

【博客·**纵横**·丛书】

CAN总线

轻松入门与实践

李真花 崔健 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

内 容 简 介

本书以基于 51 单片机的 CAN 总线系统设计为讲解对象,在内容安排上兼顾理论与实践,循序渐进地将其呈现给读者。第 1~3 章主要讲述 CAN 总线底层协议规范、应用层协议以及 CAN 总线控制器和驱动器,为 CAN 总线系统设计奠定基础。第 4~9 章重点讲解 CAN 总线系统硬件设计和程序设计,以帮助读者熟悉该系统的硬件资源,也是单片机学习者很好的学习范例;同时给出 CAN 总线自发收、两点通信、多点通信、CAN-RS232 网桥、温控系统等综合实例,让读者在学习和实践中理解 CAN 总线的精髓。最后一章与读者分享作者的一些设计感悟,并对网友常见问题进行解答。

与本书相关的学习资料、电路原理图以及实验例程,可以在 CAN 总线学习小组和书友会 <http://group.ednchina.com/684/> 下载。

本书适合 CAN 总线设计的初学者、提高者,以及对 CAN 总线感兴趣的所有电子爱好者阅读。

图书在版编目(CIP)数据

CAN 总线轻松入门与实践 / 李真花,崔健编著. --
北京:北京航空航天大学出版社,2011.1

ISBN 978-7-5124-0268-3

I. ①C… II. ①李… ②崔… III. ①总线—技术
IV. ①TP336

中国版本图书馆 CIP 数据核字(2010)第 232844 号

版权所有,侵权必究。

CAN 总线轻松入门与实践

李真花 崔健 编著

责任编辑 刘星

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@gmail.com 邮购电话:(010)82316936

印刷有限公司印装 各地书店经销

*

开本:787×960 1/16 印张:16 字数:358 千字

2011 年 1 月第 1 版 2011 年 1 月第 1 次印刷 印数:4 000 册

ISBN 978-7-5124-0268-3 定价:32.00 元

前言

CAN 总线是近些年来非常流行的几种现场总线之一。CAN 总线是一种多主方式的串行总线,可以组建多主对等的总线通信系统;具有非破坏性总线仲裁技术,让优先级高的信息得到更加快速的处理;具有强大的错误检测机制,可以检测到总线上的任何错误;采用短帧结构、位填充和 CRC 校验等措施,使传输具有高可靠性。这些优点使 CAN 总线在众多工业领域,尤其是汽车、航天等产业中得到了广泛的推广和应用。

一、如何开始你的 CAN 总线学习

回想在做本科设计的时候,第一次接触到 CAN 总线,那个时候市面上讲解 CAN 总线技术的书籍很少,尤其缺少一种针对在校学生或是初学者的入门书籍。很多书籍的协议和理论讲解占其内容的大部分,而且这些协议和理论比较难理解,读起来是一头雾水;另外这些书籍的实例也比较难,不太适合初学者。

当时书中没有简单的实例可以参考,身边也没有这方面的高手指点,因此就只能一遍遍地阅读琢磨 CAN 总线的协议和理论,一次次地修改程序并进行验证性试验。当两个 CAN 总线节点第一次通信成功时,感觉无比兴奋;然后再进行不断的细化完善,使设计的通信系统更加稳定可靠。回头再阅读 CAN 总线协议,感觉它已经不像当初那么隐晦枯燥了,也真实地体会到 CAN 总线比 RS485 更加简单好用。

由于当初学习环境的限制,使得学习 CAN 总线的过程漫长且曲折,相信很多初学者都有与 PIAE 类似的学习经历和感受。后来,经过与许多 CAN 总线学习者的交流和沟通,PIAE 认为学习 CAN 总线是一个循序渐进的过程,不可急于求成。在此与大家分享一下 CAN 总线的学习方法:

① 准备合适的学习资源。这些资源包括:完整的 CAN 总线协议;稳定可靠的 CAN 总线硬件,可以用它进行通信试验;简单的通信实例;一些设计中需要注意的事项等。

② 必须要通读一遍 CAN 总线协议,了解 CAN 总线的一些特点和主要功能。对于初学者当然不可能完全理解,也不需要完全理解。

③ 在稳定的 CAN 总线硬件上进行简单通信试验,感受 CAN 总线通信成功带来的快乐,并且结合实例逐渐掌握之前不理解的地方。

④ 多次阅读 CAN 总线协议,并且设计针对性的试验不断加深对 CAN 总线的理解。

综上所述,整个学习过程就是不断地阅读总线协议,并通过实例验证加深理解的过程。一些初学者可能不太注重总线协议的学习,只关注硬件和程序的设计,其实这是一个误区。硬件和程序都是根据总线协议设计的,它们只是协议的实现实例,是初学者学习和实践的工具。因此,建议大家要注重 CAN 总线协议的学习和验证。

二、本书的由来

读研究生的时候,很多师弟师妹们问 PIAE 关于 CAN 系统设计的问题,有许多问题都被重复提到,所以当时就想干脆为实验室写个设计指南的小文章,省得自己总是回答同一个问题。碰巧这个时候,Lief 提出能不能做一个单片机的开发板来给初学者学习,PIAE 立即同意,并建议把 CAN 总线的功能加上,于是最终设计了一个 CAN 总线的学习板。根据从菜鸟到入门再到后来逐渐进步的过程,我们开发了这款 CAN 总线学习板,同时写了相应的教程,并发布在 EDN 上。此板一经发售,立即得到了众多网友的大力支持,每天都有很多网友给我们写信讨论技术问题,建立的群也迅速爆满,这个效果是我们始料未及的。

这时 EDN 刚开展助学活动,比较有名的就是 wangjin 的 51 单片机助学和射频设计助学、圈圈的 USB 设计助学等。EDN 的负责人趁热打铁,直接联系我们建议也做一个 CAN 总线助学活动,让更多的网友能从中受益。我们觉得这是一件非常好的事情,于是欣然接受,在 EDN 上成立了 CAN 总线学习小组,当时我们非常兴奋,每天工作到很晚来整理小组的文章并回答网友们的提问。随着 CAN 助学活动的开展,我们小组的人气也越来越旺,这里非常感谢 EDN 网友对我们 CAN 小组的支持。

同时,随着技术文章越来越多,有的时候我们自己都找不到想要的帖子,而且还出现的一个问题就是:仍然有很多初学者问同样的问题。这时突然收到北京航空航天大学出版社编辑给我们的来信,说我们的活动很不错,能不能像圈圈一样写一本关于 CAN 总线学习的书籍。听后我们非常高兴,因为出书可以把我们这几年的设计经验同广大的朋友分享,而且也是对自己所学东西的一次全面总结。我们从学生年代走过来,非常知道初学者想看什么书,也知道初学者需要什么样的实例。因此在写这本书的时候,完全从初学者的角度来考虑;当然对于有一定基础的读者来说,本书也可以作为设计参考。

三、本书的主要内容

全书以基于 51 单片机的 CAN 总线系统为主体,分为 10 章。在内容安排上兼顾理论与实践,循序渐进地将其呈现给读者。

前两章为 CAN 的基本理论部分,使读者对 CAN 总线有一个基本的认识,尤其是对 CAN 总线通信机制的了解。

第 3 章结合前两章的理论详细介绍本书用到的 CAN 总线控制器 SJA1000 和驱动器

82C250,为后续的实例设计打好基础。

第4章结合CAN总线学习板的硬件资源,从实用性出发来介绍相关的硬件设计。

第5章是基础实验部分,主要是想通过一些基本实验使读者熟练掌握CAN总线学习板的硬件资源、程序调试以及下载软件的使用,为后面的综合实验打好基础。

第6~9章为4个经典的CAN实例设计,从简单的自收发开始,直到实用的CAN总线温控网络系统;只要理解并且掌握了这4个实例,读者一定有能力去设计自己的CAN总线网络系统。

最后一章是作者之前在网上发表的一些设计心得,受到了很多网友的肯定,于是又重新整理收于书中;同时根据这几年收到的初学者们的一些常见问题,进行了整理,汇编在10.3节“答网友问”。

四、本书的读者对象和相关资源

本书主要针对即将步入CAN总线设计的初学者,但阅读本书也需要一定的电子技术基础和编程基础。如果读者具备单片机的基本开发技能,可事半功倍地阅读本书。同时与大家分享CAN学习板的电路图、书中所讲解的完整程序代码以及一些相关的开发小软件的下载地址,相应的资源可以到我们EDN上的博客或是CAN学习小组里去下载,也可以到北京航空航天大学出版社网站“下载中心”下载。

全书由李真花和崔健编写;李朋对全书的电路图进行了校对,并调试和验证了相关例程;赵倩和李黎在校对全文、部分素材的翻译和整理方面做了大量的工作,在此一并对他们表示感谢!由于时间有限,加之作者水平有限,书中难免有一些不足之处,欢迎广大读者批评和指正,我们会非常感谢和珍惜大家所提出的建议并尽最大的努力去改正。如果读者发现错误或者有相关建议可以到我们的博客上去留言,也可以加入我们的CAN总线学习小组和书友会来发帖进行讨论。

个人博客地址: <http://blog.ednchina.com/PIAE/>。

CAN总线学习小组和书友会: <http://group.ednchina.com/684/>。

五、致 谢

在此特别感谢北京航空航天大学出版社工作人员对本书的关心和支持,使作者有机会把自己多年的开发经验同广大读者分享;

感谢EDN的敬荣强、彩云和黄娜对CAN总线学习小组的支持,没有你们,我们不会有这么好的一个技术平台来进行交流;

感谢CAN总线学习小组的组员们,没有大家的支持,小组不会这么热闹,这么充满生机;最后感谢所有给过我们帮助的人,正因为有了你们的支持,才能让我们走得更远。

PIAE

2010年10月于北京

目 录

第 1 章 CAN 总线概述与协议规范	1
1.1 计算机网络体系结构与拓扑结构	1
1.1.1 计算机网络体系结构	1
1.1.2 网络互联设备	6
1.1.3 网络拓扑结构	7
1.2 CAN 总线简介	9
1.2.1 CAN 总线是什么	9
1.2.2 CAN 总线的特点	10
1.2.3 CAN 总线传输介质	11
1.2.4 CAN 总线拓扑结构与设备	13
1.3 报文传输	14
1.3.1 帧类型	14
1.3.2 帧格式	15
1.3.3 帧优先级仲裁	21
1.4 报文滤波与校验	22
1.5 编码——位填充	23
1.6 错误处理与故障界定	23
1.6.1 错误类型	23
1.6.2 节点错误处理	25
1.6.3 故障界定方法	25
1.7 位定时要求	26
本章小结	29
第 2 章 CAN 总线应用层协议	30
2.1 为什么构建 CAN 应用层协议	30

2.2	常用的 CAN 总线应用层协议	30
2.3	iCAN 总线协议概要	32
2.4	iCAN 协议的报文格式	33
2.4.1	iCAN 报文标识符分配	34
2.4.2	iCAN 报文数据部分定义	36
2.4.3	iCAN 报文格式详细说明	38
2.5	iCAN 协议的报文传输协议	42
2.5.1	iCAN 协议通信模式	42
2.5.2	iCAN 协议报文处理流程	45
2.6	iCAN 协议的设备定义	46
2.6.1	I/O 资源说明	47
2.6.2	配置资源说明	49
2.7	iCAN 网络管理	51
2.7.1	节点控制	52
2.7.2	通信控制	53
	本章小结	56
第 3 章	CAN 控制器和驱动器	57
3.1	CAN 控制器和驱动器作用	57
3.2	CAN 总线控制器和驱动器选型	57
3.3	CAN 控制器 SJA1000 芯片详述	60
3.3.1	芯片 SJA1000 性能	60
3.3.2	SJA1000 的引脚定义	61
3.3.3	SJA1000 的内部结构及各个模块功能	62
3.3.4	BasicCAN 和 PeliCAN 模式的区别	64
3.3.5	BasicCAN 的寄存器及其功能详述	65
3.3.6	PeliCAN 的寄存器及其功能详述	77
3.3.7	公共寄存器	103
3.4	CAN 总线驱动器 82C250 详述	108
3.4.1	82C250 特性	108
3.4.2	82C250 功能框图	109
3.4.3	82C250 功能详述	110
	本章小结	111
第 4 章	硬件系统设计与实践	112
4.1	总线通信系统的硬件构成	112

4.2	CAN 总线学习板介绍	114
4.3	单片机及其最小系统	115
4.3.1	回顾一下老朋友——五彩缤纷的单片机	115
4.3.2	单片机最小系统设计	118
4.3.3	复位电路	119
4.3.4	时钟电路	121
4.3.5	EA 引脚的设置	122
4.4	系统人机界面设计	122
4.4.1	数码管显示设计	122
4.4.2	LED 灯显示设计	124
4.4.3	按键接口设计	127
4.5	电源部分	128
4.6	RS232 串口通信接口设计	132
4.6.1	RS232 总线简介	132
4.6.2	RS232 通信电路设计	133
4.7	RS485 通信接口设计	136
4.7.1	RS485 总线简介	137
4.7.2	RS485 通信电路设计	138
4.8	单总线温度传感器——DS18B20	139
4.9	继电器及无线扩展口部分	139
4.10	SJA1000 与单片机的连接设计	141
4.11	电路的安装、焊接与调试	143
4.12	CAN 总线系统的抗干扰设计	147
4.12.1	电源和地隔离技术	147
4.12.2	输入/输出通道隔离技术	148
4.12.3	PCB 设计的一些注意事项	148
4.12.4	软件抗干扰技术	150
	本章小结	150
第 5 章	基础实验实践	151
5.1	开发需要哪些软件	151
5.1.1	Keil 开发环境简介	151
5.1.2	如何建立一个工程	151
5.1.3	STC 单片机下载软件使用	157
5.1.4	串口通信软件使用	159

5.2 第1个实例：点亮一盏“灯”	160
5.2.1 实例讲解	160
5.2.2 程序设计	160
5.2.3 操作调试及结果	161
5.3 第2个实例：数码管显示	161
5.3.1 实例讲解	161
5.3.2 程序设计	161
5.3.3 操作调试及结果	163
5.4 第3个实例：用按键实现中断	163
5.4.1 实例讲解	163
5.4.2 程序设计	164
5.4.3 操作调试及结果	166
5.5 第4个实例：与PC机通信——串口驱动编写	166
5.5.1 实例讲解	166
5.5.2 程序设计	166
5.5.3 操作调试及结果	171
5.6 第5个实例：RS485通信	171
5.6.1 实例讲解	171
5.6.2 程序设计	171
5.6.3 操作调试及结果	176
5.7 第6个实例：电子温度计	177
5.7.1 实例讲解	177
5.7.2 程序设计	177
5.7.3 操作调试及结果	183
本章小结	183
第6章 CAN总线节点的自收发实例设计	184
6.1 系统设计目的及要求	184
6.2 程序设计	184
6.2.1 实例功能分析	184
6.2.2 程序流程规划	185
6.2.3 CAN节点初始化	186
6.2.4 CAN节点发送程序设计	188
6.2.5 CAN节点接收程序设计	190
6.3 系统调试与结果	191

6.4 扩展实例：自接收模式参数验证	192
6.4.1 试验目的和要求	192
6.4.2 配置参数及验证结果	192
本章小结	195
第7章 CAN 总线两节点通信实例设计	196
7.1 系统设计目的和要求	196
7.2 通信协议设计	196
7.2.1 协议报文格式	197
7.2.2 通信模式	198
7.2.3 基于本节实例的参数设计	198
7.3 通信程序设计	199
7.3.1 程序流程规划	199
7.3.2 CAN 节点初始化	200
7.3.3 CAN 节点发送程序设计	207
7.3.4 CAN 节点接收程序设计	208
7.4 系统调试与结果	210
7.5 扩展实例：多节点通信	210
7.5.1 试验目的和要求	210
7.5.2 配置参数	211
7.5.3 验证	211
本章小结	212
第8章 CAN - RS232 网桥设计	213
8.1 系统设计目的和要求	213
8.2 CAN - RS232 网桥硬件结构	213
8.3 CAN - RS232 网桥通信协议设计	214
8.4 程序流程设计	215
8.5 扩展实例：CAN 总线简单分析仪设计	218
本章小结	219
第9章 基于 iCAN 协议的温控系统设计	220
9.1 系统设计目的和要求	220
9.2 系统网络结构	221
9.3 通信协议和系统网络参数配置	222
9.4 检测节点设计	223

9.4.1 硬件设计方案	223
9.4.2 节点程序流程设计	225
9.5 上层软件设计	227
本章小结	227
第 10 章 感悟设计	228
10.1 培养我们的项目工程意识	228
10.1.1 电子类项目的指标及其分类	228
10.1.2 如何进行资料搜索	230
10.1.3 硬件及软件设计的一些建议	231
10.1.4 开始你的文档整理	232
10.2 关于电子类学生如何在大学中学习	233
10.2.1 关于专业	233
10.2.2 关于电子类专业学生的学习	234
10.2.3 关于单片机学习	236
10.3 答网友问	237
10.3.1 问题 1——单片机型号兼容	237
10.3.2 问题 2——远程帧功能	237
10.3.3 问题 3——SJA1000 的读/写控制	238
10.3.4 问题 4——SJA1000 初始化程序	239
10.3.5 问题 5——CAN 总线的关闭	239
10.3.6 问题 6——广播通信功能	239
10.3.7 问题 7——PCA82C250 发热	239
附 录 邮政系统与 CAN 总线通信系统对比	241
后 记	243
参考文献	244

CAN 总线概述与协议规范

本章首先简单介绍计算机网络体系架构和拓扑结构,让读者了解网络中每层的功能和网络节点的连接方式,为将要讲解的 CAN 总线协议做准备;然后重点讲解 CAN 总线的特点和协议规范。

1.1 计算机网络体系结构与拓扑结构

本节重点介绍网络体系结构、网络拓扑结构以及在网络连接中用到的设备。

1.1.1 计算机网络体系结构

计算机网络的体系结构是指计算机网络层次结构模型和各层协议的集合,也就是计算机网络及其部件所应实现的功能定义和抽象。网络协议是指为了进行计算机网络中的数据交换而建立的规则、标准或约定的集合,是计算机网络中不可缺少的组成部分。它包括语法、语义和时序 3 个要素。语法是指用户数据与控制信息的结构和格式;语义是指需要发送何种控制信息,以及完成的动作与所做出的响应;时序是指对事件实现顺序的详细说明。

国际标准化组织(ISO)在 1977 年成立了专门机构来研究网络体系结构和网络协议的国际标准化问题。在此不久,ISO 就提出了开放系统互连参考模型 OSI-RM(Open Systems Interconnection Reference Model),“开放”是指遵循 OSI 标准的任何系统之间均可通信,“系统”是指各系统中与互联有关的部分。这一标准定义了网络互联的 7 层框架,ISO 开放系统互联模型实现了系统间的互联性、互操作性和可移植性。

开放系统互联模型是一个抽象的概念,在 OSI 参考模型中,包括了体系结构、服务定义和协议规范 3 级抽象,如图 1.1.1 所示。

OSI 体系结构定义了一个 7 层模型,用以进行进程间的通信,并作为一个概念性框架来协调各层标准的制定;OSI 服务定义了每一层提供的服务,某一层的服务是指该层及其以下各层提供给上一层的服务,层间的服务通过定义好的层间抽象接口完成,交互时使



图 1.1.1 OSI 的 3 个抽象层次图

用服务原语,各种服务不考虑服务的具体实现;OSI 协议规范说明控制信息的内容。

OSI 整个网络分为 7 层,划分的原则是:

- ① 网络各节点都有相同的层次,相同层次具有相同的功能。
- ② 同一节点内相邻层次间通过接口通信。
- ③ 每一层使用下层提供的服务,并向上层提供服务。
- ④ 不同节点的同等层按照协议实现对等层之间的通信。

OSI 开放系统互联参考模型的 7 层次依次为:物理层(Physical layer)、数据链路层(Data-link layer)、网络层(Network layer)、传输层(Transport layer)、会话层(Session layer)、表示层(Presentation layer)、应用层(Application layer),如图 1.1.2 所示。

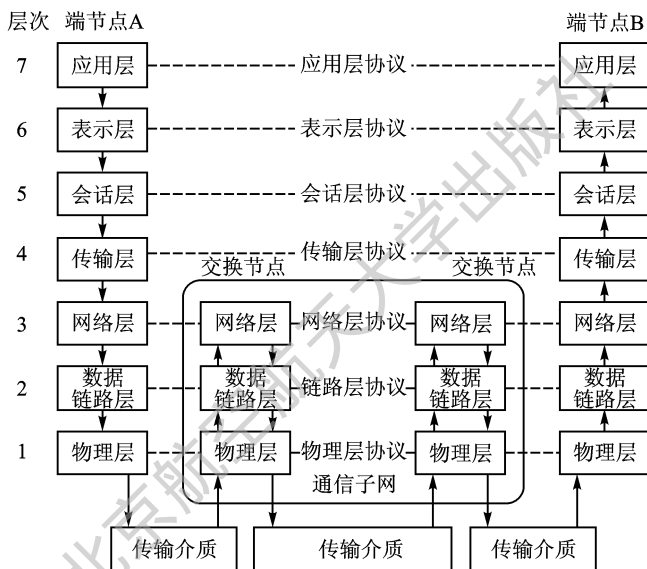


图 1.1.2 OSI 参考模型分层结构图

层次越靠上,其与信息处理的关系越密切;层次越靠下,其与通信的关系越密切。因为在两个端点通信时,上层使用下层提供的服务,同一层通过对应的协议通信,所以同一层次通信时,并不关心下层的具体实现,这就是网络通信中的“相对透明”。

下面介绍每层的功能,并与日常生活中熟悉的信件邮递系统对比说明。邮政系统与 CAN 总线通信系统对比见附录。

(1) 物理层

物理层并不是物理媒体本身,它是开放系统利用物理传输介质实现物理连接的功能描述和执行连接的规程。物理层提供用于建立、保持和断开物理连接的机械的、电气的、功能的和规程的条件。总之,物理层为数据链路层提供物理连接,透明地传输比特流。

物理层的功能主要包括:物理连接的建立与拆除,当数据链路层实体提出建立连接请求时,物理层使用相关的协议完成连接的建立过程,在数据信号传输过程中维持这个连接,传输结束后拆除这个连接;物理层数据单元的传送,物理层定义了编码的类型、位同步方式、数据传输速率,采用的单工、半双工、全双工传输方式也在物理层说明;另外,物理层还包括传输中出现差错后的处理等管理功能。通信系统的物理层就如同邮政系统使用的公路、水路、航空等交通路线;其单工、半双工、全双工传输方式是系统布线方面的考虑,而目前邮政系统的交通比较发达,资源比较丰富,完全采用全双工或者多双工方式;通信系统物理层的驱动能力,就好像邮政系统使用的汽车、轮船、飞机等交通工具的运输能力。

物理层接口协议重点定义了通信设备的一些特性,这些特性包括:机械特性、电气特性、功能特性和规程特性。其中机械特性详细规定了插头和插座的形状和尺寸,插针或者插孔的数目及其排列,固定或锁定装置等。电气特性规定了在物理连接传输二进制比特流时线路上信号电平高低,驱动器与接收器的阻抗及阻抗匹配、传输速率与接口线距离限制等。功能特性规定了设备间各条接口信号线的功能分配和确切定义。规程特性规定了利用信号线进行二进制比特流传输的一组操作过程。常用的物理接口标准有 RS232、RS422、RS485 等。

(2) 数据链路层

数据链路层是 OSI 参考模型中的第 2 层,介于物理层和网络层之间。设立数据链路层的目的是在物理层提供的物理链路连接和比特流传输功能的基础上,为网络层之间建立、维持和释放点对点的数据链路连接和传输提供方法。总之,数据链路层的功能是为网络层提供透明的、可靠的数据传输服务。数据链路层通常分为上下两个子层:逻辑链路控制(LLC, Logic Link Control)子层和介质访问控制(MAC, Media Access Control)子层。

数据链路层的主要功能包括:链路管理、帧同步、寻址、访问控制、差错控制和流量控制、透明传输。通信系统的数据链路层功能如同邮政系统中信封上的信息功能,用于标识该信件的写信地址和收信地址等。

① 链路管理。

链路管理是指在两端点通信之前,通过必要的信息交换建立起数据链路连接;在数据传输阶段要维持该数据链路的连接;在数据传输结束后还要释放这一连接。数据链路的建立、维持和释放称为链路管理。

② 帧同步。

数据链路层传输数据的单元称为帧,它将物理层传输的比特流按一定格式分割形成信息块。帧同步就是接收方可以从接收到的比特流中准确区分出一帧的开始和结束,确定帧的边界位置。

③ 寻址。

数据链路层的帧包含必要的信息部分,这部分保证每一帧都能发送到正确的目的站,而且目的站也能知道发送方是哪个站。

④ 访问控制。

当多个设备连接到同一条链路上时,数据链路层协议能够决定出哪个设备可以取得链路
的控制权。

⑤ 差错控制和流量控制。

由于信道中干扰的存在,就会不可避免地出现数据传输错误,数据链路层可以通过纠错或
检错重发两种办法实现差错控制,差错控制还包括传输过程中帧的丢失和帧的重复接收处理;
流量控制是发送方的发送速率高于接收方的接收速率时,为避免由于过载造成的数据丢失而
采取的控制措施。

⑥ 透明传输。

透明传输是指无论链路上传输的是何种比特组合,都能够正常传输。当所传数据中的比
特组合恰巧与某个控制信息完全一样时,必须采取有效措施,使接收方不会误将数据信息当作
控制信息,从而保证数据链路层的透明传输。根据传输中的信息的基本单位,数据链路层的传
输控制分为面向字符型的数据链路控制和面向比特型的数据链路控制。

(3) 网络层

网络层是 OSI 参考模型中的第 3 层,介于数据链路层和传输层之间。设立网络层的目的
是:使报文分组以最佳路径通过通信子网达到目的主机,网络用户不必关心网络拓扑结构及
所使用的通信介质,通过网络层的控制实现不同网络之间的数据交换。网络层的主要功能包
括:寻址;路由选择与中继;流量控制;网络连接建立与管理。通信系统的网络层类似于邮政系
统的邮局,为所收集到的信件选择最佳的路径到达目的地址。

① 寻址。

数据链路层完成的寻址是处理网络内的寻址问题,如果传输信息的发送者和接收者属于
不同的网络,就需要网络层参与解决寻址问题。

② 路由选择与中继。

当传输的数据单元经过通信子网时,各个中继节点在存储转发数据的过程中,采用合适的
路由选择算法,保证数据单元以某些指标最优的方式通过子网,路由选择就是网络层的主要功
能之一。

③ 流量控制。

网络层的流量控制是对进入分组交换网的通信量进行控制,使通信子网稳定运行,尽量防
止通信量过大造成通信子网性能下降。

④ 网络连接建立与管理。

网络层的服务可以划分为面向连接的网络服务(CONS, Connection-Oriented Network
Service)和无连接的网络服务(CLNS, Connectionless Network Service)。在面向连接服务中,
网络连接是指传输实体之间传递数据逻辑、贯通通信子网的端对端通信通道。

(4) 传输层

传输层是 OSI 参考模型中的第 4 层,介于网络层和会话层之间。在 OSI 参考模型中,通常称 1~3 层的功能为低层功能,它们是通信网络完成数据通信功能的集合;通常称 5~7 层的功能为高层功能,它们是由各端系统内部进程完成的面向应用的集合。传输层位于通信子网和资源子网之间,是低层与高层之间的特殊一层,从面向通信和面向信息处理角度来看,传输层应划在低层;而从用户功能和网络功能角度分类,传输层应划在高层。

设立传输层的目的是通过补充和完善下层网络通信服务质量的差异和不足,向上一层提供统一服务质量(QOS)的、透明的数据传输服务。总之,传输层为主机进程之间提供可靠的端对端通信,向会话层提供独立于网络的运输服务。因此传输层的主要功能是:对一个进行的对话或连接提供可靠的运输服务,在通向网络的单一物理连接上实现该连接的复用,在单一连接上提供端到端的序号与流量控制、端到端的差错控制及恢复等服务。该层的功能类似于邮政系统为了更好地实现信件的传递而增加的一些其他辅助工具或者措施。

(5) 会话层

会话层是 OSI 参考模型中的第 5 层,介于传输层和表示层之间。设立会话层的目的是:在两个应用进程进行相互通信时组织它们的会话,为它们之间的交互过程创建一个十分协调的环境。会话层的主要功能包括:提供会话双方之间的会话连接建立、数据传送和释放功能;管理会话双方的对话活动,主要是对令牌管理及单工、半双工或全双工数据传送方式的设立;在数据传送中插入适当的同步点,发生差错时,会话可以在双方同意的同步点重新开始;适当中断一个对话,在预定好的同步点重新开始。通信系统会话层的作用类似于在信件交流中通用规则的作用。比如通常情况下,当甲给乙发了一封信后,乙会给甲回一封信;而当乙收到银行发的账单信函时,乙不会回信给银行。

(6) 表示层

表示层是 OSI 参考模型中的第 6 层,介于会话层和应用层之间。设立表示层的目的是:处理通信中的语法,解决通信双方之间的数据表示问题,使描述的数据结构与机器无关,按照一些编码规则定义,在通信中传送这些信息所需要的编码。因此表示层的功能包括:语法转换(将抽象语法转换成传输语法,并在对方实现相反的反转换);语法协商(根据应用层的要求协商选用合适的上下文,即确定传输语法并传送);连接管理(利用会话层服务建立表示连接,管理在这个连接之上的数据传输和同步控制,以及正常或异常地终止这个连接)。通信系统的表示层类似于信件中使用的语言。如果写信双方在信件中使用对方不懂的语言,他们可能能够收到对方的来信,但不能理解对方的意思。

(7) 应用层

应用层是 OSI 参考模型中的第 7 层,也是最高的一个功能层。其直接面向用户,是计算机网络与本地操作环境和应用系统间的界面。功能包括:为应用进程访问 OSI 环境提供手段;为应用进程提供服务。应用层提供的服务包括文件传输访问和管理(FTAM)、电子文电

处理系统(MHS)、虚拟终端协议(VIP)、联系控制服务(ACSE)、可靠传输服务(RTSE)和远程操作服务元素(ROSE)等。通信系统的应用层类似于该信件的功能,比如信件可以分为个人一般信函、账单信函、邀请信函等。

1.1.2 网络互联设备

网络互联是指将不同的网络或相同的网络用互联设备连接在一起而形成一个范围更大的网络,实现网络资源的共享;也可以是为增加网络性能和易于管理而将一个原来很大的网络划分为几个子网或网段。

要实现网络互联必须实现以下几个方面:

- ① 在互联的网络提供链路,至少有物理线路和数据链路;
- ② 在不同网络节点的进程之间提供适当的路由来交换数据;
- ③ 提供网络记账服务,记录网络资源使用情况;
- ④ 提供各种互联服务的同时应尽可能不改变网络的结构。

网络互联类型主要有:局域网与局域网互联;局域网与广域网互联;局域网通过广域网与局域网的互联;广域网与广域网的互联等。要实现网络互联通常需要以下设备:中继器,网桥,路由器及网间协议变换器。下面将简要介绍这些设备。

(1) 中继器

中继器是用于实现物理层干线段之间的连接,通过信号的放大和整形延长网络来实现物理层的连接。

(2) 网 桥

网桥在数据链路层上实现不同网络的互联,两个网络数据链路层的协议、传输介质、传输速率都可不同。网桥是一种存储转发设备,从一个局域网接收到一个信息帧后,对信息帧头进行检验修改,然后转发到另一个局域网或者丢弃该帧。网桥一般可以实现相同类型局域网和不同类型局域网的连接,可以分隔两个网络之间的通信量,有利于改善网络的性能与安全性。

(3) 路由器

路由器在网络层上实现多个网络之间的互联,基本特征是:

- ① 路由器为两个或两个以上网络之间的数据传输提供最佳路径选择。
- ② 路由器与网桥的主要区别是:网桥独立于高层协议,向用户提供一个大的逻辑网络;路由器则是从路径选择角度为不同独立的子网用户之间的数据传输提供传输的路线,并且比网桥有更大的流量控制能力和更强的异种网互联能力。
- ③ 路由器要求节点网络层以上的各层使用相同或兼容的协议。

(4) 网 关

网关也叫网间协议变换器,是比网桥和路由器更复杂的网络互联设备。它可以实现不同

协议的网络之间的互联,包括不同网络操作系统的网络之间互联,也可以实现局域网与主机、局域网与广域网之间的互联。

1.1.3 网络拓扑结构

网络的拓扑结构是抛开网络物理连接来讨论网络系统的连接形式,网络中各站点相互连接的方法和形式称为网络拓扑。网络的拓扑结构与网络的性能、可靠性以及成本有很大关系。常见的计算机网络拓扑结构有:星型拓扑、总线拓扑、环型拓扑、树型拓扑及网型拓扑。下面分别介绍这几种拓扑结构的特点。

(1) 星型拓扑

星形拓扑是由中央节点和通过点到点通信线路接到中央节点的各个站点组成,如图 1.1.3 所示。中央节点负责整个网络的通信控制,是网络的控制中枢,可以与网络中任何节点进行通信。非中央节点的两个节点不能直接通信,但可以在中央节点的控制下,通过中央节点进行数据交换。

星型拓扑结构的优点为:控制简单;故障诊断和隔离容易;方便服务。其缺点为:电缆长度和安装工作量可观;中央节点的负担较重,形成瓶颈;各站点的分布处理能力较低。

(2) 总线拓扑

总线结构使用同一媒体或电缆作为传输介质,所有的站点都通过相应的硬件接口直接连接到该传输介质上,如图 1.1.4 所示。总线上的任何一个站点都可以向总线上发出数据,发出的数据信号沿着传输介质传播,总线上的所有站点都可以接收到数据信号。一般在发出的数据报文中携带有目的地址,目的站点识别到数据报文中的目的地址与自己的地址相符时,就复制数据报文中的内容。总线上的所有站点共享一条公用的传输介质,多个站点同时传输数据时会造成信号的混乱,因此必须采用一定的控制策略保证同一时刻只有一个节点在占用总线。

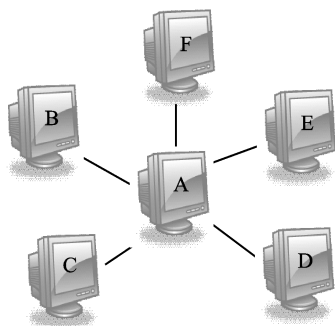


图 1.1.3 星型拓扑结构图

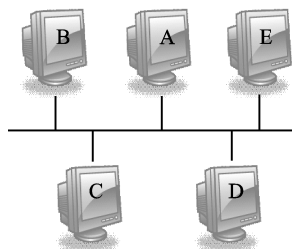


图 1.1.4 总线拓扑结构图

总线拓扑结构的优点为：总线结构所需电缆数量少；结构简单又是无源工作，有较高的可靠性；易于扩充，增减用户方便。其缺点为：传输距离有限，通信范围受到限制；故障诊断和隔离困难；分布式协议不保证信息及时传送，不具备实时功能；站点必须是智能的，要有媒体访问控制功能，增加站点软件和硬件的开销。

(3) 环型拓扑

在环型拓扑网络中，每一个节点都有输入端口和输出端口，节点间通过线路首尾相连，组成一个闭合的环路，如图 1.1.5 所示。每个站点能够接收从一条链路传来的数据，并以相同的速率串行地把该数据送到另一端链路上，数据一般以一个方向传输。和总线型网络类似，环型网络中传输的数据报文也带有目的地址，数据沿环路的节点依次传递，目的节点在识别到带有自己地址的数据报文时，在转发到同时也将其中的数据复制，这一数据报文最终回到数据发送节点。网络中各个节点共享这一环路信道，因此也需要一定的控制策略保证只有一个节点使用该信道。

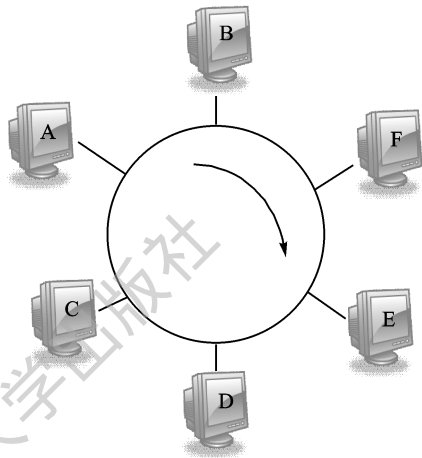


图 1.1.5 环型拓扑结构图

环型拓扑的优点为：电缆长度短；增加或减少工作站时，仅需简单的连接操作；可使用光纤作为通信介质。其缺点为：节点的故障会引起全网故障；故障检测困难；环型拓扑结构的媒体访问控制协议都采用令牌传达室传递的方式，在负载很轻时，信道利用率相对来说就比较低。

(4) 树型拓扑

在树型拓扑结构中，整个网络形状如同一棵倒置的树，顶端相当于树根，树根以下带分支，每个分支还可带子分支，如图 1.1.6 所示。每个节点只能和它的根节点或子节点直接通信，同一层次的节点不能直接通信，各个层次的根节点负责与其子节点间的通信控制。

树型拓扑结构的优点为：易于扩展，故障隔离较容易。其缺点为：各个节点对根的依赖性太大。

(5) 网型拓扑

在网型拓扑结构中，每个节点都通过多条线路与网中的其他节点连接，如图 1.1.7 所示。

网型拓扑结构的优点为：具有很高的可靠性。其缺点为：结构复杂，成本较高；节点必须具有路由选择算法和流量控制，要求较高。

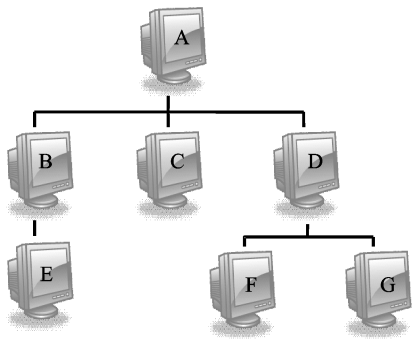


图 1.1.6 树型拓扑结构图

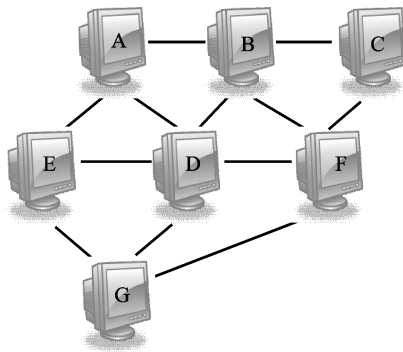


图 1.1.7 网型拓扑结构图

1.2 CAN 总线简介

本节从总体上介绍 CAN 总线的特点、CAN 总线网络结构及设备。

1.2.1 CAN 总线是什么

CAN(Controller Area Network)属于现场总线的范畴,是一种高性能、高可靠性、易开发和低成本的串行通信总线。

CAN 是由德国 BOSCH 公司在 1986 年为汽车监测和控制而设计的。由于其高性能、高可靠性、实时性等优点,随后也广泛应用于工业自动化、多种控制设备、交通工具、医疗仪器以及建筑、环境控制等众多领域,其在每个领域的广泛使用促进了标准化的进程。

1991 年 9 月 PHILIPS SEMICONDUCTORS 制订并发布了 CAN 技术规范 V2.0,该技术规范包括 A 和 B 两部分。2.0A 给出了曾在 CAN 技术规范版本 1.2 中定义的 CAN 报文格式,提供 11 位地址;而 2.0B 给出了标准的和扩展的两种报文格式,提供 29 位地址。此后,1993 年 11 月 ISO 正式发布了 CAN 的国际标准 ISO11898。除了 CAN 协议外,它也规定了最高为 1 Mbps 波特率时的物理层。同时,在 ISO11519-2 中规定了 CAN 数据传输中的容错方法。1995 年,ISO11898 标准进行了扩展,以附录的形式说明了 29 位的 CAN 标识符,其中,ISO11898-1 说明了“CAN 数据链路层”;ISO11898-2 定义了“非容错 CAN 物理层”;ISO 11898-3 规定了“容错 CAN 物理层”。

CAN 总线协议只定义了物理层和数据链路层标准,要将 CAN 总线应用于工程项目和产品中必须制定上层的应用协议。目前典型的应用协议有:SAE J1939/ISO11783、CANOpen、CANaerospace、DeviceNet、NMEA 2000 等。

1.2.2 CAN 总线的特点

CAN 是一种多主方式的串行通信总线,基本设计规范要求有高的位速率、高抗电磁干扰性能,而且能够检测总线上的任何错误,因此其具有高速性、高可靠性等特点。下面具体介绍 CAN 总线的特点,并与大家熟悉的 RS485 总线进行对比,进一步了解 CAN 总线的优势。

(1) CAN 总线的特点

① 多主工作方式。网络上的任一节点均可主动发送报文;在总线空闲时,所有的节点都可开始发送报文;最先访问总线的节点可获得发送权;多个节点同时发送时,依据报文的优先权而不是节点的优先权进行总线访问控制。

② 非破坏性总线仲裁技术。当总线发生冲突时,高优先级报文可以不受影响地进行传输,保证了高优先级报文的实时性要求;而低优先级的报文退出传输。

③ 具有点对点、一点对多点及全局广播等多种传输方式。

④ 远程数据请求。CAN 总线可以通过发送“远程帧”,请求其他节点的数据。

⑤ 高效的短帧结构。每个数据帧数据域最长为 8 字节,传送短报文时效率高。

⑥ 高可靠性。短帧传输时间短,受干扰概率低;每帧都有位填充、CRC 校验等措施,保证了极低的出错率;发送期间丢失仲裁或者由于出错而遭破坏的帧可自动重发。

⑦ 自动关闭。CAN 总线可以判断出总线上错误的类型是暂时的数据错误(如外部噪声等)还是持续的数据错误(如单元内部故障、驱动器故障、断线等)。当节点发生持续数据错误时,可自动关闭,脱离总线。

⑧ CAN 总线上的节点数取决于总线驱动电路,目前可达 110 个。在标准帧报文标识符有 11 位,而在扩展帧的报文标识符(29 位)的个数几乎不受限制。

⑨ 总线配置灵活。

(2) CAN 总线与 RS485 总线比较

CAN 是一种有效支持分布式控制或实时控制的串行通信网络。较之 RS485 基于 R 线构建的分布式控制系统而言,基于 CAN 总线的分布式控制系统在以下方面具有明显的优越性。

① CAN 控制器工作于多主方式,网络中的各节点都可根据总线访问优先权(取决于报文标识符)采用无损结构的逐位仲裁方式竞争向总线发送数据,且 CAN 协议废除了站地址编码,而代之以对通信数据进行编码,这可使不同的节点同时接收到相同的数据。这些特点使得 CAN 总线构成的网络各节点之间的数据通信实时性强,并且容易构成冗余结构,提高系统的可靠性和灵活性。而利用 RS485 只能构成主从式结构系统,通信方式也只能以主站轮询的方式进行,系统的实时性、可靠性较差。

② CAN 总线控制器通过驱动器芯片的两个输出端 CANH 和 CANL 与物理总线相连,而 CANH 端的状态只能是高电平或悬浮状态,CANL 端只能是低电平或悬浮状态。这就保证不会在 RS485 网络中出现以下现象:当系统有错误出现多节点同时向总线发送数据时,导致总

线呈现短路,从而损坏某些节点。而且 CAN 节点在错误严重的情况下具有自动关闭输出的功能,以使总线上其他节点的操作不受影响,从而保证在网络中不会因个别节点出现问题使得总线处于“死锁”的状态。

③ CAN 具有完善的通信协议,可由 CAN 控制器芯片及其接口芯片来实现,从而大大降低了系统的开发难度,缩短了开发周期,这些特点是仅具有电气协议的 RS485 所无法比拟的。

④ 其他性能方面的比较如表 1.2.1 所列。

表 1.2.1 RS485 与 CAN 性能对比

特 性	RS485	CAN	特 性	RS485	CAN
单点成本	低廉	稍高	通信失败率	高	极低
系统成本	高	较低	节点错误的影响	导致整个网络的瘫痪	无任何影响
总线利用率	低	高	通信距离	<1.5 km	可达 10 km(5 kbps)
网络特性	单主网络	多主网络	网络调试	困难	非常容易
数据传输率	低	高	开发难度	标准 Mod-bus 协议	标准 CAN-bus 协议
容错机制	无	可靠的错误处理和检错机制	后期维护成本	高	低

1.2.3 CAN 总线传输介质

表示“隐性”和“显性”逻辑的能力是 CAN 总线仲裁方法的基本先决条件,即所有节点都为“隐性”逻辑时,总线介质才处于“隐性”状态;只要有一个节点发送了“显性”逻辑,总线就呈现为“显性”状态。使用电气介质时,高电平为“隐性”状态,低电平为“显性”状态。使用光学介质时,“暗”表示“隐性”状态,“亮”表示“显性”状态。

电气传输介质中包括以下使用方式:两线总线;单线总线;共用一条线传输信号和供电。

(1) 两线总线

两线总线提供差分信号的传输,因此可以抑制共模误差(对共模误差的抑制只能在共模范围内起作用),即使在非常低的信号电平下也能保证信号的可靠传输。在采用两线传输时,必须在总线的每一端接一个电阻(建议值为 120 Ω),避免出现信号反射。此外,通过双绞线可以补偿由电磁辐射造成的干扰,这样就提高了抗干扰的能力。当使用适当的总线错误管理功能时,即使一条线断路或者短路,通信仍然能够在抗干扰能力下降的情况下继续进行。

(2) 单线总线

单线总线在车辆的车身电器系统中应用特别广泛。该方案假定各个节点都有一个公共地,因此单线总线的解决方案只能孤立地应用。由于单线总线在没有屏蔽的情况下易受到电磁感应辐射的干扰,所以必须提高信号电平差以改善信噪比。

(3) 共用一条线输出信号和供电

目前多种现场总线都可以采用同一条线缆提供电源和数据传输, CAN 总线也可以采用这种方式。但是由于 CAN 协议使用位仲裁, 同时进行供电和数据传输是比较困难的, 所以在载波频带中使用“开关键控法”以及在基带中使用对称的显性位传输数据都是可行的方案。然而, 这种电路成本高、抗干扰能力差, 因此在实际使用中无法与带单独电源线的方案进行竞争。

在现场应用中, 使用最多的传输介质是双绞线, 是因为其具有以下特点:

- ① 技术上容易实现、造价低廉;
- ② 理论上节点数无限制, 对环境电磁辐射具有一定的抗干扰能力;
- ③ 随着频率的增长, 双绞线线对的衰减迅速增高;
- ④ 双绞线有所谓的近端串扰;
- ⑤ 适合 CAN 总线网络 5 kbps~1 Mbps 的传输速率。

ISO/DIS 11898 对线缆的直流电压参数、终端电阻和波特率有一些推荐值。与传输线缆长度相关的电缆直流参数推荐表如表 1.2.2 所列。

表 1.2.2 与传输长度相关的电缆直流参数推荐

总线长度/m	电 缆		终端电阻/ Ω	最大波特率/bps
	直流电阻/($m\Omega/m$)	导线截面积/ mm^2		
0~40	70	0.25~0.34(AWG23, AWG22)	124	1 M(40 m)
40~300	<60	0.34~0.6(AWG22, AWG20)	127	>500 k(100 m)
300~600	<40	0.5~0.6(AWG20)	127	>100 k(500 m)
600~1000	<20	0.75~0.8(AWG18)	127	>100 k(500 m)

选择双绞线电缆时要注意以下几个方面:

- ① 线长。如果工业现场通信的线缆长度较长(>0.3 m), 建议使用带屏蔽的双绞线。
 - ② 波特率。由于取决于传输线的延迟时间, CAN 总线的通信距离可能会随着波特率降低而增加。
 - ③ 外界干扰。在工业使用现场必须考虑外界干扰, 例如其他电气负载引起的电磁干扰, 尤其是大功率电机运行或其他设备开关时引起供电线路上电压变化的场合。
 - ④ 特征阻抗。所采用的传输线的特征阻抗约为 120Ω 。由于 CAN 总线接头的使用使 CAN 总线的特征阻抗可能发生变化, 所以, 不能过高估计所使用电缆的特征阻抗。
 - ⑤ 有效电阻。所使用电缆的电阻必须足够小, 以避免因线路压降过大影响位于总线末端的接收器件。为了确定接收端的线路压降, 避免信号反射, 在总线两端需要连接终端电阻。
- 根据 CAN 总线通信时对线缆的要求, 许多线缆制造商都有针对 CAN 通信的线缆, 如美国百通电缆公司和日本电线公司。表 1.2.3 列举了日本电线 CANC 系列线缆。

表 1.2.3 CANC 系列电缆规格

用途		固定布线用					可动部用
型号		CANC-22		CANC-50		CANC-75	CANC-50SF
导体尺寸/mm ²		0.22		0.5		0.75	0.5
对数		1p	2p	1p	2p	1p	1p
电缆外径/mm		6.0	8.5	8.0	11	9.0	8.0
概算质量/(kg/km)		45	75	70	125	85	70
特性阻抗		120 Ω ± 10% (@1 MHz)					
容许张力/kg		3.0	6.0	7.9	15.8	10.5	7.0
最小弯曲半径/mm	铺设时	60	85	80	110	90	80
	固定时	24	34	32	44	36	32

使用双绞线时需要注意到几个问题：

① 双绞线采用抗干扰的差分信号传输方式。

② 如果使用非屏蔽双绞线作为物理层，只需要有 2 根线缆作为差分信号线传输；如果使用带屏蔽双绞线作为物理层，除需要 2 根差分信号线连接以外，还需要注意在同一网段中的屏蔽层单点接地问题。

③ CAN 网络的两端必须有两个范围为 118~130 Ω 的终端电阻。

④ 支线必须尽量短。

⑤ 确保不要在干扰源附近布置 CAN 总线，如果因环境限制必须这样做，就应该使用双层屏蔽电缆。

⑥ 使用适当的电缆类型，必须确定电缆的电压衰减。

1.2.4 CAN 总线拓扑结构与设备

(1) CAN 中继器

CAN 中继器适用于 CAN 主网与 CAN 子网的连接，或者 2 个相同通信速率的平行 CAN 网络进行互联。CAN 通过中继器硬件电路级联，提升总线的电气信号，从而实现 CAN 报文数据的转发，可以成倍地提高 CAN 通信距离，增加 CAN 节点数目，是非常有效的 CAN 网络连接。

例如，图 1.2.1(a) 为一个连接许多节点的网络地理分布结构。线形网络所需要的总线总长度为 400 m。如果按照图 1.2.1(b) 使用中继器进行连接，网络的总长度只有 350 m，信号传输的最大距离只有 150 m。

通过该例子可以看出，使用中继器可以很好地适应地理分布的限制。中继器非常适合于

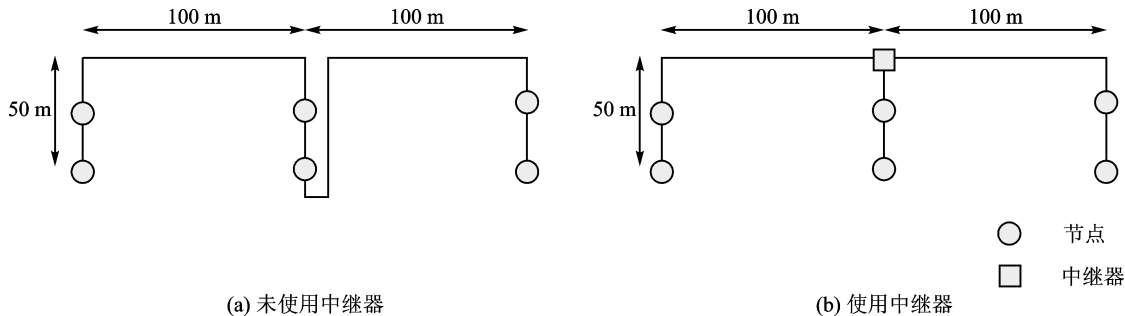


图 1.2.1 中继器使用示意图

设计优化的扩展网络拓扑结构,使用中继电器还可增加每个网络段所挂的节点数。

(2) CAN 网桥

网桥将一个独立的网络连接到数据链路层。网桥提供存储功能,并在网络段之间转发全部或部分报文;而中继器转发所有的电气信号。

通过从一个网络段向另外一个网络段转发它所需要的报文,集成了滤波功能的网桥可以实现多端网络的组织结构。

(3) CAN 网关

CAN 网关提供不同协议的网络之间的连接。CAN 网关将不同通信系统之间的协议数据节点进行转换。现在流行的 CAN 网关有:CAN 转 RS232 网关;CAN 转 RS485 网关;CAN 转 Ethernet 网关。

1.3 报文传输

在 CAN 网络中,一个发出报文的节点称为该报文的发送器,并且保持该身份直到总线空闲或丢失仲裁。如果一个节点不是这条报文的发送器,而且总线不为空闲,则该节点称为接收器。

1.3.1 帧类型

CAN 总线在报文传输中由以下 4 种不同的帧类型表示和控制:

数据帧 数据帧将数据从发送器传送到各个接收器。

远程帧 远程帧请求具有相同标识符的数据帧的发送。

错误帧 任何节点在检测到总线错误时就发送错误帧。

过载帧 过载帧用于在先行的和后续的数据帧(或远程帧)之间提供附加的延时。

1.3.2 帧格式

在 CAN2.0B 规范中,有两种帧格式,其区别主要在于标识符的长度:具有 11 位标识符的帧称为标准帧,而具有 29 位标识符的帧则称为扩展帧。数据帧和远程帧都可以使用标准帧格式或者扩展帧格式,它们通过帧间间隔与先前帧区分开。

1. 数据帧格式

数据帧由 7 个不同的部分组成:帧起始、仲裁域、控制域、数据域、CRC 域、应答域和帧结束,如图 1.3.1 所示。数据域的长度可以为 0。

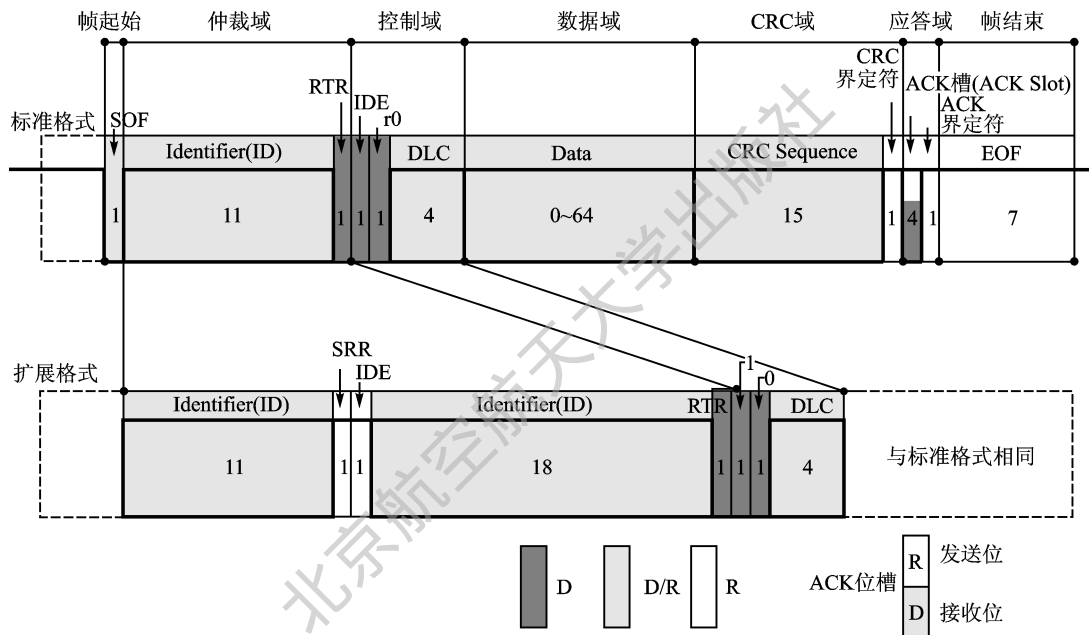


图 1.3.1 数据帧格式

(1) 帧起始

帧起始(SOF)表示数据帧的起始,由一个单独的“显性”位组成。一个 CAN 节点只在总线空闲时才允许开始发送报文,所有的节点必须同步于首先开始发送的那个节点的帧起始前沿(即总线值从隐性变为显性时产生的跳变沿)。

(2) 仲裁域

仲裁域表示数据帧的优先级。标准帧与扩展帧的仲裁域不同。

在标准帧格式中,仲裁域由 11 位标识符(ID)和远程发送请求位(RTR)组成,如图 1.3.2 所示。

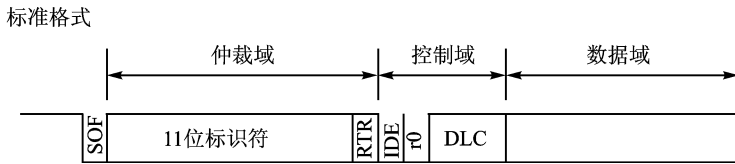


图 1.3.2 标准数据帧格式

在扩展帧格式中,仲裁域由 29 位标识符(ID)、SRR 位、标识符扩展位(IDE)、远程发送请求位(RTR)组成。扩展帧标识符由 11 位的基本 ID 和 18 位的扩展 ID 组成,如图 1.3.3 所示。

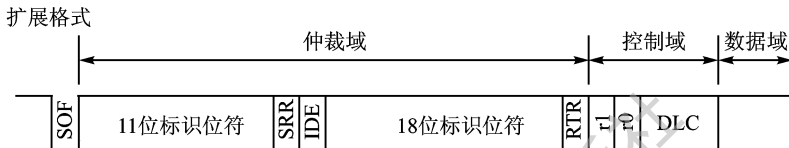


图 1.3.3 扩展数据帧格式

标识符(ID):表示数据帧的优先级。先发送标识符 ID 的高位,后发送低位。

远程发送请求位(RTR):表示该报文为数据帧或者远程帧。当为显性时,表示为数据帧;为隐性时,表示为远程帧。

代替远程请求位(SRR):SRR 位是一个隐性位。在扩展帧中 11 位标识符的下一位是 SRR 位,而在标准帧中 11 位标识符的下一位是 RTR 位,因此 SRR 称作代替远程请求位。当标准帧发送与扩展帧发送相互冲突,并且扩展帧的基本标识符与标准帧的标识符相同时,标准帧优先级高于扩展帧。

标识符扩展位(IDE):表示该报文为标准格式或者扩展格式。为显性时,表示为标准格式;为隐性时,表示为扩展格式。

(3) 控制域

控制域表示数据帧中数据域的字节长度。由保留位和数据长度码共 6 位组成,如图 1.3.4 所示。标准格式和扩展格式的控制域结构相同但位置不同。

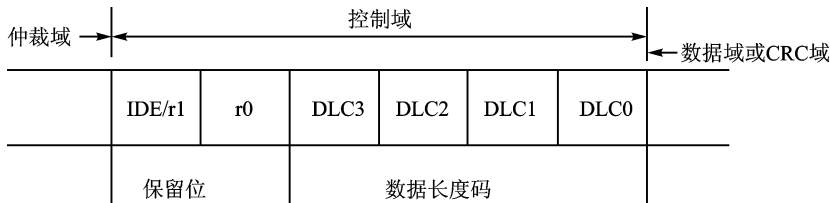


图 1.3.4 控制域结构

保留位:在标准格式里由 r0 组成;在扩展格式里由 r1(显性)和 r0(显性)组成。

数据长度码表示数据域中的字节数,由 4 位组成,采用二进制编码。数据字节数和数据长度码对应关系如表 1.3.1 所列。数据字节数只允许为 0~8,不能使用其他数值。

表 1.3.1 数据长度编码

字节数	数据长度码				字节数	数据长度码			
	DLC3	DLC2	DLC1	DLC0		DLC3	DLC2	DLC1	DLC0
0	显性	显性	显性	显性	5	显性	隐性	显性	隐性
1	显性	显性	显性	隐性	6	显性	隐性	隐性	显性
2	显性	显性	隐性	显性	7	显性	隐性	隐性	隐性
3	显性	显性	隐性	隐性	8	显性	显性	显性	显性
4	显性	隐性	显性	显性					

(4) 数据域

数据域由数据帧中要发送的数据组成。它可以包括 0~8 个字节,每字节 8 位,高位在前。

(5) CRC 域

CRC 域包括 CRC 序列和随后的 CRC 界定符,如图 1.3.5 所示。

CRC 序列:计算 CRC 序列时起始值为 0,依次计算帧起始、仲裁域、控制域、数据域。

CRC 界定符:CRC 界定符为一个单独的隐性位。

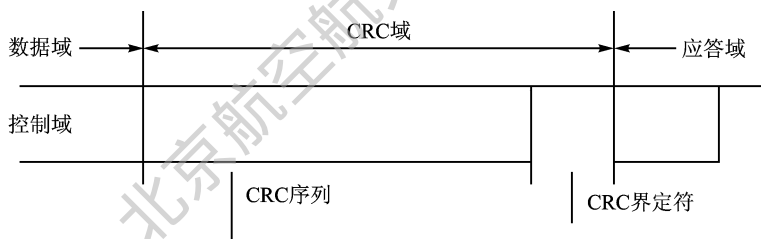


图 1.3.5 CRC 域结构

(6) 应答域

应答域长度为 2 位,包括应答位和应答界定符,如图 1.3.6 所示。

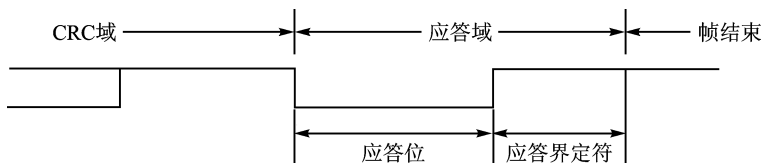


图 1.3.6 应答域结构

在应答域中,发送器发出两个隐性位。当接收器正确地接收到有效的报文,该接收器就会在应答位期间发送显性位(即发送应答)来通知发送器。

应答位:所有接收到匹配 CRC 序列的节点会在应答位期间,用一显性位写在发送器的隐性位置上作为回应。

应答界定符:应答界定符必须是一个隐性位。

(7) 帧结束

帧结束由 7 个隐性位组成,每个数据帧和远程帧都由帧结束界定。

2. 远程帧格式

远程帧的功能:作为某数据接收节点,可以通过发送远程帧启动其资源节点发送数据。

远程帧也有标准格式和扩展格式。远程帧包括:帧起始、仲裁域、控制域、CRC 域、应答域和帧结束,如图 1.3.7 所示。

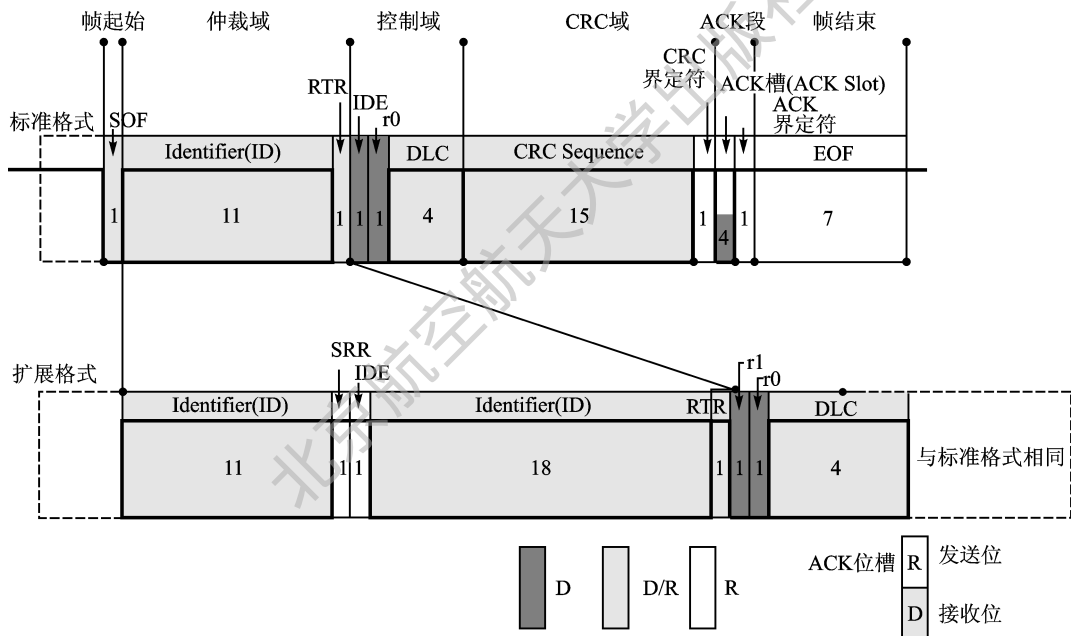


图 1.3.7 远程帧结构

远程帧中的帧起始、仲裁域的标识符、控制域、CRC 域、应答域和帧结束与数据帧相同,因此这里就不重复说明了。

远程帧中的 RTR 位必须为隐性位。远程帧没有数据域,与数据长度码的值没有关系,该数值表示所请求数据帧的数据长度。

3. 错误帧

错误帧由两个域组成:第 1 个域是由来自不同节点的出错标志叠加而成,第 2 个域是出错界定符,如图 1.3.8 所示。

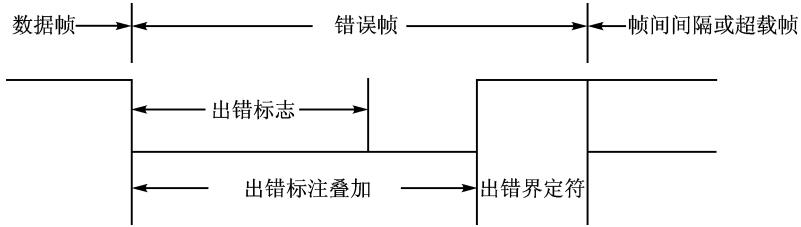


图 1.3.8 错误帧结构

出错标志有两种形式:主动错误标志和被动错误标志。主动错误标志由 6 个连续的显性位组成;而被动出错标志由 6 个连续的隐性位组成,除非被其他节点的显性位重写。

检测到错误条件的“主动错误状态”的节点通过发送主动错误标志,以指示错误。错误标志的形式破坏了从帧起始到 CRC 界定符的位填充规则,或者破坏了应答域或帧结束的固定形式。所有其他的节点由此检测到错误条件并与此同时开始发送错误标志。因此,显性位的序列导致一个结果,这个结果就是把各个单独站发送的不同错误标志叠加在一起。这个顺序的总长度最小为 6 位,最大为 12 位。

检测到错误条件的“被动错误状态”的节点试图通过发送被动错误标志,以指示错误。“被动错误状态”的节点等待 6 个相同极性的连续位(这 6 位处于被动错误标志的开始)。当这 6 个相同的位被检测到时,被动错误标志的发送就完成了。

错误界定符由 8 个隐性位组成。错误标志发送完毕以后,每一节点就发送隐性位并一直监视总线直到检测出一个隐性位为止,然后就开始发送剩余的 7 个隐性位。

4. 超载帧

超载帧包含两个域:超载标志和超载界定符,如图 1.3.9 所示。

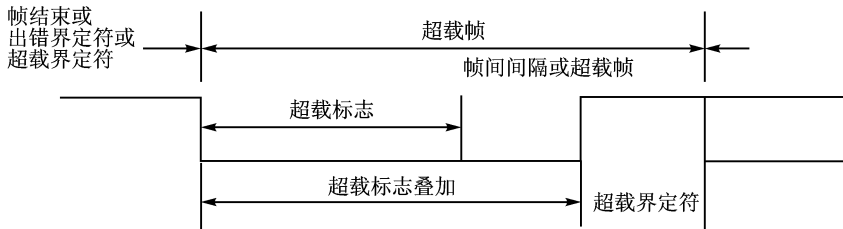


图 1.3.9 超载帧格式

存在以下 3 种超载条件,它们都会引起超载帧的发送:

- ① 接收器要求延迟下一数据帧或远程帧的到达;
- ② 在帧间间隔间歇域的第 1 和第 2 位检测到显性位;
- ③ 如果一个 CAN 节点在出错界定符或超载界定符的第 8 位(最后 1 位)采样到 1 个显性位,节点会发送 1 个超载(而不是出错帧),错误计数器不会增加。

超载标志由 6 个显性位组成。超载标志的所有形式和主动错误标志的一样。超载标志的形式破坏了间歇域的固定形式,因此,所有其他的节点都检测到一过载条件并与此同时发出超载标志。如果在帧间间隔间歇域的第 3 位期间检测到一个显性位,则该位将解释为帧起始。

超载界定符由 8 个隐性位组成。超载界定符的形式和错误界定符的形式一样。在超载标志被发送后,节点就一直监测总线,直到检测到一个从显性位到隐性位的跳变。此时,总线上的每一个节点都完成了过载标志的发送,并开始同时发送剩余的 7 个隐性位。

5. 帧间间隔

数据帧和远程帧通过帧间间隔与总线上存在的其他帧(数据帧、远程帧、出错帧和超载帧)区分开。所不同的是,超载帧和出错帧之前没有帧间间隔,多个超载帧之间也没有帧间间隔。

帧间间隔包括间歇域和总线空闲域。如果“被动错误状态”的节点已作为前一报文的发送器时,则其帧空间除了间歇、总线空闲外,还包括暂停发送域。

非错误消极节点或者作为前一报文接收器的节点的帧间间隔如图 1.3.10 所示。

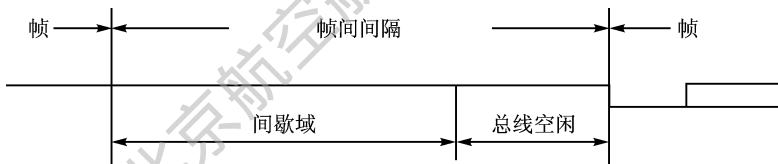


图 1.3.10 非错误消极节点帧间间隔

作为前一报文发送器的错误消极节点的帧间间隔如图 1.3.11 所示。

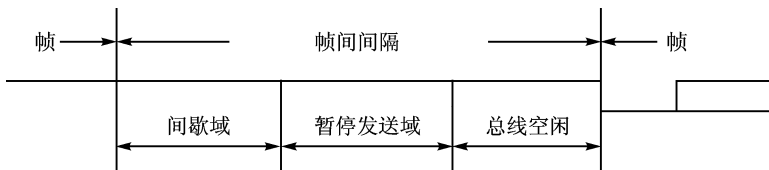


图 1.3.11 错误消极节点帧间间隔

间歇域包括 3 个隐性位。间歇期间,所有的节点均不允许发送数据帧或远程帧,其唯一的作用是标示一个超载条件。

总线空闲的周期可为任意长度。在此期间,总线被认为是空闲的,任何需要发送报文的节点都可以访问总线。一个因其他报文正在发送而被挂起的报文,将在间歇域后的第 1 位开始发送,此时检测到总线上有一个显性位将解释为一个帧起始。

暂停发送域是指“被动错误状态”节点发送一个报文后,在开始发送下一报文或者认可总线处于空闲之前,在间歇域后发出的 8 个隐性位。如果在此期间一次由其他节点引起的发送开始了,该节点将称为报文接收器。暂停发送域降低了“被动错误状态”节点向总线发送报文的频率,减少了“被动错误状态”节点因自身故障干扰总线的可能性。

1.3.3 帧优先级仲裁

在总线空闲时,最先开始发送报文的节点获得发送权。

多个节点同时开始发送时,各发送节点从仲裁域的第 1 位开始进行仲裁,连续输出显性电平最多的节点可继续发送。仲裁过程如图 1.3.12 所示。

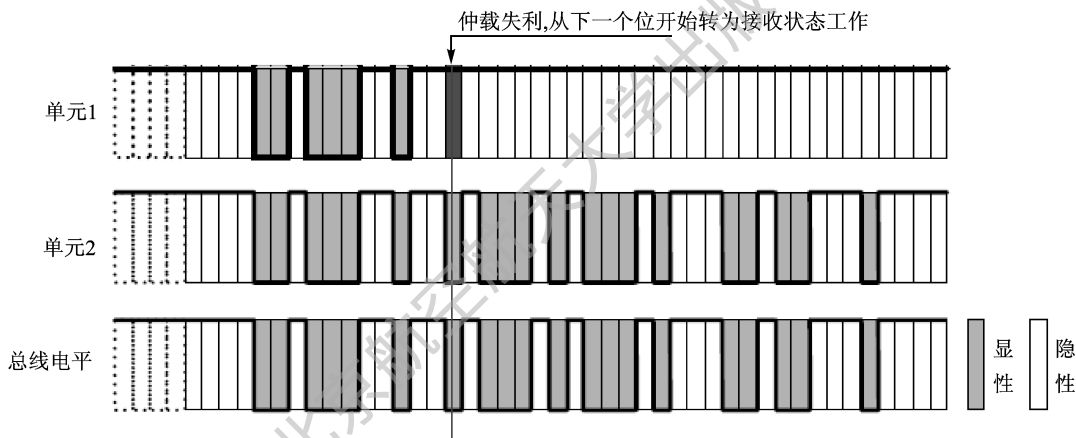


图 1.3.12 仲裁过程图

(1) 数据帧和远程帧的优先级

具有相同标识符 ID 的数据帧和远程帧在总线上竞争时,仲裁域的最后一位(RTR)为显性位的数据帧具有优先权,可继续发送。

数据帧和远程帧的仲裁过程如图 1.3.13 所示。

(2) 标准格式和扩展格式的优先级

标准格式标识符 ID 与具有相同 ID 的远程帧或者扩展格式的数据帧在总线上竞争时,标准格式的 RTR 位为显性位的具有优先权,可继续发送。

标准格式和扩展格式的仲裁过程如图 1.3.14 所示。

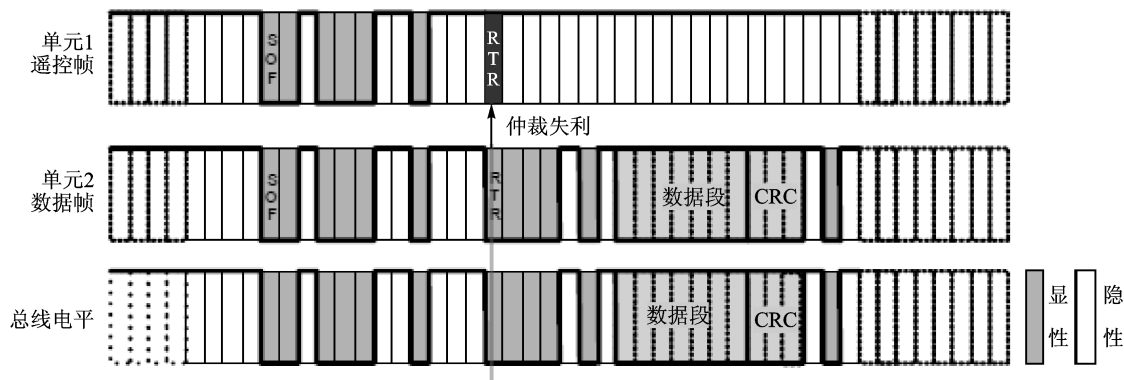


图 1.3.13 数据帧和远程帧的仲裁过程

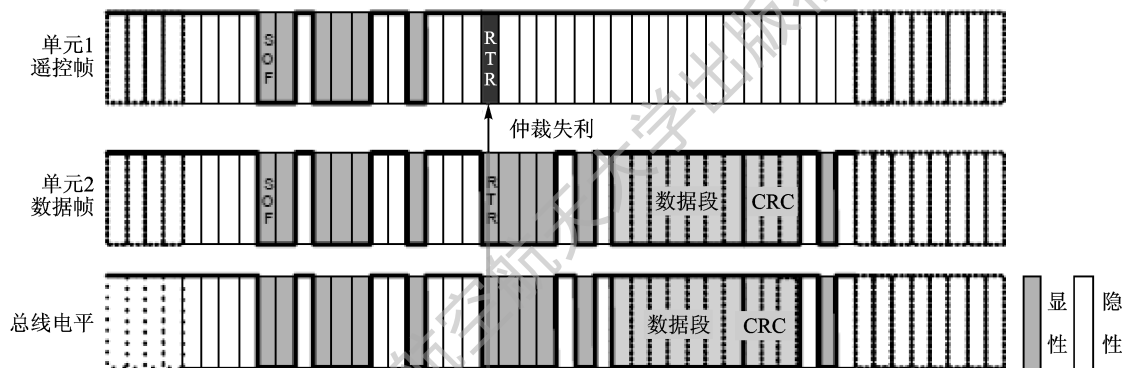


图 1.3.14 标准格式和扩展格式的仲裁过程

1.4 报文滤波与校验

1. 报文滤波

报文滤波取决于整个标识符。允许在报文滤波中将任何的标识符位设置为“不考虑”的可选屏蔽寄存器。可以选择多组的标识符,使之被映射到隶属的接收缓冲器里。

如果使用屏蔽寄存器,它的每一位必须是可编程的,即它们能够被允许或者禁止报文滤波。屏蔽寄存器的长度可以包含整个标识符,也可以包含部分标识符。

2. 报文校验

校验报文是否有效的时间点,发送器与接收器各不相同。

(1) 发送器

如果直到帧的末尾位均没有错误,则此报文对于发送器有效。如果报文破损,则报文会根据优先权自动重发。为了能够和其他信息竞争总线,重新传输必须在总线空闲时启动。

(2) 接收器

如果直到最后,所有的位(除了帧末尾位)均没有错误,则报文对于接收器有效。

1.5 编码——位填充

位流编码:帧的部分,诸如帧起始、仲裁域、控制域、数据域以及CRC序列,均通过位填充的方法编码。无论何时,发送器只要检测到位流里有5个连续识别值的位,便自动在位流里插入一补码位。

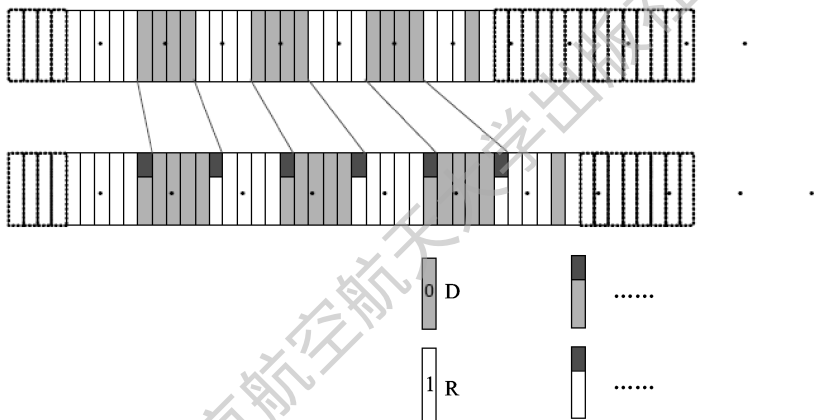


图 1.5.1 位填充示意图

数据帧或远程帧(CRC 界定符、应答场和帧末尾)的剩余位域形式相同,不填充。错误帧和过载帧的形式也相同,但并不通过位填充的方法进行编码。

CAN 总线报文里的位流根据“不归零”(NRZ)的方法来编码。这就是说,在整个位时间里,位电平不是“显性”就是“隐性”。

1.6 错误处理与故障界定

1.6.1 错误类型

CAN 总线通信过程中存在以下 5 种不同的错误类型(相互之间不排斥)。

(1) 位错误

CAN 节点在发送位的同时也对总线进行监视,如果所发送的位值与所监视的位值不相符合,则在此位时间里检测到一个位错误(Bit Error)。位错误是由向总线上发出数据帧、远程帧、错误帧、过载帧的节点和发出 ACK 的节点、发出错误的节点来检测。但以下情况除外:在仲裁域发出隐性电平,但检测出显性电平时,将被视为仲裁失利,而不是位错误;在仲裁域作为填充位发出隐性电平,但检测出显性电平时,将不视为位错误,而是填充错误;发送节点在 ACK 段发出隐性电平,但检测到显性电平时,将被判断为其他单元的 ACK 应答,而非位错误;发出被动错误标志(6 位隐性位)但检测出显性电平时,将遵从错误标志的结束条件,等待检测出连续相同 6 个位的值(显性或隐性),并不视为位错误。

(2) 填充错误

如果在使用位填充法进行编码的报文中,出现了第 6 个连续相同的位电平时,将检测到一个填充错误(Stuff Error)。

(3) CRC 错误

CRC 序列包括发送器的 CRC 计算结果,接收器计算 CRC 的方法与发送器相同。如果计算结果与接收到 CRC 序列的结果不相符,则检测到一个 CRC 错误(CRC Error)。

(4) 形式错误

当一个固定形式的位域含有 1 个或多个非法位,则检测到一个形式错误(Form Error)。但以下情况除外:即使接收单元检测出 EOF(7 位的隐性位)的最后一位(第 8 位)为显性电平,也不视为格式错误;即使接收单元检测出数据长度码(DLC)为 9~15 的数值时,也不视为格式错误。

(5) 应答错误

只要在应答间隙期间所监视的位不为“显性”,则发送器会检测到一个应答错误(Acknowledgment Error)。

对以上错误进行分类归纳如表 1.6.1 所列。

表 1.6.1 错误类型

错误类型	错误内容	错误的检测帧类型	检测节点
位错误	比较输出电平和总线电平(不含填充位),当两电平不一样时所检测到的错误	数据帧(SOF~EOF);远程帧(SOF~EOF); 错误帧;超载帧	发送节点; 接收节点
填充错误	在需要位填充的域内,连续检测到 6 位相同的电平时所检测到的错误	数据帧(SOF~CRC 顺序); 远程帧(SOF~CRC 顺序)	发送节点; 接收节点
CRC 错误	从接收到的数据计算出的 CRC 结果与接收到的 CRC 顺序不同时所检测到的错误	数据帧(CRC 顺序); 远程帧(CRC 顺序)	接收节点

错误类型	错误内容	错误的检测帧类型	检测节点
格式错误	检测出与固定格式的位域相反的格式时所检测到错误	数据帧(CRC 界定符、ACR 界定符、EOF); 远程帧(CRC 界定符、ACR 界定符、EOF); 错误界定符;超载界定符	接收节点
ACK 错误	发送单元在 ACK 间隙期间检测出隐性电平时所检测到的错误(ACK 未被传递过来时所检测到的错误)	数据帧(ACK 间隙); 远程帧(ACK 间隙)	发送节点

1.6.2 节点错误处理

如果一个节点检测到总线上的信号满足错误条件,则会输出错误标志来指示总线错误。

处于“主动错误状态”的节点输出的错误标志为主动错误标志;处于“被动错误状态”的节点输出的错误标志为被动错误标志。发送单元发送完错误帧后,将再次发送数据帧或遥控帧。错误标志输出时序如表 1.6.2 所列。

表 1.6.2 错误标志输出时序

错误的种类	输出时序
位错误,填充错误,格式错误,ACK 错误	从检测出错误后的下一位开始输出错误标志
CRC 错误	ACK 界定符后的下一位开始输出错误标志

1.6.3 故障界定方法

造成 CAN 节点出错的原因可能是总线上暂时的扰动,也可能是节点不可恢复的故障,CAN 规范定义了详细的状态转换规则用于故障界定。一个 CAN 节点发现出错后可能处于以下 3 种状态中的一种:主动错误状态;被动错误状态;总线关闭状态。

主动错误状态节点可以正常地参与总线通信并在检测到错误时发出主动错误标志;错误节点参与总线通信并且在被检测到错误时只发出被动错误标志,而且被动错误状态节点将在下一个报文发送之前处于等待状态;总线关闭节点不允许对总线有任何影响(比如关闭输出驱动器)。

为了实现故障界定,每个 CAN 节点都进行两种计数:发送错误计数和接收错误计数,这些计数按以下规则改变(注意:在给定的报文传输中,可能要用到的规则不只一个):

① 当接收器检测到一个错误,接收错误计数就加 1。在发送主动错误标志或过载标志期间所检测到的错误为位错误时,接收错误计数器值不加 1。

② 当接收器在发送出错误标志结束后的第 1 个位期间检测到显性位时,接收错误计数值加 8。

③ 当发送器发送一个错误标志时,发送错误计数器值加 8。这里有两个情况例外:一是当发送器处于被动错误状态,并且检测到一个应答错误或者发送被动错误状态标志期间没有检测到显性位;二是发送器因为仲裁期间发生的填充错误而发送错误标志。在发生以上两个情况时,发送错误计数值不改变。

④ 如果发送器在发送主动错误标志或超载标志时,检测到一个位错误,则发送错误计数器值加 8。

⑤ 如果接收器在发送主动错误标志或超载标志时,检测到一个位错误,则接收错误计数器值加 8。

⑥ 任何节点在发送主动错误标志、被动错误标志或超载标志以后,最多容许 7 个连续的显性位。当检测到第 14 个连续的显性位后(若是主动错误标志或超载错误标志),或者在检测到跟随着被动错误标志的第 8 个连续的显性位后,以及在每一个由额外的 8 个连续的显性位组成的序列后,每个发送器将发送错误计数值加 8,每个接收器将接收错误计数器值加 8。

⑦ 在报文成功传送后(得到正确的 ACK 以及直到帧末尾结束没有错误),发送错误计数器值减 1,除非发送错误计数器值已经是 0。

⑧ 在成功接收到报文后(直到应答间隙接收没有错误以及成功地发送了 ACK 位),如果接收错误计数值为 1~127,则接收错误计数器值减 1;如果接收错误计数器值是 0,则保持 0;如果大于 127,则它会被设置为一个 119~127 的值。

⑨ 当发送错误计数器值等于或大于 128 时,或当接收错误计数器值等于或大于 128 时,节点处于“错误被动”状态。注意,让节点成为“被动错误状态”节点的最后一个错误条件仍然致使节点发出主动错误标志。

⑩ 当发送错误计数器值大于或等于 256 时,节点处于“总线关闭”状态。

⑪ 当发送错误计数器值和接收错误计数器值都小于或等于 127 时,“被动错误状态”节点再次成为“错误主动”节点。

⑫ 总线关闭节点在检测到 128 次出现 11 个连续隐性位之后,该节点可以变成主动错误状态节点(不再是总线关闭节点),它的发送错误计数值和接收错误计数值被设置为 0。一般这种处理需要外界干预,决定一个总线关闭节点是否重新上线。

需要注意:如果错误计数值大于 96,则表明总线被严重干扰,这也为我们提供了一种对总线的辅助测试手段。如果启动期间总线上只有 1 个节点在线,这个节点发送报文后,它将得不到应答,并且检测到 1 个错误而重发报文。由于这个原因,该节点可能会处于被动错误状态,但不会处于总线关闭状态。

1.7 位定时要求

CAN 规范中有关位定时的论述包括以下一些主要内容。

(1) 标称位速率

标称位速率是一个理想的发送器在没有重新同步的情况下每秒发送的位数量。

(2) 标称位时间

标称位时间与标称位速率有以下关系：

$$\text{标称位时间} = 1 / \text{标称位速率}$$

标称位时间可以划分为 4 个不重叠的时间段，分别是：同步段；传播时间段；相位缓冲段 1；相位缓冲段 2。标称位时间的划分如图 1.7.1 所示。

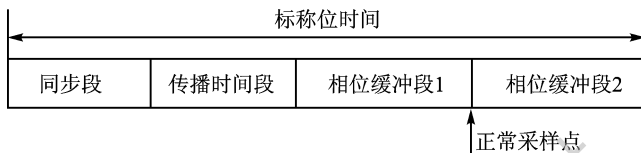


图 1.7.1 标称位时间的划分

同步段：位时间的同步段用于同步总线上不同的节点。这一段内要有一个跳变沿。

传播段：传播段用于补偿网络内的物理延时时间。它是总线上输入比较器延时和输出驱动器延时总和的两倍。

相位缓冲段 1/2：相位缓冲段用于补偿边沿阶段的误差。这两个段可以通过重新同步加长或缩短。

采样点：采样点是读总线电平并解释各位的值得一个时间点。采样点位于相位缓冲段 1 之后。

(3) 信息处理时间

信息处理时间是一个以采样点作为起始点的时间段。采样点用于计算后续位的位电平。

(4) 时间份额

时间份额是与振荡器周期有关的固定时间长度。存在有一个可编程的预比例因子，其整数数值范围为 1~32，以最小时间份额为起点，时间份额的长度为：

$$\text{时间份额} = m \times \text{最小时间份额} (m \text{ 为预比例因子})$$

(5) 时间段的长度

同步段的长度为 1 个时间份额；传播段的长度可编程设置为 1, 2, ..., 8 个时间份额；相位缓冲段 1 的长度可编程设置为 1, 2, ..., 8 个时间份额；相位缓冲段 2 的长度为相位缓冲段 1 和信息处理时间两者中的最大值；信息处理时间的长度小于或等于 2 个时间份额。因此，一个位时间的总长度可以编程设置在 8~25 个时间份额范围内。

目前，在控制单元的设计中，趋向于现场 CPU 和通信器件里使用同一振荡器。因此，CAN 器件的振荡频率往往就是现场 CPU 的振荡频率，而 CPU 的振荡器频率主要取决于控制单元的需求。为了得到所需的比特率，位定时的可编程设置性是有必要的。另一方面，由于这

些器件允许选择外部的振荡器以便于被调整到合适的比特率,所以对于这些部件,可配置性不是必要的。

但是,应该将所有节点的采样点选择在共有的位置。为此,SLIO 器件(串行连接 I/O 器件)必须兼容以下的位时间定义,如图 1.7.2 所示。



图 1.7.2 位时间定义

(6) 硬同步

硬同步后,内部的位时间从同步段重新开始。因此,硬同步强迫引起硬同步的跳变沿处于重新开始的位时间同步段之内。

(7) 重新同步跳转宽度

重新同步的结果使相位缓冲段 1 增长,或使相位缓冲段 2 缩短。相位缓冲段加长或缩短的数量有一个上限,此上限由重新同步跳转宽度给定。重新同步跳转宽度应编程设置为 1 到最小值之间(此最小值为 4,相位缓冲段 1)。

时钟信息可以从一个位值到另一个位值的跳变得来。根据位填充的规则,总线上具有相同值的连续位的数量是确定的,这个属性使在帧发送/接收期间总线节点重新同步于位流成为可能。可用于重新同步的两个跳变之间的最大长度为 29 个位时间。

(8) 沿相位误差

一个沿的相位误差由相关于同步段的沿的位置给出,以时间额度量。相位误差定义如下:

- ① 如果跳变沿处于同步段内,则 $e=0$;
- ② 如果跳变沿位于采样点之前,同步段之后,则 $e>0$;
- ③ 如果跳变沿处于前一个位的采样点之后,当前位的同步段之前,则 $e<0$ 。

(9) 重新同步

当引起重新同步沿的相位误差的幅值小于或等于重新同步跳转宽度的编程设定值时,重新同步和硬同步的作用相同,不进行同步调整。当相位错误的幅度大于重新同步跳转宽度时:

- ① 如果相位误差为正,则相位缓冲段 1 被增长。增长的范围为与重新同步跳转宽度相等的值。
- ② 如果相位误差为负,则相位缓冲段 2 被缩短。缩短的范围为与重新同步跳转宽度相等的值。

(10) 同步的原则

硬同步和重新同步是同步的两种形式,共同遵循以下规则:

- ① 在一个位时间里只允许一种同步。
- ② 只有在采样点之前探测到的值与紧跟沿之后的总线值不相符合时,才把沿用作于同步。
- ③ 在总线空闲期间,只要有从隐性到显性的跳变沿,硬同步就会被执行。
- ④ 符合规则 1 和规则 2 的所有从隐性到显性的跳变沿将用作重新同步。有一个例外情况,当发送一个显性位的节点不执行重新同步而导致一个隐性到显性的跳变沿时,此沿具有正的相位误差,不能用作重新同步。

本章小结

本章主要讲解了 CAN 总线协议,需要重点掌握的内容包括:

- ① CAN 总线报文传输中的帧类型、帧格式以及帧优先级仲裁规则。
- ② CAN 总线错误处理和故障界定方法。
- ③ CAN 总线网络的拓扑结构和物理层线缆的选择。

北京航空航天大学出版社

第 2 章

CAN 总线应用层协议

2.1 为什么构建 CAN 应用层协议

通过第 1 章的讲解,了解到 CAN 总线协议规范只定义了数据链路层和物理层,并不完整。在实际应用中,必然要根据实际系统的需求制定合适的应用层协议,哪怕是非常简单的应用层协议。用户可以建立基于 CAN2.0 规范的数据通信,但这种数据通信的传输内容一般不能灵活修改,只适合于固定的通信方式。拿大家熟悉的 RS232 通信来举例,协议规定了传输每个字节时的开始位、停止位以及奇偶校验位,在使用其通信时,一般会定义报文的报文头和报文尾,以表示报文的开始和结束。这个定义就是最简单的应用层协议,只是每人的定义可能不同,不同应用协议互联通信则会出现问题。因此,构建 CAN 应用层协议的首要目的就是要使设计的系统能够稳定可靠的相互通信。如果制定的应用协议稳定可靠,并且被大多数人采纳,那就有可能形成一种规范。

CAN 总线应用层协议应该具备以下功能:

- ① CAN 总线的网络控制域管理;
- ② 确认数据发送成功;
- ③ 能够发送大于 8 字节的数据块;
- ④ 为不同的 CAN 物理节点分配不同的报文标识符;
- ⑤ 定义报文的内容和含义;
- ⑥ CAN 网络节点的监控、故障诊断和标识。

因此,为保证 CAN 总线网络的可靠运行,必须制定严谨的 CAN 总线应用层协议。

2.2 常用的 CAN 总线应用层协议

CAN 应用层协议是在 CAN 规范的基础上发展起来的。许多系统可以制定一个合适的应用层,但对于许多行业来说,这种方法是不经济的,一些组织已经研究并开放了应用层标准,以使系统的综合应用变得十分容易。表 2.2.1 列举了一些常用的 CAN 总线应用层协议。

表 2.2.1 常用的 CAN 总线应用层协议

协议名称	制定组织	协议名称	制定组织
DeviceNet 协议	ODVA	CANKingdom 协议	Kvaser
CAL 协议	CiA	J1939 协议	SAE
CANopen 协议	CiA	SDS 协议	Honeywell

DeviceNet 协议和 CANopen 协议是真正占领市场的高层协议。标准的应用层协议具有以下特点：

- ① 一个开发的、标准化的 CAN 总线应用层协议；
- ② 制定属于特定行业的 CAN 总线应用方案；
- ③ 支持不同厂商设备的互用性、可交换性；
- ④ 完善地定义标准设备模型；
- ⑤ 开放扩充自定义设备的接口；
- ⑥ 对总线仲裁、数据交换、错误处理有明确的限定。

下面主要介绍 DeviceNet 协议和 CANopen 协议的特点。

(1) DeviceNet 协议

DeviceNet 协议是为工厂自动控制而特别定制的，目前已经成为美国自动化领域的领导者，在其他领域也逐步得到了推广和应用。DeviceNetTM 是一个非常成熟的开放式网络，它根据抽象模型来定义，这个模型是可用的通信服务和一个 DeviceNet 节点的外部可见行为。相应的设备子协议规定同类设备的行为。DeviceNet 允许多个复杂设备互联，也允许简单设备互换。

DeviceNet 协议的主要技术特点如下：

- ① 网络大小：最多支持 64 个节点；
- ② 网络模型：生产者/消费者模型；
- ③ 波特率：125 kbps(500 m)，250 kbps(150 m)，500 kbps(100 m)；
- ④ 数据包：0~8 字节；
- ⑤ 总线拓扑结构：线性干线/支线，电源和信号在同一网络电缆中；
- ⑥ 总线寻址：带多点传送（一对多）的点对点，多主站和主/从轮询或状态改变（基于事件）；
- ⑦ 系统特性：支持设备的热插拔，无需网络断电。

(2) CANopen 协议

CANopen 协议是一个基于 CAL 的子协议，用于产品部件的内部网络空盒子。CANopen 不仅定义了应用层和通信子协议，也为可编程系统、不同器件、接口、应用子协议定义了页/帧状态，这就是工业领域使用 CANopen 协议的一个重要原因。

CANopen 协议的主要技术特点如下：

- ① 网络结构简单；
- ② 设备参数存取简单；
- ③ 网络设备可同步操作；
- ④ 实时数据交换可循环和事件触发等多种传输方式；
- ⑤ 网络容量高达 127 个节点；
- ⑥ 高传输速率(最高可达 1 Mbps)；
- ⑦ 极强的抗干扰能力和总线冲突仲裁方式；
- ⑧ 统一的机构管理(CIA)管理维护；
- ⑨ 开放性结构,对不同厂家设备具有好的兼容性。

2.3 iCAN 总线协议概要

iCAN 协议全称“Industry CAN - bus Application Protocol”，即工业 CAN - bus 应用层协议。iCAN 协议沿用了 CAN 协议标准所规定的总线网络的物理层和数据链路层,对 CAN - bus 报文中的帧标识符和数据域进行了重定义。iCAN 协议位于 OSI 模型的第 7 层,如图 2.3.1 所示。

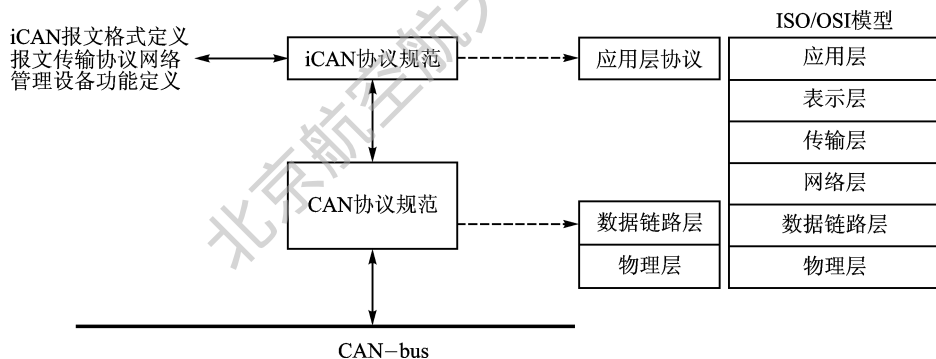


图 2.3.1 iCAN 协议模型

iCAN 协议详细地定义了 CAN 总线报文中帧标识符以及数据的分配和应用,建立了一个统一的设备模型,定义了设备的 I/O 资源和访问规则,通信模型如图 2.3.2 所示。

iCAN 协议规范主要内容如下：

- ① iCAN 报文格式定义:规定了 iCAN 协议规范中使用的 CAN 帧类型、帧 ID、报文数据的使用等；
- ② 报文传输协议:规定了基于 iCAN 协议的设备之间的通信方式；

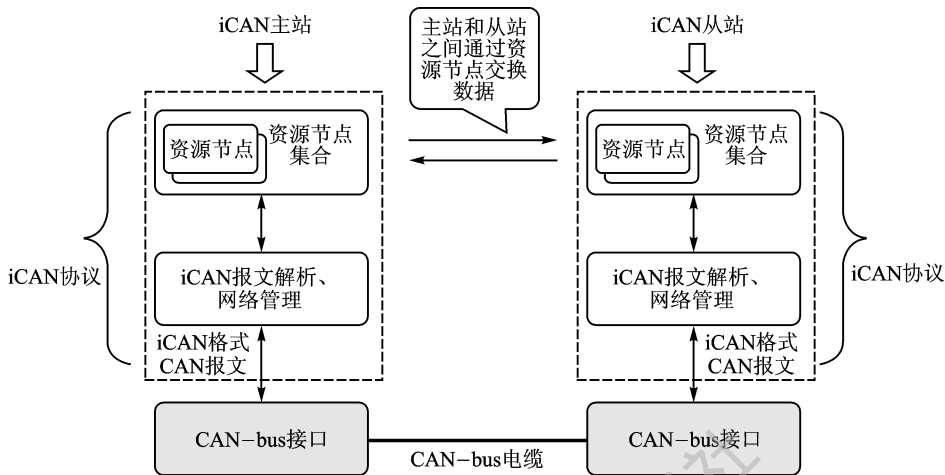


图 2.3.2 iCAN 协议通信模型

③ 设备功能定义:设备标识、设备应用单元、设备通信、应用参数以及定义标准设备类型,区分网络上设备具有的不同功能或者产品类型;

④ 网络管理:规定了设备通信监控和错误管理。

iCAN 网络具有以下特点:

- ① 基于节点地址的设备寻址,iCAN 网络中支持多达 64 个节点;
- ② 支持轮询和事件触发通信方式,保证了数据通信的实时性;
- ③ 基于资源节点的设备数据寻址方式,简化了主站和从站间数据交换方式;
- ④ 统一的设备模型,提高了设备的易用性和互换性;
- ⑤ 完善的网络管理,有效监控所有通信节点,保证了数据通信的可靠性。

2.4 iCAN 协议的报文格式

iCAN 协议报文格式定义了 CAN 报文的标识符以及数据部分使用原则和功能含义。对报文格式的定义,使网络中 CAN 报文的每个组成元素都有其特定功能和意义。CAN 报文格式的定义也是 iCAN 协议中最基本的组成部分,CAN 报文格式的定义包括两个方面:报文标识符和报文数据部分的分配。

iCAN 协议中只使用了扩展帧格式 CAN 报文,没有使用标准帧格式 CAN 报文,也没有使用远程帧格式的报文。

iCAN 协议中,对 CAN 报文的 29 位标识符和报文数据部分的使用都做了详细的规定,iCAN 协议中报文的格式规定如图 2.4.1 所示。

帧标识符	帧 ID 位	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21							
	功能	00			SrcMACID (源节点编码)											
	帧 ID 位	ID20	ID19	ID18	ID17	ID16	ID15	ID14	ID13							
	功能	00			DestMACID (目标节点编码)											
	帧 ID 位	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0		
	功能	ACK	Func ID (功能编码)				Source ID (资源节点编码)									
帧数据部分	RTR = 0															
	DLC															
	Byte 0 (SegFlag)															
	Byte1 (LengthFlag、ErrID)															
	Byte2															
	Byte3															
	Byte4															
	Byte5															
Byte6																
Byte7																

图 2.4.1 报文帧格式

2.4.1 iCAN 报文标识符分配

报文标识符规定了数据通信中的源节点 MAC ID 和目标节点 MAC ID,并指示了报文的功
能以及所要访问的资源。报文标识符分为 5 个部分:SrcMACID(源节点地址编码)、Dest-
MACID(目标节点地址编码)、ACK 位、FUNCID(功能编码)、SourceID(资源节点编码)。

iCAN 报文标识符分配如表 2.4.1 所列。

表 2.4.1 iCAN 报文标识符分配

帧 ID 位编码	ID28	ID27	ID26	ID24	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16	ID15	ID14	ID13
功 能	00			SrcMACID(源节点编码)				00		DestMACID(目标节点编码)					
帧 ID 位编码	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR	
功 能	ACK	Func ID(功能编码)				Source ID(资源节点编码)						0			

(1) MAC ID

MAC ID(节点地址编码)是设备在网络上的唯一标识,由 6 位组成,其范围为 0x00~
0x3F;0xFF 有特殊用途。因此,可以知道一个 iCAN 网络中最多容纳 64 个节点。

在通信报文的标识符中指定了源节点(发送节点)和目标节点(接收节点)的地址。在每次
通信过程中,通信双方都必须检查 SrcMACID(源节点地址编码)和 DestMACID(目标节点地
址编码)的值与已知连接的两端点是否相同。

SrcMACID 是指发送报文的节点地址编码,占用标识符中的位 ID28~ID21,如表 2.4.1
所列。

源节点的 MAC ID 由 6 位组成,数值范围为 0x00~0x3F。

DestMACID 是指接收节点的地址编码,占用标识符中的位 ID20~ID13,如表 2.4.1 所列。

目标节点的 MAC ID 由 6 位组成,数值范围为 0x00~0x3F。若 DestMACID 为 0xFF 时,表示本次发送的是广播帧(广播帧不需要应答)。

(2) ACK

ACK(响应标识位):用于区分帧类型为命令帧还是响应帧,并说明是否需要应答本帧。其由 1 位组成,占用标识符位 ID12,如表 2.4.2 所列。

表 2.4.2 ACK 响应标识位功能

ACK	功能说明
0	用于命令帧,本帧需要应答;但对于广播帧,此值无意义
1	用于响应帧,本帧不需要应答;或者不需要应答的命令帧(如广播帧)

(3) FUNC ID

FUNC ID(功能编码):用于指示报文所要实现的功能,接收报文节点根据报文中的功能码进行相应的处理。其由 4 位组成,占用标识符位 ID11~ID8,如表 2.4.3 所列。

表 2.4.3 功能编码说明

功能编码 ID11~ID8	功能名称	详细描述
0x00	保留	—
0x01	连续写端口	用于对单个或者多个资源节点的数据写入
0x02	连续读端口	用于读取单个或者多个资源节点的数据
0x03	输入端口事件触发传送	用于输入端口定时循环或者状态改变传送
0x04	建立连接	用于与 iCAN 节点建立通信
0x05	删除连接	用于删除与 iCAN 节点建立的通信
0x06	设备复位	用于复位 iCAN 节点
0x07	MAC ID 检测	用于检测网络上是否有相同的 MAC ID 节点
0x08~0x0E	保留	—
0x0F	出错响应	用于指示为出错响应

(4) SourceID

SourceID(资源节点编码):用于指示所要操作的设备内容单元,由 8 位组成,占用标识符位 ID7~ID0,如表 2.4.1 所列。根据报文中资源节点地址标号,接收报文的节点对设备内部相应的单元进行操作。在 iCAN 协议中基本资源节点占用 256 字节空间。

2.4.2 iCAN 报文数据部分定义

在 iCAN 协议中报文的数据部分主要用于传送与功能编码相关的参数。帧的数据最多可以包含 8 字节数据,不同位置的字节具有不同的功能,可以分为 2 部分,如表 2.4.4 所列。

表 2.4.4 iCAN 帧数据部分分配

iCAN 帧数据字节	功能说明
字节 0(SegFlag)	分段编码
字节 1(LengthFlag,ErrID)	与功能编码相关的参数
字节 2~7	

在帧的数据部分中,第 1 个字节被固定为分段码参数,因此 iCAN 报文帧一次最大传输数据的长度为 7 字节。

(1) 字节 0

帧数据部分字节 0 的功能定义为分段编码(SegFlag)。分段编码 SegFlag 用于实现大于 7 字节长度的数据传输。其中 SegPolo(分段标志)由 2 位组成;SegNum(分段编号)由 6 位组成。SegFlag 的格式定义如表 2.4.5 所列。

表 2.4.5 分段编码 SegFlag

位	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
描述	SegPolo(分段标志)			SegNum(分段编号)				

SegPolo(分段标志)的位值表如表 2.4.6 所列。

表 2.4.6 分段标志功能说明

SegPolo 位值	功能
0b00	本次数据传输没有分段
0b01	批量数据传输的第 1 个分段,此时,SegNum=0x00
0b10	中间分段。SegNum 值从 0x01 开始,以区分分段数
0b11	批量传输的最后分段

因此,当传送数据长度不超过 7 字节时,帧数据部分中的 SegFlag=0x00;当报文分帧传送时,目标节点(接收节点)只在接收完成整个批量传输时才做出响应。采用分段传输时,分段传送到最大报文长度为 448(64×7)字节。

(2) 字节 1~7

报文数据中的字节 1~7 通常作为功能码的参数,不同的功能码对应的参数意义不同,如表 2.4.7 所列。

表 2.4.7 iCAN 报文数据部分含义

功能码	iCAN 报文数据部分含义			
	字节 0	字节 1	字节 2	字节 3~7
0x01(连续写端口,访问资源子节点)	SegFlag	SubSourceID	—	
0x02(连续读端口,访问资源节点)	SegFlag	LengthFlag	—	
0x02(连续读端口,访问资源子节点)	SegFlag	SubSourceID	LengthFlag	—
0x0F(异常响应)	SegFlag	ErrID	—	

在某些特定的帧中,数据字节 1 可以具有以下功能:

- ① LengthFlag(长度标志),只在“连续读端口”命令中出现,由 1 字节组成,位于数据区字节 1 的位置。LengthFlag 表示需要读出的字节数。
- ② SubSourceID(资源子节点),只在 0xF9~0xFF 范围内对资源节点的资源子节点访问时使用该参数。在“连续读端口”命令中,SubSourceID 位于 iCAN 报文数据部分的字节 1;在“连续写端口”命令中,SubSourceID 位于 iCAN 报文数据部分的字节 1,对于采用分段传输的“连续写端口”命令,SubSourceID 只出现在第 1 段命令报文中。
- ③ ErrID(错误响应编码),在错误响应报文中使用,用于说明错误响应的类型,如表 2.4.8 所列。

表 2.4.8 错误响应编码说明

ErrID	功能说明
0x01	功能码不存在。使用该错误响应编码的情况:命令帧中使用了未定义的功能码;iCAN 协议中已经使用的功能码(见表 2.4.6)
0x02	资源不存在。使用该错误响应编码的情况:命令帧中所要访问的资源节点在目标设备中不存在
0x03	命令不支持。使用该错误响应编码的情况:目标设备不支持该功能码;访问的资源不支持相应的功能码操作(如向只写属性的资源进行读操作)
0x04	功能码参数非法。使用该错误响应编码的情况:命令帧中功能码附加的参数为非法参数(如参数的长度不对)
0x05	连接不存在。如果主站和从站之间的连接尚未建立,从站设备在响应错误帧中使用该错误编码
0x06	操作无效。对串口资源进行写操作时,如果从站的串口缓冲区满,使用此错误编码;对串口资源进行读操作时,如果从站的串口缓冲区空,使用此错误编码
0x07	分段传输过程出错

2.4.3 iCAN 报文格式详细说明

(1) 关于 ACK 位的帧格式

在 iCAN 网络的源节点负责发起通信,在网络上发送“命令帧”的一般为主站点。“命令帧”格式如下:

SrcMACID	DestID	ACK=0	FuncID	SourceID	DLC	分段编码	0~7 个数据
----------	--------	-------	--------	----------	-----	------	---------

“广播帧”格式如下:

SrcMACID	0xFF	ACK=1	FuncID	SourceID	DLC	分段编码	0~7 个数据
----------	------	-------	--------	----------	-----	------	---------

“正常响应帧”格式如下:

SrcMACID	DestID	ACK=1	FuncID	SourceID	DLC	分段编码	0~7 个数据
----------	--------	-------	--------	----------	-----	------	---------

备注: 正常响应帧的功能编码(FuncID)与命令帧的功能编码(FuncID)相同。

“出错响应帧”格式如下:

SrcMACID	DestID	ACK=1	FuncID=0x0F	SourceID	DLC	0x00	ErrID
----------	--------	-------	-------------	----------	-----	------	-------

备注: 出错响应帧的功能编码(FuncID)为 0xFF。

(2) iCAN 分段帧格式

“命令帧”格式如下:

SrcMACID	DestID	ACK=0	FuncID	SourceID	DLC	分段编码 SegFlag	0~7 个数据
----------	--------	-------	--------	----------	-----	-----------------	---------

“响应帧”格式如下:

SrcMACID	DestID	ACK=1	FuncID	SourceID	DLC	分段编码 SegFlag	0~7 个数据
----------	--------	-------	--------	----------	-----	-----------------	---------

备注: 正常响应帧的功能编码(FuncID)与命令帧的功能编码(FuncID)相同。

① 源节点(SrcMACID)向目标节点(DestID)的内部资源(SourceID)写入数据 0x11。

帧类型	CAN 帧 ID					DLC	CAN 帧数据部分	
	SrcMACID	DestID	ACK	FuncID	SourceID		SegFlag	1~7 个数据
命令帧	SrcMACID	DestID	0	0x01	SourceID	2	0x00	0x11
正常响应帧	SrcMACID	DestID	1	0x01	SourceID	1	0x00	—
出错响应帧	SrcMACID	DestID	1	0x0F	SourceID	2	0x00	ErrID

② 源节点(SrcMACID)向目标节点(DestID)的内部资源(SourceID)连续写入 20 个数据。

帧类型	CAN 帧 ID					DLC	CAN 帧数据部分	
	SrcMACID	DestID	ACK	FuncID	SourceID		SegFlag	1~7 个数据
命令帧(1)	SrcMACID	DestID	0	0x01	SourceID	8	0x40	数据 1~7
命令帧(2)	SrcMACID	DestID	0	0x01	SourceID	8	0x81	数据 8~14
命令帧(3)	SrcMACID	DestID	0	0x01	SourceID	7	0xc0	数据 15~20
正常响应帧	SrcMACID	DestID	1	0x01	SourceID	1	0x00	—
出错响应帧	SrcMACID	DestID	1	0x0F	SourceID	2	0x00	ErrID

(3)“连续写端口”帧格式

“命令帧”格式如下：

SrcMACID	DestID	ACK=0	FuncID=0x01	SourceID	DLC		分段编码	0~7 个数据
----------	--------	-------	-------------	----------	-----	--	------	---------

“正常响应帧”格式如下：

SrcMACID	DestID	ACK=1	FuncID=0x01	SourceID	DLC		分段编码	0~7 个数据
----------	--------	-------	-------------	----------	-----	--	------	---------

备注：正常响应帧的功能编码(FuncID)与命令帧的功能编码(FuncID)相同。

“出错响应帧”格式如下：

SrcMACID	DestID	ACK=1	FuncID=0x0F	SourceID	DLC		0x00	ErrID
----------	--------	-------	-------------	----------	-----	--	------	-------

备注：出错响应帧的功能编码(FuncID)为 0xFF。

(4)“连续读端口”帧格式

“命令帧”格式如下：

SrcMACID	DestID	ACK=0	FuncID=0x02	SourceID	DLC		分段编码	0~7 个数据
----------	--------	-------	-------------	----------	-----	--	------	---------

“正常响应帧”格式如下：

SrcMACID	DestID	ACK=1	FuncID=0x02	SourceID	DLC		分段编码	0~7 个数据
----------	--------	-------	-------------	----------	-----	--	------	---------

备注：正常响应帧的功能编码(FuncID)与命令帧的功能编码(FuncID)相同。

“出错响应帧”格式如下：

SrcMACID	DestID	ACK=1	FuncID=0x0F	SourceID	DLC		0x00	ErrID
----------	--------	-------	-------------	----------	-----	--	------	-------

备注：出错响应帧的功能编码(FuncID)为 0xFF。

① 源节点 (SrcMACID) 读取目标节点 (DestID) 内部资源 (SourceID) 中的数据。

帧类型	CAN 帧 ID					DLC	CAN 帧数据部分	
	SrcMACID	DestID	ACK	FuncID	SourceID		SegFlag	1~7 个数据
命令帧	SrcMACID	DestID	0	0x02	SourceID	2	0x00	LengthFlag=0x01
正常响应帧	SrcMACID	DestID	1	0x02	SourceID	2	0x00	1 个数据
出错响应帧	SrcMACID	DestID	1	0x0F	SourceID	2	0x00	ErrID

② 源节点 (SrcMACID) 读取目标节点 (DestID) 中自内部资源 (SourceID) 起到连续 20 个数据。

帧类型	CAN 帧 ID					DLC	CAN 帧数据部分	
	SrcMACID	DestID	ACK	FuncID	SourceID		SegFlag	1~7 个数据
命令帧	SrcMACID	DestID	0	0x02	SourceID	2	0x00	LengthFlag=0x14
响应帧(1)	SrcMACID	DestID	1	0x02	SourceID	8	0x40	数据 1~7
响应帧(2)	SrcMACID	DestID	1	0x02	SourceID	8	0x81	数据 8~14
响应帧(3)	SrcMACID	DestID	1	0x02	SourceID	7	0xc0	数据 15~20
出错响应帧	SrcMACID	DestID	1	0x0F	SourceID	2	0x00	ErrID

(5) “输入端口事件触发传送”命令格式

“输入端口循环命令”仅对输入端口有效。当正确配置了设备的循环参数后,设备可以定时将输入数据上传到主站。

“命令帧”格式如下:

SrcMACID	DestID	ACK=0	FuncID=0x03	SourceID	DLC	分段编码	0~7 个数据
----------	--------	-------	-------------	----------	-----	------	---------

“循环帧”格式如下:

SrcMACID	DestID	ACK=1	FuncID=0x03	SourceID	DLC	分段编码	0~7 个数据
----------	--------	-------	-------------	----------	-----	------	---------

(6) “建立连接”命令格式

“建立连接”命令帧用于建立主站设备与从站设备之间的通信。

“命令帧”格式如下:

CAN 帧 ID					DLC=3	CAN 数据部分		
SrcMACID	DestID	ACK=0	FuncID=0x04	SourceID=0xF7		SegFlg=0x00	SrcMACID	CyclicMasterd 定时循环参数