

天津大学
硕士学位论文
基于CANopen协议模块化CAN控制器的研究
姓名：刘莉
申请学位级别：硕士
专业：控制理论与控制工程
指导教师：吴爱国;黄瑞祥
20030101

摘 要

现场总线适应工业控制系统的分散化、网络化和智能化的要求,促使目前的自动化仪表、集散控制系统等产品面临体系结构和功能结构的重大变革,导致了工业自动化产品的又一次更新换代,已经成为世界范围内研究的热点。对现场总线技术的协议,应用和网络实时性问题进行研究,具有较大的理论和实践意义。

本文以 CAN 总线为研究对象,设计开发了具有较强通用性的模块化 CAN 总线控制器;以 CANopen 协议为标准编写并在软件实现了 CAN 总线应用层协议;并对由此组成的 CAN 总线控制系统的实时性做了初步的研究。

该 CAN 总线控制器采用二级总线的结构形式,即系统总线和 I/O 总线分离,将其分为 CAN 节点控制模块和 I/O 模块两个相互独立的部分,通过 485 总线进行数据通讯。用户可根据需要灵活组合 I/O 模块形成 I/O 端口,组成完整的 CAN 节点,其结构灵活,通用性强,每个 CAN 节点可带的 I/O 端口增多,使得系统中 CAN 节点相对减少,系统实时性大大增强。本文在硬件上设计开发了 CAN 节点控制模块和具有四个输入输出通道的独立的 I/O 模块(包括 DI、DO、AI 和 AO 四种类型)。

本文对 CANopen 协议进行分析,并根据常规控制系统中通讯和网络管理的需求,以此为标准编写了 CAN 通讯的应用层协议。软件部分由单片机 C 语言 C96 和 C51 编写,实现此应用层协议,完成各节点之间的 CAN 通讯以及节点控制模块与各 I/O 模块之间的 485 通讯。由于主节点和从节点的任务不同,软件功能也有很大的不同,因此分开考虑,在软件上分开实现。在整个 CAN 总线控制系统中,与上位机相连的 PC 适配卡作为主节点,主要完成网络设置和管理功能;其它为从节点,本设计中主要实现从节点 CANopen 协议中主要的 CANopen 通讯和网络管理功能。

最后,从 CAN 总线应用层的角度初步研究讨论了 CAN 总线控制系统的实时性问题,给出了以 CANopen 应用层协议为基础的 CAN 信息通讯最大延迟时间的计算方法。

关键词: 现场总线

CAN 总线

CAN 总线控制器

CANopen 应用层协议

ABSTRACT

With the control systems becoming decentralized, networked and intelligent, the technology of fieldbus has become the research focus all over the world. The research on fieldbus protocol, application and real-time has great theoretical and practical significance.

The thesis designs and develops a universal modularized controller based on CAN bus. The software of CAN application layer based on CAN standard is implemented and the real-time of the control system is studied.

The CAN controller is designed in two-level structure, i.e. system bus and I/O bus. The CAN controller node and the I/O modules are designed individually and they communicate with each other by RS-485. The users can constitute the I/O modules flexibly according to different needs. Each CAN controller node can configure with many I/O ports, as a result, the number of CAN nodes is decreased and the real-time is increased. The thesis mainly designs the hardware including CAN nodes and four independent I/O modules(DI, DO, AI and AO).

The thesis analyzes CANOpen protocol and programmed the application layer protocol in accordance with the needs of the communication and network management of control systems. The software is designed in microcontroller C language: C96 & C51. The main function of application protocol software is the communication among CAN nodes and communication between CAN nodes and I/O modules. In the control system, the master node is the adapter connected to the PC, responsible for the network configuration and management. All other nodes are slave. Since the master node and slave nodes have different tasks, their software is designed separately. The thesis mainly designs the CANOpen communication and network management software.

At last the real-time of the control system is discussed from the view of CAN application layer. The thesis also gives the computation of the maximum delay of CAN data.

Keywords: fieldbus, CAN bus, CAN controller, CAN application layer protocol

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得天津大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：刘新 签字日期：2003年1月15日

学位论文版权使用授权书

本学位论文作者完全了解 天津大学 有关保留、使用学位论文的规定。特授权 天津大学 可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：

导师签名：吴晋国

签字日期： 年 月 日

签字日期：2003年1月11日

第一章 绪 论

1.1 课题相关技术的现状和发展方向

随着控制、计算机、通信、网络等技术的发展,信息交换沟通领域正在迅速的覆盖从工厂的现场设备到控制管理的各个层次。信息技术的发展,引起了自动化系统结构的变革,逐步形成了以网络集成自动化系统为基础的企业信息系统。现场总线技术就是顺应这种形势发展起来的。

现场总线控制系统(FCS)既是一个开放的通讯网络,又是一个全分布的控制系统。由于现场总线适应了工业控制系统向分散化、网络化、智能化的发展方向,它一经产生就成为全球自动化领域的热点,受到了普遍的关注。它的出现,对该领域的发展产生重要的影响。

1.1.1 现场总线

一、现场总线的定义^{[6][7]}:

现场总线是用于现场仪表与控制室系统之间的一种开放、全数字化、双向通信与多站的通信系统。现场总线在企业网(Intranet)内仅处于最低层的控制系统(Infranet)。遵循 ISO 的 OSI 开放系统互连参考模型的全部或部分通讯协议。FCS 则是用开放的现场总线控制通讯网络将自动化最底层的现场控制器和现场智能仪表设备互连的实时网络控制系统。

二、现场总线的特点及发展现状:

1、现场总线的特点^{[1][3]}:

- (1) 具有开放性;
- (2) 全数字化;
- (3) 双向通信;
- (4) 可智能化现场仪表。
- (5) 是一种实时性网络,具有传输速度快,响应时间短,巡回时间短的特点。

2、应用现场总线所带来的优势^[3]:

- (1) 可以减少 1/2~2/3 的隔离器、端子柜、I/O 终端、I/O 卡件、I/O 文件及 I/O 柜子,这样就节省了 I/O 装置及装置室的空间;
- (2) 由于免去了 D/A 及 A/D 转换,使精确度可以从 $\pm 0.5\%$ 提高到 $\pm 0.1\%$;可以减少电缆的用量,费用可以节省 66%或更多;
- (3) 可以产生先进的、智能化现场仪表,如多变量变送器和执行器,使维修预报(predicted maintenance)成为可能;
- (4) 由于可以将 PID 功能植入变送器或执行器中去,使控制周期大为缩短。

目前可以从 DCS 的每秒调节 2~3 次增加到每秒调节 10~20 次,从而改善了调节性能;由于免除了主机入口处的瓶颈现象,既可以提高系统的安全性和可靠性,同时也可使主机腾出手来从事优化等工作以提高效益;

(5) 组态简单,安装、运行、维护简便,用户可以择优选择,达到最佳集成。

现场总线的出现,给控制领域带来了又一次革命,其深度与广度将超过历史上的任何一次变革,将开创自动化的新纪元。基于现场总线的控制系统有最终取代传统 DCS 的趋势。

3、现场总线的发展现状^{[1][2][3]}:

现场总线发展迅速,现处于群雄并起、百家争鸣的阶段。目前已开发出 40 多种现场总线,已通过 IEC 审核和投票表决作为 IEC 国际标准的共有 12 种,而目前较常用的现场总线有基金会现场总线 FF、CAN 总线、LonWorks 总线、PROFIBUS 总线和 HART 总线几种。它们各自有各自的特点和应用领域。

虽然现场总线有很多的优势,但由于它刚刚起步,还存在很多问题^[3]:

(1) 目前存在的现场总线的种类繁多,始终没有一种总线能覆盖所有的应用面,解决应用中出现的所有问题,占据主流。多标准就等于没有标准,这使用户在选择时遇到很大困难。

(2) 在现场总线的应用中存在着三个瓶颈:当总线因故障切断时,系统有可能产生不可预知的后果;在本安防爆应用中,现有的防爆规定限制总线的长度和总线上所挂设备的数量限制了现场总线节省线缆优点的发挥;系统组态参数过分复杂,不容易掌握,如果组态参数设定不当,对系统性能影响很大。

(3) 由于现场总线种类繁多,异种现场总线之间不能进行互操作,使得现场总线的互操作性难以实现。

(4) 许多现场总线的协议不公开,使用户的开发使用受到很多限制,必须始终依赖国外技术。而且相应的硬件,软件和开发工具都十分昂贵,许多企业由于资金不足的原因无法采用现场总线,从而影响了现场总线的推广。

(5) 由于现场总线标准本身尚在发展中,从而给产品的开发和测试带来难度。这在一定程度上造成产品开发商、生产商少,产品品种单一而且价格昂贵。

(6) 由于熟练掌握应用技术和开发技术的人才少和现场总线本身的复杂性,使得在现场总线的开发,调试、运行和维护遇到很多困难,大多数企业在大系统、关键系统中不敢用现场总线,现场总线的优势无法体现。

三、我国现场总线的应用情况和发展趋势^{[1][2][5]}:

现场总线技术在中国正在健康地发展,中国已经成为各种现场总线激烈争夺的重要战场,而且一些总线在中国的推广已取得了一定的成绩。但在现场总线的应用中依然存在着以上提到的问题。在多种总线共存的格局已在国际和国内成为

定式的情况下，我国的做法是选择协议真正开放，在技术上有作为的，支持国内生产各类装置的企业开发总线产品的现场总线，建立一个具有预测性、开放性、动态性的中国现场总线标准体系。让各种现场总线在各自适合的领域为我国服务，并在实践中逐步解决现场总线应用中所存在的问题。

1.1.2 CAN 总线

一、CAN 总线概述^[7]：

CAN (Control Area Network) 即控制器局域网，是由德国的 BOSCH 公司首先提出的用于汽车控制、并有效支持分布式控制和实时控制的串行通讯网络。它是一种多主总线，通信介质可以是双绞线、同轴电缆或光导纤维。通信速率最高可达 1Mbps。CAN 总线的技术规范包括 CAN 2.0A 和 CAN 2.0B 两部分，CAN 协议只有物理层和数据链路层两个底层协议，规定了对数据通讯的成帧处理，包括位填充、数据块编码、循环冗余检验、优先级判别等工作标准。

二、CAN 总线的特点及优势^{[7][8]}：

由于 CAN 总线采用了许多新技术及独特设计，CAN 总线的数据通信具有突出的可靠性、实时性和灵活性。已成为国际标准化组织 ISO1898 标准，其具体如下特性：

1、CAN 采用多主方式工作，网络上任意节点均可以在任意时刻主动地向网络上的其它节点发送信息，而不分主从，通讯方式灵活；

2、CAN 信息帧采用短帧结构，每一帧的有效字节数为 8 个，这样传输时间短，受干扰的概率低；

3、CAN 协议废除了传统的站地址编码，而对通讯数据块进行编码，使网络中的节点个数在理论上不受限制，网络中的不同节点同时接到相同的数据，使总线上传输的信息总量减少；

4、CAN 网络上的信息可分成不同的优先级，满足不同的实时性要求；

5、CAN 采用非破坏性总线裁决技术 (CSMA/CD)，大大节省了总线冲突裁决时间；最重要的是在网络负载很重的情况下也不会出现网络瘫痪情况(以太网则可能)；

6、CAN 网络具有点对点、一点对多点 and 全局广播等几种通讯方式；

7、具有极好的检错效果，CAN 的每帧信息都具有 CRC 校验和其他检错措施，保证了错误的输出率极低；

8、CAN 的直接通讯距离最远可达 10km (速率 5kbps 以下)；通讯速率最高可达 1Mbps (此时距离易长 40m)；最多可接节点达 110 个。

三、CAN 总线存在的不足：

1、CAN 总线中没有路由器，网关等网络连接设备，网络规模有限，大型组

网能力和网络处理能力差。

2、较适合于小型系统，当用于节点数较多的大型系统时，性能将会下降。

3、由于 CAN 总线是针对相对较少信息的发送而设计优化的一种串行通讯协议，传输大数据量的能力较差。

4、由于 CAN 协议的物理层和数据链路层已被封装在硬件中，核心内容不透明，人为对底层协议的开发和处理的余地小。

5、与其它的现场总线协议和以太网 TCP/IP 协议的互操作能力差，限制了网络规模的扩大和通过 TCP/IP 与异地网络的信息共享。

四、CAN 总线的应用领域：

由于具有以上特性，CAN 总线越来越受到人们的重视，目前已有许多大公司的产品采用了这一技术。CAN 总线的应用范围很广，其中汽车和交通领域占应用总数的 80%以上，还用在工业控制领域，楼宇自动化，机器人领域，装载器械，嵌入式网络，混合引擎控制，和医学电子，电话系统，模拟战争，高能量物理等一些特殊行业。目前，支持 CAN 协议的有 Intel, Motorola, Philips, Siemens, NEC, Silioni, Honeywell 等百余家国际著名大公司。

总之，CAN 总线是现场总线中比较有前途的总线之一，受到了越来越广泛的重视。

1.1.3 CANopen 高层协议

由于 CAN 本身并非一个完整的协议，只包括物理层和数据链路层两个底层协议，要进行高效率的通讯还需要进一步开发高层协议。目前已有几种已经成为标准的高层协议，比如 CANopen、Devicenet、CAL、CAN kingdom、SDS、SAE J1939。其中 CANopen、Devicenet 两种应用较多。

一、CANopen 高层协议：

CANopen 是由从事工业控制的 CiA (CAN in Automation) 的会员开发的基于 CAN 总线的应用层协议。简单的说，CANopen 就是将 CAN 网络中通讯数据的组建和传输进行标准化，并给用户很大的自由度，允许用户随时添加自己所需要的新功能。CANopen 协议主要有以下内容^{[23][24]}：

(1) 以对象字典为核心概念，用来定义设备和通讯的相关参数和与各类操作之间的接口，其它操作均需对对象字典进行访问；

(2) 针对数据流量需要和实时性要求的不同，定义了用于过程数据通讯 (PDO) 和服务型数据通讯 (SDO) 两种主要的通讯模式；

(3) 定义了状态管理、节点保护和特殊标识符分配 (DBT) 等网络管理服务 and 应急对象、时间标记同步信息 (SYNC) 预操作对象。

CANopen 高层协议具有以下特点：

(1) 它的最突出的特点是：它既支持管理控制层的数据传输，同时也兼容同一物理层比较小的传感器和执行器。这就避免了传感器执行器与高层通讯之间网关的不必要的花费；

(2) 既支持对设备参数的直接读取，也支持实时过程的数据通讯；

(3) 既支持周期性通讯也支持事件触发通讯，这可以最大可能的减小总线上的负载，同时保证最小的响应时间，使 CANopen 网络在波特率较低的情况下也可以获得比较好的通讯特性，特别适合于实时性的工业自动化；

(4) 在 CAN 总线系统的所有硬件设备不变的情况下，能够开发出 CAN 所有的潜在的功能，而对物理层的硬件连接无特殊的要求。

二、CANopen 的国内外研究动态：

CANopen 应用层协议在国外已有很多方面的应用，受到了足够的重视。在欧洲，CANopen 协议已被广泛的应用于医疗装置中，并进一步扩展应用到保安控制系统中；在美国，CANopen 协议已经成为装载机械和公共运输设备的协议标准，同时也应用于嵌入式系统的控制。目前，国外已有许多大公司开发了 CANopen 软件和硬件产品，比如：Northampton 公司的 CANopen 开发工具，Downers Grove 公司的 CANopen 控制模块，Elkhart 公司的 CANopen 的开发工具和软件代码；还有一些公司开发了 CANopen 协议的组态软件和配套的硬件下载工具，比如：MicroControl 公司的 u CAN.Open.er 和 Philips 公司的 CANopenIA Developer's Kit。

在我国 CAN 的应用层协议主要应用 DeviceNet，很少用到 CANopen，因此 CANopen 协议需要在我国进一步推广。

1. 2 课题的提出及意义

1. 2. 1 课题的提出

一、课题的提出：

现场总线技术是控制领域的新热点，是工业现场级设备通信的一场数字化革命，代表了自动化的发展方向。在我国，现场总线技术已受到了广泛的关注，并有了很大的发展；但依然存在着很多问题，大多数企业在大系统、关键系统中不敢用现场总线，很大程度上影响了现场总线的推广。因此本论文选择一种有前途的现场总线进行开发研究，掌握开发应用现场总线的关键技术，并尽力解决现场总线应用中存在的一些问题。

二、课题选择 CAN 总线为研究的对象的原因：

1、CAN 总线是由德国的 BOSCH 公司于 80 年代提出，并已在汽车及其它各个领域应用，是一种发展成熟的现场总线，其总线规范已被 ISO 国际标准组织制定为国际标准。

2、CAN 总线设计独特，可靠性高，被公认为是最有前途的现场总线之一。

3、CAN 总线应用十分广泛，且应用的数量和范围有不断增长的趋势。

4、CAN 总线协议完全公开且易于获得，协议只包括物理层和数据链路层两个底层协议，应用层协议可以由用户根据具体应用自己定义，也可以采用几种已成为标准的高层协议，给用户自主开发设计的余地很大。

5、支持 CAN 总线的大公司很多，产品种类十分丰富，而且已有许多厂家把 CAN 控制器嵌入到微处理器或 DSP 中，使应用更为方便。比如，Intel 公司、Philips 公司、Microchip 公司、Infineon 公司等都有相关产品，而且价格便宜，易于买到，这样使用户在做开发研究工作时，硬件设备容易获得，且不会受到经济方面的制约。

6、CAN 总线产品时不需要专门的开发工具，用户无须在开发工具上做额外的投资。且开发简单，易于掌握，周期短。可以较容易的构建 CAN 总线控制系统，并以此为平台研究解决现场总线中存在的问题；也可以进一步完善，产品化，形成适合我国国情的现场总线控制系统的解决方案。

三、选择 CANopen 作为 CAN 总线高层协议的原因：

CANopen 和 Devicenet 应用较为广泛，选择 CANopen 为标准编写 CAN 网络的应用层协议，因为 CANopen 与 Devicenet 相比有以下几个优势：

1、Devicenet 网络中最多可接 64 个节点，而 CANopen 最多可接 127 个节点。

2、通过参数的设置，CANopen 协议可以根据用户的特殊需要最大限度的将系统专用化和最优化，无须改变硬件设置，就可以获得最优的通讯特性。

3、Devicenet 对物理层的接口设备要求严格，硬件设备必须符合 Devicenet 物理层协议标准，而 CANopen 对此没有特殊要求，可在 CAN 总线控制系统的硬件基础上直接构建。

4、Devicenet 协议并不是对所有的人都是公开的，而 CANopen 协议则完全公开，很容易在各个网站获得。

四、课题的具体内容：

选择 CAN 总线为研究对象，针对存在的 CAN 控制器的不足，设计开发基于 CANopen 应用层协议、具有较强通用性的模块化 CAN 总线控制器。组成具有二级总线结构形式的 CAN 总线网络控制系统；并以对 CANopen 进行分析，结合工业控制的常规情况，以此为标准编写应用层协议，实现高效的数据通讯，完善网络管理，并使网络中的数据通讯和网络管理标准化模块化。同时，通过对 CANopen 协议的参数设置初步解决 CAN 总线控制系统中的实时性问题。

1. 2. 2 课题的意义

通过本课题的研究，实现了通用性较强的具有二级总线结构的模块化 CAN

总线控制器，并分析和开发了以 CANopen 高层协议为基础的标准的 CAN 总线应用层协议，使 CAN 总线的应用层协议的开发摆脱了各家各不相同的现象，实现了标准化和通用性。切实的掌握 CAN 总线的硬件和软件的开发应用技术，为形成产品化的 CAN 总线控制系统打下了坚实的基础。并可将研究成果推广到其它类型的现场总线中，对于促进现场总线在我国的应用和推广，摆脱国外大公司对现场总线领域的垄断，具有较大的理论和实际意义。

1.3 课题的任务及本系统的优点

一、论文的主要研究内容：

1、CAN 控制器的控制模块的设计与实现。本文采用 Intel 公司的 80C196KB 作为 CPU，外部扩展了 CY7C199、A28F512 作为外部数据/程序存储器；CAN 总线通讯由 SJA1000、82C250 配合两个高速光耦来实现，用来完成数据通讯和控制任务，此控制模块可用于各种类型的控制系统，具有通用性。

2、智能 I/O 模块的设计与实现。设计四种不同类型的 I/O 模块，DI、DO、AI、AO，每个模块都以 Intel 公司的 87C52 作为 CPU，A/D、D/A 转换器选择 Maxim 公司的 MAX529 和 MAX1294 芯片，完成数据的输入、输出、暂存和与控制模块进行 485 通讯的功能，不对数据进行处理。

3、分析 CANopen 高层协议，并以此为标准编写 CAN 通讯的应用层协议。软件上实现 CANopen 网络中从节点的功能，完成 CAN 总线数据通讯、网络管理和相应的控制任务。

4、实现 CAN 控制器控制模块与智能 I/O 模块之间的 485 通讯。即以主控模块为主节点，其它 I/O 模块为从节点的主从式多机通讯任务。并采取一定的措施解决 485 通讯中的可靠性和实时性的问题。

二、本 CAN 总线控制器的主要优点：

1、采用系统总线和 I/O 分离的思想，系统可以自由的增加控制点数，使得 CAN 控制器主板的利用率得以提高。并可根据具体应用任意组合，结构灵活，通用性强。

2、由于采用二级总线的结构形式，可使每个 CAN 节点所接的 I/O 端口数量增多，从而使整个控制系统所需的 CAN 节点减少，CAN 总线上的信息冲突减少，系统的实时性大大增强。

3、应用层的程序按照 CANopen 高层协议编写，使该节点的应用层协议符合国际公认的协议标准，可以与其它 CANopen 产品相兼容。

第二章 模块化 CAN 总线控制器的总体设计

2.1 CAN 总线控制系统的总体结构

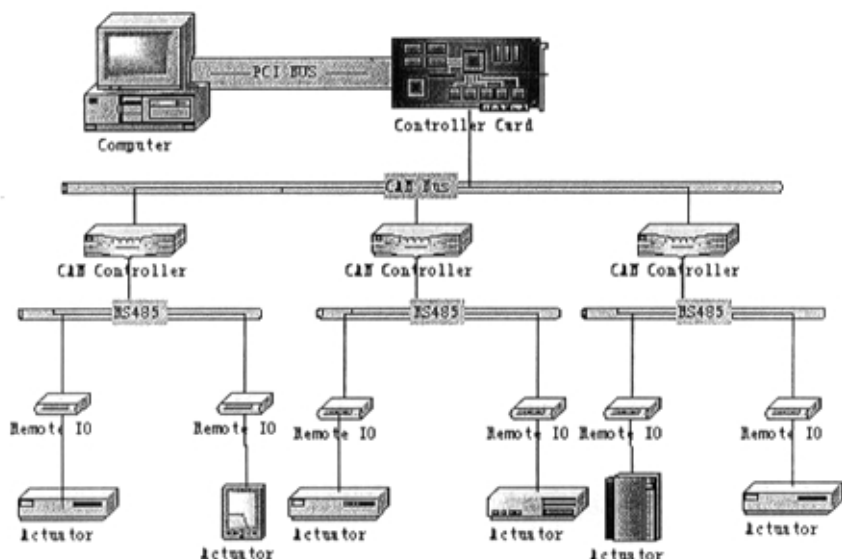


图 2-1 CAN 总线控制系统结构简图

如图 2-1，CAN 总线控制系统由两级总线组成，CAN 总线上根据系统的需要连接若干 CAN 节点（即本文所说的 CAN 总线控制器），通过 PC 适配卡与上位机相连，各 CAN 节点可与上位机通讯，也可以相互通讯。CAN 节点分为节点控制模块和独立的 I/O 模块两部分，两者通过 485 总线进行通讯。本文所设计的部分即 CAN 总线与现场设备之间的 CAN 节点控制模块以及与 485 总线连接的 I/O 模块两部分，完成 CAN 节点控制模块和 I/O 模块的硬件设计，实现两者之间的数据通讯以及 CAN 总线控制器之间的 CAN 通讯。

2.2 组合式 CAN 节点的设计要求

一、完成现场信号的采集和处理：

1、现场信号的采集和处理的任務：

** AI 信号的采集，信号质量的检查、滤波、线性化、工程单位的转换、报警极限的检查和處理。

** AO 信号的量程检查、报警检查输出。

** DI、DO 信号处理。

2、I/O 模块的性能指标:

** 4 路通用模拟输入信道 (AI):

输入信号: 0~10V 或 0(4)~20mA, 12 位并行 A/D, 精度保证在 0.2%。

** 4 路模拟输出信道 (AO):

输出信号: 0~10V 或接 500 Ω 电阻转成 0~20mA 电流, 12 位并行 D/A, 精度保证在 0.2%。

** 4 路数字输入信道 (DI):

支持脉冲输入, 能处理 DC。

逻辑电平: 满足 TTL 电平的要求。

** 4 路数字输出信道 (DO)

二、通讯功能:

1、实时数据的传输:

** 监控现场设备状态值的变化, 并定时传送给上位机。

** 在对某一设备进行闭环控制时, 周期性的将被控对象的状态值传送给控制器, 经过特定的控制算法, 再将所得控制信息及时传送给被控对象的控制器。

** 当现场设备的状态值发生异常变化时, 将这一变化值实时传送给上位机。

** 当上位机或某一 CAN 节点需要了解其它 CAN 节点的状态时, 可以通过远程请求获得其它节点的状态信息。

2、非实时数据的传输:

** 上位机对 CAN 节点的初始化设置所进行的数据传输。

** 上位机对 CAN 节点运行状态的重新设置所进行的数据传输。

** 上位机对 CAN 节点的程序下载。

3、CAN 节点控制模块和 I/O 模块之间的数据传输:

** 输入模块将现场采集到的数据传送给 CAN 节点控制模块。

** CAN 节点控制模块将控制信息实时的传送给输出模块。

** CAN 节点控制模块定期的向输出模块传送检测数据帧, 以检测输出模块是否正常工作。

** 当 I/O 模块发生异常时, 向 CAN 节点控制模块发送报警信息。

三、网络管理功能:

1、时间禁止服务: 即当优先级较高的数据由于信息量大, 占用总线时间过长, 超过了预先设定的禁止时间, 自动停止传输, 将总线让与其它数据。

2、状态管理: 即控制网络中各从节点的初始化、预操作和操作状态的转换。

3、节点保护: 确定被保护的从节点是否处于正常的工作状态。

4、紧急错误处理: 当网络中的节点发生严重错误时, 报警告知其它节点并要

求进行处理。

2.3 CAN 节点的总体设计思想与软硬件方案

2.3.1 总体设计思想

一、目前市面上的 CAN 总线控制器存在的缺点：

1、目前市面上的 CAN 总线控制器基本都属于集中式设计，即将控制器、CAN 收发器、I/O 端口等集成于一块板卡之中，其 I/O 口形式固定，通用性差。

2、大部分 CAN 总线控制器都是根据特定的需要编写自己的应用层协议，不能标准化和统一化，无法推广。

3、由于每个 CAN 总线控制器的 I/O 端口不能太多，因此一个系统中所需的 CAN 节点的数目多，导致系统庞大，实时性差、造价高。

二、本文的实现方案：

为了解决以上问题，本文将做如下设计：

1、CAN 节点控制模块和 I/O 模块相互独立，两部分之间通过全双工的 485 总线进行数据通讯，共同形成一个 CAN 节点。

2、独立的 I/O 模块分为 DO、DI、AO、AI 四种类型，用户可根据具体的控制对象选择 DO、DI、AO、AI 的类型与数量，进行优化组合。

3、应用层的程序以 CANopen 高层协议为标准编写，使本文中的 CAN 节点的应用层协议符合国际公认的协议标准，可以在不同的控制系统中推广使用。

其原理如图 2-2 所示：

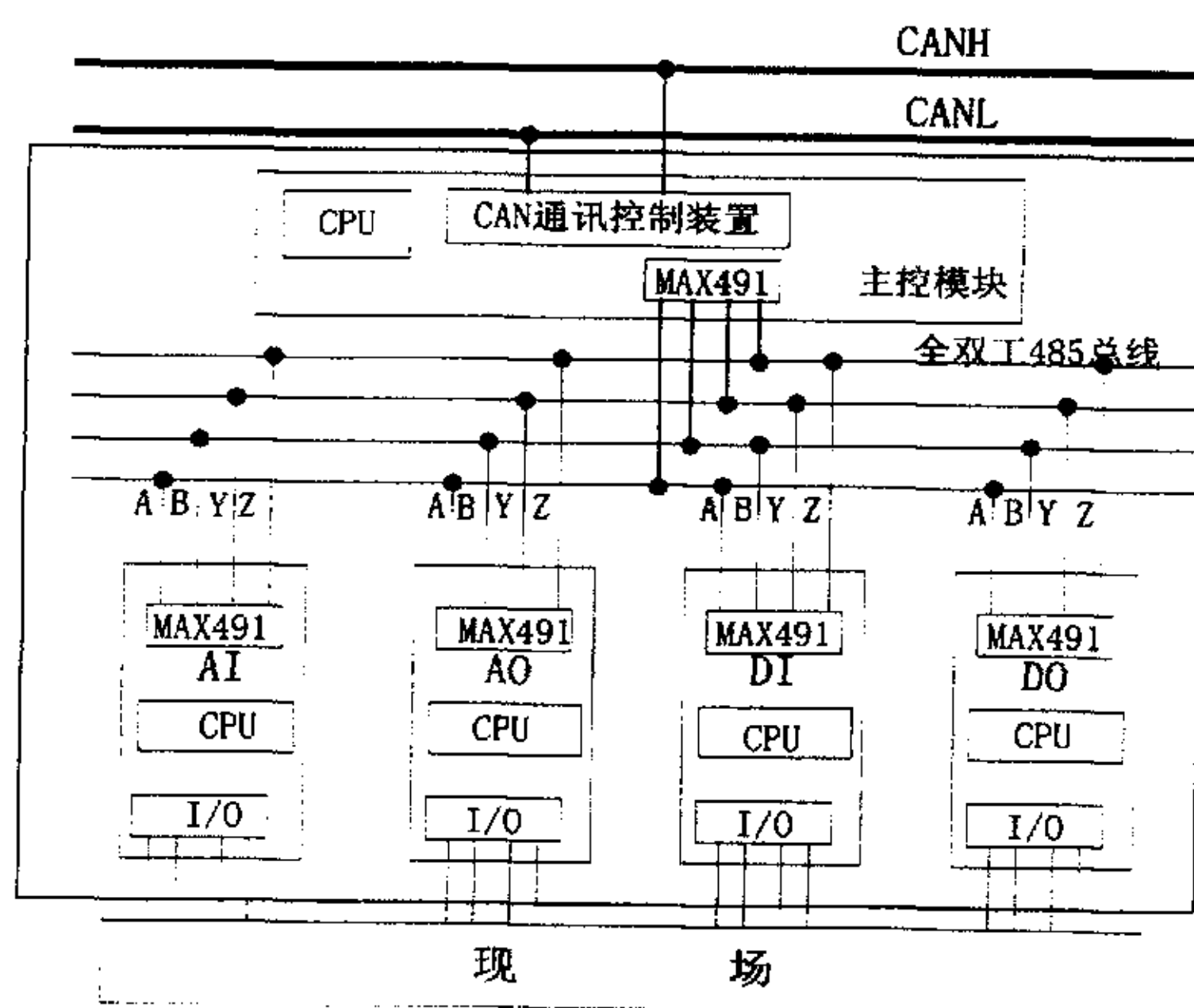


图 2-2 组合式 CAN 节点的总体设计框图

2.3.2 硬件方案设计

一、CAN 控制器主板的设计：

结合我国国情，开发低价位的 CAN 节点并实现上述功能，硬件方案如下：

1、微处理器的选择：

根据系统的性能要求和控制任务，决定选择 Intel 公司的 80C196KB 作为 CAN 节点控制模块的微处理器。80C196KB 是 MCS96 系列中第一代 CHMOS 产品，功耗低，数据总线可配置成 16 位或 8 位，CPU 的操作直接面向 256 字节的寄存器，具有较高的操作速度和数据传输能力。

2、外部存储器的选择：

由于系统既要实现 CANopen 高层协议，又要完成数据采集运算和必要的控制算法，还要考虑到以后系统的扩展和控制算法的改善，需要比较大的程序空间，因此选用 64K 的 8 位 Flash- TMS28F512A 作为外部程序存储器；另外，由于程序中 CANopen 的对象字典列表在上电初始化后要放在数据存储器中，需要占用较大的空间，还需要临时存储数据，因此外部数据存储器选择 32K 的 Cy7c199。

3、CAN 总线通讯部分的硬件设计：

CAN 总线通讯部分选用 Philips 公司的独立式 CAN 控制器 SJA1000 配合 80C196KB 单片机完成 CAN 通讯中的数据发送与接收功能，采用 CAN 总线控制器接口 82C250 作为总线驱动器，完成数据与物理总线的电平信号之间的转换，并在 82C250 与 SJA1000 之间加入两个高速光耦起到隔离干扰的作用。

二、I/O 模块的设计思想：

在本设计中充分考虑到传统集中式板卡将控制器、收发器、I/O 等集于一块板卡所带来的问题，采用 RS-485 总线技术，将现场控制级的设备进一步分散化，分成现场控制级设备（即节点控制模块）和现场的 I/O 级设备（即 I/O 模块）这两个级别，并将 RS-485 总线作为它们之间数据传输的桥梁。

I/O 模块与 CAN 节点控制模块相独立，具有自己的 CPU，可以独立进行数据采集和通讯。本文设计了四种不同类型的 I/O 模块：数字输入模块（DI）、数字输出模块（DO）、模拟输入模块（AI）、模拟输出模块（AO）。每一种模块都采用 Intel 51 单片机 89C52 作为微处理器，负责数据采集和 485 通讯；采用 RS-485 接口 MAX491 进行 485 通讯。输入输出转换的硬件部分属于常规设计，将在第三章中作详细介绍。

2.3.3 软件设计思想

本文中的软件分为两个部分，CAN 总线通讯部分和 485 总线通讯部分。485 通讯采用主从式全双工串行的多机通讯，属于常规设计，将在第五章做详细设计。

CAN 总线通讯部分软件设计思想如下:

CAN 总线控制器作为 CAN 总线控制系统中的一个节点具有控制和通讯的功能。本文意图设计一个通用的 CAN 节点,其通讯功能可以适合各种控制系统,而不同的控制系统具有不同控制算法。因此本文中的软件部分主要实现 CAN 总线控制系统中普遍存在的数据通讯和网络管理的功能。

由于 CAN 协议只有物理层和数据链路层两层,并非是一个完整的协议,要实现高效的通讯必须开发 CAN 应用层协议。为了使软件具有通用性,选用已被广泛接受为 CAN 应用层协议标准的 CANopen 为标准,实现 CAN 总线控制系统中的数据通讯和网络管理的功能。

在 CANopen 网络控制系统中,由一个或多个节点作为主节点,负责对整个网络的设置和管理,它的功能与从节点有所区别,两者的软件结构和实现的功能也有很大的不同。在本设计中,与上位机相连的 PC 适配卡作为主节点,通过上位机对整个网络进行监控和管理。其它节点为从节点,本文中设计的 CAN 总线控制器属于从节点,因此,本文主要实现从节点的软件设计,但为了整个系统的完整性,主节点的软件也做一个构想,还需要进一步完善。

CAN 总线通讯的应用层协议部分采用功能块的概念,遵循以下原则:任何模块间的数据交换,都通过对象字典来完成;任务级软件是共同的,不同的功能尽可能放在功能模块中处理;绝大部分模块是可重复使用的,重复使用的模块程序相同但被程序处理的数据不同;当模块的工作涉及中断服务子程序时,在中断服务子程序中调用功能模块;本设计中依据 CAN 总线控制系统的通讯需求和 CANopen 协议标准,包括以下功能模块:

**** 初始化模块:**包括硬件的初始化和通讯参数的初始化。

**** 对象字典的组建:**定义对象字典中规定的各个参数,并为程序运行中的变量的存储分配相应的空间。

**** 过程数据通讯 (PDO):**系统运行中实时性数据的传输。包括对数据的封装和不同模式的数据发送与接收。包括 PDO 同步通讯模式、时间触发模式和查询模式,满足系统中不同类型数据的传输要求。对实时性要求较高,优先级较高。

**** 服务数据通讯 (SDO):**用于主节点读写访问其它从节点的对象字典和往主节点下载所需的用户程序。对实时性要求较低,优先级较低。

**** 应急数据通讯:**当网络发生致命错误时,报告错误类型并进行错误处理。

**** 节点控制功能块:**主要完成 CANopen 规定的网络管理 (NMT) 功能。主要包括:状态管理、节点保护等功能使网络系统正常运行。

第三章 CAN 总线控制器的硬件设计

CAN 总线控制器分为节点控制模块和独立的 I/O 模块，两者通过 485 总线进行通讯。节点控制模块用可分为 CPU 及外围扩展部分和 CAN 通讯两个部分。本文将分为 CPU 及外围扩展部分、CAN 通讯和 I/O 模块三个部分分别介绍：

节点控制模块的硬件结构图如下：

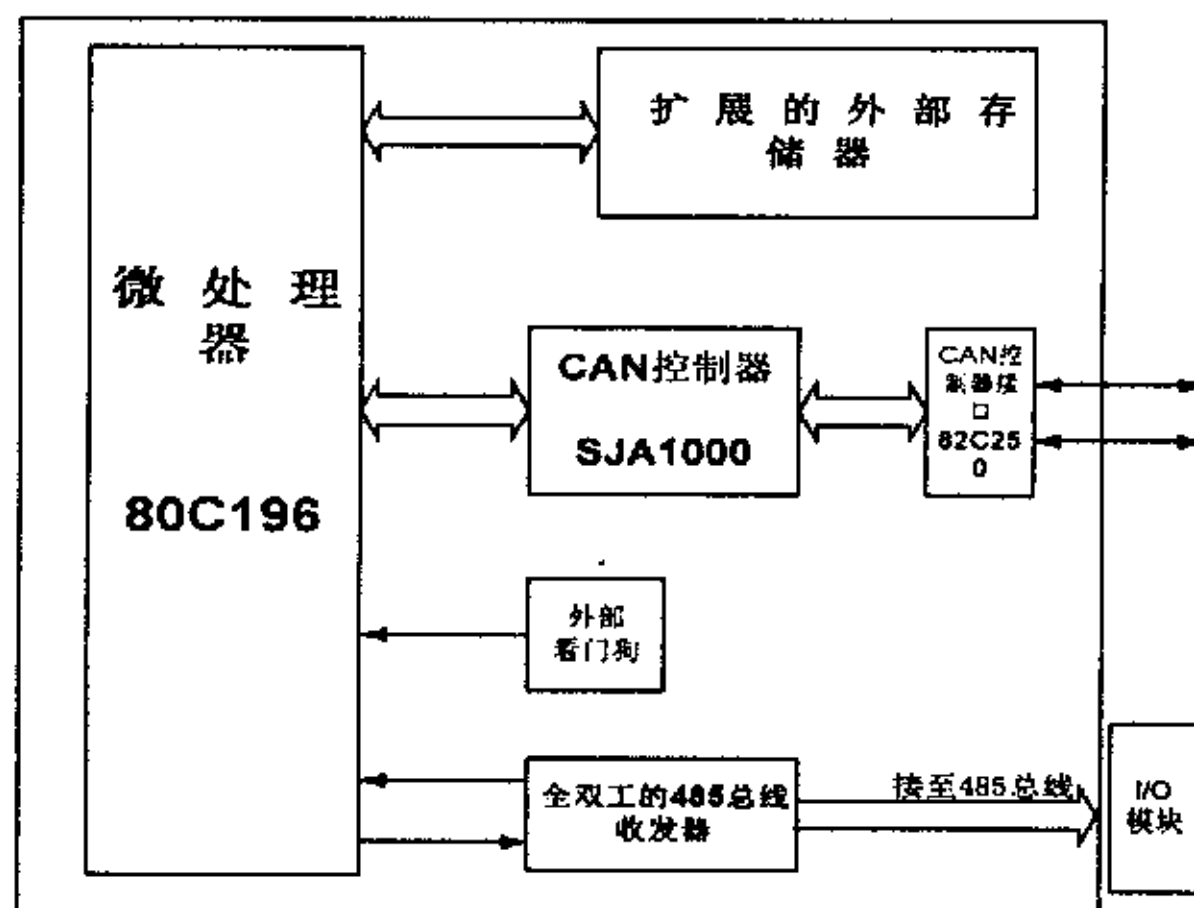


图 3-1 CAN 总线控制器节点控制模块的硬件结构图

3.1 节点控制模块 CPU 及外围扩展芯片的设计

3.1.1 芯片的选择与介绍

一、主处理器 CPU 的选择^{[15][16][25]}：

本设计中选用 80C196KB 作为微处理器，80C196KB 是高速度、高性能的单片机，且增加了待机和掉电两种节电工作方式，进一步减少了芯片的功耗，并支持 HOLD/HLDA 协议，使用更加灵活方便。其硬件结构特点如下：

1、具有比 MCS-51 单片机更强的数据处理功能，指令速度更快，指令平均执行时间为 0.5~1.5us，增加了包括双字比较和块传送在内的 6 种新指令；

2、提供了 28 个中断源，18 个中断向量，15 个优先级。其中非屏蔽中断 NMI、软件陷阱 TRAP 和非法操作码是 3 个特殊的中断源，各占了一个中断向量，其余 25 个中断源分享另外 15 个中断向量；

3、系统总线可工作于三种方式：标准 16 位总线方式，8 位总线方式和 8 位/16 位动态切换方式；

4、专用寄存器映射在 3 个 SFR 的窗口内，所有窗口的 14H 单元是一个窗口选择器 WSR(Windows Select Register)，用于控制各窗口之间的转换。通过 80C196 新添的两条指令 PUSHA 和 POPA 指令可控制 WSR 的入栈和出栈操作，窗口之

间的转换很容易实现。

5、有与 MCS-51 兼容的全双工的串行口；

6、具有以先进先出队列（FIFO）为基础的高速输入方式（HSI），以内容寻址存储器（CAM）为基础的高速输出方式（HSO）；

7、具有两个 16 位的定时器；4 个软件定时器，其受高速输出器（HSO）的控制，到达预定时间时，设置相应的软件定时标志，激活软件定时中断；

8、具有高效而丰富的指令系统。有 16 位乘 16 位和 32 位除 16 位的乘除指令，有符号扩展指令，还有数据规格化指令等等；其很多指令既可以用双操作数也可以用 3 操作数，大大提高了编程的效率。

3.1.2 单片机系统外围芯片的设计

一、CPU 各引脚的分配：

P3、P4 口：作为用于外部存储器的扩展，分别与 74LS373、74ACT245 和静态存储器-CY7C199、Flash -TMS28F512 相接。

P0、P1 口：作为外围芯片的控制信号线。其中：

P1.0、P1.1 与 MAX491 的 RE#和 DE 相接，用于选通 485 串行通讯的输入和输出。

P1.2 与 74LS138 相接，作为片选信号控制线。

P1.5、P1.6 用于输出外部看门狗的复位脉冲。

P0.2、P0.3、P0.4 用于输出控制信号选择三个不同的 CAN 通讯速率。

P2 口：使用它的多功能口的替换功能 TXD、RXD 和 EXINT，其中 EXINT 与 CAN 控制器的 SJA1000 的外部中断输出引脚相接，CAN 通讯中产生的中断由此输入 80C196KB。

XTAL1：由此输入 16M 的外部时钟信号，XTAL2 脚接地。

RESET#：由此输入外部看门狗的复位信号。

二、外部扩展存储器的设计：

Flash 用于存储程序。由于本系统的 CPU 要完成 CAN 通讯（实现 CANopen 高层协议）、完成 485 通讯和对下层 I/O 模块的管理以及针对于底层设备的较复杂的控制算法，还考虑到为以后新增的功能留出必要的程序空间。因此本文采用 64K×8 的 TMS28F512，可存储 25000-50000 行的 C 语言代码，这种规模的程序需要一个人几年的工作，对于本系统足够了。

数据存储器采用 32K×8 的 CY7C199，用于存储程序变量、系统启动变量、CANopen 对象字典、堆栈等。

本文用 74LS373 锁存器锁存低 8 位地址；用 74ACT245 与高 8 位地址相连，起到了总线驱动的作用；读、写、片选等信号线通过 74ALS244 与各芯片相连，

以增强其驱动能力。本文中外部存储器的扩展属于常规设计。

三、外部看门狗电路的设计:

当系统由于干扰导致软件运行紊乱时, 由外部看门狗电路使系统自动恢复。80C196KB 内部也具有看门狗(监视定时器), 它可以用软件起动或禁止, 使用外部看门狗的目的在于当内部看门狗也失效时使系统依然能够自动复位, 保证了系统运行的可靠性。本文使用两个看门狗, 只要其中有一个看门狗正常工作就可使系统复位, 可靠性更高。外部看门狗电路的电路设计如下:

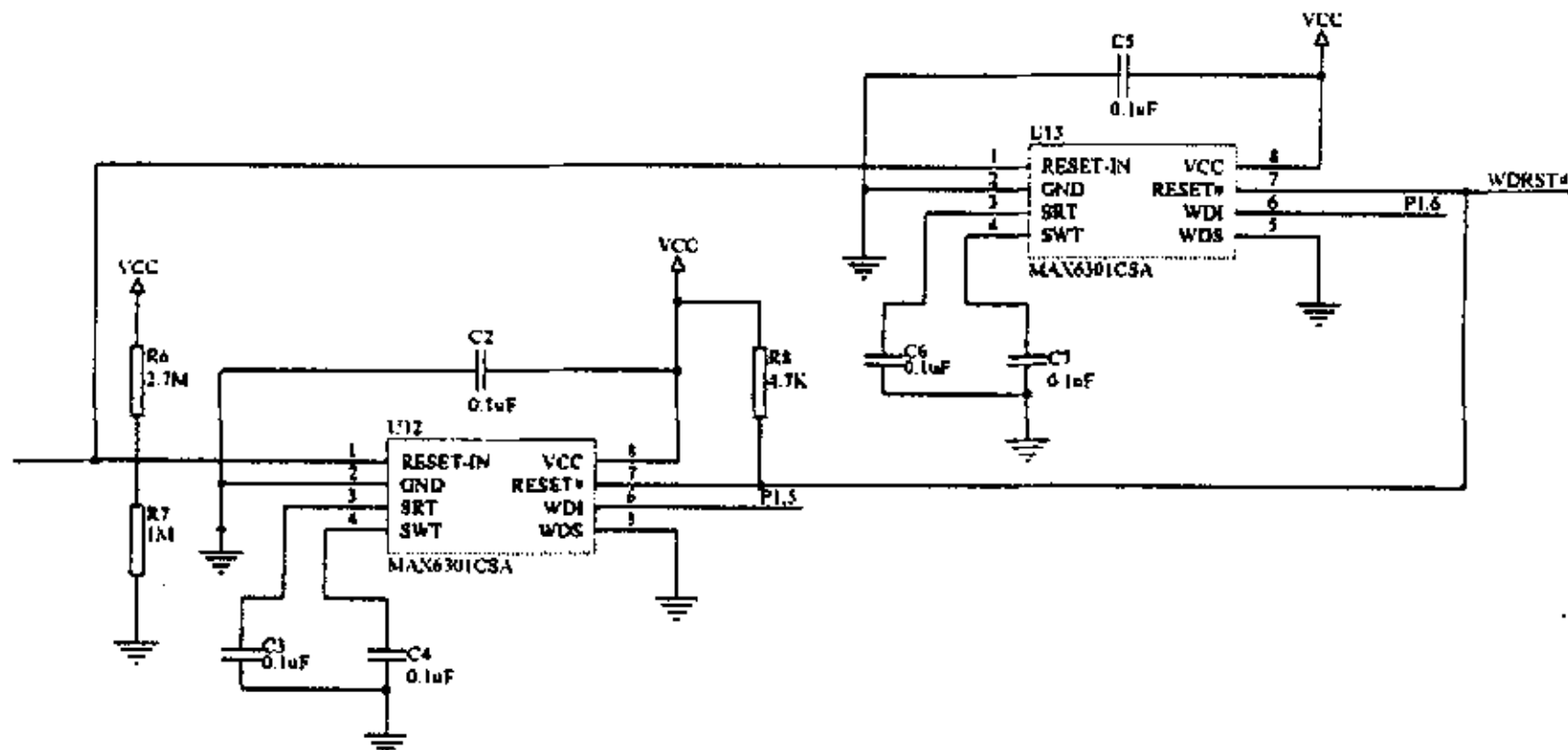


图 3-2 外部看门狗电路的设计

四、单片机存储空间的分配及片选电路的设计:

80C196 单片机具有逻辑上完全统一的存储器空间, 可寻址范围为 64K。本设计中选用了 64K 的外部程序存储器和 32K 的外部数据存储器。超过了 80C196 自身的可寻址能力范围, 因此需要另外设计片选电路。为了方便, 虽然 SJA1000 只占 128 个字节, 也为其分配了 32K 的地址空间。本文采用了页面技术, 把 128KB 的存储空间分为 4 页, 每页为 32KB。利用 INST 引脚把程序和数据空间分开。

MCS-96 系列单片机的低地址部分 (0000H~7FFFH) 包含有保留的存储区, 如中断向量区、芯片配置字节、密钥、复位后的程序执行的起始单元等, 这些保留单元只应出现在一个页面上, 称之为主页(或 0 页), 它的逻辑地址就是 0000H~7FFFH, 其它的物理存储器 (32KB 的高位程序存储器、32KB 的数据存储器、为 SJA1000 分配的存储空间) 都映射到 8000H~FFFFH 逻辑空间, 由 NINST(经过 74ALS244 驱动后的 INST 信号) 和 P1.2 两个引脚来区分。

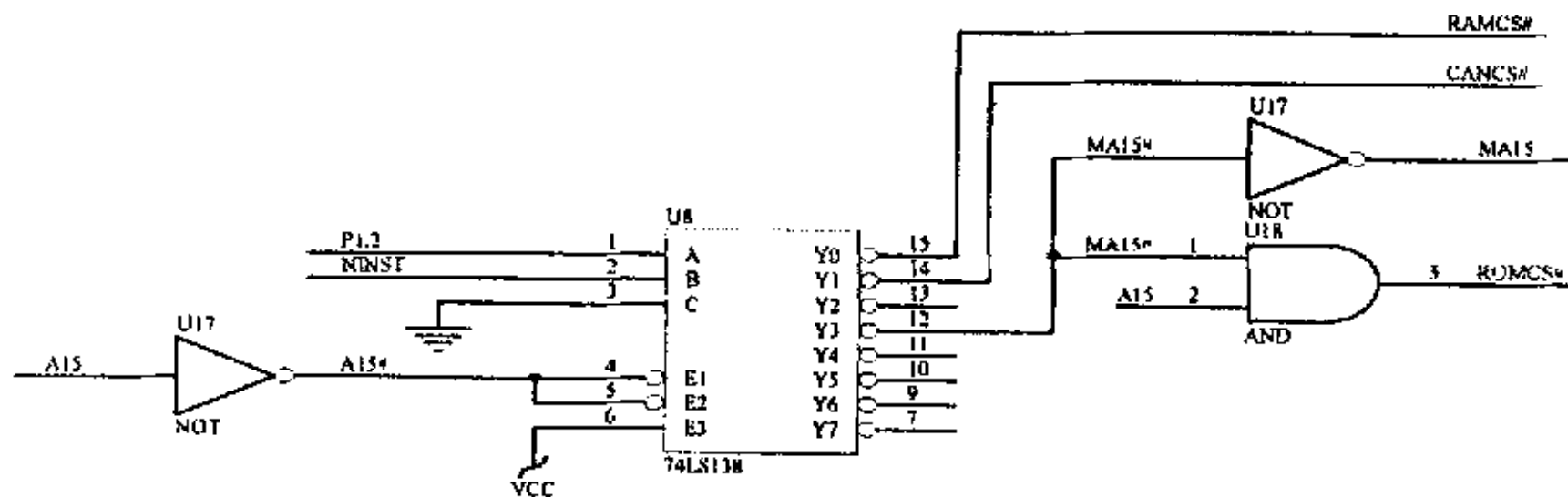


图 3-3 芯片片选电路的设计

如图 3-3, A15 是锁存后的最高位的地址线, NINST 是经过 74LS244 驱动后

的 INST 引脚, MA15 是 Flash 的第 16 根地址选通线。

A15	NINST	PL2	MA15	ROMCS#	RAMCS#	CANCS#
0	×	×	0	0	1	1
1	0	0	0	1	1	0
1	0	1	0	1	0	1
1	1	1	1	0	1	1

表 3-1 芯片片选真值表

MA15 = 0 and ROMCS = 0: Flash 的前 32KB 工作;

MA15 = 1 and ROMCS = 0: Flash 的后 32KB 工作;

RAMCS# = 0: 数据存储器工作; CANCS# = 0: SJA1000 工作。

3.2 CAN 总线通讯电路的设计

3.2.1 芯片的选择与介绍

本文中选用 CAN 通讯控制器 SJA1000、CAN 控制器接口 82C250、高速光耦和主要由 74LS123 组成的显示电路来实现。主要芯片介绍如下:

一、CAN 通讯控制器 SJA1000^[25]:

SJA1000 是由 PHILIPS 公司生产的应用于汽车和一般工业环境的独立 CAN 总线控制器。其基本 CAN 模式(BasicCAN)和 PCA82C200 兼容。同时,新增加的增强 CAN 模式(PeliCAN)还可支持 CAN2.0B 协议。SJA1000 是一种适用于一般工业环境局域网的高度集成独立控制器,具有完成高性能的 CAN 通信协议所要求的全部必要特性。其结构方框图如下:

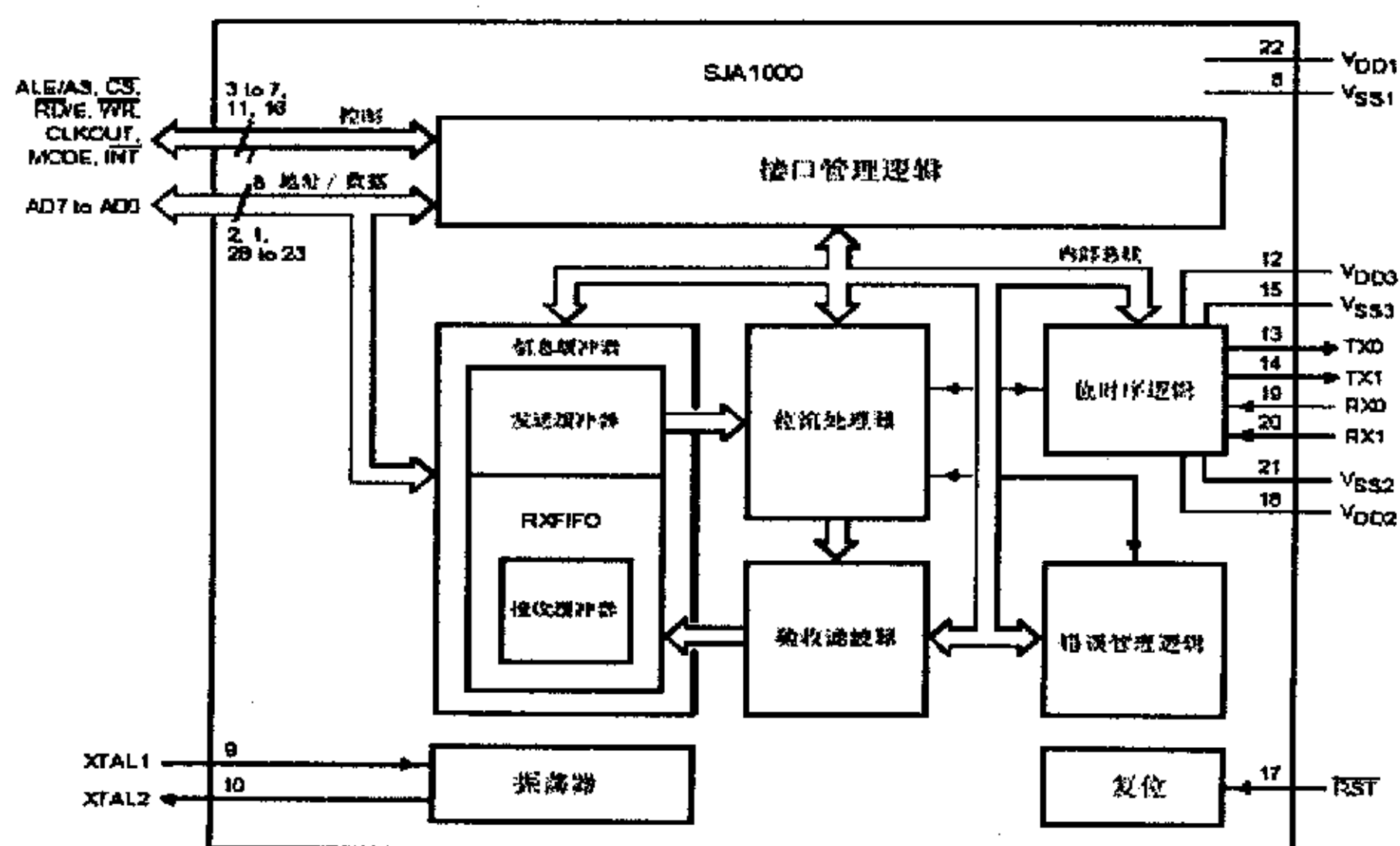


图 3-4: SJA1000 的结构方框图

1、SJA1000 的工作原理:

SJA1000 的控制模块及其功能如下:

(1) 接口管理逻辑 (IML)

接口管理逻辑解释来自CPU的命令, 控制CAN 寄存器的寻址, 向节点控制器提供中断信息和状态信息。

(2) 发送缓冲器 (TXB)

发送缓冲器是CPU和BSP (位流处理器) 之间的接口。能够存储发送到CAN 网络上的完整信息, 缓冲器长13个字节, 由CPU 写入BSP 读出。

(3) 接收缓冲器 (RXB, RXFIFO)

接收缓冲器是验收滤波器和CPU 之间的接口。用来储存从CAN 总线上接收和接收的信息, 接收缓冲器 (RXB 13个字节) 作为接收 (FIFO, RXFIFO长64 字节) 的一个窗口可被CPU 访问。

CPU在此FIFO的支持下可以在处理信息的时候接收其它信息

(4) 验收滤波器 (ACF)

验收滤波器把它其中的数据和接收的识别码的内容相比较, 以决定是否接收信息。在纯粹的接收测试中, 所有的信息都保存在RXFIFO中。

(5) 位流处理器 (BSP)

位流处理器是一个在发送缓冲器。在RXFIFO和CAN总线之间控制数据流的程序装置。它还在CAN总线上执行错误检测、仲裁、填充和错误处理。

(6) 位时序逻辑 (BTL)

位时序逻辑监视串口的CAN总线和处理与总线有关的位时序。它在信息开头“弱势-支配”的总线传输时同步CAN 总线位流 (硬同步), 接收信息时再次同步下一次传送 (软同步)。BTL 还提供了可编程的时间段来补偿传播延迟时间、相位转换 (例如, 由于振荡漂移) 和定义采样点和一位时间内的采样次数。

(7) 错误管理逻辑 (EML)

EML负责传送层模块的错误管制, 它接收BSP的出错报告, 通知BSP和IML进行错误统计。

2、SJA1000 的性能特点:

- ** 管脚及电气特性与独立 CAN 总线控制器 PCA82C200 兼容;
- ** 软件与 PCA82C200 兼容(缺省为基本 CAN 模式);
- ** 同时支持 11 位和 29 位标识符;
- ** 位通讯速率为 1Mbits/s;
- ** 采用 24MHz 时钟频率;
- ** 支持多种微处理器接口;
- ** 可编程 CAN 输出驱动配置;

- ** 工作温度范围为-40 ~ +125℃;
- ** 支持 CAN2.0B 协议;
- ** 扩展接收缓冲器(64 字节 FIFO);
- ** 增强 CAN 模式(PeliCAN);
- ** 增强的验收滤波功能(单滤波方式和双滤波方式),可以对标准帧和扩展帧实现更复杂的滤波功能;

** 增强的错误处理能力,增加了一些新的特殊功能寄存器,包括:仲裁丢失捕捉寄存器(ALC),出错码捕捉寄存器(ECC),错误警告极限寄存器(EWLR),RX 出错计数寄存器(RXERR)和 TX 出错计数寄存器(TXERR)等;

3、SJA1000 的引脚结构:

8	VSS1	AD0	23
21	VSS2	AD1	24
13	VSS3	AD2	25
12	VDD3	AD3	26
18	VDD2	AD4	27
22	VDD1	AD5	28
14	TX1	AD6	1
13	TX0	AD7	2
20	RX1		
19	RX0	RST#	17
11	MODE	INT#	16
		WR#	6
10	XTAL2	RD#	5
9	XTAL1	CS#	4
7	CLKOUT	ALE/AS	3

图 3-5 SJA1000 的引脚结构

二、CAN 控制器接口 82C250^[8]:

82C250 是 CAN 总线控制器和物理总线之间的接口,由 PHILIPS 公司设计生产,最初为汽车高速通信(最高达成 1Mbps, 40m)应用而设计。其通信介质为阻抗为 120 欧双绞线,器件可以提供对 CAN 总线的差动发送/接收能力。

82C250 驱动器的两个信号输出端的电平不是相异的,在 CANH 端,它的两个状态是高电平和悬浮状态;而在 CANL 端,它的两个状态则分别为低电平和悬浮状态。由于 82C250 采用了上述措施,由它构成的分布式测控系统,即使多个节点同时向网络发送数据,也不会发生短路现象。

82C250 主要特性如下:

- ** 与 ISO / DIS 11898 标准全兼容;
- ** 最少可连接 110 个节点;
- ** 最高通讯速率可达 1M bps;
- ** 具有抗汽车环境下的瞬间干扰,保护总线的能力;
- ** 防护电池与地之间发生短路;
- ** 网络中的某一个节点掉电不会影响整个网络的工作;
- ** 斜率控制,抗射频干扰;
- ** 低功耗待机;
- ** 差分发送与差分接收,抗电磁干扰(EMI)能力强。

82C250 的引脚结构如下:

8	RS	VCC	3
5	VREF	CANH	7
4	RXD	CANL	6
1	TXD	GND	2

图 3-6: 82C250 的引脚结构

3.2.2 CAN 总线通讯的电路设计^{[11][12][13]}

一、CAN 总线通讯的电路设计如图 3-7 所示:

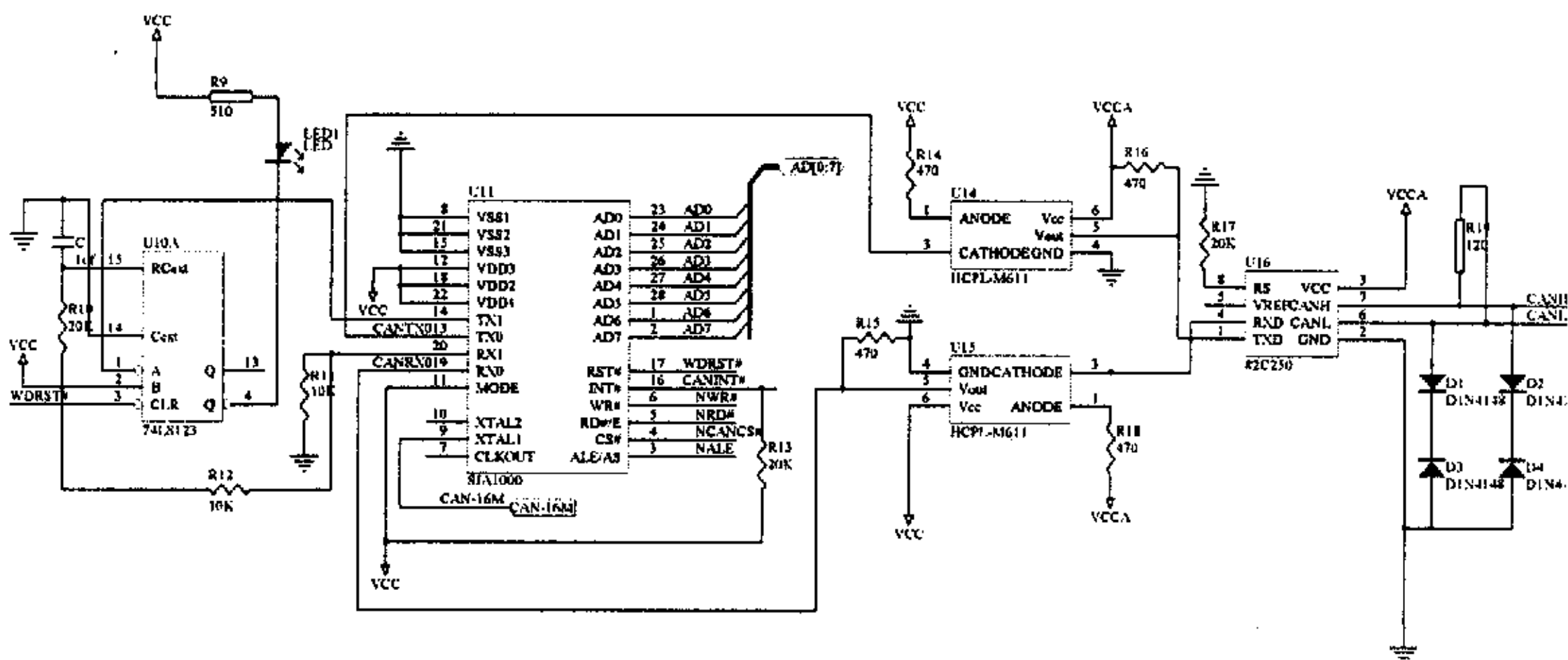


图 3-7 CAN 总线通讯的电路设计

二、CAN 总线的通讯原理^{[7][8][25]}:

在整个 CAN 节点中, SJA1000 相当于与外部数据/程序存储器一样的单片机外围设备。对于单片机来讲, CAN 数据帧的发送与接收就是对 SJA1000 的发送/接收缓冲器读/写访问; 而单片机对 SJA1000 的监控通过对 SJA1000 的控制寄存器的某一位的读/写来完成; 反之, 当 SJA1000 接收到数据或检测到错误时, 以中断的形式通过单片机的外部中断引脚告之单片机。

CAN 协议中物理层和数据链路层的功能, 如数据封装/拆装、帧编码(填充/解除填充)、总线竞争、无损仲裁、接收滤波、超载通知、错误检测、出错标定、应答、串行化/解除串行化等信息发送和接收的主要任务都由 SJA1000 硬件来完成, 无需软件干预。

SJA1000 支持 CAN 2.0A 和 CAN2.0B 协议, 本文应用 CAN2.0B 协议。在使用 SJA1000 可以选用两种模式: BasicCAN 模式和 PeliCAN 模式, 本设计中选用 PeliCAN 模式。

1、数据的发送:

发送数据帧时, 单片机 80C196 负责按照 CAN 协议将要发送的数据封装成标准的 CAN 信息帧, 即定义 CAN 信息帧的标识符、选择要发送的 8 个字节的数据并存储在单片机内部的发送数据存储区内; 需要发送时, 将 CAN 信息帧按

规定格式写入 SJA1000 的发送缓冲器，然后将 SJA1000 控制寄存器中的发送命令位 CMR.0 置 1。其剩余的工作如竞争总线，发送及竞争失败重新发送的过程由 SJA1000 完成

2、数据的验收滤波：

在接收数据的过程中，数据的验收滤波很关键，所谓验收滤波是指CAN总线上的某一节点对接收到的数据进行识别，接收发给该节点的数据，丢弃与该节点无关的无效数据，减轻了CAN控制器的信息处理任务。这一过程由SJA1000中的验收屏蔽寄存器（AMR）和验收代码寄存器（ACR）联合起来完成。

滤波器由验收代码和屏蔽寄存器根据规定算法来控制。接收到的数据自动和验收代码寄存器中的值进行逐位比较。接收屏蔽寄存器定义接收到数据的相应位是否需要经过验收滤波：0 =相关（表示需要进行验收滤波），1 =不相关（表示不需要进行验收滤波），只有收到信息中与接收屏蔽寄存器的相关位的值与验收代码寄存器位相同，这条信息才会被接收。

BasicCAN 模式里滤波器是由两个8 位寄存器控制-验收代码寄存器（ACR） 和验收屏蔽寄存器（AMR）， CAN 信息标识符的8个最高位和这些寄存器里值相比较。

对于Pelican 模式验收滤波被扩展4个8 位验收代码寄存器（ACR0， ACR1， ACR2 和ACR3）和4个8位的验收屏蔽寄存器（AMR0， AMR1， AMR2和AMR3）。这些寄存器可组成单滤波器和双滤波器两种模式。

单滤波器是指将ACR_n和AMR_n配置成一个4字节的长滤波器，滤波器字节和信息字节之间位的对应关系取决于当前接收帧格式。双滤波器是指将ACR_n和AMR_n配置成两个滤波器，第一个滤波器用到ACR0/AMR0、ACR1/AMR1和ACR3/AMR3的低四位，第二个滤波器用到ACR0/AMR0和ACR3/AMR3的高四位，两个滤波器是“或”的关系。一条接收的信息要和两个滤波器比较来决定是否放入接收缓冲器中，至少有一个滤波器发出接受信号，接收的信息才有效。在本文中用到的是单滤波器，其示意图如图3-8所示：

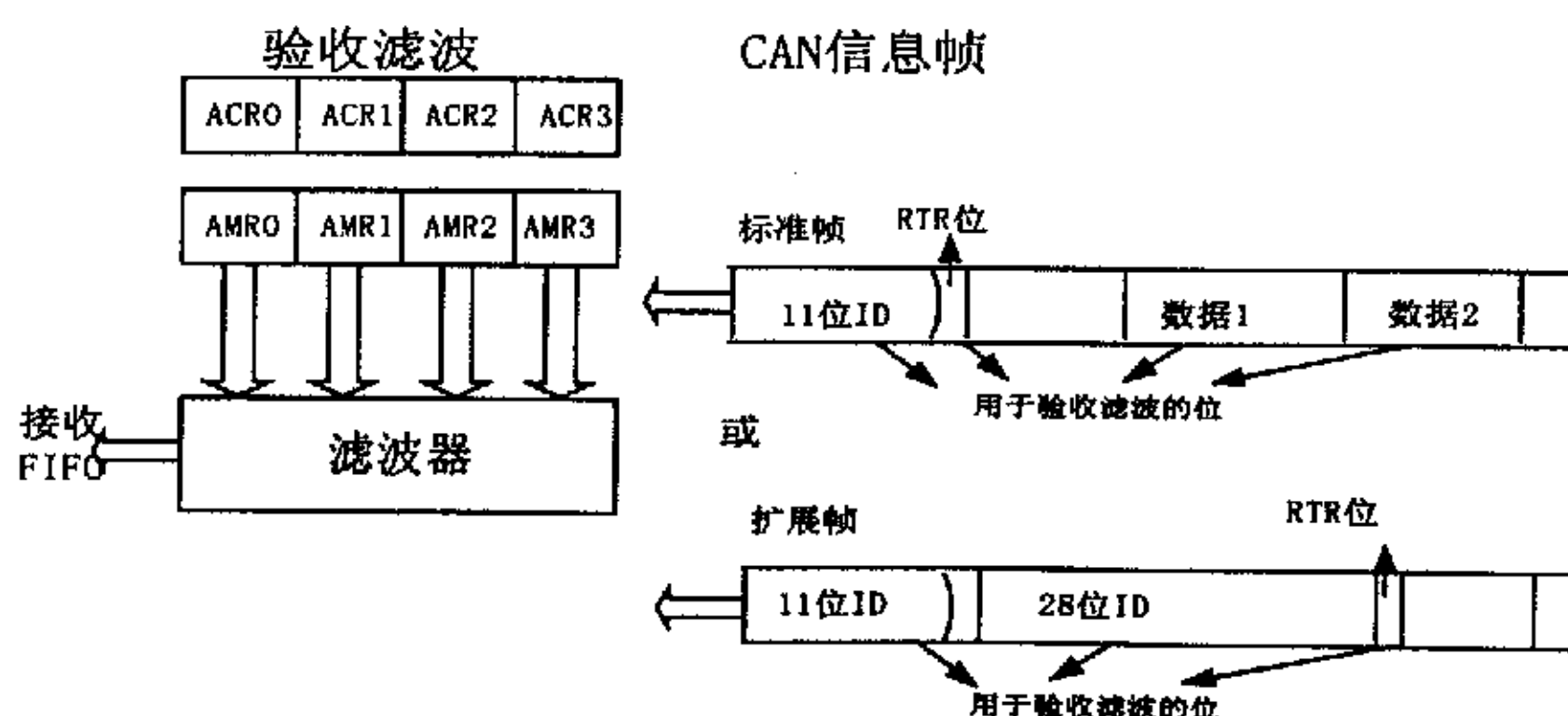


图 3-8 在 Pelican 模式下的单滤波的验收滤波

在单滤波器中，标准帧的验收滤波可以包括RTR 位和两个数据字节。不需要经过验收滤波信息位必须在验收屏蔽寄存器的相应位置“1”；扩展帧的验收滤波只包括29位的标识符，不包括数据字节在内。如果一条信息不包括数据字节（例如在一个远程帧或如果数据长度代码为零）但是验收滤波器中有数据字节，如果ID到RTR位有效，则信息会被接收。

在具体的应用中，只需选择合适的滤波模式，把在接收数据时需要验收的位在AMRn中相应的位置“0”，不需经过验收的位置“1”；而在与AMRn值为“0”对应的ACRn位上设置特定的值。当接收到数据时，前面四个字节的数值依次与AMRn值为“0”对应的ACRn位相比较，相同则接收，不同则丢弃。设定AMRn值的任务由单片机的软件完成，比较验收的任务由SJA1000硬件自动完成。

数据的接收过程主要就是验收滤波过程，验收滤波通过，数据就会被暂存在接收缓冲器（FIFO）中，并设置一个中断，通过单片机的外部中断引脚通知单片机进行处理。

在SJA1000的PeliCAN模式中，RXFIFO共有64个信息字节的空间，若单片机没有及时将接收到的数据转移，下一帧信息可以接着存储在RXFIFO中。一次可以存储多少条信息取决于数据的长度。如果RXFIFO中没有足够的空间来存储新的信息，CAN 控制器会产生数据溢出条件，此时这条信息有效且接受检测为肯定，已写入RXFIFO 的信息将被删除。这种情况可以通过状态寄存器和数据超限中断（中断允许）反应到CPU，由单片机进行处理。

3.3 I/O 模块的设计

本文设计了 A/I、A/O、D/I、D/O 四种不同类型的 I/O 模块，每个 I/O 模块包括四个输入或输出通道，通过 MAX491 与 485 总线相连，与节点控制模块进行 485 通讯。

3.3.1 485 通讯的原理及电路设计

一、节点控制模块与 I/O 模块的通讯原理：

如图 3-9 所示，本文中采用主从式的 485 通讯，为了提高系统的通讯速度和可靠性，采用全双工的通讯方式，即 422 通讯。节点控制模块是多机通讯网络中的主节点，其它各 I/O 模块为从节点，节点控制模块对网络有控制权，并且对其他节点的传输允许进行授权，同时对 485 总线上的所有从节点进行管理。通讯功能有 MAX491 系列收发器完成。组网时，一对导线将这个主节点的驱动器和所有的从节点的接收器相连。在另一个方向上，另一对导线将从节点的驱动器和这个主节点的接收器相连。

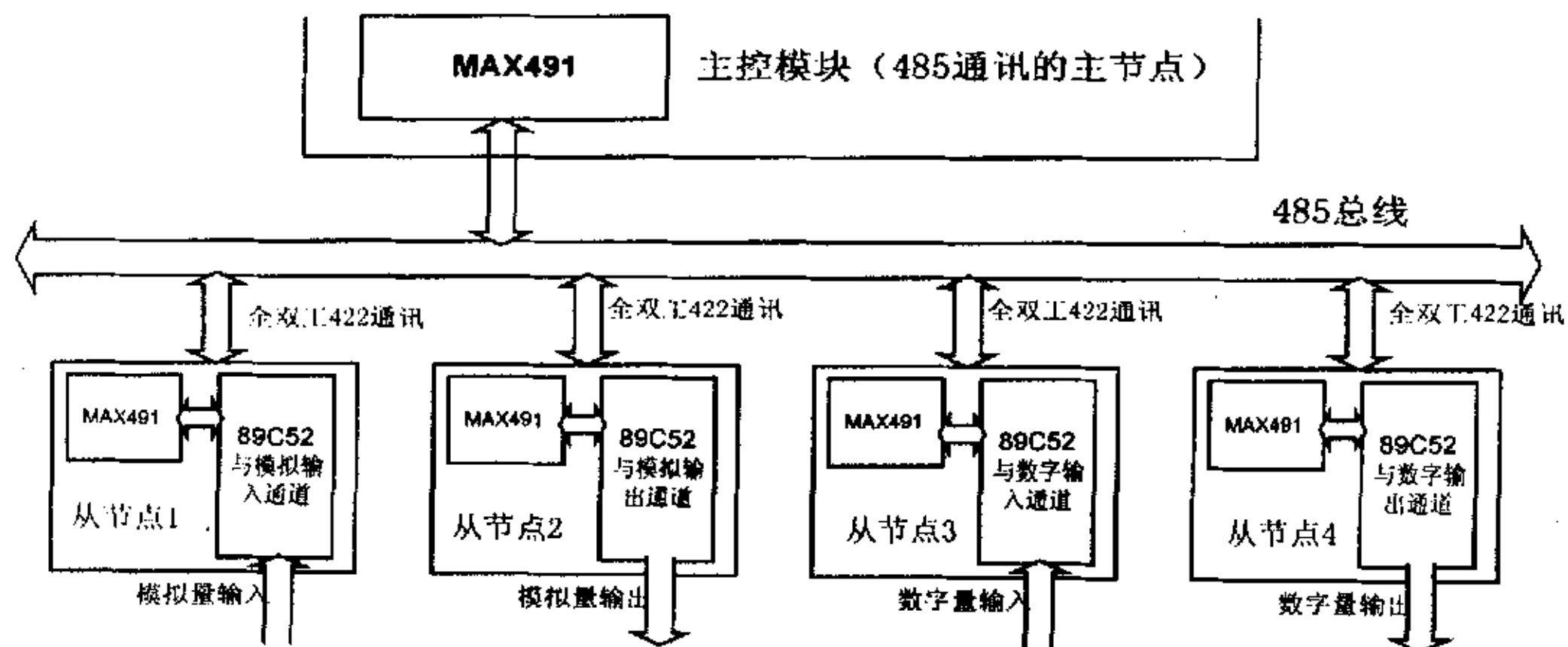


图 3-9 I/O 模块的 485 通讯原理图

二、MAX491 收发器芯片介绍：

1、MAX491 收发器的性能特点：

- ** 单+5V 电源供电；
- ** 低功耗：工作电流 120-500uA；电流 300uA；
- ** 驱动器过载保护；
- ** 可组成半双工 / 全双工通信电路；
- ** 输入电压范围：-7V—— +12V。

2、利用 MAX491 收发器进行全双工通讯组网方式：

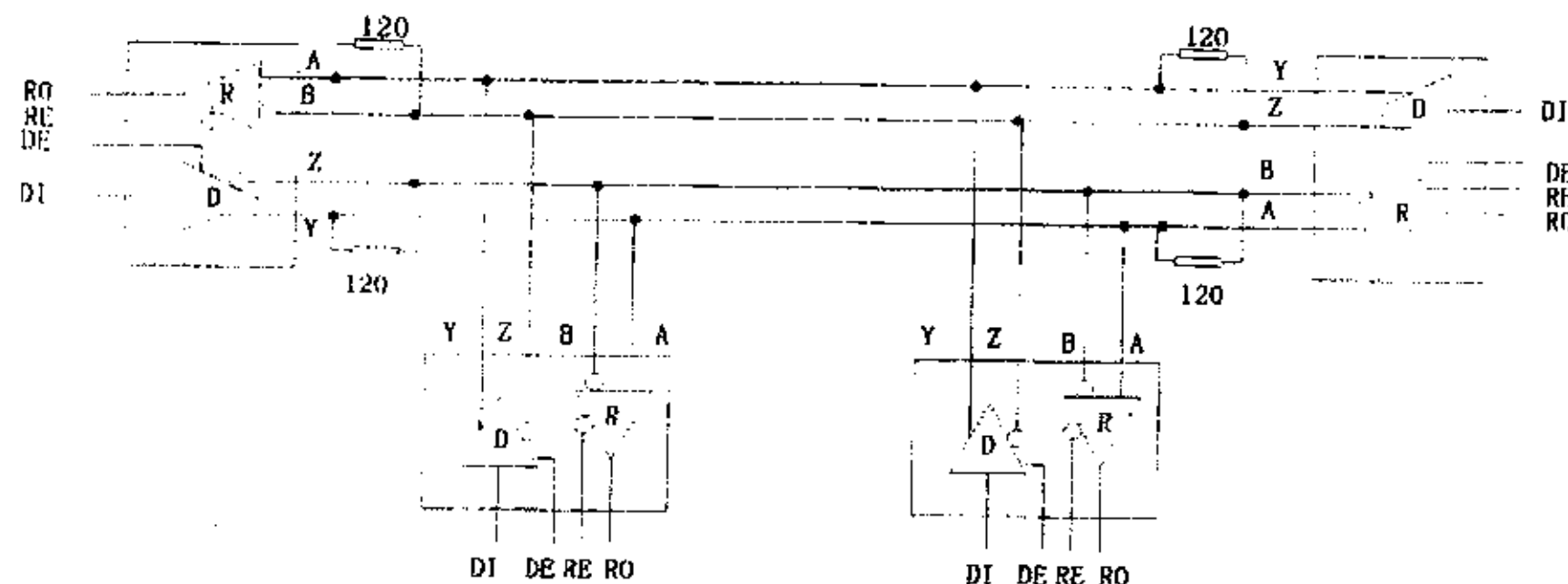


图 3-10：全双工 RS-485 通信网

MAX491 为 14 脚封装，提供了差动驱动器和接收器，驱动器的输出 Y、Z 和接收器的输入 A、B 是分开的，便于组成全双工通信，其连线方式如图 3-10 所示。

3、485 总线的通讯原理^[55]：

(1) 485 总线的电流要求：

一个 RS-485 使用的总电流随着连接中元件阻抗的变化而变化，这些元件包括驱动器、电缆、接收器以及终端负载元件。

终端负载元件在使用的时候对电流影响最大。因此 RS-485 连接有一个 120Ω

的电阻，这个电阻横跨连接两端的差动连线，电阻的并联值为 60Ω 。这些终端负载电阻从这个带有逻辑高电平输出的驱动器进入带有逻辑低电平输出的驱动器，生成了一条低阻抗通路。在没有终端负载元件的时候，接收器的输入阻抗对整个串联电阻影响最大。总的输入电阻随着被启用接收器的数量和它们的输入阻抗的变化而变化。

一个 RS-485 驱动器可以驱动 32 个单位负载。每个 RS-485 驱动器根据需要的电流定义了一个单位负载。

(2) 485 总线的电压要求：

RS-485 接口一般使用一个单一的 5V 电源，对于一个有效的输出，输出 A 和 B 之间的压差必须至少为 1.5V。

图 3-11 显示了一个 RS-485 驱动器的 A 和 B 输出，每个输出都以信号地为参考。每个输出的振幅大约为 3V，变化范围为从 +1V 到 +4V 或者从 -1V 到 -4V。驱动器的电源为 +5V。这个差压信号波谷到波峰的幅度大约为 6V，或者单独一个信号波谷到波峰幅度的两倍。

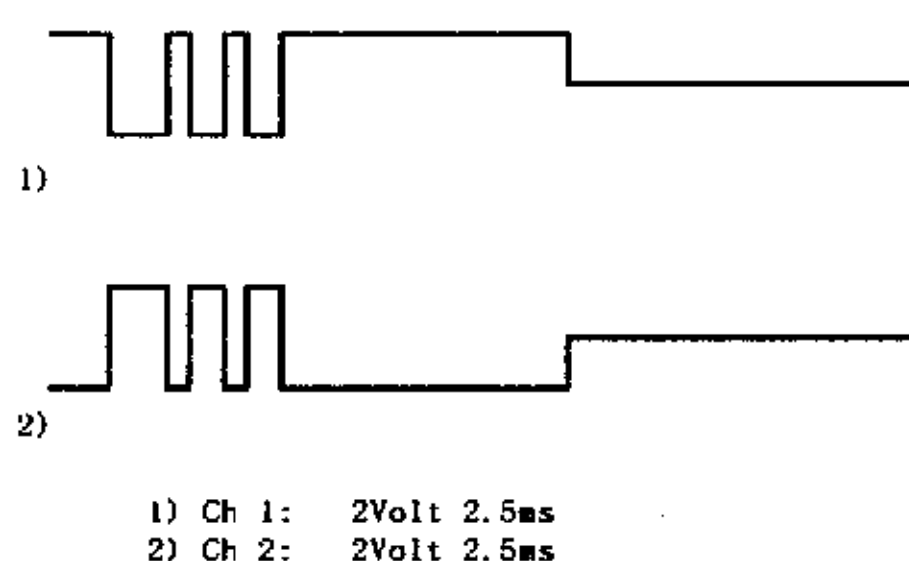


图3-11： 一个RS-485驱动器的输出，以接地电压为参考。线B（底部）的电压为线A（顶部）的电压取反

在 RS-485 的接收器端，A 和 B 输入的差异只需要 0.2V。如果 A 至少比 B 高 0.2V，接收器就认为它是逻辑 1，如果 B 至少比 A 高 0.2V，那么接收器就认为它是逻辑 0。如果 A 和 B 之间的电压差低于 0.2V，这个逻辑电平就属于没有定义。

RS-485 接口芯片是反向器，因为在这个芯片的 RS-485 端 $B > A$ 对应于在 TTL 端的一个逻辑低电平，而在 RS-485 端 $A > B$ 对应于在 TTL 端的一个逻辑高电平。

(3) 485 总线的通讯原理：

所有的从节点必须读取来自主节点的消息来找出它们的信息要发送给哪个节点。被指定接收信息的从节点在相对的导线对上做出响应。这种方案的优点是它为从节点节省了时间，因为它们没有必要读取其他从节点的响应。如果所有的节点共享一条数据通道，从节点就必须读取所有的网络通话，以便监视一条来自主节点的消息。

主从 MAX491 之间用平衡（差动）方式传输。RS-485 的电气标准规定：A、

B 之间的压差不得小于 1.5V。接收器对这些电压的压差做出反应。也就是说，当发送端 $Z=0$ 且 $Y=1$ 时，接收端 $V_A - V_B \leq -0.2V$ ，此时接收为 0，反之为 1。这样一位一位地将 8 位数据由从动模块上传到节点控制模块。至此，这一过程结束。如下表所示：

发送功能表					接收功能表			
输 入			输 出		输 入			输 出
RE	DE	DI	Z	Y	RE	DE	A - B	RO
X	1	1	0	1	X	1	$\geq +0.2V$	1
X	1	0	0	1	X	1	$\leq -0.2V$	0
0	0	X	高阻	高阻	0	0	输入路	1
1	0	X	高阻	高阻	1	0	X	高阻

表3-2 485总线的发送/接收功能表

3.3.2 模拟输入模块（AI）电路设计

一、主要芯片介绍：

1、P89C52^[18]：

(1) 简述

P89C52 属于 8 位的 MCS-51 系列微控制器，它包含一个非易失性的快闪程序存储器，具有以下特性：

- ** 具有 8K Bytes 可重复编程的闪存，速度可达 33MHz；
- ** RAM 可达 256 k Byte；
- ** 4 个优先中断级，6 个中断源；
- ** 4 组 8 位共 32 条可编程 I/O 线；
- ** 全双工方式的串行接口（UART）；
- ** 3 个 16 位计数/定时器；
- ** 具有时钟可被停止和唤醒、低功率停顿模式和功率下降模式三种功率控制模式。

2、MAX1294：

MAX1294 为具有 6 路单极性转换通道的 12 位 A/D 转换器，具有 12 路高速并行接口，工作于+5V 电压，共有 28 个管脚。其特点如下：

- ** 单+5V 操作；
- ** 内部+2.5V 基准；
- ** 可编程模拟输入复用器；

** 可编程单极/双极模拟输入;

** 12 路并行接口。

二、基本原理及功能:

模拟量输入的原理图 3-12 所示:

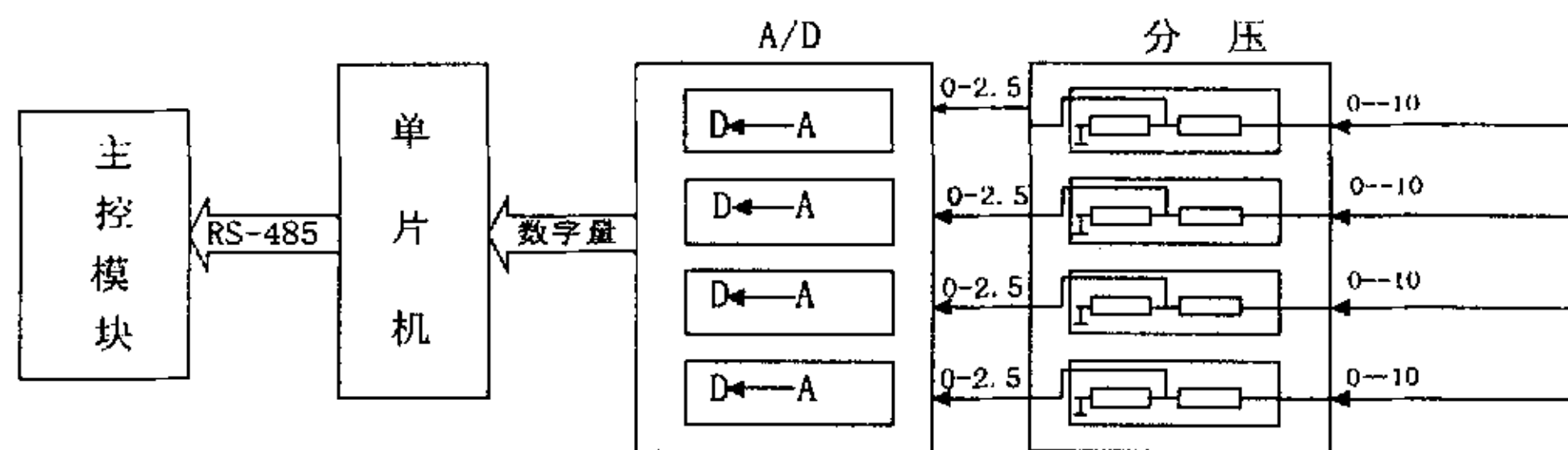


图 3-12 模拟量输入原理图

现场电压的技术指标规定为 0~10V，而 A/D 转换器 MAX1294 所需的输入信号为 0~2.5V，所以在四路输入中，均加入一个分压电路，将输入 MAX1294 的电压降为 0~2.5V。

现场的四路输入信号，经过压降后进入 MAX1294 中，分别存储在其内部的四路 12 位 DAC 寄存器中，经过缓冲（即将 A→D 的过程），将结果暂存于寄存器中，然后向 CPU 发送中断请求，当收到发送允许指令后，将结果依次经过 12 位输出端口，传输到从动模块 AI 的 CPU 中。此时，在 RS-485 通讯中，从动模块处于发送状态。由 CPU 经 MAX491 以串行方式将数据传输给节点控制模块。

3.3.3 模拟输出模块（AO）电路设计

一、MAX527:

（1）一般说明:

MAX527 包含四个 12 位、电压输出型数模转换器（DAC）。片内包含有精密的输出缓冲放大器，用来提供电压输出。MAX527 工作于 $\pm 5V$ 的电源电压。

器件具有 12 位输入寄存器和 12 位 DAC 寄存器的双缓冲接口逻辑。DAC 寄存器中的数据决定了 DAC 输出电压。MAX527 是 8 位数据总线。通过两次写操作，第一次写 8 位 LSB，第二次写 4 位 MSB，将 12 位数据装载到输入寄存器。异步装载 DAC 输入端（LDAC）将数据从输入寄存器传送到 DAC 寄存器。所有的逻辑输入都与 TTL 和 CMOS 兼容。

MAX527 具有 24 脚、300mil 塑料 DIP、陶瓷 SB、以及宽 SO 封装形式。

（2）性能特点:

- ** 全 12 位性能，并且不需调整
- ** 缓冲电压输出
- ** 输出快速建立（5 μ S）

- ** 双缓冲数字输入
- ** 微处理器及 TTL/CMOS 兼容
- ** $\pm 5V$ 电源工作

二、基本原理及功能:

模拟量输出的原理图如图 3-13 所示:

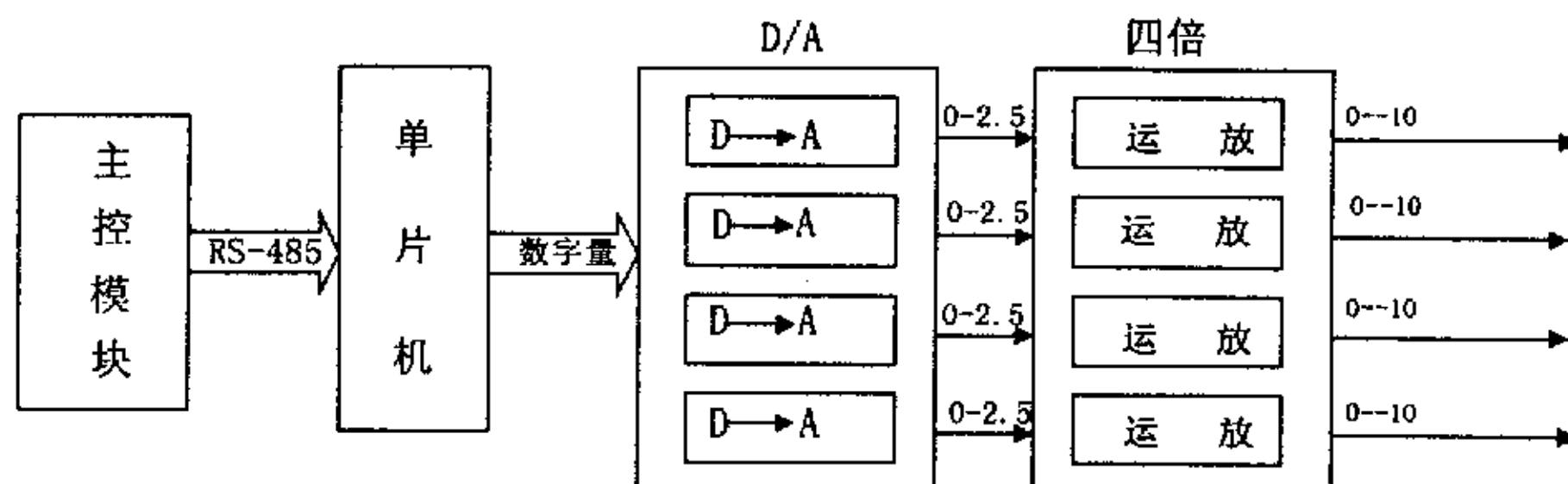


图 3-13 模拟量输出原理图

此时, RS-485 通信中过程即为 AI 中 RS-485 通讯过程的反过程, 节点控制模块处于发送状态, 从动模块 AO 处于接收状态。

当从动 CPU 接收到节点控制模块发送的数据量后, 将其低 8 位与高 4 位分时存入 MAX527 的 12 位 DAC 寄存器中。MAX527 共有 4 路 12 位 DAC 寄存器。经 MAX527 的缓冲变换 (即 D→A) 后, 输出的信号为 0~2.5V, 这并不符合现场技术指标, 因此用 LM224J 即一个 4 路 4 倍运放组芯片, 将其放大至 0~10V, 输出到现场设备。

3.3.4 数字量输出模块 (DO) 的设计

一、原理示意图:

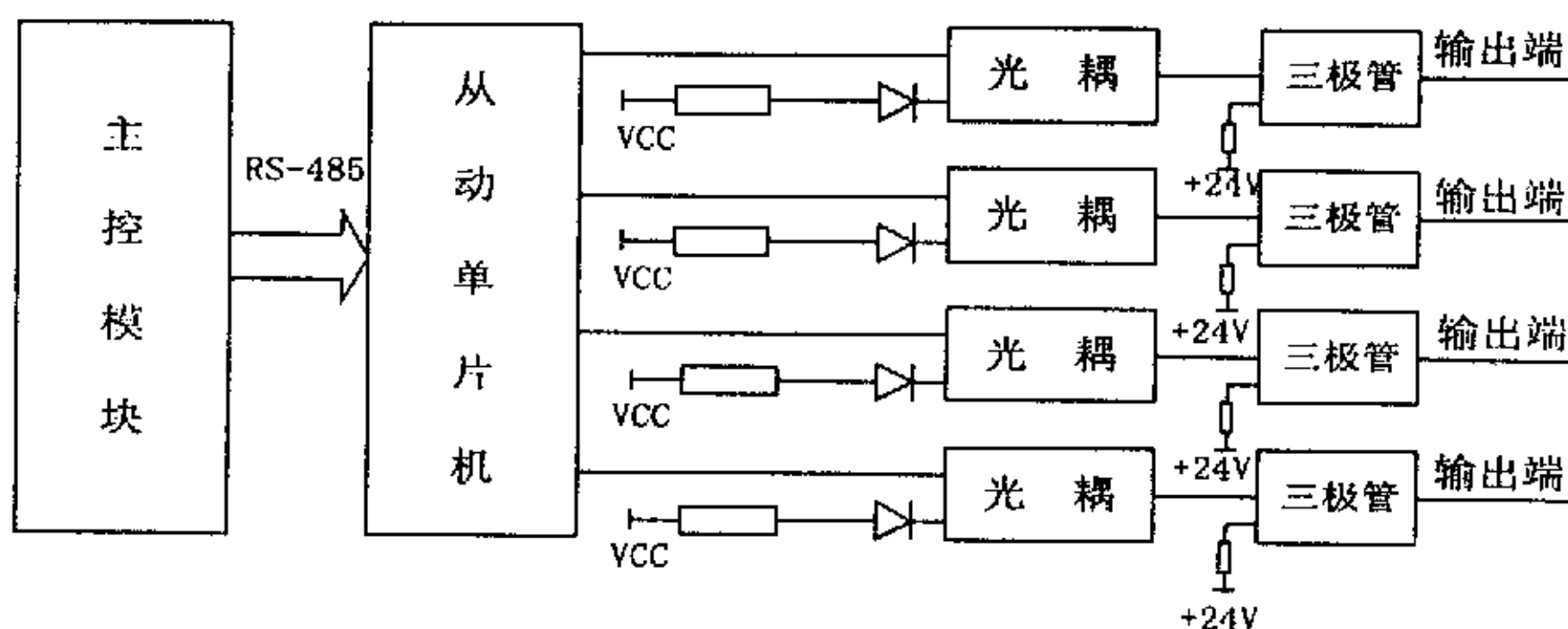


图 3-14 原理示意图

二、数字量输出模块原理:

当节点控制模块将数据经 MAX491 收发器传输至从动模块单片机后, P89C52 将“0”(0-0.3V)和“1”(5V 左右)经光耦以电子开关形式输出为开关量。LED1~LED4 用以显示光耦内部的通断情况, 外部的发光二极管与光耦内部的发光二极管同步开通或关断。以其中一路为例:

当 P89C52 输出为“0”时，由于 VCC 的作用，光耦一端发光二极管发光，另一端晶体管导通。由外部+24V 电源供电经终端 3K 阻抗限流，使外部电路电流工作于 1mA 左右。此时，外部电流无法驱动输出晶体管，相当于输出为“0”。

当 P89C52 输出为“1”时，光耦无法导通。在外部电源作用下，输出晶体管导通，即输出为“1”。

3.3.5 数字量输入模块设计

一、原理示意图：

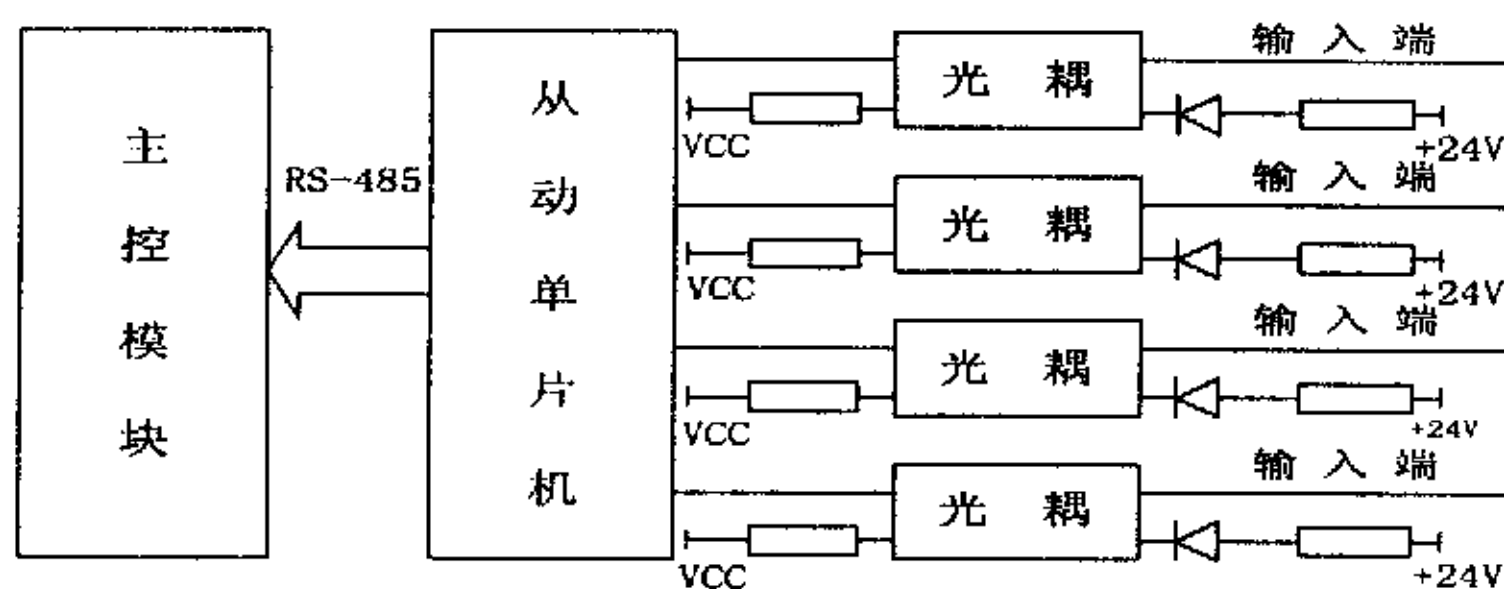


图 3-15 数字量输入模块原理示意图

二、数字量输入模块原理：

当外部输入信号为“1”时，在外部+24V 电源作用下，外部电路电流不足 10mA，所以光耦内部发光二极管无法导通。由于光耦晶体管无电流，内部电路无法导通，CPU 输入电压在 VCC 作用下为 5V，即此时输入为“1”。

当外部输入“0”时，外部电流可以驱动光耦工作，光耦内部发光二极管导通。在光耦的另一端，内部晶体管导通，使得内部电路导通。内部电源提供的 5V 电压均降在了内部终端阻抗上。此时，CPU 输入端的电压很小，可以认为输入为“0”。

第四章 通信协议

4.1 概述

CAN 协议包括 CAN 2.0A 和 CAN2.0B 两部分内容，只定义了数据链路层和物理层，不是一个完整的网络协议，缺少应用层和网络管理部分，用户可以根据实际应用开发自己的应用层协议，领先的用户们定义他们自己的协议并进行推广，使之也成为一种标准。今天存在的一些应用层协议有：DeviceNet, SDS, CAL/CANopen, CAN Kingdom, 其得到 CiA (CAN in Automation) 的支持。

本文采用 CAN2.0B 和 CANopen 应用层协议，组成一个完整的网络协议，CAN2.0B 主要由硬件 SJA1000 实现，CANopen 主要由软件编程实现。

4.2 CAN 2.0B 协议的介绍^{[7][8][9][22]}

4.2.1 CAN 2.0B 协议的分层结构

CAN 遵循 ISO/OSI 标准模型，成为数据链路层（包括：逻辑链控制（LLC）子层和媒体访问控制（MAC）子层）和物理层。

LLC 子层的主要功能包括：帧接收滤波、超载通告和恢复管理。

帧接收滤波：帧内容由标识符确定，每个接收器通过帧接收滤波确定此帧是否与其有关；超载通告：当数据帧和远程帧的发送需要延迟时，由 LLC 子层开始发送超载帧，其最多可以产生两个超载帧；恢复管理：在发送期间，对于丢失仲裁或被错误干扰的帧，LLC 子层具有自动重发的功能。

MAC 子层的功能主要是确定传送规则，即发送前确认总线是否空闲，控制帧的结构、数据的封装及卸装，执行仲裁、应答校验，错误检测、出错标定、故障界定、输出或接收串行位流、位定时等。MAC 子层特性不存在修改的灵活性。

在现场总线协议中 MAC 子层最重要，其特性直接与总线的实时能力相关。CAN 总线的 MAC 层协议属于事件触发方式中的载波检测、多主掌控/冲突检测（CSMA/CD）的通讯模式，采用这种通讯模式，当多个节点同时向总线发信息时，优先级较低的节点会自动退出发送，而最高优先级的节点可不受影响地继续传输数据，从而大大节省了总线冲突仲裁时间。但也由于缺乏全局控制，在发生信息碰撞时容易导致响应时间不确定，在控制系统中产生不确定性延迟。

物理层的功能是规定有关全部电气特性，不同节点间位的实际传送和位的编码/解码，CAN 协议并没有对物理层电气特性做具体的规定，但一个网络中所有节点的物理层特性必须是相同的。

4.2.2 CAN 的帧格式

CAN 2.0B 协议中存在两种不同的帧格式，其区别在于仲裁场中标识符的长度：具有 11 位标识符的帧称为标准帧，具有 29 位标识符的帧称为扩展帧。

在本文中用到标准格式，其格式如下：

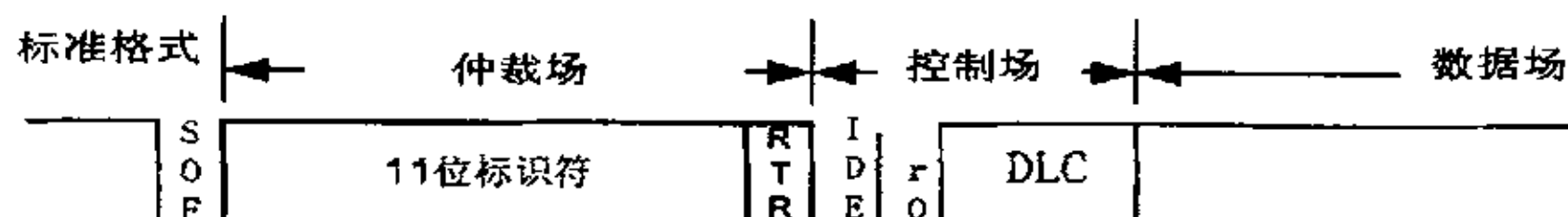


图 4-1 CAN 标准帧的数据格式

SOF “隐性”电平是帧起始位，11 位的标识符一般包括自定义的帧类型和目标节点的 ID 地址，RTR 为 “显性” 电平代表此帧为数据帧，为 “隐性” 电平代表此帧为远程帧。控制场的前两位作为控制总线发送电平的备用位，后面四位是数据长度码，用于指定数据场中的字节数目（0~8）。

4.2.3 CAN 的帧类型

CAN 的信息帧有四种帧类型：数据帧、远程帧、出错帧和超载帧。数据帧携带数据由发送器到接收器，有 7 个不同的位场组成：帧起始、仲裁场、控制场、数据场、CRC 场、应答场和帧结束，数据场的长度可为 0 到 8 个字节。其中仲裁场的标识符和数据场的具体数据由用户由单片机通过软件定义，其他位场均由 CAN 通讯控制器 SJA1000 硬件自动生成。

远程帧通过总线发送给相应的节点，以请求该节点中具有相同标识符的数据帧；远程帧与数据帧格式的区别是，前者的 RTR 位必须为 “隐性” 电平而后者必须为 “显性” 电平，且远程帧不存在数据场。

出错帧由通过检测发现总线错误的任何节点发送；由两个不同的场组成：第一个场由来自各站的错误标志叠加得到，第二个场是出错界定符。

超载帧用于提供当前的和后续的数据帧和远程帧的附加延时，包括两个位场：超载标志和超载界定符。

4.2.4 CAN 总线访问冲突仲裁方式

CAN 采用非破坏性的 CSMA/CD 竞争方式解决潜在的总线访问冲突，不丢失数据和带宽。若有两个或更多的 CAN 节点同时开始向总线发送数据，总线访问冲突通过仲裁场发送期间位仲裁的处理方法予以解决。具体方法是：标识符被逐位发送，每个位都可以是 “0” 值（显性位）或 “1” 值（隐性位）。当一个显性位和一个隐性位被同时发送到网络上时，网络上的最终结果呈现为显性位。在进行发送时，每个发送方检测网络当前值是否和自己已经发送位的值相向，若相同则继续发送；反之则终止当前发送进程，转为接收器，然后等待其它节点当前的发送完毕后再重新开始发送。依据以上规则的仲裁，最终 11 位标识符实际数

值较小的帧在仲裁中获胜，获得总线访问权。其它节点的信息将重新发送，如果再次与其它的信息帧发生冲突，则依以上规则进行新一轮的仲裁。

4.3 CANopen 应用层协议介绍^{[23][24][26][27][33]}

4.3.1 概述

CANopen是建立在CAN协议的物理层和数据链路层基础上的标准化的应用层协议。CANopen的标准化是指设备接口和通讯协议(CiA DS-4XX)定义了标准化的应用对象和基本功能。CANopen 网络管理服务(NMT)简化了项目设计系统集成和诊断。在每个分散的控制应用中都有各自所需的不同的通信模式(PDO、SDO)，在CANopen 总线上所有这些通信模式的相关参数都在对象字典中详尽地进行了描述。对象字典是整个CANopen协议的核心。

CANopen协议的框架和应用如图4-2所示：

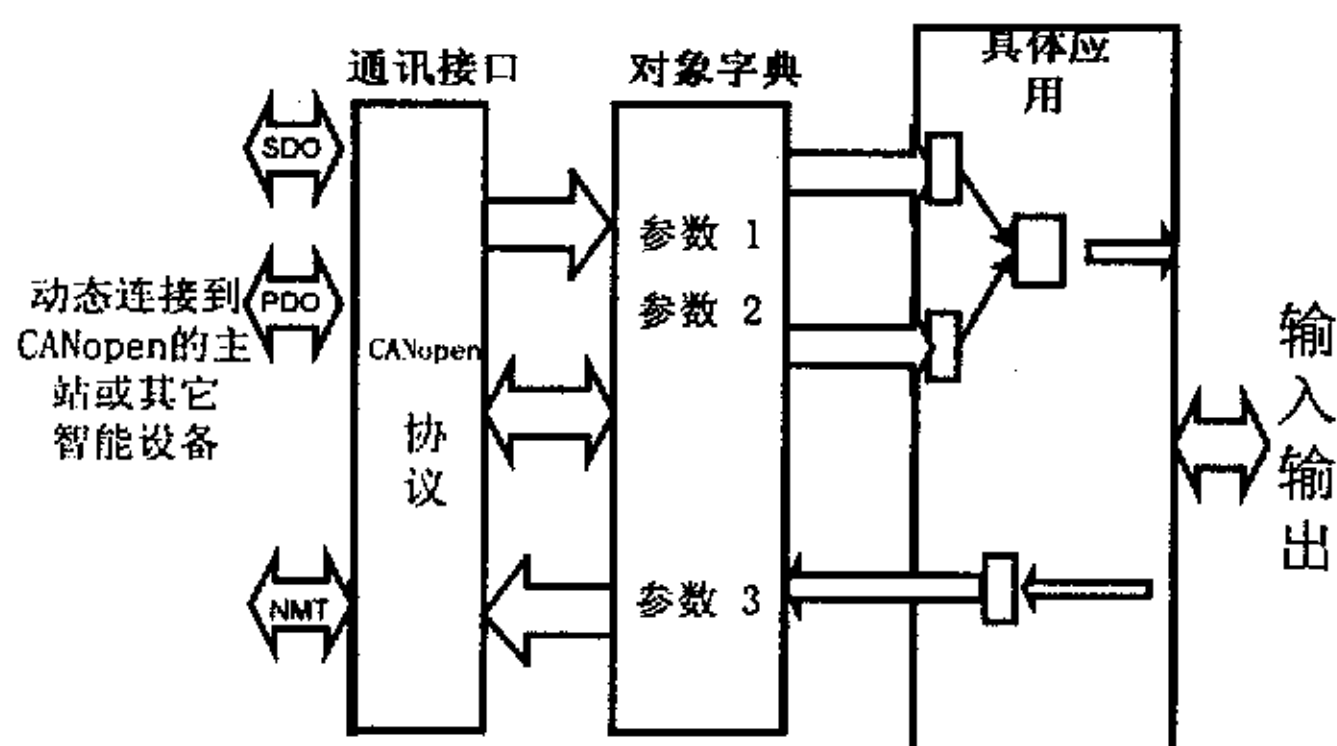


图 4-2 CANopen 协议应用示意图

4.3.2 对象字典

CANopen 对象字典是一个参数列表，定义了设备参数、各个通讯参数并存储应用中输入输出的数据，描述了设备的全部应用对象。可以用一个16 位的索引进行存取，另外用一个8 位的子索引描述每一个参数具体定义的子项。对象字典最大占64K的存储空间，特定的参数类型分布在临近的寻址区域，如表4-1所示：

索引区域	参数功能
0001H~009FH	定义数据类型
00A0H~0FFFH	保留
1000H~1FFFH	与通讯相关的参数
2000H~5FFFH	用户自己定义的特定参数
6000H~9FFFH	与特定设备相关的参数，用于存储相关数据
A000H~FFFFH	保留

表4-1 数据字典索引的定义

对象字典虽然占很大的空间，但并不是所有的参数都要定义。其中必须要实现只有 0001H~009FH 和 1000H~1FFFH 两部分，2000H~5FFFH 是用户自己定义的特定参数，用户可以根据自己的需要进行相应的定义，这一部分各个系统都有所不同。对象字典的常用参数如表 4-2 所示：

索引	子索引	参数名称	类型和特征	范围	默认值	M/O
1000H	0H	设备类型	U32, ro,	No	No	M
1001H	0H	错误寄存器	U8, ro	No	No	M
1003H	0H	预定义错误域	U8, ro	1	No	M
	1H	错误历史	U32	No	No	M
1004H	0H	支持PDO数目	U32, ro	0H-1FF0 1FFH	No	O
	1H	同步PDO数目	U32, ro	同上	No	O
	2H	异步PDO数目	U32, ro	同上	No	O
1005H	0H	SYNC的标识符	U32, rw	No	80H	O
1006H	0H	通讯循环周期	U32, rw	No	00000000H	O
1007H	0H	同步时间窗宽度	U32, rw	No	00000000H	O
1008H	0H	设备名称	Vstr, ro	No	No	O
1009H	0H	硬件版本号	Vstr, ro	No	No	O
100AH	0H	软件版本号	Vstr, ro	No	No	O
100BH	0H	节点地址	Vstr, ro	1-127	No	
100CH	0H	节点保护时间	U16, rw	No	0H	O
100DH	0H	保护时间因数	U8, rw	No	0H	O
100EH	0H	节点保护标识符	U32, rw	No	700H+节点ID	O
100FH	0H	支持的SDO数目	U32, ro	Se 1-80h Cl 0-80h	No	O
1012H	0H	时间标记帧标识符	U32, rw	No	100H	O
1014H	0H	应急对象标识符	U32, rw,	No	80h+节点ID	O
1015H	0H	应急对象禁止时间	U16, rw,	No	No	O
1200H- 127FH (SDO 服务器 参数)	0H	入口数目	U8, ro	2	No	O
	1H	Ser到Cli的 标识符	U32, 1200H ro 1201~127Fh rw	No	1200H 600h+节点ID 其余无	O
	2H	Cli到Ser的 标识符	U32, 1200H ro 1201~127Fh rw	No	1200H 580h+节点ID 其余无	O
	3H	SDO客户的节点ID	U8, rw	No	No	O
1280H- 12FFH SDO客户 参数	0H	入口数目	U8, ro	3	3	O
	1H	Ser到Cli的标识符	U32, rw	No	No	O
	2H	Cli到Ser的标识符	U32, rw	No	No	O
	3H	SDO服务器的 节点ID	U8, rw	No	No	O

接收PDO 通讯参数	0H	入口数目	U8, ro	2-4	No	O
	1H	接收到PDO的标识符	U32, rw	No	1400H 200H+节点ID 1401H 300H+节点ID	O
	2H	传输类型	U8, rw	No	依设备协议而定	O
	3H	禁止时间	U16, rw	No	No	O
	4H	CAN协议优先级	U8, rw	0-7	No	O
1600H- 17FFH	0H	PDO映射的数目	U32, rw	1-64	依设备协议而定	O
	1H 40H	第n个PDO信息映射的格式	U32, rw	No	依设备协议而定	O
发送PDO 通讯参数	0H	入口数目	U8, ro	2-4	No	O
	1H	要发送PDO信息的标识符	U32, rw	No	1800H 180H+节点ID 1801H 280H+节点ID	O
	2H	传输类型	U8, rw	No	依设备协议而定	O
	3H	禁止时间	U16, rw	No	No	O
	4H	CAN协议优先级	U8, rw	0-7	No	O
1A00H- 1BFFH	1H- 64H	发送PDO信息的映射格式	U32, rw	No	依设备协议而定	O

表4-2 对象字典中的常用参数

其中, 类型与特征中U32 (16, 8) 是指数据为无符号的32 (16, 8) 位数据; rw表示只写型数据, ro表示只读型数据; 此表中的默认值是指系统上电进行初始化时即设定此值, 有规定的是CANopen协议对参数所要求的值, 没有规定的大部分还要在实际应用中加以设定。M/O是指该参数是CANopen规定的必选值 (M) 还是可选值 (O)。通讯时对对象字典的操作在每一种通讯模式中分别阐述。

4.3.3 CANopen网络中主节点和从节点的概念

在CANopen网络中, 主节点负责各种网络管理, 并对网络中的通讯做合理的调度。在本论文中设定上位机为主节点, 其它节点为从节点。每个从节点具有固定的节点ID。

主节点主要的功能是: 初始化时控制从节点的状态; 程序运行中间启动或终止从节点, 或改变从节点的状态; 通过SDO通讯对对象字典的参数进行读取和修改; 监视从节点的运行状况和是否发生错误 (即进行节点保护); 由从节点中获得相应的信息和运行状态; 启动和终止SDO通讯; 发送SYNC报文; 提供分配特定标识符 (DBT) 等网络管理的服务。

从节点主要负责底层的网络通讯和控制任务。每个从节点只完成属于自己范围内的特定任务, 进行实时数据的传输 (PDO通讯), 响应主节点发送的管理信

息帧，对自己负责的底层设备进行数据采集和控制。

对于 CANopen 网络中必须实现的管理功能，一般由一个节点来实现，但也可以由几个节点分别承担。尤其对于 SYNC 报文传输和标识符的分配，分开来实现会更好。一个节点是否为主节点在对象字典中设置，在软件编程中，主节点和从节点有很大的不同。

4.3.4 PDO通讯

在CANopen网络中，有两种主要的通讯模式：PDO（Process Data Object）通讯和SDO（Service Data Object）通讯。两者比较如下：

PDO通讯	SDO通讯
传输实时数据	以索引和子索引的方式访问和设置对象字典
传送优先级较高的信息	传送优先级较低的信息
分为同步和非同步两种传送模式	非同步传送模式
分为周期和非周期两种传送模式	非周期传送模式
数据内容可由SDO设置	数据内容由CMS协议决定
和在CAN总线上任意两节点间进行	只限于主节点和从节点间进行
一次可传送0~8个字节	一次可传送任意多个字节

表4-3 PDO通讯和SDO通讯的比较

一、PDO通讯具有的三种模式：

1、同步通讯模式：

同步通讯模式要求在同步时间窗内周期性的发送数据。主要应用于与时间有关的闭环控制和运动控制。其通讯过程如图4-3所示：

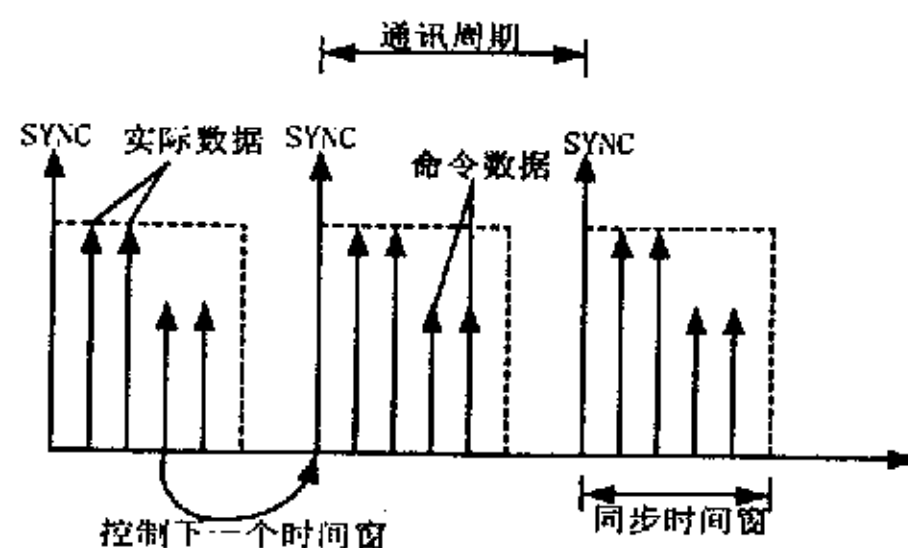


图4-3 PDO同步通讯模式

同步报文(SYNC)：在网络上周期性发送。

实际数据：代表节点的当前状态。

命令数据：代表节点将要到达的新状态。

该模式用于闭环控制，例如节点1是闭环中的检测单元，节点2是执行单元，那么节点1首先将检测到的实际数据发送给节点2，节点2根据这个数据经过一系列的控制算法得出节点1应达到的状态，并作为命令数据发送给节点2，希望在下

一个时间窗内得到的实际数据的是命令数据的值。

即一个时间窗内包括这个周期内检测到的实际数据,和下一个周期内将要达到的期望值。实际数据必须在收到同步报文(SYNC)后才可以发送,SYNC的发送频率由对象字典中的“传输类型”的参数定义。

2、事件触发模式:

通讯由某一事件触发。例如数字I/O口状态改变,超过预先设定的值都会触发节点发送一个状态短信。这种模式可使总线负载达到最小,在相对低的波特率下获得比较高的通讯特性。事件驱动模式和同步通讯模式可以在网络中自由的混合使用。

3、查询模式:

网络中的节点可以通过查询模式从其它任意节点上读取过程数据。该节点发送一个CAN远程帧,当其它节点收到此远程帧时,就会向总线发送相应的PDO数据。

三种通讯模式的区别如图4-4所示:

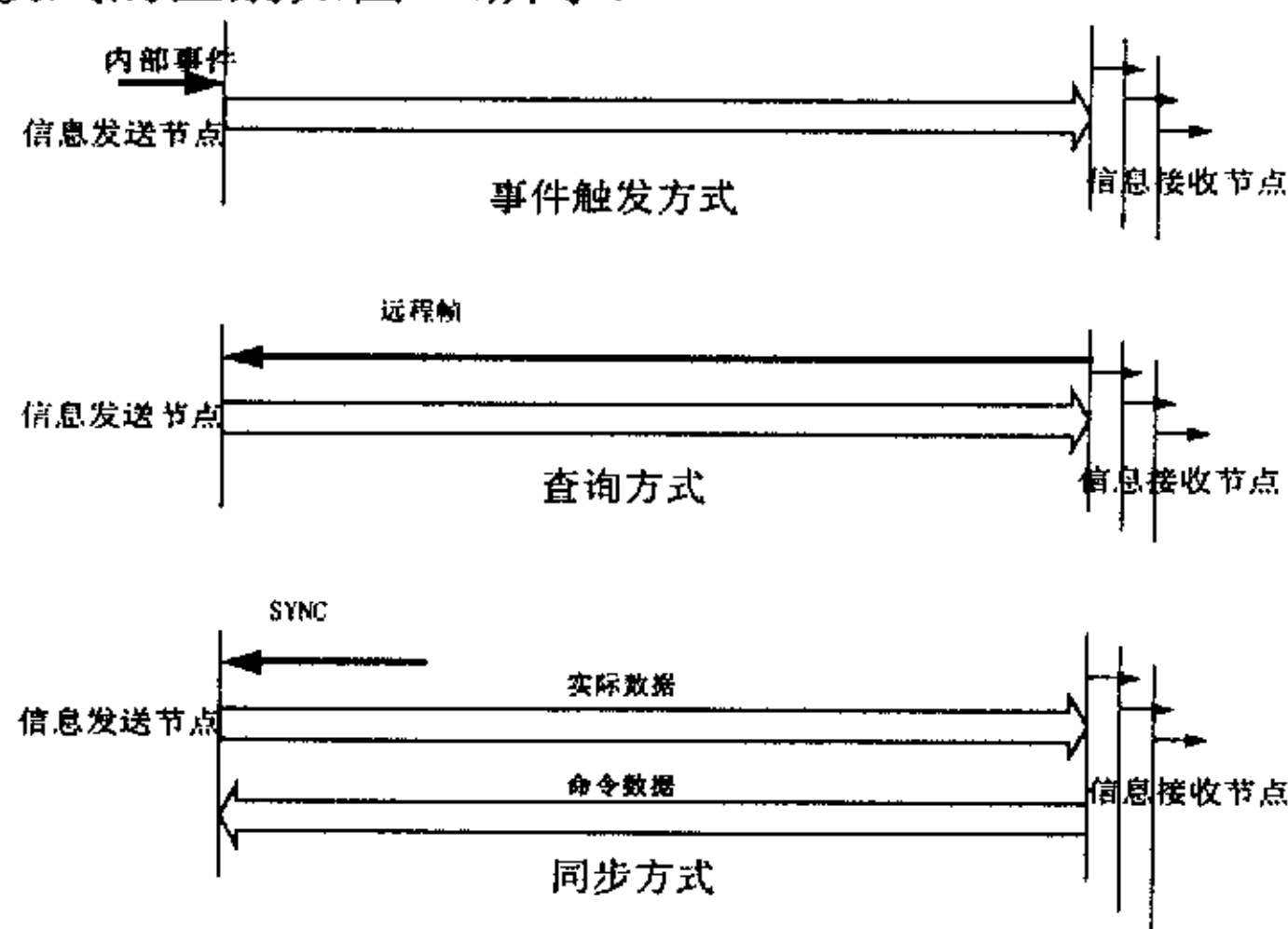


图4-4 三种PDO方式的比较

二、标识符的确定:

PDO标识符的确定有两种方式:默认值方式和动态分配(DBT)的方式。

默认值方式是指针对简单的网络,为了减少网络设置的麻烦,CANopen定义一种标识符的默认分配方式。标识符的默认值在系统初始化后的预操作状态中在对象字典的相应位置上设定,通讯时由PDO的映射获得。其包括功能代码部分和节点地址部分,其中功能代码部分占第一字节的高4位,代表着数据帧的优先级,节点地址部分占后七位,用于区别不同的节点。

动态分配方式有DBT服务来完成,由主节点与需要动态分配标识符的从节点建立起点对点的对话通讯,通过改写对象字典的相关项分配标识符。这种方法需要大量的软件编程。

三、禁止时间：

PDO通讯的优先级较高，当其数据量很大时，很容易长时间占据总线使得其它数据无法进行传输。应用禁止时间服务可以避免这种问题的出现。禁止时间服务是指在对象字典中定义一个时间域，当某种标识符的数据的传输时间超过了这个时间域时，将被禁止传输，将总线释放，允许其它标识符数据的传输。

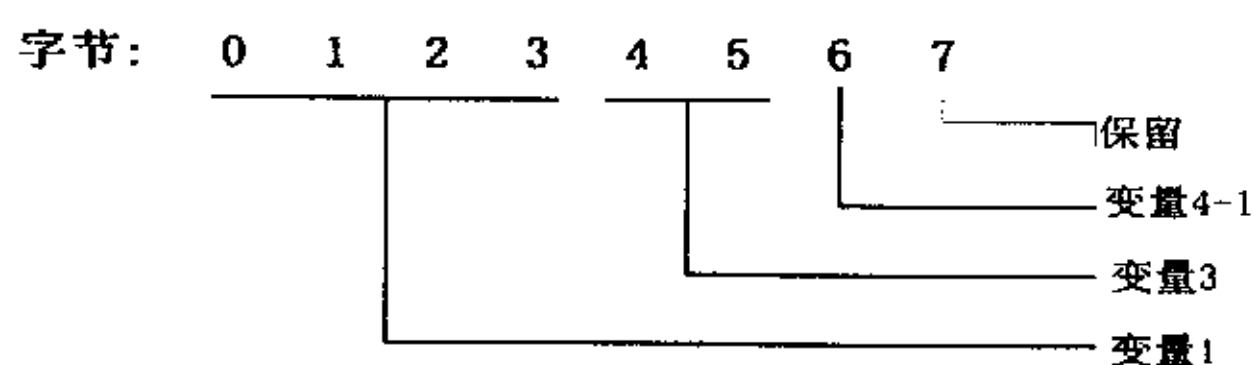
四、PDO通讯的映射封装：

当进行PDO通讯时，需要确定其通讯的具体方式（包括通讯模式、标识符、禁止时间、优先级等），以及8个字节数据帧的具体内容。通讯方式通过读取对象字典中1400H～15FFH（接收PDO）和1800H～19FFH（发送PDO）的相应参数的默认值确定；数据帧的具体内容由单片机根据实际应用确定，并存放在对象字典中6000H～9FFFH的相应位置上，通讯时，将确定通讯模式的相关信息和具体数据帧的地址信息按CANopen的映射规则写入1600H～17FFH（接收PDO）和1A00H～1BFFH（发送PDO）中。这个确定通讯方式和数据帧内容的过程叫做PDO的映射。其映射过程如下所示：

例如：下表是某节点对象字典的一部分：

索引	子索引	对象名称	数据类型
6060H	-	变量1	Unsigned16
6091H	-	变量2	Unsigned32
60AEH	-	变量3	Unsigned32
60CAH	0H	变量4-0	Unsigned8
	1H	变量4-1	Unsigned8
	2H	变量4-2	Unsigned8

现要按以下格式设置要发送的PDO数据帧：



那么须往对象字典中PDOmapping结构的1A00H中写入以下值：

子索引	写入值	注 释
0H	3	输出数据对象的个数：3
1H	60 60 00 10	第一个对象映射在：索引 6060.0 （长度为32位）
2H	60 AE 00 20	第二个对象映射在：索引 60AE.0 （长度为16位）
3H	60 CA 01 08	第三个对象映射在：索引 60CA.1 （长度为8位）

图4-5 PDO映射规则的示例

4.3.5 SYNC对象

SYNC对象由作为SYNC生产者的节点，同时发送给多个SYNC消费者。为SYNC消费者（如同步PDO通讯）提供同步信号，使其周期性的执行同步任务。

与SYNC对象相关参数由对象字典的1005H项确定：

位	31	30	29	28 ~ 11	10 ~ 0
11位标识符	0/1	0/1	0	00000000000000000000	11位标识符
29位标识符	0/1	0/1	1	29位标识符	

表4-4 对象字典的1005H项中SYNC对象相关参数的含义

其中31位为“0”表示该节点不消费SYNC对象，为“1”表示消费SYNC对象；30位为“0”表示给该节点不产生SYNC对象，为“1”表示产生SYNC对象。

SYNC对象只具有包含标识符的相应字节，不包含数据项，由于SYNC对象要与其他节点提供同步信号，对实时性要求很高，因此SYNC对象具有很高的优先级，一般默认为128。

4.3.6 SDO通讯

一、SDO 通讯的概述：

SDO 通讯用于主节点对从节点对象字典的读写访问，以实现从节点参数的设置，下载程序，定义 PDO 通讯类型和数据格式等，优先级较低。SDO 通讯需建立起两个节点间点对点的通讯，允许传送任意长度的数据，可以多于 8 个字节。传输中使用客户和服务器的概念，两节点中主动的节点是客户，被动的节点是服务器。

二、SDO 通讯数据结构和总体传输模式：

传输数据的结构如表 4-5：

CAN 标识符	控制信息	索引和子索引	组态数据	控制信息	组态数据	以后各段	与第二段相似
第一段 0 字节	1~3 字节	4~6 字节	第二段 0 字节	1~7 字节			

表 4-5：SDO 数据格式

SDO通讯可以传送任意长度的数据，如表4-5所示，当小于等于4个字节时，可以由一帧数据进行传送；当大于4个字节时，需要分段进行传送。SDO通讯需要响应确认，确认的方式有两种：每发送一帧的数据段确认一次；发送一个数据帧的数目小于127的数据块确认一次。

另外一种发送模式是以数据块单位进行发送，每发送一个 8 个字节的数据帧不需要确认，当数据块整体发送完以后进行总体确认，并进行 CRC 错误检验。

三、SDO通讯的标识符分配：

SDO 通讯的标识符一般有两种:

数据由主节点传送到从节点: 1100h + 从节点 ID

数据由主节点传送到从节点: 1101h + 从节点 ID

四、SDO 通讯的读对象字典的操作:

主节点为客户从节点为服务器时, 即主节点由从节点的对象字典中读取数据; 读对象字典的操作因读取数据长度的不同而不同。

当数据小于等于 4 个字节时其传送过程如图 4-6 所示:

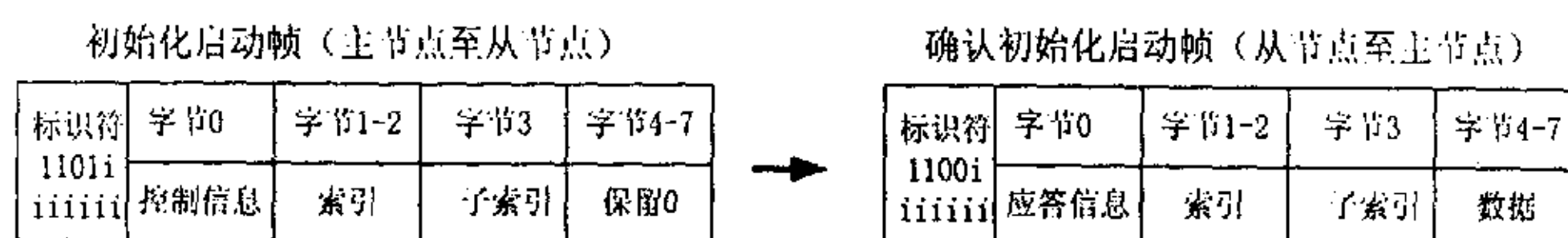


图 4-6 数据长度不大于 4 字节的读对象字典的 SDO 通讯

其中控制信息是 aaabbbbb: aaa = 010 (ccs = 2: 读对象字典初始化), bbbbb = 00000 (保留); 应答信息是 aaabcces: aaa = 010 (scs = 2: 读对象字典初始化确认), b = 0 (保留), cc = 数据字节的个数 (4~7), e = 1 (e = 1: 快速传输), s = 1 (s = 1: 数据长度被指定)。

当数据大于 4 个字节时其传送过程如图 4-7 所示:

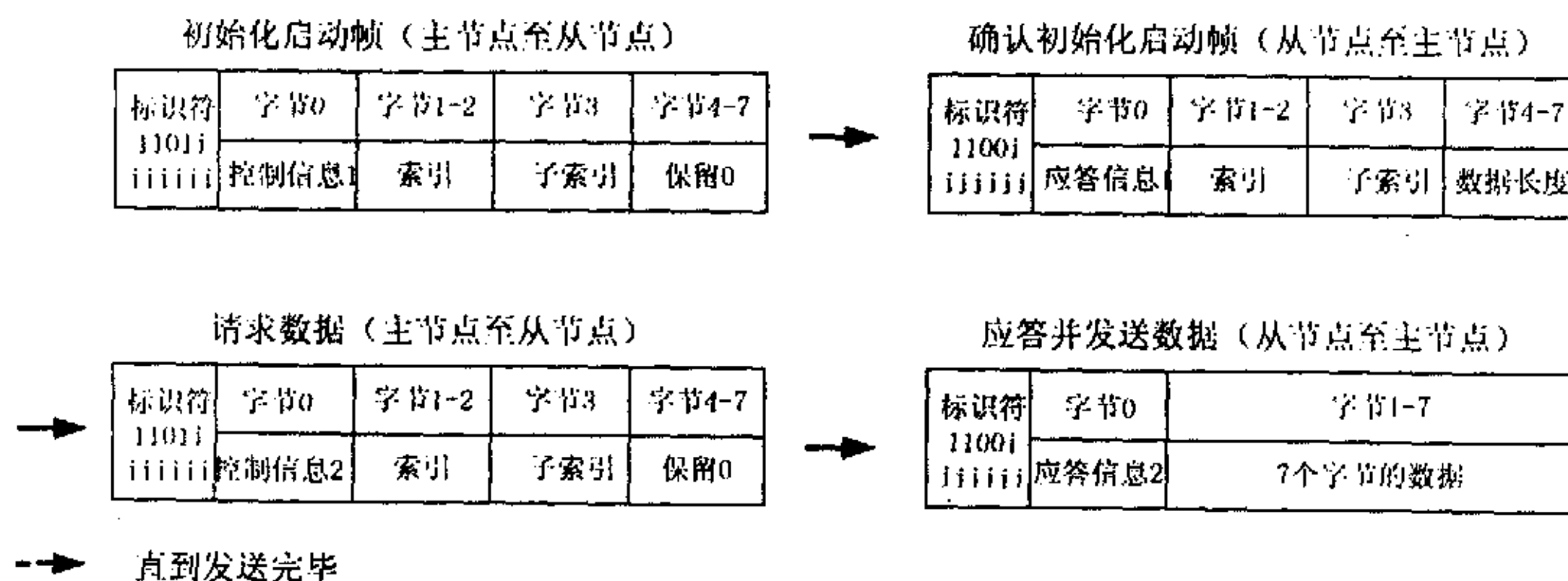


图 4-7 数据长度大于 4 字节的读对象字典的 SDO 通讯

控制信息 1 同上, 应答信息 1 中的 cc = 00 (无效), e = 0 (正常传送); 控制信息 2 为 aaabccccc, aaa = 011 (ccs = 3: 上传数据请求帧), b = 0/1 (为双态位, 第一次发送为 0, 若连续传送其状态改变, 从节点应答时与收到的状态相同, 若出现异常的数值, 则认为其产生错误, 放弃传输。) ccc = 0000 (保留); 应答信息 2 为 aaabccce, aaa = 000 (scs = 0: 响应上传数据请求帧), b = 0/1 (为双态位), ccc 表示上传数据的字节数, e = 1 (分段数据的最后一帧)。

6. SDO 通讯的写对象字典的操作:

从节点为客户主节点为服务器时, 即主节点往从节点的对象字典中写入数据以修改对象字典。

写对象字典的操作也分为小于等于 4 个字节和大于 4 个字节两种情况, 其传

输帧结构与读对象字典相似；过程相反，由主节点到从节点传送数据，由从节点到主节点的相应字节为保留值 0。

其控制信息与读对象字典的应答信息相似，只是 $aaa = 001$ ($ccs = 1$: 初始化下载数据请求)，应答信息的 $aaa = 011$ ($scs = 3$: 响应下载数据请求)。

4.3.7 主要的网络管理 (NMT)

NMT 用于管理和监控网络中的各个节点。网络中确定一个节点为 NMT 主节点 (即网络管理员)，其作为主节点时不和其它节点一起参与仲裁，只具有管理功能，不执行管理功能时主节点也可以运行应用程序参与 SDO 和 PDO 通讯。

网络管理有如下几种：

1. 状态管理：

每个 CANopen 从节点都有初始化、预操作、操作和停止四个状态，状态管理由主节点控制各状态之间的转换。

(1) 各状态的所完成的功能：

初始化状态分为三个子状态完成对节点的初始化。应用初始化由厂家设定设备相关参数的默认值，使设备标准化；通讯初始化对与通讯有关参数进行设置；初始化子状态完成硬件设备的参数设置。三个子状态一般依次进行。

在预操作状态中：

** 将 CANopen 通讯中信息标识符设为默认值，或通过 SDO 通讯改变对象字典的参数设置。

** 模块准备接收或发送有关状态管理的信息。

** 可进行 SDO 通讯，不可进行 PDO 两种通讯。

操作状态中所有的通讯功能将被激活，允许进行 SDO 和 PDO 通讯。

停止状态将禁止 SDO 和 PDO 通讯，使节点完成特定的不需要进行通讯的功能。

(2) 各状态之间的转换，各状态之间的转换如图 4-8 所示：

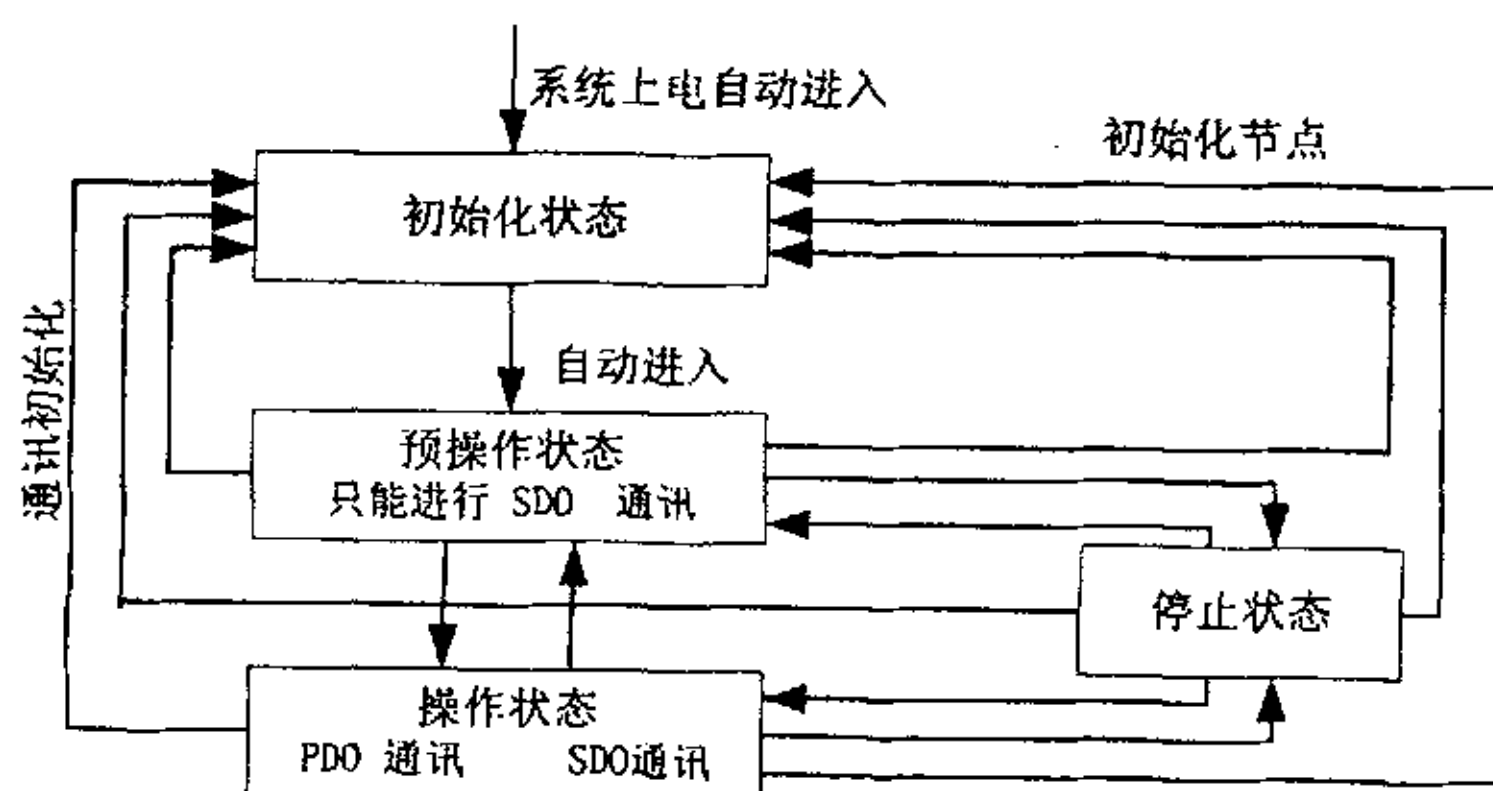


图 4-8 节点的工作状态及相互转换

各个状态可以在主站的控制下进行转换。当系统上电后，依次完成三个初始化子状态，自动进入预操作状态中并发送一个标志信息；由预操作状态进入操作状态需要主节点发送一个两字节的 NMT 报文启动远程节点；反之主节点发送另外一个两字节的 NMT 报文使节点进入预操作状态；节点可以由任意一个状态进入初始化状态中的后两个子状态，完成节点的硬件初始化和通讯初始化。

2. 节点保护：

节点保护通过主节点定期发送远程帧从节点进行响应的方式来检测网络中的通讯错误。

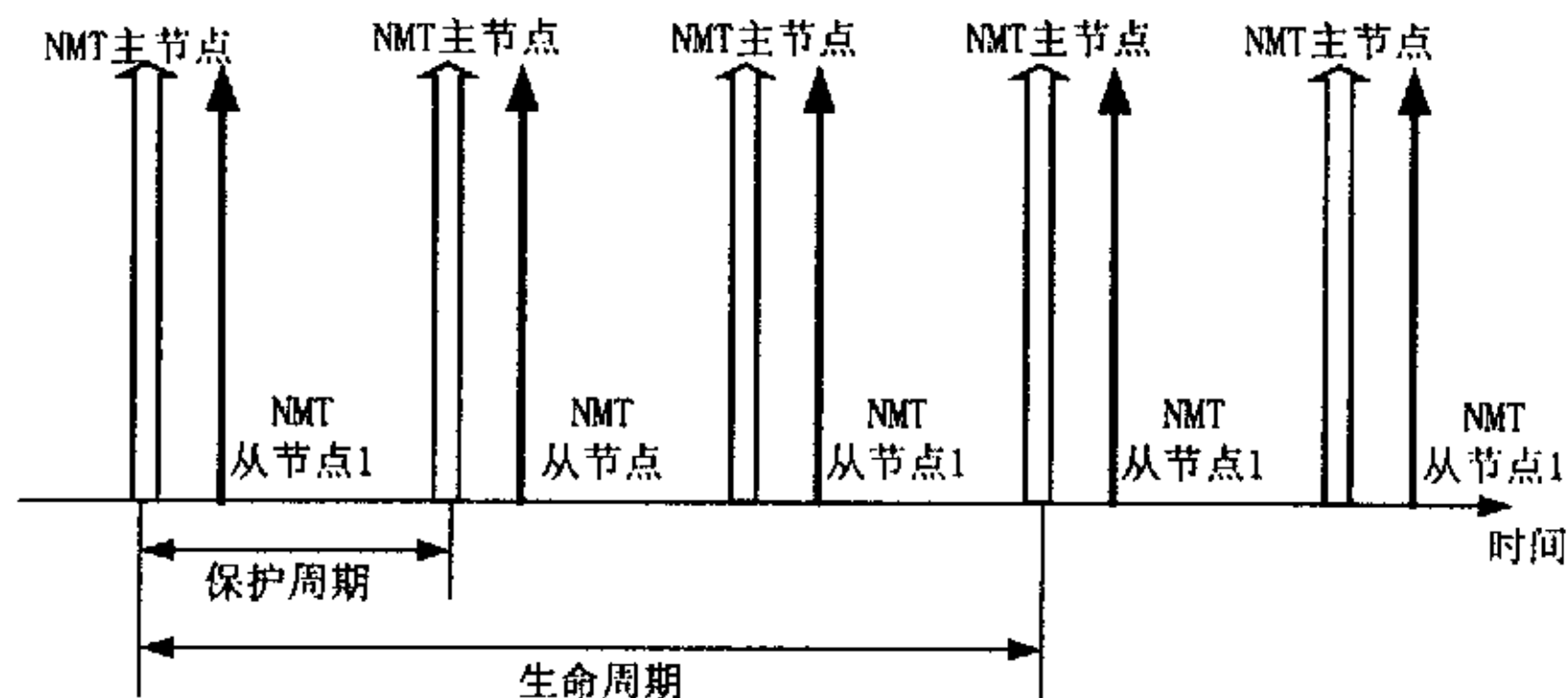


图 4-9 节点保护示意图

CANopen 网络中，主节点中建立一个数据库，包括了其它各从节点在一段时间内期望达到状态的信息。主节点周期性的向从节点发送远程帧，从节点则向主节点回送一个包括其状态信息的字节作为响应。如图 4-9 所示，主节点将收到的状态信息与自己数据库中信息进行比较，若从站没有在如图规定的保护周期内响应或其操作状态不正确，则断定从节点发生错误。主节点用“网络事件”服务发送远程错误帧标定从节点错误；同样的，从节点若在规定的生命周期内没有收到主节点发送的远程帧，也可确定主节点发生错误。从节点用“节点事件”服务发送远程错误帧标定主节点发生错误。

保护周期由对象字典中的 100CH 确定，生命周期由 100DH 确定。

主节点的信息帧格式是：1100iiiiiii，包括 RTR 位，不包括数据信息，其中 iiiiii 是从节点的地址；从节点的信息帧格式是：1100iiiiiii tsss.ssss，其含义如下：

t 是双态位，与 SDO 通讯中的含义相同；sss.ssss 表示从节点的状态：

3 表示预操作状态

4 表示准备状态

5 表示操作状态，其中 50h 表示等待 ADC 响应；

51h 表示在时间闭环中等待内部操作的完成；

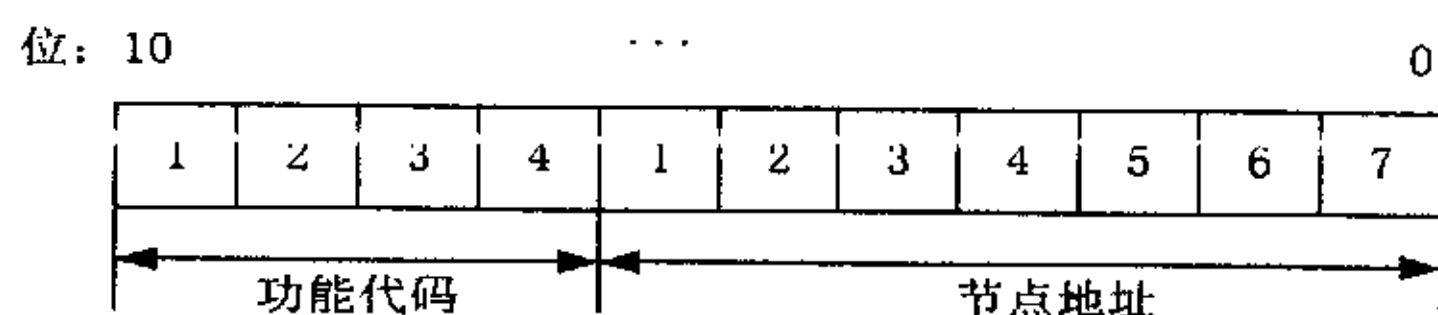
55h 表示等待 SYNC 信息帧。

4.3.8 各信息帧的优先级及标识符的默认值

CAN总线信息帧的优先级由CAN标识符来确定，标识符数值较小的优先级较大，在CANopen信息帧中用标识符的前四位表示信息帧的类型，因为它放在标识符的最高位上，其值的大小代表了信息帧的优先级。

本文根据常规控制系统中各种信息帧对实时性的要求，确定各信息帧优先级的大小和标识符中的功能代码，使优先级以标识符实际数值大小的形式得到体现。

11位的标识符格式如下：



标识符的默认值如下表：

通讯对象	功能代码	标识符的实际值	对象字典中相关参数的索引
NMT	0000	0	1005H, 1006H, 1007H
SYNC	0001	128 (80h)	1012H, 1013H
应急对象	0001	129 (81h)–255 (FFh)	1014H, 1015H
节点保护	0010	256 (100h)–384 (180h)	1016H, 1017H
PD01 (触发模式tx)	0011	385 (181h)–511 (1FFh)	1800H
PD01 (触发模式rx)	0100	513 (201h)–639 (639h)	1400H
PD02 (同步模式tx)	0101	641 (281h)–767 (2FFh)	1801H
PD02 (同步模式rx)	0110	769 (301h)–895 (37Fh)	1401H
PD03 (查询模式tx)	0111	897 (381h)–1023 (3FFh)	1802H
PD03 (查询模式tx)	1000	1025 (401h)–1151 (47Fh)	1402H
PD04 (保留tx)	1001	1153 (481h)–1279 (4FFh)	1803H
PD04 (保留rx)	1010	1281 (501h)–1407 (57Fh)	1403
SDO (tx)	1011	1409 (581h)–1535 (5FFh)	1200H
SDO (rx)	1100	1537 (601h)–1663 (67Fh)	1200H

表4-6 本系统各信息帧标识符的默认值

第五章 CAN控制器的软件设计

5.1 软件的设计思想

5.1.1 概述^{[17][19][20][21]}

CAN控制器的软件主要实现CANopen协议总线通讯和网络管理的功能，以及CAN控制器的节点控制模块和模块之间485通讯的功能。由单片机C语言C96、C51进行编写，采用模块化的结构。

CANopen协议内容十分丰富，可以是用于各种控制系统，完成各种功能，在一般的系统中，以下几种是常用的：初始化；对象字典的组建和读写访问；PDO通讯；SDO通讯；网络管理；错误的检测和处理。

在CANopen网络中，各种通讯的标识符可采用默认值，也可利用DBT服务进行动态配置，DBT服务需要大量的软件编程来实现，而在大多数系统中采用默认标识符的模式即可满足通讯的需要，因此在本设计中采用标识符的默认模式。

在本设计中485通讯用于完成节点控制模块和模块之间的数据传输，采用主从式全双工的串行通讯，软件编程属于常规设计。

5.1.2 CANopen网络中主节点和从节点的软件设计^{[27][28][29][30]}

在CANopen网络中，其主节点和从节点要实现的功能有差异，其软件编程也有很大的区别，在本文中分为主节点和从节点来说明软件的设计的思想。

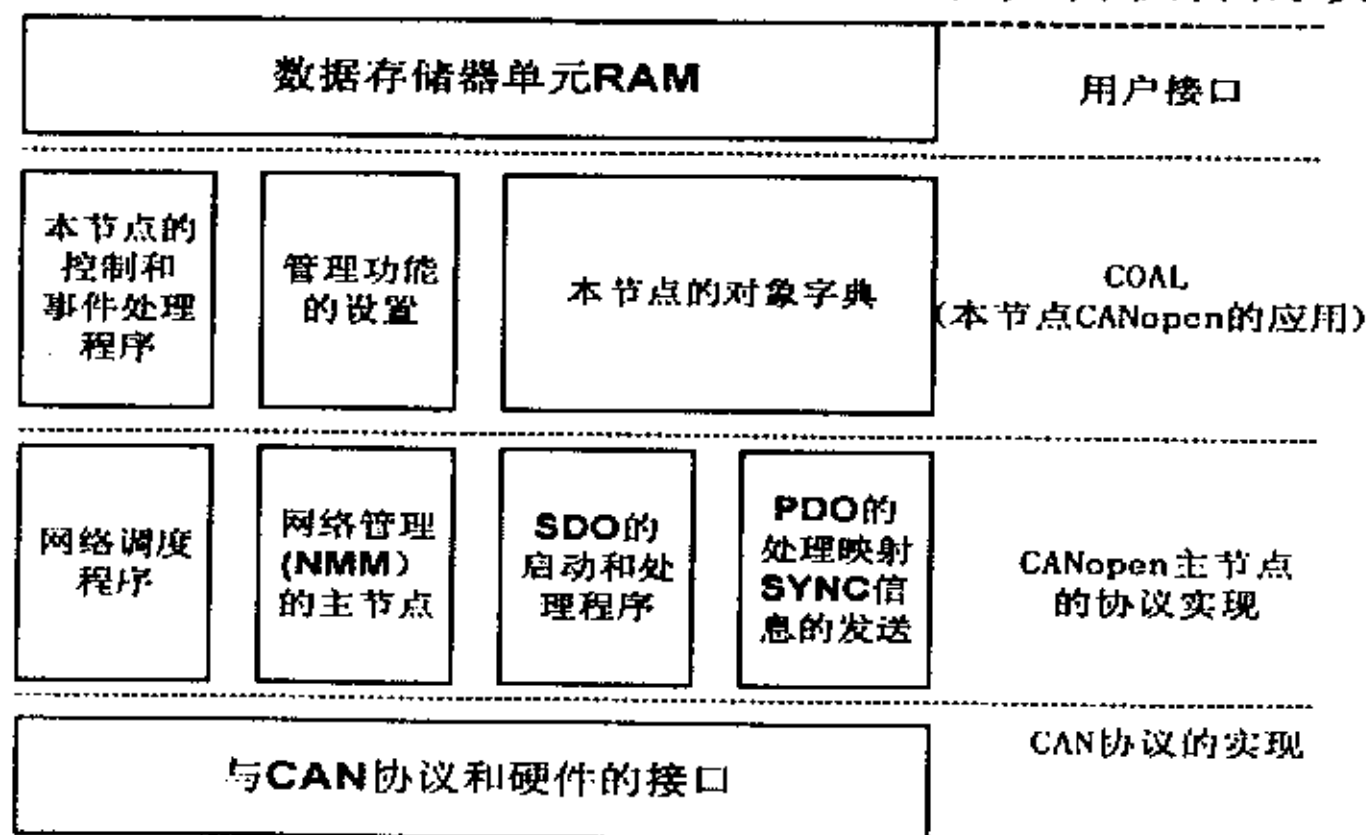


图5-1 主节点的软件结构图

主节点可以有一个节点来承担，也可以有多个节点承担，在本设计中采用PC适配卡作为主节点。其完成对整个网络中所有从节点的控制、监测和管理。主要包括初始化CANopen网络中的各个从节点并控制它们所处的状态；对各个从节点设置ID地址；在CANopen网络中减少或增加从节点的个数；启动SDO通讯；通过启动节点保护，检测网络中存在的错误；由从节点中获得需要的信息；周期性

的发送同步信息SYNC；进行NMT服务以控制网络上相连从节点的活动等。其程序结构如图5-1所示。

从节点主要完成实时数据的PDO通讯、控制底层设备；响应主节点相应的SDO通讯和网络管理；通过485通讯与底层I/O模块通讯并对其进行管理；辅助以上功能也需要对对象字典进行组建和访问以及对该节点自身的初始化。程序结构如图5-2所示，其中控制底层设备和进行485通讯的程序包含在框图中的应用程序接口中。

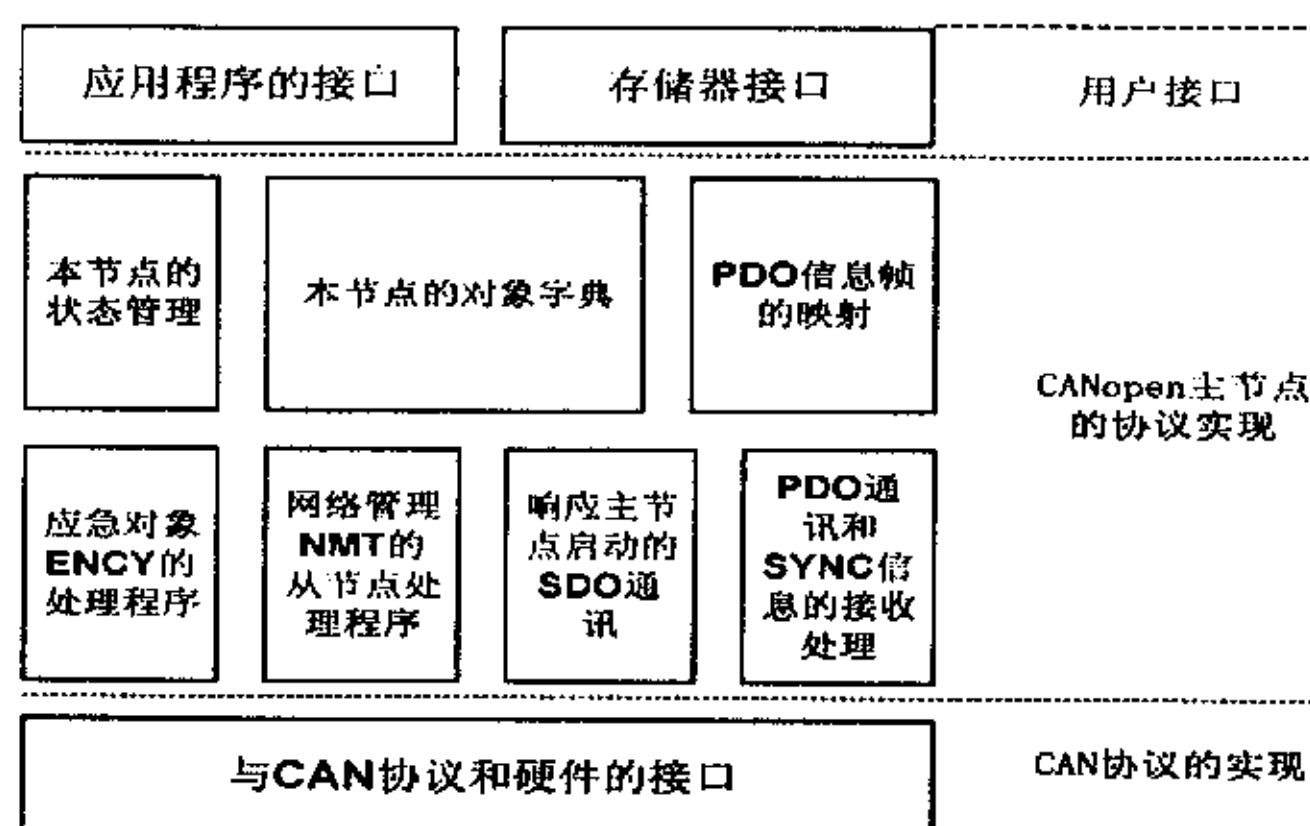


图 5-2 从节点的软件结构图

本文所设计的CAN控制器在整个CANopen控制系统中属于从节点，因此在软件设计上只对从节点进行具体设计，主节点只是在程序的结构和实现上做初步的设想，使整个系统的软件结构具有一定的完整性。

5.1.3 485通讯与I/O模块管理的软件设计

本设计中的I/O模块主要完成CAN控制器的数据输入和输出功能，与传统的集中式CAN总线控制器上的I/O端口的功能一致。因此节点控制模块和I/O模块的软件设计的组合就是完成整个CAN控制器与现场设备之间的数据交换功能。

节点控制模块在整个485通讯中是主节点，负责任何一次的数据发送与接收，并对与其连接的I/O模块进行监控和管理，使其能够正常的工作。同时对接收到的数据进行存储和处理。

I/O模块在整个485通讯属于从节点，每一个I/O模块都有自己的ID地址，在通讯中作为本节点的标识。I/O模块由现场设备中采集各种类型的数据，进行存储并组合成数据帧，在主节点请求时发送给节点控制模块；同时接收由节点控制模块发送的数据，并传输给现场设备。在每个从节点的软件设计中还包括不同类型的数据转换，比如AD，DA转换等。

5.2 主节点的软件设计

5.2.1 概述

主节点的软件可以在从节点软件的基础上加上网络管理部分得到,也可以作为网络管理的独立软件存在,独立的软件不具有PDO通讯模块,SDO响应模块和应急对象的发送模块,只完成网络的管理和监控的功能。在本设计中实现独立的网络管理监控功能。

在独立的网络管理软件中,不存在和从节点相似的对象字典,取而代之,设定各个功能模块的相应操作被集中存储在特定的数据库内,当要实现某个功能时,需要从数据库中调用相应的设定值。

本软件设计需要分配与各个功能相对应的数据存储区域。本文中设计了结果存储区(Post-Message-buff)和根据主节点调用功能模块动态分配的存储区。当某个功能模块真正被调用时,其所得数据存放在功能模块动态分配的存储区,当该模块的程序运行结束时,将以后有用的数据存放在结果存储区内,然后释放动态存储区,以备下一次程序调用时使用。这样保证了合理利用有限的存储资源。

主节点的软件设计共包括如下程序模块:[1] 初始化功能模块;[2] SDO通讯模块;[3] 存储应急对象处理模块;[4] 程序结果处理模块;[5] 节点保护模块;[6] 同步信息发送模块;[7] 网络NMT服务模块。

5.2.2 各个功能模块的介绍

1、初始化功能模块:

其实现对CANopen网络和主节点自身的初始化,又分为以下子模块:

- (1) Init_Canopen(): 初始化网络中与通讯相关的所有参数,包括CAN通讯速率、各个从节点的ID地址等。
- (2) Init_Master(): 在完成对网络初始化后,对主节点自身进行初始化。
- (3) Add_Node(): 在主节点中存储从节点地址的数据库中增加新地址,即识别在网络中新增加的从节点。

2、SDO通讯模块:

- (1) Add_Master_SDO(): 调用SDO通讯功能。
- (2) SDO_Ini_Upload_req(): 启动读对象字典的SDO通讯,可同时向多个从节点发送读对象字典的初始化数据帧。
- (3) SDO_Ini_Download_req(): 启动写对象字典的SDO通讯,对从节点的运行方式进行设置。
- (4) Abort_SDO(): 在SDO通讯过程中发生错误时中止SDO通讯。

3、接收存储应急对象模块:

- (1) Get_Emergency_Msg(): 接收从节点发送的应急对象信息帧,并发送一个接收确认标志。
- (2) Mana_Emergency_Msg(): 对接收到的应急对象信息帧进行分析并作相应

的处理。

4、程序结果处理模块：

当调用某个功能模块后，一些结果性数据需要保存起来，以后需要时重新读出供其它程序使用，该模块完成以上功能。

- (1) Init_PostMessage(): 初始化结果存储区 (Post-Message-buff)。
- (2) Write_PostMessage(): 将程序结束后的结果数据存储到结果存储区相应的位置。
- (3) Read_PostMessage(): 需要时从结果存储区中读出相应的数据并作处理。

5、节点保护模块

- (1) Set_Lifeguard_Parameter(): 设置节点保护数据帧，发送数据的时间间隔等信息。
- (2) Send_Lifeguarddata(): 周期性的封装并发送节点保护数据帧。

6、同步信息发送模块：

- (1) Add_Sync_Msg(): 封装同步信息并发送。

7、网络NMT服务模块：

用于控制各个从节点的活动，比如从节点的状态，重新初始化以及设定PDO通讯的开始以及中止等。

- (1) NMT_Network_Service (): 设置相关参数并发送命令数据帧。

5.3 从节点的软件设计

从节点中实现CANopen协议的软件内容十分丰富，从功能上可分为三个组成部分：基本功能部分，其包括节点的通讯初始化和硬件设备初始化；定义和访问对象字典；PDO通讯；SDO通讯；节点保护等功能；错误检测处理和节点管理功能部分，当系统软件发生错误或系统状态需要发生改变时，调用这部分软件。这一部分需要用户根据实际应用的具体情况进行编写；扩展功能部分，这一部分用于在系统的硬件发生变化时对节点的状态和相关参数进行重新设置，并管理对象字典进行相应的管理。

本文属于研究性设计，没有具体的应用对象，因此软件设计只完成最基本的通讯、初始化和部分网络管理功能，即上述基本功能部分。

5.3.1 软件设计思想

一、软件包括的功能部分：

**** 初始化部分：**包括初始化CANopen协议中的通讯相关参数，其中包括对象字典、PDO通讯参数、SDO通讯参数等；初始化CAN控制器硬件，包括单片机的内部寄存器、定时器、串行口，SJA1000设置等；

- ** 组建对象字典：定义对象字典的各个变量并为每个变量设定一个指针；
- ** 访问本节点的对象字典：包括从本节点对象字典中获得某一参数的地址和具体数值，或修改相应的参数值。
- ** PDO信息帧的发送：将需要发送的数据封装成PDO数据帧的形式等待发送，需要时间触发方式的数据立即可以发送，需要同步模式的数据则需等接收到SYNC信息帧才进行发送；
- ** PDO远程信息帧的发送：产生并发送CAN远程帧。
- ** PDO信息帧的接收：当SJA1000产生接收中断时，单片机接收并断定为PDO信息帧并进行分析存储；
- ** SYNC信息的分析处理：从节点对接收到的SYNC信息帧进行分析并启动相应的操作。
- ** SDO通讯：当收到主节点的SDO初始化信息帧，响应并接收或发送主节点要求的数据帧；
- ** 节点保护：响应主节点发送的节点保护信息帧，发送本节点的状态信息；
- ** 启动和中止CANopen节点：在没有主节点参与的情况下，启动或中止本节点。
- ** 根据具体的需要设计相应的控制算法；
- ** 完成与I/O模块的通讯与管理。

二、软件结构设计^{[46][47][48][50][51][53]}：

本文按如下结构设计软件，完成上述功能：

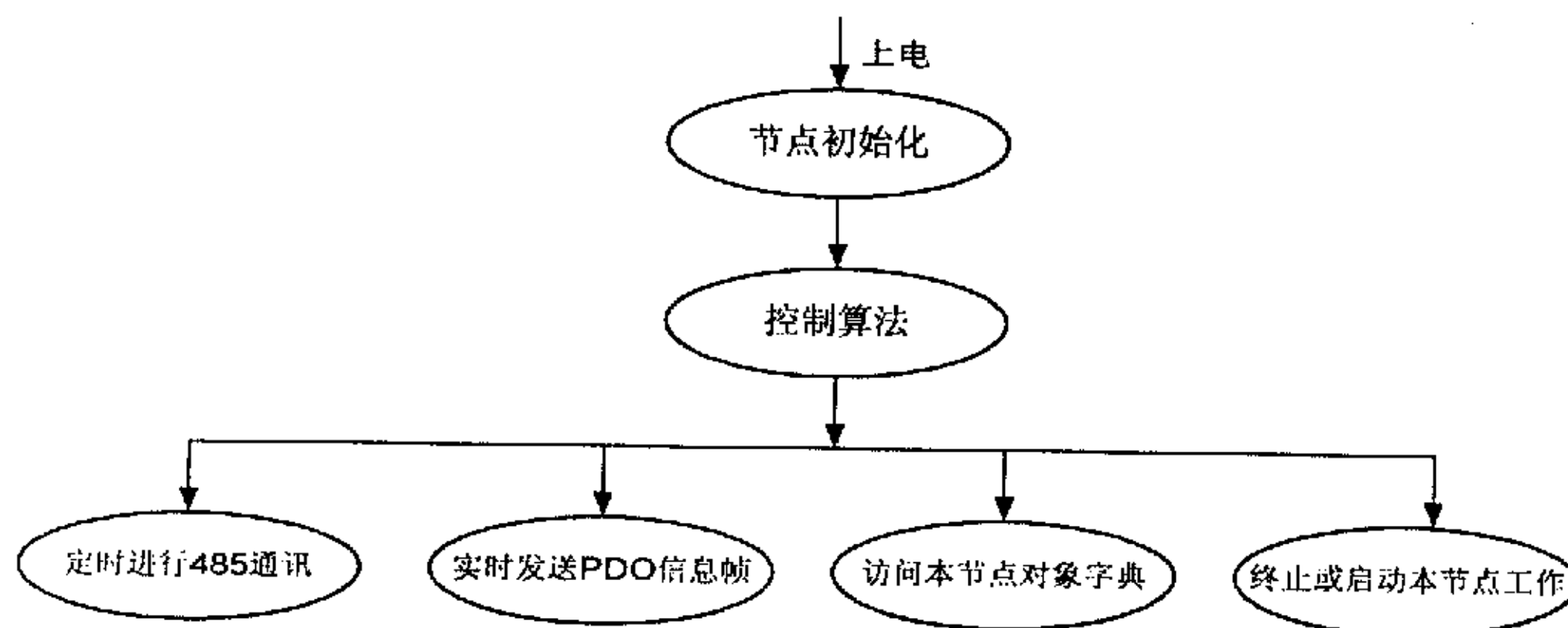


图5-3 从节点的主程序结构图

如图5-3所示，组建对象字典的程序文件作为一个头文件在主程序里进行调用，主程序主要完成对本节点的初始化和根据具体应用采取的相应控制任务，启动节点保护，定时调用485通讯模块，并根据控制任务要求实时的进行PDO通讯。

其它的功能都需要在收到主节点或其它从节点发来的信息帧以后调用相应的子程序才能完成，从节点收到信息帧产生中断，微处理器调用中断服务程序，接收此信息帧，进行分析，并根据信息帧的具体类型调用相应的子程序。

中断服务程序的结构如图5-4所示:

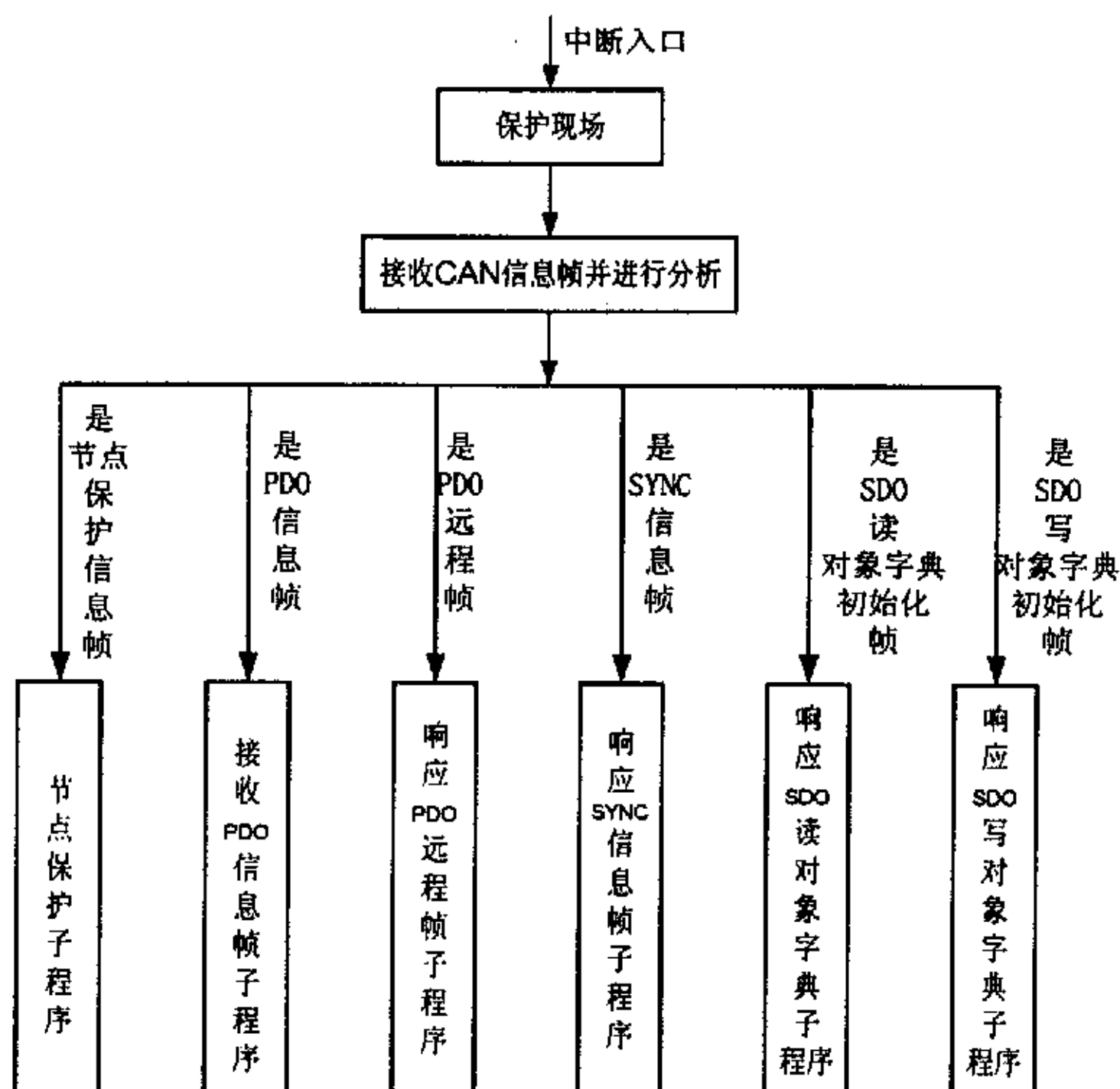


图 5-4 从节点 CAN 信息帧中断服务程序结构

本软件结构的主要框架是上述主程序和中断服务程序,大部分功能以子程序的形式实现,在主程序和中断服务中进行调用。下面主要介绍主程序和子程序中各个功能模块的构成和实现。

5.3.2 各个功能模块构成

1. 初始化功能模块:

- (1) Init_Canopen(): 设置本节点地址、CAN通讯传输速率和对象字典、PDO、SDO通讯相关参数的默认值。
- (2) Init_Hardware(): 设置本节点微处理器寄存器的相关参数。
- (3) Init_Timer(): 初始化微处理器中的定时器。
- (4) Init_Seriell(): 初始化微处理器中的串行口。

2. 访问本节点对象字典模块:

- (1) Get_Obj(): 由对象字典中得到具体参数对象的索引、子索引地址和此参数对象的大小。
- (2) Set_Obj(): 在没有SDO通讯的情况下,将对象字典中的某一个值根据具体应用从默认值改为实际需要的值。

3. PDO信息帧的封装与发送:

- (1) PDO_Mapping(): 按映射规则通过PDO映射封装需要传送的数据。

(2) `Process_Send_PDO()`: 按定义的PDO通讯模式进行数据的传输, 三种模式对应三个不同的模块。

4. PDO远程帧发送模块:

(1) `Process_RemoteRecv_PDO()`: 产生并发送一个CAN-RTR远程申请帧从其它节点申请一个本节点所需的数据。

5. CAN信息帧接收处理:

(1) `Process_Recv_Queue()`: 当SJA1000接收到CAN信息帧后通过中断通知单片机, 单片机进行接收, 对其进行分析并调用子程序作相应的处理。

6. SYNC信息帧处理模块:

(1) `Process_SYNC()`: 当节点收到SYNC信息帧时, PDO同步通讯模式发送PDO信息帧。

7. SDO通讯:

(1) `Process_SDO_Reci()`: 当节点收到的是主节点SDO初始化信息帧时, 根据要求对本节点进行读写操作。

8. 节点保护:

(1) `Process_LifeGuard()`: 当节点收到的是主节点SDO初始化信息帧时, 响应接收到的主节点的节点保护信息帧, 发送本节点的状态帧。

9. 启动终止CANopen节点:

(1) `Boot_Network()`: 在没有主节点参与的情况下, 设置、重新初始化或部分初始化本节点。

(2) `Stop_Network()`: 在没有主节点参与的情况下, 设置时本节点处于停止状态。

5.3.3 各个功能模块的实现

这一节介绍几个主要功能模块的实现及程序流程图:

一、初始化模块:

系统的初始化包括两种情况: 一种当网络刚上电时从节点在主节点的监控下进行初始化, 另一种是根据相应的命令由其他状态进入初始化的某个子状态, 对系统进行完全或部分初始化。从节点初始化的总体流程如图5-5所示。在整个初始化的过程中, 对SJA1000的初始化具有它的特殊性, 其初始化的流程图如图5-6所示:

二、数据的发送与接收:

无论是进行哪一种模式的通讯, 发送一帧数据的过程都是一样的, 只是由于数据的类型、实时性和在实际应用中的作用不同, 而在发送时顺序和优先级的不同从而形成不同的通讯模式。各种数据帧软件实现时发送一帧数据的子程序是相

同的,在进行不同模式的通讯时,只是对发送一帧数据的子程序中的变量赋予不同的值,按照不同的顺序和方式调用而已。

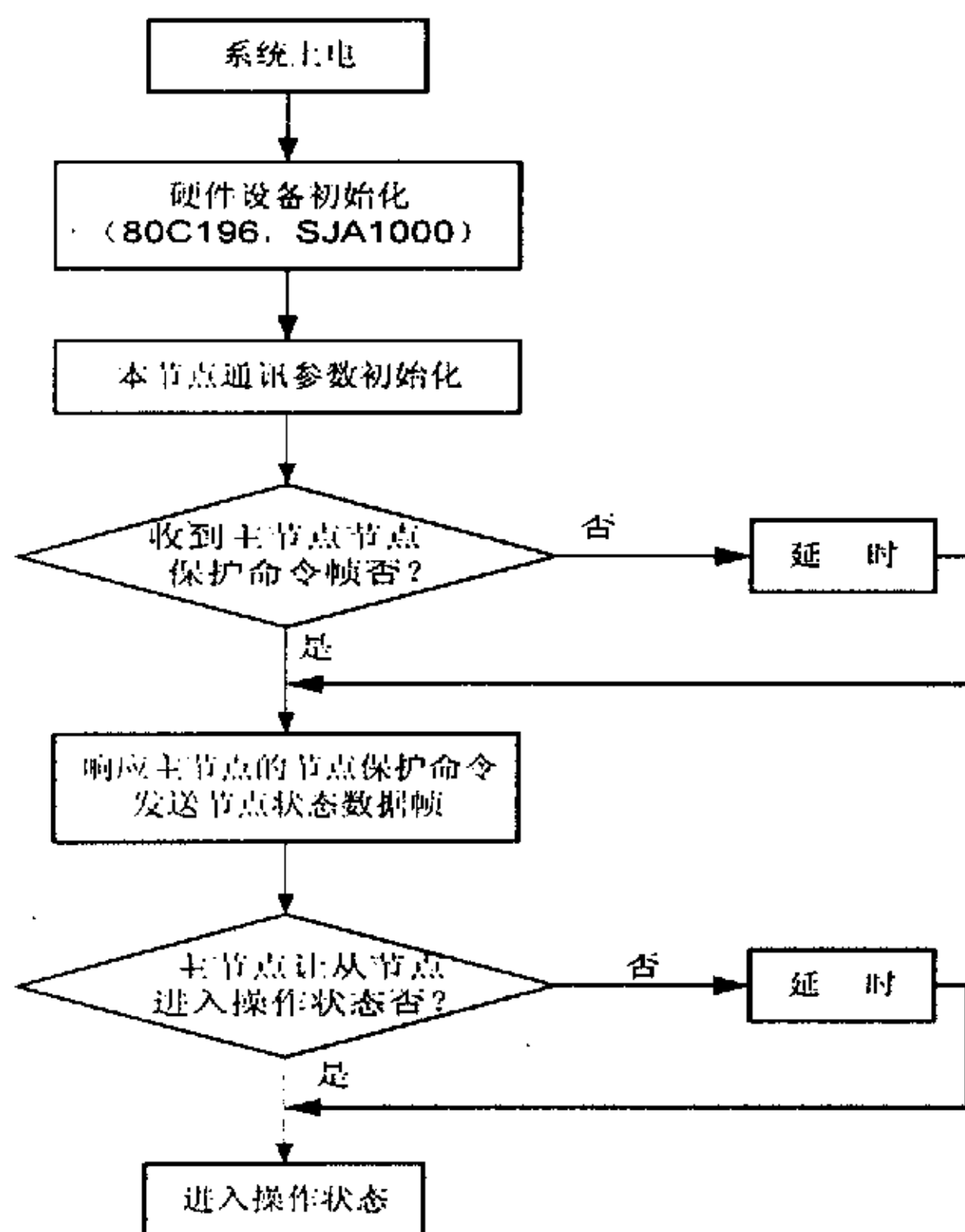


图5-5 从节点初始化总体流程图

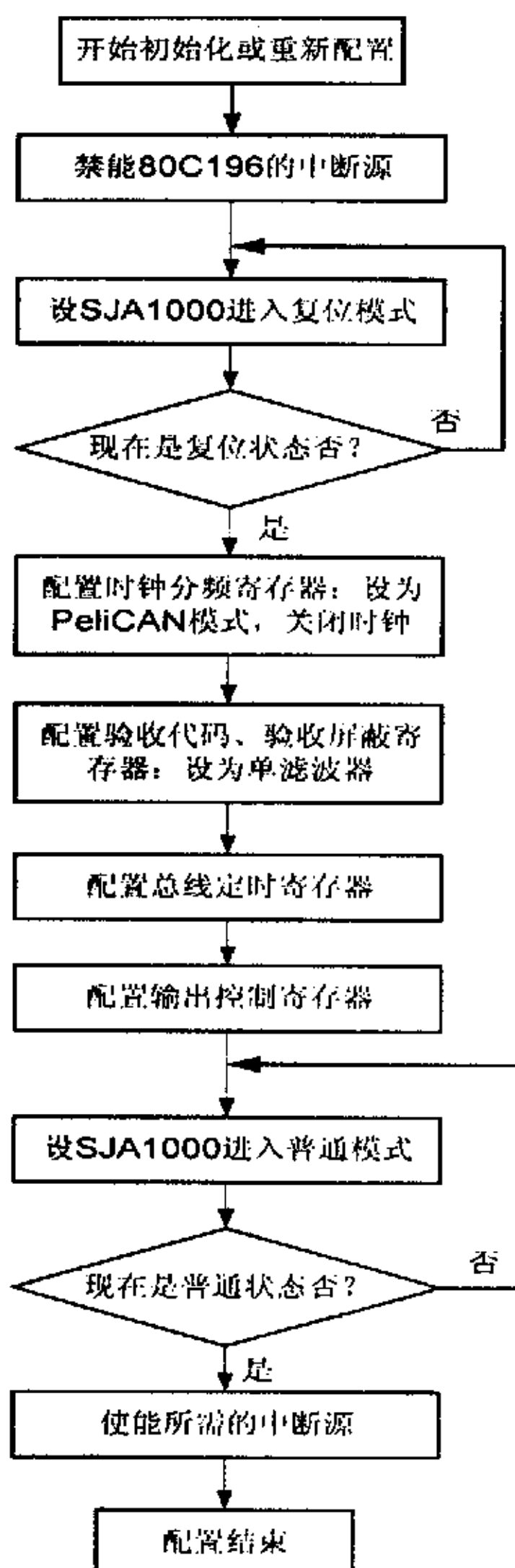


图5-6 SJA1000的初始化流程图

同样,相应的网络管理也是数据的发送与接收,只是数据帧的格式和发送的顺序不同,接收到数据以后所进行的处理也不同而已。

这一部分的软件设计的核心程序就是一帧数据的发送与接收,不同的通讯模式和网络管理依据上一章所述通讯协议所规定的帧的格式封装数据,按协议所规定的帧的发送顺序进行发送和接收处理。

CAN信息帧的发送程序流程图如图5-7所示:

当连续发送多帧数据时将定时器设为要发送数据帧的个数,当发送一帧数据时,将其设为“1”。

接收的过程完全由SJA1000硬件完成,当接收完一帧数据时, SJA1000用中断的方式通知单片机,单片机将数据由SJA1000读入单片机内部的接收数据存储区,并对其进行分析,根据数据不同的类型调用相应的子程序进行进一步的处理。

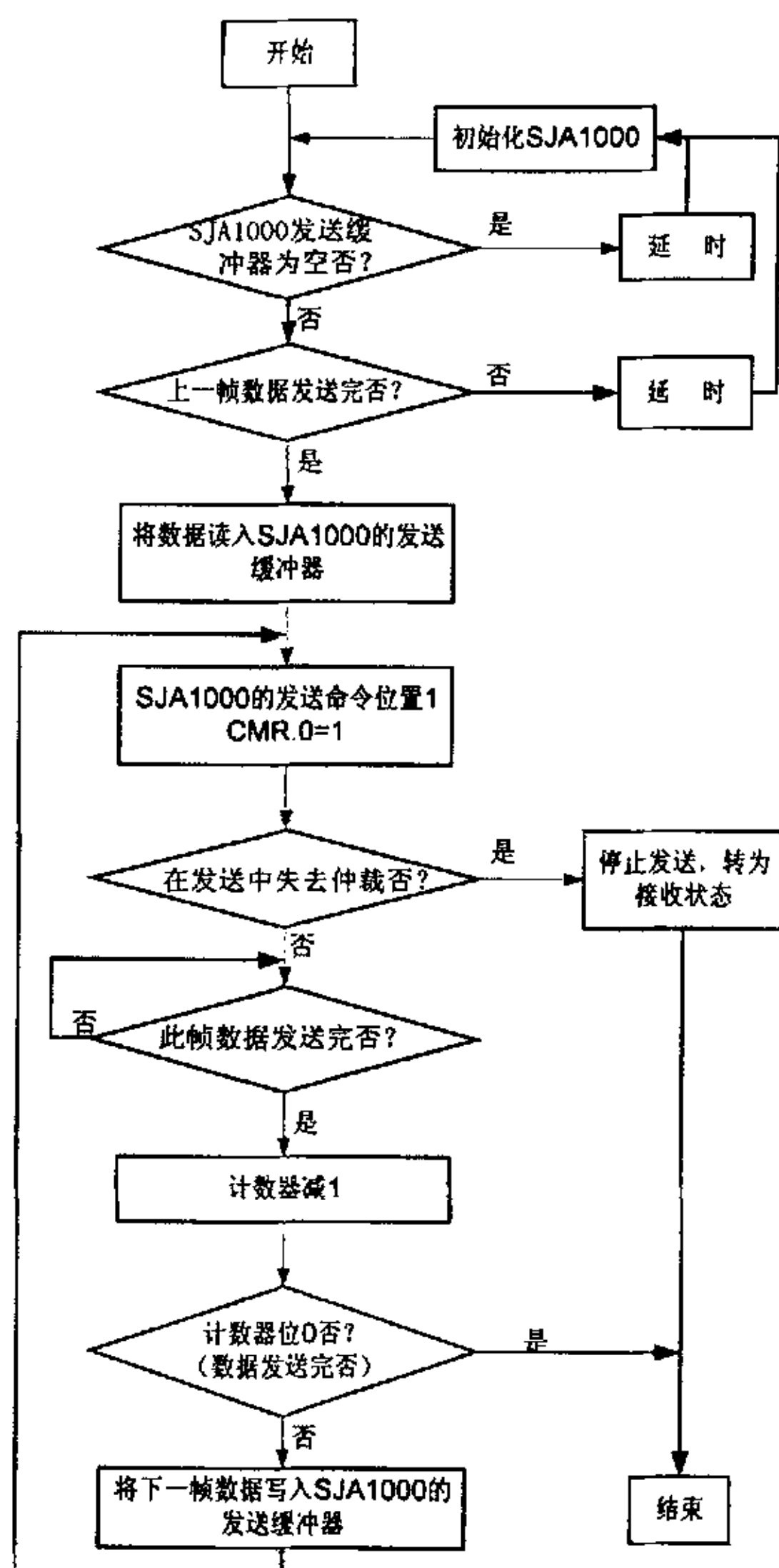


图5-7 数据帧发送流程图

5.4 节点控制模块485通讯的软件设计

5.4.1 概述

该软件主要完成节点控制模块与I/O模块之间的数据通讯和对I/O模块的管理功能。数据通讯采用全双工多机串行方式，CAN节点控制模块作为主机，各I/O模块作为从机。主机定期的从各个输入模块读入数据，同时也定期的往输出模块发送检测数据帧，以确定各个输出模块是否正常工作。若主机在此发送或接收的过程中发现错误，则命令从节点复位，并使程序重新运行。若接连三次发生错误，则中止程序，通知上位机进行相应的处理。

该程序定时功能有80C196的定时器1完成，每个20ms启动一次主机的发送或接收操作，定时的程序属于节点控制模块主程序的一部分，发送或接收程序作为子程序由定时中断调用。当节点控制模块主动要向I/O模块发送数据或输出模块

主动向节点控制模块发送数据,由节点控制模块的主程序直接调用发送接收子程序。当I/O模块在特殊需要时主动向节点控制模块发送数据时,由设置中断通知节点控制模块,由节点控制模块调用发送接收子程序进行数据的接收。

5.4.2 多机通讯协议^{[15][18]}

在本设计中,主机和多个作为从机的P89C52单片机通信过程有如下约定:

** 从机输入模块的地址设定为奇数,输出模块的设定为偶数。

** 发送和接收均不设奇偶校验。

** 如果发送失败(返回的命令和地址不对),发送0xff给从机,从机重新进行初始化,主机重新进行整个发送过程。如果连续三次发送失败,则中止发送过程,并告知上位机,相应地址的I/O模块发生错误,要求进行处理。

** 通讯中的命令:

0x01: 主机发送,从机准备接收;

0x02: 主机接收,从机准备发送;从机发送时先发送数据的长度;主机回应0x03后再发送数据;

0xff: 从机复位,复位后给主机一个就绪信号;

0x03: 从机可以发送数据;

从机准备就绪向主机发送“0x11”;

没有准备就绪向主机发送“0x00”。

其它均为非法命令。

** 首先使主机处于模式2,所有从机的SM2位置1,主机发送地址,从机处于只接收地址帧的监听状态。

** 主机向从机发送一帧地址信息,其中包括8位地址,可编程的第9位为“1”(TB8=1),表示发送的是地址,这样可以中断所有从机。

** 从机接收到地址后,判断主机发来的地址信息是否与本从机地址相符。若为本机地址,则清除SM2,进入正式通信状态,并把本机的地址发送回主机作为应答信号。然后主机发送数据帧,从机接收主机发送过来的数据或命令信息。其他从机由于地址不符,它们的SM2=1保持不变,无法与主机通信,从中断返回;

** 数据的格式为:由11位构成,1位起始位(0),8位数据位(地位在先),1位可编程位和1位停止位;

** 通信的各机之间必须以相同的帧(字符)格式及波特率进行通信;

** 发送缓冲区的地址是:50H--69H;

** 接收缓冲区的地址是:70H--89H。

在本设计中,节点控制模块和各个I/O模块还配接RS-485的标准接口(所

使用的是 MAX491 适配器)。所以在编程的过程中, 还考虑到了信号先传到 MAX491 的过程。

5.4.3 程序结构框图

在节点控制模块的主程序中设定定时器1为20ms中断一次，每中断一次在中断服务程序中调用一次485通讯的发送接收子程序。在中断服务程序中包括了由从机地址的奇偶判断是输入还是输出的功能。假设节点控制模块下带有4个从机，两个输入模块，两个输出模块。输入模块的地址设定为偶数，输出模块的设定为奇数，并在初始化时设定地址变量为0，每调用一次子程序地址变量加1，在中断服务程序通过判断地址变量的奇偶设置发送具体命令。

程序中还规定，发送数据放置在片内 RAM 区内，首地址为 50H 单元，第一个数据为发送数据块的长度；接收数据存放在片内 RAM 区内，首址为 60H 单元，接收第一个数据为接收数据块的长度。

这里还需特别说明的是：由于本设计，RS-485 采用的是 MAX491 电平转换器，所以在每次发送和接收数据时，都需要对 MAX491 的片选端事先进行设置。具体过程如下：

发送数据时, DE (P2.6) = 1, $\overline{\text{RE}}$ 任意。

接收数据时, $\overline{\text{DE}}(\text{P2.6})=0$, $\overline{\text{RE}}(\text{P2.5})=0$ 。

在程序中，主要变量的含义为：

ioslave: I/O模块的地址; iocomm: 主机命令;

中断服务程序为: timer1 io communi(void);

485通讯的发送接收子程序为: iocommuni(ioslave), 其中以I/O模块的地址变量ioslave为程序的返回值。

中断服务程序为: timer1_io_communi(void)如图5-8所示, 发送接收程序如图5-9所示。

5.5 I/O模块的程序设计

5.5.1 485 通讯程序设计

P89C52 的全双工串行通信接口具有多机通信功能。在多机通信中,为了保证主机与所选择的从机实现可靠的通信,必须保证通信接口具有识别功能,可以通过控制 P89C52 的串行口控制寄存器 SCON 中的 SM2 位来实现多机通信的功能,其控制原理简述如下:

利用P89C52串行口方式3及串行口控制寄存器SCON中的SM2和RB8的配合可完成主从式多机通信。串行口以方式3接收时，若SM2为1，则仅当从机接收到

的第九位数据（在RB8中）为1时，数据才装入接收缓冲器SBUF，并置RI=1，向CPU申请中断；如果接收到的第九位数据为0，则不置位中断标志RI，信息将丢失。而SM2为0时，则接收到一个数据字节后，不管第九位数据是“1”还是“0”都产生中断标志RI，接收到的数据装入SBUF。应用这个特点便可实现多机通讯。

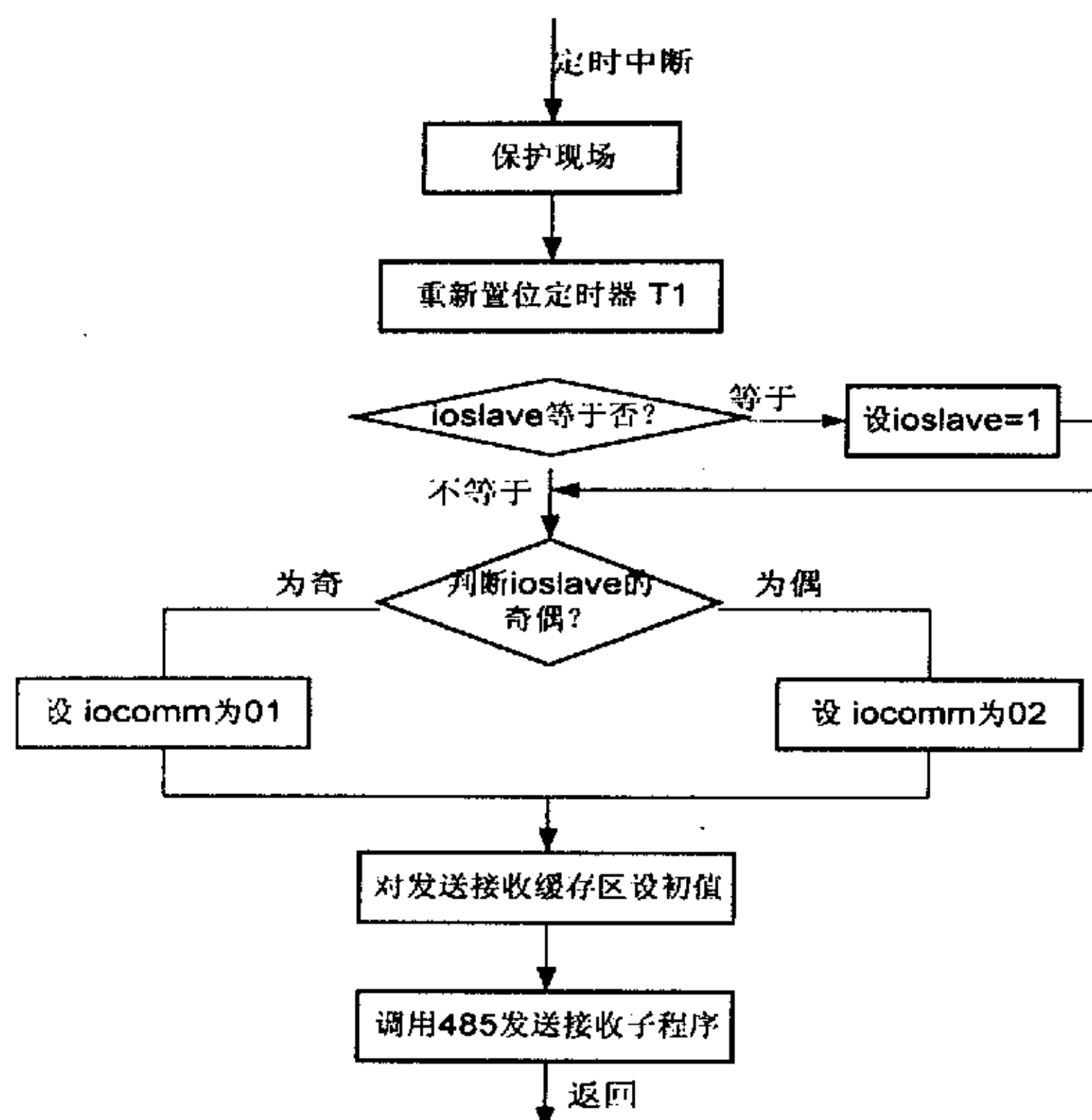


图 5-8 节点控制模块定时中断服务程序框图

各个 I/O 模块作为多机通讯的从机共同遵守上述多机通信协议。在本设计中，P89C52 采用的中断方式进行通信，其流程图如图 5-10 所示。在串行通信启动后采用查询方式接收或发送数据块。初始化程序安排在主程序中，中断服务程序选用工作寄存器区 1。本程序中用标志位 PSW. 1 为发送准备就绪标志，PSW. 5 作为接收准备就绪标志，由主程序置位。

程序中还规定，发送数据放置在片内 RAM 区内，首地址为 50H 单元，第一个数据为发送数据块的长度；接收数据存放在片内 RAM 区内，首址为 60H 单元，接收第一个数据为接收数据块的长度。

这里还需特别说明的是：由于本设计，RS-485 采用的是 MAX491 电平转换器，所以在每次发送和接收数据时，都需要对 MAX491 的片选端事先进行设置。具体过程如下：

发送数据时，DE (P2.6) = 1， $\overline{\text{RE}}$ 任意。

接收数据时，DE (P2.6) = 0， $\overline{\text{RE}}$ (P2.5) = 0。

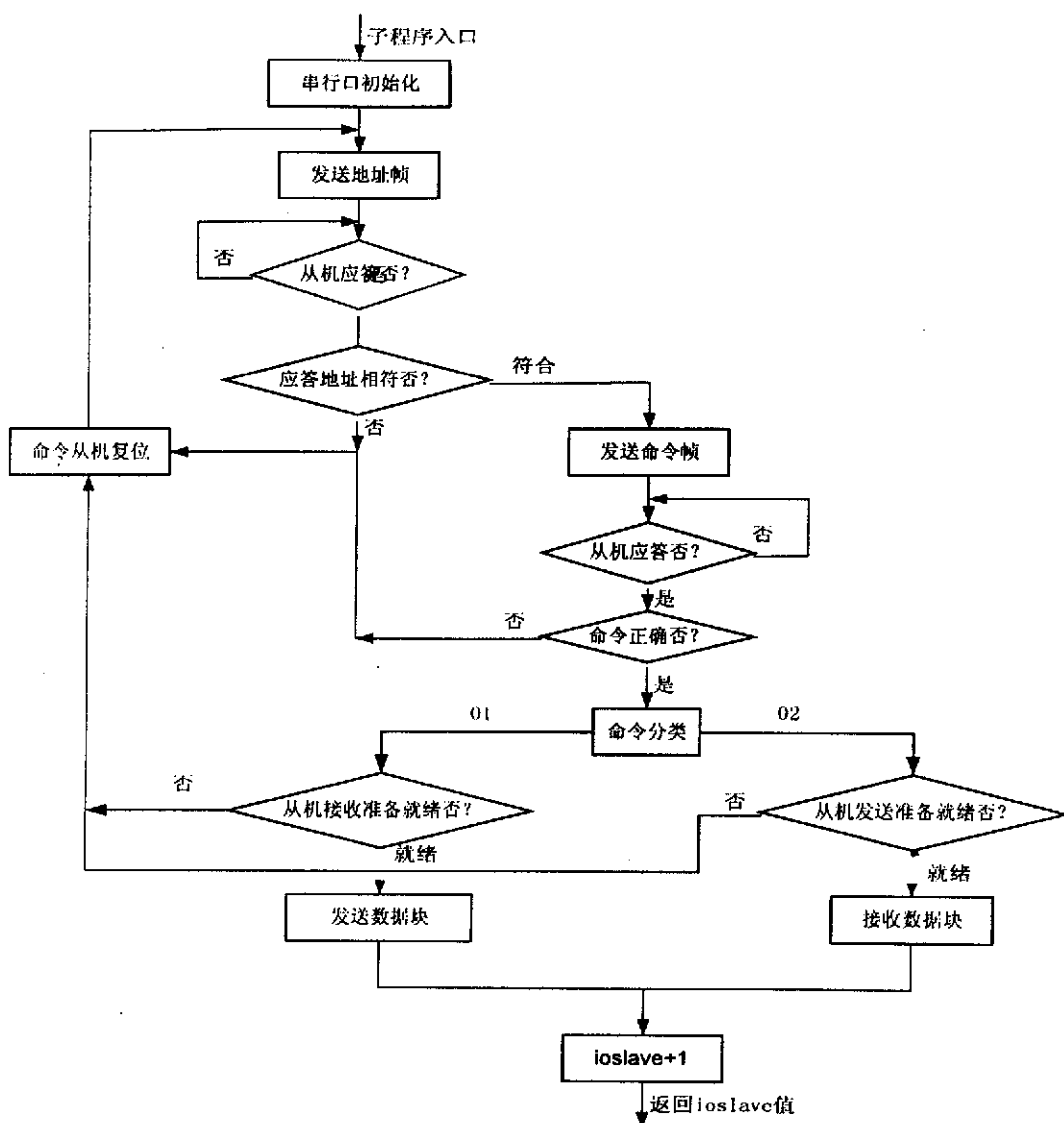


图 5-9 节点控制模块的 485 发送接收程序框图

5.5.2 输入模块程序设计

四个模块中数据输入包括DI模块的数字量输入和AI模块的模拟量输入。

一、数字量的输入：

数字量的输入处理相对简单。数字量是具有两种状态的量，数字输入的功能就是把数字量信号输入计算机中，以检测开关量的状态并进行报警。数字量的报警检测，只要判别一下当前值与系统所设定的报警值是否一样即可，如果一样即设定报警位。

二、模拟量的输入：

模拟量的输入相对复杂。通过微处理器的控制对象进行控制首先把控制回路的模拟信号变为数字信号，即进行A/D转换；然后送入计算机，为了防止干扰和提高精度，还要用软件进行一系列的处理，如数字滤波，标度转换等；最后结

果由通过485通讯传给节点控制模块参与控制算法，实施控制。

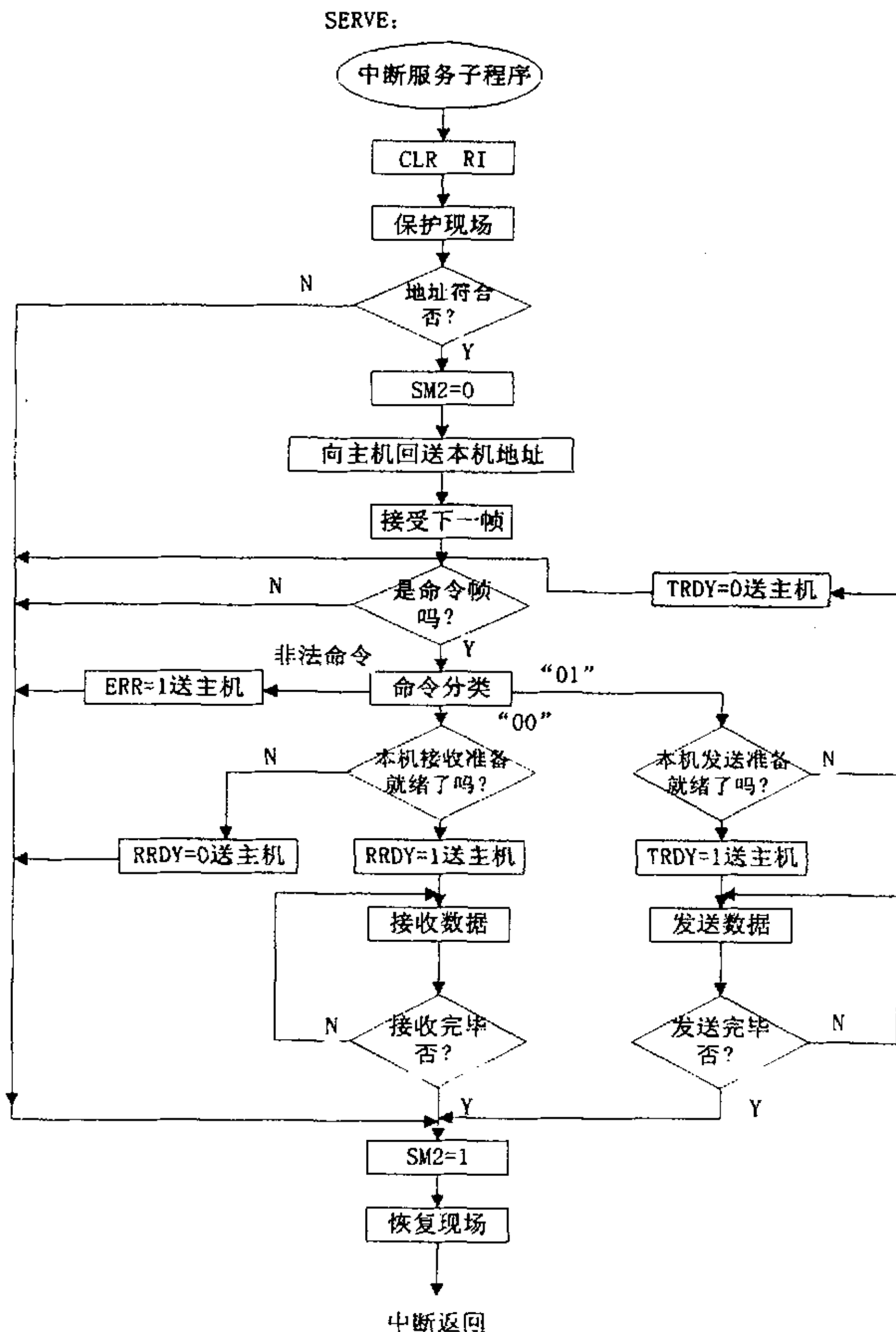


图 5-10 I/O 模块的 485 通讯流程图

本模块的A/D转换采用12位并行A/D转换器MAX1294，由MAX1294期间自身完成A/D转换并分两个字节分别传入单片机，在由程序将其变成单片机可以识别的变量。

经过A/D转换的数据由于各种因素的影响可能并不是所需的有效数据，所以需要一系列的软件处理使其变成有效的数据。

(1) 尖峰信号的抑制:

在连续性模拟信号的输入中，虽然过程参数一般不可能瞬间突变，但是附近

有产生强烈电感电压的干扰,则可能在信号传输线上产生尖峰干扰。尖峰信号一般持续的时间很短,如果恰好出现在采样时刻,就会产生较大的失真。所以应有抑制尖峰信号的措施。

在本设计中采用的方法是:在每一时刻,同时保留上一周期的值和一个允许信号变化的范围。如果本周期的A/D转换结果超出了上一周期的允许变化范围,则取消本次输入的A/D转换结果,延时1毫秒,重新采集该通道的输入值,再进行比较。如果几个毫秒后其值恢复正常,则取该正常范围内的值为有效输入。否则,几个重复采集输入都不正常,则说明传感器或输入通道上出现了故障,应当报警。

(2) 数字滤波:

由于各种电气干扰和设备不稳,信号输入计算机时,不可避免会引入一些微小的波动,有时电源的波动也会在信号上产生一个周期性的波动。虽然在硬件设计上已经采取一定的滤波措施,在软件设计上仍然需要软件滤波部分,以便进一步得到较好的信号值。

加权平均滤波:加权平均是在运行平均(即采用本周期的实际采样值和前几个周期的平均值进行算术平均得到的值)的基础上增加的近期输入所产生的影响,从而可以提高反应速度的一种滤波方法。公式如下:

$$y_k = \frac{1}{2}x_k + \frac{1}{4}y_{k-1} + \frac{1}{8}y_{k-2} + \frac{1}{16}y_{k-3} + \frac{1}{16}y_{k-4}$$

(3) 标度变换:

A/D转换输入的是电压信号,接下来的工作是根据各I/O通道对应的信号,将电压信号转换成信号的工程单位对应的物理量。根据不同的信号类型,本设计中采用了线性变换、流量信号变换的转换方式。

5.5.3 输出模块程序设计

输出模块包括DO数字量输出和AO模拟量输出。由于数字量输出较为简单,本文只介绍模拟量输出。

模拟量的输出多为线性模块。本设计的控制输出信号为4-20mA电流信号或0-10V电压信号,在控制输出以前也需要经过标度反变换和D/A转换器才能送到控制通道上。本设计中采用精度12位的4路D/A转换器MAX527,采用并行的输出方式。

因为经过控制算法得出的结果并不是最后的有效数据,需要应用标度反变换转换成与输出变量的变化范围相一致的数据。对应于12位的D/A转换器和0-10V的输出电压范围,一个二进制码对应 $10/4096=0.002441$ (V)的电压值。所以应先求出控制结果对应的二进制编码,再送到D/A通道。

第六章 CAN控制器的抗干扰设计和实时性研究

6.1 CAN控制器的抗干扰设计^[49]

CAN控制器属于单片机系统，常用于工业控制和各种远距离测控系统，其抗干扰设计也是系统设计的重要内容之一，本设计在硬件设计和软件设计中都采取了一定的措施。

6.1.1 硬件抗干扰设计

1. 电路抗干扰设计：

在本设计中，主要通过如下几点进行硬件抗干扰：

** 在 CAN 通讯中，外部信号进入 CAN 控制器时，在 82C250 与 SJA1000 之间加入两个高速光耦进行隔离。在数字输入输出模块中也加入光耦对输入输出信号进行了隔离。

** 在 CAN 通讯和 485 通讯时，都加入 120 欧终端匹配阻抗以消除长线干扰。

** 在输入输出电路中，加入 RC 电路进行滤波。

** 在 5V 电源输入每一个芯片之前，都先经过 1uf 的电容滤波以减少干扰。

** 在 RS-485 通信中，采用平衡差分电路，可以降低甚至消除反射。

** 加入外接看门狗电路，并配合软件措施以确保 CPU 的正常工作。

2. 印刷电路板的抗干扰设计^[51]：

(1) 印刷电路板的布局：

印刷电路板在设计过程中，从尺寸的选择、元器件的布置、电源线的布置，以及去耦电容的配置等各方面均充分考虑系统可靠性的要求。

** 印刷电路板大小要适中：过大使电路板上线条长，阻抗增加，导致抗干扰能力下降，成本增加；过小则散热效果差，安装调试不方便，元器件之间、临近线条之间距离过小，易受干扰；

** 在器件布置方面：把相互有关的器件就近放置，如时钟发生器、晶振和 CPU 的时钟输入端都易产生噪声，互相靠近放置；发热量大的器件考虑了散热问题，并合理放置。同时注意了将输入通道和输出通道分开；

** 对双面布线的电路板使两面线条垂直交叉，以减少磁场耦合效应。

(2) 布线方面：

** 输入、输出线分开；输入阻抗高时，电路输出端引线和临近引线分开；大电流的输出线和临近引线分开，以减小它们之间的耦合；

** 电路之间的连线尽量短，避免为了美观而将走线一律整齐排列；

** 布线时首先安排好电源线和地线，两者都应尽量加粗，尽量安排双面板的上下两面上，避免平行；

** 线与线相接或拐弯处要尽量使其成为钝角；

** 电源线尽量缩短，远离输入输出和晶振的时钟输出信号。

(3) 接地方法：

** 本文设计的五块电路板上都带有 CPU，都采用大面积地线敷铜，尽量减少干扰；

** 电路中的数字地和模拟地尽量分开，但最后采用一点接地。

6.1.2 软件抗干扰设计

1、指令冗余技术

当 CPU 受到干扰后，往往将一些操作数当作指令码来执行，引起程序混乱。当程序弹飞到某一单字节指令上时，便自动纳入正轨。当弹飞到某一双字节指令上时，有可能落到其操作数上，从而继续出错。但程序弹飞到三字节指令上时，因它有两个操作数，继续出错的机会就更大。因此，在本设计的软件设计中，多采用单字节指令，并在关键的地方人为地插入一些单字节指令（NOP）或将有效单字节指令重复书写，这便是指令冗余。指令冗余无疑降低系统的效率，但在绝大多数情况下，CPU 还不至于忙到不能执行几条指令的程度，故采用此种方法。走线一律整齐排列。

在某些对系统工作状态重要的指令前也可插入两条 NOP 指令，以保证正确执行。指令冗余技术可以减少程序弹飞的次数，使其很快纳入程序轨道。

2、软件陷阱技术

指令冗余使弹飞的程序安定下来是有条件的。首先，弹飞的程序必须落到程序区，其次，必须执行到冗余指令。而在本设计中，采用的另一种软件抗干扰措施就是软件陷阱。所谓软件陷阱就是一条引导指令，强行将捕获的程序引向一个指定的地址，在那里有一段专门对程序出错进行处理的程序。本设计中，将这段程序的入口标号记为 ERR。软件陷阱是一条无条件转移指令，为了加强其捕捉效果，一般还在它前面加两条 NOP 指令，因此软件陷阱由三条指令构成：

NOP

NOP

JMP ERR

在本设计中，将软件陷阱安排在了以下几个地方：

(1) 未使用的中断向量区。

(2) 未使用的大片 ROM 空间。

(3) 程序区。

由于软件陷阱都安排在正常程序执行不到的地方，所以不影响程序执行效率。

3、程序系统任务监视：

实时程序系统中，每个任务都有其产生时刻和最大可能生存期。在正常情况下，任务的执行时间不会超过该生存期。但是由于某种原因，程序“死掉”的现象也是有的。

为了防止系统死锁，如果一个任务生成后，其生存期超过最大可能生存期，则认为该任务状态异常，系统将强迫其停止运行，恢复系统正常运行。

任务监视的具体方法为：设置任务 A 的最大生存期为 $\text{Max } \Delta T_a$ ，当生成任务 A 时，记下当时时刻 T_b ，时钟程序每次执行，检查任务 A，若对当前时刻 T 存在 $T - T_b > \text{Max } \Delta T_a$ 时，则认为 A 超时，将其强行退出。由于时钟中断具有很高的可靠性，且执行频率为每秒 18.2 次，因此，它能有效地对任务进行监视，保证系统正常运行。

6.2 CAN 总线控制系统的实时性研究

6.2.1 概述

一、实时系统的概念^{[9][10]}：

支持实时应用，具有满足时间限制能力的系统被称为实时系统。所谓实时应用是指在应用中不强调公平性，必须保证重要的任务先执行，必要时可以放弃不重要的任务。分布实时系统是由一组分布的计算设备组成，每个计算设备执行一系列的任务，且具有实时能力的系统。现场总线就是典型的实时工业通讯网络之一。

在实时系统中时间、可靠性是影响系统性能的两大关键因素。首先，时间是实时系统中最重要资源，任务必须合理分配与调度，以保证任务的截止期要求。任务的正确性不但依赖于结果在逻辑与数值上的正确性，还依赖于结果产生的时间。其次，高度可靠是实时系统的重要特征，可靠性决定了实时系统能否正确支持和保证实时任务的完成。

二、实时应用中任务的特征与类型^{[9][43]}：

实时应用通常由一系列相互作用的任务组成，其特征是任务具有时间要求，并且以截止期的形式出现。根据任务在截止期能否完成对系统性能的影响，任务可分为三类：

**** 严格实时任务：**如果任务不能在截止期内完成将会产生巨大的危害。

**** 硬实时任务：**如果任务在截止期后完成，其完成的价值和产生的危害性都为 0。

**** 软实时任务：**对任务的截止期没有严格的要求，在截止期之后完成的任务仍然有一定价值。

在上述 CANopen 网络的通讯类型中，周期任务如 PDO 同步通讯和 I/O 模块周期性的上传数据，其特征是具有时间关键性，属于严格实时任务；非周期任务若是时间关键性的，一般属于硬实时任务。比如，PDO 通讯中的时间触发方式用于对某一状态值的监控和报警就属于此种情况。非周期任务最苛刻的运行情况是，所有的非周期任务在瞬间同时触发，这种情况对时间的要求比较严格，需要在任务调度上特别处理。PDO 的查询模式和节点保护信息对实行也有一定的要求，属于硬实时任务；SDO 通讯对时间的要求都不严格，属于软实时任务。应急对象和网络管理帧实时性要求也比较高，但它们优先级在所有数据帧也是最高的，因此一般满足实时性要求。

三、分布实时系统中的通信延迟^[9]：

在分布式实时系统中，不同计算设备之间的任务交互是通过通信网络，以信息传递的方式实现的，为了满足任务的实时要求，要求任务之间的信息传递必须在一定的通信延迟时间内完成。从信息传送到信息接收之间的全部通信延迟称作端对端的通信延迟。它主要包括下面四方面的因素：

- ** 产生延迟：**从发送端的任务产生信息到把此信息放入排队队列所需的时间。
- ** 排队延迟：**从信息进入排队队列到此信息获取现场通信设备所需的时间。
- ** 传输延迟：**信息在现场通信设备间传输所需的时间。
- ** 发送延迟：**接收端的任务处理信息，到被该信息发送给目的任务所需的时间。

在 CANopen 网络中，产生延迟、传输延迟和发送延迟都是可预测的，在讨论实时问题时不是主要的因素；信息的排队延迟主要受本节点内的信息竞争和不同节点间的信息竞争产生延迟的影响。其中，节点内信息的延迟由排队策略决定，节点间的信息延迟主要由通讯网络的 MAC 协议影响。

四、现场总线的实时性要求^{[40][42][45]}：

现场总线控制系统是典型的实时工业通讯网络，直接支持底层信息通讯，实时是最根本最重要的要求。因此，现场总线作为连接现场测量的控制设备，不但完成非实时信息通信，而且支持实时控制信息通信的一类特殊通信协议。具体表现为：

- ** 短信息帧的高效处理；**
- ** 周期与非周期信息的混合处理；**
- ** 有限的响应时间。**

要满足上述现场设备对通信系统的要求,客观上要求现场总线网络协议能够处理上述特点带来的新问题,具备以下能力:

- ** 实时响应外部环境,包括任务变化、网络节点的增/减、网络失效诊断等;
- ** 具有确定的时态特性,主要指在时间上、任务上具有行为可预测性;
- ** 简单的网络通信方式,可以有效地减小通信处理延迟。

五、解决网络延迟的几种方法^{[34][37][39][42]}:

对于实时网络控制系统中的延迟,目前学者从三个方面加以解决:

** 通过改善控制系统中的控制算法和控制策略解决。由于控制系统网络的引入带来具有不确定性的延时,使控制系统的性能变差,甚至原本稳定的系统变的不稳定,该方法不考虑网络本身,只是针对已经存在不确定性延时的控制系统从控制算法和策略上加以改进,使控制系统具有较好的性能指标,如利用随机控制,预测算法,预估补偿等。

** 通过改进网络协议的 MAC 层协议进行解决。由于现场总线的通讯延迟中主要的排队延迟主要由通讯协议中的 MAC 子层决定,MAC 子层的特征直接与现场总线的实时能力相关。因此部分学者对不同类型的 MAC 层协议进行深入的分析,加以改进,以提高通讯协议的实时能力。比如,对于事件触发方式缺乏全局控制的现象,提出了确定性冲突解决和一致的发送时间等方法;在分布式令牌和虚拟令牌中,通过控制通信节点的运行周期来保证信息的响应时间等。

** 通过改进通讯协议中的应用层协议部分进行解决。由于在实际应用中很多情况 MAC 层协议已固定在总线控制器的硬件中,无法改变,部分学者在应用中将应用层协议部分内容加以改进,以提高系统的实时性。

本文从 CAN 总线应用层协议的角度对 CAN 总线控制系统的实时性问题做初步的研究。

6.2.2 CAN 总线控制系统的实时性研究^{[36][38][42][43]}

本文应用结构分析的方法研究 CANopen 应用层协议的实时性。

一、结构分析方法的概述^[35]:

结构分析是一种传统的软件建模技术,主要是引入数据流图标(DFDs)的概念来描述系统中的信息的传输和处理过程,其对信息的描述如图 6-1 所示:

DFDs 将输入节点的信息称作外部信息,经过节点的数据传输和内部处理,一部分改变形式输出本节点,依然称作外部信息,另一部分存储在本节点内部称作数据存储。DFDs 还可以进一步的细分,以描述系统局部的数据流,描述系统的细节部分,从而描述整个系统的数据关系。DFDs 只代表数据信息流,并不能描述系统中控制的流向。

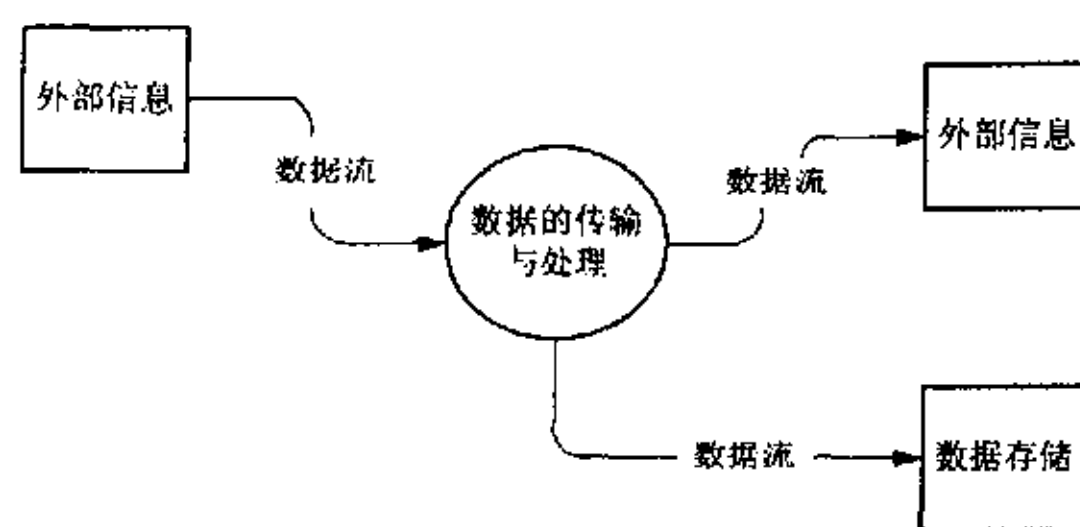


图 6-1: 结构分析方法的数据模型

所谓实时性分析，首先系统应用中的每个数据流，并将其映射成为 DFDs 形式，然后利用 DFDs 计算每个任务的响应时间。已知任务具体的响应时间，就可以利用 CANopen 中的禁止时间服务调节网络中各个信息传输的时间，进行任务调度，使任务中需要的数据在实时性要求范围的截止期内到达，满足系统的实时性要求。因此，控制系统实时性的分析首先要计算一个通讯任务的响应时间。

首先将 CANopen 的信息映射为 DFDs:

**** 数据流:** 用于描述系统中的实时性数据，即 PDO 通讯传输的数据。当系统定义数据长度时，同时定义 DFDs 中的数据类型和用户类型:

若数据流来源于应用中的处理程序，则为只写型数据，客户端是指数据的源节点，服务器是指数据流的目的节点。

否则，数据为只读型，客户端是指数据的目的节点，服务器是指数据流的源节点。

**** 事件流:** 用于描述系统中的事件性数据，分为三种类型: 使能/禁能型数据; 触发型数据和信号型数据。在 CANopen 协议中主节点的激活或禁止从节点的信息属于使能/禁能型数据; SYNC 信息帧、节点保护的启动帧、SDO 的起始帧等属于触发型数据; 当某一节点发生错误发送的应急对象帧属于信号型数据。

**** 数据存储:** 应用数据存储可以将源节点的数据变为全局数据，本节点和其他节点都可以通过数据流的传输进行访问和使用。在 CANopen 协议中，那些数据定义为这种形式需依据具体应用而定。

**** 事件存储:** 因为 CANopen 协议不支持一个节点既为客户端又为服务器端，因此当某一事件流需要在本节点进行处理时，将事件流设为事件存储。

CANopen 应用层协议的实时性分析以 CAN 协议的实时性分析为基础，因此先介绍 CAN 协议的通讯任务响应时间的计算。

二、CAN 协议的实时性分析:

CAN 协议数据信息传输的响应时间的计算方法如下^[35]:

首先作如下定义:

R_m = 信息 m 的传输响应的最长时间

C_m = 信息 m 的物理传输时间

S_m = 信息 m 的数据字节数

τ = 位传输时间长度 = 1/波特率

$$C_m = \left(\left\lceil \frac{34 + 8 * S_m}{5} \right\rceil + 47 + 8 * S_m \right) \tau \quad \text{公式 1}$$

若要计算地优先级信息的响应时间，需作如下限制：所有信息的传输都有一个时间域；偶发型信息的传输具有一个最小释放时间，而信息 m 的禁止时间必须小于最小释放时间，否则发送缓冲器的数据将被下一个数据覆盖。定义：

T_m = 信息 m 的最小释放时间

则信息 m 的排队延迟分为以下两部分：

** 当前本节点的发送缓冲器的信息发送完毕的时间

$$B_m = \max_{\forall k \in M} (C_k)$$

** 所有其它节点高优先级的信息发送完毕的时间

$$LQ_m = \sum_{\forall j \in hp(m)} \left\lceil \frac{Q_m + J_j + \tau}{T_j} \right\rceil * C_j$$

则信息 m 的排队延迟为：

$$Q_m = B_m + LQ_m$$

$$\text{解方程得： } Q_m^{n+1} = B_m + LQ_m^n \quad Q_m^0 = 0$$

由此得出信息 m 传输响应的最长时间为：

$$R_m = C_m + Q_m^{n+1} \quad \text{其中： } Q_m^{n+1} - Q_m^n = 0 \quad \text{公式 2}$$

当节点中有远程帧传输时，网络中会出现具有两个相同标识符的数据传输的情况，具有一定的特殊性，因此网络中的远程帧的传输，响应时间计算如下：

定义：

C_m = 信息 m 的物理传输时间（无 RTR 申请帧）

$C_{RTR(m)}$ = 信息 m 的物理传输时间（有 RTR 申请帧）

$$\text{则： } LQ_m = \sum_{\forall j \in hp(m)} \left\lceil \frac{Q_m + J_j + \tau}{T_j} \right\rceil * (C_j + C_{RTR(m)})$$

信息 m 的传输最长响应时间为：

$$R_m = C_m + C_{RTR(m)} + Q_m^{n+1} \quad \text{其中： } Q_m^{n+1} - Q_m^n = 0 \quad \text{公式 3}$$

三、CANopen 应用层协议的实时性分析：

在 CAN 协议实时性分析的基础上，对 CANopen 应用层协议进行实时性分析。如上一节所述，CANopen 应用层协议只需对严格实时任务和硬实时任务进行实时性分析，即其中的 PDO 通讯和节点保护信息帧。

1、节点保护的实时性分析：

因为节点保护需要主节点发送起始帧,从节点响应,因此要同时考虑到网络传输延迟和信息处理的延迟。对于主节点,因为本文中设计的主节点专用于网络管理,而没有信息处理功能,因此没有信息处理延迟。又因为节点保护信息帧是除启动/禁止从节点信息帧以外优先级最高的,排队延迟也很小,因此传输延迟占主要。对于从节点还要同时考虑到其响应主节点和封装状态帧的处理延迟时间。对于一个 CANopen 网络,节点保护响应时间与网络中需要进行节点保护的节点数成反比,若节点数越多,则响应的时间就越长。

在本设计中的硬件环境下,用前面的公式估算得:

主节点的节点保护信息帧的发送时间在 3~4us 之间,而从节点的响应信息帧传送时间在 4~6us 之间,从节点响应主节点和封装状态帧的处理延迟时间一般在 18~92us 之间。则若网络中只有一个节点需要进行节点保护,整个过程最长的响应时间为 102us,随着网络中进行节点保护的节点数的增加,主节点的任务加重,其保护周期也需要响应的增加,再加上一定的预量,一般保护周期几百毫秒。

2、单线程运行 CANopen 协议的实时性分析:

对于 CAN 控制器中单片机单线程运行的控制系统进行实时性分析,首先假定单片机中运行如下无限循环的程序:

```
for(;;) {
    CheckNodeState();
    /* Start user code section*/
    /* End user code section*/
    CMS_Main();
}
```

CMS_Main()主要功能为:(1)对要输出的数据信息进行排队传送;(2)接收处理输入的数据信息,并初始化与之相关的功能。

当系统运行以上程序时,可将 CANopen 对接收数据的处理过程进一步细化:

假定 CMS_Main()一次只处理一帧接收到的数据并释放其所占的接收缓冲器,则节点 n 的接收处理闭环的延迟时间,即对接收的 COB 处理最长时间定义如下:

CLDn = 接收处理闭环的延迟时间

CNSn = 检测节点状态的运行时间

CMS_MainAn = 对接收缓冲器中的数据进行接收并释放其所占接收缓冲器的时间

CMS_MainBn = 对接收到的数据进行识别并初始化相应的处理程序的时间

UCn = 用户程序代码运行的时间

则 $CLD_n = CNS_n + CMS_MainAn + CMS_MainBn + UC_n$

另外定义:

$REQ_{COB(x)}$ = 调用请求处理通讯对象 (COBx) 功能的程序运行时间

$ReqTX_{COB(x)}$ = 调用对通讯对象 (COBx) 进行排队和发送程序的时间 (由控制系统中的一个实时过程产生)

$OL_{COB(x)}$ = COBx 的发送请求和发送重叠的时间

$RxQueue(n)$ = 节点 n 中接收缓冲器的数据设定的个数

H_y = 发送确认信息帧 COB(y) 的时间

则节点 n 对接收到的一帧类型不确定的数据进行处理所需的最长时间为:

$$REQ_{COB(x)} + ReqTX_{COB(x)} - OL_{COB(x)} + \sum_{y \in RxQueue(n)} (CLD_n + H_y) \quad \text{公式 4}$$

那么对于一项确定的服务, 比如同步 PDO 通讯模式的接收过程, 查询模式等, 需要附加定义:

$RES_{COB(x)}$ = 响应 COBx 的程序运行时间

$ResTX_{COB(x)}$ = 响应 COBx 进行排队和发送的时间

则响应时间是:

$$REQ_{COB(x)} + ReqTX_{COB(x)} - OL_{COB(x)} + \sum_{y \in RxQueue(n)} (CLD_n + H_y) + ResTX_{COB(x)} - OL_{COB(x)} + \sum_{y \in RxQueue(n)} (CLD_n + H_y) \quad \text{公式 5}$$

根据前人做的一些工作, 结合本系统的硬件环境, 对于只读型数据流的处理的平均时间大约是 325~385us, 对只写型数据流处理的平均时间大约是 626~733us, 所有数据流总体的平均时间为 655us。

根据以上公式, 可以对一个具体系统的具体通讯过程的最大响应时间进行估算, 以此为依据分配具体通讯的禁止时间, 合理调度控制系统中的通讯任务, 以满足系统的实时性要求。

6.3 论文总结及实验验证

本论文设计开发了 CAN 总线控制系统中的底层节点 (即 CAN 总线控制器), 包括 CAN 节点控制模块和独立的 I/O 模块两部分, 实现了 CAN 总线控制系统中基本的通讯功能。为构建完整的基于 CANopen 高层的 CAN 总线控制系统作了初步的工作, 本系统已经初具规模, 但还需要在今后进一步完善, 现将所做的工作总结如下:

一、硬件设计与开发:

本文设计开发并调试了 CAN 节点控制模块、数字输入模块 (DI)、数字输出模块 (DO), 其硬件的外观图如图 6-2 所示, 模拟输入模块 (AI) 和模拟输出

模块(AO)进行了原理图 PCB 图的设计并制作了电路板。对整个 CAN 节点的验证由 DI 和 DO 模块来完成。

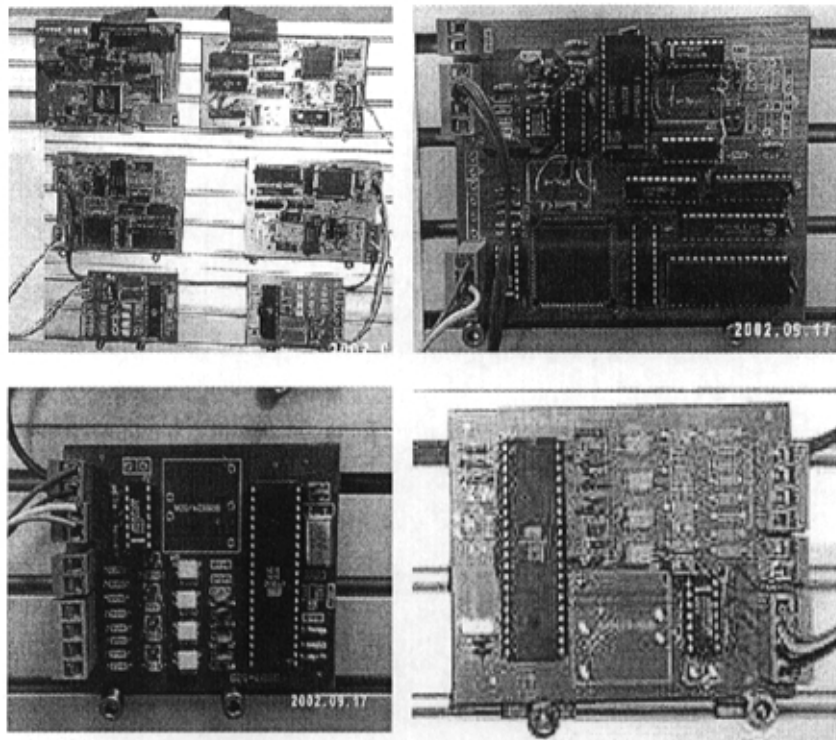


图 6-2 系统构建图和各模块的外观图

二：CAN 节点控制模块与 I/O 模块 485 通讯的实验验证：

本文采用模块化设计，将 CAN 节点分为节点控制模块和 I/O 模块两部分，通过 485 通讯实现两者之间的数据通讯，与传统的集中式 CAN 节点相比，其灵活性和通用性大大增强，但同时实时性和可靠性必然会受到影响。本文通过两个实验对这部分进行验证。

实验一，验证硬件的有效性：

令 CAN 节点控制模块带一个 DI 模块和一个 DO 模块，由 DI 模块输入一个开关量，CAN 节点控制模块进行接收，并将其发送给 DO 模块，由 DO 模块接收，由模块上的信号灯指示接收到的值。经验证，当由 DI 模块输入“1”时，DO 模块的信号灯亮，即输出值为 1；由 DI 模块输入“0”时，DO 模块的信号灯不亮，即输出值为 0；本实验证明硬件有效。

实验二，验证硬件的实时性和可靠性：

令 CAN 节点控制模块带两个 DI 模块和两个 DO 模块，DI 模块的地址设为 1 和 3，DO 模块的地址设为 2 和 4，采用全双工的 422 通讯，传输波特率设为

9600bps, I/O 模块晶振的频率为 11.0592MHz。运行第五章中所编写的程序, 由节点控制模块 100ms 启动一次发送或接收数据帧的过程, 发送的数据为一个字节, 数值为 55H; 接收 DI 模块中一个固定数据 A0H。通过示波器检测发送和接收数据的时间。发送一个字节 9 位数据需 0.93ms, 经验证, 整个发送程序的启动确认需要 5 个字节的发送或接收, 因此发送一个字节数据所需的时间为 4.7ms, 接收一个字节数据所需的时间为 5.1ms。实验测得的发送时间为 4.94ms, 接收时间为 5.22ms。实验中令节点控制模块接收 100 帧数据, 没有发现丢帧和错帧的现象。本文中选用的 AD 和 DA 转换器都是微秒级的, 不会带来太多的延迟。而一般控制系统对 I/O 数据的实时性要求为秒级即可, 因此引入二级总线结构虽然带来了一定的延迟, 但基本满足实时性和可靠性的要求, 日后进行进一步改善会得到更好的特性。

三、CAN 协议数据传输的实验验证:

实验的硬件平台为一个 CAN 总线 PC 控制卡、两个 CAN 总线控制器、一台 Duron750 PC 机。设两个 CAN 总线控制器的网络 ID 分别为 5 和 7。由于 CAN 总线传输的距离很近, 可以达到 1Mbps 的传输率。设 CAN 总线的传输速率选择为 500kbps。两个 CAN 总线控制器连续向 PC 控制卡发出数据, PC 控制卡收到数据后上传到 PC 机中并显示。同时在控制卡中将收到的数据按位取反 (用于替代数据处理部分), 再回发到源控制器中。在传输率为 500kbps 的情况下发送 1000 帧未出现丢帧和错帧现象。在传输率为 1Mbps 的情况下发送 1000 帧, 出现一帧的数据错误。可靠性满足要求。

四、CAN 协议数据传输的实验验证:

由于 CANopen 网络中必须有一个主节点, 发送同步信号, 启动网络管理和 SDO 通讯, 而本系统中没有在软件上对主节点进行 CANopen 协议的实现, 所以很大一部分软件功能无法进行测试, 只能对 PDO 通讯的事件触发模式和查询模式进行测试。PDO 通讯的事件触发模式的验证如下: 设定当节点 5 的 DI 模块的状态值发生变化时, 触发一个 PDO 通讯, 将此值发送给上位机, 经验证, 状态值发生 500 次变化引起的传输没有发现与变化值不同的情况。查询模式的验证如下: 由节点 5 向节点 7 发出远程申请一帧数据, 在节点 7 设定状态值为 AAH, 经过查询模式 PDO 通讯节点 5 接收的数据为 AAH, 又改变数据重复以上实验, 结果均正确。

通过以上的实验验证, 本系统硬件设计和软件设计均可满足一般工业控制系统的实时性和可靠性的需求, 若经过进一步的完善, 一定会得到更好的性能。

结 束 语

现场总线技术已成为工业数据总线领域的新热点,为工业控制领域带来了一次重大的变革,受到了越来越广泛的关注。CAN 总线是现场总线中最有前途的总线之一;应用范围很广,具有很好的发展前景。

在论文研究阶段笔者以 CAN 总线在以下几个方面进行了探讨:

1、针对目前多数 CAN 总线控制器存在的缺陷,在硬件设计上进行了改进,引入二级总线的概念,采用模块化的硬件结构,使 CAN 总线控制器可以根据不同的应用对象灵活的改变结构,节省了开发的时间和成本,通用性和实时性增强。

2、对 CANopen 高层协议做了分析和研究,以此为标准并结合工业控制应用中的普遍需求编写了适合大多数控制系统的 CAN 总线应用层协议。

3、在软件上实现了从节点基本的通讯和网络管理的功能,对主节点的功能和软件实现做了初步的设想。

4、对 CAN 总线以及现场总线中的实时性问题做了初步的探讨,给出了以 CANopen 应用层协议为基础的 CAN 通讯的最大延迟时间的计算方法。

在完成以上工作以后,笔者认为接下来的研究应该集中在:

1、由于二级总线结构的引入,在使系统灵活性通用性增强的同时,也会使系统数据的输入输出的实时性变差,如何解决这个问题在本文只是做了初步的工作,今后还可在硬件设计做进一步的改进,在软件设计上进一步考虑主控模块对 I/O 模块的管理和 485 通讯协议的改进以改善数据的输入输出的实时性。

2、CANopen 高层协议的功能十分丰富,本文实现了从节点的基本的通讯和管理功能,今后还可在以下几个方面进行完善:

(1) 完善主节点的管理功能,使主节点可对网络进行动态设置,在有特殊需要时对信息的标识符进行动态分配;设置时间标记服务,使系统的同步具有更高的精度;当网络中的节点数目增加或减少时,主节点对其进行标识。

(2) 从节点中对节点本身的错误进行监测和判别并进行进一步的处理;在没有主节点的情况下对本节点进行初始化和状态设置。

(3) 将 CANopen 高层协议的软件实现进行产品化,使其可以经过适当组态后适用于不同的控制系统。

3、将主节点和从节点软件联合起来进行调试,发现问题并进行改进。

4、对 CANopen 高层协议做深入的研究与改进,以提高控制系统的实时性。

由于本人水平所限,论文中有疏忽和错误之处,望各位老师和同学能不吝指正赐教,将不胜感激。

参 考 文 献

- [1] 范铠, 现场总线的发展趋势, 自动化仪表, 2000.2: 1~4
- [2] 范铠, 现场总线技术在国内的应用情况, 自动化博览, 2001.9: 1~5
- [3] 夏德海, 现场总线的困惑、无奈与出路, 仪器仪表标准化与计量, 2001.5: 10~15
- [4] 沈德耀 金敏, 现场总线纵横谈, 基础自动化, 2000.7 (4): 1~5
- [5] 冯晓升, 中国现场总线标准体系的形成与实施, 世界仪表与自动化, 2001.005 (012): 68~71
- [6] 程正群 许宝祥, 现场总线与 DCS 控制系统, 化工自动化及仪表, 1999.26 (3): 1~3
- [7] 阳宪惠, 现场总线技术及应用, 清华大学出版社, 1996.1
- [8] 邬明宽, CAN 总线原理和应用系统设计, 北京航空航天大学出版社, 1996.11
- [9] 王智 王天然, 工业实时通讯网络(现场总线)的基础理论研究现状(上), 信息与控制, 2002.4: 146~152 163
- [10] 王智 王天然, 工业实时通讯网络(现场总线)的基础理论研究现状(下), 信息与控制, 2002.6: 241~249
- [11] 邬明宽, CAN 总线系统设计中的几个问题, 1998.9: 18~20
- [12] 周凤余 鲁守银 李贻斌, CAN 总线系统智能节点设计与实现, 网络与通讯, 1999.6 9~10 32
- [13] 肖海荣, 基于SJA1000的CAN总线智能节点设计, 计算机自动测量与控制, 2001, 9 (2): 48~50
- [14] 徐应跃 吴爱国, 现场总线 MAC 层的分析, 化工自动化及仪表, 2001.28(2): 4~7
- [15] 孙涵芳, Intel 16 位单片机, 北京航空航天大学出版社, 1995.11
- [16] 金磐石 王永明, Intel 96 系列单片微型机应用详解, 电子工业出版社, 1992.4
- [17] 程军, Intel 80C196 单片机应用与 C 语言开发, 北京航空航天大学出版社, 2000.11
- [18] 孙涵芳, 徐爱卿, MCS-51/96 系列单片机原理及应用, 北京航空航天大学出版社, 1996.2
- [19] Intel 80C196 单片机应用实践与 C 语言开发
- [20] 马忠梅, 籍顺心, 张凯, 马岩, 单片机的 C 语言应用程序设计, 北京航空航天大学出版社, 1999, 1
- [21] Kirk Zurell, 嵌入式系统的 C 程序设计, 北京: 机械工业出版社 2002
- [22] CiA DS-202 TO DS-207, CAN Application Layer Specification, 1996

- [23] M.Farsi, K.Ratcliff, Manuel Barbosa, An introduction to CANopen, Computer & Control Engineering Journal, August 1999. 161~168
- [24] CANopen Communication Profile for Industry System Based on CAL, CiA Draft Standard 301 October 1996.
- [25] Philips Semiconductor, SJA1000 Data Sheet, Philips Semiconductor, 2000
- [26] Mohammad Farsi, Karl Ratcliff, An introduction to CANopen and CANopen Communication Issues, 1997 The Institution of Electrical Engineers, Print and published by The IEE, Savoy Place, London WC2R 0BL, UK.
- [27] Karl Ratcliff, CANopen Implementation Issues, 1997 The Institution of Electrical Engineers, Print by The IEE, Savoy Place, London WC2R 0BL, UK.
- [28] Luis Vazquez, Leandro Rojas, Samuel Galceran, Simplified CANopen Application Layer Model for Educational Proposals, 2000 IEEE 255~259
- [29] Zdenek Hanzalek, Tomas Pacha, Use of the Fieldbus Systems in Academic Setting, 1999 IEEE, 93~97
- [30] M Karsi, K Ratecliff, CANopen: Configure and Device Testing, 373~380.
- [31] Manuel B.M.Parbosa, Adriano da Silva Carvalho, Mohammed Farsi, A CANopen IO Module: Simple and Efficient System Integration, 1998 IEEE, 155~159.
- [32] Prof. Dr. -Ing. Gerhard Gruhler, CANopen based Distributed Control System, 1998 The Institution of Electrical Engineers, Print and published by The IEE, Savoy Place, London WC2R 0BL, UK.
- [33] FARSI M. and BARBOSA M., CANopen implementation application to industrial networks, Research Studies Press Ltd. ISBN 0-86380-247-8 2000
- [34] D.Heffeman and A.Bohannon, Real-Time design in a distributed control network application layer environment, IEE Proc.Softw., Vol.148.No5.October 2001. 150~155
- [35] TINDELL,K: Calculating controller area (CAN) message response times, 1995,Control Eng.Practice,3, (8) pp.1163-1169.
- [36] Ian Broster and Alan Burns, Timely use of the CAN Protocol in Hard Real-time Systems with Faults, 2001 IEEE, 95~102.
- [37] Gianluca Cena and Adriano Valenzano, Integrating the CAN and MUST Access Techniques in a Single Fieldbus, 2001 IEEE, 231~239
- [38] Paulo Pedreiras and Luis Almeida, EDF message scheduling on controlker area network, Computing & Control Engineering August 2002. 163~170

- [39] Frank Robler and Birgit Geppert, Applying Quality of Service Architectures to the Field-bus Domain, 1997, IEEE. 39~48
- [40] DeMACRO,T., Structured analysis and system specification, Prentice Hall, 1979
- [41] TINDELL.K. and BURNS A., and WELLINGA A., Calculating controller area network(CAN) message response times, Distributed computer Control Systems, Spain September 1994
- [42] TINDELL.K. and BURNS A., Guaranteeing message latencies on control area network(CAN), 1st International CAN conference, Germany, Sept,1994
- [43] Ian Broster, Guillem Bernat and Alan Burns, Weakly Hard Real-time Constraints on Controller Area Network, Proceedings of the 14 th Euromicro Conference on Real-Time Systems 1068-3070/02, 2002 IEEE
- [44] G. Bernat. Weakly hard real-time systems. IEEE Transactions on Computers, 50(3):308-321, March 2001.
- [45] M. Gergeleit and H. Steich. Implementing a distributed high-resolution real-time clock using the can bus. In Proceedings of the 1st International CAN Conference. CiA,1994.
- [46] 钱归平 毛维杰 苏宏业, 工业控制中嵌入式系统软件的研究, 计算机应用研究 2002.7 , 21~23
- [47] 张云生, 实时控制系统软件设计原理及应用, 国防工业出版社, 1998, 12
- [48] 郝忠孝, 微型机实时软件设计, 机械工业出版社, 1989, 2
- [49] 李孔安, 现场总线网可靠性设计与分析, 基础自动化, 1996, No.2
- [50] 周岳斌, 盛艳, 于滢, PC 机与多单片机构成主从式通讯系统, 仪表技术, 1998, 4
- [51] Wayne Wolf, 嵌入式计算系统设计原理, 北京: 机械工业出版社 2002
- [52] 孙继业, 高速数字系统印刷电路板的设计要点, 电子工程师, 2001, 27(12): 51~54
- [53] 同向前 薛钧义, 嵌入式系统测控软件的结构化设计, 计算机工程, 2002, 28 43~44, 213
- [54] 薛雷, CAN 总线的动态优先权分配机制与非实时数据的传输, 计算机工程与应用, 1999, 12 , 33~35
- [55] RS-422 and RS-485 Application Note, B&B Electronics Mfg. Co. Inc.P.O. Box 1040 -- Ottawa, IL 61350PH (815) 433-5100 -- FAX (815) 434-7094

发表论文及参加科研情况说明

研究生期间发表论文一篇：

《CAN 总线控制系统的应用层协议 CANopen 的剖析》，微计算机信息。2003.4

参加的科研项目包括：

- 1、基于以太网的嵌入式智能小区控制器的开发
- 2、基于 CANopen 协议模块化 CAN 控制器的研究

致 谢

本文的研究工作是在吴爱国教授的悉心关怀和精心指导下完成的。无论是在学习还是在生活上,都给予作者很大的关心,尤其是导师严谨求实的科学作风、勤勤恳恳的工作态度、广博的知识以及对课题发展方向敏锐的洞察力,都给予作者以莫大的感染和鼓舞。在此,谨向我的导师致以最衷心的感谢和诚挚的敬意。

同时,本文的研究还得到了黄瑞祥副教授的悉心指导,在此表示衷心的感谢。在设计过程中,还得到了同实验室的同学刘剑峰、周鹏鹏和张辉等的大力支持,在我遇到困难时曾经给予我很大的帮助和鼓励,在此也向他们表示深深的感谢。

还要感谢我的父母和家人。在我求学期间,他们时刻关注我的学业,是我坚实的后盾,感谢他们对我的爱护和支持。

另外,还有我的同学和朋友们对我的毕业设计的关心和帮助。再次,我要对所有关心和帮助我的人们说一声:衷心的感谢你们!