

学校代号: 10532

学 号: S08093022

密 级:

湖南大学硕士学位论文

基于 CANopen 总线的智能接触器 的设计与开发

学位申请人姓名: 田莉

导师姓名及职称: 戴瑜兴教授

培 养 单 位: 电气与信息工程学院

专 业 名 称: 电子科学与技术

论文提交日期: 2011 年 04 月 21 日

论文答辩日期: 2011 年 05 月 06 日

答辩委员会主席: 刘宏立教授



Y1906518

The Design and Development of Intelligent Contactor
Based on CANopen Field-Bus

by
TIAN Li

B.E.(Hunan University)2008

A thesis submitted in partial satisfaction of the

Requirements for the degree of

Master of Science

in

Electronic Science and Technology

in the

Graduate School

of

Hunan University

Supervisor

Professor DAI Yuxing

April, 2011

湖南大学

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

作者签名：田莉

日期：2011 年 5 月 26 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权湖南大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于

1、保密口，在_____年解密后适用本授权书。

2、不保密☒。

(请在以上相应方框内打“√”)

作者签名：田莉

日期：2011 年 5 月 26 日

导师签名：

日期：2011 年 5 月 28 日

摘 要

交流接触器是低压电气系统中应用最广泛的电器之一,其性能好坏不仅有直接的经济利益,还影响到电力系统电能质量与电力资源。交流接触器的节能、低耗、长寿命及保护功能是目前研究中的重点,本文在回顾和总结了交流接触器的发展历程后,致力于交流接触器的新型节能方案的实现和智能控制方案的设计,开发了一种基于 CANopen 总线的智能接触器。

(1)通过分析交流接触器的研究现状和发展趋势,看到国产接触器与国外同类产品存在的差距,针对我国传统交流接触器存在的易烧毁、易误动作、能耗大、保护功能不够等不足,以及电力电子技术和现场总线技术所带来的优势和改进,提出了智能接触器的 PWM 节能技术和 CANopen 总线技术的实现方法。

(2)根据功能要求和具体指标,提出了智能接触器的总体方案。设计了以 MC9S08DZ60 单片机为核心的硬件系统,给出了整个电路系统的原理图,包括主控电路、节能模块、信号采样与调理模块、CANopen 通信模块、人机接口模块、电源模块等,并详细阐述了各模块电路的组成原理与具体电气电路。

(3)在嵌入式平台 $\mu\text{C}/\text{OS-II}$ 操作系统的基础上,结合硬件电路,开发了智能接触器的软件系统。按照功能划分,各个任务首先独立编写调试,然后分配优先级及实现任务间的相互通信,最后联调实现所有的功能。嵌入式软件的设计主要包括 $\mu\text{C}/\text{OS-II}$ 在 DZ60 上的移植、各个用户任务的设计(PWM 节能任务、A/D 采样及数据处理任务、CANopen 实时通信任务),其中详细阐述了 CANopen 协议栈的实现及各个通信对象的设计。

(4)采用 Freescale 公司的开发软件 CodeWarrior 开发系统程序,并搭建了节能模块的测试平台和 CANopen 从站通信的测试平台。对节能模块的相关指标进行测试及分析,并通过模拟主站发送各项命令来进行远程控制,利用 ZLGCANTest 观察总线上传输的报文,结果表明这些报文的帧 ID、数据字节的定义以及整个网络的运行过程都符合 CANopen 规范。

最终的测试表明,本文所设计的智能接触器能够稳定的运行并实现节能、低耗、长寿命等功能,且能通过主站实时监测并保护接触器,达到了预期的目标。

关键词: CANopen 协议; PWM 节能技术; 交流接触器; 通信对象; 直流吸持

Abstract

AC contactor is one of the most widely used appliances in low voltage electrical system, its performance not only has a direct economic benefit, but also affects quality and resources of power systems. The emphasis of the present research is energy saving, low consumption and protection function of AC contactor. After reviewing and summarizing the development of AC contactor, this thesis devoted to the implementation of new energy-saving solutions about the AC contactor and the design of intelligent control scheme, developed a kind of intelligent contactor based on CANopen fieldbus.

Firstly, through analysing the AC contactor's research status and development trend, the thesis found out the existing gap between domestic contactor and overseas similar product, according to the traditional contactor's deficiencies such as easy burned, easy misoperation, large energy consumption, poor protection function and so on; and the advantages and improvements from the power electronic technology and fieldbus technology, then proposed a PWM energy conservation technology of energy-saving contactor and a implementation method of CANopen fieldbus technology.

Secondly, according to the function requirement and specific indicators, this thesis proposed overall scheme of the intelligent contactor. Meanwhile this thesis designed the hardware system with a core of MC9S08DZ60, provided the whole circuit principle diagram including the main control circuit, energy saving module, signal sampling and modulation module, CANopen communication module, human interface module, power module, elaborated the composition principle of each module circuit and specific electrical circuit.

Thirdly, on the basis of $\mu C/OS - II$ operating system in embedded platform, combined with the hardware circuit, the thesis developed software system of intelligent contactor. According to the functional division, firstly, it made the credit-all debugging for each task, then distributed priority and accomplished the mutual communication between tasks, finally attained all functions by joint debugging. Embedded software design included porting $\mu C/OS-II$ in DZ60, the design of each user tasks (PWM energy-saving task, A/D sampling and data processing task, CANopen real-time communications task), in this part it expounded the realization of CANopen protocol stack and the design of each communication object.

Fourthly, the thesis adopted CodeWarrior of freescale company for developing system program, and built the testing platforms of energy-saving module and CANopen slave station's communication. Made the test and analysis of energy-saving module's related indicators, realized remote control by sending each command from simulation host, used ZLGCANTest to observe the message on the bus transmission, the results showed that frame ID of these messages, definition of data bytes and operation process of the whole network all meet CANopen norms.

The final testing shows that the intelligent contactor which this thesis designed can fulfill stable operation and realize the functions such as energy saving, low consumption, long life and so on, moreover, can conduct real-time monitoring and protect the contactor. This design has achieved the desired goal.

Key words: CANopen Protocol; PWM Energy Saving Technology; AC Contactor; Communication Object; DC Holding

目 录

学位论文原创性声明和学位论文授权使用授权书	I
摘 要	II
Abstract	III
第 1 章 绪 论	1
1.1 选题背景	1
1.2 课题的国内外研究现状及发展趋势	1
1.3 课题来源及意义	4
1.4 本文主要研究内容	5
第 2 章 智能接触器设计相关技术	6
2.1 交流接触器 PWM 节能技术	6
2.1.1 交流接触器节能技术	6
2.1.2 PWM 节能技术	7
2.2 嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$	8
2.3 CANopen 协议	9
2.3.1 CANopen 物理层及数据链路层	9
2.3.2 CANopen 应用层协议	10
2.3.3 CANopen 从站的设计方案分析	13
2.3.4 CANopen 网络的功能分析	13
2.4 本章小结	14
第 3 章 基于 CANopen 的智能接触器硬件设计	15
3.1 智能接触器总体设计	15
3.1.1 系统设计的功能要求	15
3.1.2 系统设计的技术指标	16
3.1.3 总体硬件结构原理	16
3.2 智能接触器硬件设计	17
3.2.1 MCU 主控电路设计	17
3.2.2 节能模块设计	19
3.2.3 信号采样与调理电路	21
3.2.4 CANopen 通信模块设计	21
3.2.5 人机接口的设置	22
3.2.6 电源设计	23

3.3 本章小结	24
第 4 章 基于 CANopen 的智能接触器嵌入式软件设计	25
4.1 软件设计思想	25
4.2 μ C/OS-II 在 DZ60 上的移植	27
4.2.1 移植条件及主要工作	27
4.2.2 任务划分及任务间通信	29
4.3 DZ60 初始化设置	31
4.3.1 单片机底层初始化及中断设置	31
4.3.2 CAN 模块初始化	32
4.4 用户应用程序设计	34
4.4.1 PWM 节能任务	34
4.4.2 A/D 采样和数据处理任务	37
4.4.3 CANopen 实时通信任务	37
4.5 CANopen 通信任务的软件设计	38
4.5.1 CANopen 协议栈的设计	38
4.5.2 处理接收的报文	39
4.5.3 发送 TPDO 报文	41
4.5.4 发送心跳报文	43
4.6 本章小结	43
第 5 章 系统测试与分析	44
5.1 系统开发环境与测试工具	44
5.2 节能模块测试与分析	45
5.2.1 节能接触器的节能能耗分析	45
5.2.2 节能接触器的电寿命、机械寿命测试与分析	47
5.2.3 EMC 测试与分析	49
5.3 CANopen 从站通信功能测试	51
5.3.1 通信测试平台的构建	51
5.3.2 通信测试结果及分析	52
5.4 测量精度及误差分析	54
5.5 本章小结	54
总结与展望	55
参考文献	57
致 谢	60

附录 A 攻读学位期间发表的论文·····	61
附录 B 攻读学位期间参加的科研项目·····	62
附录 C 部分程序代码·····	63
附录 D 部分 EMC 测试报告·····	71

第1章 绪论

1.1 选题背景

接触器是一种能频繁接通、承载和切断正常运行条件(包括过载)下主电路和较大容量控制电路的电器。据统计,电力系统的能量有一半以上要通过各种接触器分配到各种用电设备中,如电动机、电热设备、照明器具等。其中电磁交流接触器是应用最广泛的一种,其具有低压释放的保护功能,适用于频繁操作和远距离控制,其性能好坏关系着直接的经济利益和电力系统中的电能质量和电能资源。

传统的交流接触器是利用电磁原理通过控制电路和可动衔铁运动来带动触头控制主电路通断的。在我国现在仍然有大量的CJ20系列交流接触器(80年代设计)在使用,同时也存在着部分CJ40系列交流接触器在使用中,这些交流接触器的性能指标与国外同类产品相比存在较大差距。

随着电力电子技术、微电子技术与计算机技术引入低压电器领域,制造出智能化、无弧化的交流接触器已成为可能。目前由于电力资源紧缺,对传统接触器提出了节能运行的要求。且随着新材料以及新工艺的使用,新型交流接触器的机械寿命得到大幅度提高,其最高可达1500万次,但电气寿命却只有机械寿命的十分之一左右,其目前已成为制约接触器发展的瓶颈。普通交流接触器的闭合过程中触头的弹跳是影响电寿命和造成接触器触头侵蚀的最主要因素,特别是二次弹跳^[1]。

自进入新世纪以来,国外一些著名的低压电器公司陆续推出了新一代的交流接触器产品,其共同特点是在这些接触器的电磁机构部分普遍使用了智能控制电子电路,主要是应用在100A以上框架的产品上。即通过引入微控制器,采用闭环控制方法以实现智能合闸操作。这种电路通过反馈方式,使吸力与反力特性达到最佳配合,使电磁铁部分吸合时冲击力达到最小,这样可以使触头振动减少,使交流接触器在AC-3条件下的电气寿命大大提高,同时实现节能^[2]。

因此,迫切需要设计、生产性能指标优良的新型交流接触器。它具有大电压吸持,小电压保持作用,节能率高达90%以上,抗电压波动能力强,可通过现场总线远程通讯和控制。由于其优良的性能指标和相应功能,有着广阔的应用前景。

1.2 课题的国内外研究现状及发展趋势

1.智能电器的国内外现状及发展趋势

智能电器从传统电器发展而来,随着微电子技术、微处理器技术的发展,智

能电器的发展也进入了一个崭新的阶段, 智能电器的概念得到了全新的发展。

智能电器以前被认为是“微机控制十开关电器”, 目前智能电器有如下定义: 智能电器元件是一种能根据故障与操作命令类别来控制电器元件操作机构动作的电器元件, 智能电器采用的是现代传感技术、微机控制技术及计算机数字通信技术, 智能电器能自动监测与识别运行周围的环境或故障类型, 能自动识别操作命令的类型, 能根据故障和操作命令类型来控制电器元件操作机构的动作^{[3][4]}。

在智能电器的发展中, 微处理器的应用和现场总线技术的应用使得智能电器发展迅速, 在单片微机控制技术、电力电子技术、计算机网络和现场总线技术发展的过程中, 电器智能化技术也与此同步发展。

从 20 世纪 80 年代初世界上第一片 8 位 MCU 问世起, 美国、日本、西欧就开始研究如何通过 VLSI 电路(Very Large Scale Integration)技术, 把单片机和其外围电路制做成可与电动机供电电器相吻合的专用 IC(Integrated Circuit)芯片, 用此替代继电器控制电路, 因继电器电路体积过于庞大, 来完成电动机启动、控制和多种保护功能。在 20 世纪 80 年代中期时美国和日本开发出单片机化电动机多功能保护装置, 其是第一代智能电器产品, 并成功的将此类产品推向市场。

在我国国内, 仅经过 20 多年时间, 智能电器早先只采用微机控制取代传统继电控制功能的单一封闭式装置, 后来发展到具有多学科交叉和较完整理论体系的电器智能化系统, 成为电气工程领域工业供电系统、电力开关设备及工业控制网络技术新的发展方向^[5]。目前, 国内对智能电器的研究主要集中在: 设备的在线监测; 新的信号采集处理方法、机理研究; 电器本体的研究; 智能电器可靠性及控制部分抗干扰能力的研究; 通信的实现方法等。

2. 交流接触器节能技术国内外研究现状及发展趋势

节能技术是交流接触器新技术及智能控制方面极为重要的一个部分, 目前国内厂家生产的绝大部分是传统电磁式交流接触器, 07 年产量 4000 多万台, 总产值为 47.6 亿元, 在现场工作的保有量超亿台。一台大、中容量的传统交流接触器吸持功率为几十伏安至几百伏安, 年耗电量为数百度。如果采用节能技术后的新一代交流接触器, 吸持功率为几伏安。按照现场工作的保有量 1 亿台、每台年耗电量 100 度计算, 每年可节省电量 100 亿度, 直接经济价值 77 亿元, 间接经济价值、环保意义和社会价值则更大。

传统交流接触器工作原理为: 通电吸合, 带电保持, 断电释放。在交流接触器的运行过程中, 总能耗的 90% 被铁心和短路环的磁滞损耗和涡流损耗占据, 且传统接触器功率因数低, 线圈温升高, 使电网线路上的电能损耗进一步加大, 其使用寿命低下。

为改善接触器的这种状况, 我国很早就开始采取节电措施。早在 1981 年 56

号文件中的第2号节电指令就指出要实施交流接触器节电措施。电力电子技术以及数字技术的广泛应用让接触器的节电技术获得了新发展。为了适应交流接触器节电产品发展的需要,我国又颁布了国家标准 GB8871—1998^{[6][7]}。从那时起,我国交流接触器的节能技术一直在长足发展中,交流接触器节电技术的发展共分为三个阶段:交流接触器节电器、交流接触器节电线圈和节电型交流接触器。

在节电型交流接触器中,共有3种节电方式:机械锁扣式、剩磁或永磁式和脉宽调制直流操作式。前两者都出现过但是由于缺点过于明显,造成了很大的经济损失和材料的浪费,因此目前着重发展脉宽调制直流操作式。

在脉宽调制直流操作式技术方面,国外发展较快且取得了很大的成果。为了适应工业自动化控制系统发展的需要,国外一些主要交流接触器生产厂家相继推陈出新一代交流接触器,如法国施耐德公司的LC1-D系列、美国Squared公司的P系列、德国西门子的3TF系列、金钟·默勒公司的DIL系列、日本三菱公司的MS-N系列等。这些产品的主要指标,如最高额定工作电压660V,机械寿命1000-1500万次,电寿命100-120万次(AC-3),最高操作频率1200次/h等,基本上已经处于相当高的水平,能满足各种控制系统的要求,具有强大的市场竞争能力。

国内近年来研究人员也展开了大量的研究性工作。福州大学的研究者提出了直流脉动电压激磁的控制理论,此理论工作原理为:先经实验得到不同电压下的最佳延时时间参数TA、TB,存储在微处理器中,接触器则可以根据不同的电压自行确定电磁机构的最佳运行参数^[8];西安交通大学则提出最小吸合电压理论^[8]。

3.现场总线在智能交流接触器的应用

进入20世纪90年代以来,因电网自动化要求,电气操作人员希望能采用远程控制方式了解和控制电网的电气状态和参数,便于集成管理,同时实现终端控制以及工厂办公自动化的发展,迫切需要智能型可通信电器的不断发展和完善。

为使低压配电系统中的主控设备与智能化电器实现高速双向数据通信,形成智能化监控、保护与信息控制网络,需要一种高速可靠的总线传输。近年来,具有各种特点的现场总线逐步涌现,给工业自动化领域带来了新鲜的动力^[9]。现场总线的运用使操作人员可以方便地在控制室控制各种低压电器设备,及时了解现场各种设备的运行情况,并实时处理故障。

现场总线是一种工业数据通信总线,由于现场总线设备智能化、数字化的特点,使其从根本上提高了测量与控制的精确度,加上系统连线简单,提高了系统的工作可靠性。同时,系统具有现场设备的在线故障诊断、报警、记录功能,并可完成现场设备的远程参数设定、修改等工作,增强了系统的可维护性^[10]。

在现行的众多现场总线中,从目前技术水平看,没有绝对先进的现场总线,每一种总线都有它的优点,也有其不足之处。目前,在工业控制领域有多种现场

总线被使用, 本文中使用 CANopen 总线作为智能接触器实现组网功能和远程控制的现场总线。CANopen 是在 CAN 总线的基础上发展而来, CAN 总线应用广泛是因为其成本低廉、通信实时性好、纠错能力强等特点, 而 CANopen 是 CAN 总线的应用层协议, 除这些特点外, CANopen 总线采用了中心授权机制, 网络管理功能由一个主节点实现, 且其通信机制简单而可靠性高。CANopen 的报文是依据优先级来发送的, 但是同一个高优先级的报文不允许一直占用总线, 从而给低优先级的报文发送时间^[1]。

因此, 选择了 CANopen 总线作为远程监控交流接触器的媒介。使交流接触器的状态和各项参数能够得到实时监测和控制, 实现智能化。使其从基本保护功能发展到智能化、网络化的保护功能。

1.3 课题来源及意义

1. 课题来源及作者承担的科研开发任务

本课题是湖南大学与浙江天正股份有限公司共同承担的校企合作项目, 来源于浙江省第二批重大科技专项“基于现场总线的智能断路器技术研究与应用”[2007C11072]。该项目于2010年11月通过浙江省科技厅验收, 研究成果已投入实际应用, 开发的各种现场总线通信接口、智能控制器及智能断路器等系列化产品已基本实现了产业化。

以上述项目为支撑, 本论文扩展其研究领域, 主要致力于低压电器中另一重要器件-交流接触器的节能和智能化研究, 并将现场总线 CANopen 协议应用于其三三相主回路的控制、保护和通信, 使之具有高性能、高可靠性、低成本、小型化、智能化、可通信的特点。作者主要研究交流接触器电磁铁部分 PWM 节能技术的软硬件设计, 使其实现节能、长寿命、耐电压波动、无声等基本功能; 同时对交流接触器的三相主回路进行采样、数据处理、CANopen 网络通讯, 增添对接触器三相主电路的保护功能、状态控制及网络通讯功能。

2. 课题意义

随着低压电器行业的不断发展和新标准的出台, 电力电子技术、计算机技术、数字通信技术在低压电器行业的不断渗透和应用, 至2012年, 我国第一代产品将面临全面淘汰, 第二代产品将成低档产品, 第三代产品降为中档产品。目前第三代产品正逐步成为市场的主力军, 随着我国第三代产品的开发与应用, 我国低压电器产品与国外先进水平差距已缩短至十年。从产品发展和不同层次市场需求考虑, 有必要发展第四代低压电器产品。第四代低压电器产品具有智能化、监控和保护功能、故障自检功能、人机交互功能等特点。基于以上考虑, 有必要研制第四代产品, 以提高我国在国际电力电器领域内技术与产品的竞争力。因此, 研制

和开发基于 CANopen 总线的智能型接触器意义重大。

1.4 本文主要研究内容

第1章 绪论。论述了课题的选题背景，给出了交流接触器的国内外发展现状以及目前研究的重点-交流接触器的节能、低耗及保护功能，并阐述了课题的研究意义。

第2章 智能接触器设计技术基础。主要分析了智能交流接触器的技术基础，包括交流接触器的 PWM 节能技术、嵌入式软件 $\mu\text{C}/\text{OS-II}$ 、CANopen 应用层协议的分析及实现原理等。

第3章 智能接触器的硬件设计。结合天正电气的企业标准，给出了系统设计的功能要求和具体指标，提出了智能接触器的总体方案。给出了整个电路系统的原理图，包括主控电路、节能模块、信号采集与调理模块、CANopen 通信模块、人机接口模块、电源模块。

第4章 智能接触器的嵌入式软件设计。在嵌入式平台 $\mu\text{C}/\text{OS-II}$ 操作系统的基础上，结合硬件电路，开发了智能接触器的软件系统。嵌入式软件的设计主要包括 $\mu\text{C}/\text{OS-II}$ 在 DZ60 上的移植、各个用户任务的设计(PWM 节能任务、A/D 采样及数据处理任务、CANopen 实时通信任务)，其中详细阐述了 CANopen 协议栈的实现及各个通信对象的设计。

第5章 系统测试与分析。采用 Freescale 公司的开发软件 CodeWarrior 开发系统程序，并搭建了节能模块的测试平台和 CANopen 从站通信的测试平台，对节能模块的相关指标进行分析，并通过模拟主站发送各项命令来进行远程控制，利用 ZLGCANTest 观察总线上传输的报文，结果表明这些报文的帧 ID、数据字节的定义以及整个网络的运行过程都符合 CANopen 规范。

最后，对本文研究工作进行总结与展望。

第 2 章 智能接触器设计相关技术

智能交流接触器是通过调节线圈电压实现大电压吸合、小电压吸持来实现节能作用的，同时对三相主回路的电压进行采样并通过单片机软件判断过压和欠压，并通过上位机进行实时控制等实现智能控制及保护功能。本章从交流接触器的 PWM 节能技术、嵌入式软件 $\mu\text{C}/\text{OS-II}$ 、CANopen 应用层协议三个方面来讨论其技术基础。

2.1 交流接触器 PWM 节能技术

2.1.1 交流接触器节能技术

传统电磁铁吸合力模型的计算公式为^[12]：

$$F = \frac{L^2 I^2}{N^2 \mu_0 S} \quad (2.1)$$

式中： L —电感量； I —线圈流过电流； N —线圈的匝数； μ_0 —空气磁导率； S —电磁衔铁与磁轭的相对极的截面积。

根据交流接触器节能技术的发展可知，采用将其操作电磁系统的电流由交流改为直流这一节电技术达到节能，但是节电操作由吸合到吸持不同的转换情况会对交流接触器的闭合性能产生影响。要从吸合状态转换到吸持状态，在转换过程中，由于电路中有电感，操作线圈中的电流不会突变，而是有一个衰减的过渡过程，见图 2.1。

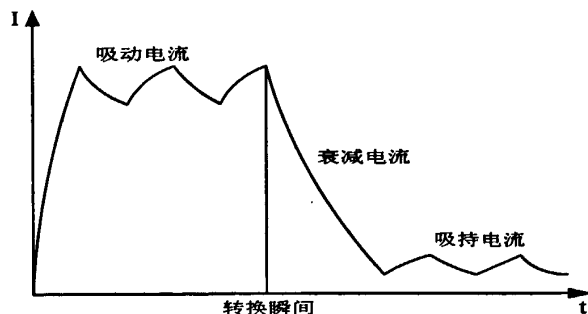


图 2.1 使用节能技术的电流变化示意图

根据 GB14048.4-2010 规定，交流接触器的电磁铁部分应在 85%~110% 额定控制电源电压下能够可靠吸合。通常工厂为保证交流接触器在较低电压下仍可以可靠工作，把电磁铁部分的吸合电压设定在额定电压的 65%~75%。图 2.2 为吸力与反力特性配合曲线。假如吸合过程的电磁铁吸力特性 $F_{\text{吸上}} = f(\delta)$ 与反力特性 $F_f = f(\delta)$ 像图 2.2 中那样配合，则当在额定电压下正常工作时，操作电磁铁的 $F_{\text{额定}}$

会大大高于反力特性，吸合过程中可动部分动能会太大，会造成不必要的电能损失，且容易引起触头弹跳^[13]。从图 2.2 可知，随着工作气隙的减小，吸力增大得很快，而电磁铁有在吸上位置电磁吸力裕度很大的特点，因此可以通过 PWM 技术使线圈电压从触头吸合后减小并保持一定的低电压，达到节能目的。

国标规定中的控制电源电压 85%~110% 波动条件下，假如能通过一个带电压反馈的控制系统，使得接触器线圈电压保持不变，并让对应的吸力特性 $F_{智能}$ 仅稍高于反力特性，这样可让电磁铁模块的可动部分的动能降低，减小触头上的弹跳和提高电寿命。一种可行的方式见图 2.3，图中为吸合过程中接触器线圈的电压变化情况。在吸合时，当外界变化时通过电压反馈系统，保持 $U_{智能}$ 不变；当电磁铁部分的动静铁芯吸上后，供给线圈的电压降低，电压维持在 $U_{保持}$ 用以节能^[14]。

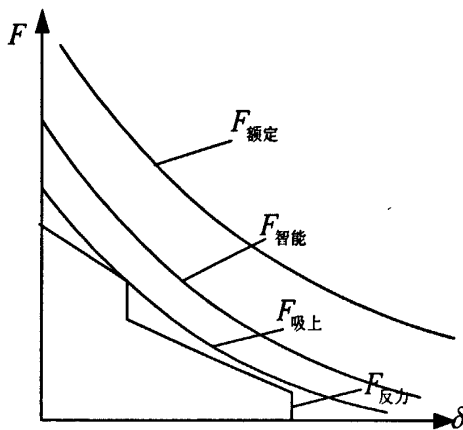


图 2.2 吸力与反力特性配合曲线

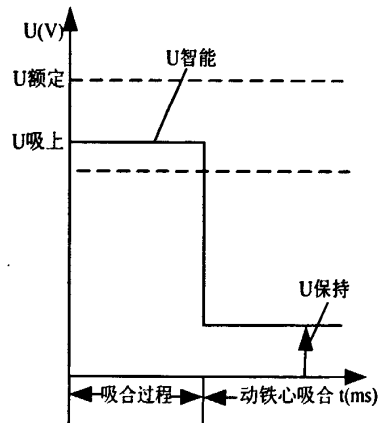


图 2.3 智能控制的线圈供电电压示意图

2.1.2 PWM 节能技术

由上面可知，交流接触器的电磁铁控制电源不论是直流还是交流，都会经历吸合（衔铁的运动阶段）、吸合后的弹跳（闭合后的稳定阶段）、吸持（改变线圈电压使其仅为克服弹簧的反力阶段）。要达到此设计目的即采用大电压吸合、小电压吸持设计方案，且转换的时间要尽量短。

PWM 技术是电力电子技术中的一种控制技术，它是利用面积等效原理将整流后的正弦半波通过控制开关管转换为等幅不等宽的脉冲波，脉冲波的宽度呈正弦变化。而这种脉冲波恰好是接触器线圈要达到大电压吸合、小电压吸持所需的脉冲波。PWM 调制技术有三种实现方案，计算法、调制法、跟踪控制法。在此采用的是调制技术。图 2.4 为 PWM 调制技术的示意图。

本文中描述的节能原理利用单片机电路和一个带反馈的调压电路来实现。交流控制电源经防浪涌抑制、滤波、整流后，通过 PWM 调制电路供电给接触器线圈，采样线圈电压并将之反馈给单片机，按照事先存储在单片机中的吸反力特性最佳

配合曲线调节 PWM 口输出脉冲占空比，以此调节线圈电压，从而实现对线圈上电压的控制。在吸合阶段输入宽脉冲电压，使触头安全可靠地吸合，减少了触头电磨损，使电寿命大大提高。在吸持阶段输入窄脉冲电压，使闭合后的触头保持在低压节能运行状态。宽脉冲吸合、窄脉冲吸持方案的波形图见图2.5。其中PWM调制的实现是通过开关管IGBT的导通和关断来实现的^[15]。

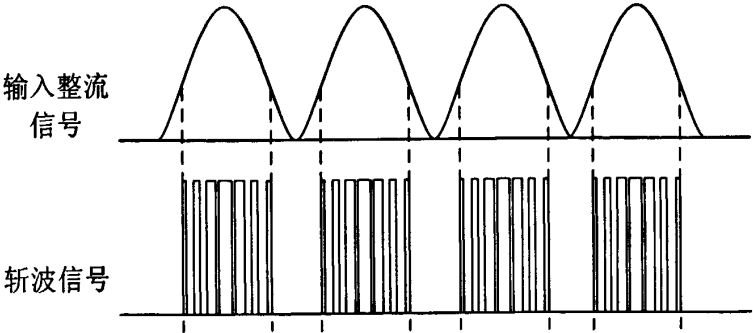


图 2.4 PWM 调制技术示意图

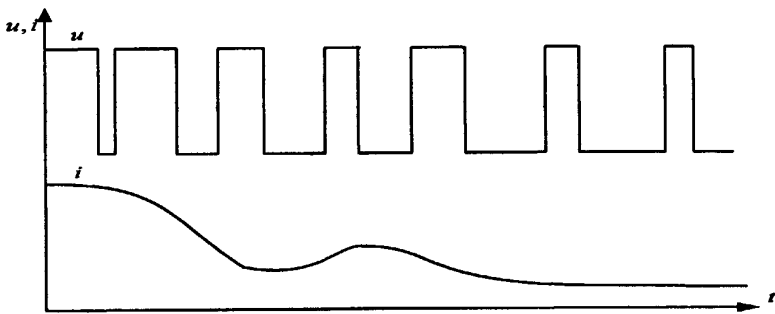


图 2.5 PWM 脉宽调制方式波形图

2.2 嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$

$\mu\text{C}/\text{OS-II}$ 是由美国工程师Jean.J.Labrosse于1992年编写的一个开源代码的、微内核嵌入式实时操作系统，于2000年得到了FAA(Federal Aviation Administration)的认证，该认证是针对用于商用飞机的、符合RTCA DO-17813标准， $\mu\text{C}/\text{OS-II}$ 的稳定性和安全性因此得到证明。 $\mu\text{C}/\text{OS-II}$ 系统采用C语言与汇编语言编写，其中绝大部分代码的编写采用C语言，汇编语言只是用来编写与处理器密切相关的极少部分代码，用户要想将之移植到各类嵌入式MCU上只要做很少的工作即可^[16]。

$\mu\text{C}/\text{OS-II}$ 操作系统是基于优先级的抢占式的硬实时内核，包含了实时内核、任务管理、时间管理、任务间通信同步(信号量，邮箱，消息队列)和内存管理等功能。 $\mu\text{C}/\text{OS-II}$ 的体系结构见图2.6。

$\mu\text{C}/\text{OS-II}$ 还有2个重要的特点：(1)其是基于优先级的可剥夺型内核；(2)区分用户空间和系统空间。系统中的所有任务都有一个唯一的优先级别，采用 $\mu\text{C}/\text{OS-II}$ 平台，各个任务可以独立工作，互不干涉，很容易实现准时而且无误执行，让实

时应用程序的设计与扩展更加容易；它区分用户空间和系统空间，所以很适合应用在比较简单的处理器上^[17]。

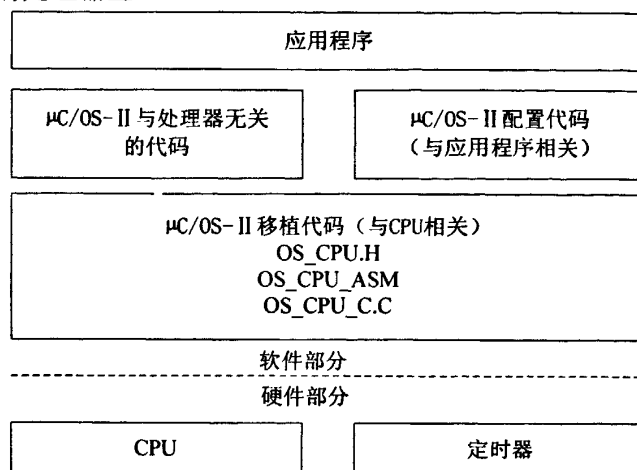


图 2.6 μC/OS-II 体系结构

2.3 CANopen 协议

由OSI网络模型可知，CAN总线仅仅定义了第1层(物理层)、第2层(数据链路层)，但应用层没有定义，是一个不完整的协议，如果有一个高层协议定义CAN报文中的11/29位标识符和数据域中8字节的具体使用的话，将会十分完整^[18]。CiA (CAN in Automation)推荐使用的CANopen协议，支持各种CAN厂商设备的互用互换，是一个开放的、标准化的高层协议，能够在CAN网络中提供标准的、统一的通讯模式，提供具体设备的功能描述方式，执行CANopen网络管理功能^[10]。

2.3.1 CANopen 物理层及数据链路层

1. CANopen 物理层

(1)收发器的规定及连接器引脚分配

根据ISO11898，使用高速收发器最高的速率时，VCAN_H与VCAN_L相差+16V，总线节点之间的隔离可选。推荐的收发器可维持线缆的不当连接，从0V到+30V。CANopen设备通讯使用的物理介质是标准的带公共回路的差分驱动两线总线，CAN-bus信号连接到5引脚OPEN5连接器^[10]。

(2)每个CANopen模块都必须支持指定位速率中的一种。

CANopen总线的直接通信距离最远可达5km，通信速率最高可达1Mbps(此时通信距离最长为25m)，CANopen总线上的节点数主要取决于总线驱动电路，最多可达127个。报文采用短帧结构，传输时间短，受干扰概率低，使得数据出错率极低得到了保证。通信介质可为双绞线、同轴电缆或光纤，选择灵活。在报文标识符上，CANopen上的节点分成不同的优先级，可满足不同的实时要求，优先级高

的数据最多可在134us内得到传输。在错误严重的情况下总线还具有自动关闭输出功能，其他节点不受影响^[10]。

(3)指示灯的规范

指示灯的通常功能是帮助服务人员不需要检查每个器件的手册而快速识别通讯错误。CANopen 协议中建议支持两种指示灯：一个红色的ERR LED(指示CAN物理层的错误状态)、一个绿色的RUN LED(指示 CANopen 网络状态机构的状态)。

2.CANopen 数据链路层

包括逻辑链路控制子层和介质访问控制子层。在CAN 2.0A规范中，数据链路层的LLC子层和MAC子层的服务及功能分别被解释为“目标层”和“传输层”。LLC子层作用范围包括：为远程数据请求以及数据传输提供服务；确定LLC子层接受的报文中哪些报文实际上被验收；为恢复管理和过载通知提供手段。MAC子层的作用主要是传送规则，即控制帧的结构、错误检测、执行仲裁、标定错误、界定故障等。

2.3.2 CANopen 应用层协议

CANopen 协议是基于CAN总线的应用层协议,其物理层是一致的,但是在 CiA Draft Standard 301、303 中更加详细的强调了CANopen 协议的一些特性^[19]。在CANopen 协议中最重要的部分莫过于对象字典和通信对象。

1.对象字典

对象字典(Object Dictionary)是CANopen 协议中最核心的部分。对象字典有如一张表，描述着CANopen 的配置和节点功能，同时承载着整个网络的信息和数据。CANopen 网络中，应用程序可以通过使用不同的通信对象读写对象字典实现CANopen 通信，通信对象、应用对象、对象字典三者一一对应。

对象字典是一个有序的、公共的、开放性的表格，每个应用对象采用一个16位的主索引（共65535个）和多个8位的子索引(256个)值来寻址。为了满足不同厂商设备间CAN模块的互联和通用性，CANopen 协议中详细的描述和规定了对象字典的构建。表2.1是一个完整的对象字典模型，建立对象字典时需要按照此模型来建立。

表 2.1 对象字典元素模型

索引	对象	名称	属性	类型	M/O
----	----	----	----	----	-----

2.通信对象

CANopen 网络的通信和管理都是通过不同的通信对象来完成的。考虑到工业自动化系统中数据流量的不同需要，CANopen 规范定义了4类标准的通信对象来实现通信、网络管理、紧急情况处理等功能。

(1)过程数据对象 (PDO)

PDO 通讯可以用生产者/消费者模型来描述。

图 2.7 为 PDO 通讯示意图, 过程数据可以从一个设备(生产者)发送到另一个设备(消费者)或者到几个其它设备(广播)。PDO 均以非确认的方式来发送。生产者发出带特定标识符的发送 (T_PDO), 它和一个或多个消费者的接收 (R_PDO)标识符相对应^[19]。

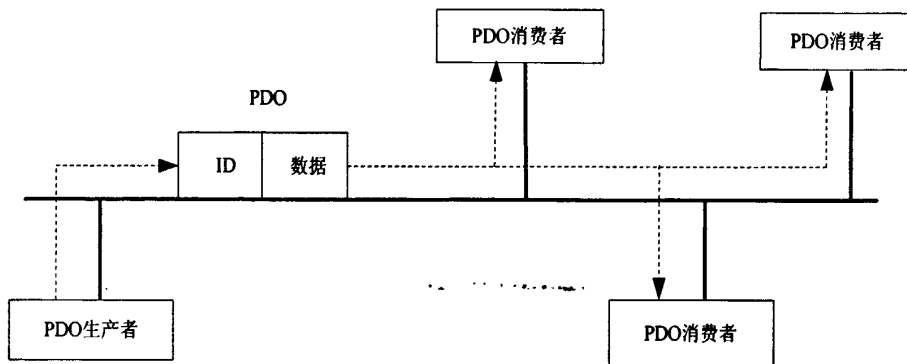


图 2.7 PDO 通讯示意图

根据实际的情况确定一个设备的 PDO 报文的个数和长度, 必须在设备描述和预定义连接集里规定好。每个 PDO 报文只可以最多传送 8 个字节的数据。PDO 在 OD 里有对应的具体元素, PDO 因此提供了对象字典和应用对象的接口。在对象字典中用 2 个对象来描述每个 PDO 报文: PDO 通讯参数和 PDO 映射参数。PDO 通讯参数用来描述 PDO 的通讯行为; PDO 映射参数用来描述所映射的应用对象的内容。只要在对象字典中定义好 PDO 的通讯参数和映射参数, 就可生成一条 PDO 发送报文。

(2)服务数据对象 (SDO)

SDO 通讯可以描述成客户/服务器模型。

两节点中主动的节点是客户, 被动的节点是服务器, 客户发出一个请求(上传/下载), 触发服务器去执行某个任务, 结束任务后服务器回应这个请求, 见图 2.8。

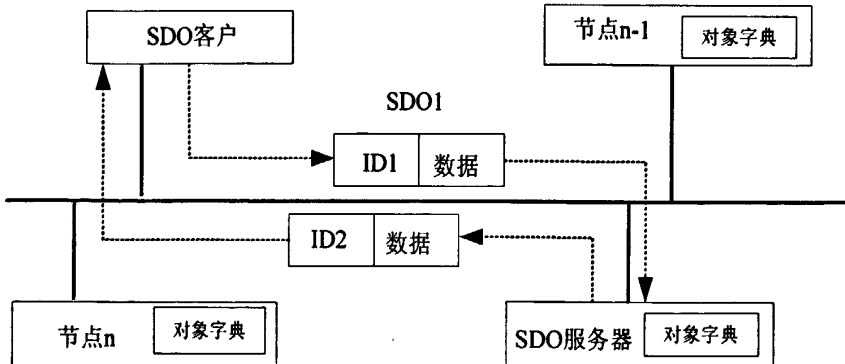


图 2.8 SDO 通讯示意图

SDO 通讯用于主节点对从节点对象字典的读写访问, 以实现从节点参数和

功能的设置，譬如定义 PDO 的通讯类型和数据格式等。SDO 报文优先级较低。从节点在使用 SDO 报文时会用到两个不同标识符的 CAN 数据帧，一个用于主节点请求数据时的操作，一个用于操作完后发响应帧给主节点。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^{[19][20]}。

(3)网络管理对象(NMT)

CANopen 网络管理是基于节点的且采用主/从结构。CANopen 网络中要求有一个设备专门作为 NMT 的管理者，其他节点都是 NMT 的从属。在网络上电过程中，NMT 主节点通过 NMT 服务控制从节点的初始化进程，为监视节点提供错误控制服务和网络通信状态，从网络中的一个模块到其他各个节点的配置数据的上传和下载的配置控制服务。每个在 CANopen 网络中的从节点都有初始化、预运行、运行和停止四个状态，由主节点发命令控制各状态间的转换。

当系统上电后，NMT 从节点依次完成初始化中的三个初始化子状态，自动进入预运行状态并发送一个标志信息。通过发送一个两字节的 NMT 报文启动远程节点请求，节点可直接转为运行状态，在运行态，所有通信对象都可以起作用。节点可以由任意一个状态进入初始化状态中的后两个子状态(复位应用程序和复位通信)，完成节点的硬件初始化和通讯初始化^[21]。

(4)心跳报文(Heartbeat)

在 NMT 网络管理服务里，主要包括 NMT 模式控制和 NMT 错误控制，NMT 模式控制在③中已经描述。NMT 错误控制包括节点保护(NodeGuarding)、心跳报文(Heartbeat)和 Boot-up 协议。一个节点不能同时支持 NodeGuarding 和 Heartbeat 协议。对于新的设计，CiA 推荐使用心跳协议，心跳协议的通讯示意图见图 2.9。

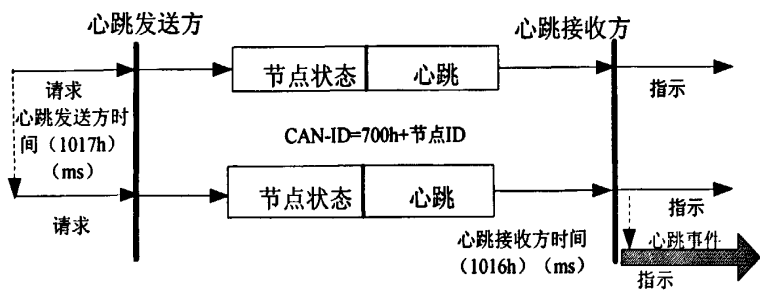


图 2.9 心跳协议通讯示意图

CANopen 设备周期性的传送其“心跳”。周期称为发送方心跳时间，在对象 1017h 中进行组态。心跳接收方在接收到心跳消息后评估相关设备是否仍在正常工作以及是否处于正确的网络状态。心跳接收方功能在对象 1016h 中进行组态。在心跳周期内，必须接收至少一条相关心跳发送方的心跳信息。

第 4 类通讯对象是特殊功能对象。CANopen 定义了 3 个特定用于同步、紧急状态表示和时间标记传送的特殊功能对象。在从节点中，一般特殊功能对象报文

使用不多，略。

2.3.3 CANopen 从站的设计方案分析

CANopen 从站节点主要采用模块化设计，主要应包括节能模块、电压采样模块、CANopen 通讯模块、MCU 处理模块等几大部分。从站的核心问题除了节能设计外，主要是如何实现 CANopen 通讯协议，通常有以下几种方案^[22]：

(1) 单片机+CAN 控制器SJA1000+CAN 收发器PCA82C250

该方案使用芯片较多，硬件设计复杂，自行开发协议栈；但成本较低。

(2)带 CAN 控制器的单片机(MC9S08DZ60)+CAN 收发器AMIS30660

该方案也是自行开发协议栈，成本较高；但是使用芯片少，硬件设计简单，抗干扰强。

(3) CANopenIA-XA CANopen 协议控制芯片+外围器件

该方案成本最高，硬件设计并不是最简单；优点是不用开发协议栈，但较难自己维护。

经过综合比较，在以上3种方案中，第二种方案最为合适。采用一款freescale 的HCS08 系列内嵌 CAN 控制器的单片机 MC9S08DZ60 与高速 CAN 收发器 AMIS30660 芯片组合实现 CANopen 通讯协议。

2.3.4 CANopen 网络的功能分析

在本设计中每个智能接触器都要接入 CANopen 网络，形成一个网络结构，因此应对 CANopen 网络进行功能分析。CANopen 通信网络一般由一个主节点和多个从节点组成^{[23][24]}。主节点负责整个网络的管理，从节点负责实时数据的采集和通信，低压电器的判断和控制。一般来讲，每个从节点分别采集现场设备的电压、电流等实时数据，并通过 MCU 进行保护，将线路状态、三相电压、三相电流、通信状态等发送给主站进行统一管理；同时接收主站来的管理命令，如遥控，参数配置、保护等，对现场设备进行控制和保护，以免设备的损坏，见图 2.10。

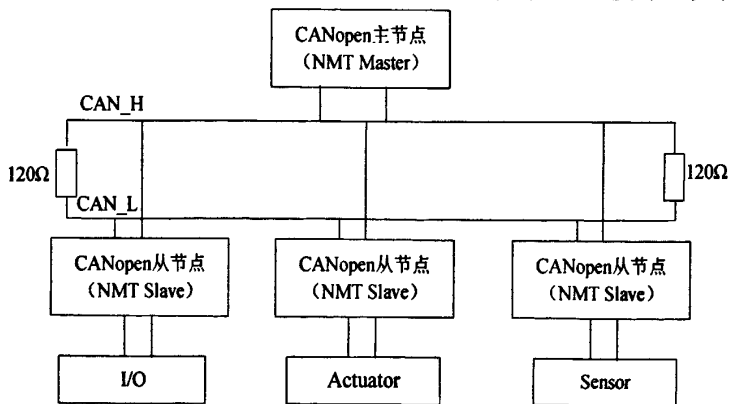


图 2.10 CANopen 网络结构

由 CANopen 协议可知，对于主从节点的通信需要规定好通信规约，所有的节点按照通信规约进行设计，主要是配置好应用对象、通信对象、对象字典三者^{[25][26]}。

一个智能接触器的从节点需要传送的数据有 A、B、C 三相电压，交流接触器规格，工作状态，吸合状态，分合闸指令等数据，而在 CANopen 协议中有 TPDO、RPDO、SDO、NMT、Heartbeat 等通讯对象，根据 CANopen 协议的子设备 DSP 401 的规定和具体的应用分析，应用对象与通信对象的分配见表 2.2。

表 2.2 应用对象与通信对象的分配表

应用对象	字节数	通讯对象
$U_{A有}$	2	TPDO
$U_{B有}$	2	
$U_{C有}$	2	
交流接触器规格	2	RPDO
工作状态	1	
吸合状态	1	
分合闸指令	2	
从节点状态		
从节点状态转换		Heartbeat
从节点参数配置		NMT
		SDO

按照预定义的连接集，通信对象和对象字典已经配置完成，需将应用对象配置到对象字典中，供 CANopen 协议栈使用，见表 2.3。同时，主从节点之间的交流使用 CANopen 协议中规定的数据帧。

表 2.3 对象字典与应用对象的分配表

节点	索引	子索引	应用对象	类型
一台地址为 0x01 的接触器	0x6000	0x01	$U_{A有}$	INT16U
	0x6001	0x01	$U_{B有}$	INT16U
	0x6002	0x01	$U_{C有}$	INT16U
	0x6003	0x01	交流接触器规格	INT8U
	0x6004	0x01	工作状态	INT8U
	0x6005	0x01	吸合状态	INT8U

2.4 本章小结

本章阐述了智能接触器的相关技术基础，重点介绍了与本文密切相关的 PWM 节能技术、嵌入式软件 $\mu C/OS-II$ 、CANopen 应用层协议三个方面，接着对 CANopen 从站的设计和 CANopen 组网的功能进行了分析，为后续的具体硬软件设计打下了坚实的基础。

第3章 基于 CANopen 的智能接触器硬件设计

智能电气监控单元需要在运行现场对不同类型、不同物理属性的参量进行测量,并根据测量结果对开关电器进行操作控制,完成对现场设备和开关电器自身的控制和保护^[27]。为了实现对交流接触器的智能控制和能耗控制,本章在了解智能监控单元硬件总体结构及功能的基础上,介绍单元中各模块的设计原理及具体的电路设计。

3.1 智能接触器总体设计

3.1.1 系统设计的功能要求

依据国家标准 GB14048.1-2006、GB14048.4-2010 以及国际标准 SEMI-F47-0200 等要求,本文设计的智能交流接触器主要用于交流 50Hz(或 60Hz)、额定工作电压达 660V,在 AC-3 使用类别下额定工作电压 380V 时,额定工作电流达 800A 的电力系统中,供远距离接通和分断电路,并可与适当的热继电器组成电磁起动器以保护可能发生过载或断相的电路。其应具有如下主要功能:

1. 直流起动、直流吸持的节能无声运行

在交流接触器电磁系统消耗的功率中,铁心损耗与分磁环损耗占总损耗的绝大部分。如果采用电子电路,线圈在保持状态所需的功耗将明显下降,有功功率和视在功率均为同规格普通交流接触器 5%~10%。这样,功耗可以降至最小值,带来不小的节能效益。同时由于线圈能耗的降低带来的电磁系统优化设计以及最佳吸反力特性的配合使触头烧损量降低等,可使铜、银等材料消耗量明显减少。符合国家提出的节能降耗政策,顺应时代发展趋势。

2. 监控及保护功能

主要通过对交流接触器三相主回路的电压进行采样、处理,并上传于上位机进行控制,从而通过上位机对接触器进行监控及保护功能。当发现接触器三相电压出现异常、骤降、骤升等,即时启动电磁铁部分的分闸电路,断开接触器,以免造成后面所接的重要器件(如电动机)等严重烧损。

3. 通信组网功能

智能交流接触器采用 CANopen 总线实现了与主控计算机双向通信的功能。主控计算机既可以显示智能交流接触器的工作状态信息,又可以控制交流接触器的接通和分断;实现远程控制。控制模块中采用了压敏电阻、TVS 管等硬件抗干扰技术,实现了在脉冲群、浪涌等恶劣电磁干扰环境中正常通信与运行,具有良好

的电磁兼容能力。另外上位机也可以同时监测多达 110 台接触器，在接触器使用较多的场合尤其适合。

3.1.2 系统设计的技术指标

根据目前低压电器领域的实际需求，本文设计的基于 CANopen 协议的智能接触器需要达到以下具体指标：

- (1) 能耗：有功功率和视在功率均为同规格普通交流接触器5%~10%；
- (2) 自带浪涌吸收功能，通过相关EMC试验；
- (3) 满足AC-3电寿命80万次、机械寿命800万次(TGC5-500)；
- (4) 吸合门槛电压(热态)：70%Usmin±5%；释放门槛电压(冷态)：55%Usmin±5%；运行极限电压范围：70% Usmin ~ 115%Usmax；满足国际标准 SEMI-F47-0200；20ms失电保护。

3.1.3 总体硬件结构原理

带通信功能的智能接触器的硬件框图见图 3.1。按结构功能可以划分为以下几个主要模块：带电压反馈PWM 节能模块、三相主回路电压采样及调理模块、MCU 主控模块、CANopen 通信模块、电源模块等。其中单片机系统为整个智能接触器的核心，CPU 的选择对提高系统工作性能及简化系统设计起着极其重要的作用。本文采用了飞思卡尔公司的 MC9S08DZ60 单片机，其具有 40M 的处理速度，60K 的 flash，内嵌 CAN 控制器，使用这种单片机可以尽量减少外围元件，从而降低系统成本，减少功耗并增加系统的可靠性。

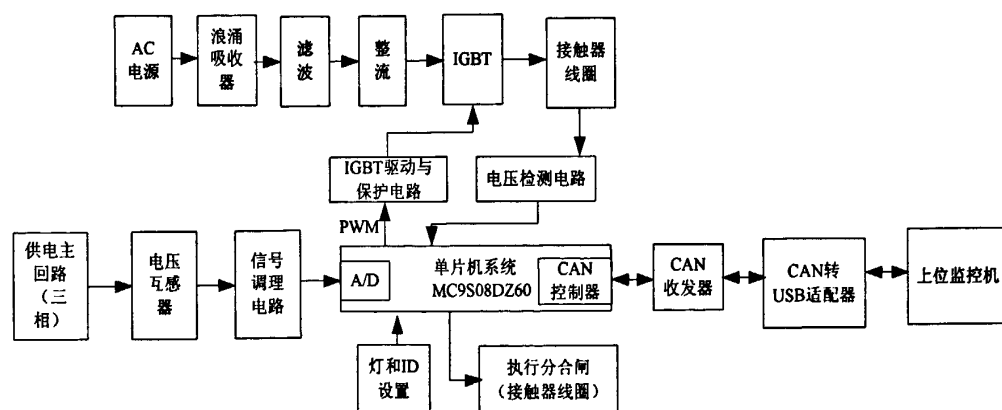


图3.1 总体硬件结构框图

接触器工作时，PWM 节能模块不断检测控制电源并输入单片机进行比较，当发现电源大于吸合电压（70%Usmin）时，通知单片机调节占空比调制输入波形控

制IGBT形成大电压带动衔铁运动，再很快接着调节占空比转为吸持状态等待释放和关闭开关管，完成接触器的节能吸合的功能。

同时，对输入端的三相主回路的实时电压采样，输入单片机进行A/D转换变成数字信号，然后进行有效值计算，计算后判断电压是否过压或是欠压，控制线圈分闸保护交流接触器。三相电压有效值同时送入CANopen总线，由上位机判断、控制状态和动作。CANopen通信模块主要完成于上位机的双向通信，将智能型接触器的状态、参数、数据传递给上位机，同时也通过上位机进行命令控制，实现远程监控。

线路板的电源由工业用电的DC24V供电，在电源模块的设计下，使线路板得到稳定的电源信号。

3.2 智能接触器硬件设计

3.2.1 MCU 主控电路设计

由于本设计中的MCU需要支持 $\mu\text{C}/\text{OS-II}$ 实时操作系统，且能支持一定的并行任务，具有实时中断，具有一定容量的内存空间，同时需要内嵌CAN控制器，减少电路所受外界干扰，但是价格因素也要考虑在内，因此本设计选用Freescale公司生产的高性能、低价位、外围设备丰富的8位单片机MC9S08DZ60作为主控制器^[28]。

1. MC9S08DZ60 单片机特点与结构

MC9S08DZ60具有抗干扰能力强、外围设备丰富、成本低、具有可靠的系统保护、易于小型化等特点，是HCS08家族中性价比较高的一员，特别适合本设计的开发。主要功能特性如下：

(1) 具有丰富的外围设备

包括：24通道，12位数模转换器(ADC)，2.5 μs 转换时间，自动比较功能，1.7 mV/ $^{\circ}\text{C}$ 温度传感器，包含内部能隙参考源通道；内嵌CAN控制器的MSCAN模块，CAN协议-V2.0 A和B；支持标准和扩展数据帧；支持远程帧；5个带有FIFO存储机制的接收缓冲器；灵活的接收识别符过滤器，可编程如下：2*32位、4*16位或8*8位；含定时器脉冲宽度调节器TPM，一个6通道(TPM1)和一个2通道(TPM2)；可支持输入捕捉，输出比较，或每个通道带缓冲的边沿对齐PWM输出。

(2) 存储空间

容量多达60KB的片内Flash存储器与最大4K的RAM，能满足多数应用要求，包括移植的 $\mu\text{C}/\text{OS-II}$ 所占空间，不必再额外扩展存储器，Flash存储器具有在线编程能力，可实时编译和调试程序，方便编写程序后的查错，可缩短开发周期。

(3) 编程方式与中断支持

拥有 32 种不同的外部与内部中断源，提高数据的处理速度。HCS08 系列为浮动式堆栈，为使用高级语言编程创造了条件，目标代码完全向上兼容 M68HC05 和 M68HC08 系列，16 位堆栈指针（64Kb 地址空间内任意规模的堆栈），且支持 C 语言中嵌套汇编语言，进行混合编程。

(4) 多种操作模式

DZ60支持4种操作模式。主动后台模式：采用低功耗设计用于代码开发；等待模式：CPU 关闭以省电，系统时钟正常运行，内部稳压器正常工作；停止模式：系统时钟被关闭，内部稳压器处于待机状态；运行模式：MC9S08DZ60 系列产品的正常操作模式。代码在复位完成后运行，并且从内存0xFFFFE - 0xFFFF上获取其起始地址。

2. MC9S08DZ60 主控电路

图 3.2 给出了整个硬件系统中最重要的主控制器电路图。在这里它主要完成三相主回路的电压实时采样、节能电路的直流脉宽占空比调制、节能电路的线圈控制电源电压采样、接触器状态控制、CANopen 双向通讯等功能。

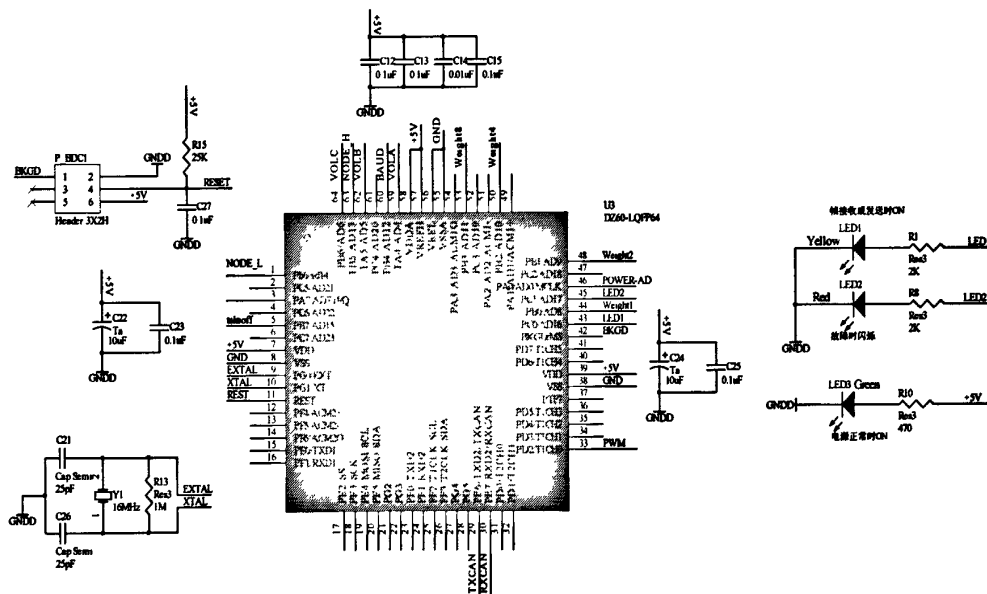


图 3.2 主控制器电路图

按照 DZ60 的 datasheet 和管脚、功能分配, 图中的 PA4-PA6 (VOLA, VOLB, VOLC) 用于调理后的三相主回路的电压采样及模数转换; Power-AD 和 PWM 引脚用于电磁铁部分的节能实现及控制电源电压采样; Weight1, weight2, weight4, weight8 以及 NODE_H, NODE_L, BAUD 用于连接编码开关, 进行每个从节点接触器的 ID 设置于波特率设置; TXCAN, RXCAN 用于连接 CAN 收发器, 与 CAN 总线进行通信; 另 PC0, PC1, PB7 分别连接 CANopen

协议中物理层规定的指示灯的设置和控制电压异常时的线圈断电控制。

3.2.2 节能模块设计

1. 节能方案设计

节能模块是本文设计中极为重要的一个模块，它涉及到现在交流接触器中研究最为热门的节能低碳的概念，如果采用电子技术能够实现节能^{[29][30]}，每年可以为国家节省电量 100 亿度，意义重大。

在此采用了一种带电压反馈智能控制的方式来实现节能，同时为提高电磁铁部分线圈吸合的可靠性，对电源电压的纹波和电压端的干扰进行隔离，避免线圈的误动作。主要分为控制电源输入模块、电压检测模块与 IGBT 驱动模块、中央控制模块，其中滤波、整流电路以及功率电子器件 IGBT 斩波电路组成了控制电源输入模块。节能方案框图见图 3.3。

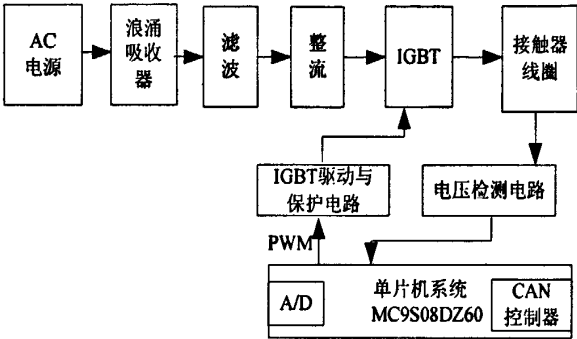


图 3.3 交流接触器节能方案框图

其工作流程是：交流电源经过一系列的滤波整流后，通过开关管供电给接触器线圈，电压检测模块检测线圈电压并送至单片机，经过比较后确定输出电压的占空比，驱动电路则提供开关管的控制脉冲。由于此处采用了电压反馈，使电磁铁的吸力和反力特性达到良好配合，减小了接触器接通过程中铁心和触头闭合时的弹跳^{[31][32]}。当接触器操作电磁铁吸上后，同样检测线圈电压调整 PWM 输出脉冲占空比，使线圈供电电压降低，可大幅度节省能量，降低功耗。

2. 具体节能电路设计

对于交流接触器的电磁铁部分，依用户使用的实际情况而言，输入的是交流电源，如 110V、220V、380V 等，因本文采用的是直流吸持、直流保持的方案，因此先要产生稳定可靠的、纹波系数小的直流脉动电压。交流接触器的电磁系统由交流改为直流，需要对交流电源实现 AC-DC 变换，提供稳定的激磁回路电源和系统供电电源，以满足系统的控制需求，其中激磁回路电源要求纹波系数小。

图 3.4 为控制电源的输入端设计，采用了可靠的电源输入，减小电源线上的传导发射干扰。由 AC1、AC2 从交流电源中取电，先经压敏电阻 R17 进行防浪涌抑

制，再加一个共模扼流圈 T1，可扼制共模噪声，接 C1、R2 进行进一步的滤波措施，进入整流电路，这里采用的是单相全波整流，采用市面上比较流行的 MB6S 整流桥，接 C2、C3 两个不同容值的钽电容，产生所需的直流脉动电压 U_{dc} 。

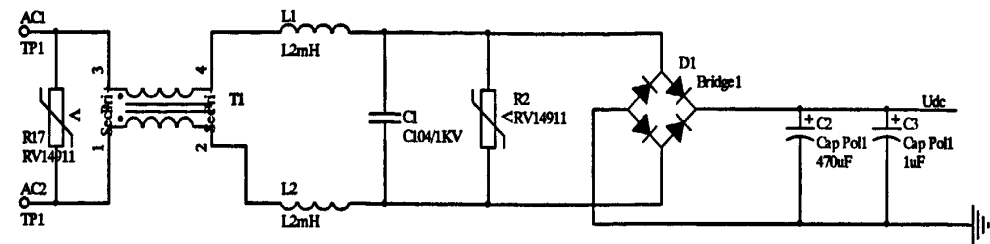


图 3.4 控制电源输入端设计

直流吸合、直流保持的节能及驱动电路见图 3.5。该硬件电路由光耦管、IGBT 场效应管等组成。接插件 J2 处接交流接触器的线圈两端，VD1 为续流二极管，在此将 VD1 反接于线圈两端，能够维持存储在线圈内的磁能，当 IGBT 关断时给线圈能量释放提供通道，并且能够限制由于关断线圈的控制而引起的过电压。在电压不足或者电压突跌的情况下，这样能够保持接触器闭合，在此接触器能够充当电压限制器，使得允许供电不足得到补偿。Q1 是 IGBT 器件，此关键器件选用东芝公司的 GT25Q101，它耐压 1200V，允许通过的最大电流为 25A，具有较高的开关速度， $t_f = 0.5\mu s(\max)$ 。由于接触器吸合过程的电流一般在 1~3A，吸持阶段的电流为 mA 级，考虑到安全裕量的问题做出的选择。利用光耦来实现其驱动，因 GT25Q101 在栅源间施加 7V 左右的电压即可导通。光耦选用 TLP521-1，当单片机端口 PWM 输出高电平时，发光二极管导通，驱动三极管导通从而电压 12V 加于 IGBT 栅源级间，驱动 IGBT 导通。减少寄生振荡是 R12 的作用，D3、D4 是用以吸收关断瞬间产生能量的 TVS 管。

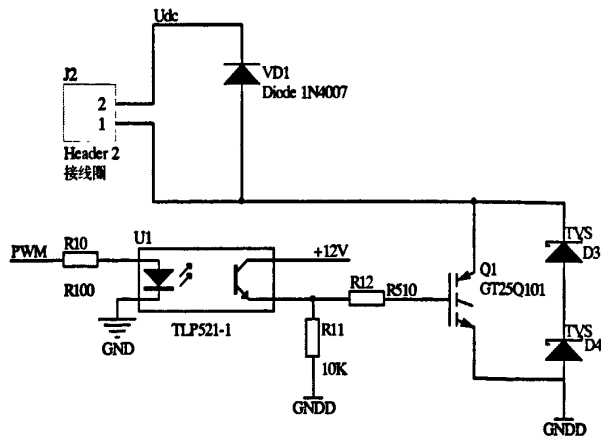


图 3.5 IGBT 驱动及操作电路

3. 控制电源检测电路

电压检测模块是用来检测控制电源的实时电压，通过电压跟随器实现并送至 MCU 模块。在此采样直流脉动电压，得到单片机可以处理的电压信号。经过由运

算放大器 LM358 构成的电压跟随器输入单片机端口 POWER-AD，经 A/D 采样后与阈值电压比较，进行是否进入吸合、衔铁是否开始运动的判断。后期释放接触器时也要检测电源电压，以判断是否需要释放或是出现异常情况。图 3.6 为电压检测模块的方框图。

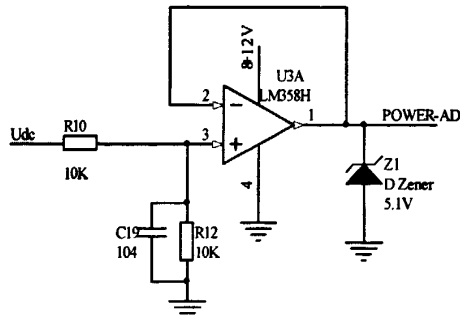


图 3.6 电压检测电路

3.2.3 信号采样与调理电路

本文采用电压互感器对线路电压进行隔离变换，同时设置信号调理电路，使互感器输出的电压信号在滤除干扰后，能变成幅值大小变化适当的电压信号。

智能电器及开关设备运行现场的被测参量，在进入中央控制模块之前，一般需要再经过一次隔离。隔离的目的主要是把模拟通道供电电源与中央控制模块电源分开，使中央控制模块的电源地线“全浮空”，提高中央处理器的抗干扰能力。对于内置 A/D 转换器的 MCU，一般在信号调理电路中采用带有内部隔离器的线性运算放大器即可。

图 3.7 为三相主回路的电压采样电路。为了监测接触器的主回路，对 A、B、C 三相电压进行采样，用于后面与基准电压的比较及上位机监测。要检测的接触器相间电压一般为 220V~660V，而 DZ60 一般只能接受 2.7V~5.5V 的直流数字信号，因此需经过转换。市电交流 380V 经互感器后，经过信号调理电路和运放组成的极性变换电路，转换为目前 A/D 转换器的模拟输入端只接受的正极性电压，再经过一个运放进行隔离放大，采样后的信号将被送至 DZ60 的 PA4-PA7 进行依次采样。

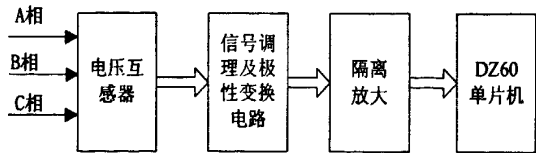


图 3.7 三相主回路电压信号采样及调理电路

3.2.4 CANopen 通信模块设计

采用了内置 CAN 控制器的单片机 DZ60 之后，CANopen 模块的硬件电路简单了许多。本设计采用国外的 AMIS-30660 高速 CAN 收发器，支持高达 1M 的速率，

抗干扰能力强, 允许挂接节点多达 110 个, 在 5kbps 的速率下, 最大传输距离高达 10km。其通信接口电路见图 3.8。

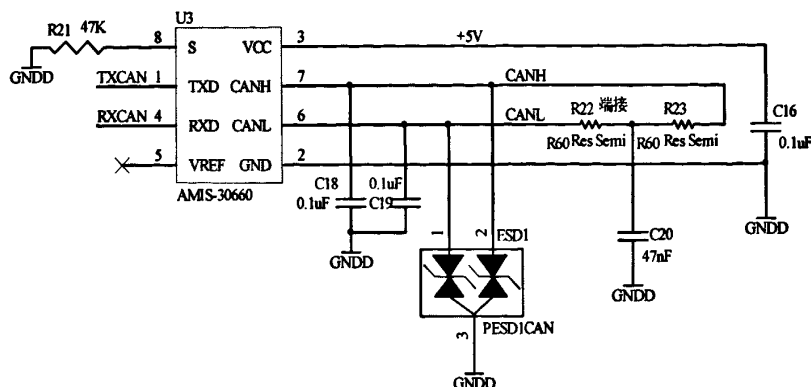


图 3.8 CANopen 通信接口电路

AMIS-30660 芯片的 TXD, RXD 直接接于内嵌 CAN 控制器的 DZ60 引脚 TXCAN, RXCAN。CANH, CANL 通过外围电路后接于 CAN 总线, 随时监测 CAN 总线状态, 准备从总线获取数据和将数据发送于 CAN 总线, 便于上位机监测。CAN 总线数据为免受其他信号的干扰, 增强其可靠性, 必须采取措施进行隔离^{[33][34]}。在电路中, CANH 和 CANL 输出端另接 PESD1CAN 保护器件, 起到 ESD 保护功能, 隔离 CAN 总线和单片机系统。

在 CAN 总线多节点系统中, 会产生阻抗反射和信号衰减的问题, 因此一般要求在最头和最尾的节点中添加终端电阻, 本设计采用 2 个 60R 的电阻并联的方式。

3.2.5 人机接口的设置

作为一个具有通信功能的交流接触器, 其地址需要唯一识别才能监测。节点地址和波特率设置电路见图 3.9。为了达到对多个智能接触器实时监测的目的, 作为下位机从节点, 在本设计中设计了随时可以修改的地址; 为了达到远程高速监控的目的, 基于 CAN 总线的特性, 设置了波特率的可选。

可修改的地址和波特率由编码开关实现, 采用 BCD 码编程。选通引脚分别由 10K 的上拉电阻接于 5V 电压, 电平低导通, 编码引脚通过 weight1、weight2、weight4、weight8 接于单片机 PB0—PB3, NODE_H、NODE_L、BAUD 接于单片机 PB4—PB6, 编程实现地址和波特率的可选值。地址的可选范围为 00-99, 波特率可选范围为 0/1/2/3, 对应 100kbps、500kbps、800kbps、1Mbps。

同时, 基于一般的电路设计习惯, 方便以后维修, 通常都会设置指示灯来表示产品的状态, 在此同时也是依据 CANopen 协议对指示灯的规范要求, 共设置了三个指示灯。设置了一个电源灯, 绿色的贴片二极管; 一个通讯正常灯, 灯的颜色为黄色; 一个通讯故障灯, 红色指示。

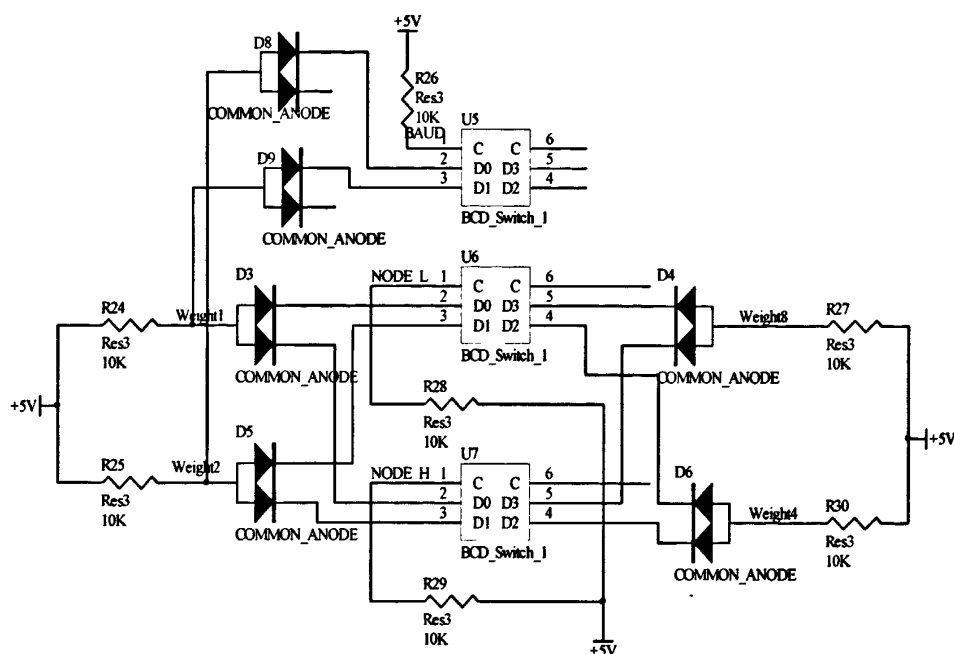


图 3.9 节点地址和波特率设置电路

3.2.6 电源设计

电源电路在单片机系统中是非常重要的一个部分。在本文的设计中，需要 2 组供电电源，分别是前面几个模块中的运放和光耦提到的+12V 直流电、单片机系统中需要的和 CAN 收发器需要的+5V。

根据国家低压电器的相关标准，采用统一的工业标准 DC24V 电源进行供电。CAN 总线接口输入电源 DC24V，通过电源转换电路将其转换为 DC12V 和 DC5V，并提供足够的驱动电流。

电源电路见图 3.10。CANbus 连接器采用 DeviceNet 协议的标准 5 针型插座与 CAN 总线相连，供电线与通信电缆线和设备屏蔽地在同一多芯电缆中传输。在实际操作中，断路器工作场所干扰比较大，特别是浪涌和雷电会对电源的可靠性产生较大影响。在 DC24V 的输入端，采用了 D1 肖特基二极管对电压进行整流和减小纹波的处理，D7 TVS 管加在电源线上，可吸收高达数千瓦的浪涌功率，防止微处理器或单片机因瞬间的脉冲。电源电路的 DC/DC 芯片采用 cosel 公司的开关电源模块，该器件电源转换效率高达 70%，可在 18V—36V 的输入范围下工作，输出电压稳定，可在 105%负荷下工作并可自动恢复，可承受高达 80V 的瞬态电压。

同时再将+12V 直流电通过 LM7805 器件获得+5V 的电压。LM7805 系列为 3 端正稳压电路，内含过流、过热和过载保护电路，带散热片时，输出电流可达 1A。

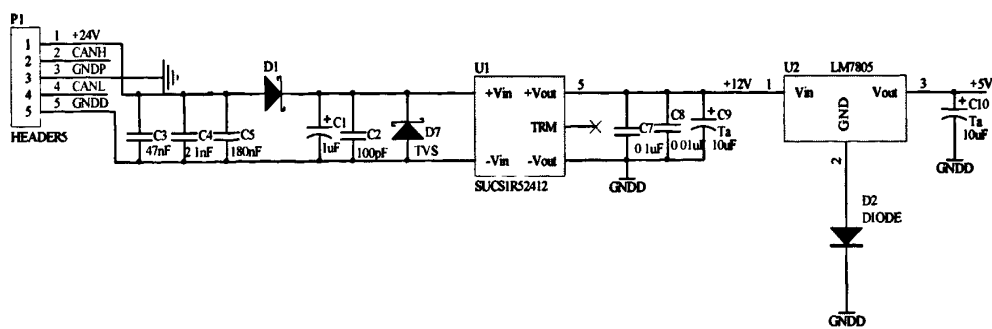


图 3.10 电源模块电路图

3.3 本章小结

本章首先给出了智能接触器的功能要求和具体技术指标。接着设计了智能接触器的整体硬件结构图，并详细阐述了以 MC9S08DZ60 单片机为核心的硬件系统以及各个硬件模块的设计，包括主控电路、节能模块、信号采样与调理模块、CANopen 通信模块、人机接口模块、电源模块等。硬件部分完全是围绕智能接触器的功能特点设计的，为之后的嵌入式软件设计提供了平台。

第 4 章 基于 CANopen 的智能接触器嵌入式软件设计

智能接触器的嵌入式软件设计包括用户应用程序的设计和嵌入式实时操作系统内核的设计。对采用微处理器设计的硬件系统,软件是整个系统的灵魂,软件设计的好坏直接影响到整个系统的性能和稳定性。智能接触器的嵌入式软件设计是本文的重点所在。

4.1 软件设计思想

智能控制单元对接触器进行控制,其实时性要求较高,而且需要处理的信息量较大,因此采用 $\mu\text{C}/\text{OS-II}$ 作为智能接触器的运行平台,所有应用程序均在 $\mu\text{C}/\text{OS-II}$ 的管理下有序运行。而传统的程序设计思路是主循环加中断顺序软件结构,执行效率低下。嵌入式软件虽然也有主程序,但其并不完成某种特定功能,只是进行一些操作系统初始化和实时变量定义工作,同时建立一个任务,启动操作系统,然后把CPU的控制权交给操作系统,永不返回主程序。各用户应用程序(任务)在 $\mu\text{C}/\text{OS-II}$ 系统的调度下合理有序的执行,以实现三相电压信号的实时采样,并对其进行计算、判断等,实现智能控制器的保护功能;同时电磁铁模块也作为任务完成PWM节能功能;信号通过CANopen总线与上位机进行通信,并执行上位机的命令。

1. 软件功能描述

整个软件体系分为两大部分,即智能接触器(通常所称的下位机)的嵌入式软件和PC机(上位机)的数据转存和分析软件。

(1) 嵌入式软件包括如下三项任务:

- ①三相电压的实时采集及数据处理;
- ②PWM节能任务;
- ③CANopen实时双向通信;

(2) PC机软件包括如下四个模块:

- ①电压显示界面;
- ②状态控制与显示界面;
- ③从机ID选择;
- ④异常时分合闸命令;

在研二阶段从事的项目中,基于CAN总线的上位机与下位机都已成功开发,上下位机的数据通讯主要是通过自己预定好的通信规约来实施的。对于上位机来讲,主要是开发控制界面,采用的是VC++,相对来讲容易一些,在这里主要分析

下位机的开发，针对下位机的实时性要求采用实时操作系统及多任务的模式，同时采用比 CAN 协议更加可靠的 CANopen 协议，它能够解析数据包里面的数据，更加方便和实用。同时也会定义好应用在智能接触器上的通信规约，以便后续上位机软件的开发。

2.嵌入式软件的结构

下面介绍本系统嵌入式软件的结构，见图 4.1。在多任务调度开始前，应对 $\mu\text{C}/\text{OS-II}$ 进行初始化，然后建立软件需要的数据处理任务、PWM 节能任务、CANopen 通讯任务。各个任务可以在多任务调度开始后由其他任务建立，也能在多任务调度开始前全部建立，但在多任务开始前一般至少需要建立一个任务。

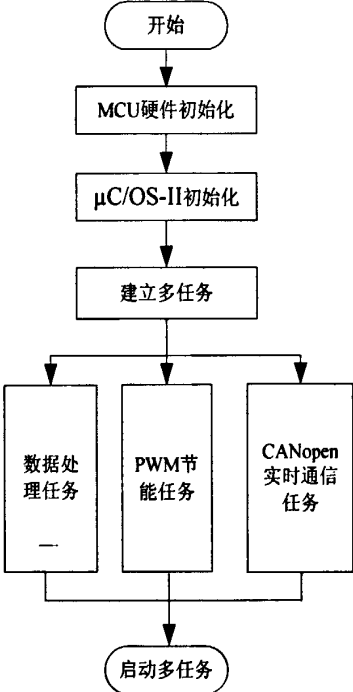


图 4.1 嵌入式软件的结构图

3.嵌入式软件开发思路

本文中下位机嵌入式软件开发采用如下基本思想：以嵌入式开发平台为基础，以任务为主脉络，以实时性为优先级，同时编程时尽量采用模块化思想，便于设计和查错等。

本文中嵌入式软件的开发步骤是：

- (1) $\mu\text{C}/\text{OS-II}$ 移植到 DZ60 上。
- (2)单片机底层配置，中断向量设置，相应的驱动程序。
- (3)在接口函数之上，开发应用程序，构成自己的嵌入式软件。

4.2 $\mu\text{C}/\text{OS-II}$ 在 DZ60 上的移植

4.2.1 移植条件及主要工作

1. 移植条件

本次移植中,使用的是 $\mu\text{C}/\text{OS-II}$ 内核的经典版本 V2.52。所谓移植,即使一个实时内核能在微控制器上稳定运行。绝大部分的 $\mu\text{C}/\text{OS-II}$ 代码采用 C 语言编写,但因在读写 MCU 的寄存器时只能通过汇编实现,仍需要采用汇编语言编写一些与 MCU 相关的代码^{[16][35]}。

要使 $\mu\text{C}/\text{OS-II}$ 在单片机上稳定运行,MCU 须满足如下要求:

- (1) MCU 的 C 编译器可以产生可重入代码;
- (2) 用 C 语言即可以开/关中断;
- (3) MCU 支持中断,且可以生成定时中断(一般在 10~100Hz);
- (4) MCU 可以支持一定数量的数据存储硬件堆栈(一般是几千 byte);
- (5) MCU 有一种指令,可将堆栈指针或其他 CPU 寄存器中的内容读出、存储到堆栈或内存里去^[16]。

很明显,由第 3 章中 DZ60 特点的介绍和 DZ60datasheet 的说明可以看到,这款单片机完全满足以上所有条件。

开发工具包括:

- (1) freescale 单片机的编译调试软件 codewarrior;
- (2) 飞锐泰克的 DZ60 开发板;
- (3) freescale 原装的在线调试、烧写器 P&E USB Multilink Interface;

2. 修改与处理器有关的代码

根据 $\mu\text{C}/\text{OS-II}$ 的要求,移植 $\mu\text{C}/\text{OS-II}$ 到一个新体系结构上,需要编写 2 个或 3 个文件: OS_CPU.H, OS_CPU.C, OS_CPU.A.SYM, 由于 DZ60 支持 C 语言和汇编语言的混合编程,所以可以将 OS_CPU.A.SYM 中的四个函数合并到 OS_CPU.C 中。

(1) 修改 OS_CPU.H

① 不依赖编译器的数据类型

由于不同的微处理器有不同的字长,所以 $\mu\text{C}/\text{OS-II}$ 的移植包含了一系列数据类型定义,以确保其可移植性。尤其是 $\mu\text{C}/\text{OS-II}$ 代码从不使用 C 语言中的 short、int 及 long 等数据类型(与编译器相关),是属于不可移植的^{[17][36]}。

首先为了定义相关的数据类型,应该参考 freescale C 编译器的文档,找到使用的数据类型,使 $\mu\text{C}/\text{OS-II}$ 得到正确的数据类型定义。

② 临界代码的保护

一般每个MCU都会提供一定的汇编指令来开/关中断，C编译器必须有一定的机制直接从C语言中执行这些操作。根据芯片DZ60的特点，采用方法一实现：

```
#define OS_CRITICAL_METHOD 1
//-----
#if OS_CRITICAL_METHOD == 1
#define OS_ENTER_CRITICAL()    __asm sei
#define OS_EXIT_CRITICAL()     __asm cli
#endif
```

③堆栈增长方向

绝大多数微处理器的堆栈是从上往下递减的，DZ60作为HCS08的一员，也不例外。

```
#define OS_STK_GROWTH 1
```

④任务切换的宏定义实现

OS_TASK_SW()是一个宏，是在从低优先级任务切换到高优先级任务时须用到的。在DZ60中，显然的，用软中断指令实现。

```
#define OS_TASK_SW() asm swi
```

(2) 修改OS_CPU_C.C与OS_CPU_A.ASM

OS_CPU_C.C文件中需要定义6个函数，包括OSTaskStkInit()、OSTaskCreateHook()、OSTaskDelHook()、OSTaskStatHook()、OSTaskSwHook()、OSTimeTickHook()。实际需要定义的只有OSTaskStkInit()，其他5个函数需要声明但不一定有实际内容。

OS_CPU_A.ASM中包含4个简单的汇编函数：OSStartHighRdy()：用来使就绪态任务中优先级最高的任务开始运行；OSCtxSw()：实现任务级的任务切换功能；OSIntCtxSw()：在中断级实现任务间的切换；OSTickISR()：时钟节拍中断服务子程序。这5个函数都编写在OS_CPU_C.C文件中，见附录C。

3. μ C/OS-II 在 DZ60 上移植代码的测试

做完 μ C/OS-II在DZ60单片机上的移植后，接着验证移植到DZ60上的 μ C/OS-II操作系统是否能正常工作，移植过程中最复杂的一步工作就是这一步了。为了得到移植成功与否的验证结果，在DZ60的开发环境编写一个简单的测试程序即可。

在调试程序时，用户必须在多任务系统启动以后再启动时钟节拍源计时，不然如果在之前启动时钟节拍源计时，因在主程序中会先开总中断，则有可能在内核启动第一个任务前时钟节拍中断触发，而此时的系统是处在一个不确定的状态，从而会造成系统崩溃。 μ C/OS-II的任务调度是基于优先级的抢占式调度算法，由 μ C/OS-II自己实现，在此只需调用相关函数进行切换即可。

在飞锐泰克的DZ60开发板上，成功的完成了 μ C/OS-II在DZ60上移植的调试。建立了2个任务，分别是驱动灯的闪烁和喇叭的滴答。代码见附录C。

4.2.2 任务划分及任务间通信

1.任务调度实现

基于μC/OS-II 操作系统的应用程序的任务调度图见图 4.2。各用户任务和中断之间通过信号量和消息邮箱进行操作，即应用层和驱动层的通信是通过信号量和消息邮箱来实现的。

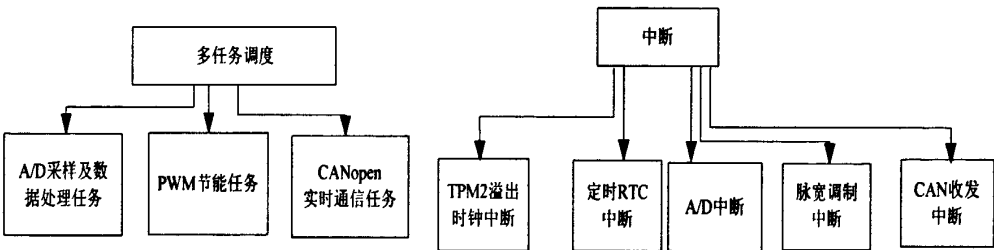


图 4.2 任务调度图

2.任务划分及优先级分配

将本设计中的优先级定为28，系统可用优先级资源为0~25，因要减掉μC/OS-II 操作系统预留空闲任务的优先级27和统计任务的优先级26。考虑到版本的升级和兼容性，一般建议保留优先级0~3，因此实际可用优先级为4~25。从表4.1中也可看出部分优先级暂未使用，方便以后当需要扩充功能时可添加新的任务。

通过分析本设计中的智能接触器所要完成的功能，可将所有任务按照实时性要求划分为实时任务和非实时任务2类。在实时任务中又可划分成强实时的任务与准实时的任务^[36]。当当前任务达到执行时间，则须马上执行，不然会引起错误的结果，引起程序的紊乱，这样的任务即称为强实时任务；准实时任务为当当前任务达到执行时间，能够延迟一小段时间执行，但必须在下一次的任务还没调用前执行完毕，这样的任务即称为准实时任务。经过分析可以将任务分为以下三类，见表4.1：

表 4.1 智能控制器任务功能及其描述

任务号	任务名称	内 容	实时要求	优先级
1	有效值计算	三相电压的有效值计算等	最高	4
2	PWM 节能	控制回路 PWM 节能软件	较高	6
3	CANopen 通信	和上位机、执行机构互相通信	高	7
4	电路自检	CPU 运行状态检测、EEPROM 读写错误检测等	一般	8
5	统计任务	系统任务	较低	26
6	空闲任务	系统任务	最低	27

强实时任务：很明显，在所有任务中，A/D中断是最不能打断的，因此三相电压有效值的计算是最高优先级的任务；

准实时任务：PWM 节能、CANopen 通信任务。这2个任务都是不需要实时性高达A/D采样的程度，但是却是需要及时处理的。对于单片机的执行速度来讲，属于准实时任务。

非实时任务：电路自检任务。

统计任务、空闲任务属于 $\mu\text{C}/\text{OS-II}$ 系统任务，可以通过这2个任务使得系统在没有用户任务的情况下有事可做，并可通过变量使应用程序访问它来了解处理器的利用率。

3.任务间通信与同步

多任务之间相互通信与同步用以协调各自的操作和工作的主要手段即任务间的通信机制。通常采用如下 2 种方法来协调任务间的通信事务：一种是共享的方式，采用信号量进行共享资源的使用和释放；另一种是通信的方式，采用消息邮箱、消息队列等传递下一个任务需要的消息^[17]。

任务间的通信包括各用户任务之间的通信和用户任务与中断之间的通信。

中断是一种硬件机制，实时系统设计中最关键的问题即硬件中断处理问题。DZ60 中的中断优先级已经设置，共有 32 个优先级可用。本文中共设计了 5 个中断向量：一个是 A/D 定时采样中断，其优先级最高。因为实时可靠的采样是计算有效值、功能保护等功能实现的基础；一个是系统时钟节拍中断，其优先级次之。每 10ms 中断一次，作为系统时钟；一个是定时/计数中断，用于各个模块的定时、延时、计数等的使用；一个是 PWM 输出脉冲中断，用于输出节能模块的控制脉冲，产生斩波信号供给线圈；一个是 CANopen 通信中断，其优先级低。 $\mu\text{C}/\text{OS-II}$ 系统的中断处理允许中断嵌套。

为了实现任务间的同步，本软件系统建立了 4 个事件，具体如下：

(1)数据计算信号量 OSADCntSem。该信号量用于任务 1 和 A/D 定时采样中断子程序进行通信。每完成一次中断采样便发出这个信号量，通知任务 1 对 A/D 采样的数据进行计算处理。当电压有效值计算后存储于 MCU，然后由功能保护模块调用此有效值数据进行判断，看此时的线路是否出现过压欠压状态，如有，则断开接触器，以免烧毁。

(2)通信数据消息邮箱 OSTPDOMbox。该消息邮箱用于任务 1 和任务 3 之间的通信，将三相电压值送给 CANopen 通信模块，输送到上位机进行监测。

(3)通信信号量 OSReceiveSem。该信号量用于任务 3 和 CANopen 驱动层的中断子程序进行通信。在适当的状态下一旦接收到上位机发过来的信号，中断子程序接受数据，并发出这个信号量，使任务 3 对接收数据进行处理。

(4)节能任务信号量 OSPWMSem。该信号量用于任务 2 和 PWM 中断程序之间的通信。任务 3 处理的信号都是通过 PWM 中断程序输出并提供给开关管的。

4.3 DZ60 初始化设置

一般的单片机软件开发分层进行,各个模块开发完成后再直接调用函数即可。同样,单片机的通用底层软件也需先编写好,然后全部放在 HardwareInit()中,供主函数调用。

4.3.1 单片机底层初始化及中断设置

1. 单片机基本底层功能初始化

DZ60 支持外部和内部时钟,支持 1~16M 的总线频率,在此采用外部时钟,用 16M 的外部晶振实现,实现总线时钟 8M,编程如下:·····

```
#define CPU_CONFIG()
{
    SOPT1 = SOPT1_VAL;
    SOPT2 = SOPT2_VAL;
}
```

通用 IO 端口供编码开关和灯的设置,编码开关的 IO 口数据方向统一设置为低电平,读入;灯的 IO 口数据方向统一设置为高电平,输出。

RTC 定时器在本文中起着极为重要的作用,它主要用于中断采样的定时采样时间间隔的设置、PWM 节能模块的延时与定时设置、CANopen 心跳报文的定时时间间隔设置、TPDO 的定时触发设置。在程序中先设置计数器为最小的采样定时 0.625ms,然后取 0.625ms 的 N 倍采用多个计数器来分别独立的实现各定时功能。

μC/OS-II 时钟需要单独设置,在此采用 TPM2 作为 μC/OS-II 的时钟,设置好 TPM2SC,设置模数寄存器(TPMxMODH: TPMxMODL),这里采用预分频 128,模数寄存器为 TPMxMODH: TPMxMODL = 0x04:0xE2,定时 μC/OS-II 时钟为 10ms。

PWM 主要用于节能模块,输出占空比不同的脉冲波。这里采用边缘对齐 PWM 模式,这类 PWM 输出使用定时器计数器的正常向上计数模式(CPWMS=0),PWM 信号的周期由模数寄存器(TPMxMODH: TPMxMODL)的值加 1 确定,占空比由定时器通道寄存器(TPMxCnVH: TPMxCnVL)中的设置确定。采用 TPM1CH0 通道进行脉冲波输出。

A/D 转换器由 2 个寄存器控制:一个是 ADCCFG 寄存器,该寄存器必须在 A/D 转换前写入值,用于设置 ADC 的运行模式、时钟源、长采样时间。一个是 ADCSC1 寄存器,该寄存器包含 ADC 的 AIEN 位、COCO 位和输入通道选择。在进行 AD 转换是,软件通过检查 ADDR 寄存器的 COCO 位是否为 1 来判断 ADC 转换是否结束(若设置 ADC 中断并置位中断位 AIEN,则 ADC 转换结束后会产生中断);当转换完成

后,从ADCRH: ADCRL寄存器中读取结果。

2. 人机接口的设置

本文采用的是台湾的十进制编码开关设置从站地址和可选波特率,这样可以减小线路板的尺寸,也方便操作。由第二章可知,地址由 2 个编码开关进行设置,范围为 00-99,此编码开关为 BCD 编码,且低有效,因此程序相对简单, ID 设置函数 ReadCodeswitch()再被各个函数调用。同时采用三个二极管进行指示标识,一个电源灯接+5V,其他 2 个红色灯、绿色灯分别采用 CANopen 协议中规定的闪烁模式进行标识。在编程时,只需控制好标识位、高低电平,用定时器进行时间闪烁的控制即可实现。

3. 中断设置

DZ60 设置了总共 32 个中断供开发人员使用,优先级别也设置好了,当要进入中断时,只需将相应中断寄存器的使能位设置为 1,在中断函数里面再将中断标志位清 0 即可。但是要注意的是,中断函数最好不要嵌套,以免出现程序崩溃。

在这里使用了 ADC、CANRX、CANTX、TPM1 的通道 0、TPM2、RTC 共 6 个中断,供不同的模块使用,以完成用户任务。

4.3.2 CAN 模块初始化

1. MSCAN 模块的初始化

CAN 模块的初始化是初始化配置中比较重要的一块。

一帧数据帧由 11 位 ID 和数据域构成,其中 11 位 ID 中的 10-7 位为各节点的通信对象的功能码,预定义好,6-0 位为各节点的实际 ID,由第三章中的编码开关进行编码,00-99,数据域中的数据用于存放应用对象等具有实际意义的数

CAN 控制器初始化主要是用来实现 CAN 工作时的参数设置,其初始化内容包括:硬件使能 CAN、波特率设置、设置 CAN 验收滤波器的工作方式(与 ID 有关)、设置 CAN 控制器的工作模式等。

首先必须先定义 CAN 报文,结构体如下所示:

```

/*****
*
*                               CAN 报文类型定义
*****
typedef struct _can_msg
{
    INT16U id;                /* 11 位的 CAN 报文标识符 */
    BOOL RTR;
    INT8U data[8];
    INT8U len;
    INT8U prty;
} can_msg;

```

使能 CAN 模块,只需编写好相应的寄存器即可,显然采用中断收发,因此主

要是收发中断使能的控制。采用函数 void CANEnable(bool RxEnable, bool TxEnable)实现。波特率设置，配置好总线计时寄存器即可。

```

/*****
*
*                               波特率参数选择
* 说明: bus clock:8M  CAN 总线波特率: 10kbps--1Mbps
* *****/
tCANBitClkParms CANBitClkSettings[]=
{
    {5, 2, 2, 1},          /* CANBAUD_1M    0 */
    {5, 2, 2, 2},          /* CANBAUD_500K  1 */
    {11, 4, 2, 2},         /* CANBAUD_250K  2 */
    {11, 4, 2, 5},         /* CANBAUD_100k  3 */
    {11, 4, 2, 4},         /* CANBAUD_125K  4 */
    {11, 4, 2, 10},        /* CANBAUD_50k   5 */
    {11, 4, 2, 20},        /* CANBAUD_25k   6 */
    {11, 4, 2, 25},        /* CANBAUD_20k   7 */
    {11, 4, 2, 50},        /* CANBAUD_10k   8 */
};

```

CAN 验收滤波器在 CAN 帧的接收中是非常重要的一块。主要是 DZ60 内嵌 CAN 控制器，所以它自己本身有着非常独特的接收数据帧的模式，主要采用三个寄存器进行设置：CANIDAC、CANIDMR0-7、CANIDAR0-7。

```

/*****
*
*                               验收滤波器设置
* *****/
// 接收滤波器设置 CANIDAC、CANIDMR、CANIDAR
CANIDAC = 0x00;          /* two 32bit acceptance */
//下面仅配置为接受标准帧格式
CANIDMR0 = 0x00;
CANIDMR1 = 0x18;          /* 屏蔽 RTR, IDE 为 0, 标准帧 */
CANIDMR2 = 0xFF;
CANIDMR3 = 0xFF;
CANIDMR4 = 0x00;
CANIDMR5 = 0x18;          /* 屏蔽 RTR, IDE 为 0, 标准帧 */
CANIDMR6 = 0xFF;
CANIDMR7 = 0xFF;
//配置节点地址
CANIDAR0=(unsigned char)(InitID>>3); /* 高 8 位 */
CANIDAR1=(unsigned char)(InitID<<5); /* 低 3 位 */
CANIDAR2 =0x00;
CANIDAR3 =0x00;
CANIDAR4 =(unsigned char)(InitID>>3);
CANIDAR5 =(unsigned char)(InitID<<5);
CANIDAR6 =0x00;
CANIDAR7 =0x00;

```

2. CAN 模块驱动层设计

CAN 驱动层设计是实现 CANopen 协议设计的基础。CAN 驱动层的设计与所使用的 CAN 控制器密切相关，本文采用的飞思卡尔的 8 位单片机 MC9S08DZ60 是一款带 CAN 控制器的芯片，因此驱动层设计相对简单，下面介绍基于该芯片的 CAN

底层驱动的实现。

DZ60 的 MSCAN 模块共有 16 个可写入的寄存器，同时还有发送报文缓冲器和接收报文缓冲器。发送报文缓冲器采用三重发送缓冲器机制，本地优先方式发送数据帧；接收报文缓冲器有 5 级输入 FIFO，配合三组接收的滤波寄存器。报文缓冲器结构共包含 16 个字节，前 13 个寄存器中，前 4 个是数据帧的标识符 ID，后 8 个是数据域的具体数据寄存器和 1 个用来存储数据长度的寄存器。通过设置 TBPR(数据发送缓冲器优先级设置寄存器)，可以确定该数送缓冲器的优先级^[28]。

对于主控制器来说，CAN 控制器是基于事件触发的，因此可以采用中断或是查询的控制方式，但采用查询方式执行效率低下。所以本设计中 CAN 驱动层采用中断控制方式，CAN 驱动层发送/接收流程图见图 4.3。

初始化好 CAN 模块之后，中断接收和发送是很容易编程实现的。只需将相应的 ID、实际数据分别填入 CAN 数据帧中即可，或是直接判断结构体 msg_receive 中的数据和标志位。函数分别为：INT8U CANsnd1DFrm(INT16U rid,INT8U *databuf,INT8U len)与 INT8U CANrcv1Frm(INT8U *CANRcvbuf)。

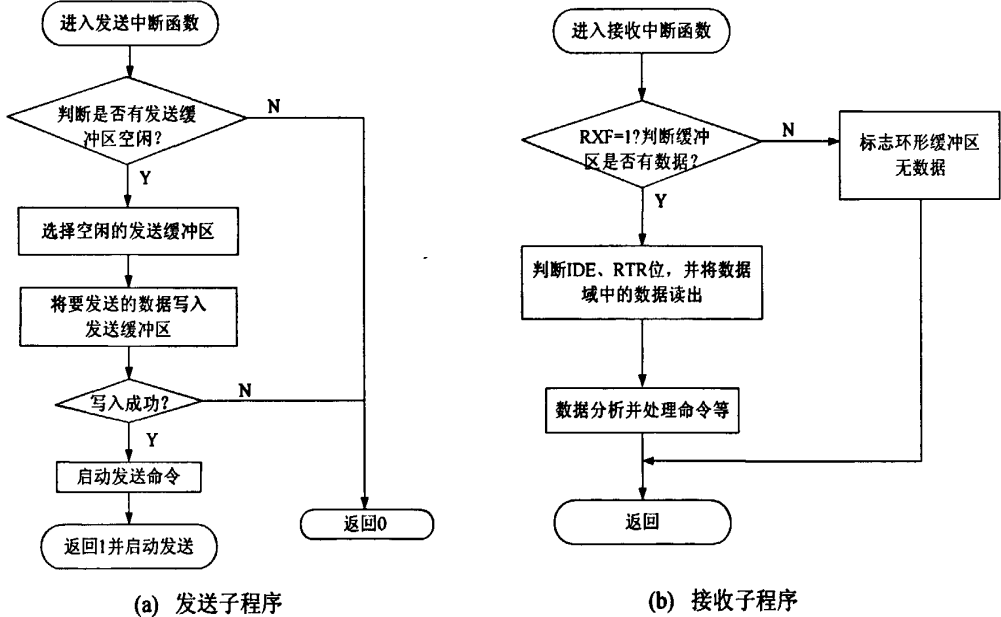


图 4.3 CAN 驱动层中断接收/发送流程图

4.4 用户应用程序设计

4.4.1 PWM 节能任务

节能接触器的电流规格一般比较为 63A-800A，操作频率不像小规格的交流接触器频繁(如使用在 AC-3 类别的 TGC5-63~250，当其控制电源电压为 220V 的时候，

要求其极限操作频率为 1200 次/h, 即 3 秒/次), 因此实现节能意义重大, 适应国家低碳的环保概念。PWM 节能任务主要实现大电压吸合、小电压吸持的功能。

1. 节能模块软件设计

PWM 节能流程图见图 4.4。原理如下:

首先, 线圈的控制电源是由用户设定的, 可以是 110V、220V、380V 等。国标 GB 14048.4-2003 有规定, 接触器必须在额定控制电源电压的 85%~110%可靠吸合。另外企业标准里增加接触器处于断开状态时, 其输入的额定控制电源电压 U_s 在 $<70\%U_{smin}$ 或 $>115\%U_{smax}$ 范围内, 接触器不应动作。因此, 在实现 PWM 节能的第一步是先检测整流后的控制电源的电压, 判断其是否大于门槛电压, 进行吸合判断。根据第 2 章中动力学的分析, 门槛电压的设置由交流接触器的试验数据得到, 且控制电源电压不同, 门槛不同。控制电源的电压由 POWER-AD 进行交流离散采样, 然后计算有效值。

第二, 当发现可以吸合之后, 铁心开始运动。由单片机确定吸合阶段的占空比, 此占空比由试验数据得到, 试验数据是先前存入单片机中满足优化的吸力与反力特性配合曲线的数据^{[37][38]}, 通过 PWM 控制口输出脉冲, 然后调制整流后的控制电源的电压, 采用 PWM 技术对开关管实行关断产生脉宽变化的斩波信号, 提供给反并联续流二极管的线圈, 产生线圈的吸合电压。

第三, 一般来讲, 在静磁芯产生的磁力吸引下, 铁心会克服反力弹簧的反力运动, 直至两者闭合, 但是由于闭合之后的速度并未马上减为 0, 因此一般会出现铁心和主触头的弹跳, 弹跳次数较多从而导致电弧对触头的侵蚀增大, 对接触器的电气寿命造成很大影响, 这也是本文采用电压反馈的原因之一。因此, 在闭合之后, 采用单片机对其进行延时等待, 减少弹跳, 再转入吸持状态。

第四, 吸持状态。延时一段时间之后, 对线圈的实时电压进行取样, 送入单片机, 经过比较, 确定吸持阶段的脉冲占空比, 同样对整流后的控制电源的电压进行调制, 产生斩波信号提供给线圈, 产生合适的吸持电压, 并定时保持, 等待释放。单片机在维持这种状态的同时, 对吸持电压及电网不平衡度进行监视, 一旦有短路、断相、电网电压不平衡度超过 30% 或者电网电压低于 160V 及起动运行电流超过设定值, 控制回路立刻输出低电平进行失电分断, 从而保护设备的安全。

吸持电压是在考虑接触器在其安装场所应能承受一定的振动和冲击的能力以及相关的辅助触头数量前提下, 而计算出的恰好能够保持接触器闭合的电压值。

第五, 当定时时间到, 检测控制电压, 若小于释放电压, 则判断是否需要关闭 IGBT, 是的话, 则输出低电平让其关断, 线圈断电, 释放衔铁, 断开接触器; 如果需要进行其他处理的话, 可以进行延时, 等待其他的处理。

这样的节能方案设计的节能接触器具有节能、无声、省材、体积小等优点。

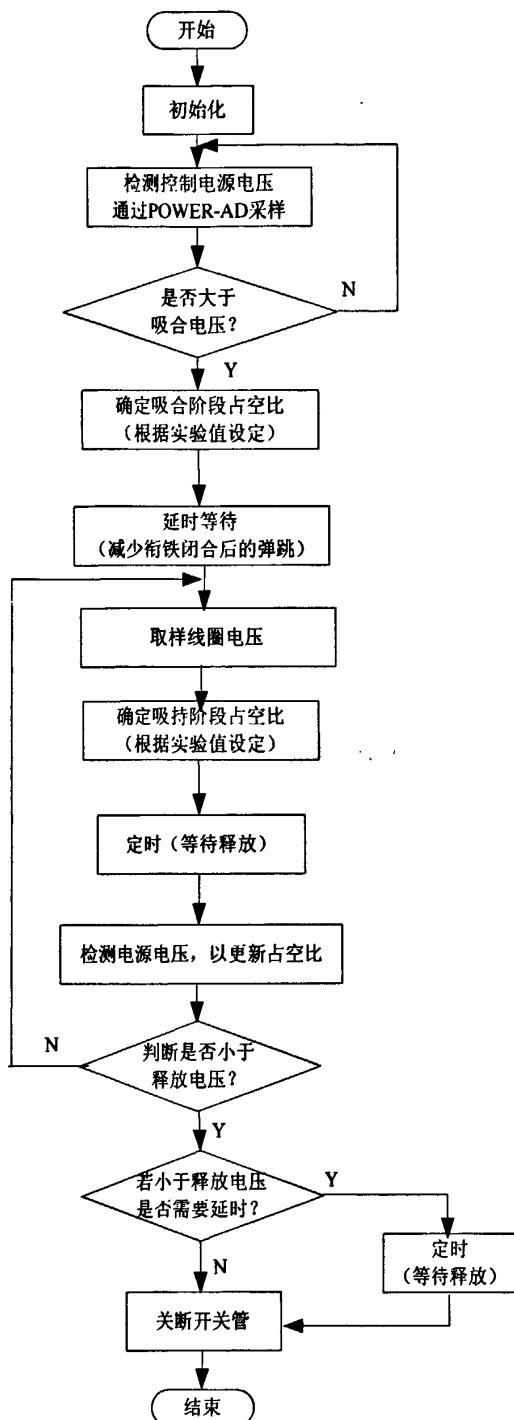


图 4.4 PWM 调节能任务的软件流程图

2. 节能模块软件设计中的注意事项

在节能设计中需要注意的是：

(1)接触器的吸上和释放电压需要根据国家标准规定，在软件中设定。接触器控制端得电后，若检测得到电压大于规定的吸上电压，则确定占空比，输出控制脉冲，使 IGBT 导通；当接触器处于吸上位置，若检测到电源电压低于规定的释放电压，则停止输出脉冲，封锁 IGBT 使接触器线圈断电。

(2)由于本文吸上与释放电压根据设定值由开通或封锁 IGBT 来实现,没有了传统接触器的抖动现象,但是为了提高接触器对瞬时电压骤降的承受能力,当电源电压下降至释放电压时,单片机可设置一个可调延时,经过延时后再停止控制脉冲的输出^{[39][40][41]},从而提高接触器的抗干扰能力,提高运行可靠度。

4.4.2 A/D 采样和数据处理任务

目前在中、低压等级电网中,6 次以上的谐波不会造成对保护精度的任何影响,所以截止频率至少可设为 6 倍的基波频率 $300\text{Hz}(6 \times 50\text{Hz} = 300\text{Hz})$ 。由 Nyquist 采样定理 $f_{s,\max} \geq 2f_{\max}$ 可知,模数转换器的采样频率应至少为 2 倍的截止频率,即 600Hz ,此采样频率完全能够满足在中、低电压等级电网中的测量精度。

本设计每个电源半波周期(在智能电器的监控中,为简化处理,一般采用 50Hz 电源的周期 T 来计算采样周期 T_s ,即 $T_s = 10\text{ms}/N$)采样 16 个点,采样频率为 1600Hz ,定时时间为 0.625ms 。用单片机定时中断对输入的电压信号进行采样,每次定时中断都依次对 A、B、C 三路电压进行采样。系统时钟为 8M ,定时器采用 8 分频,为 1M 。定时器每 $1\mu\text{s}$ 计一个数,要实现定时 0.625ms ,就在模数寄存器 RTCMOD 中填入 $0x0271$ ($1\mu\text{s} \times 625 = 0.625\text{ms}$)。当 16 次采样完成时(一个周期),通过调用函数 OSSemPost()将信号量 OSADCntSem 置 1,用以通知数据处理任务进行操作。

智能电器对电参量的测量是根据对被测参量的采样结果,实时的计算出被测量的当前值,然后才能进行电参量的保护和判断等。在本设计中,主要是电压有效值计算。MCU 是根据被测量的采样值进行计算的。求有效值,即计算其均方根值,离散的有效值计算表达式为式 4.1:

$$U = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} u_k^2} \quad (4.1)$$

注: N : 每个电源周期的采样点数; u_k : 电压在第 k 点的采样值。

4.4.3 CANopen 实时通信任务

图 4.5 为 CANopen 从节点总体设计流程图。在 CANopen 协议栈中,共有 4 类通信对象,且有 4 种运行状态,各个通信对象都需要在一定的状态下运行和操作。在本文中,在初始化完成后,开总中断,进入各运行状态。传统的软件设计中,在主循环中一般是以各状态作为模块,各个通信对象在相应的状态下完成相应操作,然后通过各个状态的转换完成协议栈的设计。本文并没有完全以各运行状态为模块进行设计,而是以各个通信对象的操作为模块,将各个通信对象执行所需的各个状态作为其使能条件,大大简化代码,并满足本文 CANopen 协议栈的设计要求。

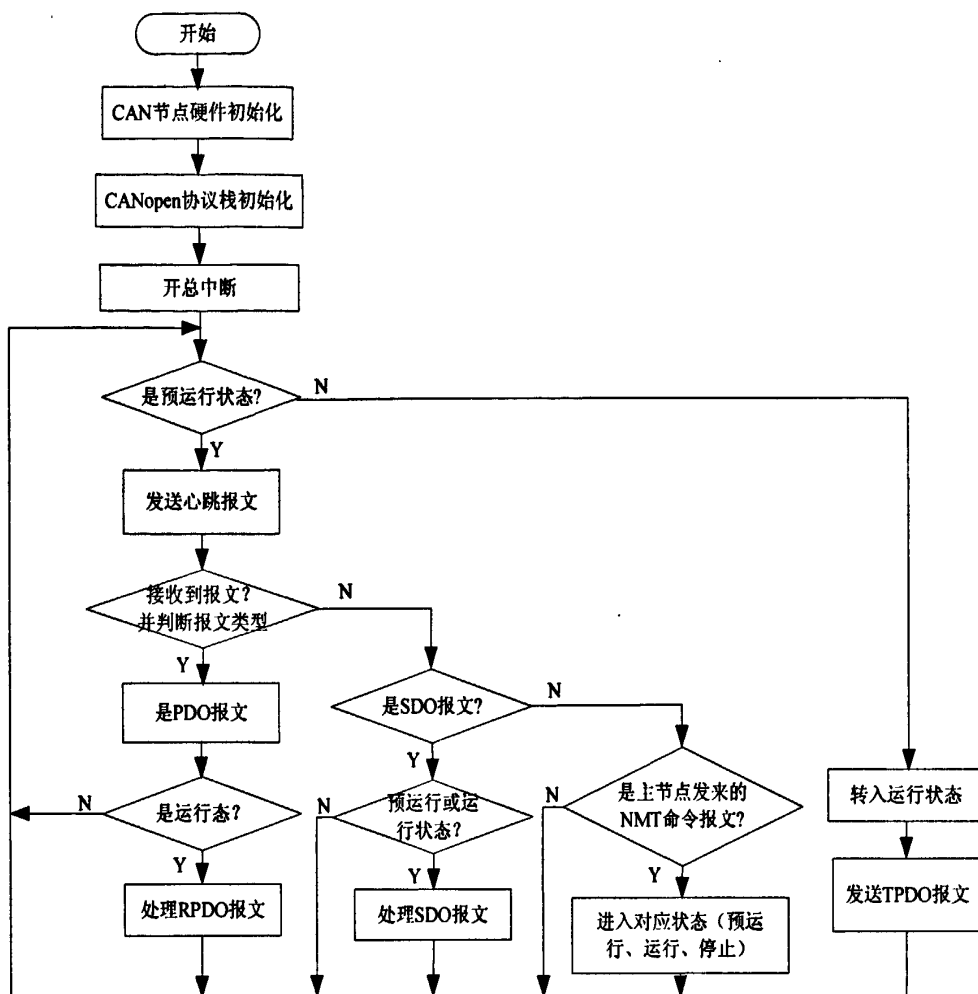


图 4.5 CANopen 通信任务总体设计流程图

4.5 CANopen 通信任务的软件设计

上面简单介绍了 CANopen 任务的总体流程图，因为 CANopen 从节点的软件设计比较复杂，所以将其单独列为一节，在此详细叙述。

4.5.1 CANopen 协议栈的设计

CANopen 设备完成初始化之后，进入了能够由用户来控制其执行的实际运行状态，以各个状态为使能条件，在主循环中处理各个通信对象的操作函数，以处理各种命令和数据，完成相应功能，见图 4.6。

在进入到预运行状态之后，可以经由主节点发送各种 CANopen 报文给其它从节点，以发送各种数据或是命令，从节点接收到属于自己的 CANopen 报文后，首先判断目前所处状态和报文类型，判断完成后进入相应的报文子函数进行通信数据的处理，完成 CANopen 主从站的实时双向通信。

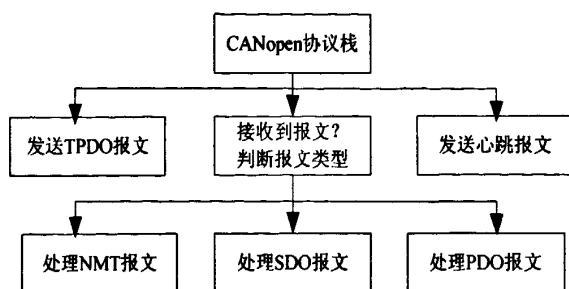


图 4.6 CANopen 处理示意图

4.5.2 处理接收的报文

1. 收到的是 PDO 报文

本协议栈使用了一个RPDO报文，用于分合闸指令的下达。当从节点接收到报文时，判断报文类型，若收到的是PDO报文，且此时的节点处于运行态，则根据传输的类型(254 或 255，异步)和标识符ID找到符合的RPDO，并找到相应的参数，再根据通讯参数解析出相应的对象数据后，将接收的数据复制到对应RPDO的过程映射数组中，激发收到的数据，对设备进行命令控制。

由于PDO通讯的实时性较高，所以当节点进入操作状态后，各节点主要采用PDO进行信息的交流和通信。

2. 收到的是 SDO 报文

当从节点接收到报文时，判断报文类型，若收到的报文是SDO报文，且节点目前处于预运行或者运行状态，允许SDO通信，进入相应的SDO报文处理函数。

从节点SDO报文处理的流程见图 4.7，在收到SDO报文请求后，获取报文的命令信息、索引、子索引(索引和子索引在对象字典中已规定好)，然后解析命令字，判断命令的具体信息，若为上传(读)或下载(写)命令，在对象的读写属性满足命令信息(命令为读或写，对象可读或可写)的条件下，则依据索引以及子索引找寻OD中相应的对象。若是读命令，则发送SDO响应报文给请求的主节点，SDO响应报文里面是赋给的对象的价值；若是写命令，则将SDO报文中的数据域中内容赋给相应的应用对象，同时发送响应报文给主节点。

SDO报文主要是用来对对象字典进行访问，它主要用于在设备过程中传递参数及大数据块。由CANopen协议知，SDO的传送机制有3种，即加速传送、分段传送和块传送。在本文设计中，由于智能接触器系统要求传输的数据量不是很大，但实时性要求相对会高一些，前两种传输方式就可以满足功能要求，块传送则不被支持。

SDO报文的具体传输模式在CANopen协议中总共有5种类型，包括：启动域下载；域分段下载；启动域上传；域分段上传；域传送中止。对它们的区分在于对SDO命令字(报文的数据部分的第一个字节)的解析，见表4.2，SDO通讯分为下

载和上传，下载是指主节点写从节点的对象字典，而上传是指主节点读从节点的对象字典。SDO 命令字中 bit5~bit7 这 3 位是固定的命令字，用其可区分上述 5 类 SDO 报文。

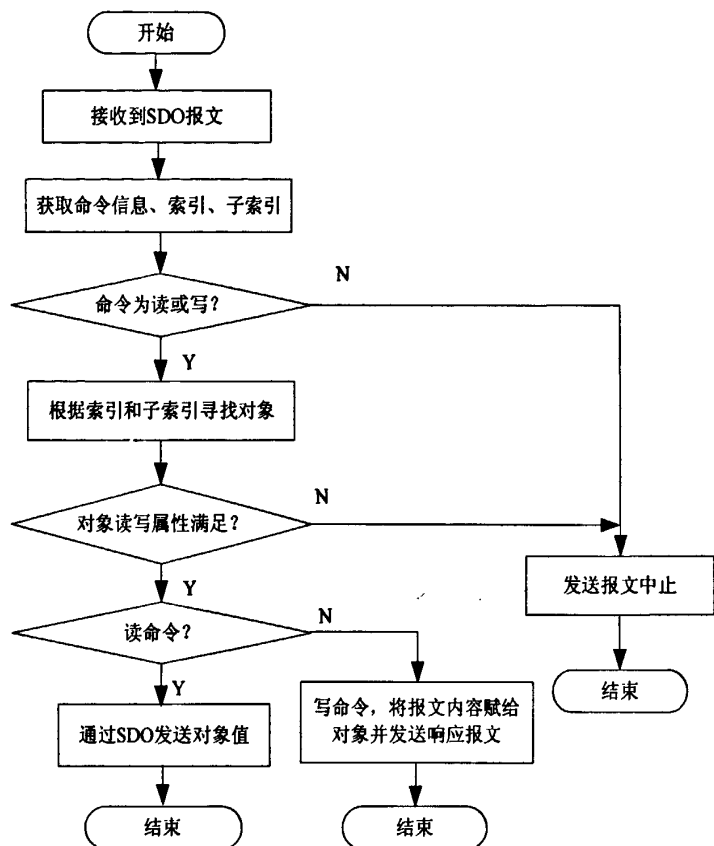


图 4.7 处理 SDO 报文流程图

表 4.2 SDO 对象数据字节的使用

字节 0			字节 1-2	字节 3	字节 4-7
SDO 命令字 (bit5-bit7)	传送方式	功能	Client->Server (请求)	Server->Client (应答)	
	启动域	下载	100	110	对象索引 数据值
		上传	010	010	
	域分段	下载	000	100	
		上传	110	000	
	域传送中止		001		

3. 收到的是 NMT 报文

当从节点接收到报文时，判断报文类型，若收到的报文为 NMT 主节点发送的控制报文，则从节点根据 COB-ID 判断并读取相应的数据帧中的数据，根据表 4.3 中的字节 0 中的命令字使设备进入相应的状态(复位节点、复位通信、预运行、运行、停止)。如果设备是进入到了运行态，可马上生成 TPDO 报文，准备随时发送出去。

对于一个连入 CANopen 网络的从节点，必须支持并使用 NMT 服务，且从节点

的一个很重要的功能即可以解析其接收到的NMT报文，然后根据解析得到的NMT命令字来改变从节点的目前状态，使其进入下一个相应的状态进行报文操作。NMT从节点处理流程图见图4.8。具体的NMT对象数据字节使用及命令字的意义见表4.3。

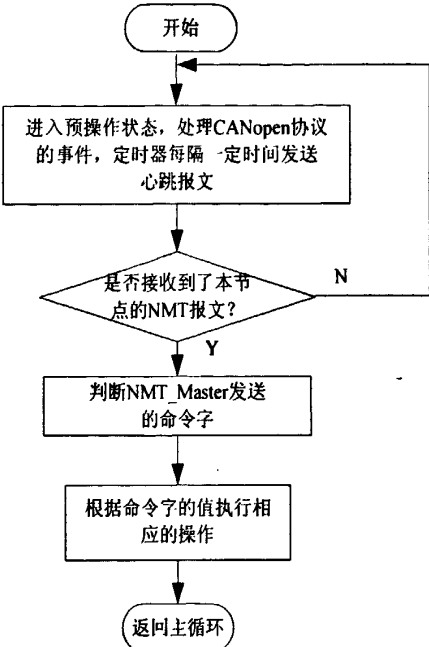


图 4.8 NMT 从节点处理流程图

表 4.3 NMT 对象数据字节的使用

标识符	字节 0(命令字)	字节 1
0x000	命令字	表示的意义
	1	运行态
	2	停止态
	128	预操作状态
	129	初始化（复位节点）状态
	130	初始化（复位通信）状态
		从节点 ID

4.5.3 发送 TPDO 报文

CANopen 协议中定义了 PDO 的传输类型，传输类型反映了 CANopen 设备何时可以发送数据。PDO 传输条件见表 4.4。TPDO 报文有 2 种发送方式-同步和异步，同步发送要求在同步中断子程序中执行；异步发送由事件触发，事件触发可能有两种，远程请求触发和对象字典项数据的改变。在 CANopen 协议栈中进行的是异步发送。对 TPDO 的发送判断需要循环进行。

一个 PDO 能够指定一个禁止时间(InhibitTime)，禁止时间是指两个连续的 PDO 报文传输时最小的间隔时间，可以避免高优先级数据的数据量太大，会一直占据总线，使得另外的优先级较低的数据竞争不到总线的问题。禁止时间

(InhibitTime)定义成 16 位无符号整数，其单位是 100us。一个PDO 能够指定一个事件定时周期(EventTime)，当超过这个定时时间，PDO 报文的传输能够直接被触发而不需要触发位。事件定时周期(EventTime) 定义成 16 位无符号整数，其单位是 1ms^[19]。

表 4.4 PDO 传输条件

触发PDO的条件					PDO 传输
传输类型	B: 两者须同时能够成立 O: 只有一个能够成立就可以触发			事件 Event	
	收到SYNC 消息	收到RTR 消息			
0	B	—	B	同步，非周期	
1~240	O	—	—	同步，周期	
241~251	—	—	—	保留	
252	B	B	—	同步，收到 RTR 之后发送	
253	—	O	—	异步，收到 RTR 之后发送	
254	—	O	O	异步，制造商特定事件	
255	—	O	O	异步，设备的子协议特定事件	

在本设计中，因为PDO 只在运行态下可用，需要先判断目前节点所处状态，保证节点是处于运行状态下，且此TPDO 报文传输类型为 254 或 255(异步传输)。

在本设计中，TPDO 报文根据第 2 章的描述生成。报文生成示意图见图 4.9。如果定义事件定时周期且事件定时周期超过，则将过程映射数组中的数据复制到TPDO 并启动发送。事件定时周期触发表示TPDO 的发送是周期性的，定时周期设为 500ms。则每隔 500ms，TPDO 会自动发送。本设计中的发送PDO 根据映射参数和通讯参数的COB-ID 来封装报文，在大于禁止时间和小于事件定时周期之内将TPDO 报文发送出去，例如：接触器从站节点把当前的三相电压发给主站。

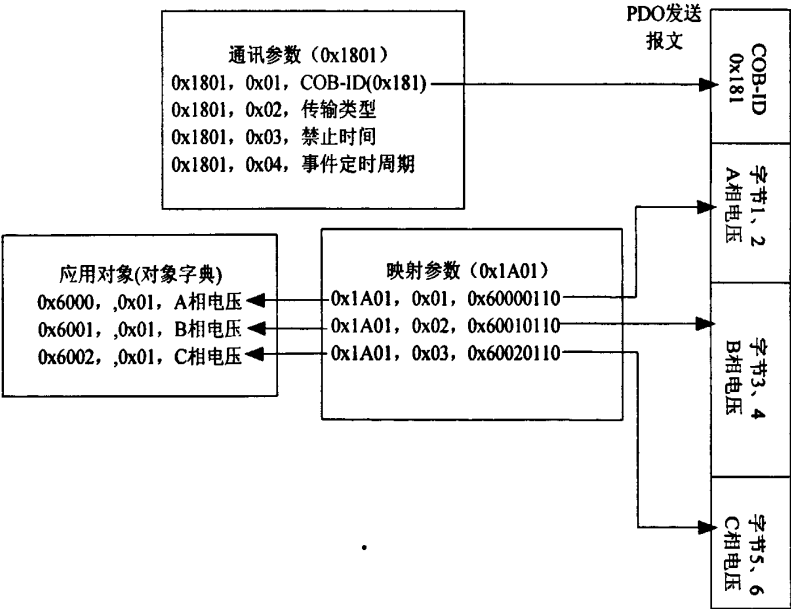


图 4.9 PDO 报文的生成图

4.5.4 发送心跳报文

由 CiA 推荐, 本文使用心跳报文作为错误控制服务。在从节点中判断心跳时间计数器是否超过心跳周期, 超过则发送心跳报文, 同时心跳时间计数器清零。

CANopen 从节点定时发送心跳报文给主节点, 由主节点完成各个从站节点的监控。对于 CANopen 网络中每一个采用心跳报文机制的从站节点, 当 CANopen 协议栈开始执行时, CANopen 主站会在启动过程中启动该从节点的心跳报文机制, 同时设置心跳报文监视函数。节点的心跳周期即为报文监视函数的等待超时时间, 在本设计中, 设置心跳周期为 500ms, 以便主站检查其状况。报文监视函数每隔一定时间检查一次等待的超时时间是否到, 在不出现异常的情况下, 在等待超时的前面, CANopen 主站应会收到相应节点发送的心跳报文。CANopen 主站的报文监视函数解析该心跳报文, 通过解析后的数据来更新 OD 中地址为 1F82h 上对应的子索引, 并会重新设置超时时间。若出现异常情况, 超过了一个心跳周期后, CANopen 主站仍没有收到相应从站的心跳报文, 该监视函数会生成一个超时错误告诉主循环进行错误处理。表 4.5 是心跳报文的数据字节的具体使用的标注及各状态相应的数值。除了 Boot-up 报文外, 在设备正常运行期间, 从节点将会处于预运行、运行和停止三个状态中的某一个。

表 4.5 心跳报文数据字节的使用

标识符	字节 0 (从节点状态)	
	数值	表示的状态
0x700+Node_ID	0	启动(Boot_up)
	4	停止(Stopped)
	5	运行(Operational)
	127	预运行(Pre_operational)

4.6 本章小结

本章论述了智能接触器的嵌入式软件的详细设计过程。

在嵌入式平台μC/OS-II操作系统的基础上, 结合硬件电路, 开发了智能接触器的软件系统。首先介绍了μC/OS-II在DZ60上的详细移植过程, 然后对各个任务进行优先级分配和阐述了任务间的相互通信机制, 最后根据智能接触器所要实现的功能要求设计了各用户任务, 包括 PWM 节能任务、A/D采样及数据处理任务、CANopen 实时通信任务, 其中详细阐述了 CANopen 协议栈的实现及各个通信对象的设计。

第 5 章 系统测试与分析

5.1 系统开发环境与测试工具

本设计选用的 DZ60 芯片为 Freescale 公司的一款性能优良的芯片。因此软件编程采用 Freescale 的集成开发环境 CodeWarrior。除了 $\mu\text{C}/\text{OS-II}$ 的移植程序中采用了汇编，其他都采用 C 语言开发。并且尽量遵循模块化、分层的开发习惯。图 5.1 为软件编译界面。

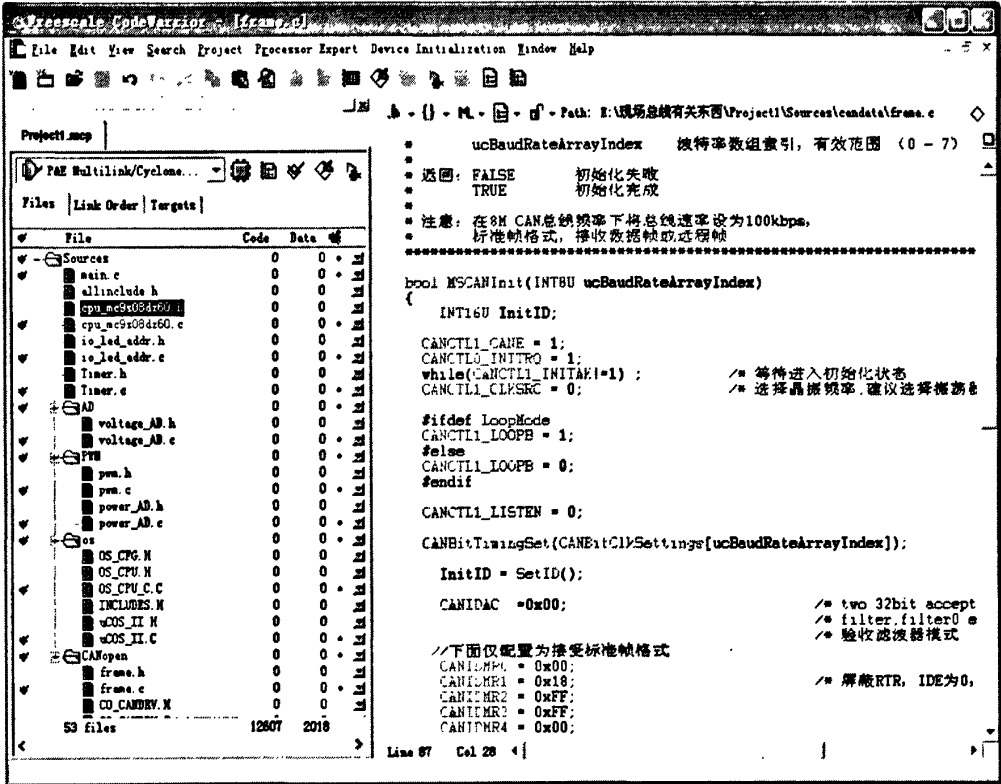


图 5.1 软件编译界面

为了验证对基于 CANopen 总线的智能接触器系统工作原理分析的正确性、系统实现的可行性，了解系统设计的软硬件工作的性能状况，本章对实验样机做了相关调试，同时对结果进行了分析。

实验中用到的仪器有：

- (1)示波器：Tektronix 公司的 TDS2004B 数字式荧光示波器 TDS2004B；
- (2)仿真器：Freescale 公司的原装 P&E USB Multilink Interface 烧写器；
- (3)万用表：EM-G0057 数字万用表；
- (4)周立功 USB-CAN 接口卡；
- (5)接触器电寿命、机械寿命测试台、电流表、电压表；

从站的测试分为功能测试和通讯测试，功能测试是验证智能接触器自身的节能

及相关功能^[42]，通讯测试是主站和从站的通讯测试，通过它来验证接触器从站与主站之间是否能够通过 CANopen 通讯实现信息的交互及远程监控。

5.2 节能模块测试与分析

5.2.1 节能接触器的节能能耗分析

1. 节能接触器吸持能耗分析

按照 GB 21518-2008《交流接触器能效限定值及能效等级》对采用智能控制后的 TGC5-500 交流接触器的吸持能耗进行测试。

试验方法：在环境温度 $25\text{C}^0 \pm 5\text{C}^0$ 下，对接触器线圈施加额定控制电源电压 ($100\%U_s$)，达到热稳定状态时，测试接触器闭合状态下的吸持功率，接触器试验接线图见图 5.2(测量用电压表、电流表精度为 0.2 级)。

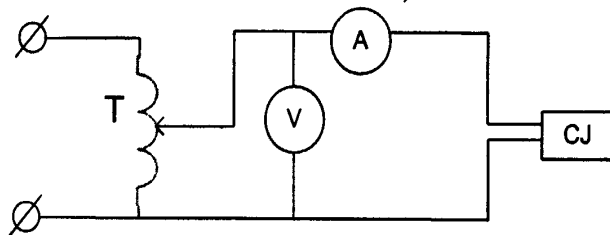


图 5.2 试验接线图

其中，T：调压器；CJ：交流接触器线圈；A：电流表；V：电压表

试验平台：节能接触器吸持能耗的测试平台见图5.3，传统的接触器按照国家标准，控制电源直接接上线圈部分，测试线圈部分的功耗即可。而节能接触器增加了电子模块，电磁铁中的线圈电压是通过线路板进行控制的，所以电子模块的输出端再接线圈两端。因此在接触器闭合状态下测试吸持能耗则是测线路板与线圈的总功耗。具体测试点即测试线路板输入端的电压电流即可，控制电源电压可由图中的调压器进行调节。

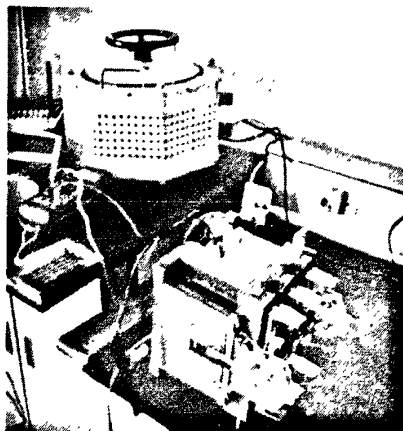


图 5.3 试验接线图

试验结果及分析:

经试验测量, 原样机(TGC5-500)在额定控制电源电压 AC380V 时的吸持能耗为 152VA。由表 5.1 可知, 采用智能控制后, 接触器在不同励磁电压下的功耗基本相同, 约为 10.6VA 左右, 较原样机节能 90%以上。经试验, 其吸持阶段的电流值基本上为 0.32A 左右, 平均电压为 33V 左右。采用智能控制后的接触器因使用了直流吸持, 取消了磁轭上的短路环, 大大降低了无智能控制时短路环所消耗的大量电磁能耗。从表 5.1 中也可看出, 采用智能控制的接触器在不同控制电压下的能耗基本保持接近, 达到了优化的目的。

表 5.1 不同电压下接触器的吸持能耗比较 (TGC5-500)

控制电压 (V)	无智能控制时/(V*A)	有智能控制时/(V*A)
300	120	7.2
380	152	10.6
400	165	11.6

2. 节能接触器动态过程测试及分析

测试节能模块时三相主回路无须接入市电, 在此选取采用智能控制后的 TGC5-500 交流接触器进行试验(控制电源为 380V)。

试验方法: 电磁铁的接触器线圈上接出一个电流互感器, 对电流信号进行比例减小, 对电流互感器的输出信号进行测试, 用示波器观察。

试验结果及分析:

用示波器观察线圈吸合全过程电流波形见图 5.4。

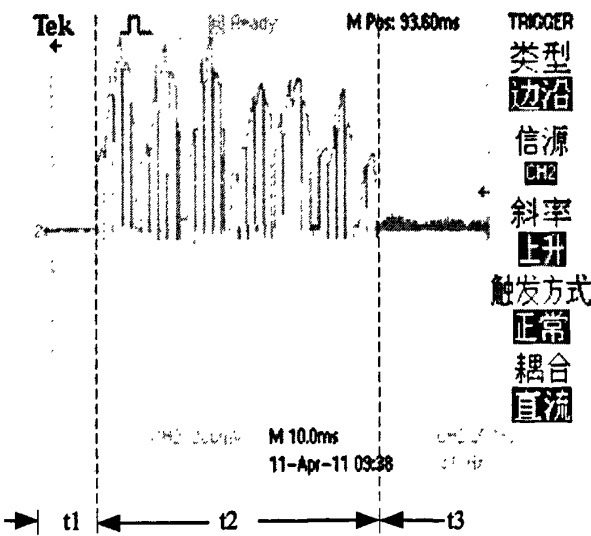


图 5.4 启动及吸合过程电流波形图

(1)电流建立阶段(t1)。线圈内电流从建立到到一定值时, 电磁吸力克服反力, 电磁铁准备开始进行吸合了。

(2)吸合阶段(t₂)。当 MCU 检测电源电压大于门槛电压时, 进入吸合阶段, 由 PWM 波调制后的半波正弦信号, 如图中所示, 有 7 个脉冲波峰, 延迟 7 个半波, 即 70ms。

(3)吸持阶段(t₃)。在检测到吸合过程完成后, 为避免触头的弹跳, 延时一段时间, 然后转入吸持状态。如图中所示, 转为小脉冲吸持, 处于低能耗保持, 实现节能目的, 直到释放。

5.2.2 节能接触器的电寿命、机械寿命测试与分析

按照 GB 14048.4-2010 对采用智能控制后的 TGC5-500 交流接触器的电寿命及机械寿命进行测试, 电寿命和机械寿命试验属于接触器的特殊试验。

1. 电寿命试验判定

试验条件: 接触器的电寿命用相应于表 5.2 所规定的条件下, 不修理或不更换任何零部件所能承受的有载操作次数来表示。

表 5.2 不同使用类别下验证电寿命的接通和分断条件

使用类别	额定工作电流 A	接 通			分 断		
		I/I _e	U/U _e	cos φ 或 L/R (ms)	I _c /I _e	U _r /U _e	cos φ 或 L/R (ms)
AC-1	全部值	1	1	0.95	1	1	0.95
AC-2	全部值	2.5	1	0.65	2.5	1	0.65
AC-3	I _e ≤17	6	1	0.65	1	0.17	0.65
	I _e >17	6	1	0.35	1	0.17	0.35
AC-4	I _e ≤17	6	1	0.65	6	1	0.65
	I _e >17	6	1	0.35	6	1	0.35
DC-1	全部值	1	1	1	1	1	1
DC-3	全部值	2.5	1	2	2.5	1	2
DC-5	全部值	2.5	1	7.5	2.5	1	7.5

注: cos φ 的误差为±0.05; L/R 的误差为±15%。

I_e: 额定工作电流。U_e: 额定工作电压。I: 接通电流。交流接通电流用有效值表示, 但对称电流的峰值在相应的电路功率因数下可能会呈现出较高值。U: 外施电压。U_r: 工频或直流恢复电压。I_c: 分断电流。

试验方法: 根据 GB 14048.4-2010 的规定, 试验一般选用 AC-3 使用类别。对于电寿命试验条件中接通与分断条件不相同的电器, 由表 5.2 可以看出, 其接通条件为 $I/I_e=6$ 、 $U/U_e=1$, 分断条件为 $I_c/I_e=1$ 、 $U_r/U_e=0.17$, 由于其接通条件与分断条件不同, 对试验电路提出了较高的要求。一般均以结构简单的电阻器与空心电抗器串联作为负载, 来模拟真实的电动机负载, 这时, 一般采用双电源的试验电路, 其主电路的原理接线图(单线图)如图 5.5 所示。

为了克服图 5.5 试验电路中损失太大的缺点, 可采用电力半导体的接触器电寿命试验装置, 即采用电力半导体取代图中的 KCI^[43], 从而减少 AC-3 类电寿命试验时陪试接触器的消耗量, 减少试验时的噪声, 可使试验时的维修工作大大减少。

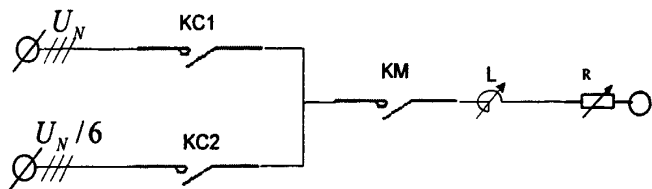


图 5.5 AC-3 类电寿命试验主电路

试验结果的判定：试验后，接触器在室温下应仍能满足标准中动作范围所规定的操作要求，并应能承受 $2U_N$ 但不小于 1000V 的介电性能试验，其施加部位如下^[44]：

- (1)主触头闭合时连接在一起的所有各极的全部带电体与接触器支架之间；
 - (2)主触头闭合时每一极与接触器连接至支架的所有其余各极之间。
2. 机械寿命试验判定

试验条件：接触器的机械寿命用其在需要维修或更换机械零件前所能承受的无载操作循环次数来表示。通常，机械耐久性试验中应有 90%以上的被试电器在不需要维修或更换任何零部件的情况下达到或超过该无载操作循环次数^[44]。推荐的机械寿命次数为(万次)：0.1，0.3，1，3，10，30，100，300 和 1000。

试验方法：接触器应按正常使用条件安装，特别是应如正常使用情况一样接好导线，试验时主电路不通电。控制电磁铁的线圈应施加额定电压，对于交流，应为额定频率的额定电压。如果有电阻或阻抗与线圈串联，不管操作时短接与否应如正常操作时一样将这些元件接入一起进行试验^[44]。

试验结果的判定：

机械寿命试验后，接触器应仍能满足室温下动作范围所规定的操作要求，且连接导线用的零部件不松动。关于试验最终合格与否的判定，可采用双三制判别法或者单八制判别法。

3. 电寿命与机械寿命测试平台

采用天正电气公司自己的电寿命、机械寿命测试平台进行测试。电寿命和机械寿命测试平台见图 5.6。

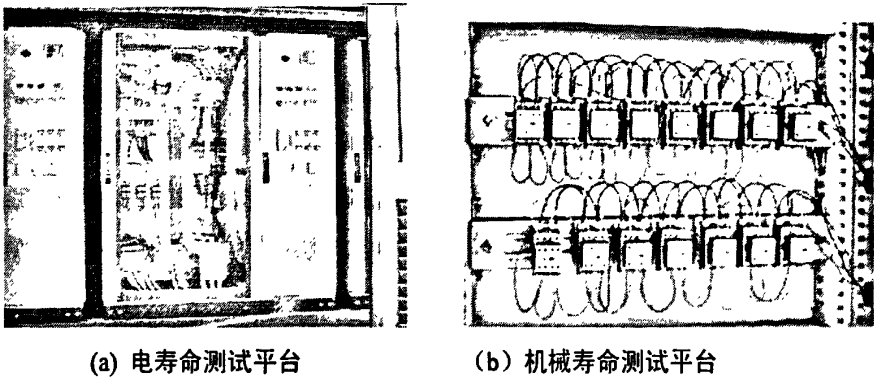


图 5.6 电寿命与机械寿命测试平台

4. 电寿命与机械寿命试验结果及分析

从表5.3可以看出，智能接触器采用节能控制后，保证了线圈供电电压在极小的范围内变化，使触头及铁芯碰撞后产生的弹跳时间、次数以及能量大大缩小，不仅保证了产品可靠性，更进一步提高产品的电气寿命和机械寿命指标。从而大大提高了接触器的使用寿命，节约了铜、铁等自然资源，更进一步避免了接触器易烧毁的故障现象。

表 5.3 节能接触器的电寿命、机械寿命试验结果

型号规格	电寿命（万次）		机械寿命（万次）	
	条件：AC-3；Ue：380V			
	无智能控制	有智能控制	无智能控制	有智能控制
TGC5-500	60	80	600	800

5.2.3 EMC 测试与分析

EMC测试即电磁兼容测试，应符合GB/T 17626电磁兼容国家标准的规定^[45]。同时本文中设计的智能接触器应依据GB/T 14048.4-2010中8.3节的EMC测试要求进行测试。在8.3节规定具有电子线路的电器的抗扰度测试应符合表5.4中规定的性能判据要求。

表5.4 抗扰度试验规定的验收判据

项目		验收判据	
		A	B C
一般要求	在规定限值内的正常性能	功能或性能出现暂时性降低或丧失，但可自行恢复	功能或性能出现暂时性降低或丧失，但需要人为介入或系统复位使其恢复，不应有任何零件（元件）的损坏
电源和控制电路的工作	无误动作 —接触器保持在预期位置； —过载继电器不应脱扣	瞬间的误动作，但不足以引起脱扣。 不允许触头误接通或分断。能自行恢复	过载继电器脱扣。 允许触头误接通或分断。 不能自行恢复
显示器和辅助电路的工作	可见的显示器信息没有变化。 仅有LED灯亮度的轻微波动或字符的漂移	暂时的可见变化，例如不期望的LED发光。 辅助触头无误动作	永久丧失显示信息 辅助触头无误动作
信息处理和检测功能	辅助触头无误动作 与外部设备进行通信和数据交换无误动作或错误信息	由于外部影响临时干扰通信，但可自行恢复	信息的错误处理，数据和（或）信息丢失，不能自行恢复

EMC测试主要包括ESD、浪涌抗扰度、快速瞬变脉冲群抗扰度、射频电磁场辐射抗扰度、射频场感应的传导骚扰抗扰度、衰减振荡波抗扰度、电压降落和短时中断抗扰度等^[45]。主要试验如下：

1. 静电放电 ESD 测试

试验环境和试验方法：使用静电放电发生器进行静电测试，见图 5.7。

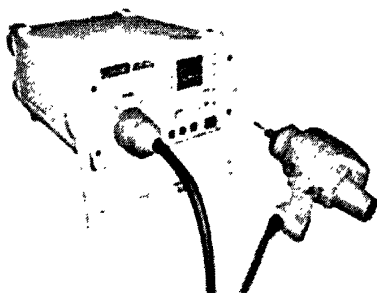


图 5.7 静电放电发生器

按 GB/T 17626.2-1998 进行测试：

空气放电：8kV；放电部位：非金属部位；

接触放电：4kV；放电部位：金属部位；

放电次数：正负各 10 次；

间隔时间： $\geq 1s$ ；

电源端子不必进行该试验，仅线圈通电。

符合 GB 14048.4-2010 中性能判据 B(见表 5.4)要求(不允许触头误接通或分断)。

试验结果及结论：

试验时接触器情况：未出现误接通或分断现象。

试后动作特性验证：满足产品原有门槛特性要求。

2. 电快速瞬变脉冲群 (EFT/B)

试验环境和试验方法：使用电快速瞬变脉冲群发生器进行脉冲群测试，见图 5.8。

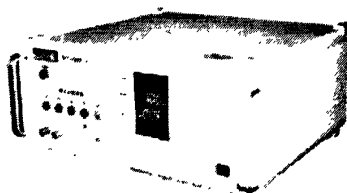


图 5.8 电快速瞬变脉冲群发生器

按 GB/T 17626.4-1998 进行测试：

试验水平：2kV；

重复频率：5kHz；

施压时间：1min；

试验部位：

试验电压施加在控制电路端子上，应符合 GB 14048.4-2010 中性能判据 B(见表 5.4)要求(不允许触头误接通或分断)；

试验结果及结论：

试验时接触器情况：未出现误接通或分断现象。

试后动作特性验证：满足产品原有门槛特性要求。

3. 浪涌

试验环境和试验方法：使用雷击浪涌发生器进行浪涌测试，见图 5.9。

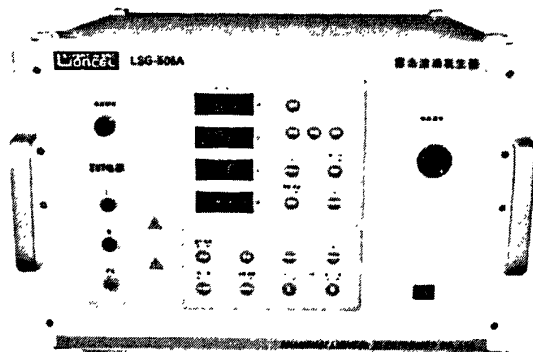


图 5.9 雷击浪涌发生器

按 GB/T 17626.5-1999 进行测试：

试验水平：2kV(共模) 1kV(差模)

正负极性各 5 次；

间隔时间：1min；

试验部位：

试验电压施加在控制电路端子上，应符合 GB 14048.4-2010 中性能判据 B(见表 5.4)要求（不允许触头误接通或分断）；

试验结果及结论：

试验时接触器情况：未出现误接通或分断现象。

试后动作特性验证：满足产品原有门槛特性要求。

5.3 CANopen 从站通信功能测试

本文是在项目的实施基础上进行设计的，因此前是在开发 CAN 下位机，线路板和驱动层都已经调试通过。但是对于应用层，因没有购买 CANopen 上位机主站卡，所以对整个产品的通讯功能的测试还有待下一步工作。在此，测试 CANopen 协议中通过上面的硬软件设计后的各个通信对象。采用周立功的 USB-CAN 接口卡进行测试，上位机软件为 ZLGCANTest。

5.3.1 通信测试平台的构建

CANopen 从站的通讯测试需要借助上位机进行，见图 5.10。TGC5-500 交流接触器的三相主回路接电压互感器及信号调理后送入单片机智能控制，通过 CANopen 模块与上位机进行通讯，智能模块中的硬件线路板外接 5 端的端子、24V

工业电源、CANH 与 CANL，智能模块的从站地址设置为 0x01。CANH 与 CANL 接 USB 转 CAN 的接口卡后，接入电脑实时控制，启动界面及电源进行数据监测。

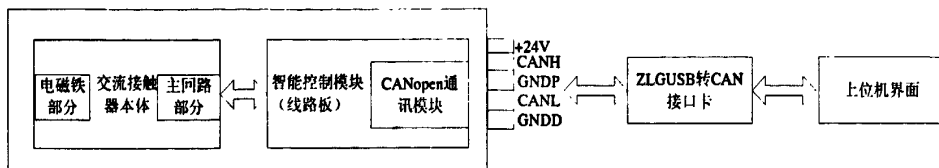


图 5.10 通信功能测试平台

先对周立功 CANTest 进行初始化配置，配置参数除设置波特率为 100kbps 外，其他默认。打开设备后，再启动 CAN。CANTest 初始化配置界面见图 5.11。

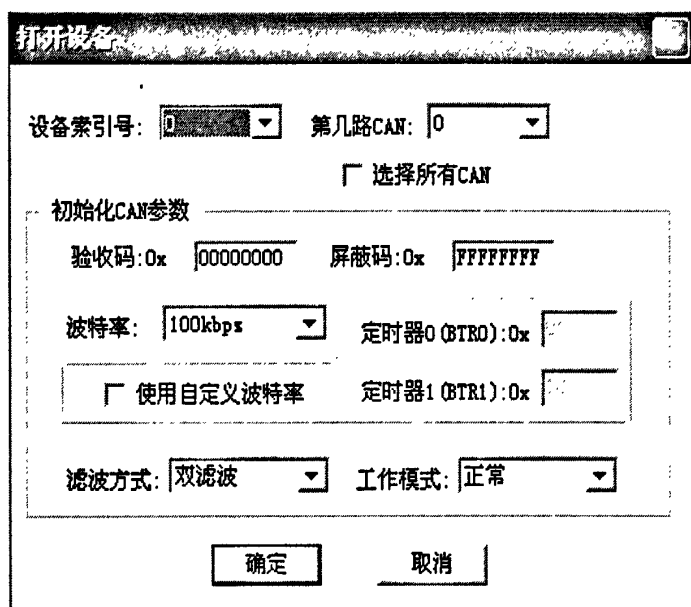


图 5.11 CANTest 软件初始化配置

5.3.2 通信测试结果及分析

将下位机节点设为 0x01。主回路接 380V 电压，在此选取采用智能控制后的 TGC5-500 交流接触器进行试验。

按照第 4 章中的 CANopen 协议栈的设计，在此进行各个报文的调试。采用 ZLGCANTest 界面进行从站的测试。CANopen 从站依据报文的 ID 来区分接收报文类型，因此可使用 ZLGCANTest 软件模拟实际的 CANopen 主站来给从站发送 NMT 启动与控制报文，只要 ZLGCANTest 发送报文时采用的报文 ID 与 CANopen 主站发送的时候采用的一致就行。

使用 ZLGCANTest 进行通讯测试时的报文显示界面见图 5.12。首先从站发送 NMT 报文中的 Boot-up 报文，该报文 ID 为 0x701 (0x700+节点 ID)，数据内容为 00，从站发送 Boot-up 报文之后自动进入了预运行状态。在图 5.12 中的心跳报文上的数据 0x7f 能够看出。从站进入预运行状态后，通过 ZLGCANTest 模拟主站发送

启动报文 0x01，使从站转入到运行状态，收到此报文后，从节点判断得出是转入运行态，则自动转入运行态，从心跳报文的数据字节变为 0x05 能够看出。

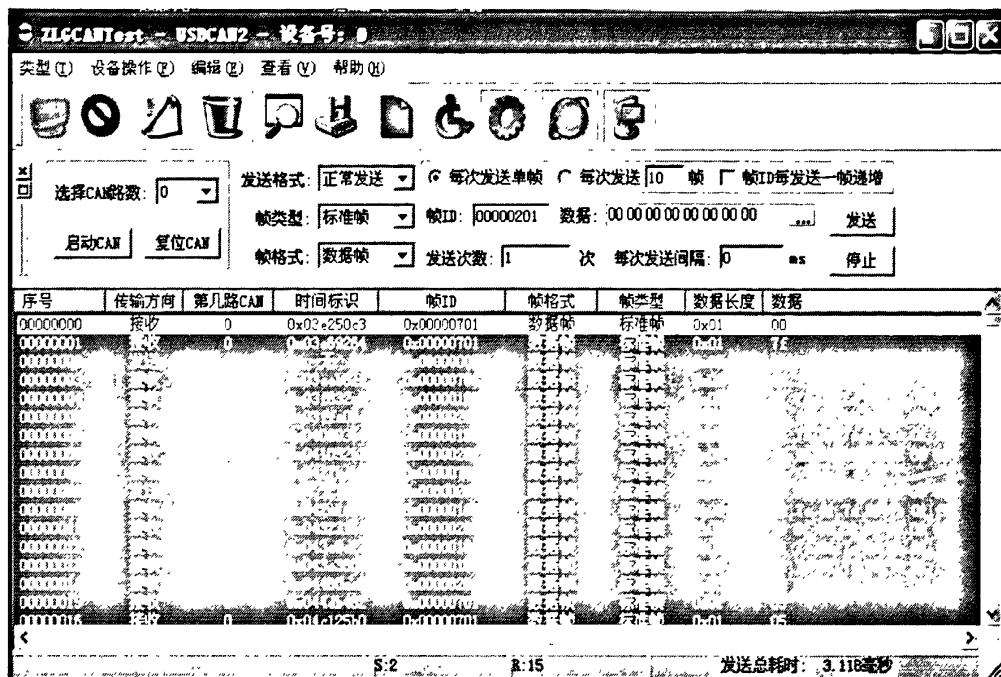


图 5.12 ZLGCANTest 通讯报文显示界面

在运行态中，PDO 有效，从节点会生成 TPDO 报文准备发送。从第 4 章中的应用对象与通信对象的分配表可知，只需 2 个 TPDO 报文即可。从界面可以看到，CANopen 主站会收到报文 ID 为 0x181 与 0x281 的 TPDO1、TPDO2，分别是三相电压、交流接触器的规格、工作状态(F0: 正常, 0F: 预报警)、吸合状态(F0: 吸持, 0F: 分断)。后续开发中，本课题会用 VC++ 直观的的实现各界面，可以直接观察三相电压和接触器的各个状态，直接选择各台接触器，并在遇到异常情况时用鼠标操作接触器的吸合。

假设遇到电路异常，过压或是欠压的状况，切断开关管，使线圈断电。在 ZLGCANTest 中发送报文 ID 为 0x201 的 RPDO1，数据为低电平 00 00，则从节点读取 RPDO1 中的指令并切断开关管，这从之后向上发送的 TPDO 中的数据中可以看到吸合态变化为 0x0f。

由上述可知，无论 Boot-up 报文、PDO 报文还是心跳报文，其帧格式、报文 ID、数据字节数都完全符合 CANopen 协议标准，同时网络启动过程和 NMT 状态机机制规定的完全一样，由此可见实现了 CANopen 从节点的通信功能，设计符合 CANopen 规范。

5.4 测量精度及误差分析

智能电器一般用于输配电或工矿企业等的供电系统中, 需要保证其长期无故障的运行。本文设计的智能接触器存在精度、稳定性等问题, 有待改善和提高。

1. 硬件误差分析

在本文的硬件系统中, 主要的误差由 A/D 采样误差产生, 且主要是由不同步采样误差带来的影响。由第 4 章可知, 在中、低压等级电网中, 截止频率一般选为至少 6 倍基波频率, 即 300 Hz, 满足中、低电压等级电网对测量精度的要求。但是在实际的采样中, 电源频率有时会发生波动, 会引起采样频率与电源频率不能保持整数倍。解决的根本办法就是使采样时间间隔随电源频率改变, 保证两者的整数倍关系不变。智能电器中常采用的实时检测电源频率, 在频率发生变化时, 用程序改变采样时间间隔。

2. 软件系统可靠性、稳定性分析

随着电器智能化系统应用范围的扩大, 用户对系统功能提出了更高、更多的要求。功能扩展使软件系统结构越来越复杂, 占有的资源也相应增加, 给软件系统的稳定性与可靠性带来了隐患。

(1) 提高程序代码的稳定性、抗非法操作性。

最好的办法是把一个大的功能模块中的功能分解为较小的、独立的子功能, 把它们分布到不同的软件结构中, 使每块功能构件中的编码简化、减小各构件间的耦合度, 提高系统的稳定性。这样做的好处是, 即使出现代码问题, 也被局限在一块功能构件中, 不至引起系统崩溃。

(2) 采用“喂狗”复位功能。

Watchdog 最早是用于单片机设计中, 以提高现场设备的软件抗干扰性。在程序主程序中设置好 Watchdog 寄存器的代码, 当操作系统、应用软件出现工作异常时, 系统将重新启动。

(3) 关键设备的双机热备份。

为了提高软件系统的可靠性, 除了保证软件稳定性之外, 还应对网络中关键的物理设备(如主站)实行双机热备份, 保证其中常用的一台出现软件故障时, 备用机可立即取代主机去工作。

5.5 本章小结

本章从两个方面对从站的性能进行了测试, 包括功能测试和通讯测试。功能测试验证了节能接触器自身的节能及相关功能, 通讯测试验证了主从站之间能否通过 CANopen 通讯实现信息的交互及远程监控, 最后给出了相关测试结果及结论。

总结与展望

1. 本文研究工作总结

本文是技术与工程设计紧密结合的实际课题。它是在浙江天正电气股份有限公司开发设计该类产品的的基础上,针对目前国内交流接触器能耗大、寿命有限、可靠性不高、易烧毁等特点,结合当前电力电子技术和通信技术的发展,设计了一款基于CANopen总线的智能交流接触器。主要完成工作如下:

(1)在分析国内外交流接触器发展的基础上,比较了国产接触器与国外接触器在能耗、寿命和可靠性等方面的差距,根据电力电子技术和现场总线技术在智能电器中的应用及我国新一代节能接触器的发展要求,提出了一种基于CANopen总线的智能交流接触器的设计方案。

(2)详细讨论了智能交流接触器的技术基础,包括交流接触器的PWM节能技术、嵌入式软件 $\mu\text{C}/\text{OS-II}$ 的相关介绍、CANopen应用层协议的分析及实现原理。

(3)根据系统设计的总体功能需求,设计了以MC9S08DZ60单片机为核心的硬件系统,给出了整个电路系统的原理图,包括主控电路、节能模块、信号采样与调理模块、CANopen通信模块、人机接口模块、电源模块等,并详细阐述了各模块电路的组成原理与具体电气电路,通过对硬件系统的介绍,使得对本文的智能接触器有了更加深刻的认识,为下面的软件设计奠定了基础。

(4)以 $\mu\text{C}/\text{OS-II}$ 嵌入式操作系统为开发平台开发了嵌入式软件。按照功能划分,各个任务首先独立编写调试,然后分配优先级及实现任务间的相互通信,最后联调实现所有的功能。软件模块化设计使得软件易于维护,大大缩短了开发周期,提高了系统的可维护性和可移植性。嵌入式软件的设计主要包括 $\mu\text{C}/\text{OS-II}$ 在DZ60上的移植、各个用户任务的设计(PWM节能任务、A/D采样及数据处理任务、CANopen实时通信任务),其中详细阐述了CANopen协议栈的实现及各个通信对象的设计。

(5)采用Freescale公司的开发软件CodeWarrior开发系统程序,并搭建了节能模块的测试平台和CANopen从站通信的测试平台。对节能模块的相关指标进行测试及分析,并通过模拟主站发送各项命令来进行远程控制,利用ZLGCANTest观察总线上传输的报文,结果表明这些报文的帧ID、数据字节的定义以及整个网络的运行过程都符合CANopen规范。

最终的测试结果表明,本文所设计的智能接触器能够稳定的运行并实现节能、低耗、长寿命等功能,且能通过主站实时监测并保护接触器,达到了预期的目标。

2. 本文研究工作展望

经过一年多的努力,本人对交流接触器的智能控制和PWM节能的开发有了一

个较为全面的认识和理解,且设计的智能接触器的各项指标也基本达到了预期的目标。然而,本课题的设计程度距离开放(易移植、易接入、易扩展)、实时(能够提供严格实时保障的通信能力)的目标,还有许多需要改进与完善的地方:

(1) 完整的 CANopen 网络还需要一个功能完善的主节点,设计中的主节点只完成了简单的通信对象的收发,还需一个具有完整管理和处理紧急错误等功能的主站。这个主站和多个下位机从站组成一个完整的 CANopen 网络,主站可以由 VC++ 设计,以一个主站界面的形式。

(2) 在本课题中, CANopen 从节点的实时性也是一个重要的内容,实时性主要在于对象字典的设计(EDS),本文中采用的数组设计实时性不够高,可以考虑用散列表等形式来设计^{[46][47]}。

(3) 本文设计的基于 CANopen 总线的智能接触器只是在干扰不是很强的工厂环境下进行的测试,对工业应用现场的抗干扰措施做的不够,有待实践中去加强硬件设计的抗干扰措施,在布局和布板中做的更加精细。

(4) 根据交流接触器的动力学方程可知,接触器的动力分析是较难的一块,本文中采用的 PWM 技术只是基本上实现了节能、无声、长寿命、小体积,后续中稳定性和弹跳的控制有待继续跟踪并加以优化,将节能接触器的性能及控制精度进一步提高。且各相主回路及控制回路的保护功能,如过压、欠压、缺相、漏电保护等功能更加完善,这都是后续阶段需要做的工作。

参考文献

- [1] 陈德桂.交流接触器技术发展动态.低压电器, 2003(4): 8-12, 16
- [2] 陈德桂. 开关电器电子化与智能化近期发展动态.电工技术杂志, 2004(7): 01-05
- [3] 许志红, 张培铭.低压电器的智能化设计与控制技术进展.电气时代, 2007(2) : 54-57
- [4] 陈德为, 张培铭.智能电器概念的发展. 低压电器, 2006(6): 15-18
- [5] 李峰.基于工业以太网的智能接触器的研究: [沈阳工业大学硕士学位论文]. 沈阳: 沈阳工业大学, 2008, 04-08
- [6] 周淑军, 王晓颖.交流接触器节点技术的发展. 电气时代, 2005(2) : 65-66
- [7] 许志红, 张培铭, 郑昕.交流接触器智能可靠性问题的分析.低压电器, 2002(3): 10-13
- [8] 林文生.智能混合式交流接触器的设计与研究: [哈尔滨工业大学硕士学位论文].哈尔滨: 哈尔滨工业大学, 2006, 01-05
- [9] 王岗.低压电器的现状及技术特点.中国高新技术企业. 2010(19) : 87-88
- [10] 饶运涛, 邹继军, 王进宏等.现场总线 CAN 原理与应用技术(第 2 版).北京: 北京航空航天大学出版社, 2007, 17-19, 378-380
- [11] 田山.基于 CANopen 协议的网络主控制器的设计: [上海交通大学硕士学位论文]. 上海: 上海交通大学, 2008, 17-18
- [12] 李强, 胡景泰, 徐涛等.一种基于 PIC16F685 微处理器的电磁铁控制方法.低压电器, 2010(2): 27-31
- [13] 申潭, 陈德桂, 冯涛.带反馈控制的智能交流接触器.低压电器, 2005(2): 3-5,10
- [14] 刘颖异, 陈德桂, 纽春萍等.带电压反馈的智能接触器动态特性及触头弹跳的仿真与研究.中国电机工程学报, 2007, 27(30): 20-25
- [15] 杨喆, 姚建军, 冯涛等.智能交流接触器专用芯片系统的测试. 低压电器, 2009(15): 20- 24
- [16] 邵贝贝.嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ (第 2 版). 北京: 北京航空航天大学出版社, 2003, 283-284
- [17] 任哲, 潘树林, 房红征.嵌入式操作系统基础 $\mu\text{C}/\text{OS-II}$ 和 Linux. 北京: 北京航空航天大学出版社, 2006, 68-69, 221-235, 141
- [18] Xu Zhe,Dong Shifeng.The Design and Implementation of a CANopen Slave Stack for Powertrain Controllern in Hybrid Electric Vehicle.In: Intelligent Computation Technology and Automation.Beijing,2010,755-758
- [19] CANopen Application Layer and Communication Profile.CiA Draft Standard 301,

Version 4.02, Feb 2002

- [20] Vazquez Luis, Rojas Leandro, Galceran Samuel, Sudria Antoni. Simplified CANopen application layer model for educational proposals. In: 8th International Conference on Emerging Technologies and Factory Automation. Antibes-Juan les Pins, 2001, 255-260
- [21] 王芳. 基于汽车总线模型的 CANopen 协议的实现: [天津理工大学硕士学位论文]. 天津: 天津理工大学, 2008, 38-50
- [22] 张云廷. 基于 CANopen 协议的从站的设计与应用: [北京工业大学硕士学位论文]. 北京: 北京工业大学, 2009, 06-07
- [23] Chen Zai Ping, Wang Feng, Jia Chao. Research of CANopen slave node based on embedded system. In: 2011 International Conference on Information Science, Automation and Material System. Zhengzhou, 2011, 629-632
- [24] Wang Baoqiang, Wang Zhirou, Yang Liangliang. Design of Embedded Meteorological Data Acquisition System Based on CANopen. In: Proceedings of the International Conference on Electrical and Control Engineering. Wuhan, 2010, 732-735
- [25] Kafader Urs. CANopen excels in microdrive motor control. Electronic Products, 2010, 52(5): 133-136
- [26] Ma Wenmin, Wang Zhigang, Wang Xinlei. Shipping monitoring system based on CANopen. Electronic Science and Technology, 2009, 22(7): 77-81
- [27] 王汝文, 宋政湘, 杨伟. 电器智能化原理及应用. 北京: 电子工业出版社, 2003, 92-93, 79-80
- [28] Freescale. MC9S08DZ60 datasheet. Freescale, 2009.10
- [29] Jinrong Shen, Jiakai Song. Design of Low Power Intelligent Compound Switch for Reactive Power Compensation Based on Single-chip ATMEGA16L. In: 2009 Second International Symposium on Computational Intelligence and Design. Changsha, 2009, 123-126
- [30] Yingyi Liu, Degui Chen, Liang Ji, Yingsan Geng. Dynamic Characteristic and Contact Bounce Analysis for an AC Contactor with PWM Controlled Coil. In: Electrical Contacts 2007. 53rd IEEE Holm Conference on Electrical Contacts. Pittsburgh, 2007, 289-293
- [31] 陈德桂, 冯涛, 耿英三等. 带电压反馈智能控制的交流接触器. 中国专利. 200510096055, 2007-11-28
- [32] 王建华, 耿英三, 姚建军等. 基于可变电流控制的智能交流接触器控制单元. 中国专利. 200710018027, 2010-05-19

- [33] 刘艳强, 马秋霞, 王建. CAN 总线通讯的电磁兼容性能分析与设计. 电子技术应用, 2006(8): 77-79
- [34] Philips Semiconductor. SJA1000 Data Sheet, Philips Semiconductor, 2000
- [35] 童晓斌, 戴瑜兴. 基于实时操作系统 $\mu\text{C}/\text{OS-II}$ 的智能控制器嵌入式软件设计. 电气应用, 2007, 26(8): 77-80
- [36] 李洪林. 基于 ARM 和 $\mu\text{C}/\text{OS-II}$ 的微机线路保护装置的研制: [合肥工业大学硕士学位论文]. 合肥: 合肥工业大学, 2009, 45-46
- [37] 许志红, 张培铭. 智能交流接触器动态吸合过程研究. 中国电机工程学报, 2007, 27(18): 108-113
- [38] 刘颖异, 陈德桂, 袁海文等. 带电流反馈的永磁接触器动态特性仿真与分析. 中国电机工程学报, 2010, 30(15): 118-124
- [39] Wada M, Yoshimoto H, Kitaide Y. Dynamic analysis and simulation of electromagnetic contactors with AC solenoids. In: Proceedings of the 28th Annual Conference of the IEEE Industrial Electronics Society. Sevilla, 2002, 2745-2751
- [40] Patrick K, Nelson Si, Nelson B, et al. Coupled field and circuit analysis considering the electromagnetic device motion. In: IEEE Transactions on Magnetics. Sapporo 2000, 1458-1461
- [41] Yoshihiro K, Tadashi Y, Kei I, et al. 3-D finite element analysis of dynamic characteristics of electromagnet with permanent magnets. IEEE Transactions on Magnetics, 2006, 42(4): 1339-1342
- [42] Taczynski W, Hejman T, Smugaia D. Computer-controlled testing system for investigating the dynamic characteristics of contactors with AC. Electromagnet drives, 2003(33): 313-323.
- [43] 陆俭国, 张乃宽, 李奎. 低压电器的试验与检测. 北京: 中国电力出版社, 2007, 153-159, 147-152
- [44] GB 14048.4-2010. 低压开关设备和控制设备第 4-1 部分: 接触器和电动机起动器. 北京: 中国国家标准化管理委员会, 2010, 65-68
- [45] GB/T 17626.4-1998. 电快速瞬变脉冲群抗扰度试验. 北京: 中国标准出版社, 2000, 92-93
- [46] 徐喆, 阎士珍, 宋威. 基于 MC9S12DP512 和 $\mu\text{C}/\text{OS-II}$ 的 CANopen 主站开发. 计算机工程与科学. 2009, 31(5): 118-120, 125
- [47] 徐喆, 阎士珍, 宋威. 基于散列表的 CANopen 对象字典的设计. 计算机工程. 2009, 35(8): 44-46

致 谢

在千年学府特有的浓厚学术氛围中，我终于完成了硕士研究生阶段的学业，在此，我对那些帮助和支持我的老师、家人、同学及朋友表示衷心的感谢！

首先，我要衷心的感谢我的指导老师戴瑜兴教授。戴瑜兴老师严谨的学术作风、确保科研成功的方法，踏实肯干的务实精神给我留下了深刻的印象，在其影响下使我形成了善于动脑、勤于动手的习惯。在三年的攻读硕士学位期间，在他的教育和熏陶下，我不仅学到了大量的知识，丰富了头脑，增强了实际动手能力，而且明白了做人的道理。更为重要的培养了我独立思考、解决问题的能力，培养了我克服困难、增强自信力的能力，为今后的学习和工作打下了扎实的基础。本论文从选题、撰写和修改到最后的定稿无不倾注了戴老师的心血。

感谢浙江天正电气股份有限公司的领导给我提供了宝贵的实践机会以及工作上的支持，特别要感谢李鹏总工、技术中心主任王卫国工程师、王兴仙工程师、杨斌工程师、陈晓军工程师在技术上的指导，生活上的关心。感谢同学好友周琴、单志华、李二强、文露、邬思奇、柴世杰、沈雁、谢云燕等，是他们给我提供了热心的帮助和鼓励。

同时感谢所有任课老师以及湖南大学电气与信息工程学院所有帮助和关心我的老师。

最后，我要感谢我的家人，尤其是我的爸爸，一直以来他对我的学业和生活给予着无私的支持和帮助，更多的还有鼓励和爱。我将用我一生的努力来回报他们。

我再一次真诚地对你们说：谢谢你们！谨以此文献给我的父母、亲人以及所有关心帮助我的人们！

田莉

2011年5月于岳麓山

附录 A 攻读学位期间发表的论文

- [1] 田莉, 戴瑜兴, 王卫国. 应用于智能断路器的CAN总线接口设计与实现. 计算机测量与控制, 2011, 19(4): 966-968。

附录 B 攻读学位期间参加的科研项目

- [1] 2009.7-2010.8 在浙江天正电气股份有限公司, 参与浙江省重大科技专项(湖南大学与浙江天正集团的校企合作项目)“基于现场总线的智能断路器技术研究与应用”[2007C11072]。主要负责CAN总线在智能断路器上的应用的研究工作。负责CAN总线方案的设计、下位机嵌入式软件的编写、硬件电路的相关设计, 后期的联调、EMC测试。本课题已于2010.11通过浙江省科技厅验收, 验收证书编号[浙科验字【2010】1879号]。

- [2] 2007.10-2008.5 参与湖南大学与深圳市华力特电气股份有限公司的校企合作项目“在线式电能质量监测仪”。负责前期的准备工作和方案设计、理论算法的研究, 主要负责三相不平衡度和闪变算法在matlab和CCS上的研究与仿真。

- [3] 实用新型专利.《一种 MODBUS/TCP 与 DeviceNet 的协议转换器》.专利号:[ZL 2008 2 0210950.7].第三人。

附录 C 部分程序代码

C.1 OS_CPU.H

```

#ifdef OS_CPU_GLOBALS
#define OS_CPU_EXT
#else
#define OS_CPU_EXT extern
#endif

#ifndef NULL
#define NULL 0
#endif

/*****
*                                定义数据类型
*****/

typedef unsigned char  BOOLEAN;
typedef unsigned char  INT8U;
typedef signed   char  INT8S;
typedef unsigned short INT16U;
typedef signed   short INT16S;
typedef unsigned long  INT32U;
typedef signed   long  INT32S;
typedef float          F32;
typedef double         F64;
typedef INT16U   OS_STK;    /* Each stack entry is 16-bit wide */
typedef INT16U   OS_CPU_SR; /* The CPU Status Word is 16-bit wide */
//-----

#define OS_TASK_SW() asm swi
//-----

#define OS_STK_GROWTH 1 /* Interrupt vector # used for context switch */
//-----

#define OS_CRITICAL_METHOD 1
//-----
#if OS_CRITICAL_METHOD == 1
#define OS_ENTER_CRITICAL() __asm sei
#define OS_EXIT_CRITICAL() __asm cli
#endif
//-----
#if OS_CRITICAL_METHOD == 2
#define OS_ENTER_CRITICAL() asm tpa;asm psha;asm sei//推CCR寄存器入栈
#define OS_EXIT_CRITICAL() asm pula;asm tap          //弹CCR寄存器出栈

```

```
#endif
//-----
#if OS_CRITICAL_METHOD == 3
#define OS_ENTER_CRITICAL() asm tpa;asm sei;asm sta cpu_sr
#define OS_EXIT_CRITICAL()  asm lda cpu_sr;asm tap
#endif
//-----
#define OS_SAVE_SP(); if(OSIntNesting==1){asm ldx OSTCBCur;asm tsx;asm txa;asm sta 0,x;}
```

C.2 OS_CPU_C.C

```
#ifndef OS_CPU_MC9S08DZ60_C
#define OS_CPU_MC9S08DZ60_C
#include "INCLUDES.H"
#include <MC9S08DZ60.h>
/*****
*
* 初始化新建任务的堆栈
* 说明: 此函数被 OSTaskCreate() 或 OSTaskCreateExt() 调用, 用于初始化新建的任务堆栈,
* 模拟一次中断时的现场保护, 分别将 PCL, PCH, X, A, CCR, H等寄存器依次入栈, 然后返回任务当前的堆栈指针
* 参数: task 任务执行函数
* pdata 传递给任务的数据块指针
* ptos 任务的栈顶地址, 实际传递的是栈顶的后一个地址
* opt 控制堆栈的初始化方式, 在 uCOS_II.H 中定义
* 返回: 返回任务堆栈指针, 实际传递的是栈顶的前一个地址
* 注意: 单片机 MC9S08DZ60 的栈顶为高地址, 入栈时堆栈指针递减
*****/
OS_STK *OSTaskStkInit ( void (*task)(void *pd),
                        void *pdata,
                        OS_STK *ptos,
                        INT16U opt )
{
    INT16U pc;
    INT8U *stk;

    (void)pdata; /* 避免编译时警告变量未使用*/
    (void)opt;

    pc = (INT16U)task;
    stk = (INT8U *)ptos;

    stk = (INT8U *) (ptos - 1); /* 获取任务实际上的栈顶*/
    *stk-- = (INT8U)pc; /* PCL */
    *stk-- = (INT8U)(pc >> 8); /* PCH */
    *stk-- = 0x00; /* X */
    *stk-- = 0x00; /* A */
}
```

```

*stk-- = 0x00;          /* CCR */
*stk = 0x00;            /* H */

return ((OS_STK *)stk); /* 入栈后返回任务当前的堆栈指针 */
}

/*****
*
* 开始执行最高优先级任务
* 说明: 获取当前最高优先级任务的 TCB 指针 *OSTCBHighRdy, 保存到 *OSTCBCur中,
        并用此指针获取任务的堆栈指针 *OSTCBExtPtr, 然后模拟一次单片机中断返回, 恢复当
        前最高优先级任务的现场, 依次将 H, CCR, A, X, PCH, PCL等寄存器出栈.
* 参数: 无
* 返回: 无
*****/

void OSStartHighRdy(void)
{
    OSTaskSwHook();      /* 调用用户自定义的任务切换钩子*/
    OSRunning++;          /* 指示多任务系统已启动 */

    asm
    {
// 获取当前最高优先级任务的 TCB 指针 *OSTCBHighRdy, 保存到 *OSTCBCur中
        LDX    OSTCBHighRdy /* X = HIGH( *OSTCBHighRdy ) */
        STX    OSTCBCur     /* HIGH( *OSTCBCur ) = X */
        PSHX                     /* [SP] = HIGH( *OSTCBCur ), SP -= 1 */
        LDX    OSTCBHighRdy:1 /* X = LOW( *OSTCBHighRdy ) */
        STX    OSTCBCur:1    /* LOW( *OSTCBCur ) = X */
        PULH                     /* H = HIGH( *OSTCBCur ), SP += 1 */
                                   /* 寄存器 H:X = *OSTCBCur, 保存了当前任务*/
                                   /* .. 的TCB指针 *OSTCBCur */

// 用当前任务的TCB指针 *OSTCBCur, 获取任务的堆栈指针 *OSTCBExtPtr
        LDA    0,X           /* A = HIGH( *OSTCBCur ) */
        PSHA                     /* [SP] = HIGH( *OSTCBCur ) SP -= 1 */
        LDX    1,X           /* X = LOW( *OSTCBCur+1 ) */
        PULH                     /* [SP] = LOW( *OSTCBCur+1 ), SP += 1 */
                                   /* 寄存器 H:X = *( *OSTCBCur ), 保存了任务*/
                                   /* .. 的堆栈指针 OSTCBExtPtr, 因为 TCB 的*/
                                   /* .. 首字节便是任务的堆栈指针 */

// 依次将 H, CCR, A, X, PCH, PCL等寄存器出栈
        TXS                     /* SP = (H:X) - 1 SP 重装 */
        PULH                     /* Pull (H), SP += 1 */
                                   /* 执行中断返回指令 RTI 时, H寄存器不能自*/
                                   /* .. 动出栈, 故需手动出栈 */
    }
}

```

```

RTI          /* Pull (CCR),          SP += 1 */
              /* Pull (A),          SP += 1 */
              /* Pull (X),          SP += 1 */
              /* Pull (PCH),        SP += 1 */
              /* Pull (PCL),        SP += 1 */

};
}

/*****
*
* 中断时的任务切换
* 说明: 将进入中断时的现场保存在当前任务的堆栈; 进入中断后, 单片机将 CCR, A, X, PCH,
PCL 硬件入栈, 编译器又自动将H寄存器入栈, 所以 SP += 6, 进入函数OSIntExit() 时,
单片机将 PCH, PCL 入栈, 编译器又自动将H寄存器入栈, 所以SP += 3, 进入本函数时,
单片机将 PCH, PCL 入栈, 编译器没有将 H 寄存器入栈, 故 SP += 2; 所以要得到进入
中断前的现场状态, 需要将 SP = 5, 即本函数的第一条指令 "AIS #5"; 然后获取最高
优先级 OSPrioHighRdy保存到当前优先级 OSPrioCur, 并获取当前最高优先级任务的
TCB 指针 OSTCBHighRdy, 保存到 *OSTCBCur中;然后模拟一次单片机中断返回, 恢复
当前最高优先级任务的现场, 依次将 H, CCR, A, X, PCH, PCL等寄存器出栈.
* 参数: 无
* 返回: 无
*****/
void OSIntCtxSw(void)
{
    asm
    {
        // 将进入中断时的现场保存在当前任务的堆栈
        AIS #5          /* 调用OSIntExit() 和本函数时, 占用了4字 */
                        /* .. 节堆栈空间,          SP += 4 */
        TSX              /* H:X = SP + 1          */
        PSHX             /* 寄存器(X)入栈          SP -= 1 */
        PSHH             /* 寄存器(H)入栈          SP -= 1 */
        LDX OSTCBCur:0    /* X = HIGH( *OSTCBCur ) */
        PSHX             /* *OSTCBCur 入栈        SP -= 1 */
        LDX OSTCBCur:1    /* X = LOW( *OSTCBCur ) */
        PULH             /* H:X = *OSTCBCur      SP += 1 */
        PULA             /* 寄存器(H)出栈, 保存到(A)中 SP += 1 */
        STA 0,X          /* HIGH( *( *OSTCBCur ) ) = H */
        PULA             /* 寄存器(X)出栈, 保存到(A)中 SP -= 1 */
        STA 1,X          /* LOW( *( *OSTCBCur ) ) = X */

        // 将就绪任务中最高的优先级别 OSPrioHighRdy
        // .. 保存到当前优先级别 OSPrioCur
        LDA OSPrioHighRdy /* OSPrioCur = OSPrioHighRdy */
        STA OSPrioCur

        // 获取当前最高优先级任务的 TCB 指针 *OSTCBHighRdy, 保存到 *OSTCBCur中
    }
}

```



```

LDX    OSTCBHighRdy    /* X = HIGH( *OSTCBHighRdy ) */
STXOSTCBCur            /* HIGH( *OSTCBCur ) = X */
PSHX                    /* [SP] = HIGH( *OSTCBCur ),    SP -= 1 */
LDX    OSTCBHighRdy:1  /* X = LOW( *OSTCBHighRdy ) */
STXOSTCBCur:1          /* LOW( *OSTCBCur ) = X */
PULH                    /* H = HIGH( *OSTCBCur ),        SP += 1 */
                        /* 寄存器 H:X = *OSTCBCur, 保存了当前任务*/
                        /* .. 的TCB指针 *OSTCBCur */

// 用当前任务的TCB指针 *OSTCBCur, 获取任务的堆栈指针 *OSTCBExtPtr
LDA     0,X              /* A = HIGH( *OSTCBCur ) */
PSHA                    /* [SP] = HIGH( *OSTCBCur )    SP -= 1 */
LDX     1,X              /* X = LOW( *(OSTCBCur+1) ) */
PULH                    /* [SP] = LOW( *(OSTCBCur+1) ), SP += 1 */
                        /* 寄存器 H:X = *( *OSTCBCur ), 保存了任务 */
                        /* .. 的堆栈指针 OSTCBExtPtr, 因为 TCB 的 */
                        /* .. 首字节便是任务的堆栈指针 */

// 依次将 H, CCR, A, X, PCH, PCL等寄存器出栈
TXS                    /* SP = (H:X) - 1 = **OSTCBCur  SP 重装 */
PULH                    /* Pull (H),                      SP += 1 */
                        /* 执行中断返回指令 RTI 时, H寄存器不能自 */
                        /* .. 动出栈, 故需手动出栈 */
RTI                    /* Pull (CCR),                      SP += 1 */
                        /* Pull (A),                        SP += 1 */
                        /* Pull (X),                        SP += 1 */
                        /* Pull (PCH),                      SP += 1 */
                        /* Pull (PCL),                      SP += 1 */

};
}
/*****
*
* 任务切换
* 说明: 使用软件中断切换任务, 将进入中断时的现场保存在当前任务的堆栈; 获取最高优先
级 OSPrioHighRdy 保存到当前优先级 OSPrioCur, 并获取当前最高优先级任务的 TCB
指针 OSTCBHighRdy, 保存到 *OSTCBCur中; 然后模拟一次单片机中断返回, 恢复当
前最高优先级任务的现场, 依次将 H, CCR, A, X, PCH, PCL等寄存器出栈.
* 参数: 无
* 返回: 无
*****/
interrupt 1 void OSCtxSw(void)
{
    asm
    {
        //PSHH            /* 寄存器(H)已由软件自动入栈    SP -= 1 */
        TSX              /* H:X = SP + 1 */
    }

```

```

PSHX          /* 寄存器(X)入栈          SP -= 1 */
PSHH          /* 寄存器(H)入栈          SP -= 1 */
LDX OSTCBCur:0 /* X = HIGH( *OSTCBCur ) */
PSHX          /* *OSTCBCur 入栈          SP -= 1 */
LDX OSTCBCur:1 /* X = LOW( *OSTCBCur ) */
PULH          /* H:X = *OSTCBCur          SP += 1 */
PULA          /* 寄存器(H)出栈, 保存到(A)中 SP += 1 */
STA 0,X       /* HIGH( *(*OSTCBCur) ) = H */
PULA          /* 寄存器(X)出栈, 保存到(A)中 SP -= 1 */
STA 1,X       /* LOW( *(*OSTCBCur) ) = X */

```

// 将就绪任务中最高的优先级别 OSPrioHighRdy

// .. 保存到当前优先级别 OSPrioCur

```

LDA OSPrioHighRdy /* OSPrioCur = OSPrioHighRdy */
STA OSPrioCur

```

// 获取当前最高优先级任务的 TCB 指针 *OSTCBHighRdy, 保存到 *OSTCBCur中

```

LDX OSTCBHighRdy /* X = HIGH( *OSTCBHighRdy ) */
STXOSTCBCur      /* HIGH( *OSTCBCur ) = X */
PSHX             /* [SP] = HIGH( *OSTCBCur ), SP -= 1 */
LDX OSTCBHighRdy:1 /* X = LOW( *OSTCBHighRdy ) */
STXOSTCBCur:1    /* LOW( *OSTCBCur ) = X */
PULH             /* H = HIGH( *OSTCBCur ), SP += 1 */
                /* 寄存器 H:X = *OSTCBCur, 保存了当前任务 */
                /* .. 的TCB指针 *OSTCBCur */

```

// 用当前任务的TCB指针 *OSTCBCur, 获取任务的堆栈指针 *OSTCBExtPtr

```

LDA 0,X          /* A = HIGH( *OSTCBCur ) */
PSHA             /* [SP] = HIGH( *OSTCBCur ) SP -= 1 */
LDX 1,X          /* X = LOW( *OSTCBCur+1 ) */
PULH             /* [SP] = LOW( *OSTCBCur+1 ), SP += 1 */
                /* 寄存器 H:X = *OSTCBCur, 保存了任务 */
                /* .. 的堆栈指针 OSTCBExtPtr, 因为 TCB 的 */
                /* .. 首字节便是任务的堆栈指针 */

```

// 依次将 H, CCR, A, X, PCH, PCL等寄存器出栈

```

TXS             /* SP = (H:X) - 1 = **OSTCBCur SP 重装 */
PULH            /* Pull (H), SP += 1 */
                /* 执行中断返回指令 RTI 时, H寄存器不能自 */
                /* .. 动出栈, 故需手动出栈 */
RTI             /* Pull (CCR), SP += 1 */
                /* Pull (A), SP += 1 */
                /* Pull (X), SP += 1 */
                /* Pull (PCH), SP += 1 */
                /* Pull (PCL), SP += 1 */

```

```

    };
}
/*****
*
*          实时计数器中断服务程序
* 说明: 实时计数器中断服务程序
* 参数: 无
* 返回: 无
*****/
interrupt 25 void CpulSrRtcTx(void)
{
    //CPU_RTC_INT_DIS()
    OSIntEnter();
    CPU_RTC_INT_CLR_FLAG() /* 清除中断标志位 */
    //OS_EXIT_CRITICAL(); /* 开中断,可选, 允许中断嵌套 */
    OSTimeTick();
    OSIntExit(); /* 退出中断并任务调度、切换 */
}

```

C.3 MCU的RTC配置, 供 μ C/OS-II时钟使用

```

/*****
*
*          实时计数器
* 实时计数器初始化 CPU_RTC_INIT()
* - 使用内部1K时钟源
* - 中断周期约10ms, 预分频1:10, 溢出值 10
*****/
#define RTCSC_VAL (INT8U)0b10001011 /* 状态和控制寄存器 */
#define RTCMOD_VAL (INT8U)0x0A /* 模数寄存器 */

#define CPU_RTC_INIT()\
{\
    RTCSC = RTCSC_VAL;\
    RTCMOD = RTCMOD_VAL;\
}

#define CPU_RTC_INT_EN() RTCSC_RTIE = 1; /* RTC溢出中断使能*/
#define CPU_RTC_INT_DIS() RTCSC_RTIE = 0; /* RTC溢出中断禁止 */
#define CPU_RTC_INT_CLR_FLAG() RTCSC_RTIF = 0; /* 清除中断标志位*/

```

C.4 μ C/OS-II在DZ60上移植代码的测试, 测试的主函数

```

#include <hidef.h>
#include "derivative.h"
#include <MC9S08DZ60.h>
#include "cpu_mc9s08dz60.h"
#include "os_cfg_r.h"
#include "os_cpu_mc9s08dz60.h"
#include "ucos_ii.h"

```

```
//-----
#define TASK_STK_SIZE 50
//-----
void StartTask(void *pdata);      //声明任务
void MyTask1(void *pdata);       //声明任务

//-----
OS_STK    StartTaskStk[TASK_STK_SIZE];  //定义开始任务的堆栈
OS_STK    MyTaskStk1[TASK_STK_SIZE];    //定义开始任务的堆栈
/*****/
void main(void) {

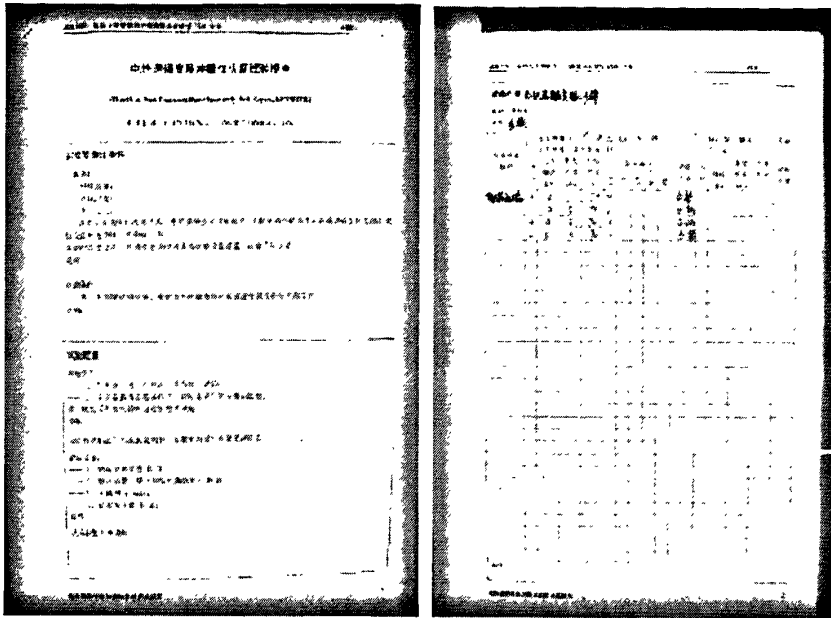
    CpuInit();                //单片机外设初始化
    PTADD_PTADD6=1;          //输出
    EnableInterrupts;
    OSInit();
    OSTaskCreate(StartTask,(void *)0,&StartTaskStk[TASK_STK_SIZE-1],1);
    OSTaskCreate(MyTask1,(void *)0,&MyTaskStk1[10],2);
    OSStart();
}/* please make sure that you never leave main */
//-----
//          task start
//-----
void StartTask(void *pdata){
    pdata = pdata;                /*为了消除警告    */
    //enable time tick for os
    CPU_RTC_INIT()                /* 实时时钟 RTC 初始化*/
    CPU_RTC_INT_EN()              /* RTC 溢出中断使能    */

    for(;;){
        PTAD_PTAD6=~PTAD_PTAD6;    //D7 灯的闪烁
        OSTimeDly(2);                //延时 20ms
    }
}
//-----
void MyTask1(void *pdata)
{

    pdata=pdata;
    for(;;)
    {
        PTBD_PTBD5=~PTBD_PTBD5;    //D5 灯的闪烁
        OSTimeDly(10);
    }
}
```

附录 D 部分 EMC 测试报告

D.1 电快速瞬变脉冲群



D.2 浪涌

