

天津大学
硕士学位论文
CAN bus高层协议CANopen的研究以及在模块化CAN控制器上的实现
姓名：宋晓强
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
20040101

## 摘 要

作为通用有效可靠及经济的平台, CAN 总线已经广泛地受到了欢迎。它可以用于汽车系统、机械、技术设备和工业自动化里几乎任何类型的数据通信。

但是, CAN芯片只提供了开放系统互连参考模型(OSI)中的物理层和链路层功能, 一般用户必须直接用驱动程序操作链路层, 不能直接满足工业控制网络的组态和产品互连要求。为了以CAN芯片为基础构成完整的工业控制现场总线系统, 必须制定相应的应用层协议, 实现系统的组态、设备互连和兼容功能。为此目的, 在德国成立了“自动化CAN用户和制定商协会”(CiA, CAN in Automation), 开始着手制定自动化CAN的应用层协议CANopen。目前CANopen协议已经被提交欧洲标准委员会讨论, 作为一种新的工业现场总线标准。

本文详细剖析了CANopen应用层(CiA DS-301)。解释了CANopen的核心一对对象字典、各类通讯对象、标识符的分配, 以及一些设备子协议(Device Profiles)。

此外, 自主开发设计了具有较强通用性的模块化CAN总线控制器。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。

最后, 在这个模块化CAN总线控制器上实现了CANopen协议中从结点的通讯和网络管理功能。并且在CAN bus数据的实时调度上进一步做了改进探索。

关键词: 现场总线

CAN 总线

CAN 总线控制器

CANopen 应用层协议

## ABSTRACT

As a kind of general, reliable and economical platform, CAN bus has been used for a long time, which appears in automobile, mechanical engineering and many other application fields.

CAN bus chips, however, only provide with the function of CAN physical layer and CAN data link layer of OSI communication model. Common users have to deal with the data link layer with the result that Can chips cannot meet the requirement of configuration and product interlinkage in industrial control networks. The corresponding application layer protocol has to be made in order to constitute the complete industrial field bus systems based on CAN chips. With this purpose, CiA (CAN in Automation) established the higher layer protocol CANopen, which has been put forward to Europe Standard Committee as a kind of new industrial field bus standard.

This thesis analyzed the CANopen application layer in detail. It explained the core of CANopen--- object dictionary, all kind of communication objects, the allocation of COB-ID and some device profiles.

In addition, a kind of modularized CAN bus controller was designed. This controller took the form of two-level structure, i.e. the system bus and I/O bus was designed separately and the whole controller had two parts: CAN node control module and I/O module. The two parts communicated with each other through RS-485. The users can configure the I/O modules freely to constitute a complete CAN node. With this controller, the CAN bus can contain more I/O ports with less nodes, so the real time performance of the bus is guaranteed.

At the end of the thesis, the CANopen slave protocol was realized in this modularized controller. The function of CANopen network management and communication was implemented. And this thesis made some attempt to modify the CANopen application protocol to improve the real time performance of CAN bus.

**Keywords:** field bus, CAN bus, CAN bus controller, CANopen 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：宋晓强 签字日期：2004 年 1 月 5 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：宋晓强

导师签名：吴云同

签字日期：2004 年 1 月 5 日

签字日期：2004 年 1 月 5 日

## 第一章 绪 论

### 1.1 课题相关技术的现状和发展前景

#### 1.1.1 CAN 总线

##### 1、CAN 总线概述<sup>[1]</sup>:

CAN (Controller Area Network) 称为控制器局域网, 属于总线式通讯网络。它是 20 世纪 80 年代初德国 BOSCH 公司提出为解决汽车内部众多控制器与测量设备之间的数据交换而开发的一种串行数据通信总线, 是一种有效支持分布式控制系统或实时控制的串行通信网络。后来, 由于其成本低、可靠性高、抗干扰能力和实时性强等特点使其应用范围逐渐扩大。Philips Semiconductors 制定的 CAN 技术规范 (Version2.0) 和 ISO 颁布的国际标准 ISO11898 为 CAN 规范化、标准化和应用系统设计铺平了道路。同时, 各种基于 CAN 协议的高层协议的开发使得 CAN 总线的功能更强, 应用范围更广, 不仅在汽车工业、过程控制、数控机床和纺织机械等领域, 而且也在向医疗、电力、海运电子设备等方面发展。

##### 2、CAN 总线的特点及优势<sup>[2][3]</sup>:

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:

- ✧ 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;
- ✧ 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;
- ✧ 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;
- ✧ 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;
- ✧ 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);
- ✧ 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;
- ✧ 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;
- ✧ 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。

### 3、CAN 总线存在的不足:

- ✧ 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。
- ✧ 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。
- ✧ 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。
- ✧ 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。
- ✧ 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。

### 1.1.2 CANopen 高层协议

CAN 协议是一个非常简单的协议。它只定义了物理层和数据链路层, 本身并不完整, 有些复杂的应用问题需要一个更高层次的协议—应用层协议来实现。比如, CAN 数据帧一次最多只能传送 8 字节; CAN 只提供了非确认的数据传输服务等。然而, CAN 的技术特点允许各厂商在 CAN 协议的基础上自行开发自己的高层应用协议, 给用户提供了一个面向应用的清晰接口。目前, 许多厂商都根据自己的意图并结合其优势纷纷推出基于 CAN 的总线产品, 如 DeviceNet(设备网), CANopen, CANKingdom, SDS 等。它们都得到 CiA (CAN in Automation) 的支持, 符合 ISO11898 标准, 同时又各具特色。

#### 1. CANopen 高层协议简述<sup>[4][5]</sup>

CANopen 协议集定义了基于 CAN 的分布式工业自动化系统的应用标准以及 CAN 应用层通信标准。CANopen 在发布后不久就获得了广泛的承认。尤其是在欧洲, CANopen 被认为是在基于 CAN 的工业系统中占领导地位的标准。CANopen 协议集基于所谓的“通信子集”, 该子集规定了基本的通信机制及其特性。

大多数重要的设备类型, 例如数字和模拟的输入输出模块, 驱动设备, 操作设备, 控制器, 可编程控制器或编码器, 都在称为“设备子集”的协议中进行描述。设备子集定义了不同类型的标准设备及其相应的功能。依靠 CANopen 协议集的支持, 可以对不同厂商的设备通过总线进行配置。

CANopen 标准最核心的部分是通过对象字典 (Object Dictionary) 对设备功能进行描述。对象字典分为两部分, 第一部分包括基本的设备信息, 例如设备 ID, 制造商, 通信参数等等。第二部分描述了特殊的设备功能。

一个 16 位的索引和一个 8 位的子索引唯一确定了对象字典的入口。通过对象字典的入口可以对设备的“应用对象”进行基本网络访问, 设备的“应用对象”

可以是输入输出信号, 设备参数, 设备功能和网络变量等等。

CANopen 设备的功能及特性以电子数据单 (EDS) 的形式描述, EDS 采用 ASCII 格式, 可以将 EDS 理解成某种形式的表格。实际的设备设置通过所谓的设备配置文件 (DCF) 进行描述。EDS 和 DCF 都可以从 Internet 上下载, 并可以存储在设备之中。

象其他知名的现场总线系统一样, CANopen 也分为两种基本的数据传输机制: 通过过程数据对象 (PDO) 对小型的数据进行高速数据交换, 以及通过服务数据对象 (SDO) 对对象字典进行访问。后者主要用于在设备配置过程中传输参数以及传输大数据块。进程数据对象通常采用事件触发、循环或请求方式发送, 作为广播对象, 它的上层并没有附加协议。一个 PDO 最大可传输 8 字节数据。在连接一个同步报文的时候, 整个网络都可以采用同步方式对 PDO 进行传送和接收 (Synchronous PDOs)。通过存储在对象字典中的 PDO 映射 (PDO Mapping) 结构, 可以对应用对象到 PDO 对象的分配 (Transmission Object) 进行调整, 这可以保证设备可以对不同的应用需求进行调整。

通过传输 SDO 可以实现可靠的数据传输, 由两个 CAN 对象在两个网络节点间通过点对点的通信来实现这一过程。通过传输对象字典的索引以及子索引, 可以定位相应的对象字典入口。通过 SDO 传送报文可以不受长度的限制, 但传送 SDO 报文需要额外的协议开销。

标准化的事件触发的高优先级突发报文用于报告设备故障, 可以通过中心时钟报文来提供整个系统的系统时钟。准备和协调建立一个分布式的自动化系统所需的功能都适于 CAN 网络管理 (NMT) 定义的机制, 同样可应用于周期性的节点维护 (node guarding)。

可以选择通过 “Heartbeat Message” 来显示 CANopen 设备的通信能力。PDO 和 SDO 的 CAN 报文标识符可以直接通过对象字典的数据结构的入口标志符进行分配, 或者, 在简单的系统中, 也可用预定义的标志符。

## 2. 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: Northhampton 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

#### 1. 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。

#### 2. 选择 CANopen 作为 CAN 总线高层协议研究的原因

CAN 在我国的应用已经比较广泛, 从工业现场、过程控制到楼宇电子设备, 但是普遍停留在比较低层次的应用上, 即简单的应用 CAN 控制器、CAN 总线收发器完成 CANbus 的一些简单应用, 并且都是自己定义的一些简单的应用层协议, 没有和国际的标准 CAN 高层协议接轨。一方面这是因为应用的场合比较简单, 要求不高; 另一方面, CAN 总线高层协议 (如 CANopen) 国内这方面的研究比较少, 应用也比较少, 即使应用的话, 也是使用现成的国际大公司的内嵌 CANopen 协议的控制器产品, 而没有自主开发的产品, 所以需要加强我国 CAN 总线高层协议标准化的进程。

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:



- ✧ Devicenet 网络中最多可接 64 个节点，而 CANopen 最多可接 127 个节点。
- ✧ 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。
- ✧ 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。
- ✧ 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。

### 3. 课题的具体内容

选择 CAN 总线为研究对象，着重研究分析 CAN 高层协议 CANopen，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，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。

另外，在 CANopen 协议的基础上，着重再对网络数据的实时调度方面进行一些探索，以增强 CANopen 在工业现场的时效性，为 CANopen 的优化抛砖引玉。

### 1.2.2 课题的意义

通过本课题的研究，分析了以 CANopen 高层协议为基础的标准 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局面，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。而且最后对 CANopen 实时数据调度的优化做的探索也对 CANopen 协议的优化做出了一定的贡献。

### 1.3 课题的任务

- 1、分析 CANopen 高层协议，本文主要实现 CANopen 从结点的必要通讯。
- 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 89C52 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、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。

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。

5、以 CANopen 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。

6、探索 CANopen 实时数据调度的改进余地，增强 CANbus 的实时性，进一步增强工业控制网络的控制效果。

## 第二章 CANopen 应用层协议剖析

CANopen 是一个基于 CAN 串行总线的网络传输系统。CANopen 假设控制器的节点的硬件有 ISO11898 规定的一个 CAN 收发器和一个 CAN 控制器。

CANopen 协议集描述了标准的通讯机制和设备功能。这个协议集是 CAN in Automation(CiA)定义和维护的, CiA 是一个国际用户和生产商组织,但是值得一提的是,这个协议是公开的,没有所谓的版权的限制。

CANopen 规定了应用层和通讯子集 (CiA DS-301)、可编程控制器的设备子集 (CiA DSP-302)、电缆和接口特性等等。还规定了额外的专用设备描述协议,这些都在相应的设备子集里规定着 (CiA DS-4XX)。本文主要介绍 CiA DS-301,即工业上应用的基本 CANopen 通讯协议。

### 2.1 协议简介<sup>[4][5][6][7][8]</sup>

#### 2.1.1 CANopen 通讯参考模型

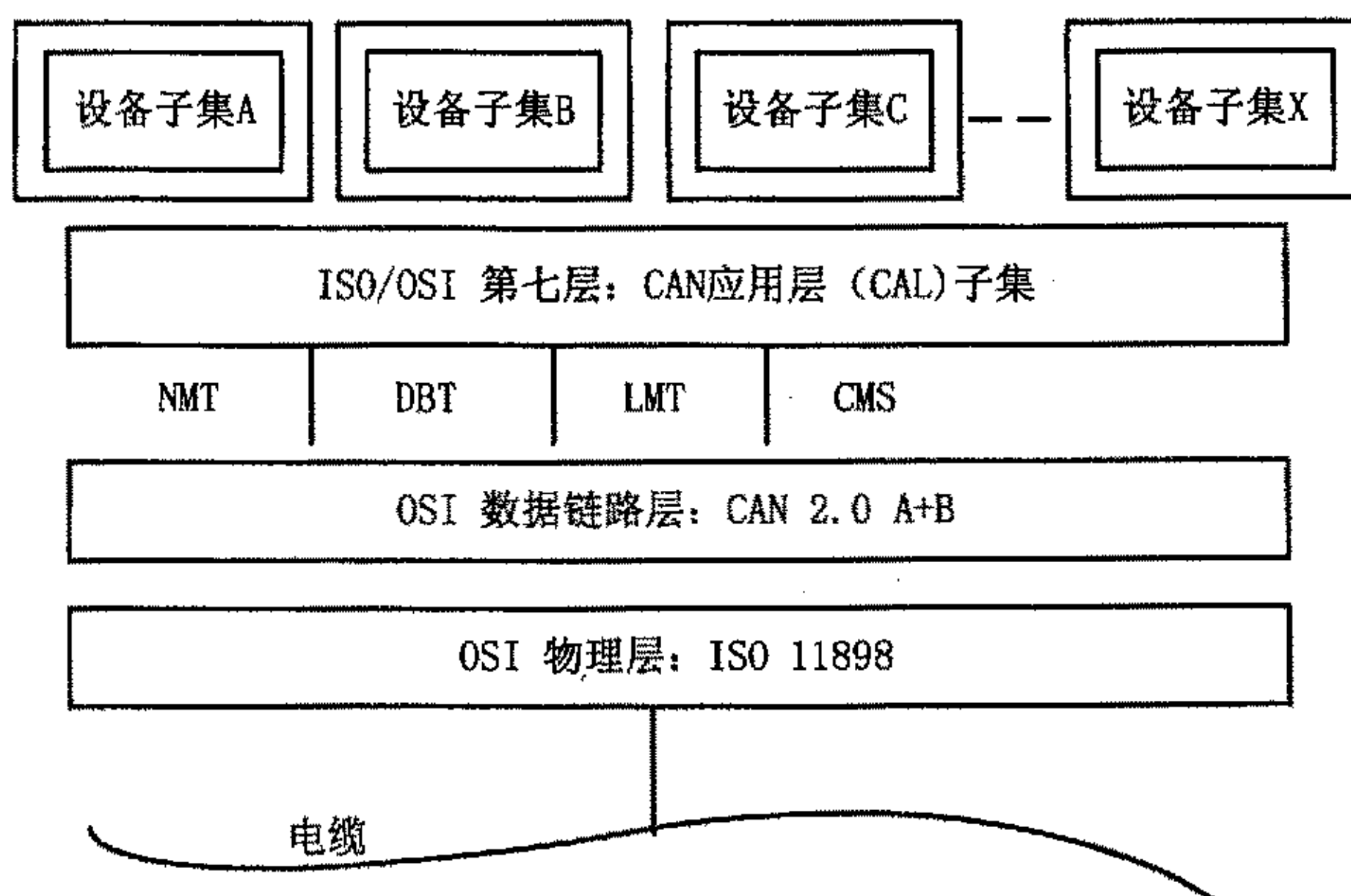


图2-1 CANopen 通讯参考模型

CANopen 通讯协议子集涵盖了配置和进行实时数据通讯的概念,以及设备节点之间进行同步的机制。基本上说,CANopen 通讯协议子集描述了设备节点之间是怎样在 CAN 应用层 (CAL) 进行通讯的。

CANbus 各层的数据传输如下图所示:

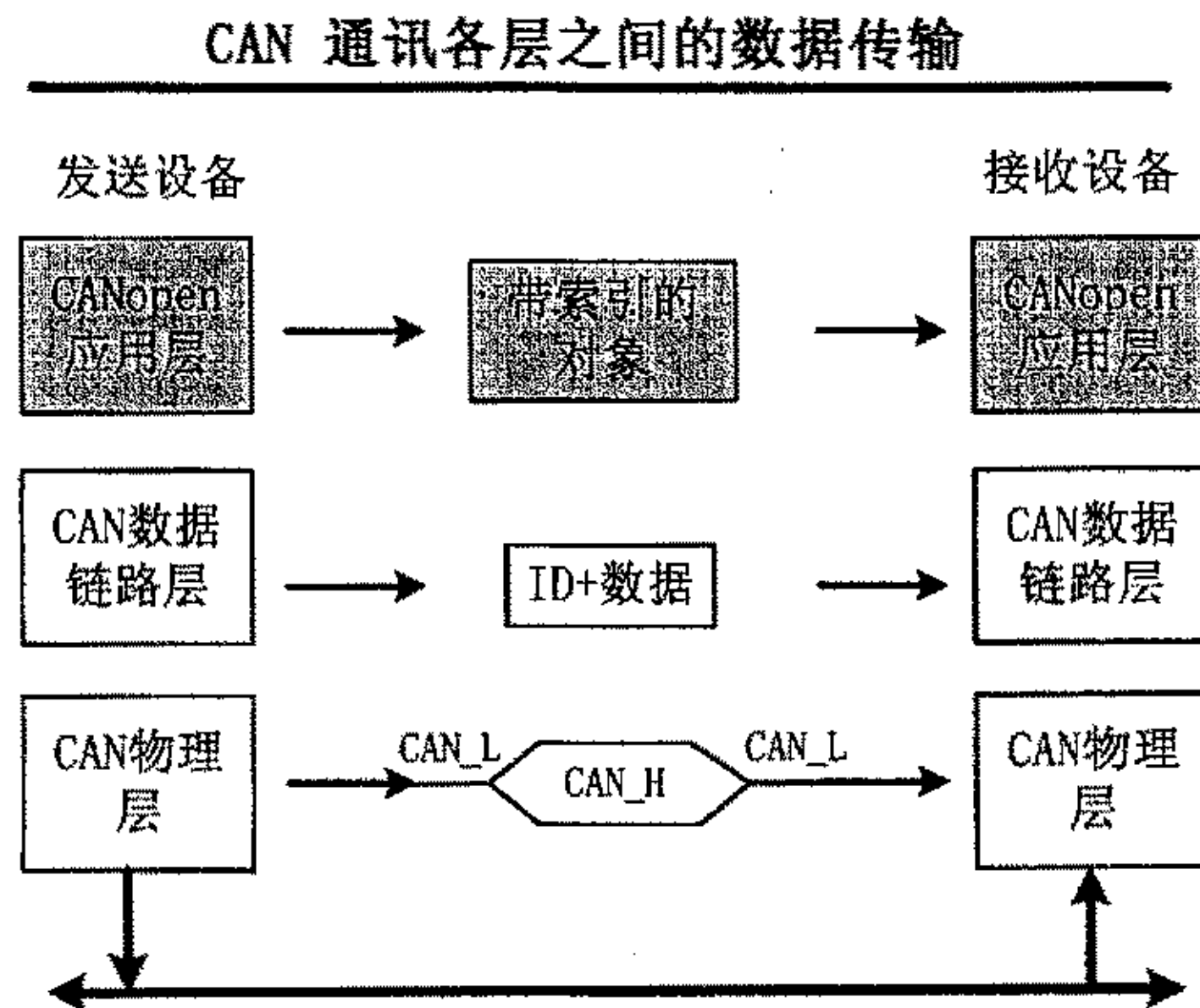


图 2-2 CANopen 各层的数据传输

在 CANopen 应用层，设备之间交换通讯和应用对象。所有这些对象由一个 16 位的主索引和 8 位子索引来定位。这些通讯对象(communication object-COB)被映射到带有标识符的一个或多个 CAN 帧，这些标识符或者是预定义好的，或者是后来配置的。CAN 物理层规定了传输位的电平和每一位占有的时间。

### 2.1.2 CANopen 设备模型

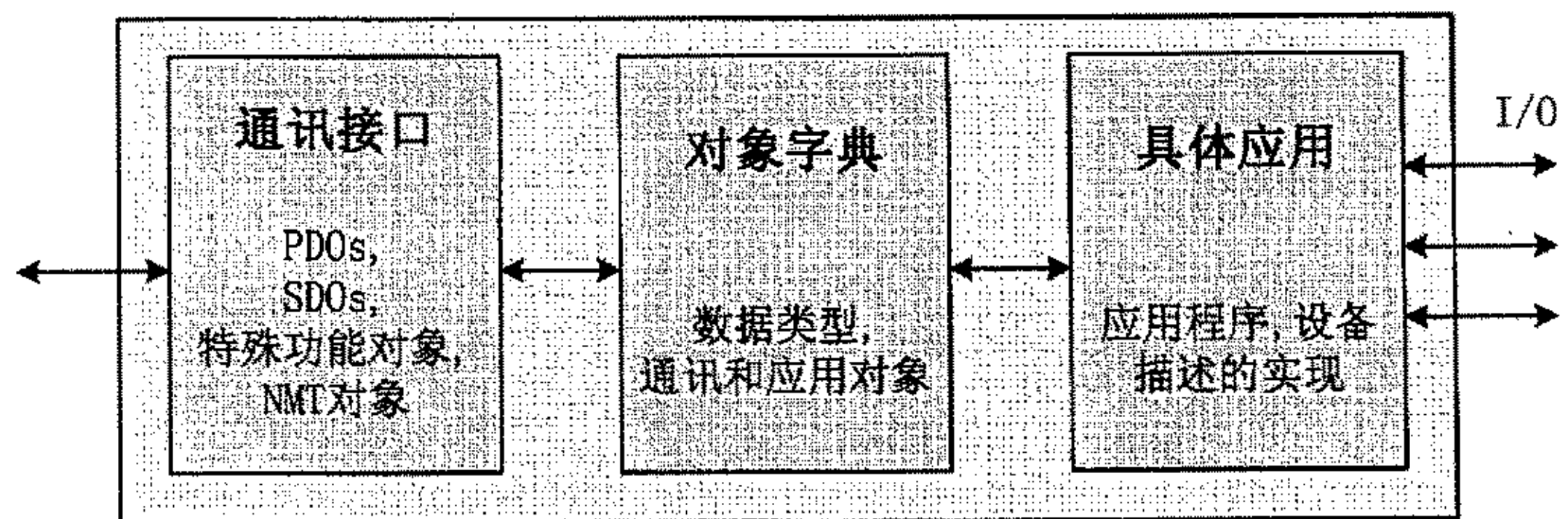


图 2-3 CANopen 设备模型

一个 CANopen 设备可以被分成以下三部分：

- 通讯接口和协议软件
- 对象字典



## ● 过程接口和应用程序

通讯接口和协议软件提供通过总线传输和接受通讯对象的服务，对象字典描述了所有的数据类型、通讯对象和设备用到的应用对象。这是和应用软件的接口。应用程序提供了内部控制的机制，也提供了和其他过程硬件的接口。

### 2.1.3 设备描述的标准化

设备描述 (device profile) 方法的两个主要好处在于系统集成和设备的标准化。如果两个独立的设备生产商想设计可以互相通信的产品，那么每个生产商必须了解另一个厂商的产品的性能规格。如果靠每个厂商来提供这个性能规格的话，那么这些规格将千差万别。设备描述的概念给那些规格提供了一个可供参考的标准。通过采用这个方法所有的生产商将采用类似的形式来规定他们的产品，这样就会在系统集成上减少很多麻烦。

另一个“设备描述”的显而易见的好处是可以指导厂商生产标准化的产品。标准化了产品的益处实在很多。可能最重要的是标准化的设备减少了许多集成商和产品生产商的纠纷。如果一个生产商不能满足需要的话，集成商可以用另外一家的产品，而不用重新构建网络软件。另一方面，生产商不用再被迫为每一个用户来实施一套独特的协议了。

一个“设备描述” (device profile) 定义了一个标准的设备。这个标准规定了每一类设备必须具有的性能机制。这个强制性的性能机制对于保证一个设备的非厂商方面的互操作性是必要的。

设备标准化的概念是用标准“设备描述”里的可选机制来扩展的。那些操作机制对于所有的厂商并不是必要的。但是如果你想让一个产品有那些机制的话，必须按标准来行事。

通过定义强制性的设备特性，基本的网络操作就可以得到保证。通过定义可选的设备特性，就可以有一定的灵活性。

## 2.2 对象字典<sup>[4][5][6]</sup>

### 2.2.1 对象字典的框架

一个 CANopen 设备最重要的部分就是对象字典 (object dictionary)。对象字典就是通过网络可以读取的一组预先定义好形式的对象。字典里的每一个对象可以用一个 16 位索引和 8 位子索引来定位。

一个标准的对象字典的框架如下表所示。这个框架跟其他的工业现场总线概念很相似：

索引 (十六进制)	对象
0000	保留

0001—001F	基本数据类型
0020—003F	复杂数据类型
0040—005F	和生产商相关的数据类型
0060—007F	设备描述的基本数据类型
0080—009F	设备描述的复杂数据类型
00A0—0FFF	保留
1000—1FFF	通讯描述区域
2000—5FFF	特定生产商设备描述区
6000—9FFF	标准设备描述区
A000—FFFF	保留

表 2-1 对象字典结构

基本的数据类型 (Static Data Types) 在 0001h 到 001Fh 的索引区域里, 包括标准的数据类型定义, 像布尔变量、整型、浮点数、字符串等等。这些对象只是做参考用, 不能读写。

复杂的数据类型 (Complex Data Types) 在 0020h 到 003Fh 的索引区域里, 都是预先定义好的结构, 其中包括标准的数据类型, 这些定义对于所有的设备都是相同的。

和生产商相关的数据类型 (Manufacturer Specific Data) 在 0040h 到 005Fh 的索引区域里, 也都是包括标准数据类型的结构, 但是, 是针对某一个特定设备的。

设备描述 (Device Profiles) 可以包括一些额外的针对某一类设备的数据类型。设备描述的基本的数据类型在 0060h 到 007Fh 区域里, 复杂数据类型在 0080h 到 009F 区域里。

一个设备可以选择性的提供它所支持的复杂数据类型 (索引 0020h 到 005Fh 和 0080h 到 0090Fh 之间), 然后子索引 0 提供一个某个索引所包括的对象数目, 接下来的子索引都是无符号 16 位寻址的数据类型。

通信描述区域在索引 1000h 到 1FFFh 之间, 包括 CAN 网络的特定通讯参数。这些对象对于所有设备是相同的。

标准的设备描述区域在索引 6000h 到 9FFFh 之间, 包括所有可以通过网络读写的数据对象, 而这些对象对于一类设备都是相同的。每一类设备的对象字典都有一些的强制的对象。这些对象保证某一类设备在某一个规定的方式下运作。

对象字典还规定了一些可选的设备特性, 这些特性意味着一个厂商不一定必须提供这样额外的服务, 但是如果提供的话, 必须按规定的方式操作。

### 2.2.2 对象字典的索引和子索引

对象字典的所有对象都用一个 16 位的索引来定位。如果是一个简单的变量,

这个索引就直接定位这个变量的值；如果是记录和数组，则这个索引就指向整个这个数据结构。

为了可以通过网络定位一个结构型数据的个别元素，CANopen 定义了子索引。对于单个对象，例如无符号 8 位数、布尔变量、32 位整型等等，子索引的值总是 0。但是对于复杂的对象，例如有多个数据域的数组、记录等等，子索引就可以用来定位这个数据结构的个别元素。例如，一个单通道的 RS232 接口模块，在索引 6092 定义了一个数据结构来描述这个模块的通讯参数。这个结构可能包括的域有：波特率、数据位、停止位、奇偶校验。子索引就可以用来定位某个参数，如下表：

主索引	子索引	变量	数据类型
6092	0	包括的对象数	Unsigned8
	1	波特率	Unsigned16
	2	数据位数	Unsigned8
	3	停止位数	Unsigned8
	4	奇偶校验	Unsigned8

表 2-2 索引和子索引举例

一个子索引也可以用来定位多通道的一个类似的数据域。例如，一个数字输出模块也许可以为不同的极性配置输出信号。对象字典的索引 6022h 就可以定义这些极性信息。某一个生产商可以把他的对象字典的定义成，通道 1 的极性是通过往主索引 6022h 的子索引 1 里写数据来设置的。同样，通道 2 的极性可以通过往子索引 3 里写数据来设置。相同类型的多通道可以看作是同类型的一批通道。

### 2.2.3 数据类型



0001	DEFTYPE	Boolean
0002	DEFTYPE	Integer8
0003	DEFTYPE	Integer16
0004	DEFTYPE	Integer32
0005	DEFTYPE	Unsigned8
0006	DEFTYPE	Unsigned16
0007	DEFTYPE	Unsigned32
0008	DEFTYPE	Real32
0009	DEFTYPE	Visible_String
000A	DEFTYPE	Octet_String
000B	DEFTYPE	Unicode_String
000C	DEFTYPE	Time_Of_Day
000D	DEFTYPE	Time_Difference
000E	DEFTYPE	Bit_String
000F	DEFTYPE	Domain
0010	DEFTYPE	Integer24
0011	DEFTYPE	Real64
0012	DEFTYPE	Integer40
0013	DEFTYPE	Integer48
0014	DEFTYPE	Integer56
0015	DEFTYPE	Integer64
0016	DEFTYPE	Unsigned24
0017		reserved
0018	DEFTYPE	Unsigned40
0019	DEFTYPE	Unsigned48
001A	DEFTYPE	Unsigned56
001B	DEFTYPE	Unsigned64
001C-001F		reserved

表 2-3 基本数据类型

数据字典的基本数据类型是用于定义用的。Boolean 型数据的值为 TRUE 或 FALSE，Integer n 型数据的值为整数，范围为  $-2^{n-1}$  到  $2^{n-1}-1$ 。Unsigned n 型数据为非负的值，范围为 0 到  $2^n-1$ ，Float 型数据的值为实型。

Visible\_String 型数据定义为：Unsigned8 为 Visible Char 型，Visible Char 的数组为 Visible String。Visible Char 型数据的允许值为 0h 和从 20h 到 7Eh 的值。

Octet String 型数据定义为 Unsigned8 型数据的数组。

Date 型数据为 56 位，包括毫秒，分钟，小时，标准或夏日的日期，天，星期，月，年和一些保留位的值。

Time\_Of\_Day 型数据代表绝对时间，包括从 1984 年 1 月 1 日开始的天数和午夜后的毫秒数。

Time\_Of\_Differences 代表一个时间差，作为天数和毫秒数的和。



0020	DEFSTRUCT	PDO_Communication_Parameter
0021	DEFSTRUCT	PDO_Mapping
0022	DEFSTRUCT	SDO_Parameter
0023	DEFSTRUCT	Identity
0024-003F		reserved
0040-005F	DEFSTRUCT	Manufacturer Specific Complex Data Types
0060-007F	DEFTYPE	Device Profile (0) Specific Standard Data Types
0080-009F	DEFSTRUCT	Device Profile (0) Specific Complex Data Types
00A0-00BF	DEFTYPE	Device Profile 1 Specific Standard Data Types
00C0-00DF	DEFSTRUCT	Device Profile 1 Specific Complex Data Types
00E0-00FF	DEFTYPE	Device Profile 2 Specific Standard Data Types
0100-011F	DEFSTRUCT	Device Profile 2 Specific Complex Data Types
0120-013F	DEFTYPE	Device Profile 3 Specific Standard Data Types
0140-015F	DEFSTRUCT	Device Profile 3 Specific Complex Data Types
0160-017F	DEFTYPE	Device Profile 4 Specific Standard Data Types
0180-019F	DEFSTRUCT	Device Profile 4 Specific Complex Data Types
01A0-01BF	DEFTYPE	Device Profile 5 Specific Standard Data Types
01C0-01DF	DEFSTRUCT	Device Profile 5 Specific Complex Data Types
01E0-01FF	DEFTYPE	Device Profile 6 Specific Standard Data Types
0200-021F	DEFSTRUCT	Device Profile 6 Specific Complex Data Types
0220-023F	DEFTYPE	Device Profile 7 Specific Standard Data Types
0240-025F	DEFSTRUCT	Device Profile 7 Specific Complex Data Types

表 2-4 复杂数据类型

CANopen 为 PDO 和 SDO 参数预定义了一些复杂数据类型。而且，对象字典还为特定的设备标准和复杂数据类型保留了一些可以扩展的空间。对于有多个设备模块的设备或者设备描述，像多轴控制器，每一个虚拟的设备就可以用他们自己的数据类型。

## 2.2.4 对象说明

### 1. 对象的说明

名称 (Name)	此对象用途的简短说明
对象类型 (Object Code)	变量，数组，记录等等
数据类型 (Data Type)	Unsigned8, Boolean, Integer16 等等
性质 (Category)	可选项，必选项，或是根据条件而定。

表 2-5 对象的描述

对象类型必须是 CANopen 特性中定义的一种，对于简单变量，这个属性仅仅出现一次，而没有子索引和子索引的类型。对于复杂数据类型，必须为每一个元素(即子索引)定义属性说明。如下表所示。

说明(Description)	子索引的说明名称
数据类型(Data Type)	Unsigned8, Boolean, Integer16, 等等
子索引的类型(Entry Category)	可选(o), 必选(m), 根据条件而定(c)
可读写性	只读(ro), 只写(wo), 读写(rw)
PDO 映射	NO, 可选, 缺省
数据值的范围	这个对象所允许的值的范围
缺省值	设备启动后的值

表 2-6 关于子索引的对象描述

### 2.2.5 和通讯参数有关的对象

在对象字典的 1000h—1FFFh 区域里, 描述了和通讯有关的各种对象, 如下表:

索引(hex)	对象	名称	类型	可读写性	M/O
1000	VAR	Device type	Unsigned32	Ro	M
1001	VAR	Error register	Unsigned8	Ro	M
1002	VAR	Manufacturer status register	Unsigned32	Ro	O
1003	ARRAY	Pre-defined error field	Unsigned32	Ro	O
1004	为以后的兼容性而保留				
1005	VAR	COB-ID SYNC-message	Unsigned32	Rw	O
1006	VAR	Communication cycle period	Unsigned32	Rw	O
1007	VAR	Synchronous window length	Unsigned32	Rw	O
1008	VAR	Manufacturer device name	Vis-String	C	O
1009	VAR	manufacturer hardware version	Vis-String	C	O
100A	VAR	manufacturer software version	Vis-String	C	O
100B	保留				
100C	VAR	guard time	Unsigned32	RW	O
100D	VAR	life time factor	Unsigned32	rw	O
100E	保留				
100F	保留				
1011	VAR	store parameters	Unsigned32	RW	O
1012	VAR	COB-ID time stamp	Unsigned32	RW	O
1013	VAR	high resolution time stamp	Unsigned32	RW	O

1014	VAR	COB-ID Emergency	Unsigned32	RW	O
1015	VAR	Inhibit Time Emergency	Unsigned16	RW	O
1016	ARRAY	Consumer Heartbeat Time	Unsigned32	RW	O
1017	VAR	Producer Heartbeat Time	Unsigned16	RW	O
1018	RECORD	Identity object	Identity	RO	M
...	...	.....	...	.....	.....

表 2-7 和通讯参数有关的对象

下面详细说明一下几个必选的参数对象。

### 1. Error Register

0	1	2	3	4	5	6	7
Generic Error	Current	Voltage	Temperature	Communication Error	Device Profile Specific	保留	Manufacturer specific
M	0	0	0	0	0	0	0

表 2-8 错误寄存器

如果哪一位被置为 1，则说明发生了相应的错误。唯一一个必选的位是 Generic error，不管发生任何错误，这一位都会被置位。

一个设备的错误寄存器在对象字典的 1001h 里，设备可以把内部错误映射到这个字节里。而且这个对象是对所有设备来说是必需的，属于应急对象的一部分。

### 2. Identity Object

索引	子索引	说明	数据类型
1018h	0	包含的对象数	Unsigned8
	1	厂商 ID	Unsigned32
	2	产品号	Unsigned32
	3	修订号	Unsigned32
	4	序列号	Unsigned32
<div> <div>3124230</div> <div> <div>部门</div> <div>公司</div> </div> <div>MSB 厂商 ID LSB</div> </div> <div> <div>3116150</div> <div> <div>主修订号</div> <div>次修订号</div> </div> <div>MSB 修订号 LSB</div> </div>			

表 2-8 ID 对象

这个必需的 ID 对象在对象字典的 1018h, 包含了设备的一般信息。厂商 ID 是一个 32 位无符号数, 包括一个唯一的公司号, 并且如果需要的话, 也包括公司内部的一个唯一的部门号。公司厂商 ID 是由 CiA 总部负责分配的。公司号和部门号都必须在 CiA 注册, 并且需要付 128 欧元和一定的德国增值税, 对 CiA 成员免费。

产品号表示了一个特定产品的版本。修订号包括一个主修订号和一个次修订号。如果设备的 CANopen 功能扩展了, 主修订号必须增加。

## 2.3 通讯协议<sup>[4][5][6]</sup>

通过 CAN 网络传输的 CANopen 通讯对象可以通过服务和协议来描述。分类如下:

- 实时数据的传输通过“过程数据对象”(PDO)协议来完成。
- 通过“服务数据对象”(SDO), 可以实现对对象字典的读写操作。
- “特殊功能对象”协议提供了特定应用的网络同步, 时间戳和应急报文传输。
- 网络管理(NMT)协议提供了网络初始化, 错误控制和设备状态控制的服务。

### 2.3.1 过程数据对象(PDO)的传输

#### 1. 生产者/消费者通讯方式

首先介绍一下生产者/消费者通讯方式。如下图:

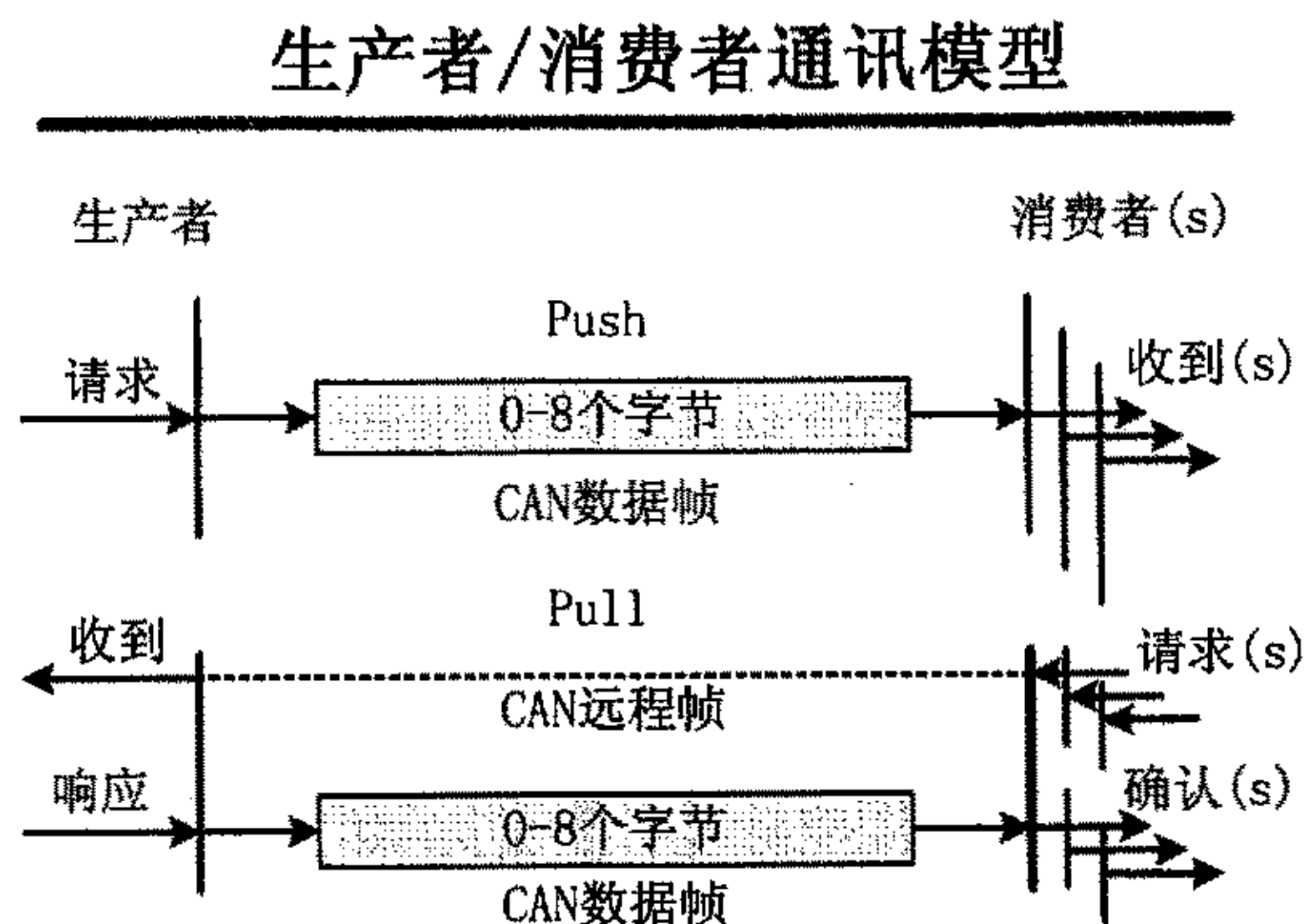


图 2-4 生产者/消费者通讯模型

生产者/消费者模型很好的描述了 CAN 的广播通讯能力。网上的每一个站都可



以侦听发送站的报文。收到后，由每个节点决定是否接收这个报文，所以在在一个 CAN 节点里接收滤波是必需的。

CAN 的广播通讯类似于一个为车辆司机发送交通堵塞信息的广播站，每个司机必须根据他想去的方向上决定广播的信息对于他是否重要。

生产者/消费者模型需要的服务有：直接发送一个报文(Push 型)或者请求一个报文(pull 型)。

## 2. 过程数据对象 PDO 的传输

PDO 通讯可以描述成生产者/消费者模型。“过程数据”可以从一个设备(生产者)发送到另一个或很多个设备。PDO 的发送是非确认的方式。如下图所示：

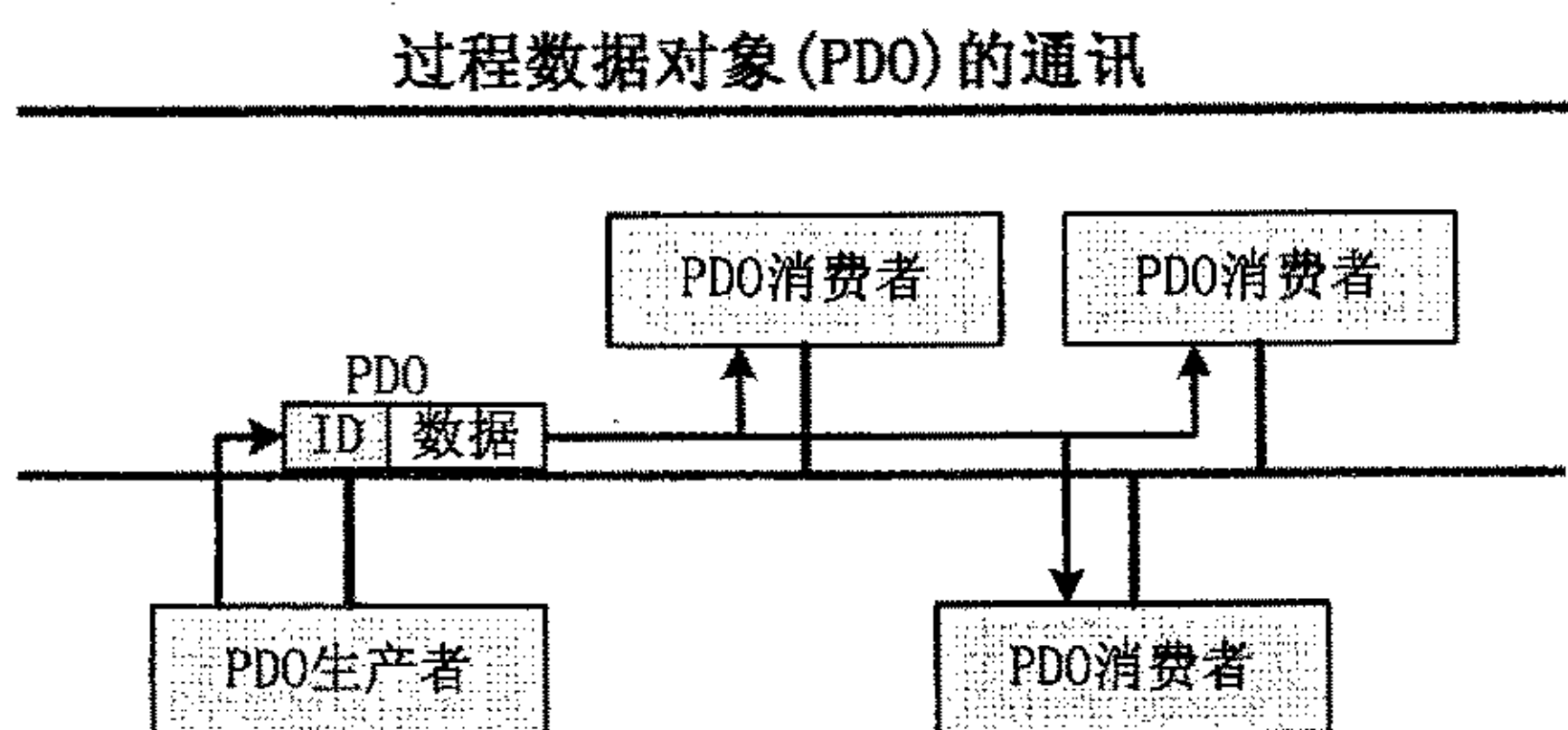


图 2-5 PDO 的发送

生产者发送一个带有特定标识符的 Transmit-PDO(T\_PDO)，这个标识符对应着消费者的 Receive-PDO(R\_PDO)的标识符。

PDO 的通讯有两种：写 PDO 和读 PDO。如下图：

## PDO 协议

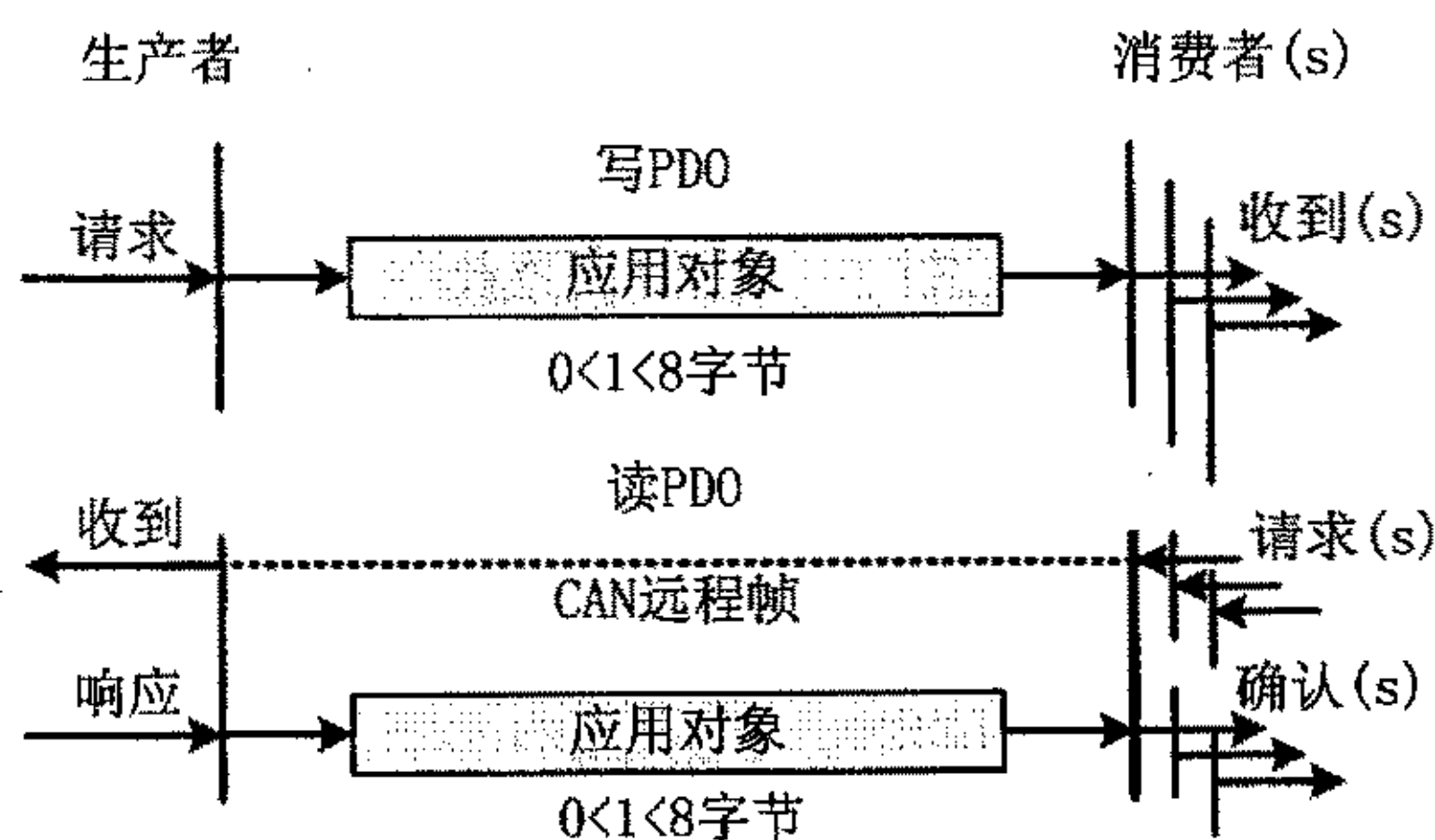


图 2-6 PDO 协议

Write\_PDO 被映射到一个 CAN 数据帧。Read-PDO 被映射到一个 CAN 远程帧 (CAN Remote Frame)，这个远程帧会被相应的 CAN 数据帧所响应。Read-PDO 是可选的，取决于设备的能力。

一个完全的数据域顶多有 8 个字节，里面包含了过程数据。一个设备的具体 PDO 的数量和长度是根据具体情况而定的，必须在设备描述 (profile device) 里规定好。

PDO 对应着对象字典里的具体元素，提供了和应用对象的接口。应用对象的数据类型和映射参数是由对象字典的缺省 PDO 映射结构来决定的。第一个 R\_PDO 的映射结构在对象字典的 1600h 里，第一个 T\_PDO 的映射结构在 1A00h 里。在一个 CANopen 网络里，最多有 512 个 T\_PDO，512 个 R\_PDO。

### 3. PDO 的调度触发模式

CANopen 的 PDO 通讯有三种报文触发模式，如下图：

## PDO 触发模式

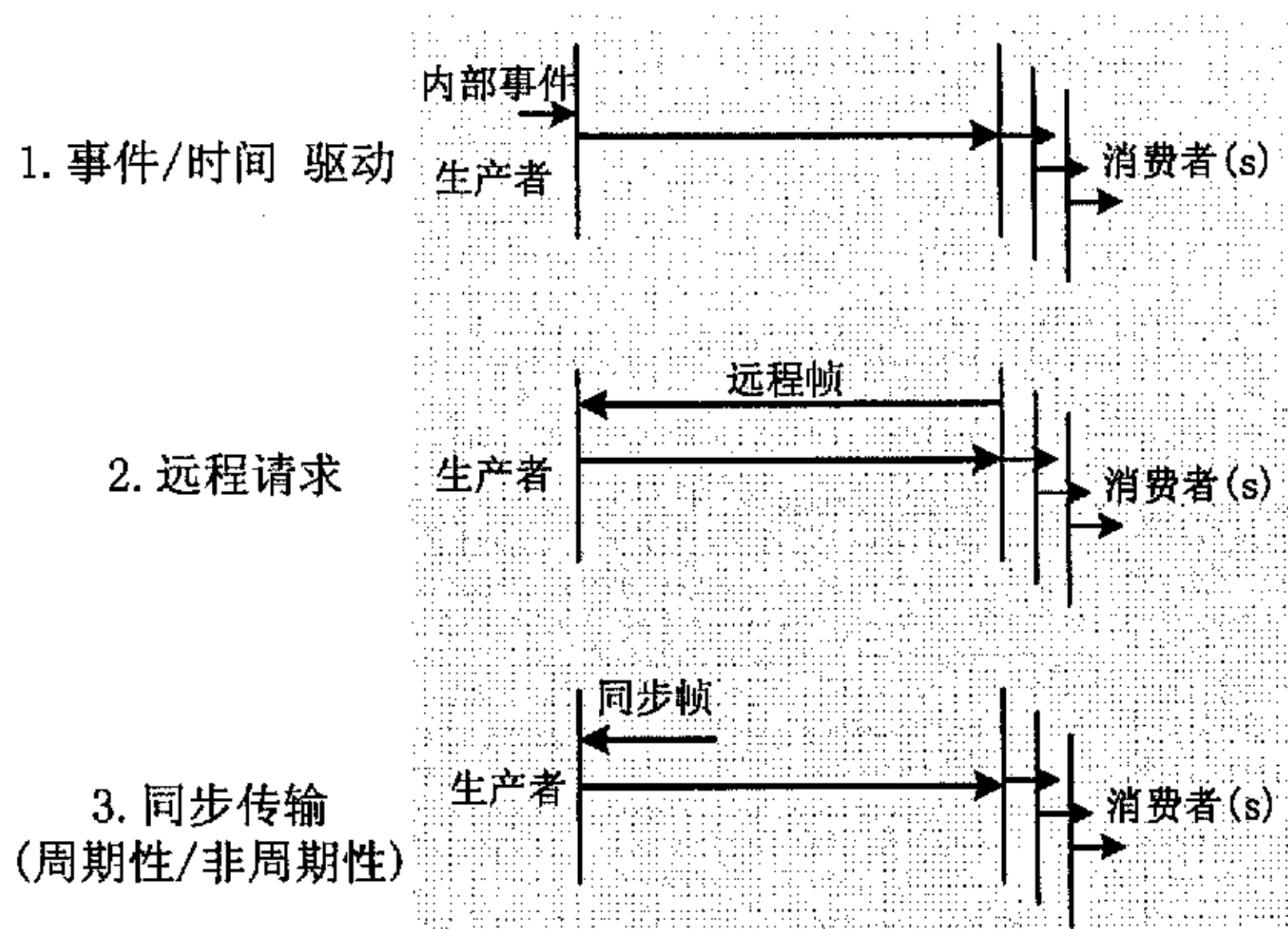


图 2-7 PDO 触发模式

- 1) PDO 的传输是由设备描述里一个特定的事件来触发的，或者，即使没有特定事件发生也可以由定时器来触发。
- 2) 异步传输的 PDO 也可以由收到另外一个节点的远程请求来启动。
- 3) 同步传输 PDO，一个指定的传输时间到了，就会触发 PDO 的传送，这个传输时间是由 SYNC(同步对象)的到来实现的。

同步和异步 PDO 如下图所示，

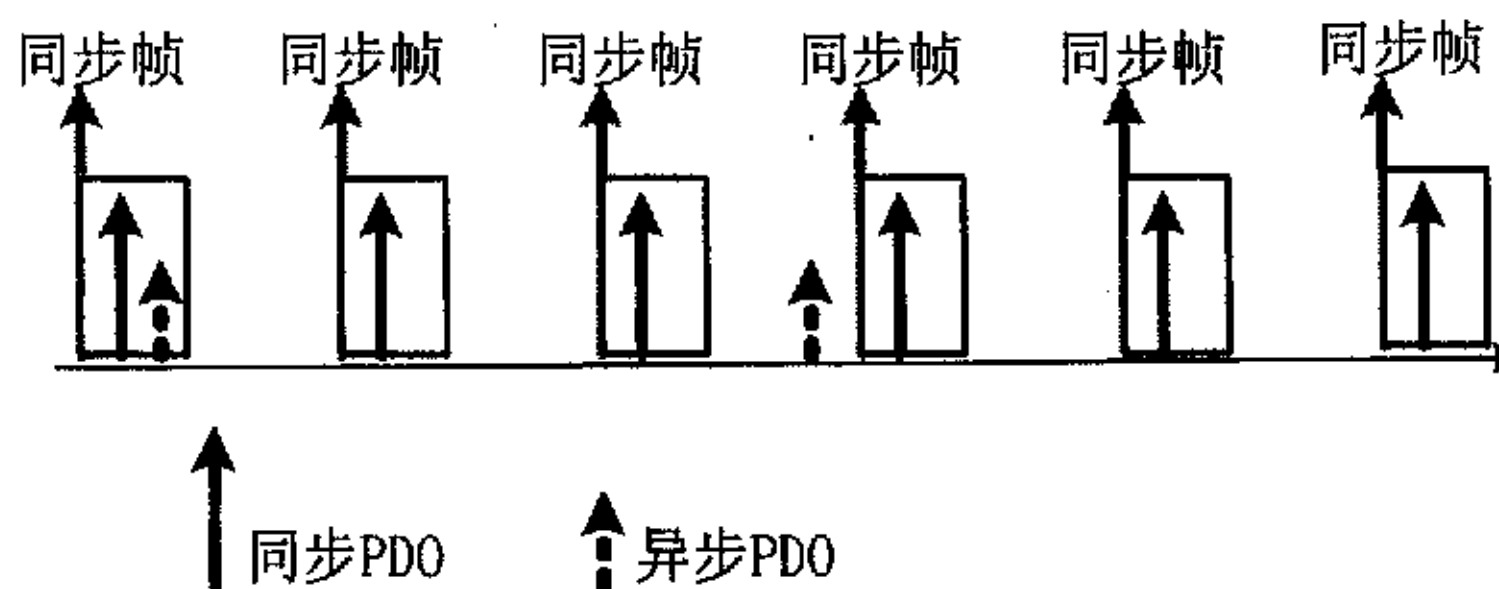


图 2-8 同步 PDO

同步 PDO 在同步对象(Sync Object)到来后的同步窗内传输。同步 PDO 的优先级应该高于异步 PDO。异步 PDO 可以根据他们的优先级在任何时候发送，SDO 也是这样的。所以他们也可能正好赶上在同步窗内传输。

另外，同步的 PDO 传输还可以分成周期和非周期传输模式。如下图。

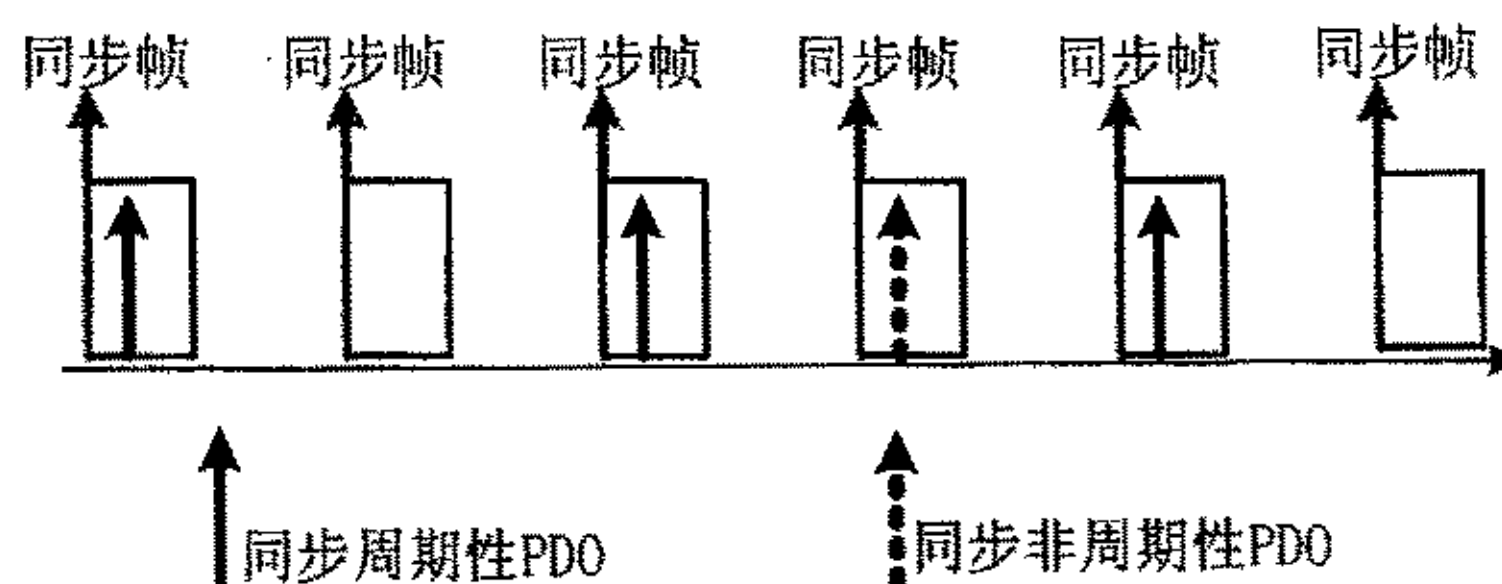


图 2-9 同步周期性 PDO 和同步非周期性 PDO 传输

同步周期性 PDO 在同步窗内传输。“传输类型”的值(1-240)表示每两个 PDO 之间的同步对象数。同步非周期性 PDO 是由特定的应用触发的。报文必须跟着同步对象发送，但是不一定每个周期都发送。

总之，这些报文的触发调度方式是由 PDO 的“传输类型”参数的值决定的，如下表所示。

传输类型值	周期性	非周期性	同步	异步	仅适用于远程帧
0		✓	✓		
1-240 <sup>1</sup>	✓		✓		
241-251			保留		
252 <sup>2</sup>			✓		✓
253 <sup>3</sup>				✓	✓
254 <sup>4</sup>				✓	
255 <sup>5</sup>				✓	

- 1 数字表示了两个 PDO 传输之间的同步对象数。 2 数据在收到同步对象后立即更新，但不发送。  
 3 数据在收到 RTR(远程帧)后更新。 4 应用事件是根据设备而定的。  
 5 应用时间是定义在设备描述里的。

表 2-9 PDO 的“传输类型”参数

同步传输(传输类型 0-240 和 252)意味着 PDO 的传输应该和同步对象有关。非周期(传输类型 0)意味着报文应该跟着同步对象同步地传输，但不是周期性地传输。一般，设备利用同步对象来触发输出，或者根据以前的 Receive-PDO 来更新将要在下一个同步 Transmit-PDO 传输的数据。具体的机制取决于设备类型。另外，提一下，PDO 的“传输类型”参数定义在 PDO 通讯参数里。(索引号，第一个 R\_PDO 为 1400h，第一个 T\_PDO 为 1800h)。

#### 4. PDO 的“禁止时间”(Inhibit Time)

为了保证那些低优先级的通讯对象不会在网络中“饿死”，即总是传输不出



去，给 PDO 分配了一个禁止时间。禁止时间定义了在两个连续的 PDO 的传输之间必须保留的最短时间。例如下图所示。

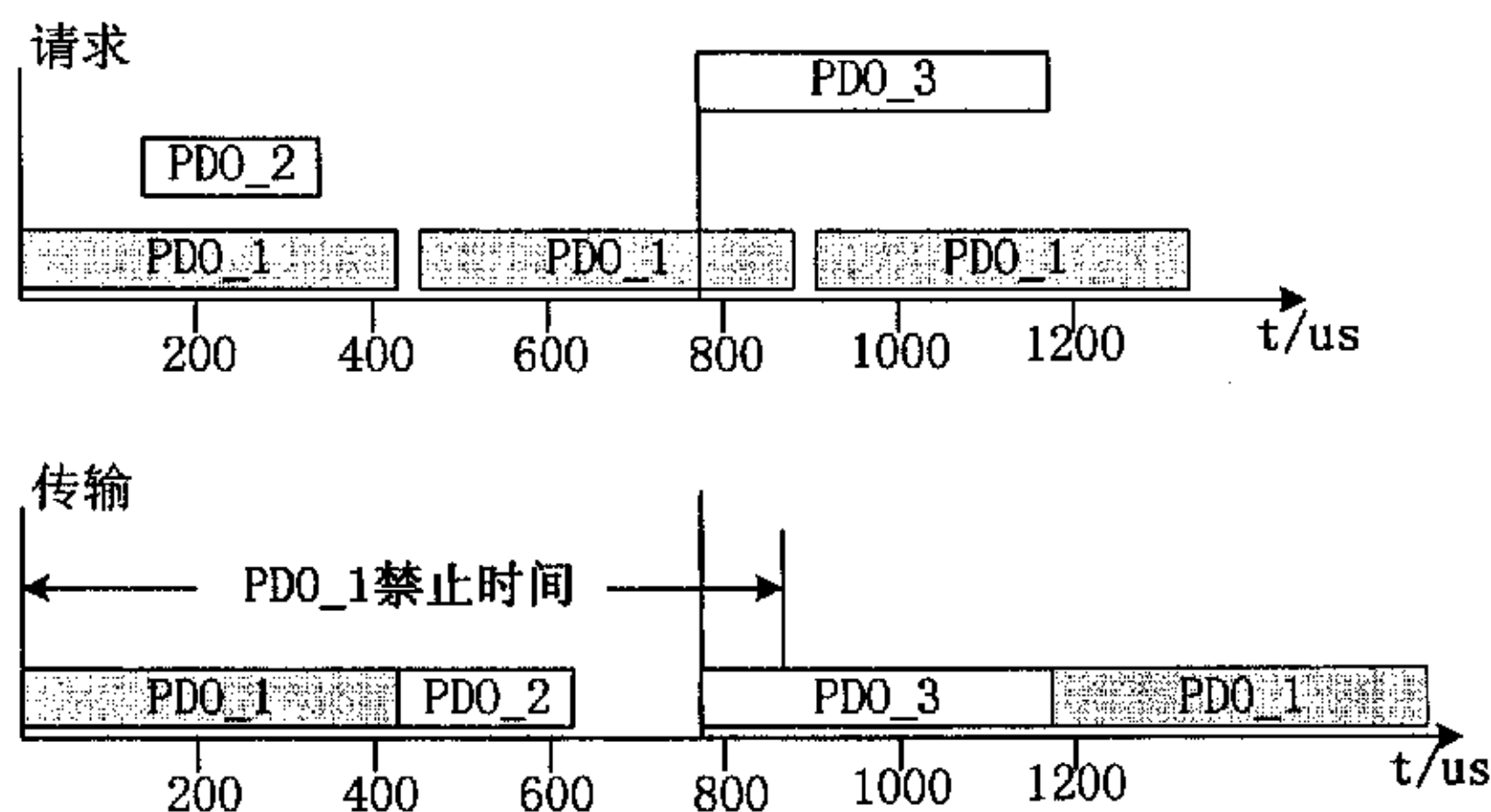


图 2-10 PDO 的“禁止时间”

在上面的例子里，PDO\_1 比 PDO\_2 和 PDO\_3 的优先级高，但是禁止时间允许 PDO\_2 和 PDO\_3 得到总线权，即使它们比 PDO\_1 的优先级低。

所以，把禁止时间和事件定时器(Event\_Timer)结合起来，系统的设计者就可以给每一个 Transmit\_PDO 分配一个虚拟窗，只有在这个窗里，PDO 才可以发送。PDO 发送的最早的时间是禁止时间过后，最晚的时间是事件定时(event-time)之前，如下图。虚拟窗可以用来减少总线负载。

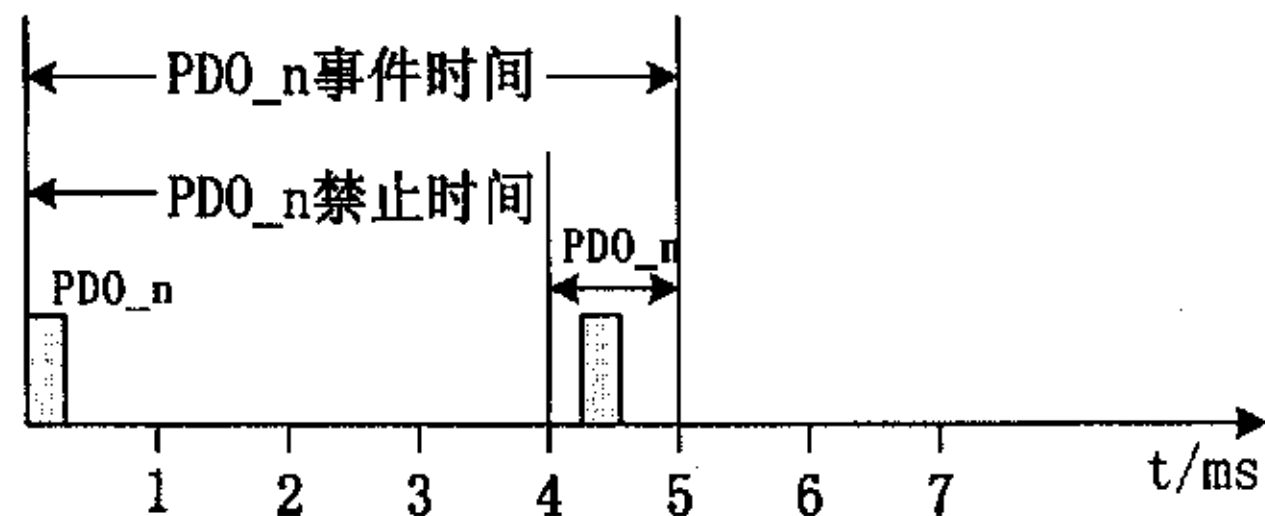


图 2-11 PDO 的虚拟传输窗口

## 5. PDO 参数集

PDO 在对象字典中的参数集如下图所示。

接收 PDO 通讯参数 (20h)					
1400	RECORD	第 1 个接收的 PDO 参数	PDO 通讯参数	RW	M/O
1401	RECORD	第 2 个接收的 PDO 参数	PDO 通讯参数	RW	M/O
...	...	.....	.....	...	...
15FF	RECORD	第 512 个接收的 PDO 参数	PDO 通讯参数	RW	M/O

接收 PDO 映射参数 (21h)					
1600	ARRAY	第 1 个接收 PDO 映射	PDO 映射参数	RW	M/O
1601	ARRAY	第 2 个接收 PDO 映射	PDO 映射参数	RW	M/O
.....	.....	.....	.....	...	.....
17FF	ARRAY	第 512 个接收 PDO 映射	PDO 映射参数	RW	M/O
发送 PDO 通讯参数 (20h)					
1800	RECORD	第 1 个发送 PDO 参数	PDO 通讯参数	RW	M/O
1801	RECORD	第 2 个发送 PDO 参数	PDO 通讯参数	RW	M/O
.....	.....	.....	.....	...	.....
19FF	RECORD	第 512 个发送 PDO 参数	PDO 通讯参数	RW	M/O
发送 PDO 映射参数 (21h)					
1A00	ARRAY	第 1 个发送 PDO 映射	PDO 映射参数	RW	M/O
1A01	ARRAY	第 2 个发送 PDO 映射	PDO 映射参数	RW	M/O
.....	.....	.....	.....	...	.....
1BFF	ARRAY	第 512 个发送 PDO 映射	PDO 映射参数	RW	M/O

表 2-10 PDO 的参数集

## 1) PDO 通讯参数

通讯参数规定了 PDO 的通讯行为, 像标识符, 传输类型, 禁止时间等等, 如下表所示。通讯参数格式在对象字典的 20h 有定义。

索引	子索引	内容	数据类型
1XXXh	0h	子索引元素数	Unsigned8
	1h	COB-ID	Unsigned32
	2h	传输类型	Unsigned8
	3h	禁止时间	Unsigned16
	4h	保留	Unsigned8
	5h	事件定时器	Unsigned16

表 2-11 PDO 的通讯参数

## 2) PDO 映射参数

PDO 映射就是把对象字典里需要发送的对象映射到一个 PDO 里, 然后发送出去。如下图所示。

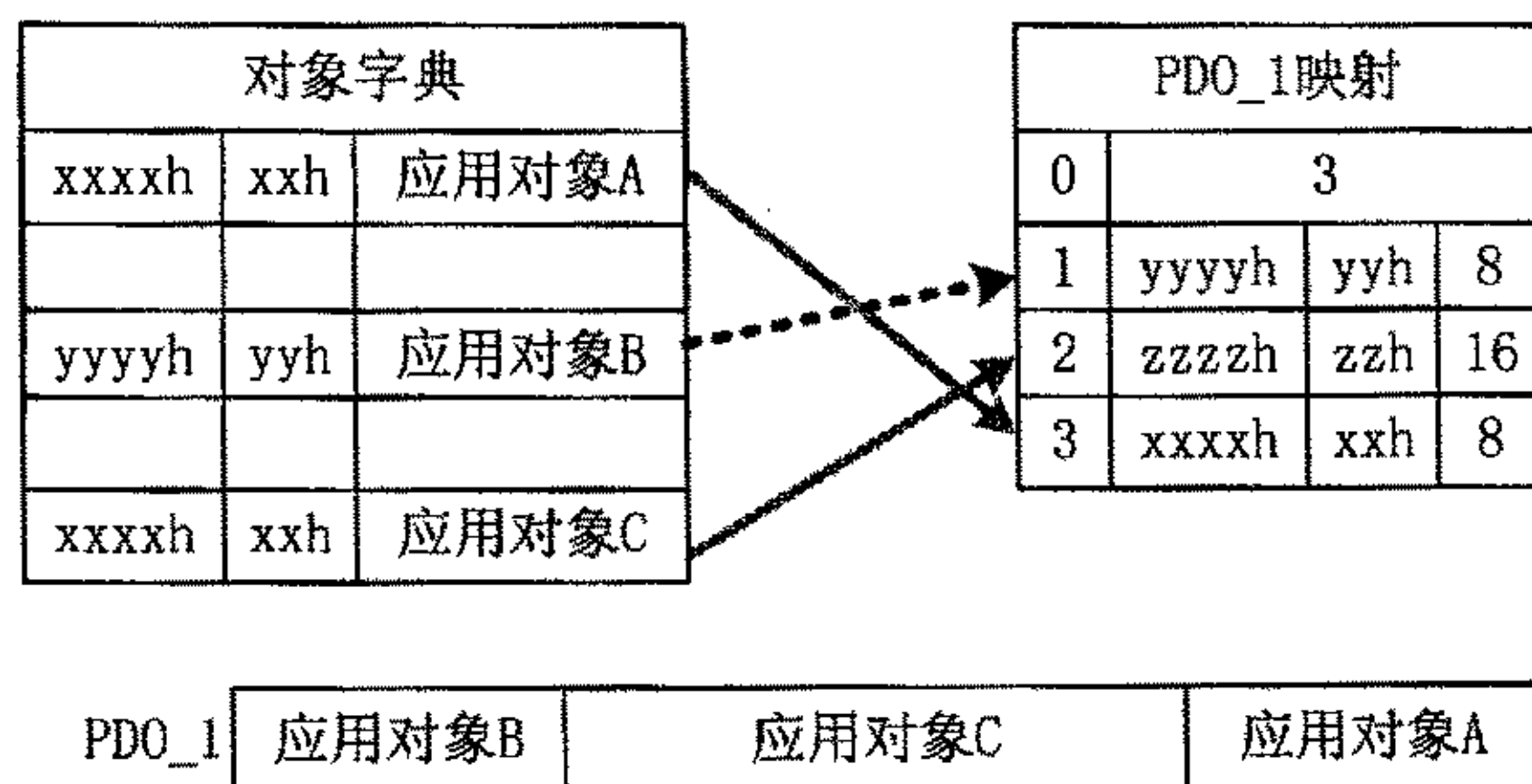


图 2-12 PDO 映射

对象字典的 21h 规定了映射参数的格式，在映射参数中规定了哪个对象被映射到一个 PDO 里。最多可以映射 64 个对象。具体映射参数如下表。

索引	子索引	内容	数据类型
1XXXh	0h	映射的数量	Unsigned8
	1h	第 1 个对象	Unsigned32
	2h	第 2 个对象	Unsigned32
	.....	.....	.....
	40h	第 64 个对象	Unsigned32

31
16 15
8 7
0

16 位索引	8 位子索引	长度
--------	--------	----

表 2-12 PDO 的映射参数

### 2.3.2 服务数据对象(SDO)的传输

#### 1. 客户/服务器通讯方式

首先介绍一下客户/服务器通讯方式。如下图所示。

## 客户/服务器通讯模型

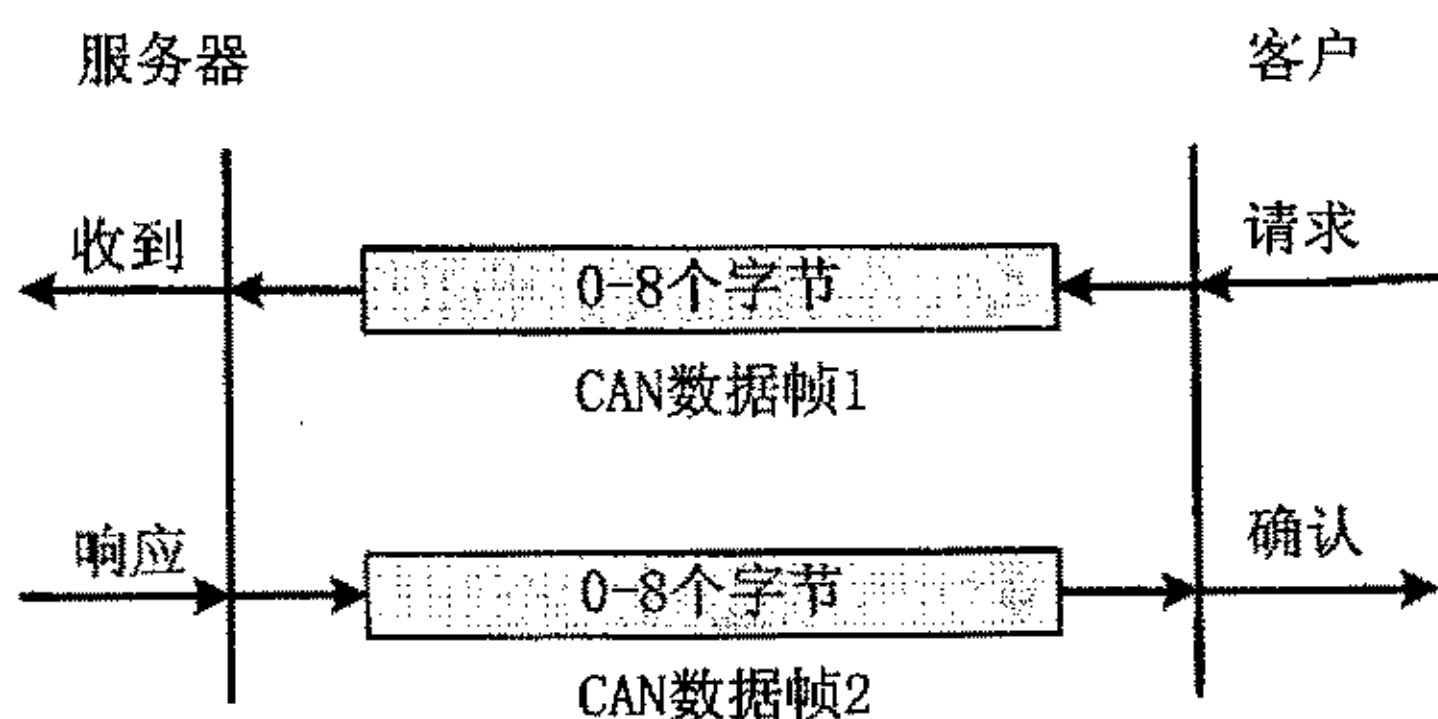


图 2-13 客户/服务器通讯方式

在经典的客户/服务器通讯模型里，客户传输一个报文，然后服务器会做出响应，客户再确认。就像发出一个命令，那个命令必须被确认，你才知道那个命令对方听懂了。

客户/服务器模型用来传输大于 8 个字节的数据。所以，原始的数据需要分段，然后带着相同的标识符分段发出去。每一段或者几段，或者全部报文一起被接受者确认。所以这是一个点对点的通讯。客户/服务器的通讯模型中的服务包括上传、下载、放弃传输。

### 2. 服务数据对象(SDO)的传输

#### 服务数据对象(SDO)的通讯

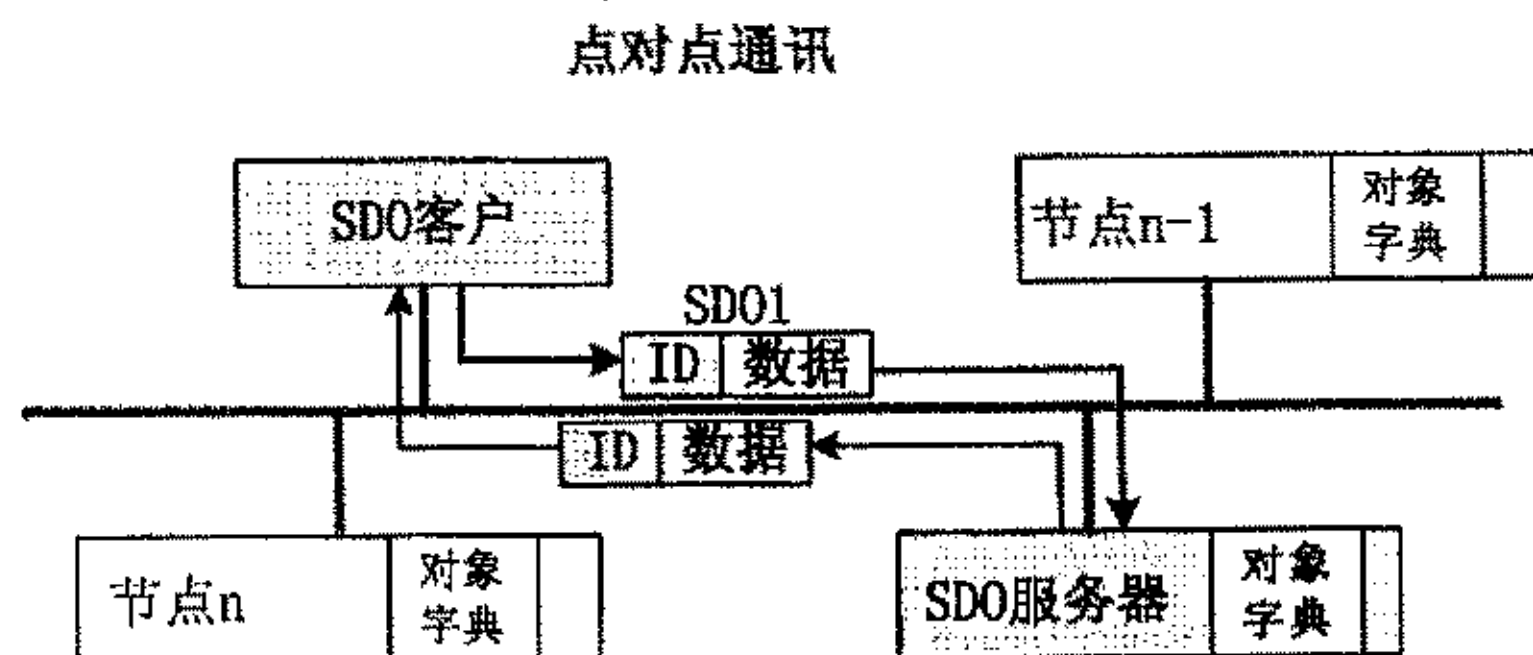


图 2-14 SDO 的通讯

通过服务数据对象(SDO)，就可以访问一个设备的对象字典。一个 SDO 用两个带有不同标识符的 CAN 数据帧，因为通讯需要确认。通过 SDO，在两个设备间就可以建立起点对点的通讯。被访问对象字典的设备是 SDO 通讯的服务器，另一方是客户。一个设备可以支持一个或多个 SDO 通讯。

SDO 通讯允许传输任意大小的数据。它们一般是很多连续的段。如果是传输的数据是 4 个以下的字节，就可以通过“开始域上传/下载”(Initiate Domain Down/Upload)来加快传输。用这个方式，数据可以在开始的通讯里直接完成。



如果数据大于 4 个字节，就必须采用一段一段的方式传输。这样，就可以传输任意长度的数据，因为数据会被分成好几个 CAN 报文。发送 SDO 的第一个 CAN 报文后，其余的段可以每个包含 7 个字节的有用的数据。最后的段可以包含一个终止符。如下图所示。

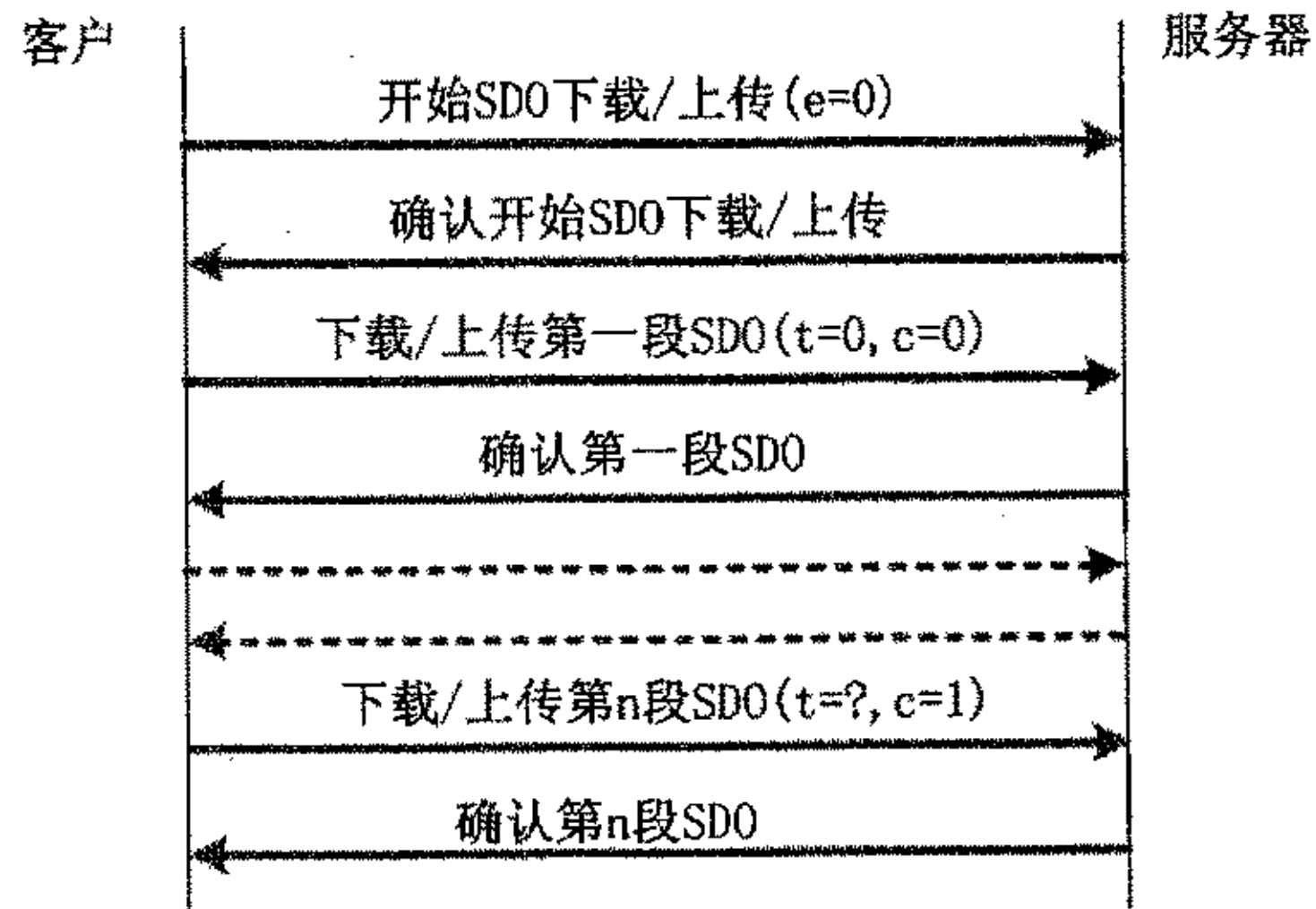


图 2-15 SDO 的分段传输

一个 SDO 通过“确认”的方式传输，也就是每段的接收通过相应的 CAN 报文来通知确认。一般总是客户启动传输过程。此外，客户和服务器都可以主动的放弃数据的传输。

或者，一个 SDO 可以作为一些块传送。每个块最多传送 127 个段，然后一起确认。SDO 的块传送在后面有详细描述。

### 3. 对象字典的读取

SDO 的数据帧格式如下图所示，

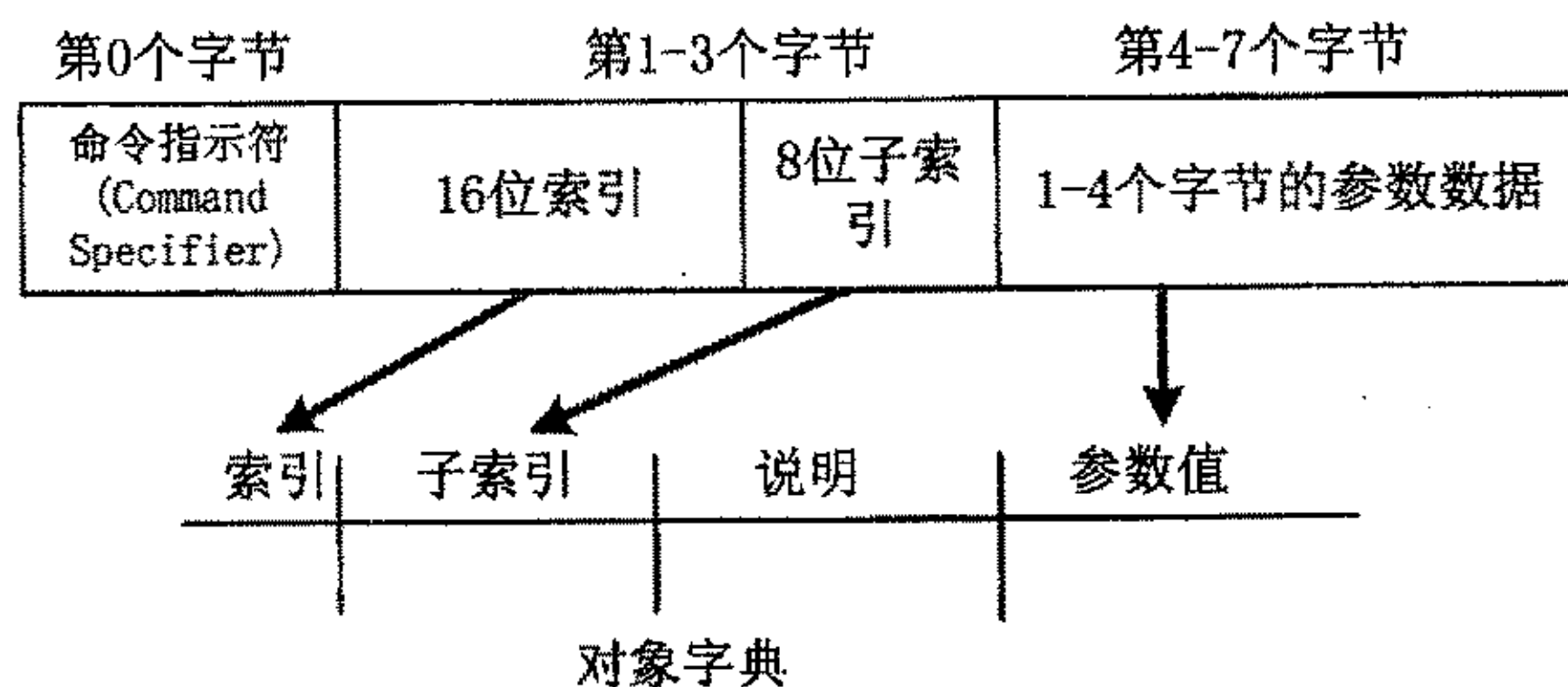


图 2-16 SDO 的数据格式

其中命令指示符包括了以下信息：

◇ 下载/上传

- ◇ 请求/响应
- ◇ 分段/成块/加速传输
- ◇ 数据字节的数目
- ◇ 终止符
- ◇ 下一个段的触发位(alternating toggle bit)

缺省的 Server-SDO(S\_SDO) 通讯参数在对象字典的 1200h, 第一个 Client-SDO 的通讯参数在对象字典的 1280h。

当读写服务器的对象字典时，必须先启动“SDO Down/Upload”。需要注意的是数据字节的传送是最高位在前，最低位在后。如下图所示。

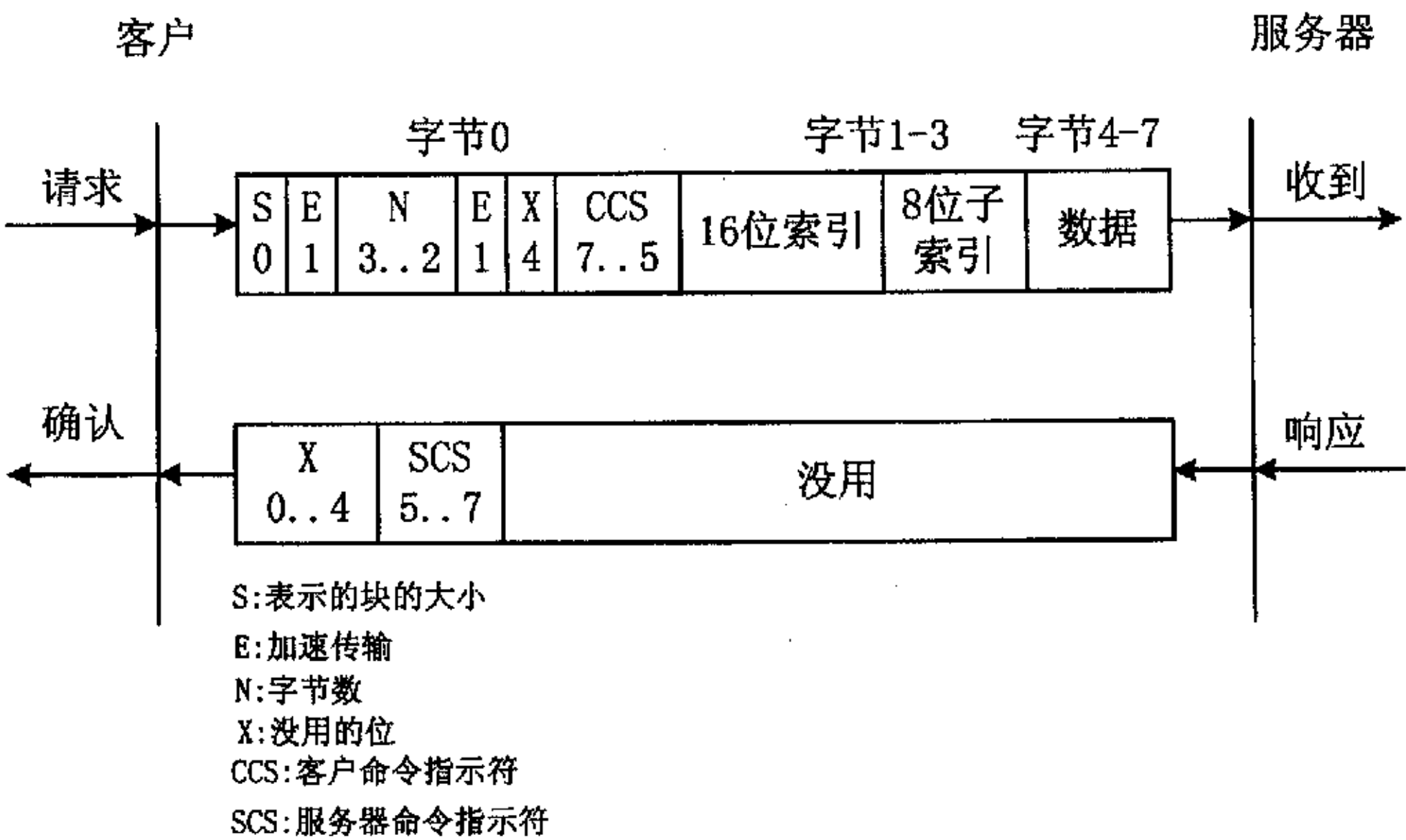


图 2-17 启动 SDO 下载

对每个 SDO 段的上传(下载)来说，数据在客户和服务器之间交换。请求方携带着数据段和分段控制信息，响应方在收到后应答。过程如下图所示。

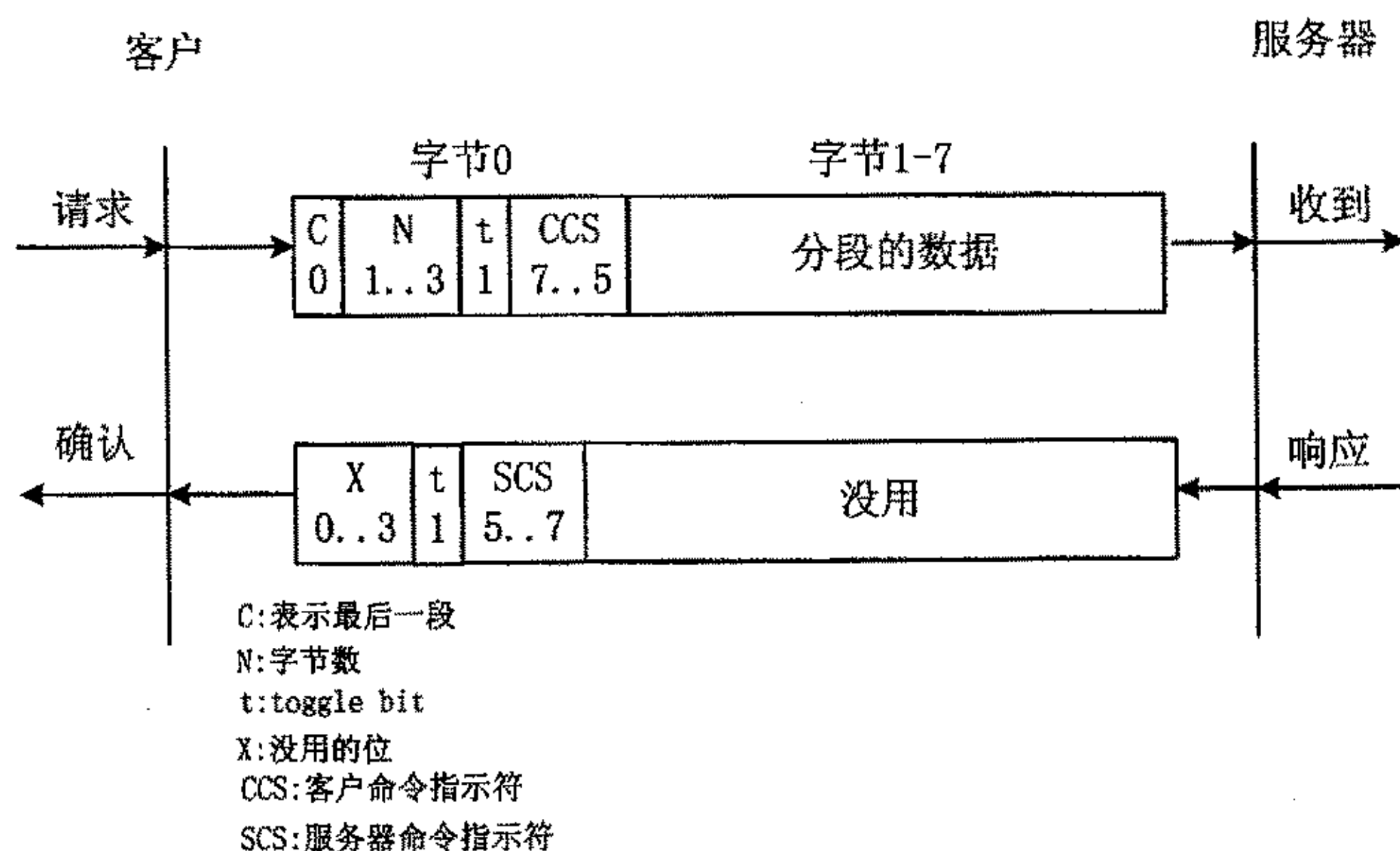


图 2-18 下载 SDO 段

SDO 的传送可以被任何一方放弃。“放弃 SDO 传送”用来通知客户或服务器传送错误。如下图。

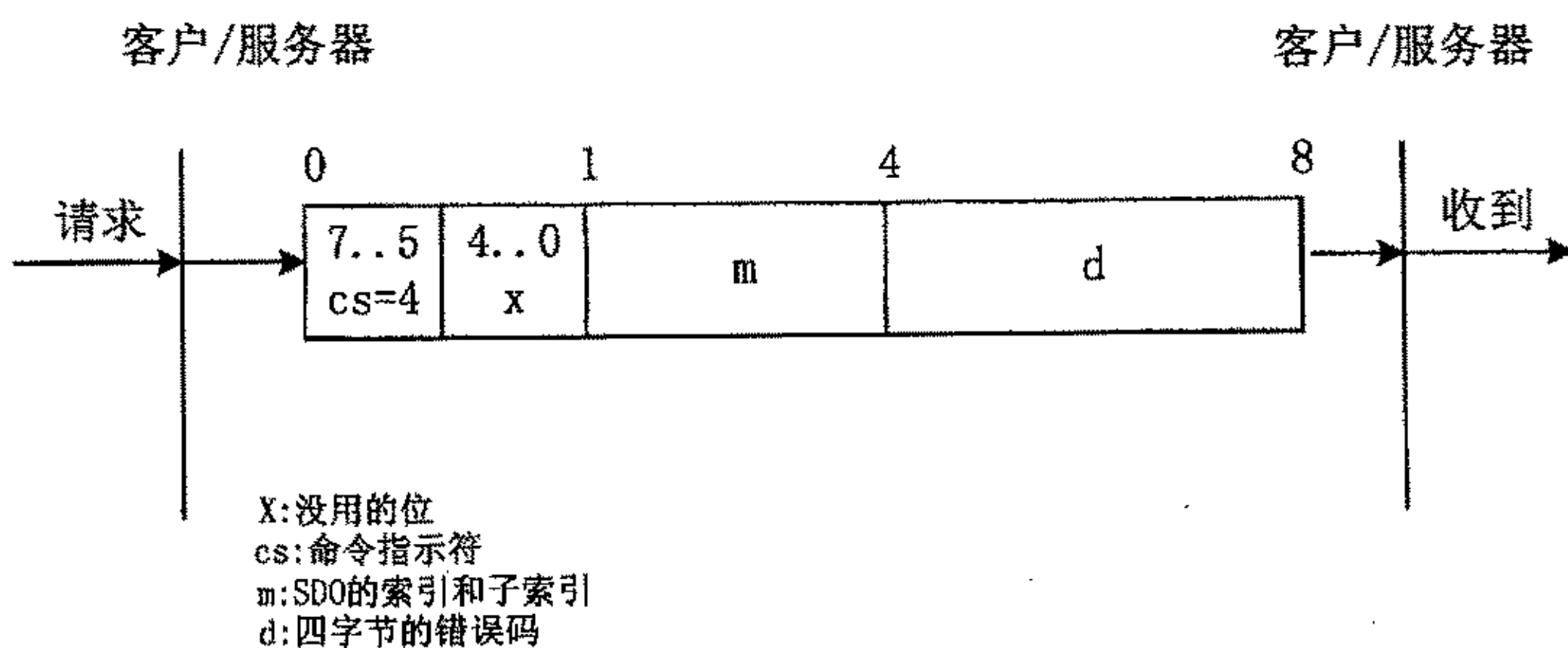


图 2-19 中止(放弃)SDO 传送

下面再说一下 SDO 的块传送方式。

SDO 的下载(上传)块协议规定由 SDO 块下载(上传)报文启动, 然后跟着一系列 SDO 下载(上传)块。最后由一个“终止 SDO 块下载”(上传)结束传送。如下图所示。

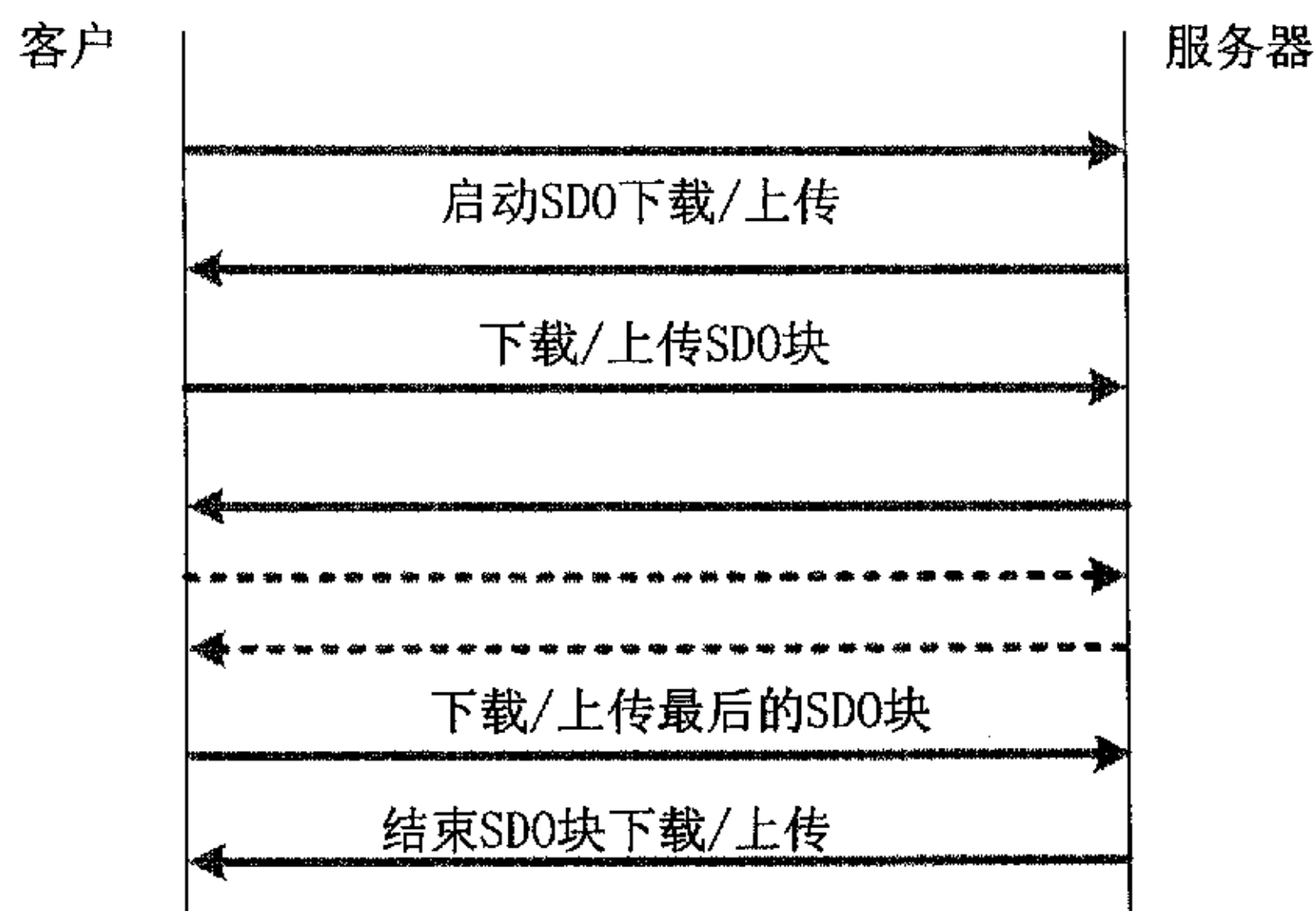


图 2-20 SDO 块传送

下载的 SDO 块最大 127 段，然后用一个报文确认。如果确认的报文的 c 位为 1，如下图，那么说明块被成功接收。不成功的接收会用“中止 SDO 传送请求/回应”来表明。

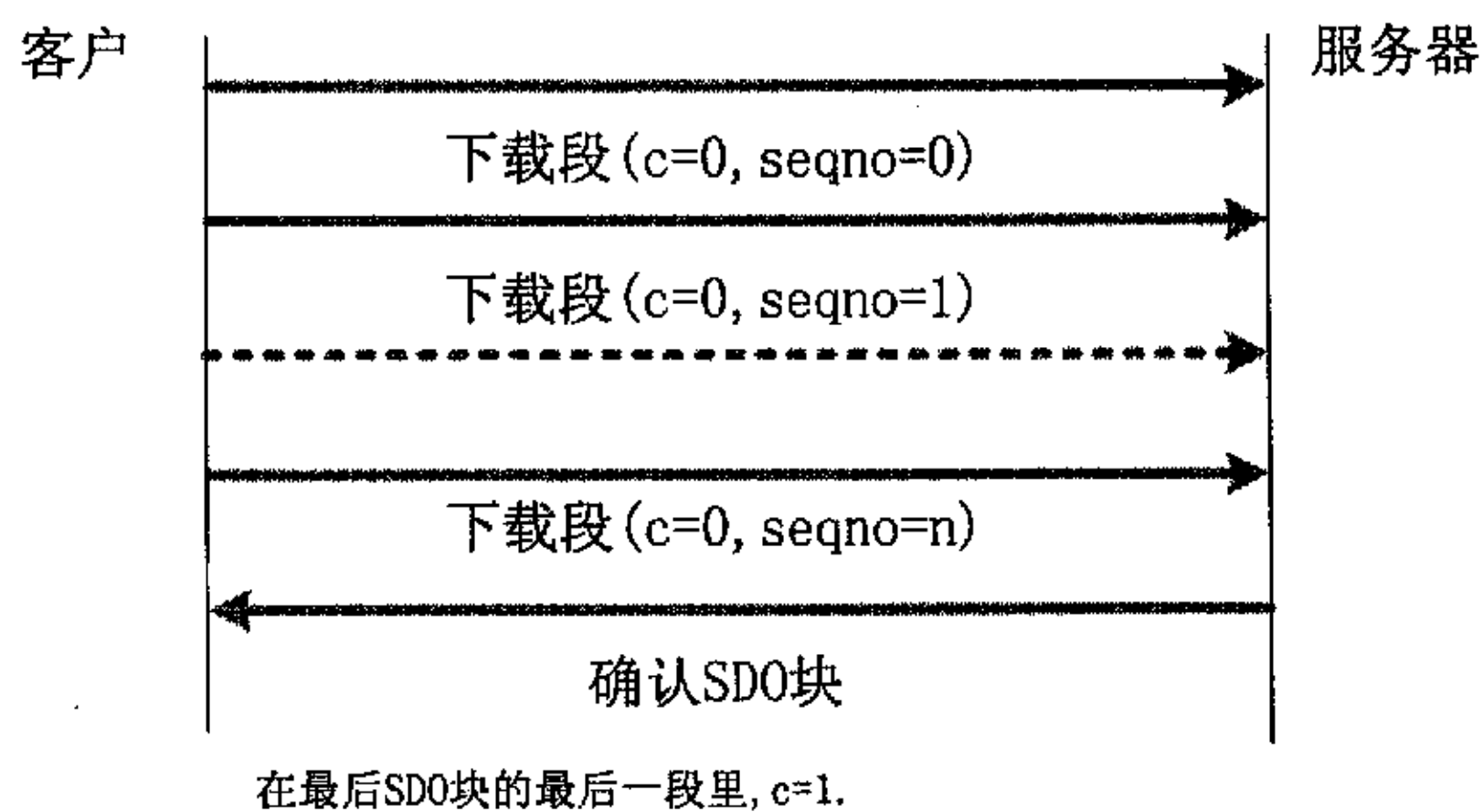


图 2-21 下载 SDO 块

结束块传送，如下图所示。



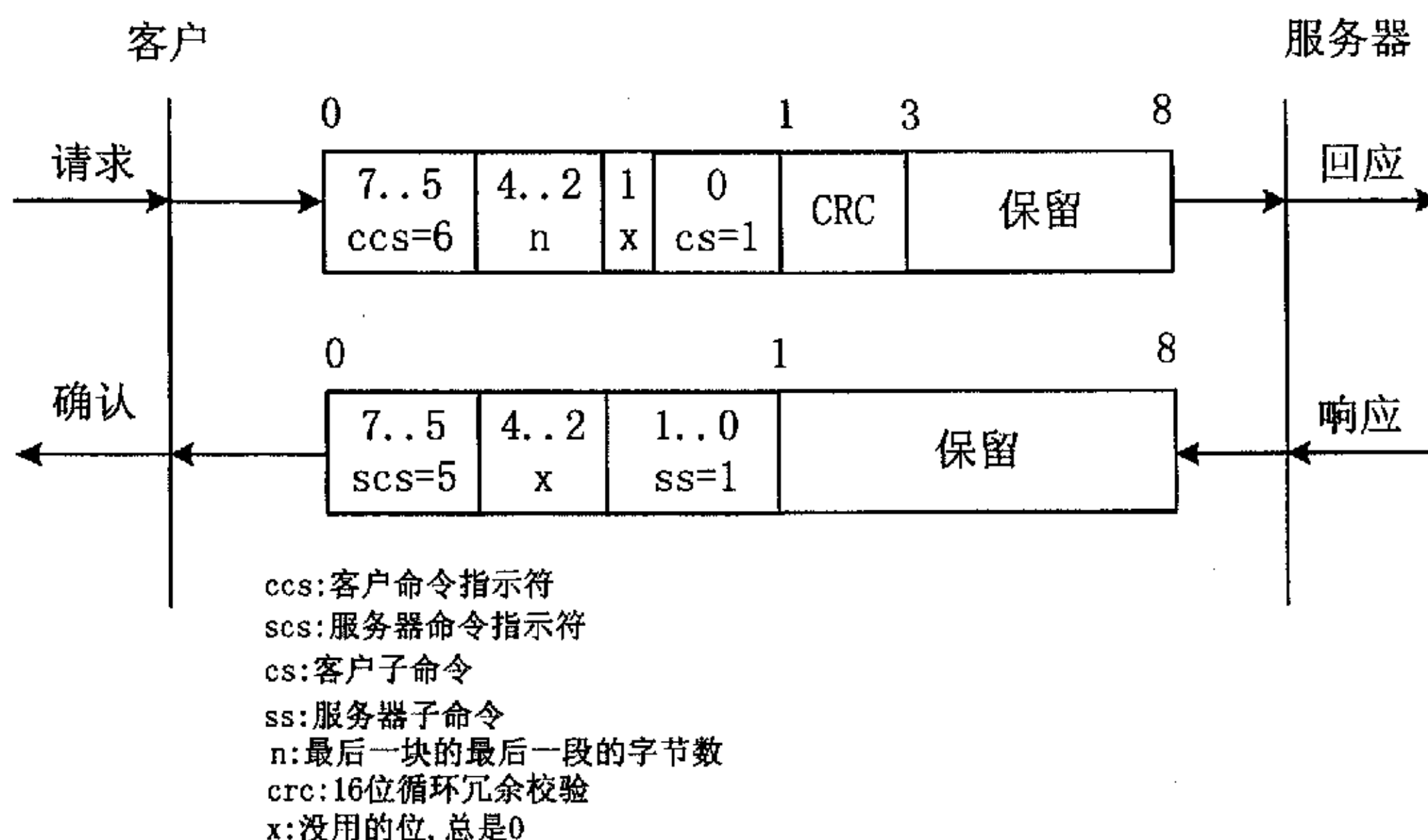


图 2-22 结束 SDO 块下载

为了保证 SDO 块传送的正确性，客户和服务端会计算 CRC，并且会在结束 SDO 块传送时交换 CRC。校验多项式公式为  $x^{16}+x^{15}+x^5+1$ 。在计算校验和之前，CRC 计算器必须清零。

#### 4. SDO 参数集

最后总结一下 SDO 的参数集，如下图所示。

服务器 SDO 参数 (22h)					
1200	RECORD	第 1 个服务器 SDO 参数	SDO 通讯参数	RO	0
1201	RECORD	第 2 个服务器 SDO 参数	SDO 通讯参数	RW	M/O
...	...	.....	.....	...	...
127F	RECORD	第 128 个服务器 SDO 参数	SDO 通讯参数	RW	M/O
客户 SDO 参数 (22h)					
1280	RECORD	第 1 个客户 SDO 参数	SDO 通讯参数	RW	M/O
1281	RECORD	第 2 个客户 SDO 参数	SDO 通讯参数	RW	M/O
.....	.....	.....	.....	...	.....
12FF	RECORD	第 128 个客户 SDO 映射	SDO 通讯参数	RW	M/O
1300		保留			
.....	.....	.....	.....	...	.....
13FF		保留			

表 2-13 SDO 参数集

SDO 通讯的参数格式在对象字典的 22h 里有定义。下面解释一下 SDO 的参数，如下表所示。

索引	子索引	内容	数据类型
1XXXh	0h	子索引元素数	Unsigned8
	1h	对象标识符(客户→服务器)	Unsigned32
	2h	对象标识符(服务器→客户)	Unsigned32
	3h	节点 ID	Unsigned8

表 2-14 SDO 参数说明

### 2.3.3 特殊功能对象

#### 1. 同步对象(同步对象)(Synchronization Object)

同步对象生产者向同步对象消费者提供了同步信号。当同步对象消费者接收到此信号后，它们进行它们的同步任务。通常，同步 PDO 报文的固定传输时间加上同步对象的周期性传输保证了传感器设备正常的采集过程信号，也保证了驱动设备按照固定的方式发出驱动信号。

另外，同步对象的标识符在对象字典的 1005h。同步对象的传输如下图所示。

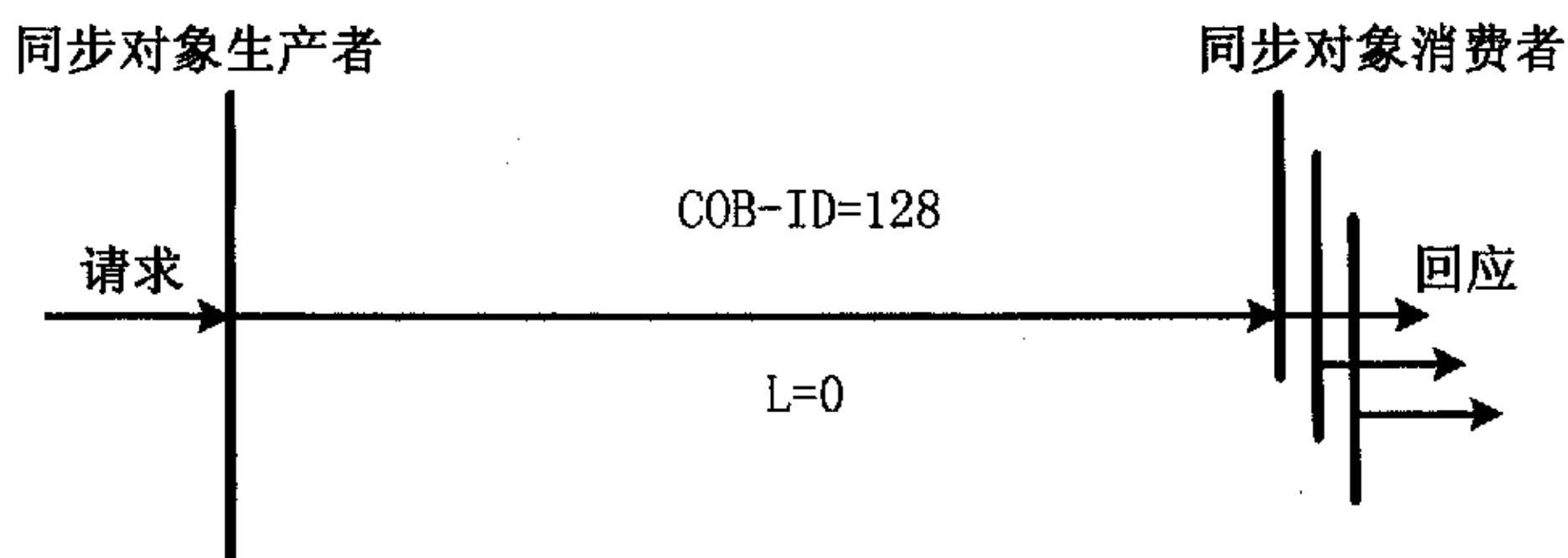


图 2-23 同步对象传输

为了保证及时地把同步对象发到总线上去，同步对象被赋予了很高的优先级标识符。CANopen 协议推荐用标识符 128，这是在预定义的优先级中几个最高中的一个。在同步对象中，不仅没有应用数据，而且，缺省情况下，不含有任何数据。

PDO 的同步传输意味着它的传输是相对于同步对象的传输是固定的。它在相对于同步对象的一个固定的时间窗内(同步窗)传输。如下图所示。

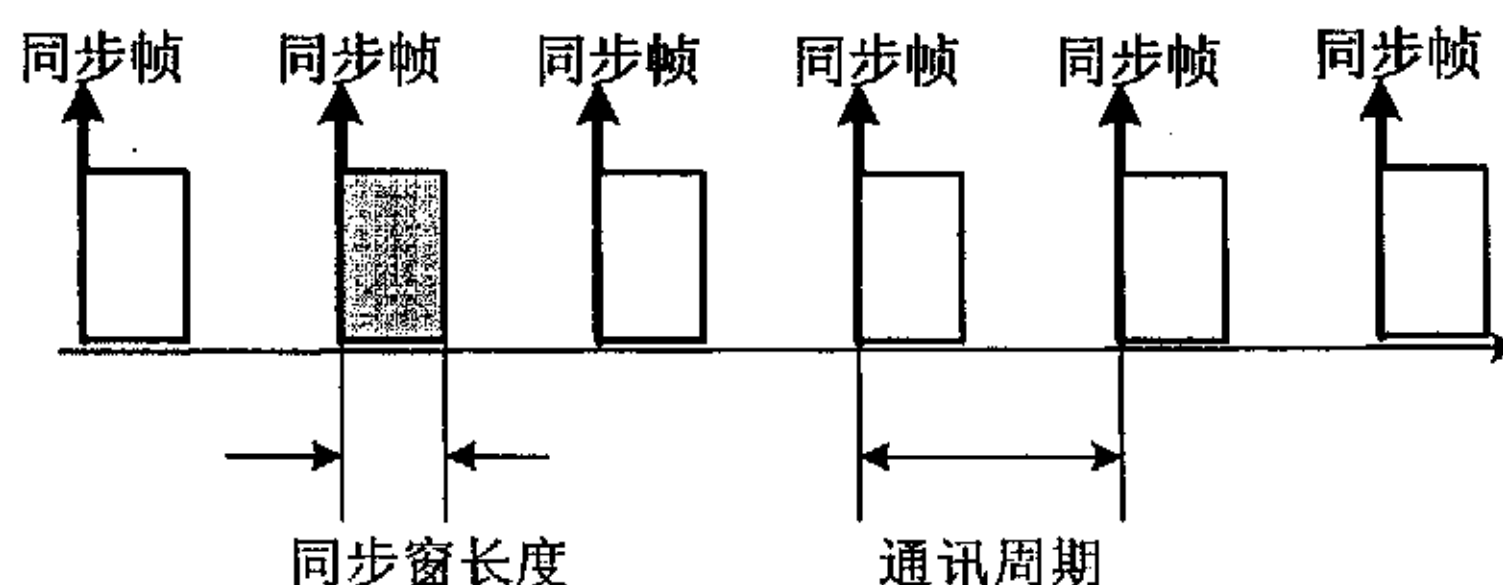


图 2-24 同步时间定义

两个同步对象之间的时间间隔在“通讯周期”参数中定义着。对象字典中的“同步窗长度”(1007h)和“通讯周期”(1006h)参数可以在设备启动过程中用配置工具来修改。另外一点，在同步对象生产者发送同步对象时候，可能出现时间抖动，这是由于在发同步对象之前可能正好有另外一个报文在发送。所以应该考虑这种情况-同步对象被发送了两遍。

循环发送的同步对象可以使支持这种功能的设备几乎同步地存储实际的输入值。然后在下一个时间窗内发送给其他设备。在下一个时间窗里，CANopen 设备把输出值传送给驱动设备，但是，直到下一个同步帧到来时才有效。如下图所示。

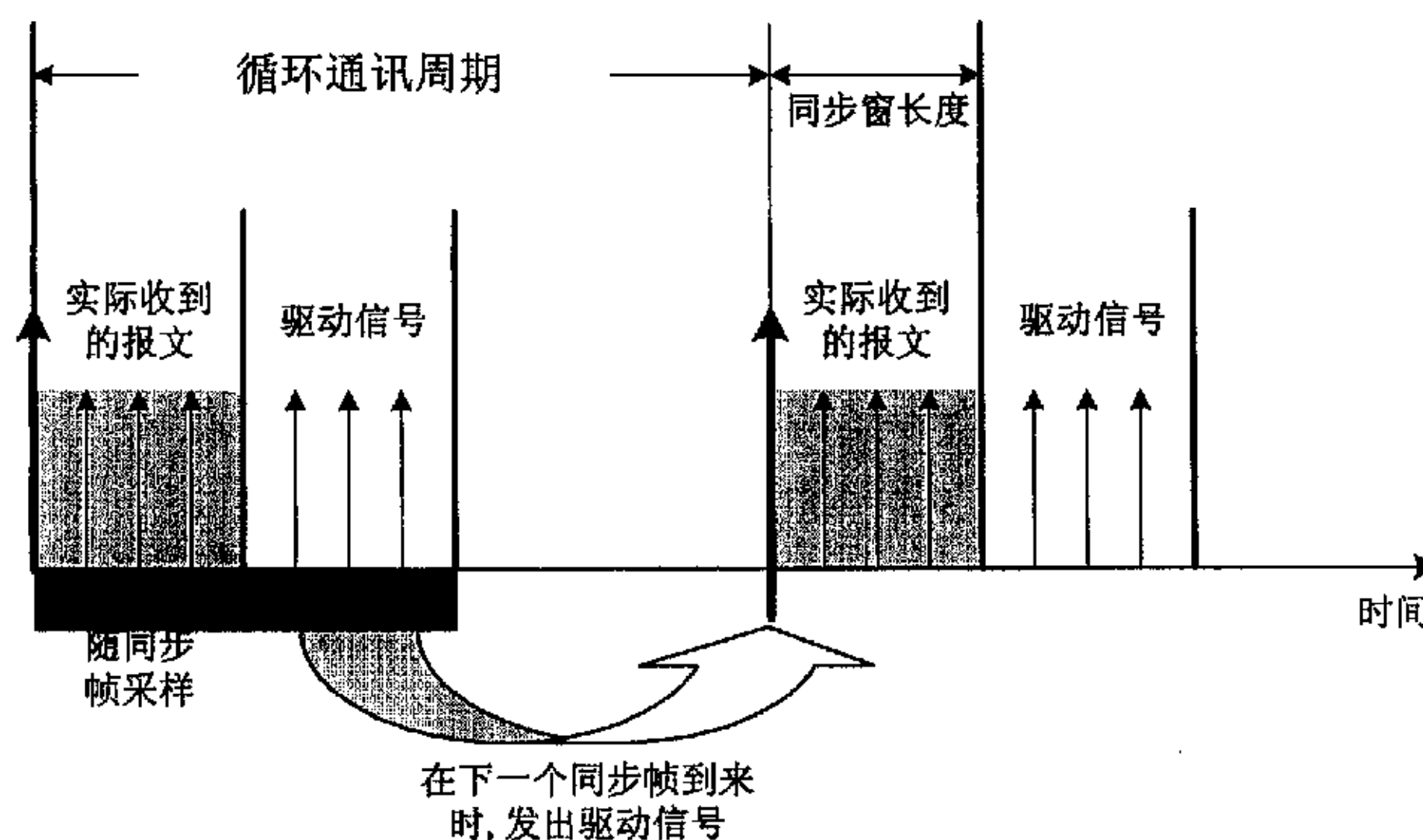


图 2-25 同步性操作

## 2. 时间戳对象(Time-Stamp)

通常，时间戳对象表示从 1984 年 1 月 1 号后的天数和午夜过后的毫秒数，一共是 6 个字节。

有时，特别在一些大型网络里，网络的传输速度比较慢，那些对时间要求苛刻的设备就要求非常准的时间同步，有可能需要把各自的时间同步在毫秒级。这就可以用这种高精度的同步协议来达到这种要求，用特殊的时间戳来调整不可避

免的时间漂移。这种高精度的时间戳是一个无符号 32 位数，精度为 1 毫秒，这意味着时间计数器每 72 分钟重新计数一回。这个高精度时间戳在对象字典的 1013h，发送时映射到一个 PDO 里。

同步时间戳的传输过程如下图所示。



图 2-26 时间戳协议

时间戳对象传输时，发送设备作为生产者，接收设备作为消费者。另外，对于时间戳对象，建议用标识符 256。

### 3. 应急对象(Emergency Object)

应急对象一般是当一个设备发生内部严重错误的情况下被触发的，用高优先级从一个设备发送到其他的设备。应急对象适用于中断类型错误警告。需要注意的是，一个应急报文只需每个错误发送一次，就是说应急报文一定不要重复。只要没有新的错误发生，就应该再没有应急报文发送了。

通过 CANopen 的通讯描述定义好的应急错误代码，错误寄存器和额外的和特定设备有关的信息都在“设备描述”里有说明。

应急报文的传输如下图所示。

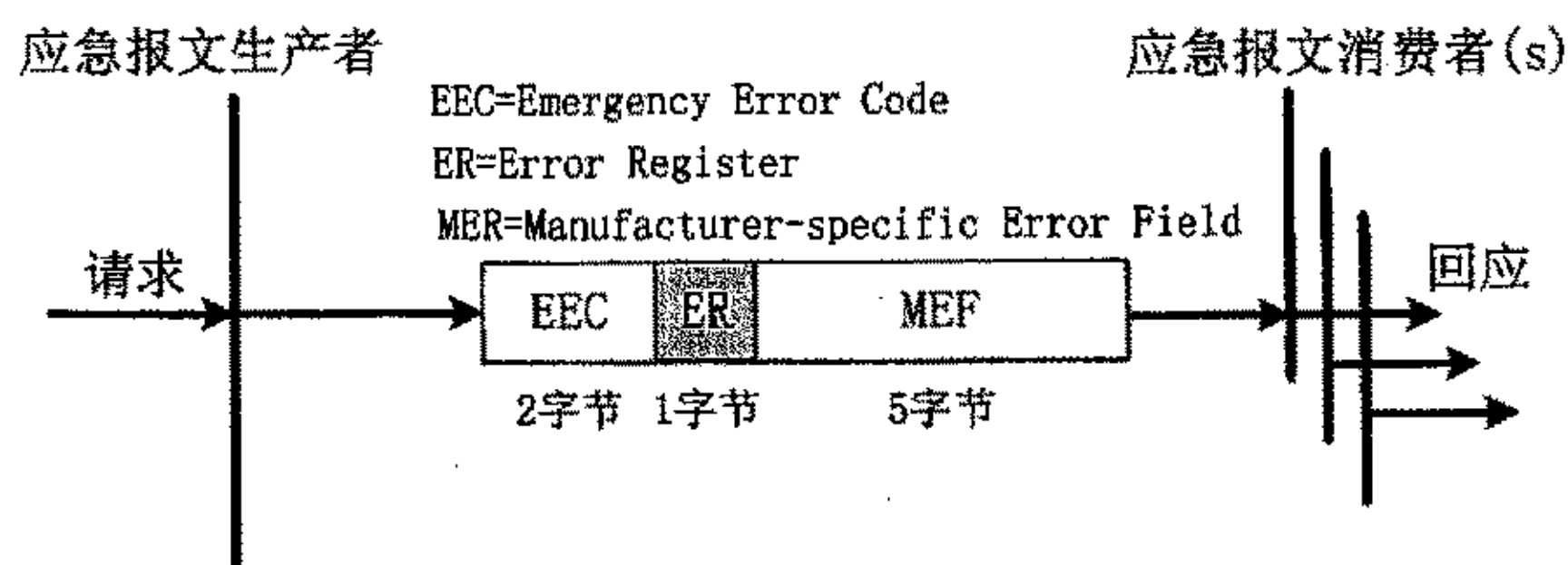


图 2-27 应急报文协议

应急对象是可选的。如果一个设备支持应急对象，它就得支持至少两个错误代码 00xx(error reset 或者没有错误)和 11xx(generic error)。此外，从上图可以看出，应急对象包括 8 个数据位。

### 2.3.4 网络管理(Network Management-NMT)



CANopen 网络管理是面向节点的，并且实施的主从通讯方式。下面让我们先了解一下主从通讯方式。

### 1. 主从通讯方式

主从方式只允许有主节点发起的通讯，从节点永远等待主节点的请求。这就像在军队一样，士兵服从命令，只有在得到允许后才可以说话。如下图所示。

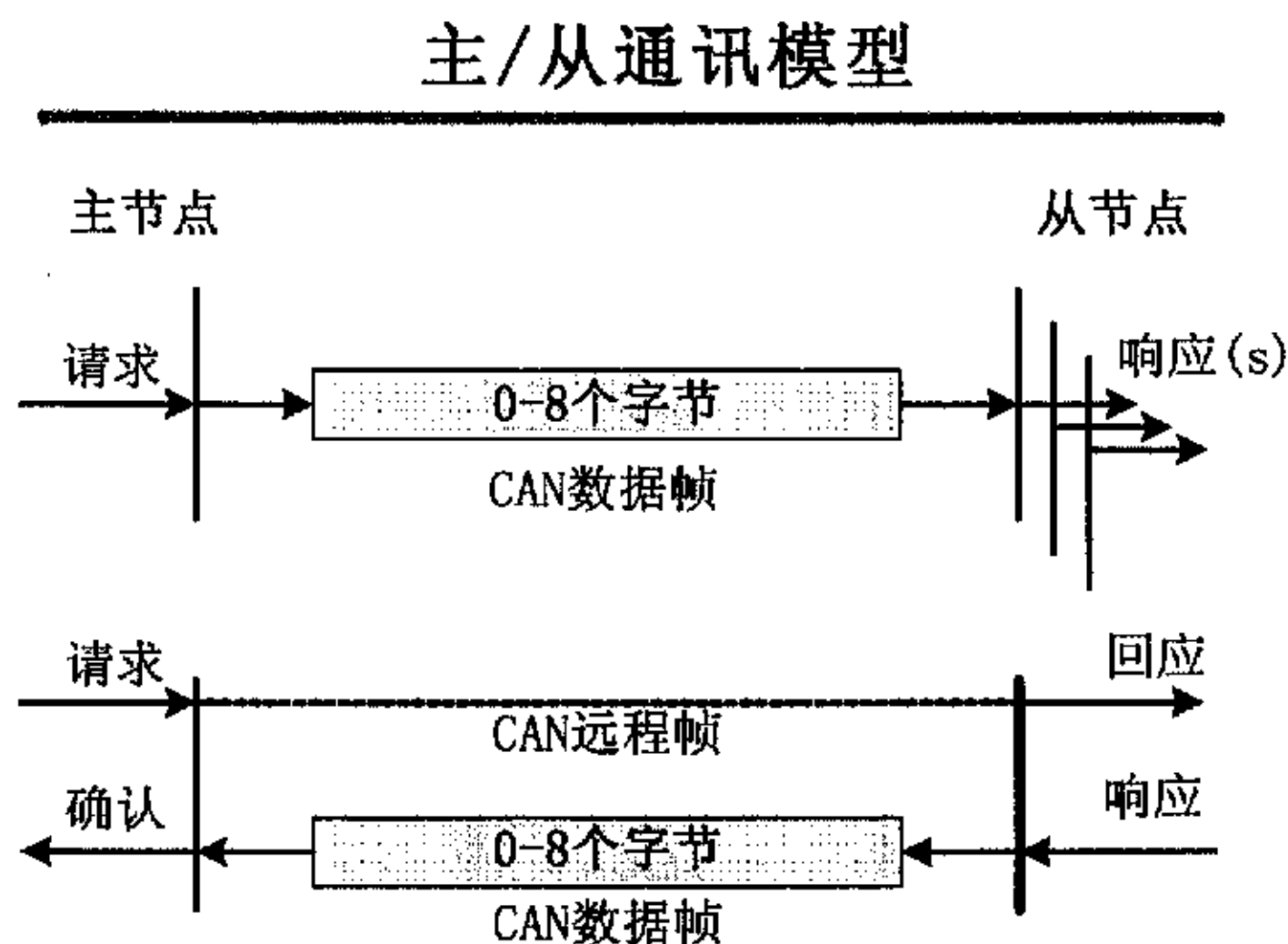


图 2-28 主从通讯方式

在基于 CAN 的网络通讯中，主从通讯可以用特定的标识符分配来实现。此外，非确认方式的主从通讯也允许广播。

### 2. 网络管理的主从通讯方式

在 CANopen 通讯中，需要一个节点履行网络管理(NMT)的主节点功能。其他节点作为 NMT 从节点。如下图所示。每个 NMT 从节点以节点 ID 来标识。

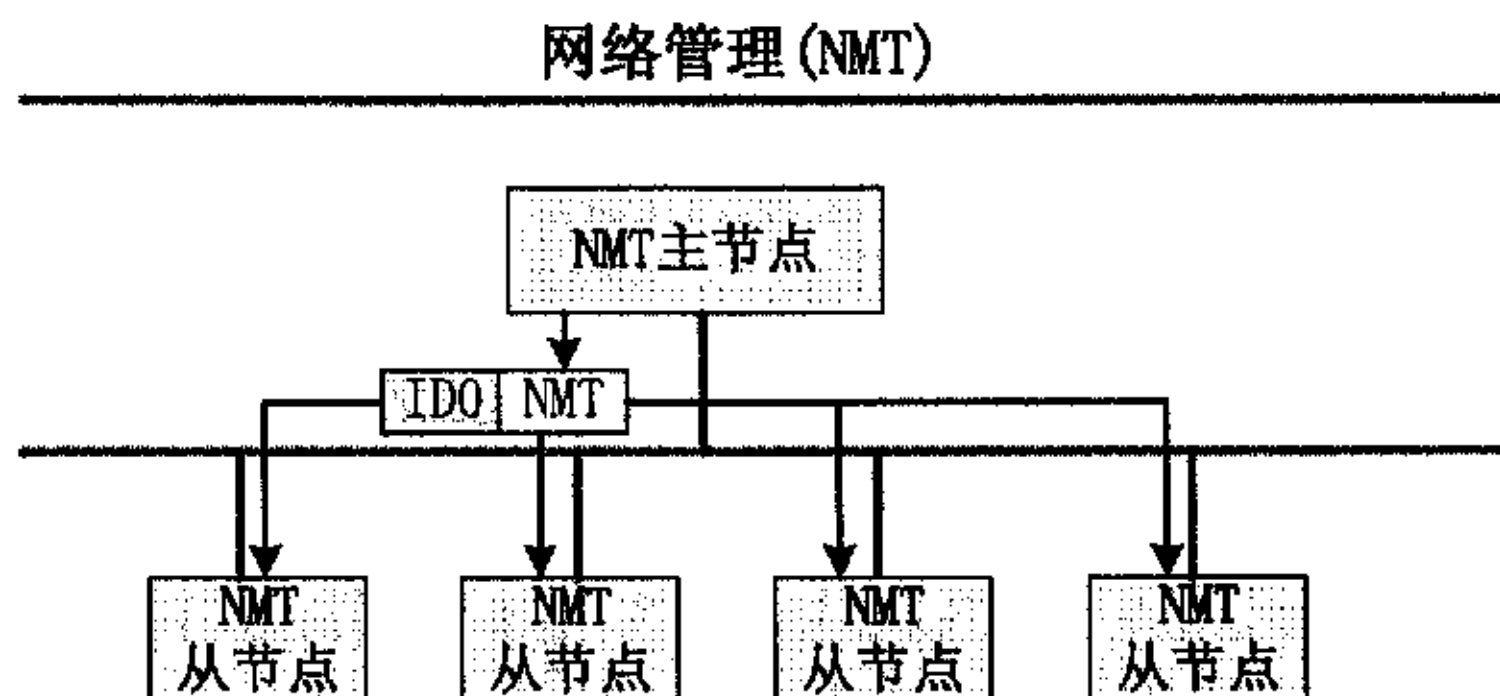


图 2-29 网络管理

网络管理提供了以下功能:模块控制服务(初始化那些想参加网络通讯的 NMT

从节点);错误控制服务(管理节点和网络通讯的状态);配置控制服务(上载和下载配置数据)

### 3. 状态管理

#### 1) 从结点状态转换

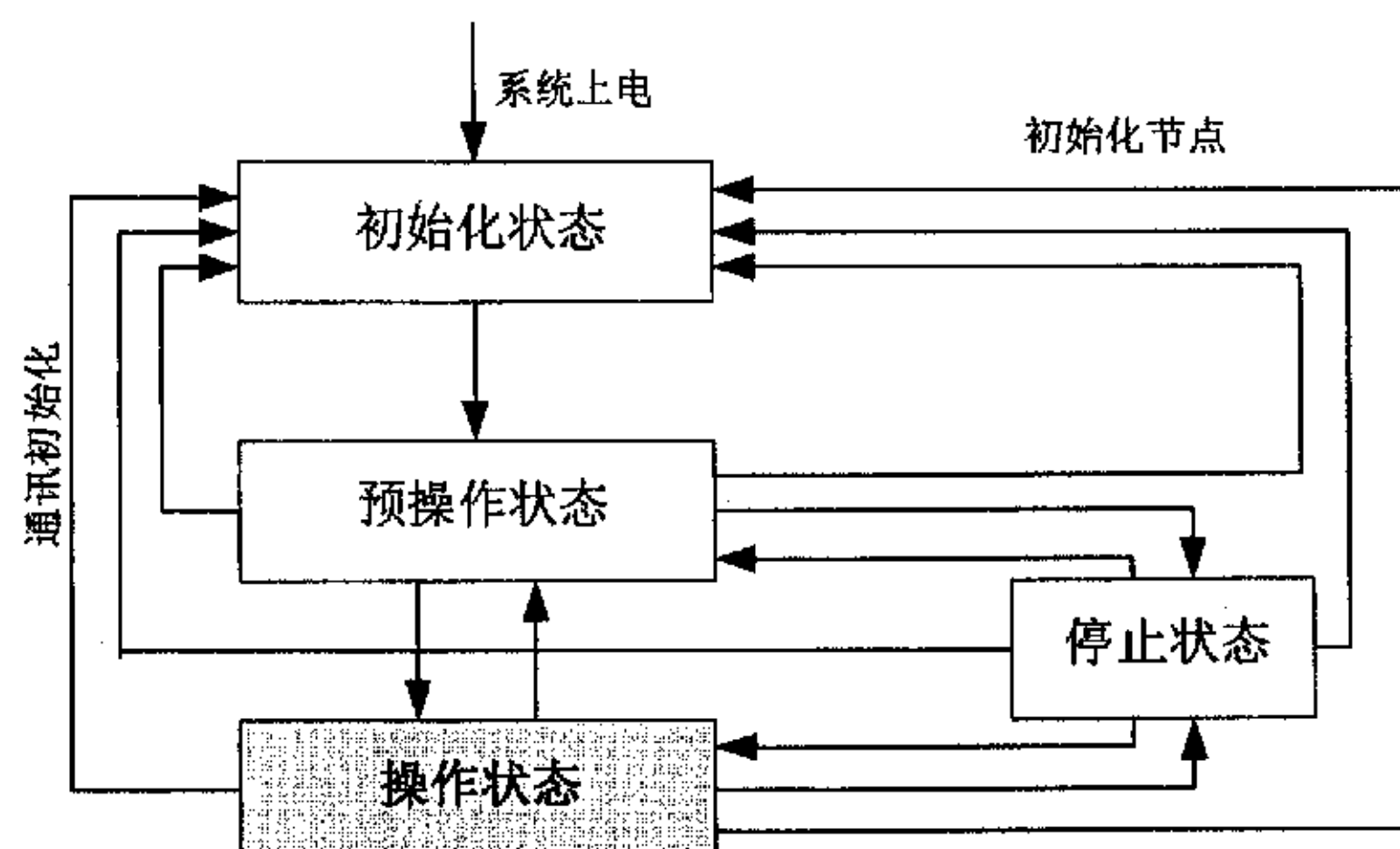


图 2-30 从结点的状态转换

每个 CANopen 的 NMT 的从节点上电和内部初始化之后自动进入预操作状态(Pre-operational state)。在这个状态里，从节点可以通过 SDO 被配置和设置参数。不允许 PDO 通讯。

NMT 主节点可以使所有的其他节点进入操作状态(Operational state)，当然也可以退出。在操作状态里，允许 PDO 通讯。主节点还可以使从节点进入停止状态(Stopped State)，从而停止 PDO 和 SDO 通讯，这个状态可以使从节点专心做自己的事情，具体的事情在设备描述里有详细定义。

在操作状态里(Operational state)，所有的通讯对象都是可通讯的，用 SDO 读写对象字典也是可以。

需要着重说一下的是初始化状态(Initialization state)，为了可以完全或者部分的初始化节点，初始化状态被分成三种子状态。在“应用初始化子状态”(Rest Application sub-state)里，生产商特定的参数和标准的设备描述参数被设置成缺省值。然后，自动进入通讯初始化子状态(Reset Communication sub-state)。在这个子状态里，通讯描述参数被设置成上电值(power on)。之后，进入“Initializing sub-state”，在这个状态里，执行基本的设备初始化。在进入预操作状态之前，发送标准的启动对象(Boot-up object)。

初始化值是上次存储的值。如果不支持参数存储或者如果初始化之前执行了“恢复缺省值命令”(对象 1011h)，上电值就是按照通讯描述和设备描述的缺省值。

#### 2) NMT 通信对象

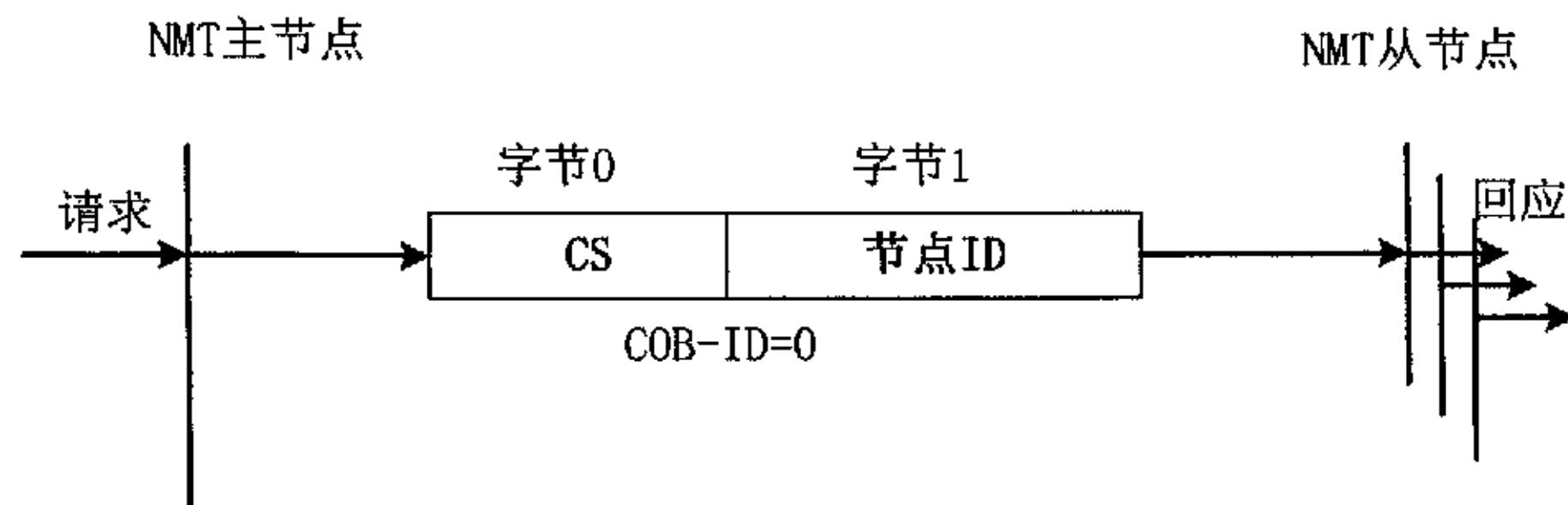


图 2-31 NMT 通信对象

CANopen 网络管理提供了以下五种服务，用命令标识(CS)来区分：

- ✧ 启动远程节点 (CS=1)
- ✧ 停止远程节点 (CS=2)
- ✧ 进入预操作状态 (CS=128)
- ✧ 初始化节点 (CS=129)
- ✧ 通讯初始化 (CS=130)

NMT 通讯对象标识符为 0，由两个字节组成。节点 ID 表示报文的目的地。如果节点 ID 为 0，则目的地是所有 NMT 从结点。

#### 4. 节点保护

为了知道某个不发送 PDO 的节点已脱离总线，NMT 主节点可以管理一个数据库，在这个数据库里除了其他信息，所有连接节点的状态在里面保存着，我们称之为节点保护。利用循环的节点保护，NMT 主节点就可以经常地管理 NMT 从节点。

为了检测 NMT 主节点的存在，从节点也检测是否节点保护是否在规定的时间内发生了。如下图所示。

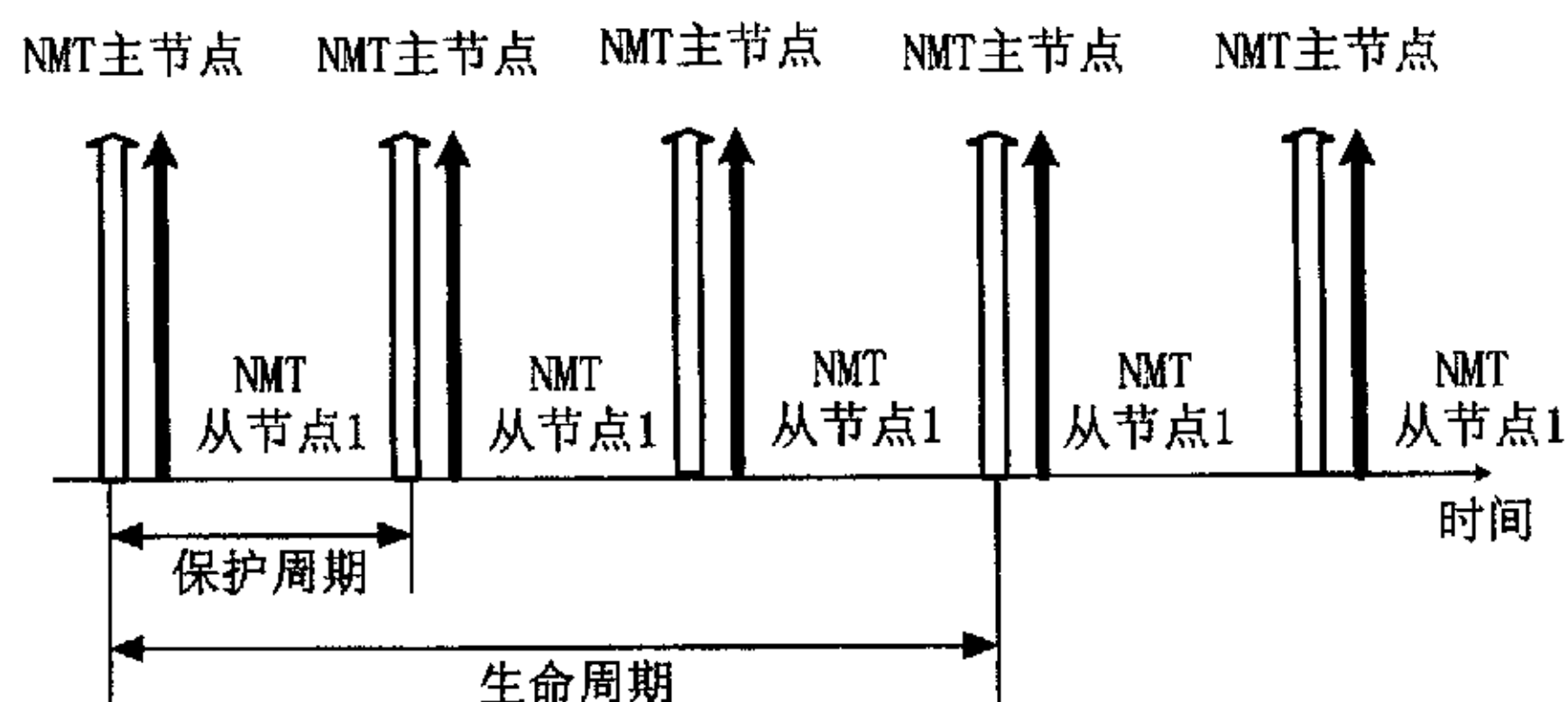


图 2-32 节点保护

NMT 主节点在从结点的预操作状态里通过发送远程帧来启动节点保护。如下图所示。节点保护在停止模式下依然有效。

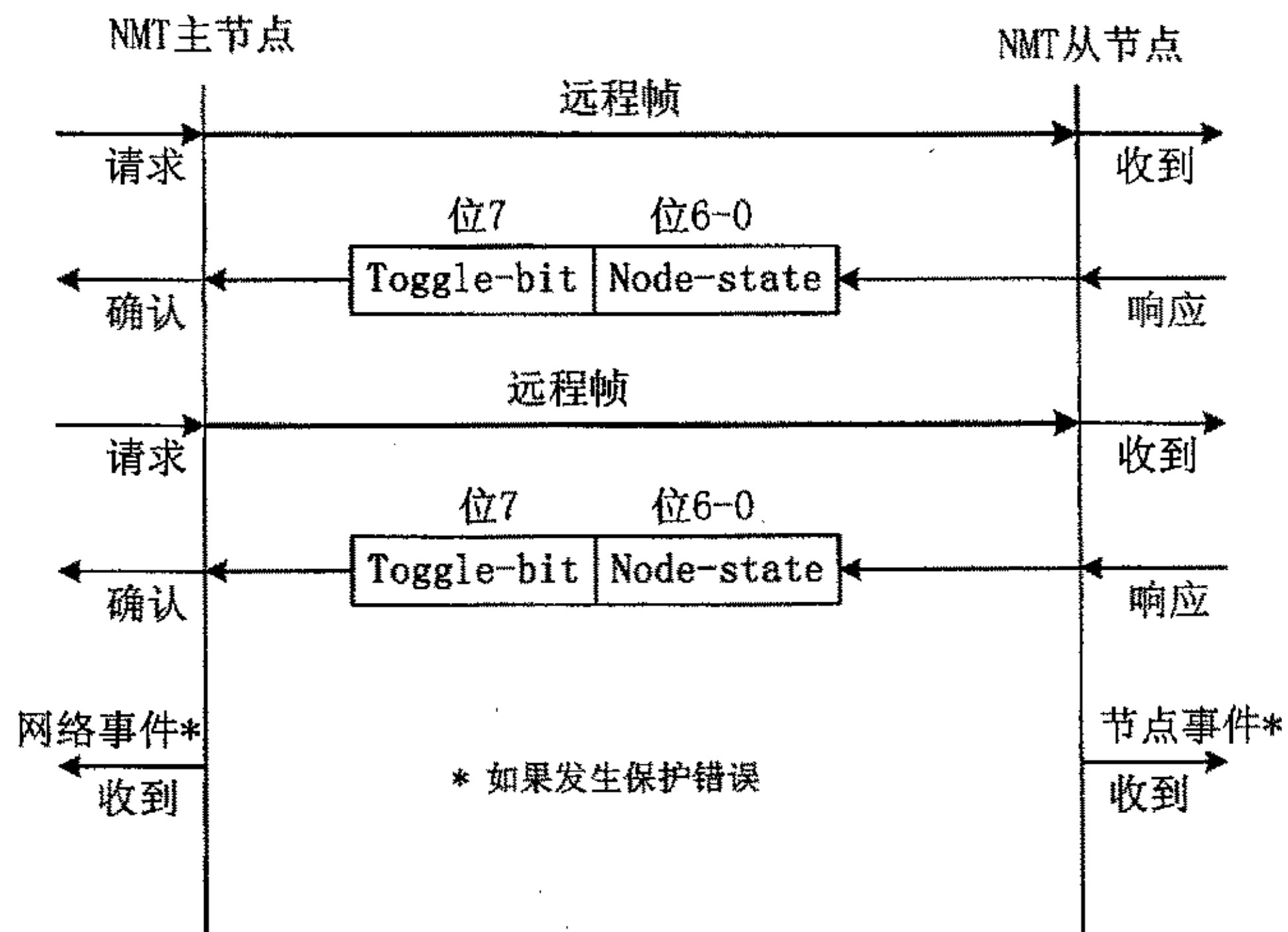


图 2-33 节点保护过程

NMT 主节点经常地通过发送远程帧来取得所有节点的实际状态，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。主节点把他们和数据库中存储的状态相比较。若从站没有在规定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。

### 5. heartbeat 对象

可选的 heartbeat 协议是用来建议代替节点保护协议的。heartbeat 对象的传输过程如下图所示。



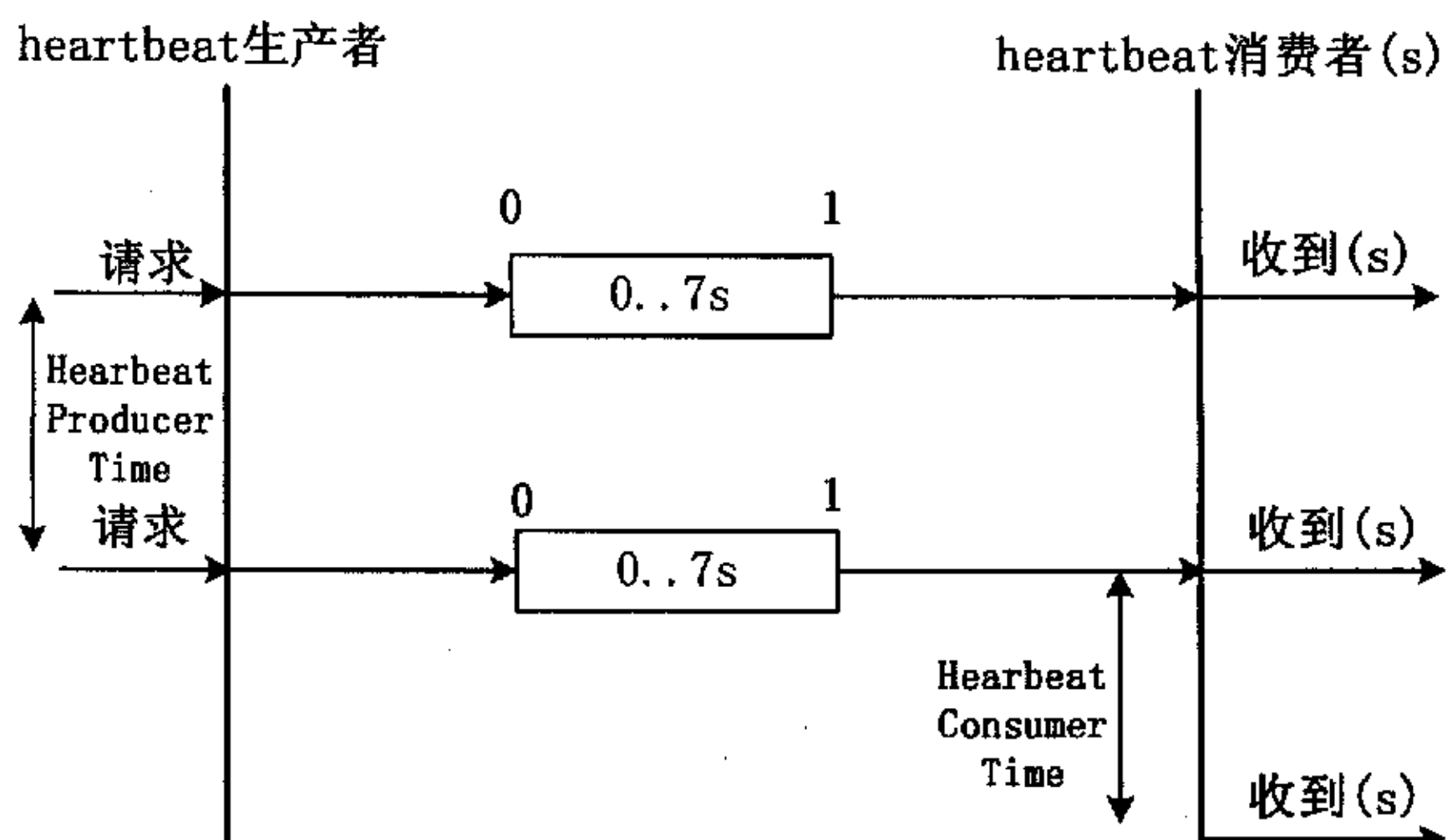


图 2-34 heartbeat 对象

一个 Heartbeat 生产者按照“Heartbeat producer time”对象定义好的频率周期性地发送 heartbeat 报文。一个或多个 heartbeat 消费者会收到。heartbeat 生产者和消费者的关系是通过对象字典配置好的。heartbeat 消费者在“heartbeat consumer time”时间内监视 heartbeat 的到来，如果在这段时间没有收到，就会触发一个 heartbeat 事件(heartbeat event)。

### 2.3.5 缺省的标识符分配方案

为了在简单的网络中减少配置的麻烦，CANopen 预定义了一个缺省的标识符分配方案。这些预定义的标识符在初始化后会在预操作状态有效。

缺省的标识符分配方案由一个功能码(function code)和一个节点 ID 组成。如下图所示。

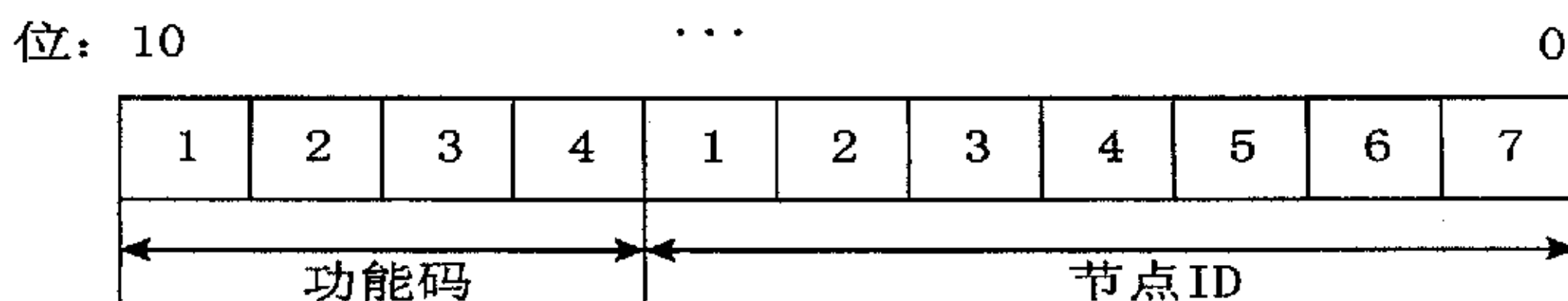


图 2-35 缺省的标识符分配方案

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

时间戳	0010	256(100h)	1012H, 1013H
应急对象	0001	129 (81h)-255 (FFh)	1014H, 1015H
PDO1 (tx)	0011	385(181h)-511(1FFh)	1800H
PDO1 (rx)	0100	513(201h)-639(639h)	1400H
PDO2 (tx)	0101	641(281h)-767(2FFh)	1801H
PDO2 (rx)	0110	769(301h)-895(37Fh)	1401H
PDO3 (tx)	0111	897(381h)-1023(3FFh)	1802H
PDO3 (rx)	1000	1025(401h)-1151(47Fh)	1402H
PDO4 (tx)	1001	1153(481h)-1279(4FFh)	1803H
PDO4 (rx)	1010	1281(501h)-1407(57Fh)	1403
SDO (tx)	1011	1409(581h)-1535(5FFh)	1200H
SDO (rx)	1100	1537(601h)-1663(67Fh)	1200H
NMT Error Control	1110	1793(701h)-1919(77Fh)	1016h, 1017h

默认的 ID 分配方案允许一个主节点最多和 127 个从节点之间点到点的通讯。也支持非确认的 NMT 服务、同步对象、时间戳以及节点保护对象的广播通讯。

## 第三章 模块化 CAN 总线控制器的硬件设计

### 3.1 CAN 总线控制器的总体结构

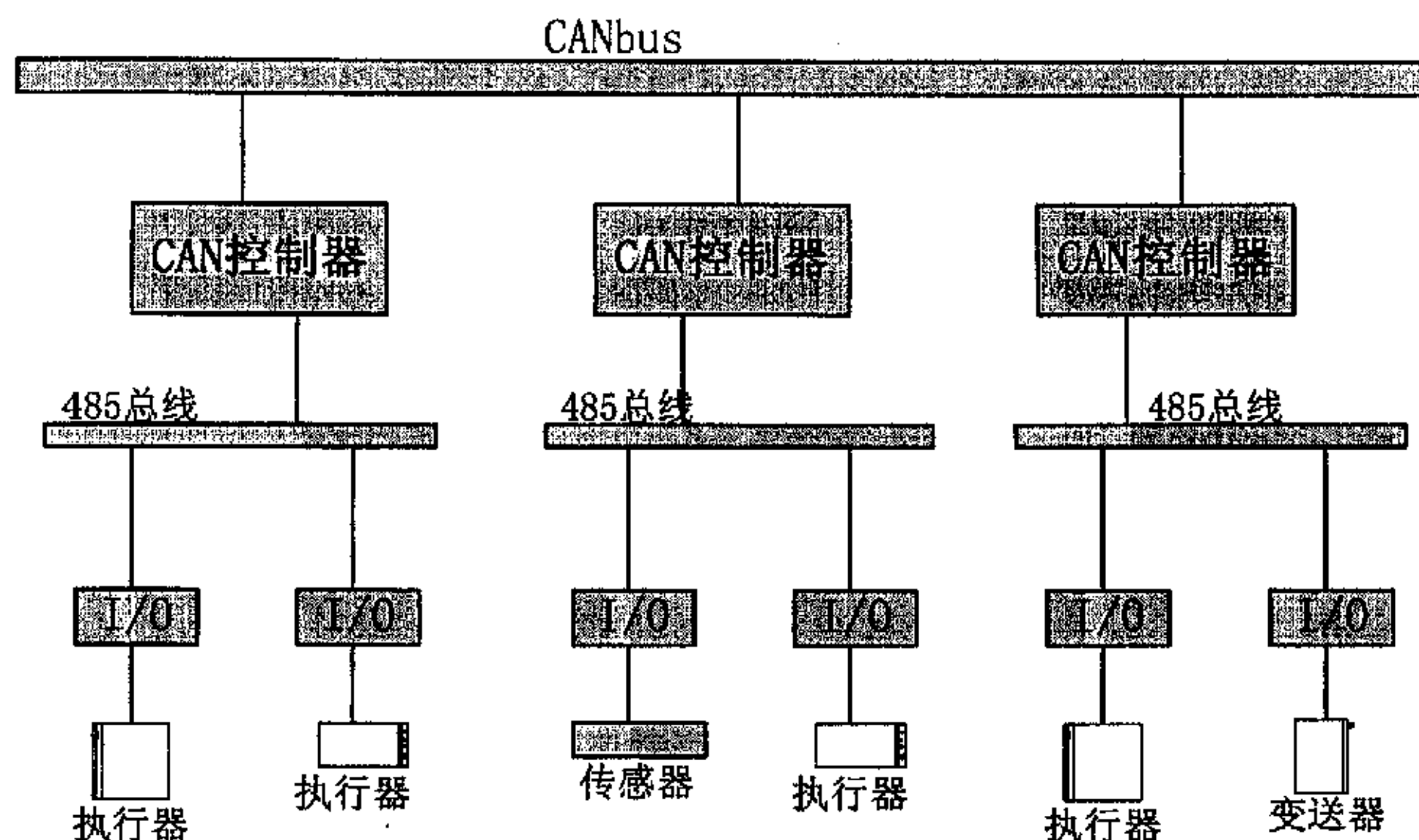


图 3-1 CAN 总线控制系统结构简图

如图 3-1，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），各 CAN 节点之间相互通讯。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。本文所设计的部分包括 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。

### 3.2 CAN 节点的总体设计思想

#### 3.2.1 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) 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。

#### 3.2.2 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。

其原理如图 3-2 所示：

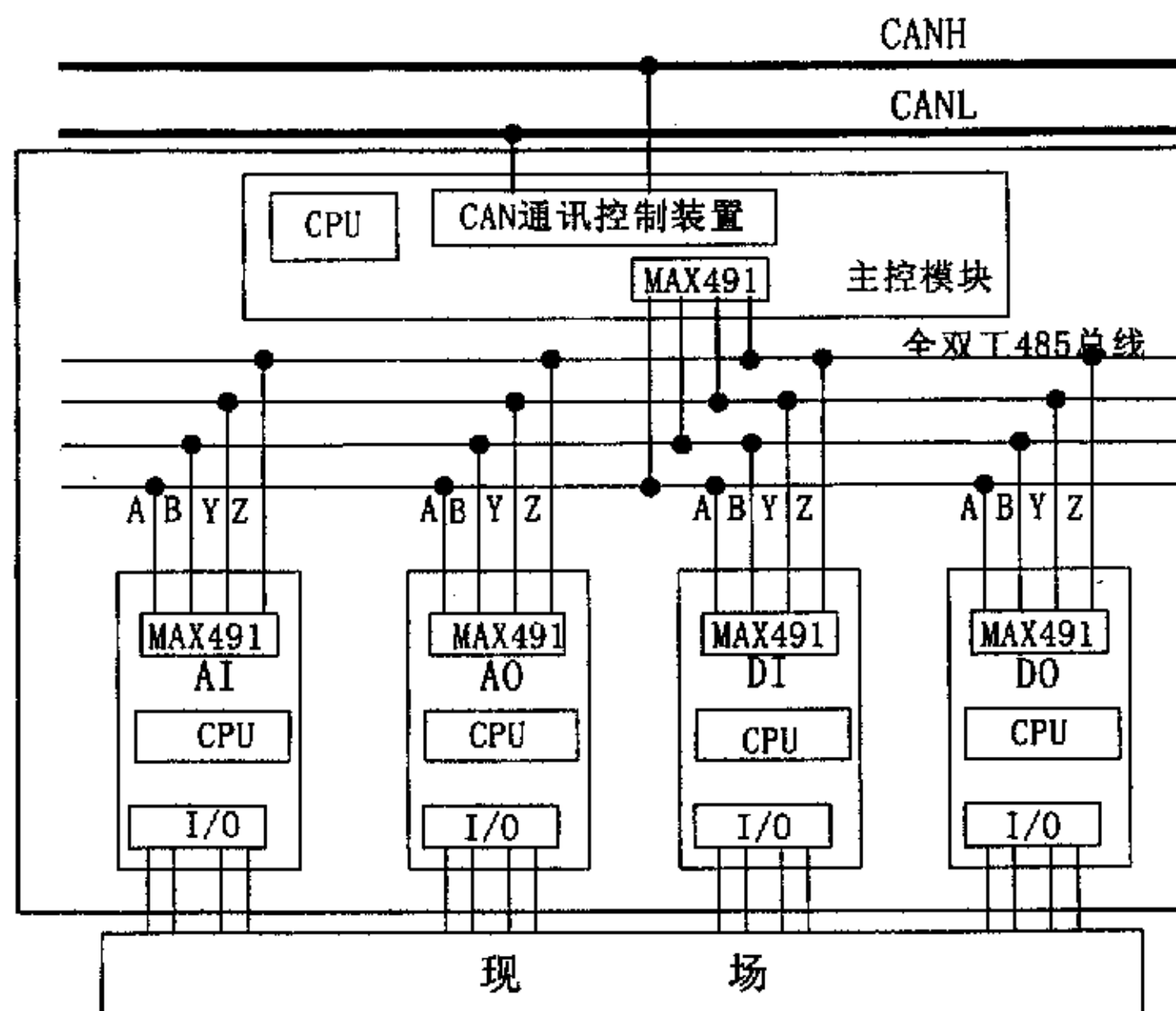


图 3-2 组合式 CAN 节点的总体设计框图

### 3.3 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。

#### 3.3.1 节点控制模块的设计

节点控制模块的硬件结构图如下：

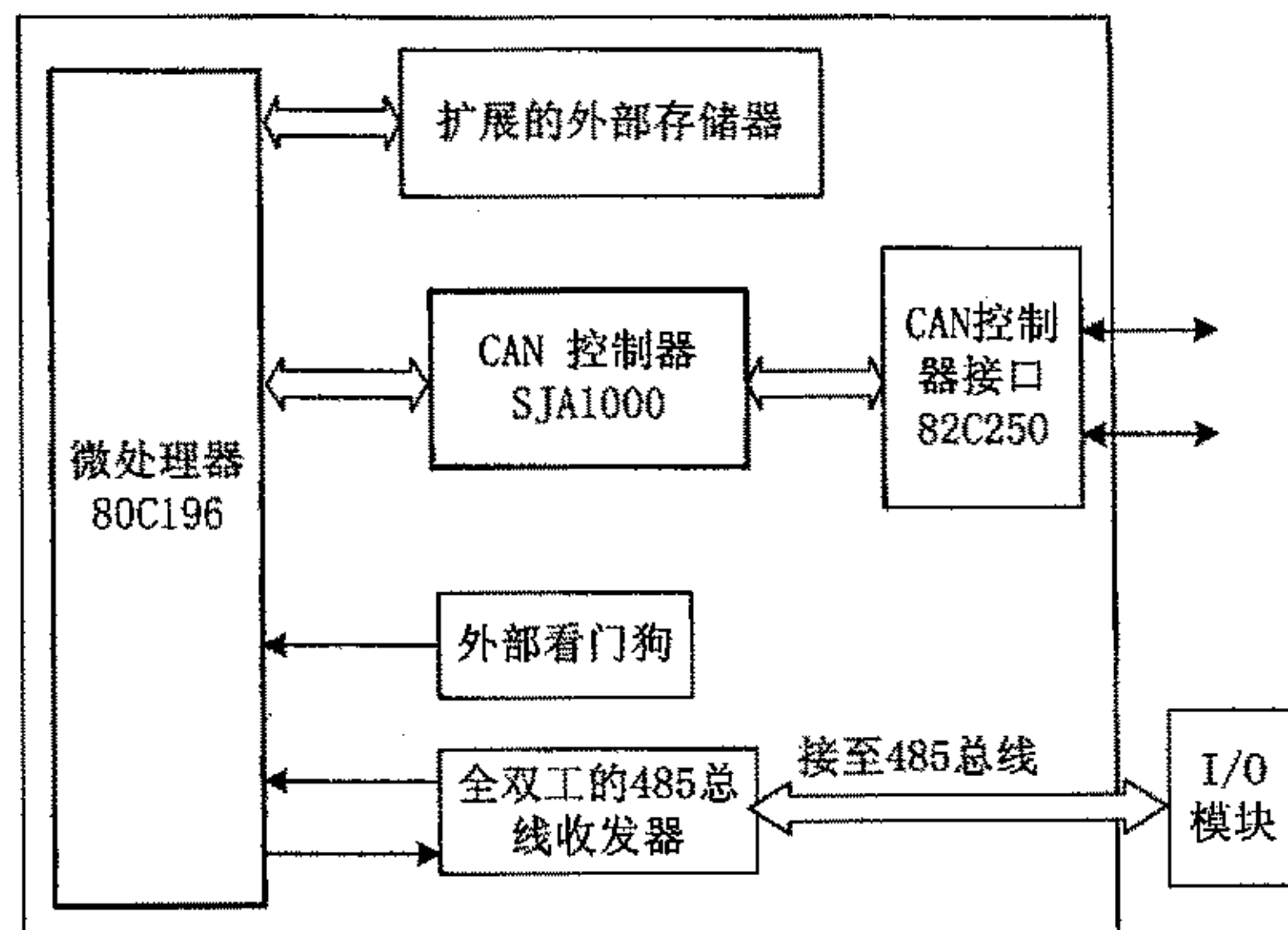


图 3-3 CAN 总线控制器节点控制模块的硬件结构图

## 1. 芯片的选择及外围芯片的设计

### 1) CPU 的选择<sup>[9][10][11]</sup>

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。

CPU 各引脚的分配如下：

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。

### 2) 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  $64\text{K} \times 8$  的 TMS28F512。

数据存储器采用  $32\text{K} \times 8$  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。外部存储器的扩展属于常规设计。

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  $64\text{K}$ 。本设计中选用了  $64\text{K}$  的外部程序存储器和  $32\text{K}$  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  $32\text{K}$  的地址空间。本文采用了页面技术，把  $128\text{KB}$  的存储空间分为 4 页，每页为  $32\text{KB}$ 。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 ( $0000\text{H} \sim 7\text{FFFH}$ ) 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  $0000\text{H} \sim 7\text{FFFH}$ ，其它的物理存储器（ $32\text{KB}$  的高位程序存储器、 $32\text{KB}$  的数据存储器、为 SJA1000 分配的存储空间）都映射到  $8000\text{H} \sim \text{FFFFH}$  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。

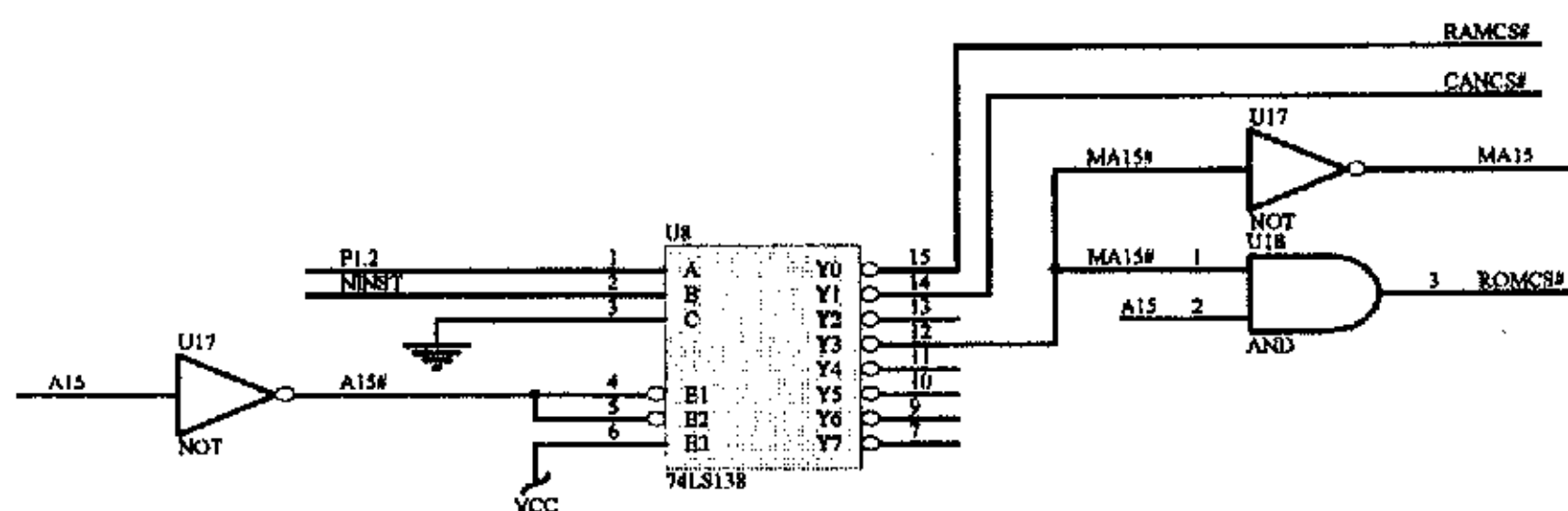


图 3-4 芯片片选电路的设计

如图 3-4，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	P1.2	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) 外部看门狗的设计

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是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:

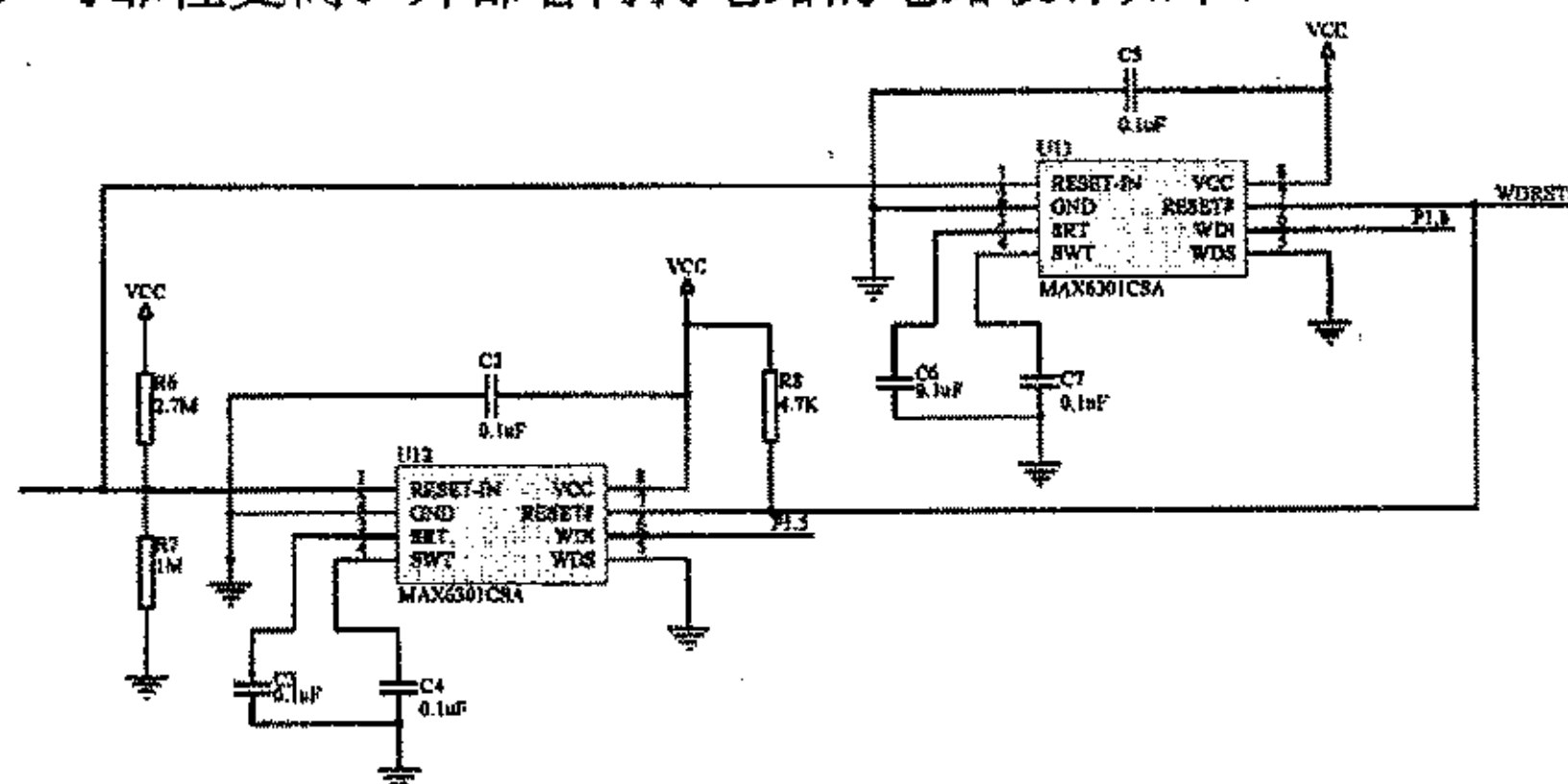


图 3-4 外部看门狗电路的设计

## 3.3.2 CAN 总线通讯电路的设计

### 1. 芯片的选择与介绍<sup>[11]</sup>

本文中选用 CAN 通讯控制器 SJA1000、CAN 控制器接口 82C250 和高速光耦等芯片,主要芯片介绍如下。

#### 1) CAN 通讯控制器 SJA1000

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:

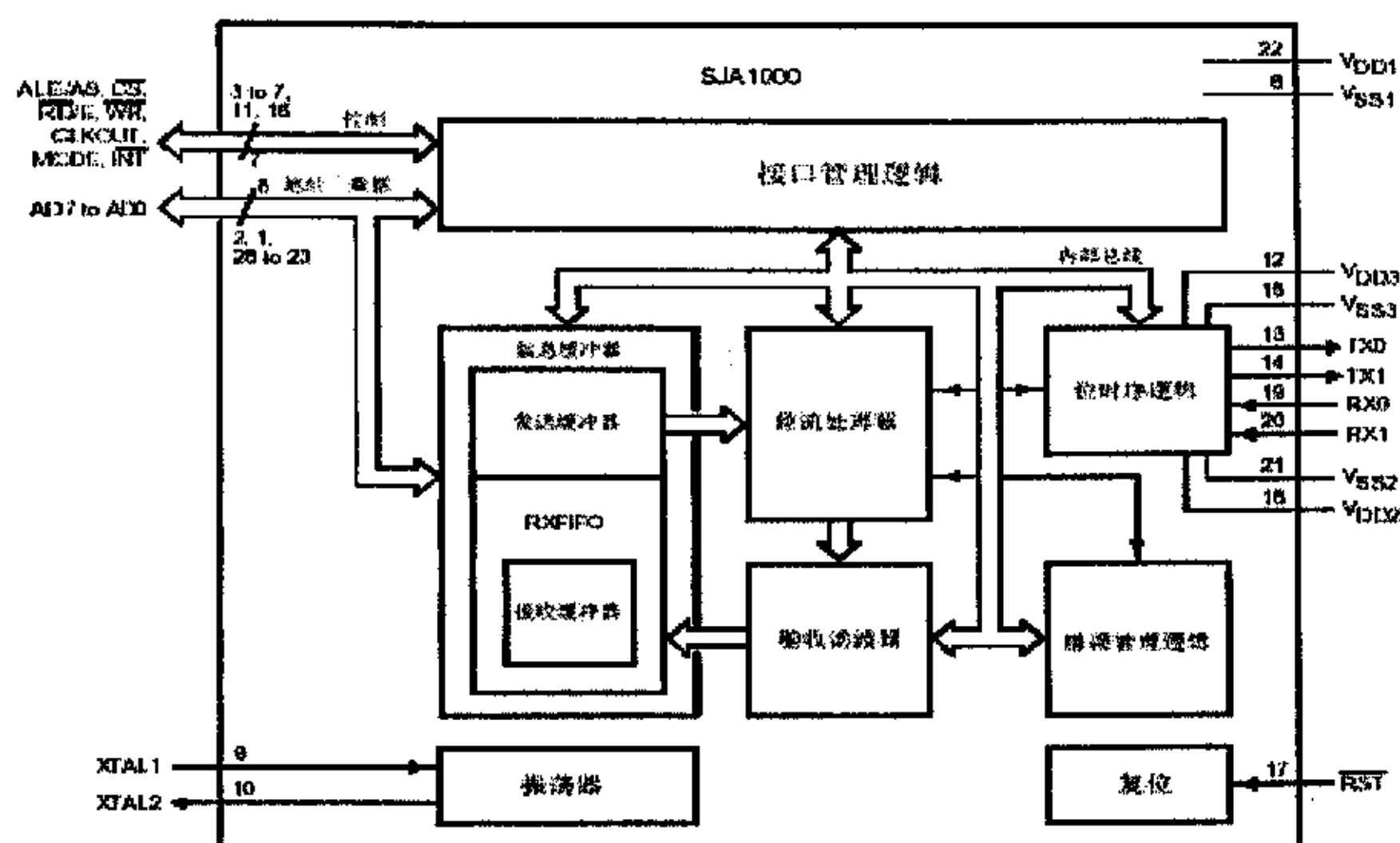


图 3-5: SJA1000 的结构方框图

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。

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 1Mbps/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;

2) CAN 控制器接口芯片 82C250<sup>[3]</sup>

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。

## 2. CAN 总线通讯的电路设计<sup>[12][13][14]</sup>

### 1) CAN 总线通讯电路, 如下图。

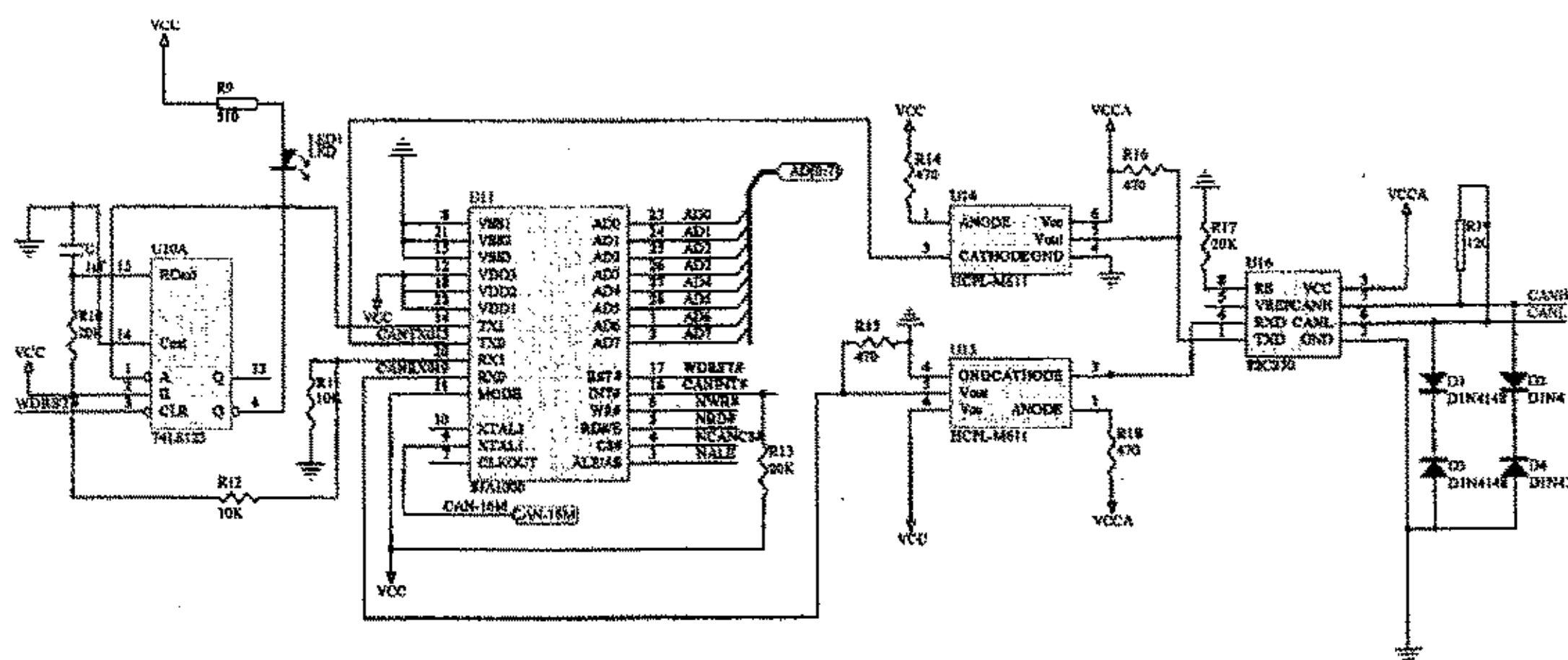


图 3-6 CAN 总线通讯的电路设计

### 2) CAN 总线通讯的原理<sup>[2][3][11]</sup>

在 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ACRn和AMRn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ACRn和AMRn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3-7所示：

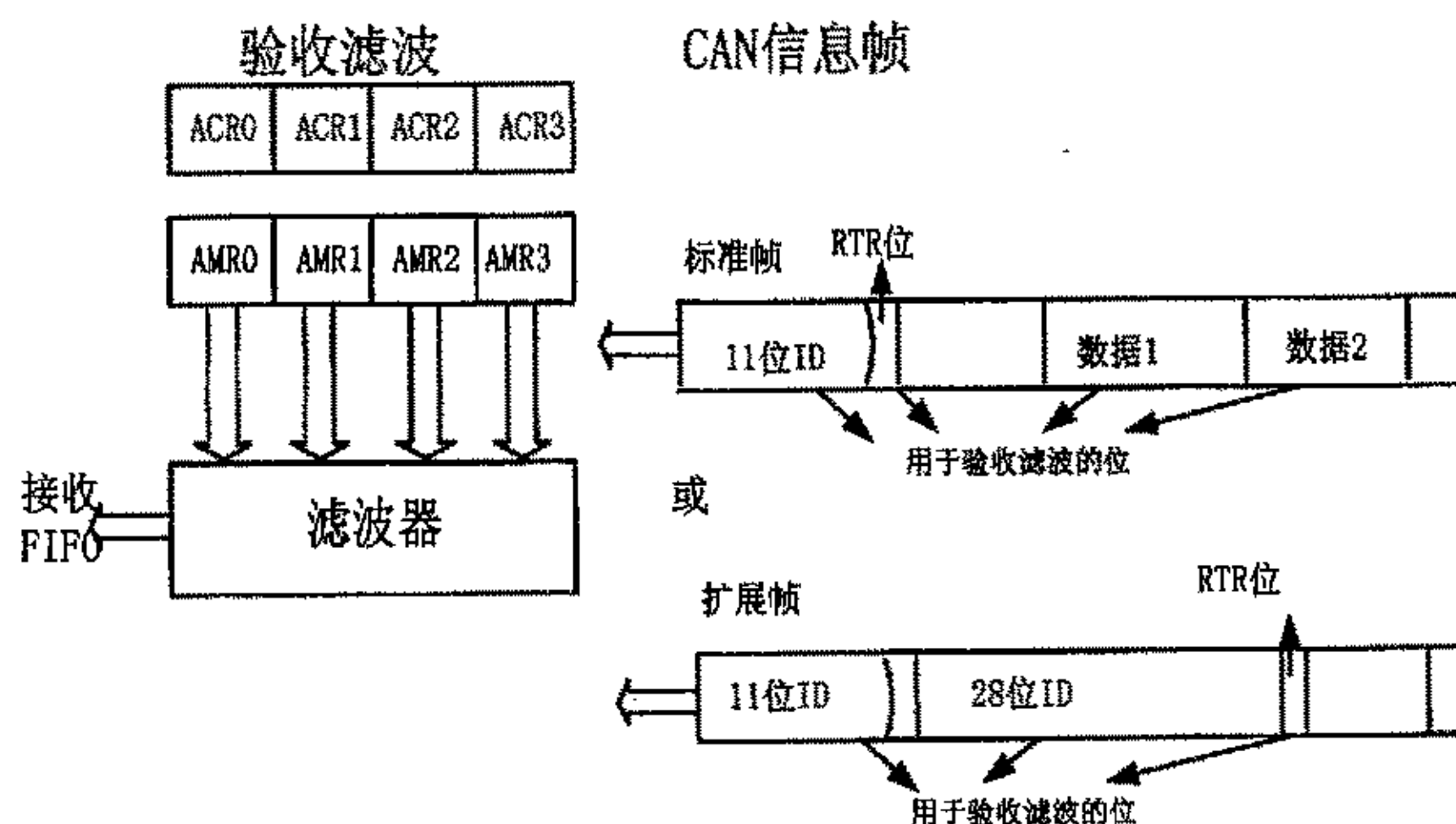


图 3-7 在 Pelican 模式下的单滤波的验收滤波

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。

### 3.3.3 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。

#### 1. 485 通讯原理及电路设计<sup>[15]</sup>

如图 3-8 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。

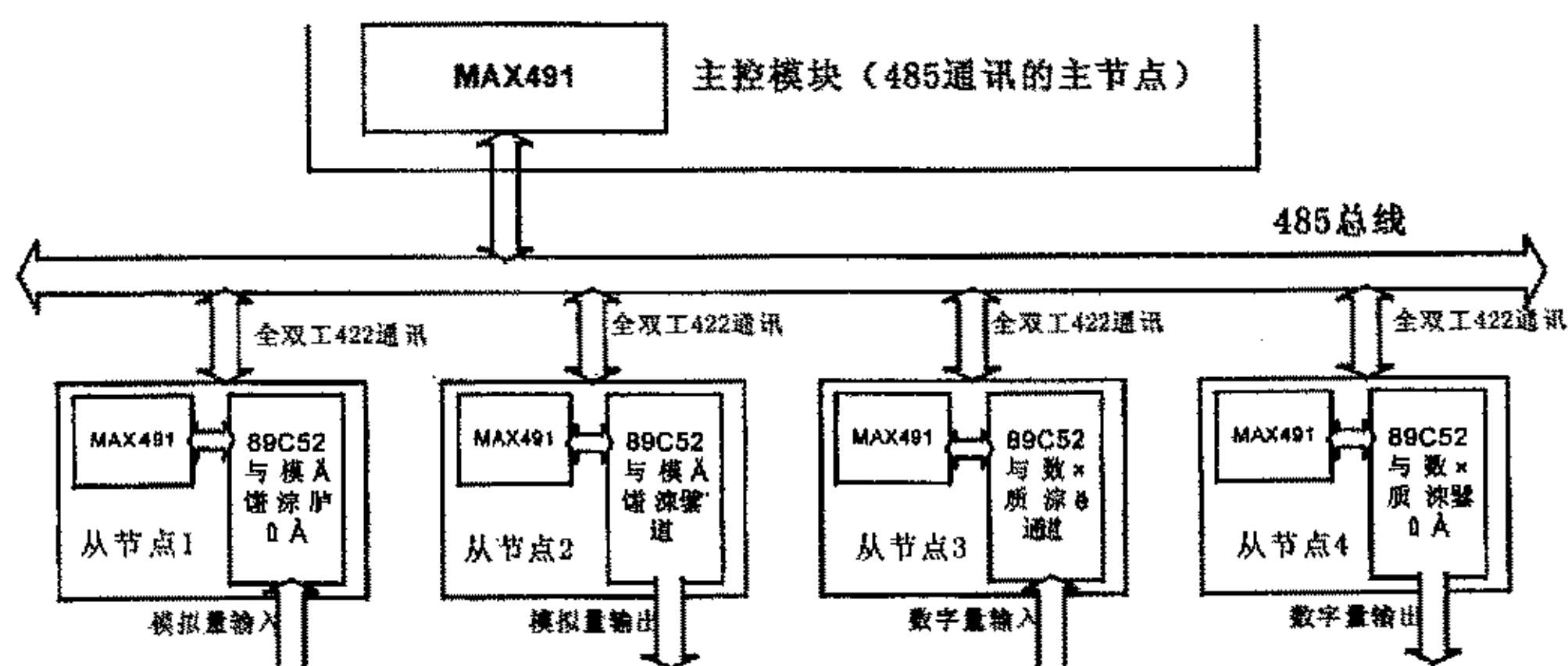


图 3-8 I/O 模块的 485 通讯原理图

## 2. 模拟输入模块(AI)电路设计<sup>[16]</sup>

微处理器采用 P89C52，A/D 采用 MAX1294。电路原理如下图所示：

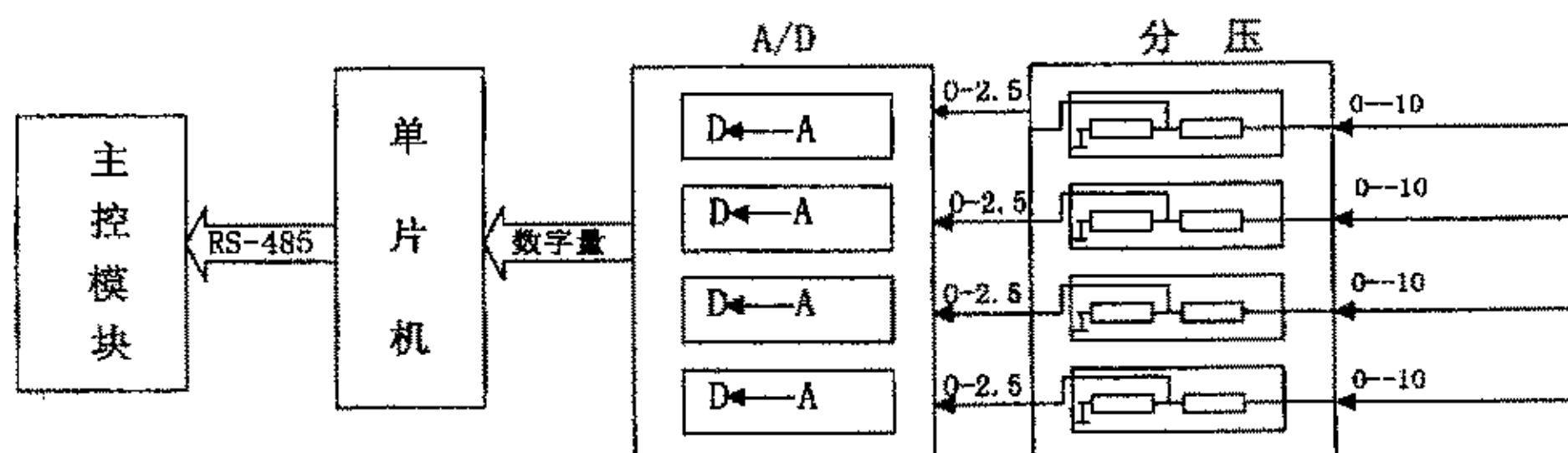


图 3-9 模拟量输入原理图

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. 模拟输入模块(AI)电路设计

微处理器通用采用 P89C52，D/A 采用 MAX527，电路原理如下图所示。

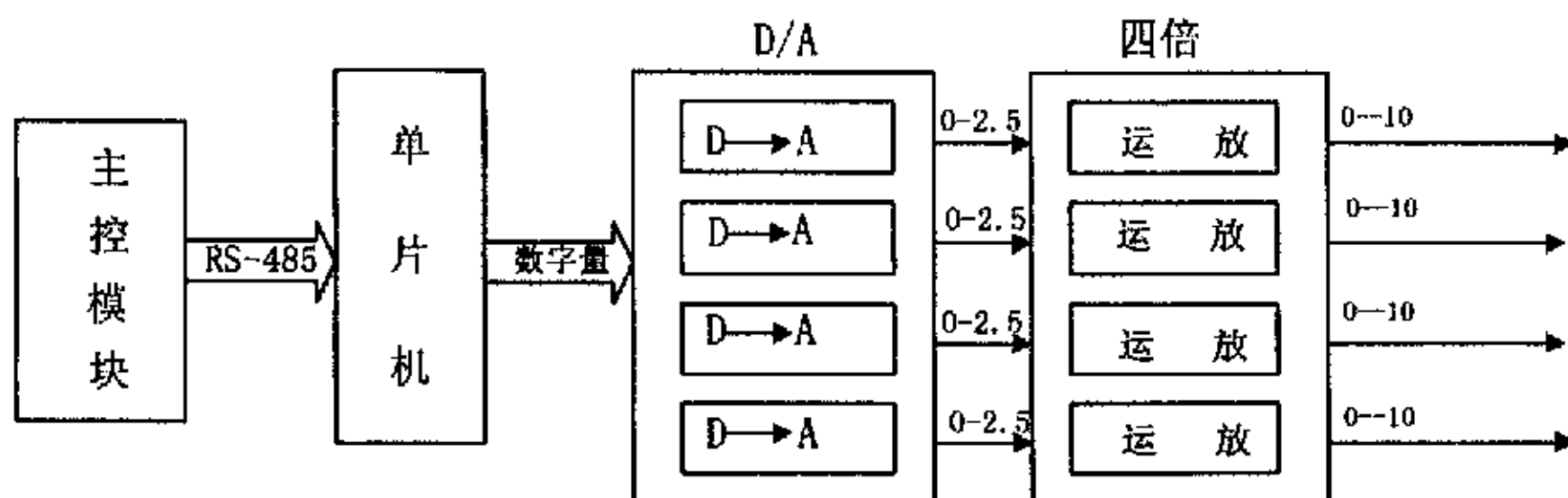


图 3-10 模拟量输出原理图

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。

#### 4. 数字量输出模块(DO)的设计

原理图如下图所示。

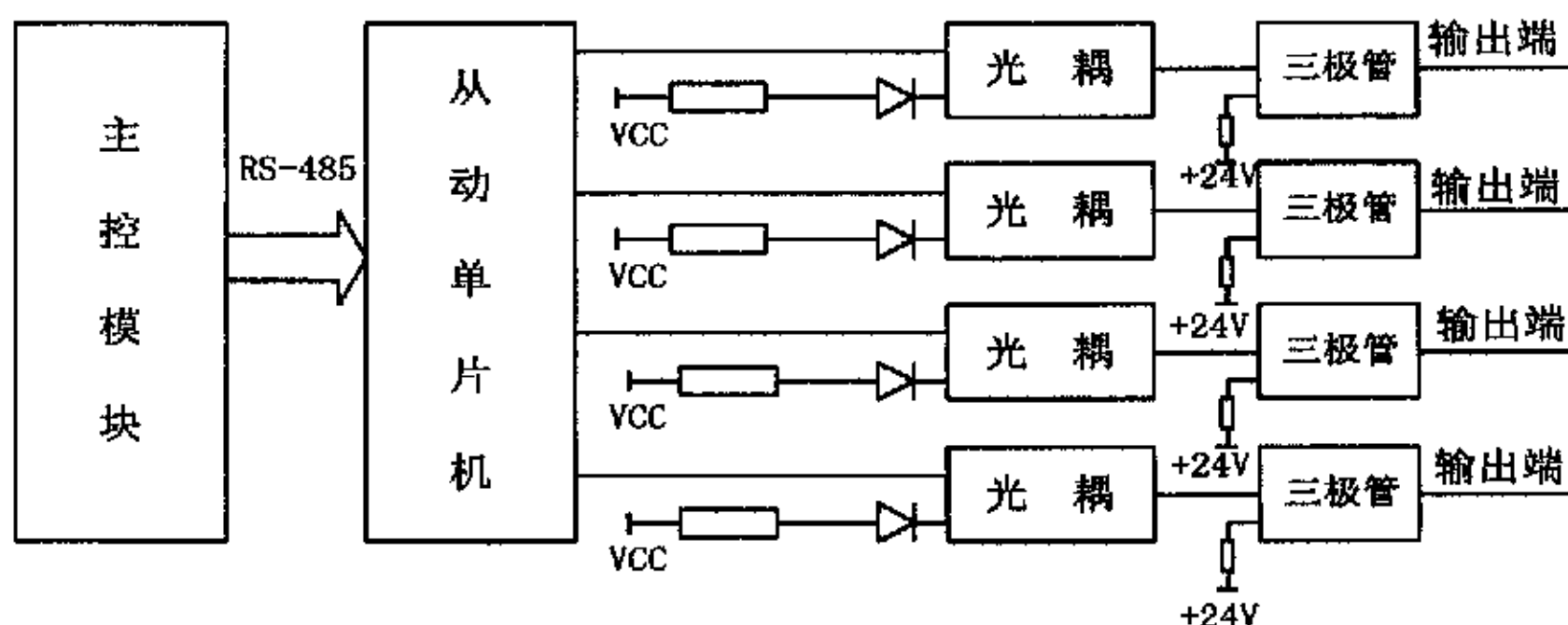


图 3-11 原理示意图

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”。

#### 5. 数字量输入模块设计

原理如下图所示。

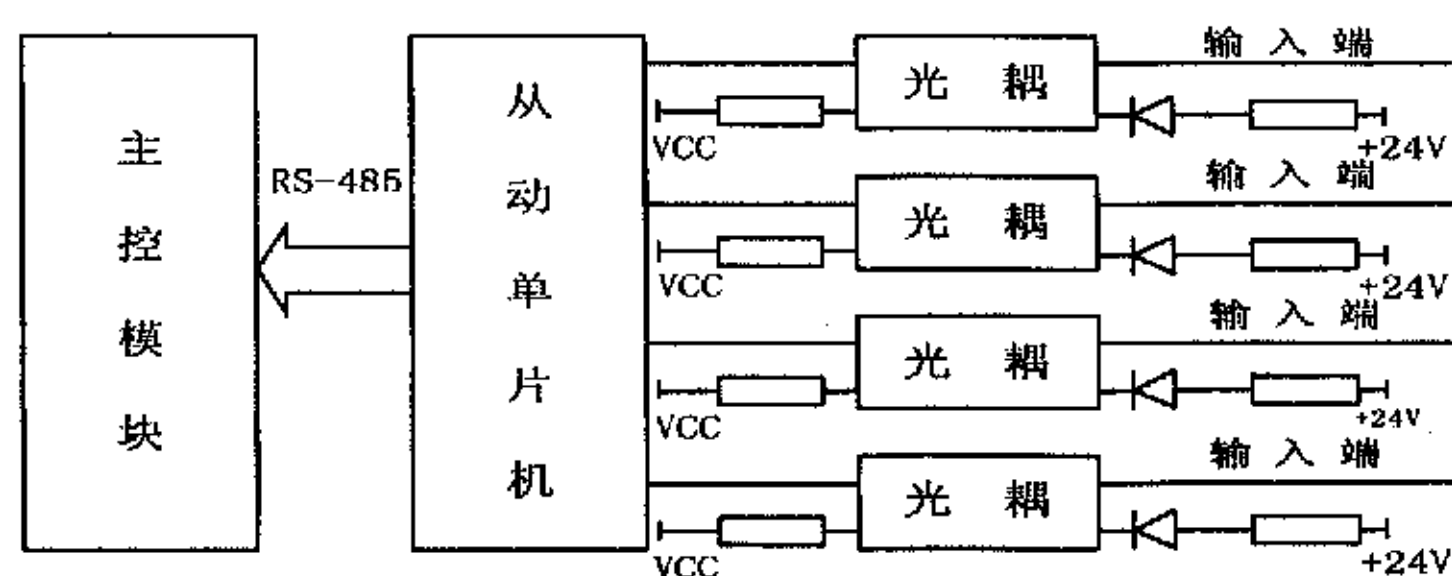


图 3-12 数字量输入模块原理示意图

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 CAN控制器的软件设计及CANopen协议的实现

### 4.1 软件的整体设计<sup>[17][18][19][20]</sup>

本文主要是在此模块化CAN控制器上实现CANopen从节点的通讯协议,以及CAN控制器的节点控制模块和I/O模块之间的485通讯的功能。主要是用C96和C51编写。

CANopen协议内容十分丰富,通讯的重点--标识符的分配可以采用默认值,也可以用DBT服务尽心动态配置,一般在小型的简单网络里,大多数使用默认的标识符,这样既简化了网络的配置,也免去了很多编程的麻烦。

节点控制模块和I/O模块的485通讯用于完成控制模块和I/O模块的数据传输,采用主从式全双工的串行通讯,软件编程属于常规设计。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。

### 4.2 CANopen通讯从结点的实现<sup>[7][21][22][23]</sup>

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。

#### 4.2.1 从节点包括的功能部分

- ✧ 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对象的发送:将需要发送的数据封装成PDO的形式等待发送,需要时间触发方式的数据立即可以发送。
- ✧ PDO远程对象的发送:产生并发送CAN远程帧。
- ✧ PDO对象的接收:当SJA1000产生接收中断时,单片机接收并断定为PDO对

象并进行分析存储;

- ✧ SYNC信息的分析处理:从节点对接收到的SYNC对象进行分析并启动相应的操作;
- ✧ SDO通讯:当收到主节点的SDO初始化对象,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。

## 4.2.2 从节点的总体流程<sup>[24][25][26][27][28]</sup>

### 1. 从节点状态转换.

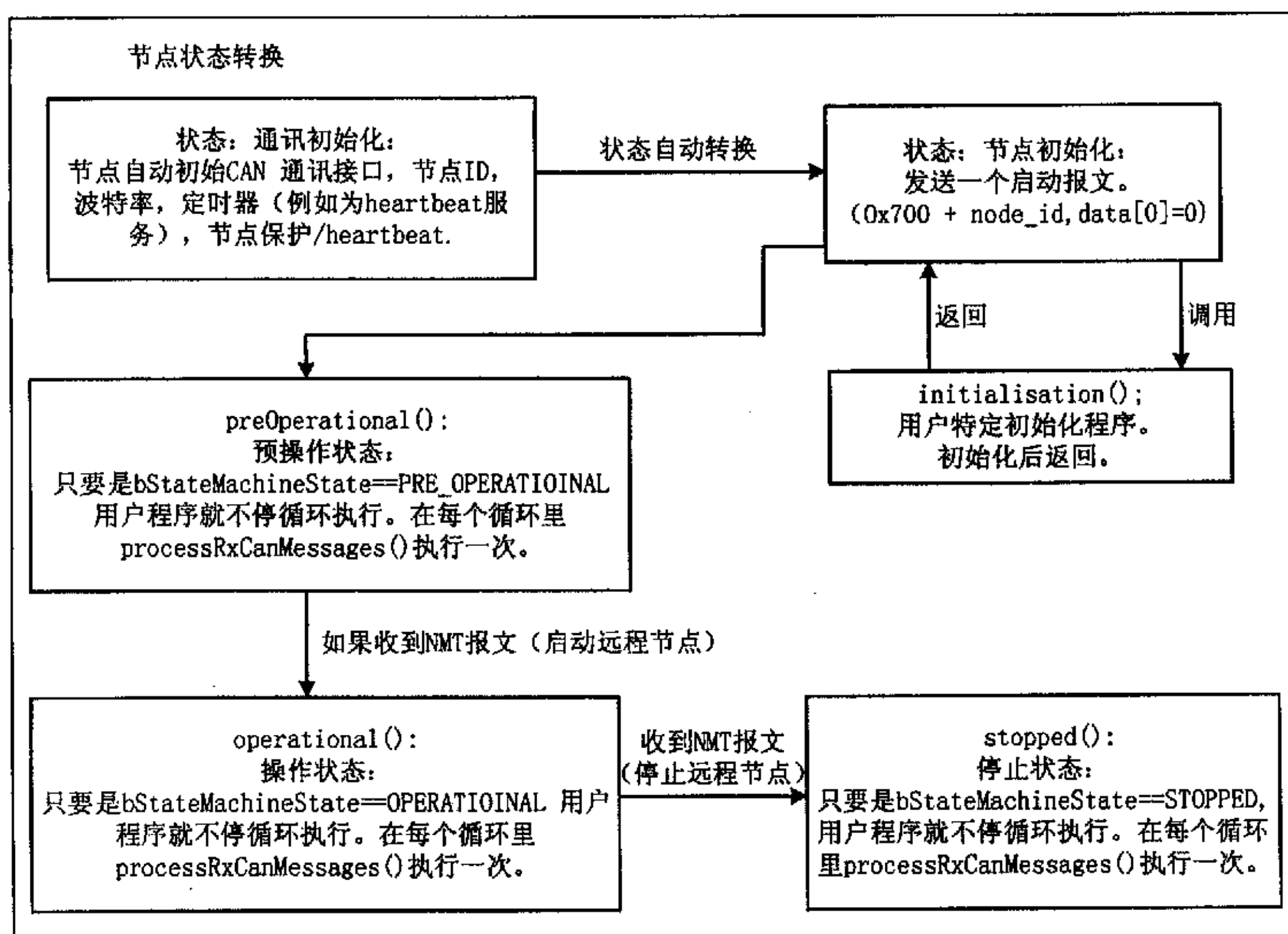


图4-1从结点状态转换流程图

在从结点上电后,首先节点自动初始化 CAN 通讯接口,节点 ID,波特率等参数,,即通讯初始化,然后自动进入节点初始化,调用 `initialisation()` 模块,进行特定的一些用户初始化,完成后,发送一个启动报文,自动进入预操作状态,调用 `preOperational()` 模块,用户的程序循环执行着,在每个循环里,调用 `processRxCanMessages()` 一次,这个模块用来检查接收到的对象,并做相应的处理.在 NMT 报文来之前一直持续在预操作状态,在这个状态里,从结点可以通过 SDO 被配置和设置参数,但是不允许发

送 PDO。在相应的 NMT 报文到来后,即收到启动远程节点命令后,进入操作状态,在这个状态就可以进行正常的通讯了,发送 PDO 了,每个周期依然调用 processRxCanMessages()一次.在收到 NMT 报文,即停止远程节点命令,从节点进入了停止状态(Stopped State),从而停止 PDO 和 SDO 通讯,这个状态可以使从结点专心做自己的事情,但是依然每个周期调用一次 processRxCanMessages()。

## 2. 从结点处理接收报文的过程

如下图所示。

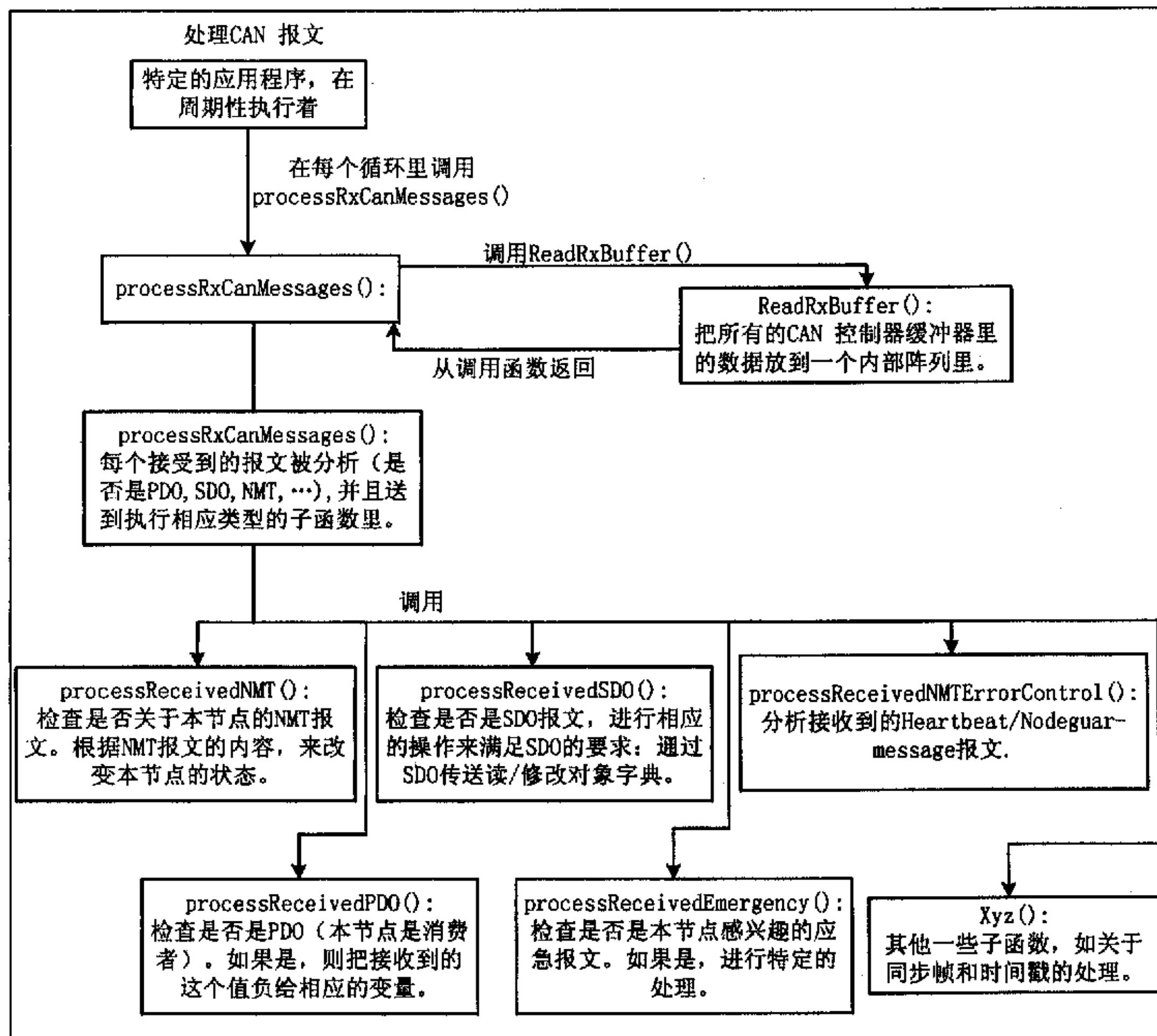


图4-2 从结点处理接收报文流程图

特定的应用程序在周期性的执行.每个周期调用一次processRxCanMessage()功能模块,各个功能模块的功能如下:

- ✧ processRxCanMessages(): 主要负责读取CAN报文,并进一步分析.是否是PDO,SDO,NMT等等.并且送到相应类型的子函数里进一步处理.
- ✧ ReadRxBuffer(): 负责把CAN缓冲区里的数据读到一个数组里.(canMessageRxBuffer).
- ✧ processReceivedNMT(): 检查是否是关于本节点的NMT报文,如果是NMT报文,

根据报文的内容, 改变本节点的状态.

- ✧ processReceivedSDO(): 检查是否是SDO报文, 如果是SDO报文, 则进行相应的操作来满足SDO的要求, 通过SDO传送读/写对象字典.
- ✧ processReceivedPDO(): 检查是否是PDO报文, 如果是PDO报文, 则根据接收到的报文的类型, 把这个值赋给相应的本地变量.
- ✧ processReceivedNMTErrControl(): 分析接收到的是否是Heartbeat/节点保护报文, 如果是, 则相应的处理.
- ✧ processReceivedEmergency(): 检查是否是本节点感兴趣的应急报文, 如果是, 则做特定应急处理.
- ✧ Xyz(): 统一处理一些其他的一些类型报文, 作为保留用. 可以分析是否是同步帧和时间戳等等.

### 3. 发送PDO的过程

- ✧ sendPDO()

当为事件触发发送方式时, 当某个本地变量由于中断或是别到原因改变了状态, 就调用此模块, 把需要发送的变量放到对象字典里, 再把他们映射封装成PDO对象, 发送到总线上去. 当为同步触发发送方式时, 等待同步帧的到来, 再把PDO发送到总线上去. 下面看一个映射封装的例子. 假设节点A的输入变量2和输入变量3要发送到总线上, 映射过程如下图所示.

节点A的对象字典

1000H	设备类型
.....	
6000H, 1	输入变量1, 8位
6000H, 2	输入变量2, 8位
6000H, 3	输入变量3, 8位
.....	

发送PDO映射参数

1A00H, 0	子索引元素数	2
1A00H, 1	第一个映射对象	60000208H
1A00H, 2	第二个映射对象	60000308H

发送PDO通讯参数

1800H, 0	子索引元素数	2
1800H, 1	标识符	01C0H
1800H, 2	传输类型	255

PDO:

标识符	数据	
01C0H	输入变量2	输入变量3

图4-3 映射封装PDO

这就把节点A的输入变量2和输入变量3映射到一个PDO里,传输标识符为01C0H,传输类型255,参考表2-9,表示传输类型为异步传输.

#### 4. 处理heartbeat过程

如下图所示.



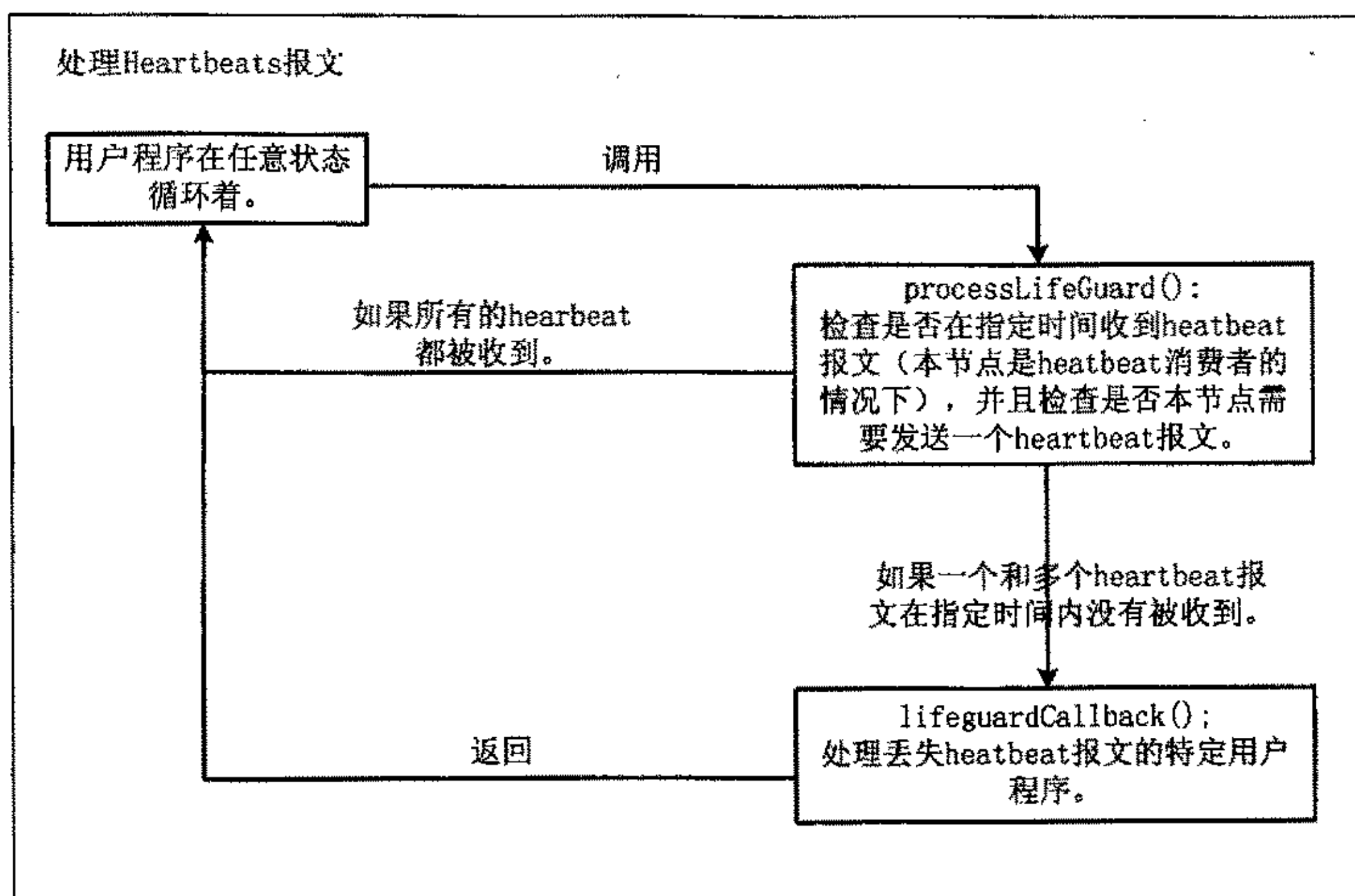


图4-4 heartbeat的处理过程

- ✧ processLifeGuard(): 当用户程序在任意状态循环着的过程中, 每周期调用一次此模块, 检查是否指定的时间收到heartbeat报文否, 并且检查是否本节点需要发送一个heartbeat报文。
- ✧ lifeguardCallback(): 如果一个或多个heartbeat报文在指定的时间没有被收到, 则调用此模块, 运行丢失heartbeat报文的特定用户程序。
- ✧ Interrupt-Service-Routine(): 这个模块每50毫秒调用一次, 对每个节点来说, 当这个节点是一个heartbeat消费者时, 增加heartbeat生产者的time-value. 也每次增加本节点的time-value, 在time-value的帮助下, 就可以知道从上次heartbeat发送或接收后, 过了多长时间。

### 4.3 节点控制模块485通讯的软件设计<sup>[9][16]</sup>

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。

#### 4.3.1 多机通信协议

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。

#### 4.3 I/O模块程序设计

### 4.3.1 数字量输入模块设计

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。

### 4.3.2 模拟量输入模块设计

模拟量的输入相对复杂。通过微处理器首先把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转精度，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。

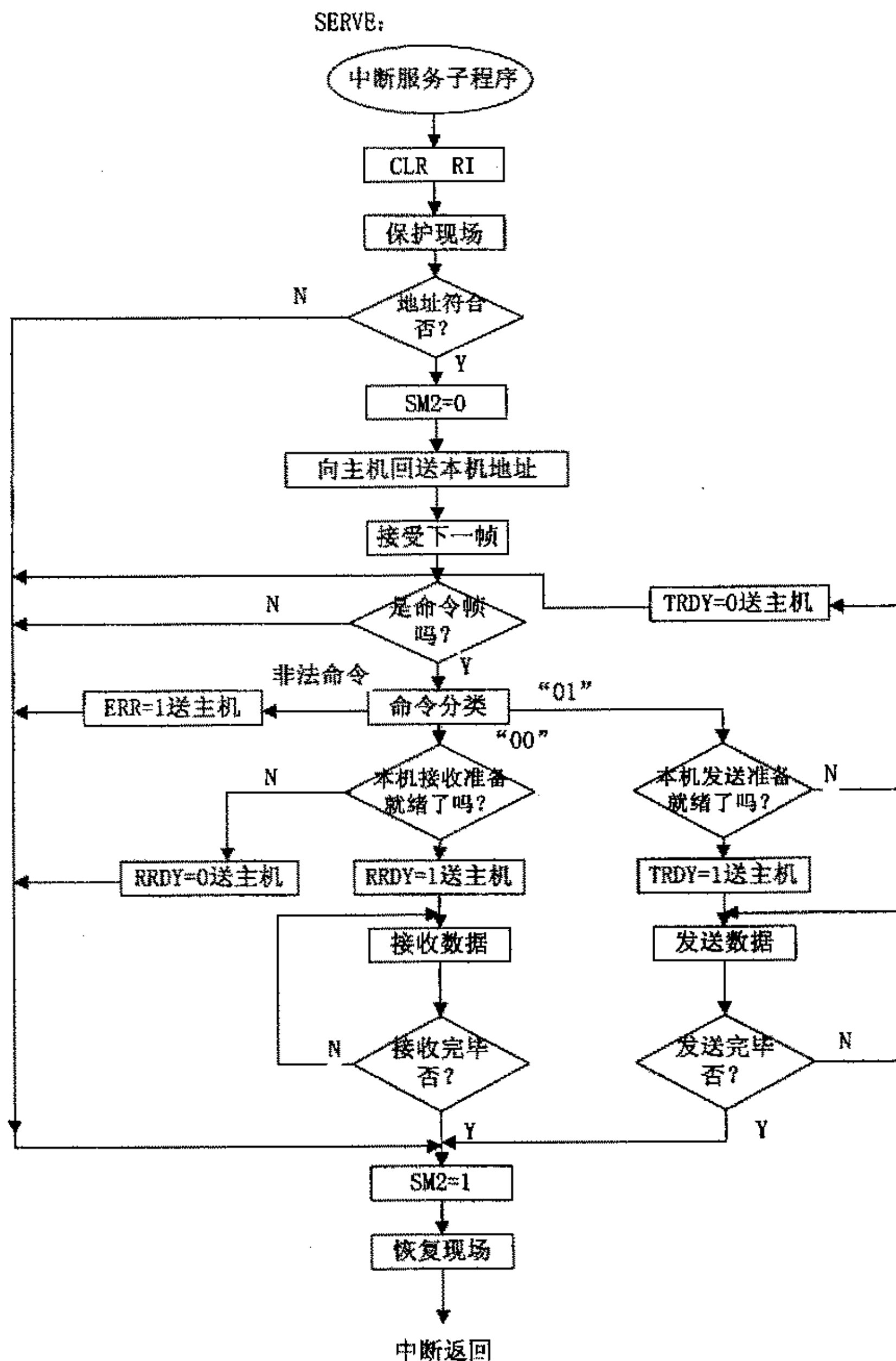


图4-5 I/O模块的485通讯流程图

### 4.3.2 输出模块设计

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。



## 第五章 CANopen 协议实时性的研究及改进

### 5.1 CAN 总线实时性<sup>[2][3][29][30]</sup>

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

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。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。

CAN 总线的 MAC 层协议属于事件触发方式中的载波检测、多路复用/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。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。

所以，数据的标识符决定了数据的优先级，是主要可做文章的地方。而且，由于 MAC 子层不存在修改的灵活性，只能在应用层进行研究改造以增加网络的实时性。

### 5.2 实时数据库的实时调度与 CANopen 协议实时性的改进<sup>[31][32][33][34]</sup>

调度(scheduling)就是为相互竞争的活动访问串行共享资源而排定次序所做的决定。分派(dispatching)是根据调度的决定，授权最合格而又准备就绪的活动立即访问共享资源。

在实时数据库的研究中，目前对实时事务的调度主要采用的是优先级的调度方案。优先级调度方法的特点就主要体现在如何组织事务的调度队列，使队列结构和其中的元素按优先级有序排列。通常在实时系统中使用的优先级措施策略有如下几种：

◇ 最早放行最优先

该策略使“可以开始执行时间”最早的事务具有最高的优先级

◇ 空余时间最短最优先

事务的空余时间是指事务的截止期和事务预计能完成时间之间的差。

◇ 截止期最早最优先

该策略使“截止期”最早的事务具有最高的优先级，截止期即实时事务应该完成的最晚时间。

◇ 可达截止期最早最优先

它是空余时间最短最优先的改进，它通过对事务的执行时间进行预分析来判断事务的截止期是否可达，对可达的事务进行截止期最早的优先调度。

◇ 价值最高最优先

构造价值函数，根据事务的价值大小来分配优先级。这种方法也可以看成是构造优先级函数来给事务指派优先级。采用函数的方法能更加综合地考虑指派优先级的因素，使优先级的措施更加公平。

我们着重再说一说实时事务的截止期，按截止期性质的不同，实时事务可分为：

- 硬(截至期 / 实时)事务超过截止期会导致恶果(价值函数取大的负值)，它对应于安全危急性活动。如下图所示。

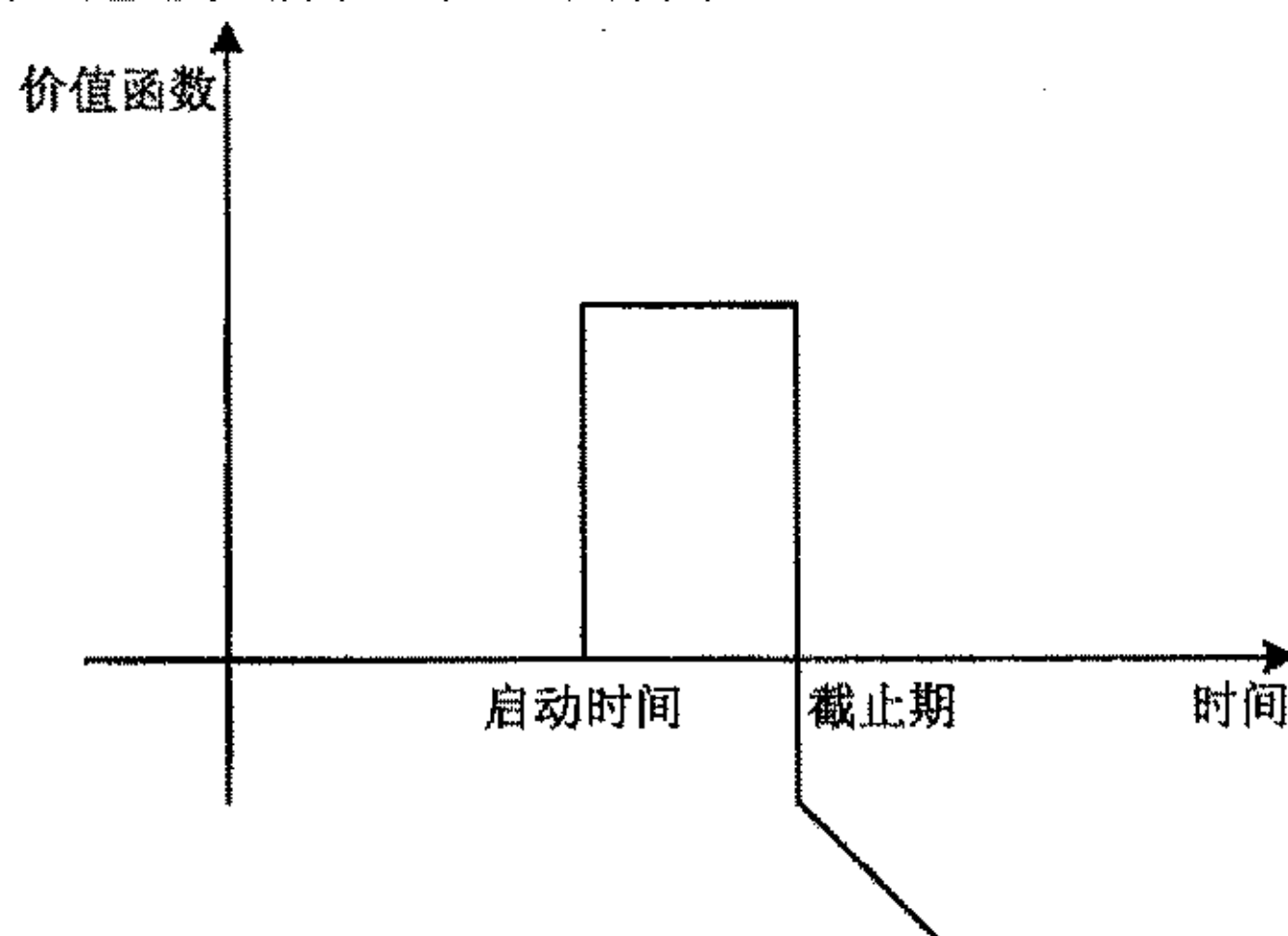


图5-1 硬实时事务的关键性

- 软(截至期 / 实时)事务超截止期仍有一定的价值，但不断下降，直到某一时刻(称为最终有效时间)，其价值降到零，此后保持为零(不会为负)。如下图所示。

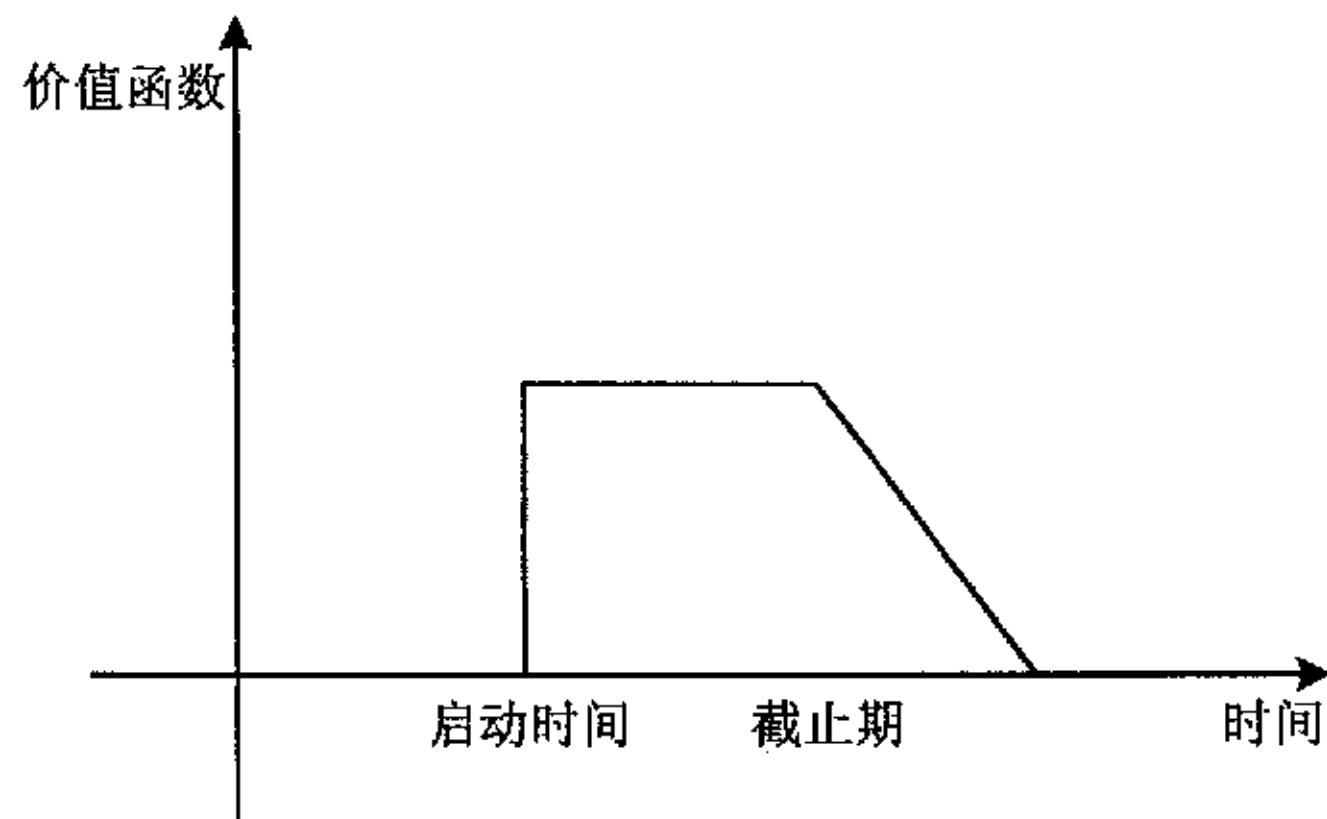


图5-2 软实时事务的关键性

- 固(截至期 / 实时)事务一旦到达截止时间，其价值立即降为零，此后固定为零(也不会为负)。

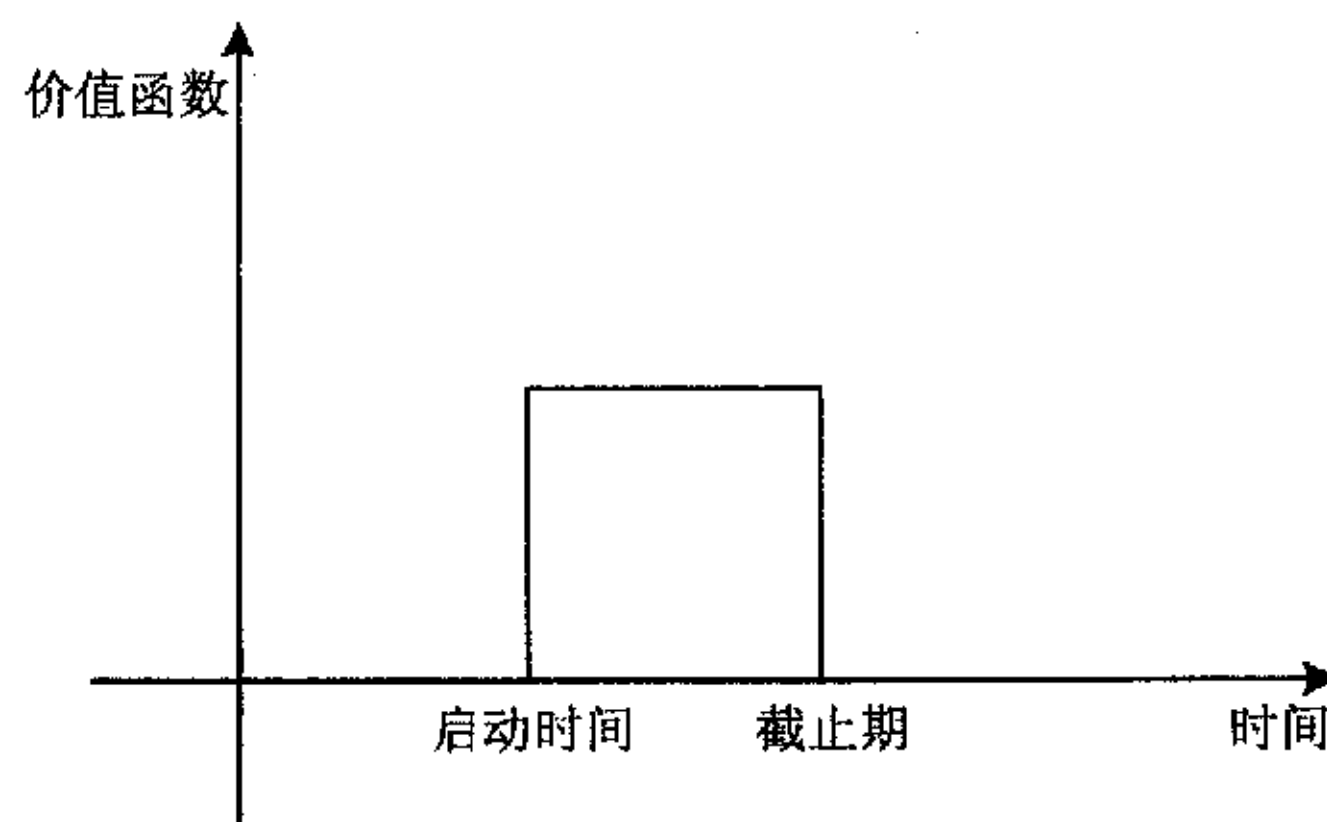


图5-3 固实时事务的关键性

从上面优先级的分配方法可以看出，各种优先级的分配方案都是各有侧重的。如最早放行最优先的方法只强调了到达时间的优先，它的实现虽然简单，但对事务的紧迫度和重要性等却不加区别。空余时间最短最优先考虑了事务的紧迫程度却又对事务的重要程度没有加以考虑。改进后的可达截止期最早最优先的调度策略对于可调度的实时事务集是一种简单而且比较有效的调度方法，可以证明它能保证满足独立事务集中各事务的截止期，但判断多个事务是否可调度是一件困难的工作。我们暂且撇开各个调度方法的优缺点，目的是把各种调度方法和CANopen协议的实时性结合起来。

在CAN总线里，由于用标识符把CAN报文的优先级区分开来。一方面在总线上仲裁时，优先级高的报文争得总线权，另一方面，在每个节点内部，我们也可以按照对象优先权的高低来分配传送的队列顺序。

关于CANopen协议PDO标识符的分配，即决定了发送对象的优先级，可以静态的实施，如在脱机状态下，作为系统设计或配置的一部分，这种PDO标识符的分配没有“自适应”和“优化”的能力，不能动态地进行事务的“可调度性”

和“可行性分析”，一般的标识符的分配方法都是按照这种静态实施的。而且确实大部分的静态分配能满足应用环境，但是在一些大型网络里，网络节点比较多，而且复杂，况且还会有新的节点的加入，如果只用静态的分配PDO标识符的方法，会区分不开硬实时和软实时的数据。造成重要的数据的PDO标识符优先级不高，不能动态的根据网络情况的变化自动提高自身的优先级。所以，PDO标识符的分配就需要动态的施行，即在系统执行期间，靠应用程序实时地进行事务的“紧急性”分析，然后通过一些动态的调度算法，来提高或是降低需要发送PDO的优先级。

下面我通过把截止期最早和“价值最高最优先”结合起来，对CANopen节点发送PDO的标识符进行动态配置来提高CANopen的实时性。

我们知道每个发送的PDO最多可以包括8个字节，具体变量数根据具体的情况而定，我们给每个变量分配一个参数D(截止期deadline)，这样最多8个截止期参数D1、D2、D3、D4、D5、D6、D7、D8，如果有5个变量则为D1、D2、D3、D4、D5。

再为每个变量的截止期性质分配一个参数E，此参数可以有三种值：硬实时事务E1；软实时事务E2；固实时事务，E3。其中硬实时事务的值最高，软实时事务值居中，固实时事务的值最低。

每个PDO的子索引里的参数还有传输类型(Transmission type)，我们称它为参数T。

PDO子索引的禁止时间(Inhibit time)，我们成为参数I。

最后我们构造一个价值函数Q。公式如下：

$$Q = p_1 D_1 + p_2 D_2 + p_3 D_3 + p_4 D_4 + p_5 D_5 + p_6 D_6 + p_7 D_7 + p_8 D_8 + p_9 E + p_{10} T + p_{11} I。$$

其中 $p_1 \dots p_{11}$ 为加权系数。最后根据得出的价值函数的高低动态分配标识符。

当然，这样动态分配标识符，会有一些问题，一方面增加了控制器的负担，如果控制器控制任务本来就较重，那么增加的计算价值函数的算法会更加加重CPU的负担。所以应该考虑控制器的处理能力。另一方面，在动态分派PDO标识符的时候，要考虑到其他对象的标识符，像同步对象(SYNC)，SDO，heartbeat对象，节点保护，应急对象等。既要保证实时数据的快速实时传送出去，也不能妨碍其他CANopen服务协议传送。

### 5.3 论文的实验验证

本论文详细剖析了CANBUS的高层协议CANopen，并自主开发了模块化CAN控制器，实现了CAN总线的基本通讯功能，在此控制器上实现了CANopen从结点的协议。

### 5.3.1 开发的CAN控制器实验平台

本文设计的CAN控制器二级总线结构，包括一个CAN节点控制模块、DI模块、DO模块，外观如下图所示。

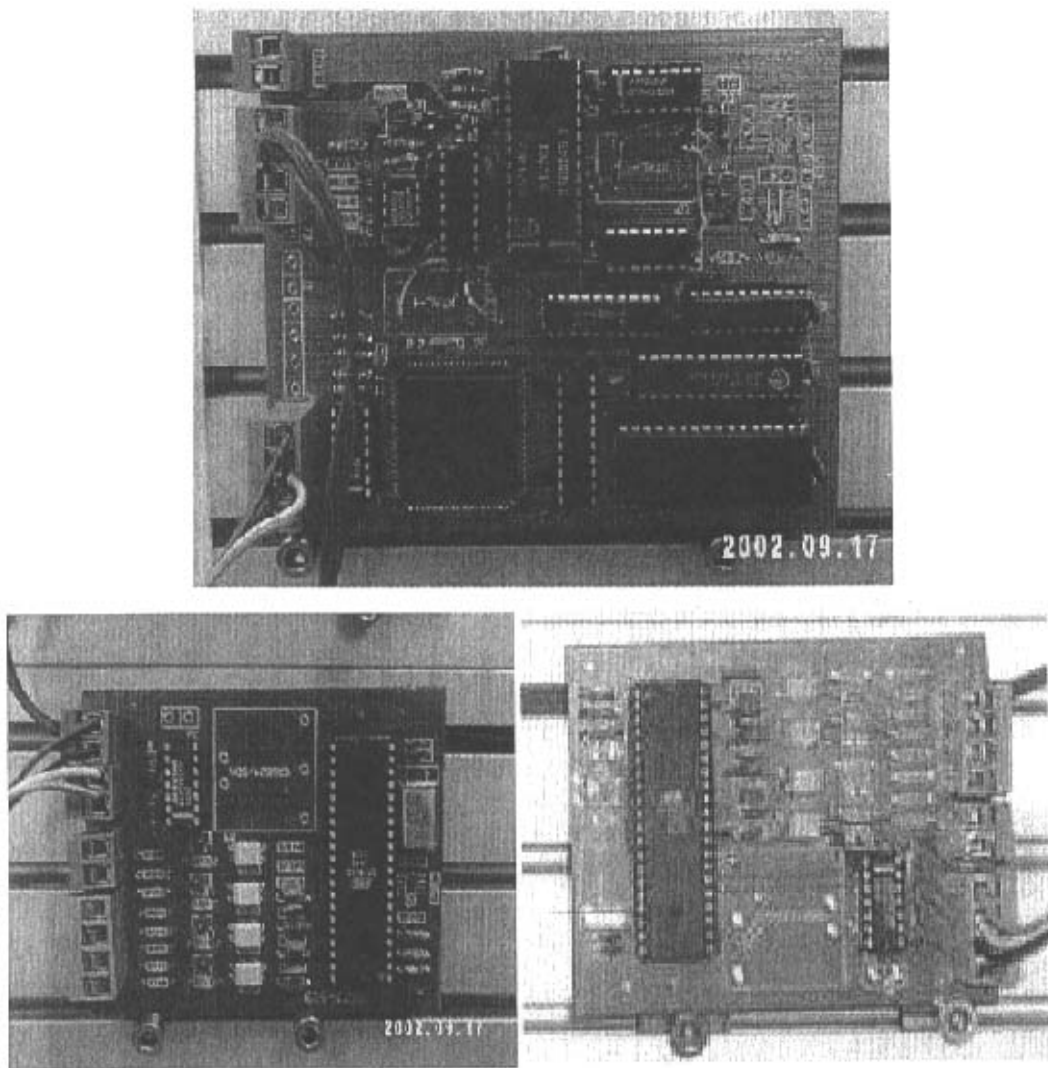


图5-4 节点控制模块与I/O模块

### 5.3.2 实验验证

首先验证节点控制模块和I/O模块的通讯的有效性，由DI模块输入开关量，通过485总线传送到节点控制模块，再由节点控制模块传送给DO模块，通过I/O指示灯，显示通讯顺利。

然后验证CANopen协议数据传输的有效性。在验证网络中，用一个带PCI-CAN



卡的计算机作为网络的主节点，本文的节点控制模块作为从节点。经过验证节点控制模块可以自动完成通讯初始化和节点初始化，然后进入预操作状态，等待收到主节点NMTA启动远程节点报文后，可以进入操作状态，在操作状态里，首先用事件触发模式进行异步PDO传送，当DI模块的值发生变化时，触发异步PDO传送，经过验证，状态变化300次，通讯正确。然后通过由主节点发送远程对象，向从节点请求数据，节点控制模块的状态值设为34H，经过主节点发送远程对象，请求，从结点发送状态值，主节点接受的数据正确，重复几次，依然正确。所以，经过验证CANopen从结点的协议可以正确通讯。

## 结 束 语

控制器局域网是一种有效支持分布式控制和实时控制的串行通信网络,由于 CAN 总线具有卓越的特性和极高的可靠性,特别适合工业过程监控设备的互连。被认为是几种最优前途的现场总线之一,已经越来越受到工业界的重视。

在本课题中,笔者进行了以下方面的研究:

- 1、对 CANbus 高层协议 CANopen 进行了仔细的剖析。
- 2、设计了一种模块化的 CAN 总线控制器,引入二级总线的概念,采用模块化的结构,使控制器可以根据应用对象改变结构,增加了通用性。
- 3、在此控制器上编写了 CANopen 从节点的协议,实现了 PDO、SDO、NMT 以及 heartbeat 对象的通讯,完成了基本的通讯和网络管理功能。
- 4、对 CAN 总线高层协议 CANopen 的实时性做了初步的改进探索。

在完成以上工作后,笔者认为接下来需要研究的工作包括:

- 1、在小型 CANbus 网络中,进一步简化网络的配置工作。
- 2、在大型网络中,节点比较多时,进一步加强 CANbus 的实时性,在高层协议方面,可以研究一些特定的算法,动态地分配 PDO 的标识符,改善网络的性能。
- 3、在硬件方面,由于二级总线结构的引入,使系统的灵活性通用性增强,但是系统的实时性可能会受影响,需要在硬件设计上做一些改进。

由于本人水平所限,论文难免有疏忽和错误之处,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] 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
- [2] 阳宪惠, 现场总线技术及应用, 清华大学出版社, 1996.1
- [3] 郭宽明, 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
- [4] M.Farsi, K.Ratcliff, Manuel Barbosa, An introduction to CANopen, Computer & Control Engineering Journal, August 1999. 161~168
- [5] CANopen Communication Profile for Industry System Based on CAN, CiA Draft Standard 301 October 1996.
- [6] 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.
- [7] Karl Ratcliff, CANopen Implementation Issues, 1997 The Institution of Electrical Engineers, Print by The IEE, Savoy Place, London WC2R 0BL, UK.
- [8] FARSI M. and BARBOSA M., CANopen implementation application to industrial networks, Research Studies Press Ltd. ISBN 0-86380-247-8 2000
- [9] 孙涵芳, Intel 16 位单片机, 北京航空航天大学出版社, 1995.11
- [10] 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
- [11] Philips Semiconductor, SJA1000 Data Sheet, Philips Semiconductor, 2000
- [12] 郭宽明, CAN 总线系统设计中的几个问题, 1998.9: 18~20
- [13] 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
- [14] 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
- [15] 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.
- [16] 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
- [17] Intel 80C196 单片机应用实践与 C 语言开发
- [18] 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
- [19] 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 [4] 沈德耀 金敏, 现场总线纵横谈, 基础自动化, 2000.7

- (4): 1~5
- [20] Kirk Zurell, 嵌入式系统的 C 程序设计, 北京: 机械工业出版社 2002
- [21] Luis Vazquez, Leandro Rojas, Samuel Galceran, Simplified CANopen Application Layer Model for Educational Proposals, 2000 IEEE 255~259
- [22] Zdenek Hanzalek, Tomas Pacha, Use of the Fieldbus Systems in Academic Setting, 1999 IEEE, 93~97
- [23] M Karsi, K Ratecliff, CANopen: Configure and Device Testing, 373~380.
- [24] 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
- [25] 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
- [26] 郝忠孝, 微型机实时软件设计, 机械工业出版社, 1989, 2
- [27] 李孔安, 现场总线网可靠性设计与分析, 基础自动化, 1996, No.2
- [28] Wayne Wolf, 嵌入式计算系统设计原理, 北京: 机械工业出版社 2002
- [29] 王智 王天然, 工业实时通讯网络(现场总线)的基础理论研究与现状(上), 信息与控制, 2002.4: 146~152 163
- [30] CiA DS-202 TO DS-207, CAN Application Layer Specification, 1996
- [31] 何炎详 刘朝阳, 分布式实时数据库系统中一种新的复制控制策略, 小型微型计算机系统, 2000, 21 (3), 298~301.
- [32] 党德鹏 刘云生, 一种新的实时事务并发控制方法, 小型微型计算机系统, 2002, 23 (7), 849~852
- [33] 张子仲 李凡, 实时数据库系统中的 I/O 调度策略研究, 华中理工大学学报, 1997, 25 (2), 73~75
- [34] 刘云生 梁爽, 实时数据库的定时调度策略, 小型微型计算机系统, 2001, 22 (11), 1375~1377
- [35] 高强 董力川, 嵌入式 CAN 总线控制系统设计, 自动化技术与应用, 2002.2: 33~34
- [36] 孙占辉 张培仁, CAN 总线在现场总线控制系统中的应用, 微计算机信息, 2002, 18 (7) : 7~8
- [37] 范铠, 现场总线的发展趋势, 自动化仪表, 2000.2: 1~4
- [38] 蔡月明 刘浩, 基于 CAN 总线的工业控制系统, 中国仪器仪表, 2001.5: 16~18
- [39] 王智 王天然, 工业实时通讯网络(现场总线)的基础理论研究与现状(下), 信息与控制, 2002.6: 241~249
- [40] 袁爱进 杨国勋, CAN 现场总线通信系统的研究, 计算机工程, 2001, 27 (9) : 129~131

- [41] 陈双叶 赵颖燕, CANBUS 分布式计算机控制系统设计及应用, 北京工业大学学报, 2002 28 (2) : 134~137
- [42] 王健 肖学军, CAN 总线技术应用, 微计算机信息, 2000, 16 (5) : 1~3
- [43] 朱敏 张崇巍, CAN 总线在数据采集与控制系统中的应用, 合肥工业大学学报, 2002 25 (3): 345~349
- [44] 王红蕾, CAN 总线技术在控制系统中的应用与实现, 现代机械, 2002, 2, 26~27
- [45] 袁爱进 唐明新, CAN 现场总线系列仪表一种通用化软件设计技术, 计算机工程, 2001, 27 (4) : 144~146
- [46] 王锐, CAN 现场总线及模拟量输入模块的设计, 金川科技, 2001, 3 : 20~23
- [47] 律德才 马峰, CAN 总线控制器 SJA1000 的原理及应用, 电测与仪表, 2002, 39 (4): 57~59
- [48] 马安平 张春, 基于 CAN 总线通讯的分布式控制系统的原理与设计, 黑龙江大学自然科学学报, 1999, 16 (1) : 67~70
- [49] Manuel B.M.Parbosa, Adriano da Silva Carvalho, Mohammed Farsi, A CANopen IO Module: Simple and Efficient System Integration, 1998 IEEE, 155~159.
- [50] 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.
- [51] 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
- [52] TINDELL,K: Calculating controller area (CAN) message response times, 1995, Control Eng.Practice, 3, (8) pp.1163-1169.
- [53] Ian Broster and Alan Burns, Timely use of the CAN Protocol in Hard Real-time Systems with Faults, 2001 IEEE, 95~102.
- [54] Gianluca Cena and Adriano Valenzano, Integrating the CAN and MUST Access Techniques in a Single Field bus, 2001 IEEE, 231~239.
- [55] Paulo Pedreiras and Luis Almeida, EDF message scheduling on controller area network, Computing & Control Engineering August 2002. 163~170.
- [56] Frank Robler and Birgit Geppert, Applying Quality of Service Architectures to the Field-bus Domain, 1997, IEEE, 39~48.
- [57] DeMACRO, T., Structured analysis and system specification, Prentice Hall, 1979



- [58] TINDELL.K. and BURNS A., and WELLINGA A., Calculating controller area network(CAN) message response times, Distributed computer Control Systems, Spain September 1994.
- [59] TINDELL.K. and BURNS A., Guaranteeing message latencies on control area network(CAN), 1<sup>st</sup> International CAN conference, Germany, Sept,1994.
- [60] Ian Broster, Guillem Bernat and Alan Burns, Weakly Hard Real-time Constraints on Controller Area Network, Proceedings of the 14th Euromicro Conference on Real-Time Systems 1068-3070/02, 2002 IEEE .
- [61] G. Bernat. Weakly hard real-time systems. IEEE Transactions on Computers, 50(3):308–321, March 2001.
- [62] 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.
- [63] 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
- [64] 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
- [65] 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, 8 43~44, 213
- [66] 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
- [67] Hasnaoui, S, Kallel, O, Kbaier, An implementation of a proposed modification of CAN protocol on CAN field bus controller component for supporting a dynamic priority policy, Industry Applications Conference, 2003. 38th IAS Annual Meeting. Conference Record of the, Volume: 1, 12-16 Oct. 2003, 23 – 31.
- [68] Hasnaoui, S, Bouallegue, A., A proposal modification of CAN protocol to support a dynamic priority policy being able to be implemented on CAN fieldbus controller components, Industry Applications Conference, 2000. Conference Record of the 2000 IEEE, Volume: 2, 8-12 Oct. 2000, 1129 – 1136.
- [69] Almeida, L., Pedreiras, The FTT-CAN protocol: why and how, Industrial Electronics, IEEE Transactions on, Volume: 49, Issue: 6, Dec. 2002, 1189 – 1201.
- [70] Cena, G., Valenzano, A., An improved CAN fieldbus for industrial applications, Industrial Electronics, IEEE Transactions on, Volume: 44, Issue: 4, Aug. 1997,

- 553 – 564.
- [71] Jong Man Jeon, Dae Won Kim, An analysis of network-based control system using CAN (controller area network) protocol, Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, Volume: 4, 2001, 3577 – 3581.
- [72] Dobrin, R., Fohler, G., Implementing off-line message scheduling on controller area network (CAN), Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on, 15-18 Oct. 2001, 241 – 245.
- [73] Di Natale, M., Scheduling the CAN bus with earliest deadline techniques, Real-Time Systems Symposium, 2000. Proceedings. The 21st IEEE, 27-30 Nov. 2000, 259 – 268.
- [74] Fonseca, J.A., Almeida, L.M., Using a planning scheduler in the CAN network, Emerging Technologies and Factory Automation, 1999. Proceedings. ETFA '99. 1999 7th IEEE International Conference on, Volume: 2, 18-21 Oct. 1999, 815 – 821.
- [75] Tindell, K.W., Hansson, H., Analyzing real-time communications: controller area network (CAN), Real-Time Systems Symposium, 1994, Proceedings. , 7-9 Dec. 1994, 259 – 263.
- [76] Cena, G., Valenzano, A., A distributed mechanism to improve fairness in CAN networks, Factory Communication Systems, 1995. WFCS '95, Proceedings, 1995 IEEE International Workshop on, 4-6 Oct. 1995, 3 – 11.
- [77] 李超锋 卢炎生, 实时数据库事务调度分析, 中南民族学院学报(自然科学版), 2001, 20 (2), 5~9.

## 发表论文及参加科研情况说明

研究生期间发表论文一篇：

《具有高可靠性和灵活性的 CAN 总线控制系统的设计与开发》，组合机床与自动化加工技术，2004.1

参加的科研项目包括：

- 1、BA-I 型智能大厦中央空调控制系统实验装置的开发
- 2、基于 CANopen 协议模块化 CAN 控制器的研究
- 3、PC-BASED BA-II 型智能大厦中央空调控制系统的开发
- 4、BA-III 型智能大厦中央空调控制系统实验装置的开发

## 致 谢

本课题从选题到最后完成都是在吴老师的细心指导下完成的，笔者从中体会到了吴老师的严谨的科研作风和勤勤恳恳的工作态度，当课题进展出现困难时，吴老师总是给予认真的帮助，使我受益匪浅。同时还得到了袁浩老师和李保林老师在科研中给与的很大帮助，金文老师为本课题的研究提供了很大的方便。

此外，实验室的师兄师姐：张召、王冬青、郑爱红、方来华，以及何熠、李长宾、刘林山、盛涛和梁瑾同学，还有师弟师妹：杜春艳、刘屏凡、王立石、张小明、和郭欣等，他们都对我的学习和生活给与了很大的帮助。

当然，我的父母和姐姐一直在我的身后默默的支持我，没有他们就没有我的一切。

最后向他们衷心的感谢，笔者虽然学艺不精，但是我会继续努力的。

# CAN bus高层协议CANopen的研究以及在模块化CAN控制器上的实现

作者：[宋晓强](#)  
学位授予单位：[天津大学](#)



## 相似文献(10条)

1. 期刊论文 [王雁平, 张永春, WANG Yan-ping, ZHANG Yong-chun 基于CAN总线的现场总线仪表设计 -常州工学院学报2009, 22\(1\)](#)  
CAN总线控制网络直接面向生产过程,因其独特的设计、高性能而被广泛应用.在CAN总线控制网络中会大量使用现场总线仪表,利用现场总线技术、微处理器技术等现代科学技术手段设计的基于CAN总线的现场总线仪表系统,较好地解决了CAN现场总线仪表本身成本高的瓶颈问题.
2. 学位论文 [李智星 CAN总线网络监控系统 and CAN总线智能适配卡的设计 2004](#)  
工业化网络是实现大规模工业生产管理的基础,是实现工业现代化的重要组成部分.本文重点研究了直接面向生产一级的工业化网络系统.现场总线控制系统.本文首先探究了现场总线控制技术的概念和特点、网络体系结构和网络模型,接着介绍了具体应用的现场总线CAN(Control Area Network)总线,之后详细地叙述了我们为南京某热处理厂设计的基于CAN总线的现场总线监控系统-钢包炉温度监控系统一的设计开发实例.本文还详细介绍了CAN总线智能适配卡的硬、软件设计,以及CAN总线应用的小结.最后本文将总结介绍现场总线系统的可靠性和抗干扰性设计.本文的研究必将对我国自行研制和使用成本低廉、灵活实用的现场总线控制系统起到积极作用.
3. 期刊论文 [周治邦 适用于安全性控制的现场总线:时间触发的CAN总线 -工业控制计算机2004, ""\(3\)](#)  
CAN总线是在工业控制领域普遍使用的著名现场总线.为了使CAN总线可以应用于有严格实时要求的工业控制场合(例如安全性自动控制装置),爱尔兰的Limerick大学正在研制一种具有确定消息传输行为的、时间触发的CAN总线.本文介绍事件触发和时间触发的概念,以及TTCAN的主要特点.
4. 学位论文 [朱海志 电动车控制系统CAN总线高层协议的研究与开发 2006](#)  
现场总线是工业现场控制领域一门新兴的通讯技术,曾被誉为控制技术新时代的开端. CAN(ControllerAreaNetwork)总线具有成本低、速度快、实时性和可靠性较高的特点,其透明开放的协议用户自行定制和扩展提供了方便,因此, CAN总线成为世界范围内较早出现的并且得到广泛使用的现场总线国际标准之一.  
控制器局域网(CAN)为串行总线通讯协议,能有效地支持具有很高安全等级的分布式实时控制.由于在CAN2.0规范标准中,只规定了物理层和数据链路层,实际上,即使是仅仅完成一些非常简单的基于CAN总线的分布式系统,也少不了更上层的协议或规约.除了这两层基本的服务之外,用户往往还要求或希望有更多功能,因此制定与开发CAN总线应用层协议是使用这种现场总线的必备的环节.  
本论文介绍了应用于多电机蓄电池车辆控制的测控系统,分析了其技术特点和应用要求;论述了现代工业系统的分布系统及其实时性问题,分析了CAN现场总线网络环境相关的通讯协议和模型,其中包括了CANopen、DeviceNet、SDS等几种流行的CAN总线高层协议;分析了在现场总线网络环境中触发调度机制.根据CAN总线通讯的特点和要求,选择制定了适合整车的控制系统的网络模型与调度机制.  
最后,在原有基础上设计开发了适合“多电机蓄电池车辆整车控制系统”的CAN应用层通信协议,根据目标系统的要求并结合本系统的特点,对所传输的信息进行分类总结,确定了信息标识符ID的分配和定位,在自定义数据格式的基础上,建立了可靠的数据收发机制、网络管理机制和错误处理机制共同维护协议的运转,实现了CAN总线实时可靠的数据传输.
5. 期刊论文 [季宏志, 范多旺 一种现场总线-控制器局域网\(CAN\)总线 -甘肃科技2003, 19\(4\)](#)  
概述了现场总线,并针对现场总线中的CAN总线的特性及报文传输时如何获得总线主权进行了分析介绍.
6. 学位论文 [黄欢 基于CAN总线网络监控系统研究与开发 2009](#)  
现场总线控制系统是新一代的全分布式控制系统,适应了控制系统向智能化、网络化、分散化发展的趋势,具有强大的生命力,它将使控制系统的体系结构以及自动化技术发生一次根本性变革.  
现场总线是连接智能现场设备和自动化系统的数字式、双向传输、多分支结构的通信网络.它是计算机技术、通信技术和控制技术高度综合与集成的产物,是应用在工业生产现场、在微机化测量控制设备之间实现双向串行多节点数字通讯的底层网络控制系统.现在总线控制系统采用双绞线、多回路的全数字化信息传输不仅提高了整个系统的可靠性和抗干扰能力,同时也大大节省了整个系统的投资、安装及后期的维护费用;现场设备的智能化及可互连和互操作性,有助于现场总线控制系统进一步向分散化、智能化、网络化方向发展.现场总线是一种开放式和分布式的新模式,是当今自动化领域技术发展的热点之一,现场总线技术的发展与应用,必将引起工业控制领域的一场深刻变革.  
CAN(控制器局域网)属于现场总线的范畴,是一种有效支持分布式控制和实时控制的串行通信网络. CAN总线具有成本低、速度快、实时性和可靠性高的特点,成为在世界范围内得到广泛使用的现场总线国际标准之一.由于其卓越的性能, CAN总线的应用范围已不再局限于汽车工业,被广泛应用于自动控制、楼宇自动化、医学设备等各个领域.因此,本文在简要介绍了现场总线技术后,重点对现场总线领域很有发展前途的CAN总线技术进行了详细阐述.  
智能控制是控制理论和技术发展的高级阶段,是一个新兴的并正在迅速发展的领域,它主要用来解决那些用传统方法难以解决的复杂系统的控制问题.本文在分析了CAN总线2.0B协议标准的基础上,给出了建立CAN总线控制网络模型的原理,在此基础上分析了CAN总线节点软、硬件的实现方法.本文采用CAN现场总线技术,设计开发了一种分布式控制系统,系统中采用了控制、通信、网络和电路设计等多领域的较先进的技术,使系统具有较完备的控制功能和丰富的监测功能,具有稳定的性能,良好的可扩展性和优良的性能价格比.  
基于CAN总线的分布式网络监控系统将CAN现场总线技术应用于核医疗设备监控、运行过程的智能化监测、保护和管理.它是由位于监控中心的主控计算机、USB-CAN转换器和位于设备各部分电子单元的控制节点构成.控制节点可以不依赖CAN网络,独立实现对设备各电子单元的监控保护功能,也可以通过CAN网络和上位机进行信息交互;USB-CAN转换卡完成CAN总线协议和USB协议之间的转换;主控计算机可以通过CAN网络与各个控制节点通信,实现集中管理功能;从而形成分散监控和集中管理于一体的分布式监控结构.  
论文主要内容概括如下:  
(1)探究现场总线控制技术的概念和特点、网络体系结构和模型.  
(2)介绍应用的现场总线CAN(Control Area Network)总线的性能特点、分层结构.  
(3)详细阐述了CAN总线的分布式监控系统设计,主要包括整体结构设计,智能节点硬件,软件设计.  
(4)根据整体设计要求,制定CAN总线应用层协议.  
(5)通过现场功能测试实验,实验表明系统具有实时性好、抗干扰能力强、可扩展性好等特点.实现预期要求.
7. 期刊论文 [张月明, 李闯 一种流行的现场总线—CAN总线 -现代电子技术2003, 26\(24\)](#)  
主要介绍了目前较为流行的一种现场总线—CAN总线的特点、结构及通信协议等.
8. 学位论文 [李爱国 基于现场总线的电梯远程监控系统的设计与实现 2004](#)  
电梯远程监控系统是当代先进科学技术的产物,对电梯安全可靠运行起着重要的作用.因此,该文对电梯远程监控系统的合理结构加以研究,尤其是对其数据采集模块进行重点研究并加以实现.该文将现场总线应用于电梯远程监控系统中,并将CAN总线作为数据采集模块的主要手段,利用其优良的性能,实现了对电梯状态数据的采集与传输.同时,对CAN总线的应用层协议CANopen进行分析,根据协议要求设计硬件电路,并按照其规范进行单片机C语言编程,实现应用层协议.该文根据数据采集模块与监控工作站通讯方式的不同而分为几种不同的方案进行设计,利用RS232总线的简单可靠性,设计了CAN总线转RS232数据采集模块,通过RS232米实现与监控工作站的通信.采用这种方案,硬件电路简单,软件编程简洁,成本低.但是,由于RS232总线的速度低,限制了监控系统的性能的进一步提高.为此,又设计了CAN总线转USB数据采集模块,采用USB总线作为与工作站通讯的手段,应用PHILIPS公司的PDIUSB012来设计USB模块,实现了USB总线的方便、接入灵活及高速的优点,并大大的提高了系统的实时性与监控电梯的数量.为了延长CAN总线的传输距离,提高系统组网的灵活性.同时,为了提高系统的兼容性,增加对现有大多数具有RS485总线接口的电梯的监控能力,又设计了具有RS422/485总线转CAN总线能力的CAN总线中继器.该文设计的所有电路模块,都采用双MCU共享双DRAM的结构,将数据采集和协议转换、故障判断及与工作站通信的任务分别交由两个MCU承担,两个MCU通过双DRAM进行数据交互.这种结构在实时性上较之单MCU的结构有了显著的提高.最后,在实验室搭建测试平台,运行采用不同数据采集模块的电梯远程监控系统,对所设计的模块以及监控系统进行测试,验证其性能.采用CAN总线转RS232数据采集模块的电梯远程监控系统已安装于现场之中,并且稳定运行.
9. 期刊论文 [吁苗, Yu Miao CAN作为现场总线在车身控制系统中的应用 -农业装备与车辆工程2006, ""\(9\)](#)  
CAN总线也是现场总线的一种,它最初被应用于汽车的控制系统中.由于其卓越的性能,CAN总线的应用范围已不再局限于汽车工业,被广泛应用于自动控制、楼宇自动化、医学设备等各个领域.本文在分析了CAN总线的协议后,探讨了CAN总线控制系统的网络实现方案,并在此基础上研究了CAN总线节点的软、硬件的实现方法,并提出了CAN节点在实现过程中值得注意的地方.试验验证了其可行性.
10. 学位论文 [吴明宇 CAN总线及其在雷达系统中的应用 2004](#)  
该文实现了一种先进的测控总线集现场总线引入实际的工程项目中,现场总线代替传统的总线是当今总线的发展趋势. 该论文讨论了基于现场总线中的一个代表集CAN总线在雷达信号处理机中的模板开发设计问题,并重点论述了如何在接口板上用CAN总线控制器SJA1000和两种器件配合形成CAN总线的方法.文中首先介绍了相关的现场总线、CAN总线的基本概念以及CAN总线系统构成模块的设计原理和框图,并详细介绍了CAN总线的传输机制.最后提出了使用单片机89C52组成CAN总线系统的方法,讨论了基于TMS320系列DSP组成CAN总线系统的方法.该文使用VHDL和汇编语言对前者进行了详细的电路原理和逻辑原理设计,并给出各个子功能模块设计以及硬件仿真调试结果.

## 引证文献(1条)

1. [王杰 基于CAN总线的船舶电站自动控制装置的设计 \[学位论文\] 硕士 2007](#)

本文链接: [http://d.g.wanfangdata.com.cn/Thesis\\_Y591258.aspx](http://d.g.wanfangdata.com.cn/Thesis_Y591258.aspx)  
授权使用: 李桂芬(wfszkjtsg), 授权号: 0fd83da9-a155-4204-9350-9e4501050b1c

下载时间: 2010年12月7日