相应的总线状态管理模块请求通信模式，总线状态管理模块是负责控制实际的总线状态

(5)提供让一个 ECU 被动进入 NO COMMUNICATION 模式的方法；

(6)通过给请求的通信模式分配所以必须得资源达到简化资源管理的目的。例如：当用户请求 FULL COMMUNICATION 模式，ComM 检查通信是否被允许，并在通信期间禁止 ECU 进入关闭状态。

4.4.2 通信管理状态机设计

为了给应用程序提供服务，结合应用程序对网络信道的使用情况，将 ComM 模块的通信进行模式的划分。通信管理模块提供 3 中不同的通信模式，最高级别的 FULL_COMMUNICATION、SILENT_COMMUNICATION 和最低级别的 NO_COMMUNICATION。ComM 状态机内部的功能对用户是透明的，用户不需要了解内部的工作模式，当有多个用户请求通信，以最高级别的通信模式作为整个通信信道的通信模式，从而保证所有用户都能正常工作。

FULL_COMMUNICATION 表示总线通信被请求，主要包含用户请求使用信道的子状态 FULL_COM_NETWORK_REQUESTED 和用户释放信道的子状态 FULL_COM_READY_SLEEP，在该模式下既允许在总线上发送通信消息，也允许接收总线上的通信消息，而 NO_COMMUNICATION 表示总线通信被释放，是进入 ComM 状态机之后的默认状态。在该状态下，不允许总线收发消息，包含 NO_COM_PENDING_REQUEST 和 NO_COM_NO_PENDING_REQUEST

两个子状态。**SILENT_COMMUNICATION** 表示总线通信被释放，只允许接收总线通信消息，不允许发送总线通信消息。

FULL_COM_READY_SLEEP 和 **SILENT_COMMUNICATION** 主要是用于同步关闭总线通信，因为如果只有一个 ECU 关闭通信，其余的 ECU 将接收不到它发送的应用信号，从而认为该节点故障并存储故障信息。

4.4.3 通信管理状态机转换

在子状态 **COM_NO_PENDING_REQUEST** 下，如果收到 ECU 模块发送的 **WAKE_UP** 指示，或者接收到网络管理模块发送的重启请求，或者用户请求进入 **FULL_COMMUNICATION** 通信模式，则 **ComM** 状态机将立即进入 **COM_PENDING_REQUEST** 状态，如果在该状态下没有收到有效地 **FULL_COMMUNICATION** 模式请求，**ComM** 状态机将维持在该状态，见图 4-4。

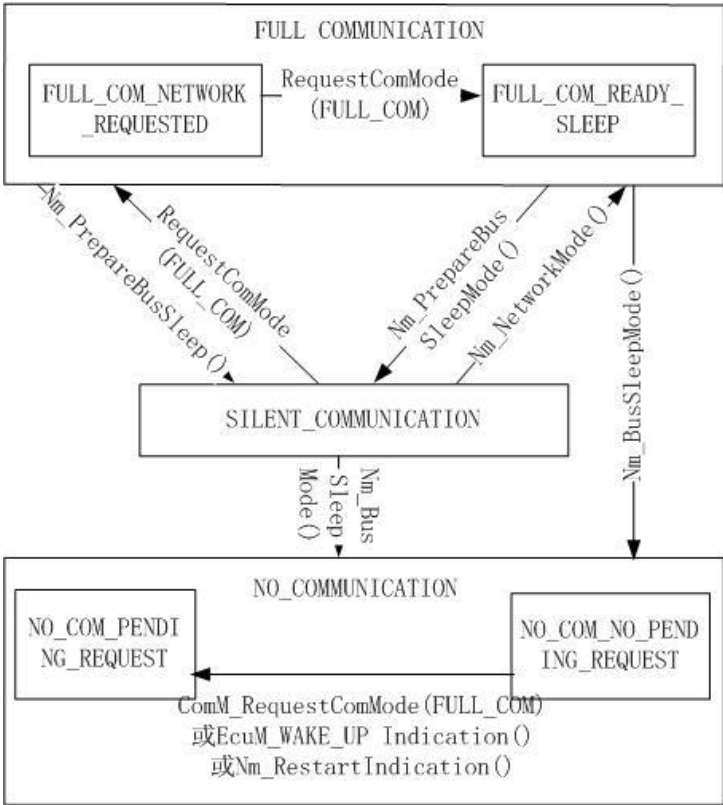


图 4-4 通信状态机转换

在 **SILENT_COMMUNICATION** 模式，收到 **FULL_COMMUNICATION** 模式请求之后，**ComM** 状态机将进入 **FULL_COMMUNICATION** 模式。如果在 **SILENT_COMMUNICATION** 模式下收到网络管理模块的总线睡眠指示，**ComM** 状态机将进入 **NO_COMMUNICATION** 通信模式。如果在通信模式 **SILENT_COMMUNICATION** 下收到网络管理模块的网络模式指示，**ComM** 将进入 **FULL_COM_READY_SLEEP** 状态。

当 **ComM** 状态机进入到 **FULL_COMMUNICATION** 模式之后，立即进入子

状态 FULL_COM_NETWORK_REQUESTED, 在 FULL_COMMUNICATION 状态, 如果收到网络管理模块的总线睡眠指示, ComM 状态机将进入到 NO_COMMUNICATION 模式。在 FULL_COM_NETWORK_REQUESTED 子状态, 如果没有用户通信请求, ComM 将进入到 FULL_COM_READY_SLEEP 状态。在 FULL_COM_READY_SLEEP 状态下收到网络管理模块的准备睡眠指示, ComM 状态机将进入到通信模式 SILENT_COMMUNICATION, 如果在 FULL_COM_READY_SLEEP 状态下收到用户的 FULL_COMMUNICATION 通信请求, ComM 状态机将返回到 FULL_COM_NETWORK_REQUESTED 状态。在 FULL_COMMUNICATION 通信模式下, ComM 模块允许在受影响的信道上进行消息的传输, 在 NO_COMMUNICATION 模式下将禁止在受影响的信道上进行消息的传输。当有一个以上的独立的用户请求 FULL_COMMUNICATION 通信模式时, ComM 模块将设定当前的通信模式为 FULL_COMMUNICATION 模式。

当 ComM 模块进入到 NO_COMMUNICATION 状态之后, 默认立即进入子状态 NO_COM_PENDING_REQUEST。在 NO_COM_NO_PENDING_REQUEST 状态下, 如果用户请求 FULL_COMMUNICATION 通信模式, ComM 状态机将进入 NO_COM_PENDING_REQUEST 在状态。如果在模式 NO_COM_NO_PENDING_REQUEST 下, 收到 ECUM 模块发送的 WAKE_UP 指示, ComM 状态机将进入到 NO_COM_PENDING_REQUEST 子状态。在 NO_COM_NO_PENDING_REQUEST 子状态下, 网络管理模块发送一个重启指示, 则 ComM 状态机进入子状态 NO_COM_PENDING_REQUEST。如果在子状态 NO_COM_PENDING_REQUEST 下没有收到 FULL_COMMUNICATION 模式请求, ComM 状态机将返回到 NO_COM_NO_PENDING_REQUEST 状态。

4.5 状态管理模块设计

状态管理模块主要负责网络的控制流抽象, 提供 API 给请求通信模式, 将逻辑上的状态转换成相对应的实际总线操作。其中, CAN 状态管理最主要的任务是处理 ComM 模块请求的网络模式转换和 CAN 接口模块通知的特定事件, 例如总线故障、模式标识。CAN 总线状态管理主要是将 ComM 模块的通信状态转换成总线的具体操作。为了与 ComM 保持一致, 将 CAN 总线状态划分为 CANSM_FULL_COMMUNICATION、CANSM_SILENT_COMMUNICATION、CANSM_NO_COMMUNICATION 三个状态, 依次与 ComM 模块中的三个状态相对应, 当 ComM 层的通信模式发生转换时, 总线状态管理模块的状态也随之发生改变。总线状态管理层的自动机转换及其条件如图 4-5 所示。

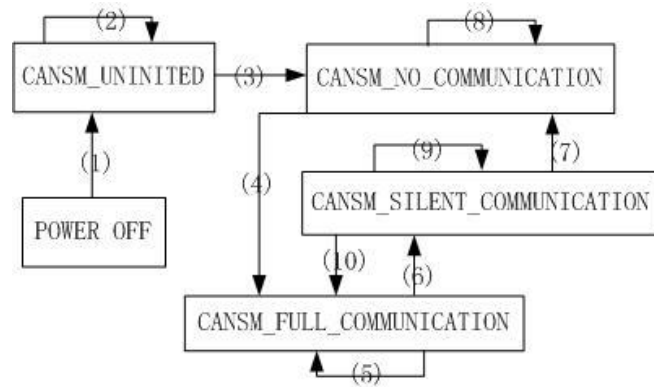


图 4-5 状态管理自动机

(1) 上电之后默认进入 CAN 状态管理未初始化状态 (2) ComM 层没有进行初始化相关操作 (3) 进行 CANSM 模块初始化之后，首先进入 CANSM_NO_COMMUNICATION 状态 (4) ComM 请求网络通信模式 FULL_COMMUNICATION，CANSM 模块即允许消息的发送也允许消息的接收 (5) ComM 层一直处于 FULL_COMMUNICATION 通信模式 (6) ComM 层请求进入 SILENT_COMMUNICATION 通信模式，CANSM 模块禁止发送消息，只允许接收消息 (7) ComM 模块进入 SILENT_COMMUNICATION 通信模式，CANSM 模块禁止消息的接收 (8) ComM 层没有通信请求，CANSM 维持在 NO_COMMUNICATION 通信模式 (9) 没有外部事件发生且 ComM 层进入模式 SILENT_COMMUNICATION (10) ComM 层请求 FULL_COMMUNICATION 通信模式，CANSM 开启消息发送功能。

4.6 CAN 接口和驱动设计

4.6.1 CAN 接口设计

CAN 接口模块包括的服务主要有初始化、传输请求服务、传输确认服务、接收指示服务、控制器模式控制服务、PDU 模式控制服务。

CAN 接口层的初始化主要包括全局初始化和控制器相关的初始化。全局初始化使用 CANIF_INIT() 函数，主要初始化所有可用的 CAN 硬件资源。控制器初始化采用 CANIF_INITCONTROLLER()，负责初始化连接在同一个网络上的 CAN 控制器以及消息接收和发送缓冲器。初始化结束之后，CAN 接口和 CAN 控制器处于停止模式，不能发送或接收 CAN 协议数据单元的。初始化过程只能在停止或未初始化模式执行。在系统重启后、CAN 接口层初始化之前处于未初始化模式。在停止模式，全局的初始化和与控制器相关的初始化都可以进行。如果初始化步骤是在启动状态被执行的，那么 CAN 接口会将模式转换到停止状态。

上层模块通过 CAN 接口的 CANIF_TRANSMIT() 函数发起传输请求，CAN 接口层将对收到的 L-PDU 进行编码，组装成符合 CAN 规范的数据帧格式之后

交给驱动层，CAN 驱动将 L-PDU 成功写入到 CAN 硬件设备时，意味着传输请求完成，然后驱动层的发送确认回调函数会通知接口层，数据已经发送成功，接口层再将发送确认一层层上传，直到初始发起传输的模块。

若在发起传输过程中没有可用的硬件资源，该传输请求将被挂起，L-PDU 将被临时存储于 CAN 接口层。如果等到上一个传输请求结束都没有可用的软件和硬件资源或者 CAN 接口初始化失败，则将拒绝该传输请求。

在成功接收到一个 CAN L-PDU 时，通过硬件过滤、等待上层通讯层读取接收到的数据之后，执行软件过滤、数据长度编码(DLC)核查、接收数据分派、选择合适的上层接收模块并将接收事件通知给相应的上层模块等操作，最后根据上层模块 AUTOSAR COM 的需求，设置相应的提示函数。CAN 接口层提供一个读取接收数据的函数 CANIF_ReadRxPduData()供上层模块读取从 CAN 网络接收到的 CAN L-PDU。

在执行传输请求时，CAN 接口层会将 L-PDU 连同相应的参数传递至 CAN 驱动，并通过 CAN 驱动将 CAN L-PDU 转发至相应的 CAN 控制器中。在接收时，CAN 接口层将从底层收到的 L-PDU 分发至上层模块。接收到的 L-PDU 具体发送至哪个上层模块是静态配置好的。在发送确认时，CAN 接口层负责通知上层模块发送动作已经成功执行。

CAN 控制器的状态由上层改变 CAN 控制器状态的函数来改变。该请求被 CAN 接口层验证后通过 CAN 驱动层的函数接口传递至各个 CAN 控制器，这些 CAN 控制器连在同一个 CAN 网络上。通过这种方式，所有这些 CAN 控制器可以依次被设置成睡眠模式或被唤醒。

当 CAN 控制器发出一个网络事件(总线中断，唤醒)的信号后，这一信号会被通知到相应的上层应用程序，在上层应用程序收到这些通知信号后，会执行相应的唤醒、恢复或关闭系统等操作。通过这种方式，上层使用这一函数的用户，如 ECU 状态管理器，COM 管理等，可以通恢复或唤醒步骤控制系统行为。

4.6.2 CAN 驱动设计

CAN 驱动层位于基础软件层的通信驱动模块之中，用于实现对硬件设备的访问，为上层模块提供独立于硬件的应用接口。CAN 接口是与 CAN 驱动交互的唯一模块。CAN 驱动为 CAN 接口提供初始化的消息传输服务，调用 CAN 接口层通知事件的回调函数。此外，CAN 驱动还负责控制属于同一个 CAN 硬件单元的所有 CAN 控制器的状态和行为。

在使用 CAN 驱动前，首先要通过初始化函数初始化 CAN 驱动模块。通用的初始化函数主要负责静态变量设置、CAN 硬件单元的一些常规设置、依据 CAN 控制器而设的不同的控制器设置等。CAN 驱动的初始化，是 CAN 接口层

在启动状态必须要完成的

在进行 L-PDU 的发送操作时，CAN 驱动负责将包含标识符和数据长度编码的 L-PDU 转换成与具体硬件匹配的格式，写入 CAN 控制器硬件单元中的缓冲区内触发数据传输操作；在进行 L-PDU 的接收操作时，CAN 驱动会调用接口层的接收指示回调函数，交由上层模块处理 L-PDU 的接收。

CAN 驱动提供控制 CAN 控制器状态的服务，总线关闭和唤醒事件都是通过回调函数通知相应模块的。

CAN 驱动的实现采用的是飞思卡尔的 MC9S08DZ60 微控制器。MC9S08DZ60 尺寸小、成本低、功能强大、资源齐全，具有很高的性能价格比，适合汽车的功能需求以及汽车的运行环境。其主要资源包括：一个 CAN 模块、一个串行外设接口 SPI 模块、两个串行通信接口 SCI 模块、多达 24 通道的 12bit 的 A/D 转换模块、一个基本时钟模块、60KB 的片上 Flash、4KB 的片上 RAM、2KB 的 E2PROM、看门狗定时器(COP Watch-dog)，另外还有 I2C 总线模块和多个定时器、计数器等。该控制器采用 PLL 锁相环技术，能够产生最高 40MHz 的总线频率，其独特的片上仿真/调试模块(BDC)更是大大简化了设计，从而确保了 MC9S08DZ60 在本设计中的应用地位。

CAN 总线收发器选用飞利浦公司 PCA82C250 产品。CAN 总线收发器是 CAN 协议控制器和物理总线之间的接口，该器件对总线提供差动发送能力并对 CAN 控制器提供差动接收能力，有很强的抗电磁干扰(EMI)的能力。

在 CAN 网络管理的具体实现过程中，采用第一个通道发送应用层消息，第二个通道发送网络管理消息，第三个通道发送诊断消息。这样能够避免报文的发送通道被其他类型消息占用，保证网络管理消息的正常发送，从而确保了系统通信的安全可靠。

4.7 本章小结

本章主要研究 CAN 网络管理方法。通过对需求结果的分析，结合 CAN 总线的事件触发的特性、CAN 网络节点地位均等等特点，在 CAN 通信协议的基础上，本章对 CAN 总线网络管理实现过程中网络管理模块、网络管理接口模块、通信管理模块、状态管理模块、CAN 接口和 CAN 驱动等模块进行了设计。

第五章 系统仿真与测试

5.1 仿真系统简介

CANoe 是一款集网络和 ECU 开发、测试、分析等多种功能于一身的专业工具，它支持总线网络从分析到实现的全过程。因此，在汽车行业、工程机械行业、轨道交通等领域得到广泛应用。CANoe 支持 CAN、LIN、MOST、FlexRay、J1708、以太网等多种总线系统和协议，其中 CAN 总线协议又包括 J1939、J1587、NMEA2000、ISO11783、CANopen、MCnet、GMLAN、CANaerospace 等，因此，CANoe 可以创建不同的总线网络并对整个网络的通信情况进行实时监测。CANoe 具有功能丰富和可配置选项众多等特点，由此受到开发工程师、网络设计工程师以及测试工程师的广泛使用和支持^[47]。CANoe 软件支持 CAPL 编程，用户可以通过 CAPL 编程语言对仿真节点进行功能实现和扩展。CAPL 使用类 C 语言和交互式的开发模式，采用事件驱动的工作方式，方便用户操作。

在开发初级阶段，可以利用 CANoe 建立仿真模型，并在此基础上进行功能评估。ECU 开发完成之后，仿真模型可以用于整个系统的测试、分析，以便尽早的发现问题并解决问题。此外，CANoe 包含测试功能集，能够简化或进行自动测试。使用该功能可以进行一系列的连续测试，并自动生成测试报告。CANoe 还具有诊断功能集，该功能可以使 CANoe 与 ECU 进行诊断通信^[48]。CANoe 支持 Vector 的所有硬件接口，适用于众多不同的 PC 接口和收发器，例如 PCMCIA、PCI 和 PXI 等，用户可以根据自身的需要进行选择，以获得最优秀的总线访问能力。

CANoe 功能十分强大，建立仿真模型之后，可以通过手动输入节点参数和导入数据库两种方式。同时 CANoe 支持两种仿真模式，一种是全部采用虚拟节点进行的全仿真模式，另一种是采用实际节点与虚拟节点混合的半仿真模式。

CANoe 支持的总线协议包括 CAN、LIN、MOST、FlexRay 和 J1708 等，能够完成生产和实践阶段的全部功能，包括建模、仿真、测试和分析。使用 CANoe 指导 ECU 开发过程，不但可以提早发现和修正错误，更可以根据评估结果获取系统架构的改进方案，给设计人员带来了极大的便利。此外，CANoe 包含的测试集功能大大简化了测试的难度，可以进行自动测试，或连续进行一系列测试并生成报告。

5.1.1 CANoe 组成

CANoe 功能强大，可以对总线进行测试、分析、仿真和记录，同时 CANoe 操作比较简单。它主要由主窗口、数据库编辑器（CANdb++ Editor）、仿真建立界面（Simulation Setup）、CAPL 编程浏览器（CAPL Browser）、CAPL 程序生成器（CAPLGenerator）和面板设计器（Vector Panel Designer）六个主要部分

组成，见如图 5-1。其中数据库编辑器主要负责创建和修改数据库，指定信号与消息的对应关系，对数据程序进行管理^[49-50]。在数据库编辑器中可以创建数据库，数据库包括节点，消息，信号，环境变量等信息，并通过对象的链接功能在网络节点、信号、消息之间建立链接。

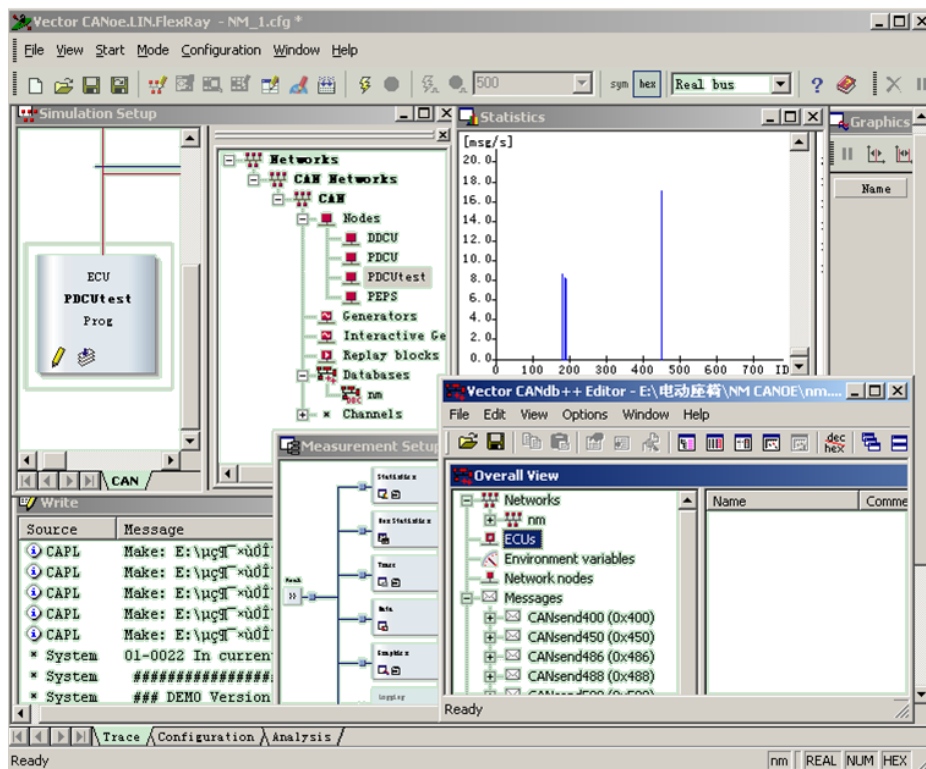


图 5-1 CANoe 界面

5.1.2 CANoe 基本操作

使用 CANoe 建立仿真模型的操作步骤如下：

- (1)启动 CANoe。
- (2)新建或者打开已有的配置文件，创建完成后保存后缀名为.cfg 的配置文件。
- (3)在仿真建立窗体中，点击 **Bus** 控件，右键选择硬件配置命令，在弹出的窗口中设置总线速率、通道、ID 滤波等参数。
- (4)在仿真建立窗体中点右键，在总线上创建需要的网络节点。
- (5)进入数据库编辑器，创建节点需要的信号、消息和环境变量，并将它们与网络节点进行关联，保存数据库。
- (6)在 CAPL 编程浏览器、程序生成器部分编写并编译仿真程序。
- (7)编译、运行、数据分析。

5.2 仿真系统架构

为验证改进之后的 AUTOSAR 网络管理方法能够满足汽车网络的要求，基于 CANoe 软件建立 CAN 总线的网络管理仿真系统。该系统能够实现本文前述网络管理的基本功能，并通过系统仿真分析运行结果，判断是否满足汽车 CAN 网络的基本要求，能否实现网络管理功能。

CANoe 软件主要设计了仿真系统中与 CAN 总线连接的各个仿真节点。在仿真过程中，数据库设计采用 CANdb++ Editor，然后通过 CAPL 语言建立 CAN 网络通信模型，最后通过 CANoe 仿真工具中的跟踪 Trace 窗口观察仿真结果，并对总线上的数据进行分析，验证本文提出的网络管理方法的可行性。

为了验证本文提出的改进方法，在 CANoe 仿真软件中建立虚拟节点和真实节点相结合的混合仿真系统，从节点故障检测、总线故障处理和协同睡眠三个方面进行仿真。仿真环境及网络拓扑结构如图 5-2 所示。其中 4 个虚拟节点分别为 IG2(右车灯控制节点)、PEPS（点火控制节点）、PEDS（左车灯控制节点）、DDCU（左门控制节点）。PDCUtest 为实际节点，采用飞思卡尔的 8 位微处理器 MC9S08DZ60 设计。网络管理中使用的定时器的定义见表 3。

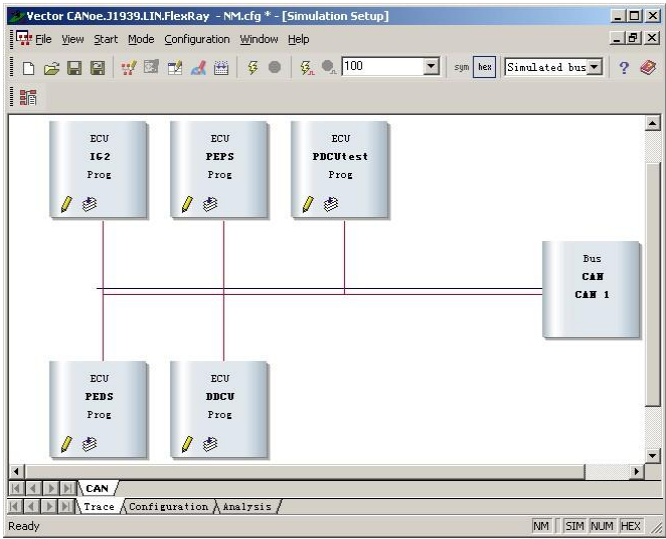


图 5-2 仿真及其拓扑结构

表 5-1 定时器参数

定时器名字	含义	定时时间
CANNM_REPEAT_TIME	在 REPEAT 状态停留时间	5000ms
CANNM_MSG_WAIT_BUSSLEEP_TIME	进入睡眠状态之前的等待时间	15000ms
CANNM_MSG_TIMEOUT_TIME	消息超时	1500ms

5.3 仿真系统设计

5.3.1 数据库建立

CAN网络使用CANdb++ Editor作为数据库支持工具，如图5-3所示。根据仿真要求，每个仿真节点都有网络管理消息和应用消息，具体设计如下：DDCU节点的应用消息DDCUapp和网络管理消息DDCU；PEPS节点的应用消息PEPSapp和网络管理消息PEPS；PEDS节点的应用消息PEDSapp和网络管理消息PEDS；IG2节点的应用消息IG2app和网络管理消息IG2。

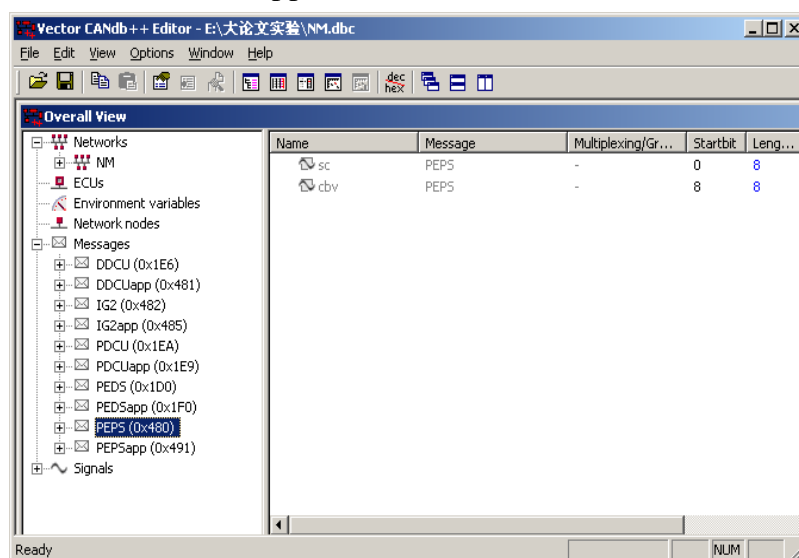


图 5-3 CANdb++ Editor 工具

5.3.2 软件设计

在CAN网络的仿真实验中，节点采用4.1节设计的状态转换图，其中最主要的是网络节点从NMNormal状态到BUS_SLEEP状态的转换。当仿真节点在NMNormal状态接收到其他节点发送的网络管理消息时，根据接收到的网络管理消息的类型以及自身的通信需求，进行相应的处理。图5-4是该网络状态转换的流程图。

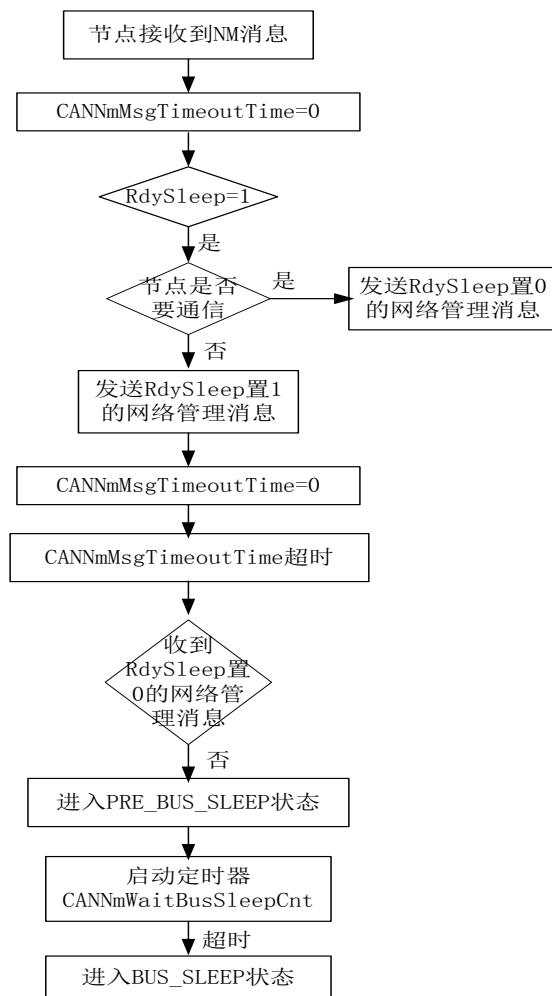


图 5-4 NMNormal 状态接收过程图

5.4 仿真过程及结果

仿真系统搭建结束后，可以进行测试实验。点击CANoe软件界面上的运行按钮，能够从Trace窗口中看到网络中各个节点消息的收发情况。对改进的网络管理功能的测试，主要分为四部分：正常通信测试、故障节点测试、总线故障测试和协同睡眠测试以及LimpHome状态测试。具体测试及结果如下。

5.4.1 节点故障检测

图5-5和图5-6分别是网络正常通信和节点故障通信的情况。在正常通信情况下，节点发送普通的网络管理消息。按下按键'k'，节点DDCU退出网络，不进行通信，此时网络中的其它节点根据故障检测方法检测出该节点发生故障。

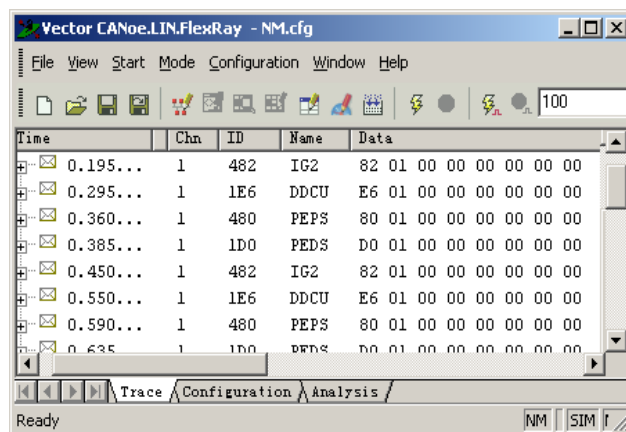


图 5-5 通信正常

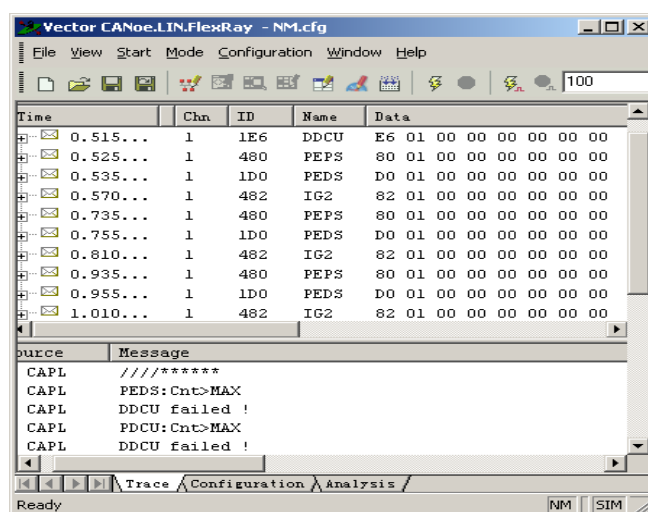


图 5-6 节点故障通信

5.4.2 总线故障

改进的网络管理方法能够及时发现由于总线故障导致的网络通信异常，并采取相应的处理措施。在正常通信情况下，按下按键'i'，网络中除了 PEDS节点外都不能通信，PEDS节点发送消息失败进入LIMPHOME状态，并在该状态下不断发送LIMPHOME消息，如图5-7所示。

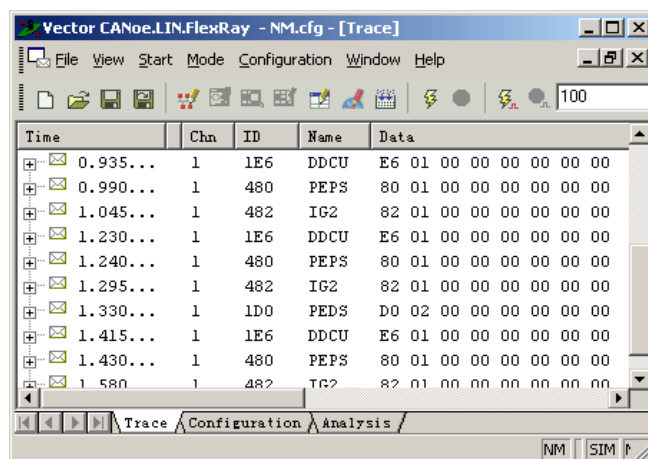
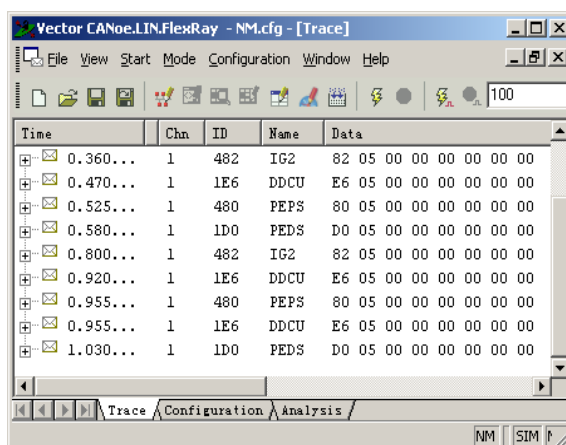


图 5-7 总线故障通信

5.4.3 协同睡眠

协同睡眠是汽车网络管理的一个重要方面。改进后的网络管理睡眠机制不但能够协同网络中的节点同时进入到睡眠模式，而且还能够及时发现由于节点故障而被误认为是请求睡眠的情况，按键'o'被按下，即节点PDCU想要进入休眠状态，此时网络中有其它节点需要通信，网络不能进入睡眠状态，请求睡眠的节点继续等待。当按下'q'，意味着网络中的其它节点同意睡眠，网络进入睡眠状态，见图5-8。



Time	Chn	ID	Name	Data
0.360...	1	482	IG2	82 05 00 00 00 00 00 00
0.470...	1	1E6	DDCU	E6 05 00 00 00 00 00 00
0.525...	1	480	PEPS	80 05 00 00 00 00 00 00
0.580...	1	1D0	PEDS	D0 05 00 00 00 00 00 00
0.800...	1	482	IG2	82 05 00 00 00 00 00 00
0.920...	1	1E6	DDCU	E6 05 00 00 00 00 00 00
0.955...	1	480	PEPS	80 05 00 00 00 00 00 00
0.955...	1	1E6	DDCU	E6 05 00 00 00 00 00 00
1.030...	1	1D0	PEDS	D0 05 00 00 00 00 00 00

图 5-8 节点同步睡眠

5.5 本章小结

本章以CANoe软件为平台，根据仿真系统的要求，建立CAN网络数据库，采用CAPL语言实现CAN总线系统的网络管理功能，并用该仿真系统实现了第四章提设计网络管理方法。通过仿真系统验证了网络管理方法的可行性。

第六章 总结与展望

6.1 工作总结

汽车总线技术在汽车工业的发展中发挥重要作用。使用汽车总线技术不仅提高了汽车性能，降低了生产成本，而且还改变着传统汽车的设计理念。由国内外各个汽车生产厂商、整车制造商、零部件供应商制定的OSEK、AUTOSAR等软件开发标准得到了广泛的使用。我国汽车电子起步晚，开发应用水平都处在起步阶段，对于国外先进的汽车软件开发方法和技术研究较少。

车载网络技术在汽车中的大规模应用改变了ECU之间的通信方式，车载网络的通信安全和可靠性成为一个重要的研究内容，因此需要对汽车总线的网络管理进行研究。本文主要针对汽车CAN总线的网络管理方面进行分析研究。具体工作如下：

- (1). 深入研究AUTOSAR标准的基本体系架构，对网络管理相关模块进行了重点分析，结合车载网络中ECU和总线的特点，确定网络管理的主要功能，为车载网络管理功能的实现奠定基础。
- (2). 研究了OSEK和AUTOSAR两种网络管理方法，分析两种方法的基本工作原理，对两种方法的优缺点进行比较，分析比较CAN总线较其他汽车总线的主要优点，并在此基础上提出基于AUTOSAR标准的CAN总线网络管理主要应该解决的问题和研究思路。
- (3). 本文针对AUTOSAR标准在网络管理睡眠过程、功能检测等方面的不足，提出一种改进的AUTOSAR网络管理方法，并给出了在CAN总线上的实现方法。
- (4). 利用CANoe软件建立仿真环境，对本文提出的CAN总线的网络管理方法进行测试。仿真结果表明，本文设计的网络管理方法是可行的。同时，也为进一步设计更加完善合理的方案奠定基础。

6.2 进一步工作

本文对CAN网络管理方案下的节点通信和休眠状态下进行了初步研究。AUTOSAR标准是近年来新兴的汽车行业的软件架构标准，AUTOSAR方法学是一门高深的学问，由于本人的知识水平和研究经历有限，本文设计的方案仅是针对整个架构的一部分，对很多细节方面的技术考虑不周，存在许多不足和缺陷，还有很多地方需进一步完善。

首先，CAN总线的网络管理方法还需进一步研究。本文提出的网络管理方法虽然能够实现一般的网络管理的基本功能，但这种方法缺乏完整性，需要进一步研究CAN网络的特点，并结合网络管理的要求，完善CAN网络管理。

其次，本文以AUTOSAR网络管理为基础，研究CAN总线在该标准下的改

进和实现，但是在实际应用过程中还涉及到应用消息、网络管理消息两者与总线负载的关系。在本文中还未加以研究。

最后，本文研究CAN总线的网络管理，研究对象比较简单。由于当前汽车内部使用的总线类型繁多，各种总线之间的节点能否进行有效的管理尚值得研究。

参考文献

- [1] 付亮,李伟.CAN 总线技术及其在现代汽车中的应用.中国汽车制造,2006(07): 23—26.
- [2] 罗峰,苏剑.汽车网络与总线标准[J].汽车工程,2003,25(4):372-376.
- [3] Rudolf Haug. 汽车网络总线技术发展研究.世界汽车技术发展,2008,143-144
- [4] 崔代福,魏学哲,孙泽昌,赵格英.LIN 协议及其在轿车车身控制中的应用[J].自动化与仪表,2002,6(17):4-6.
- [5] 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,2006:14-36.
- [6] 阎树田,黄新春,康会峰.基于 Flexray 总线的嵌入式汽车线控制动技术研究[J].计算机测量与控制,2009,17(3):487-488
- [7] 中国科学技术协会.车辆工程学科发展报告.第一版.北京:中国科学技术出版社,2008,142-143
- [8] 李茗.汽车电子产品的开发.汽车工程,2004(26):367~372
- [9] 杨艳玲,韩莉.车载网络的应用现状及发展趋势.科技资讯,2008,(17):89-89
- [10] Application note. Determination of Bit Timing Parameter for the CAN ControllerSJA1054. Products for CAN Application. 1998
- [11] 高青春.基于 CAN/LIN 总线的汽车车身通信网络的研究及应用[D].重庆:重庆大学,2006,30-35
- [12] D.John.OSEK/VDX History and Structure[C].IEEE Seminar.1998,20:1-2
- [13] 姚敏.参照 AUTOSAR 标准的汽车电子通信与应用:[浙江大学硕士学位论文].杭州:浙江大学,2008,25-47
- [14] 苟广鹏. 控制器局域网(CAN)在汽车中应用的基础研究.同济大学硕士论文.2000,10-11
- [15] MOST Cooperation, MOST High Protocol Specification Rev.2.1[EB/OL]. 2001.01
- [16] 章亮飞,李银国.嵌入式实时操作系统 AutoOSEK 的设计[J].计算机工程,2007,33(16):53-55.
- [17] 熊毅,郭杏荣,张倪.基于 OSEK 规范的网络管理的研究与改进.计算机应用研究[J].2007,9:217
- [18] 杨凯,孙晓民.TH OSEK 车用网络管理系统设计与实现.计算机工程与设计[J].2008,9:4395
- [19] 李佳,朱元,田光宇.CAN 与 TTCAN 通信延迟时间的分析.清华大学学报[J].2006,46:2

- [20] 郭杏荣.OSEK 研究及基于 Hopen 的实现:[中国科学院计算机研究所].北京:中国科学院计算机研究所,2006.4
- [21] B.Emaus.Introduction to OSEK Network Management[J].Society of Automotive Engineers.2000,1:15-24
- [22] 王锴,王宏,徐皑东.下一代车载网络 FlexRay 及其应用研究[J].计算机工程与应用,2008,44(20):20-24.
- [23] 高美芹,明平顺,周明应.车载网络 FlexRay 及其在线控制制动系统中的应用[J].武汉理工大学学报,2007,29(1):78-81.
- [24] 陈筠翰,刘衍珩,曲良东等.车载 CAN 网络中直接 NM 逻辑环的实现[J].计算机工程,2010,36(13):114-116.
- [25] F.Laqrade,A.Radermacher,S.Robert,S.Gerard,D.Servat. Issues in mapping CORBA component model to OSEK[J].Communication of the ACM.2005,8:434-437
- [26] K.M.Zuberi,P.Pillai,K.G.Shietal.EmERALDS-OSEK: A Small Real-Time Operating System for Automotive Control and Monitoring[C].Society of Automotive Engineers Congress and Exposition. 199911:1-7
- [27] The OSEK Group.OSEK/VDX System Generation OIL:OSEK Implementation Language.2.5.2004
- [28] 周海娟.参照 AUTOSAR 标准的总线通信协议栈的设计与实现:[浙江大学硕士学位论文].杭州:浙江大学,2008,22-43
- [29] The AUTOSAR Group .AUTOSAR Technical Overview.2.2.1.2008
- [30] Stefan Roget .AUTOSAR and the Automotive Tool Chain[C].EDAA 2010.
- [31] Jorgen Moessinger. AUTOSAR.The Standard for Global Cooperation in Automotive SW Development[C].ATI 2008.
- [32] Heinekens H.Automotive Open System Architecture-An Industry Wide Initiative to Manage the Complexity of Emerging Automotive E/E Architectures[C].SAE Paper,2004-21-0042.
- [33] 方遒,姜瑜涛.汽车电子基础软件标准体系研究[J].信息技术与标准化,2011.08.
- [34] AUTOSAR GbR.AUTOSAR Specification of CAN Interface Layer 2.0.1.AUTOSAR Administration,2006.
- [35] AUTOSAR GbR.AUTOSAR Specification of CAN Driver Layer 2.0.1.AUTOSARAdministration,2006.
- [36] 高焕吉.基于 AUTOSAR 的汽车电子控制系统嵌入式软件开发[J].汽车电器,2010,(05)
- [37] AUTOSAR GbR.AUTOSAR Diagnostic Communication

- Manager2.0.1.AUTOSARAdministration,2006.
- [38] AUTOSAR GbR.AUTOSAR Specification of RTE Software 2.0.1. AUTOSAR Administration,2006.
- [39] POLEDNA STEFAN. The Time-Triggered Communication Protocol TTP/c[M].1998.
- [40] Michigan Garson, Stefan John, Bernd Deutschmann.Methodology to predict EME effects in CAN bus system using VHDL-AMS.IEEE Transactions on electromagnetic compatibility.2008,50
- [41] 徐华,王耀南,陈洁平等.混合电动汽车中的 CAN 总线研究与应用[J].自动化技术与应用.2003,22(4):33-35
- [42] FlexRay Consortium. FlexRay Protocol Specification.V2.1 rev A.2005
- [43] 罗端,李红,方正,邓俊,胡琦,唐凯.基于 AUTOSAR 的汽车电子诊断系统的开发[J].汽车工程,2012,(02)
- [44] S.Suzuki,W.Nagaura,T.Imai,S.Kuragaki,T.Yokoyama,K.G.Shin.A Distributed Control System Framework for Automotive Power train Control with OSEK Standard and CAN Network. Society of Automotive Engineers Congress and Exposition[C]. SAE International Congress&Exhibition.1999,44:01-1276
- [45] Marko H H, Dirk G. FlexRay NM Hanser Automotive[J]. Sensing and Control. 2007,2:1-7
- [46] 徐鑫朋,王翔,陆建华,徐军,张晓先.基于 AUTOSAR 方法论的应用组件配置[J].计算机工程,2010,(18)
- [47] 丁志华,罗峰,孙泽昌.基于 CANoe 的汽车故障诊断系统研制[J].汽车工程.2007,29:449-452
- [48] 谢雪松 胡长阳 . 基于驱动程序的协议栈设计 [J]. 《电子技术应用》.2000,10:17-19
- [49] 张东来.基于单芯片以太网协议栈的远程环境监测系统[J].仪器仪表学报.第 24 卷第 3 期增刊.2003,8:537-539
- [50] 林凯,罗禹贡,杨殿阁,李克强,连小珉.基于 CANoe 的混合动力电动车 TTCAN 建模与仿真[J].汽车工程.2007,29:1032-1035

攻读硕士学位期间发表的论文

- [1] 张建军,于萍,张本宏.基于带宽利用率的 Flexray 静态段研究[J].计算机应用研究.2012,29(12):4574-4576
- [2] 一种改进的 AUTOSAR 车载网络管理方法,电子测量与仪器学报,已录用。

读研期间参与的科研工作

- [1]项目名称：具有远程监控功能的汽车智能多节点网络系统
时间：2011.11-2013.4
- [2]项目名称：基于时间隔离的汽车实时网络防危调度研究
时间：2011.11-2013.4

特别声明

本学位论文是在我的导师指导下独立完成的。在研究生学习期间，我的导师要求我坚决抵制学术不端行为。在此，我郑重声明，本论文无任何学术不端行为，如果被发现有学术不端行为，一切责任完全由本人承担。

学位论文作者签名

签名日期：