

摘要

随着人民生活水平的提高以及公众环保意识的增强,太阳能热水器以其诸多优点受到人们欢迎,逐渐走进了千家万户。然而目前太阳能热水器主要以家庭为单元安装使用,系统的智能化程度相对较低,功能简单,不具备网络控制和远程监控功能,而且太阳能集热器的效率较低。

基于上述原因,本文依据模糊 PID 控制算法和无线传感器网络(WSN)技术,以基于 ARM9 内核的 S3C2410 微处理器为核心,设计一套具有太阳能加热与电加热结合的太阳能集热系统,对建筑内所有用户集中提供热水。控制系统由主控制器与从控制器两部分组成,主控制器与从控制器间构成主从式多机无线传感器网络。本文首先介绍了太阳能热水系统的国内外发展现状以及太阳能集热系统的组成、原理和控制系统设计方案,然后从硬件和软件两部分对控制器进行设计。

硬件设计,包括主控制器和从控制器两部分。主控制器外围硬件电路的设计包括电源电路、存储器电路、XE1205 无线传输电路、以太网接口电路、水位检测电路、温度检测电路、键盘输入和报警电路、复位电路、输出控制电路等;从控制器硬件电路设计,主要包括以下电路模块:XE1205 无线传输模块、存储器模块、输出控制模块、遥控接收模块等。

软件设计,包括主控制器软件设计和从控制器软件设计。主控制器软件设计首先进行控制器软件平台的搭建,即 Linux 操作系统的建立,包括交叉编译环境的建立、系统引导程序 Bootloader 移植、Linux 内核移植和文件系统的建立四部分工作。接着编写在操作系统平台上的应用程序,包括 Qt 开发环境的搭建和具体应用程序的编写。从控制器软件设计主要是用 C 语言编写与主控制器间的通信程序、输出控制程序、实时显示程序等。

最后,对模糊 PID 控制算法进行了分析和研究,采用了二输入三输出的模糊控制器,确定了输入、输出变量的论域及隶属度函数,给出了 PID 参数的自整定规则,完成了模糊 PID 控制算法的设计及其实验仿真。

关键词: 太阳能, S3C2410, 嵌入式 Linux, WSN, 模糊 PID

The research and application of new solar control system

Lv Changjiang (Dection Technology&Automation Equipment)

Directed by Duan Chenxu

ABSTRACT

With the improvement of living standards and awareness of environmental protection, the solar water heater has gradually entered into thousands family with its many popular advantages. However, at present, the solar water heater system has been mainly used for the family units which have relatively low intelligent ability, simple function, unavailable network control and remote monitoring function, and that poor efficiency of the solar energy collector.

For these reasons, in this paper, based on fuzzy PID control algorithm and wireless sensor network (WSN) technology, with S3C2410 microprocessor based on ARM9 kernel as the core, the solar energy collection system with solar heating and electric heating is designed for the whole buildings and provide hot water for all users in the building. The control system includes main-controller and slave-controller, and the main-controller as well as the slave-controller constitutes the master-slave multi-motor wireless sensor network. This article firstly introduces the development situation of solar water heater system at home and abroad and the constitution, principles as well as the design scheme of the solar energy collection systems, and then the controller is designed from the aspect of hardware and software.

The hardware design includes main-controller and slave-controller. The design of peripheral hardware circuit in the main-controller includes power circuit, memory circuit, XE1205 wireless transmission circuit, Ethernet interface circuit, water level detection circuit, temperature detection circuit, keyboard input and alarm circuit, reset circuit, output control circuit and so on. The hardware circuit design of slave-controller system is made up of XE1205 wireless transmission module, storage module, output control module, remote controlled receiving module and so on.

The software design consists of main-controller and slave-controller. The first thing of main-controller software design is to set up the controller software platform, namely, the establishment of Linux operating system, which comprises the following four components:

the establishment of cross-compiler environment, the transplant of system bootloader, the transplant of Linux operating system kernel and the establishment of file system. Then the applications of operating system platform are written, including Qt structures of the development environment and the written of concrete application program. The software design of slave-controller mainly is used C language for communications program, output control program, real-time display program etc. between the main-controllers.

Finally, this paper makes an analysis and research on the fuzzy PID control algorithm, which uses two input three output fuzzy controller to confirm the universe of discourse of the input and output variables and the membership function, provide the PID parameters-self-adjusting rule, and accomplish the design and the experimental simulation of the fuzzy PID control algorithm.

Keywords: solar energy, S3C2410, embedded Linux, WSN, fuzzy PID

原创性声明

本人郑重声明：所提交的学位论文是本人在导师的指导下，独立进行研究取得的成果。除文中已经注明引用的内容外，论文中不含其他人已经发表或撰写过的研究成果，也不包含为获得山东建筑大学或其他教育机构的学位证书而使用过的材料。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。

学位论文作者签名：吕昌江 日期 2010.6.22

学位论文使用授权声明

本学位论文作者完全了解山东建筑大学有关保留、使用学位论文的规定，即：山东建筑大学有权保留并向国家有关部门或机构送交学位论文的复印件和磁盘，允许论文被查阅和借阅。本人授权山东建筑大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其它手段保存、汇编学位论文。

保密论文在解密后遵守此声明。

学位论文作者签名：吕昌江 日期 2010.6.22

导师签名：段志远 日期 2010.6.22

第 1 章 绪论

1.1 论文研究的背景及意义

能源作为国民经济的基础，对于社会经济发展和提高人民生活质量都极为重要，在高速增长的经济背景下，能源工业面临经济增长与环境保护双重压力。能源问题、环保问题是当今世界各国面临的重大问题之一。能源危机日益加剧，环境污染已威胁到自然界的生态平衡。统计数据表明，目前世界上 75% 的能源来自矿物燃料的燃烧，排放出大量有害气体颗粒，是人类最大的健康污染源，也是地球“温室效应”的祸首。能源战略中的可持续发展问题、能源发展过程中的环境保护问题已经成为联合国首脑会议的重要议题之一。

因此，节约能源、提高能源利用效率，尽可能多地利用洁净的新能源，是我国经济建设的基本国策。太阳能以其清洁、源源不断、安全等显著优势，成为关注重点^[1]。太阳能具有资源丰富、无需开采和运输、不污染环境、不破坏生态平衡等特点，因此太阳能的开发利用将有巨大的市场前景，它不仅带来很好的社会效益、环境效益，而且还具有明显的经济效益。

经过二十多年的研究与开发，我国太阳能热利用取得了很大成就，已经达到世界先进水平。其中，太阳能热水器作为节能、环保、低成本的绿色能源产品，已经成为我国可再生能源领域发展最快、规模最大、市场成熟最早的行业^[2]。特别是“九五”计划将大力开发能源和可再生资源列入我国重点项目以来，在企业规模化、产品标准化、技术国际化的产业发展指导思想的指导下，我国太阳能热水器产业迅速发展，初步形成了原材料加工、生产、制造、销售和安装服务相配套的产业化体系。我国太阳能热水器产业，从无到有，从规模较小到规模较大，随着市场需求的迅速增长和公众意识的不断加强而不断发展。我国已经成为太阳能热水器第一使用国和生产国，太阳能热水器已被国家列为重点推广的项目。《中华人民共和国可再生能源法》已于 2006 年 1 月 1 日施行，其中第十七条规定：国家鼓励单位和个人安装和使用太阳能热水系统、太阳能供热采暖和制冷系统、太阳能光伏发电系统等太阳能利用系统^[3]。今后，国家还将出台一些有利政策来鼓励太阳热水器的推广应用。这些不仅为太阳热水器的生产与安装提供了一个很有利的政治法律环境，而且为太阳热水器的推广应用提供了法律保障。

1.2 国内外发展现状与发展趋势

1、国内发展现状

中国太阳能热水器市场发展很快，从20世纪70年代后期就开始开发家用太阳能热水器。到20世纪90年，它不仅可为家庭和机关、旅社、医院等提供洗澡、洗衣、炊事等所需热水同时还可用于空调、干燥、农业种植、水产养殖、海水淡化等^[4]。截止到2001年底，我国太阳能热水器保有量约300万平方米，占世界保有量的百分之七十，年产量达到780万平方米，形成了年约500万吨标准煤的能力。全行业实现总产值近100亿元，从业人员超过15万，产品销售量是欧洲的10倍，太阳能热水器产品的出口额近年来以百分之四十至百分之五十的速率增长。

随着科学技术的进步，太阳能热水器也随之不断改进，由集热和贮热合为一体逐步改进为集热和贮热部分分开，采用先进的太阳能集热器，先进的保温材料和冬季防冻技术及其他辅助装置等^[5]。现如今生产的先进的太阳能热水器已实现了全年运行，大大提高了太阳能热水器的热效率和利用效率。目前，太阳能热水器是可再生能源技术领域商业化程度最高，推广应用最普遍的技术之一^[6]。

有专家预计到2015年，仅全国住宅用太阳热水器将达到2.32亿平方米的拥有量，普及率将达到20-30%，太阳能热水器行业前景和“钱”景都被双重看好^[1]。

2、国外发展现状

世界各国对太阳热水器的开发也十分重视，生产发展很快。各国纷纷制定太阳热水器技术产业的发展规划，大大地加速了该行业的发展^[7]。例如：澳大利亚政府规定，在北部地区新建房屋一定要设置太阳能热水器，西澳大利亚已有25%的新住宅安装了太阳能热水器^{[10][11]}。日本现在每年安装太阳能热水器近50万台，现在有20%的家庭安装了太阳能热水器，计划今后普及率达到25%，按照日本的“阳光计划”还将为公寓，办公楼安装6500套太阳能热水系统，为工厂安装1900套工业用太阳能热水系统。美国的“百万太阳能屋顶计划”实施将带动大批相关工业的发展。欧共体提出了“太阳热水系统热能供给保证体制”，以合同的形式对用户实现供热水的承诺，先后在法国、西班牙和德国加以实施，已取得了良好的效果。以色列是一个化石能源资源较少而太阳能资源十分丰富的国家，政府对太阳能热水器的开发利用进行了直接干预，规定所有新建筑物必须配备太阳能热水器，目前普及率已超过60%。韩国，东南亚各国也制定了支持开发利用太阳能热水器的优惠政策，纷纷吸收外资到本国开办太阳能企业随着科学技术的进步，太阳能热水器也随之不断改进。英、法、德、意、希腊五国到2000年底推广热水器600万

平方米，比 1990 年增长 2 倍多。

太阳能热利用已经发展成为一项成熟、可靠的技术。目前已经开发出各种类型的太阳能热水系统，并根据不同的需要和全球气象条件进行有效的调整。

(1)自然循环、直接换热、紧凑式太阳能热水器

这种方式是太阳能热水器中出现最早，也是最简单的系统。适用于没有霜冻问题的低纬度地区，比如南欧国家、以色列和澳大利亚^[12]。这类热水器安装在平屋顶建筑上，储热水箱高于集热器，对建筑外观一定的影响，不设置循环泵，运行方式简单，无须专门的维护管理。

(2)强制循环、二次换热、分离式太阳能热水系统

在有霜冻问题的地区，如中北欧，多采用此种系统。集热器安装在建筑屋面上，储热水箱设置在房屋内部。集热器系统通过循环泵，以电力驱动传热介质(水/防冻液)在管路中循环，通过储热水箱中的换热器将集热系统获得的热量传递到水中，使水升温。该系统符合建筑美观与使用功能的要求，又适合于不同气候条件，在国外普通采用。

(3)集中太阳能供暖和供热水系统

以德国为代表的欧洲国家早在 20 世纪 80 年代就已经开始在工程中应用集中太阳能供暖和供热技术。在一些住宅小区中，将大型太阳能加热系统与社区热力网相连接，通过增大系统的规格，实现系统性能的提高及成本的降低。到目前为止，根据系统的蓄热能力已开发出当天蓄热、一周蓄热、季节蓄热三种主要系统，均采用强制循环、二次换热的运行方式。

3、未来发展趋势

随着可持续发展战略在世界范围内的实施，太阳能的开发利用将被推到新的高度。至本世纪中叶，世界范围内的能源问题、环境问题的最终解决将依靠可再生洁净能源特别是太阳能的开发利用。随着越来越多的国家和有识之士的重视，太阳能的利用技术也有望在短期内获得较大进展。太阳能建筑将得到普及，太阳能建筑集成已成为国际新的技术领域，将有无限广阔的前景。太阳能建筑不仅要求有高性能的太阳能部件，同时要求高效的功能材料和专用部件。如隔热材料、透光材料、储能材料、智能窗(变色玻璃)、透明隔热材料等，这些都是未来技术开发的内容。

1.3 选题依据及主要研究内容

在研究了国内外相关领域的发展现状及未来发展趋势后发现，智能化家庭住宅模式

将成为新的家庭住宅模式的发展趋势。太阳能热水器虽然已经非常普及，但是其智能化程度相对较低，原因是国内太阳能热水器主要以家庭为单元安装使用，功能简单，而且主要以手动或半自动实现对热水器的操作，给用户使用带来许多不便和安全隐患，因此提高太阳能热水系统的智能化程度，为建筑物整体设计太阳能集热系统，对建筑内所有用户集中提供热水的智能太阳能集热控制器的研究非常有意义。针对上述问题，本课题将对以下几个方面内容进行研究，研究的内容分为七章：

第一章介绍课题研究的背景及意义，以及太阳能热水系统的国内外发展现状和未来发展趋势，然后确定课题的研究内容；

第二章介绍太阳能集热系统的原理、构成以及控制系统的组成和它要实现的功能；

第三章进行主控制器各外围电路的硬件设计；

第四章进行从控制器的软硬件设计以及制定其与主控制器通信协议；

第五章进行主控制器软件平台的搭建及应用程序的编写；

第六章介绍了模糊 PID 控制器的设计及系统的实验仿真；

第七章对本论文工作做了总结和展望。

1.4 本章小结

本章首先介绍了课题研究的背景及意义，然后分析了太阳能热水系统的国内外发展现状和未来发展趋势，以及目前国内外太阳能热水系统的应用情况，最后确定了课题要研究的主要内容。

第 2 章 太阳能集热控制系统

2.1 太阳能热水器的原理

太阳能热水器就是利用温室原理，把太阳能转变为热能，并向水传递热量，从而获得热水的一种装置。所谓温室原理就是指由于对流、辐射损失减少，使热量聚积，温度逐渐升高的一种自然现象^[2]。具体过程是太阳光透过玻璃进入密封的集热器，大部分能量被集热器吸收，然后将热量传给冷水，冷水加热后重量变轻，自动流入水箱上部，水箱下部的冷水由于密度大而自动流入吸热体继续获得热量，周而复始，水箱内的全部冷水将被加热。

2.2 太阳能集热系统组成及工作原理

太阳能集热系统主要由控制器、集热器、集热水箱、供热水箱、循环泵、支架、电加热器、电磁阀、电动三通阀、控制阀、温度传感器、水位传感器等设备构成。系统通过太阳能集热器收集能量，集热器内的热水收集于供热水箱中为用户提供热水。当系统储存热水不能满足供热指标时，通过辅助电加热系统提高水温。系统具有供热水循环回路，启动回水循环泵可更换供水管道中的水。图 2.1 为太阳能集热系统的结构框图。

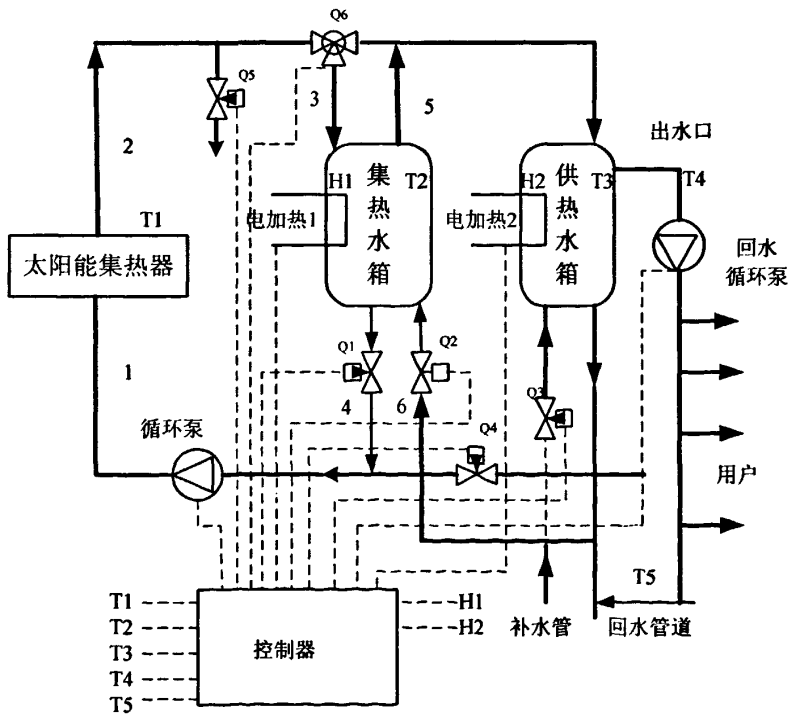


图 2.1 太阳能集热系统结构框图

太阳能集热器置于建筑物顶端，集热水箱和集热器中的水通过管路 1、2、3、4 以及控制阀 Q1 和三通阀 Q6 形成循环，使得集热器中被加热的热水循环至集热水箱中，经过一段时间，水箱中形成明显的温度分层，上部的水温已达到可供使用的水平，当集热水箱的上层水温高于 60℃ 时，打开控制阀 Q2，自来水经过管路 6 进入集热水箱底部，将其中的热水压出，通过管路 5 将加热的热水压送至供热水箱中。在运行过程中，传感器 T1、T2、T3、T4、T5 分别测量出集热器、集热水箱、供热水箱、出水口以及回水的温度，H1、H2 测得集热水箱和供热水箱中的水位，并传递给控制器，由控制器对系统的运行进行相应的控制。供热水箱为恒温水箱，将其热水温度控制在 60℃。控制阀 Q3 用于控制自来水的流量，避免供热水箱中的水温超过 60℃。当遇阴雨天气，供热水箱的水温低于 60℃ 时，电加热 2 开启，利用辅助能源加热供热水箱中的水。系统具有供热水循环回路，启动回水循环泵可更换供水管道不合供热指标的热水。供热水箱中的热水由出水口流向建筑内各用户的热热水调节入口，用户可以通过从控制器控制冷、热水阀门开度大小对水温进行控制。供热终端则有计量装置，计量每户热水的用量。由于各终端用水时间不尽相同，则总体用热量在各时间段可趋于自然平衡，太阳能集热器的热效率可充分发挥出来。图 2.2 为单用户的恒温调节器原理图。

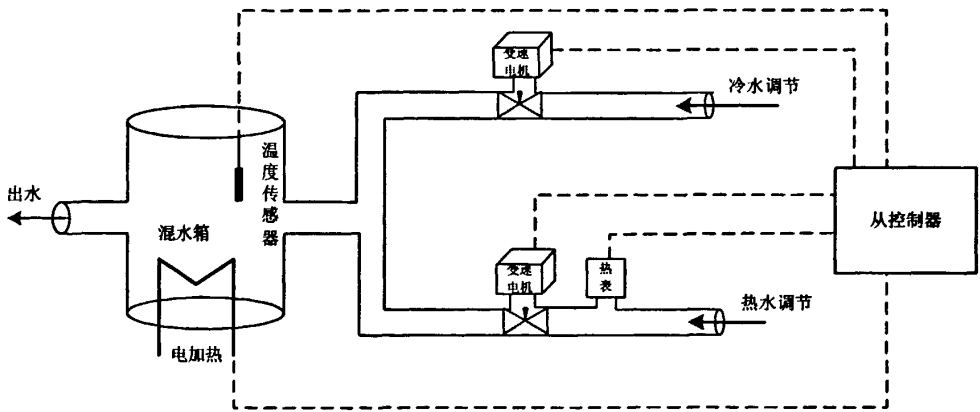


图 2.2 单用户恒温调节器原理图

2.3 太阳能集热系统的控制系统

太阳能集热系统的控制系统由主控制器与从控制器两部分组成。主控制器与太阳能热水器连接，负责与从控制器进行信息交互以及对热水器进行相应的控制。从控制器实现对单用户恒温调节器的控制、以及近程的显示、功能设置、报警等功能。主、从控制器间通信采取无线通信的方式。图 2.3 为控制系统框图。

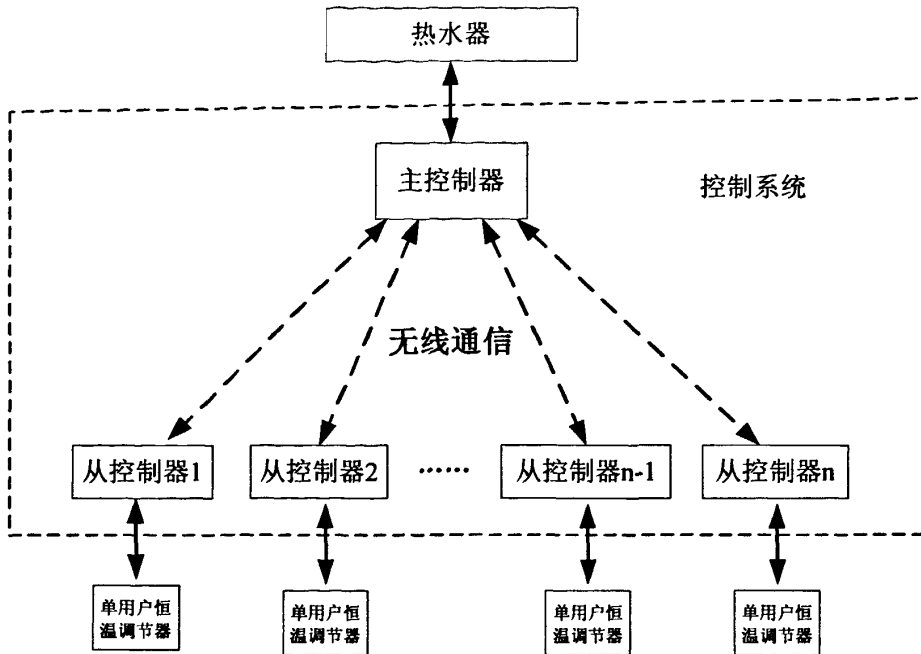


图 2.3 控制系统框图

2.4 本章小结

本章首先介绍了太阳能热水器的原理和太阳能集热系统的构成及工作原理，然后介绍了控制系统的组成和系统框图以及主、从控制器的作用。

第 3 章 主控制器硬件电路设计

3.1 主控制器硬件电路总体设计

太阳能集热系统主控制器的设计以三星公司生产的基于 ARM920T 内核的 S3C2410 微处理器为核心，并由电源电路、存储器电路、XE1205 无线传输电路、以太网接口电路、LCD(触摸屏)接口电路、键盘输入及报警电路、输出控制电路、复位电路、温度采集电路、水位检测及传送电路等外围电路组成。图 3.1 是主控制器硬件电路设计框图。

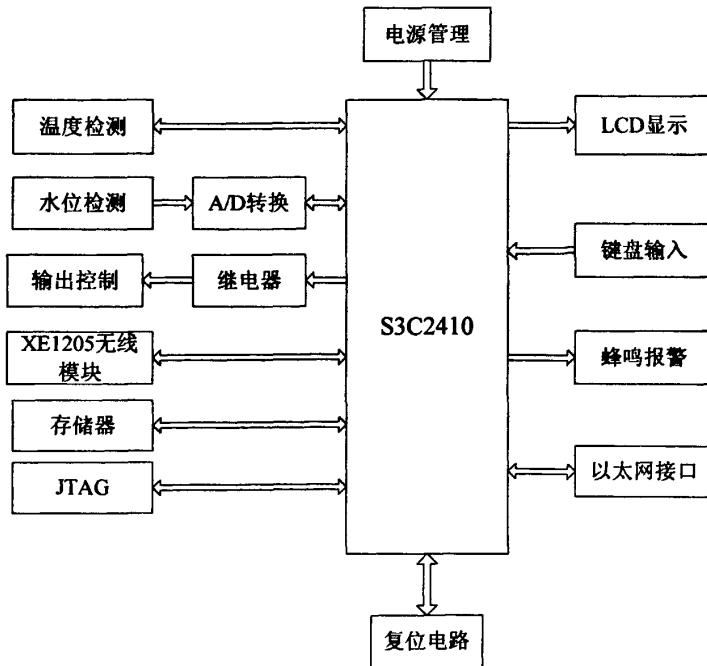


图 3.1 主控制器硬件电路框图

控制系统中 S3C2410 是系统核心，负责各种数据的处理；存储器用来存储系统程序代码以及各种数据；XE1205 无线模块实现与从控制器间数据的无线传输；温度采集采用 DS18B20，它检测到的信号为数字信号，可直接与处理器连接；水位检测把液位传感器采集到的电压信号经 TLC1549 变成数字信号后由 S3C2410 进行处理；键盘输入采用独立式按键设计，用来进行参数的设定；LCD（触摸屏）显示用来实现良好的人机互动以及各种数据的实时显示；以太网接口主要是实现系统的远程网络监控；输出控制通过继电器输出完成，主要包括电加热控制、上水控制、循环泵控制等。

3.2 S3C2410 芯片

主控制器采用 S3C2410 微处理器，其内核为 ARM920T，工作频率最高可达

203MHz，内部集成了丰富的模块，主要特点如下^[13]：

- 1 个 LCD 控制器（支持 STN 和 TFT 带有触摸屏的液晶显示器）
- SDRAM 控制器
- 3 个通道的 UART
- 4 个通道的 DMA
- 4 个具有 PWM 功能的计时器和 1 个内部时钟
- 8 通道的 10 位 ADC
- 触摸屏接口
- I²C 总线接口
- 1 个 USB 主机接口，1 个 USB 设备接口
- 2 个 SPI 接口
- SD 接口和 MMC 卡接口
- 117 位通用 I/O 口和 24 位外部中断源

在时钟方面，S3C2410 芯片集成了一个具有日历功能的 RTC 和具有 PLL(MPLL 和 UPLL)的芯片时钟发生器。MPLL 产生主时钟，能够使处理器工作频率最高达到 203MHz，这个工作频率能够使处理器轻松运行于 WinCE、Linux 等操作系统以及进行较复杂的信息处理；UPLL 产生实现主从 USB 功能的时钟。

3.3 电源电路设计

一般的变压器输出的电流杂波相当多，质量差，需要经过电源稳压芯片稳压后得到波形较好的电源后才可使用。

本系统所需 5V 电源采用降压开关型集成稳压电路 LM2576 芯片，它使用简单，只需极少的外部元件，且包含内置频率补偿和一个固定频率的晶振。LM2576 的输出开关不仅包括逐周期电流限制，在故障发生时还采用热关断为器件提供充分的保护。图 3.2 为它的电路设计原理图。

系统所需 3.3 伏电源采用低压降线性稳压器 LM1117-33 芯片将 5V 电压转化为 3.3V 电压，电路设计原理图如 3.3 所示。

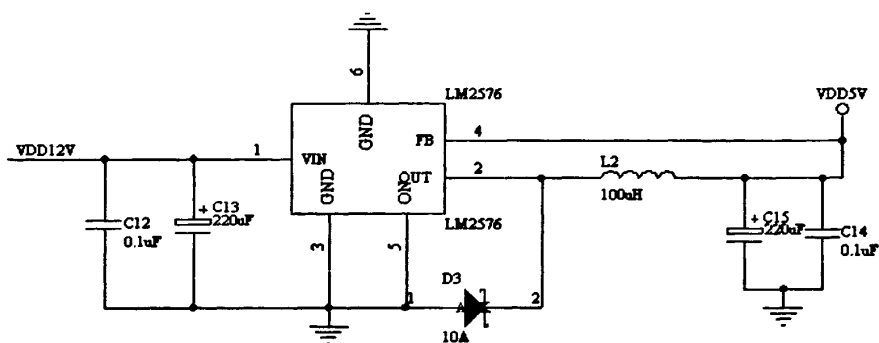


图 3.2 5V 电源电路原理图

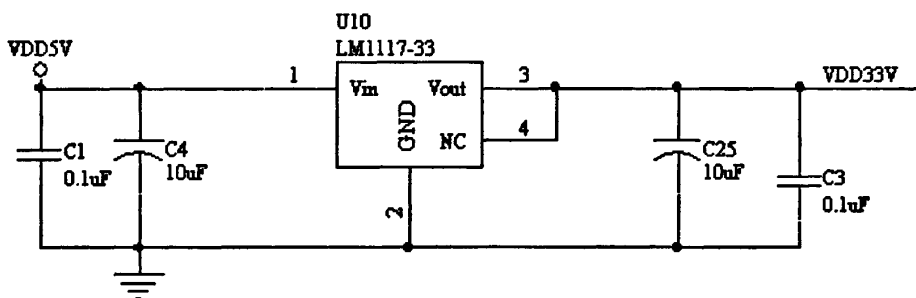


图 3.3 3.3V 电源电路原理图

3.4 存储器电路设计

主控制器的存储器电路由两部分组成：SDRAM和Nand Flash，采用NAND Flash与SDRAM组合的方式，可以获得较低的性价比。NAND Flash在系统中用于存放程序代码、常量表以及一些在系统掉电后需要保存的用户数据等。SDRAM主要用来存放执行代码和变量，是系统启动后主要进行存取操作的存储器。

3.4.1 NAND Flash 接口电路设计

Flash 存储器是一种可在系统（In-System）进行电擦写，掉电后信息不丢失的存储器。它具有低功耗、大容量、擦写速度快、可整片或分扇区在系统编程、擦除等特点。

现在市场上两种主要的非易失性闪存技术是 NOR 和 NAND^[14]。NOR 的特点是芯片内执行，这样应用程序可以直接在 Flash 内运行，不必再把代码读到系统 RAM 中。NOR 的传输效率很高，在 1~4MB 的小容量时具有很高的成本效益，但是很低的写入和擦除速度大大影响了它的性能。NAND 结构能提供极高的单元密度，可以达到高存储密度，并且写入和擦除的速度快。

主控制器设计采用 K9F1208U0C^[15] NAND Flash 芯片，共有 64MB 的存储空间，能

够满足启动代码、操作系统 Linux 内核和文件系统映像的存储需要，图 3.4 为 K9F1208U0C 与 S3C2410 的电路连接图。S3C2410 的 NANDFlash 控制器与外部的接口信号如下所示。

- nFCE: NANDFlash 片选
- nFRE: NANDFlash 读使能
- nFWE: NANDFlash 写使能
- D[7:0]: 数据/命令/地址输入输出端口
- ALE: 地址锁存使能
- CLE: 命令锁存使能

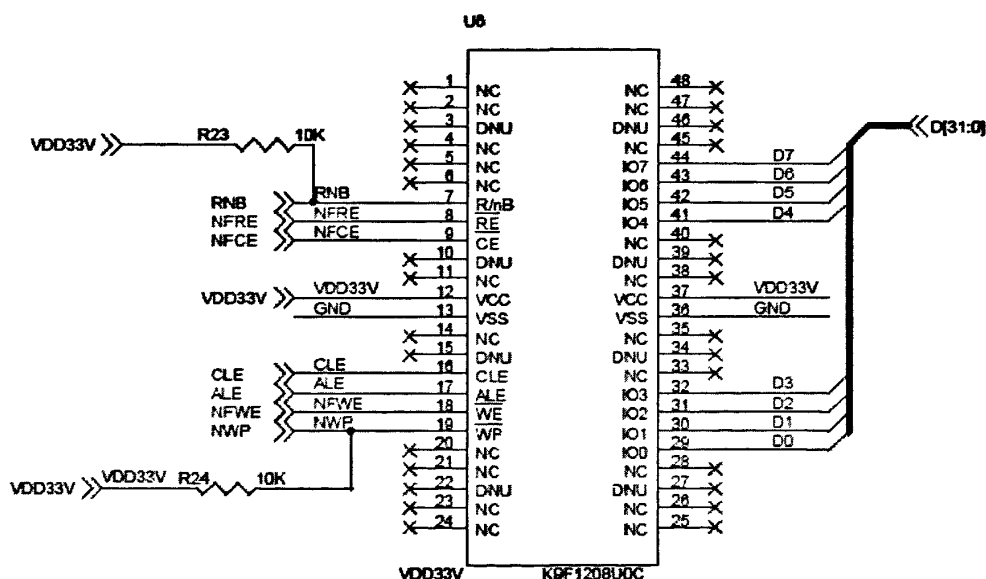


图 3.4 K9F1208 与 S3C2410 的连接电路

3.4.2 SDRAM 接口电路设计

SDRAM 具有容量大、存取速度快、成本低的特点，因而应用比较广泛。S3C2410 芯片带有 SDRAM 刷新控制逻辑，可方便地与 SDRAM 接口。电路设计中 SDRAM 由两片 K4S561632C-TC75 芯片^[16]构建 32 位的 SDRAM 存储系统。该芯片外部同步时钟速率可在一定的频率范围内连续变化，最高频率可达到 133MHz，每块 SDRAM 内含四个独立的 Bank，它的基本存储单元都是按照阵列排列的，数据位宽和整个存储器的位宽相同，所有的输入信号均以时钟的上升沿为基准，这使得地址、控制和数据输入到缓冲器的时间可保持一致且建立和保持的时间很小。另外，它还具有突发长度可编程、延

- 编码方式：NSR；
- 接收时，有高IIP3 和高灵敏度两种选项（A、B 模式）；
- 发射电流62mA@15dBm；
- 接收电流14mA；
- 休眠电流0.2uA；
- 传输速率可以达到152.3kbit/s；
- 输入输出驻波 ≤ 1.5 ；
- 收发状态均有DCLK；
- 调制方式：2-FSK；
- 通信距离长：市区环境1500m，开阔地3000m。

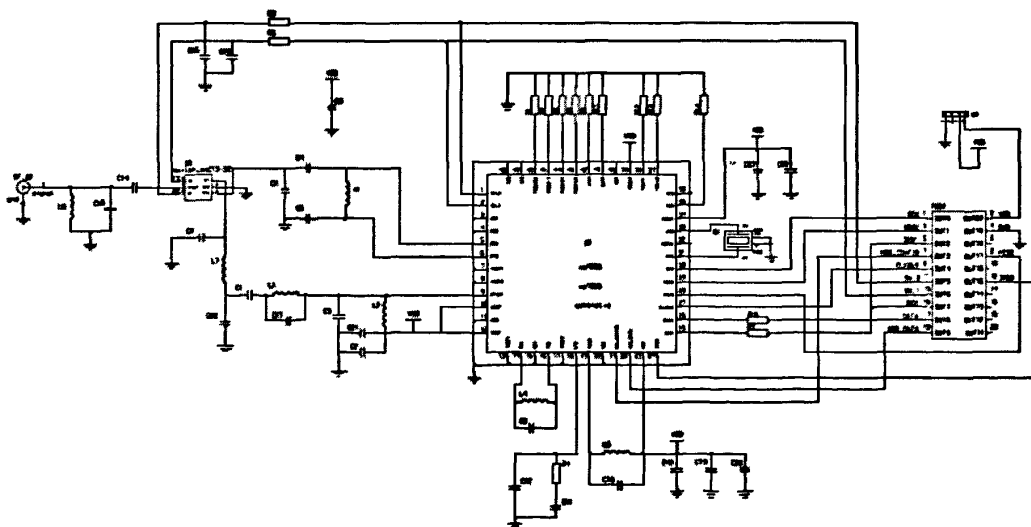


图 3.6 XE1205 无线模块电路原理图

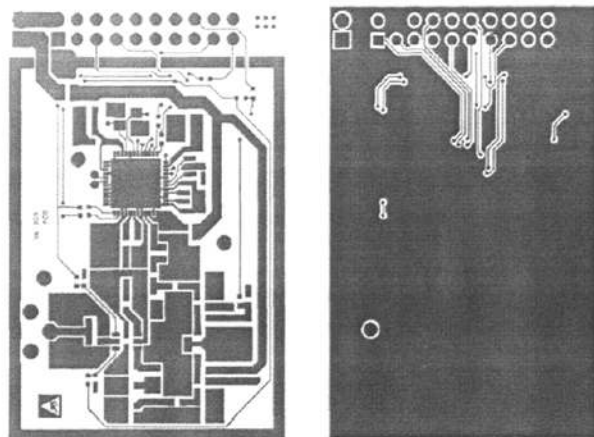


图 3.7 XE1205 无线模块电路 PCB 图

3.6 以太网接口电路设计

为控制器引入网络功能,通过网络传输和网络接口实现远程控制已经成为一种发展趋势。以太网接口控制器主要包括 MAC 层和 PHY 层两部分, S3C2410 内没有集成 MAC 层和 PHY 层控制器,所以要使用带有 MAC 层和 PHY 层的控制器以太网芯片 CS8900A^[17]扩展。

CS8900A 是 CIRRUS LOGIC 公司生产的 16 位以太网控制芯片^[18]。它支持 ISA 总线标准,提供 10Mbps 的以太网接入带宽,支持全双工通信,内部集成了支持 IEEE 802.3 协议标准的 MAC 引擎,负责处理有关以太网数据帧的发送和接收,包括:冲突检测、帧头的产生和检测, CRC 校验码的生成和验证。通过硬件实现 MAC 层协议,使得硬件平台能够节约更多的 CPU 资源。

它的工作原理是:收到由主机发来的数据报(从目的地域到数据域)后,侦听网络线路。如果线路忙,就等到线路空闲为止,否则立即发送该数据帧。发送过程中,首先添加以太网帧头(包括先导字段和帧开始标志),然后生成 CRC 校验码,最后将此数据帧发送到以太网上。接收时,它将从以太网收到的数据帧再经过解码、去掉帧头和地址检验等步骤后缓存在片内。通过 CRC 校验后,它会根据初始化配置情况,通知主机 CS8900 收到了数据帧。

CS8900A 有两种工作模式可以选择:内存模式和 I/O 模式,系统启动后默认是 I/O 工作模式,默认的地址是 0x300,本系统选择 I/O 模式。CS8900A 的初始化由 SBHE 引脚参与,初始化时在该引脚上产生一个由高到低,然后由低到高的电平,完成以后就可

以对芯片进行读写操作。正常工作时，以字为单位寻址，数据线为SD[15:0]。CS8900A的片选信号连接S3C2410A的nGCS3引脚，复位后的映射存储区是Bank3，S3C2410X的地址线A[24:0]中的A[19:0]与CS8900A的地址线连接，A24参与片选，所以CS8900A占用CPU的地址为0x19000000。

在I/O模式下，CS8900A只有6个寄存器可以直接寻址访问，对这些寄存器采用直接寻址方式来实现对这些寄存器的直接寻址。地址的计算公式为：片选地址(0x19000000)+I/O基址(0x300)+偏移地址。

CS8900A的片内寄存器可分为4类：总线接口寄存器组、状态与控制寄存器组、发送初始化寄存器组和地址过滤寄存器组。这些寄存器需要进行间接寻址，将这些寄存器地址放入片内寄存器指针中，即可实现网络数据收发的功能。CS8900A与S3C2410电路连接如图3.8所示。

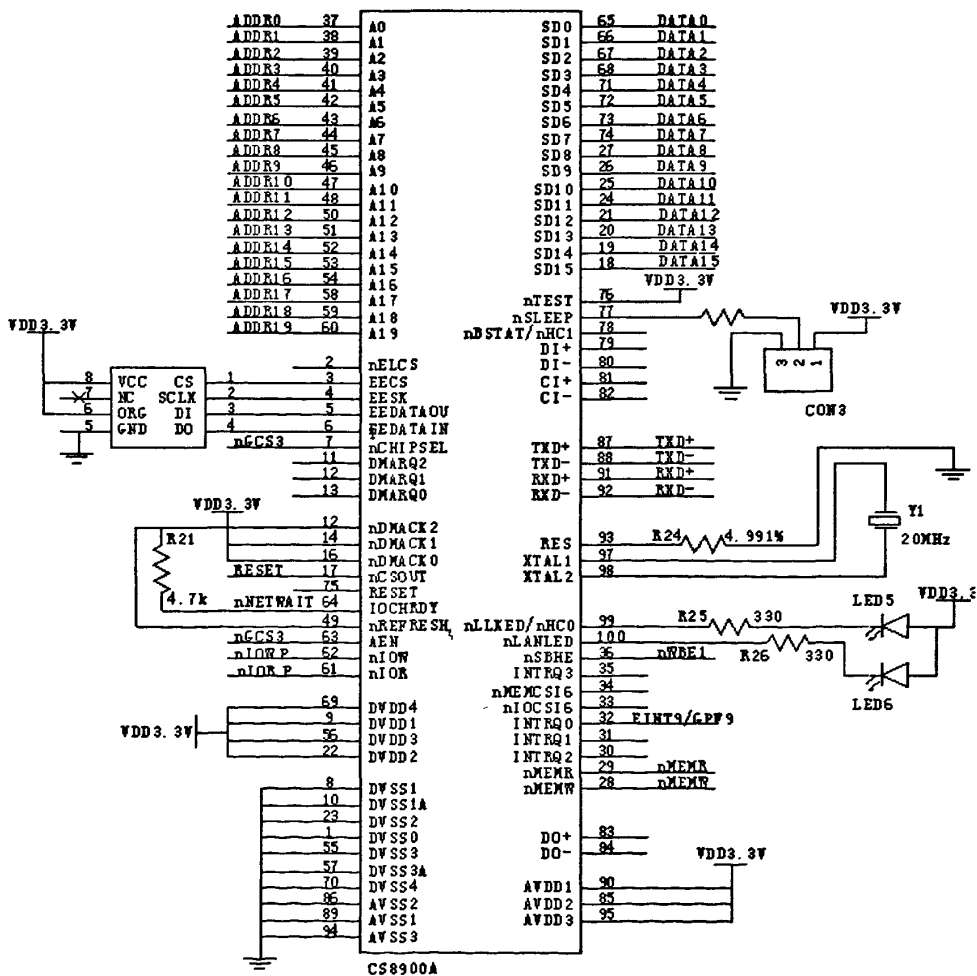


图 3.8 CS8900A 与 S3C2410 电路连接图

3.7 LCD（触摸屏）接口电路设计

S3C2410支持TFT/STN型^[42]的LCD及触摸屏，本设计中引出了S3C2410内部LCD 相关的信号线，通过41pin的插座作为LCD与触摸屏接口，来连接外部LCD 模块。本系统采用三星公司的LTV350QV_F04触摸屏，320RGBx240像素，16.7M色，TMR技术，内置驱动IC(S6F2002)，开发简单。采用白光二极管背光，功耗很低。图3.9为LCD接口电路原理图。

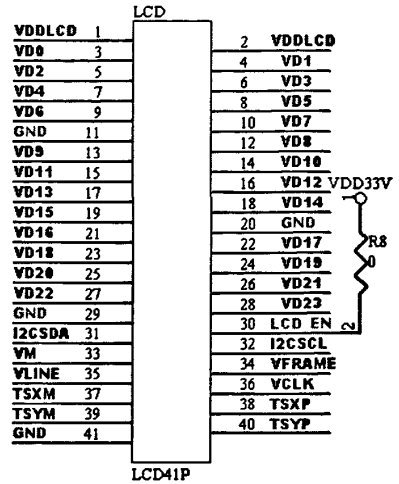


图 3.9 LCD 接口电路原理图

3.8 键盘输入及报警电路设计

键盘设计采用独立式接法，共设计了功能键、数字增加键、数字减少键、确定键、取消键和手动键 6 个按键。当有按键被按下时，会实时产生中断请求信号，通知处理器处理。报警电路采用蜂鸣器设计。电路原理图分别如图 3.10，图 3.11 所示。

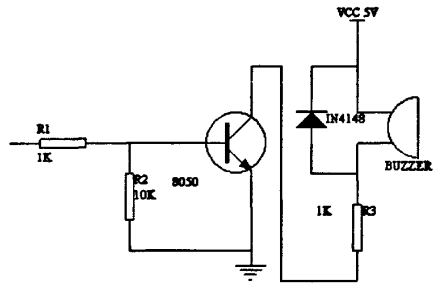


图 3.11 蜂鸣器报警电路图

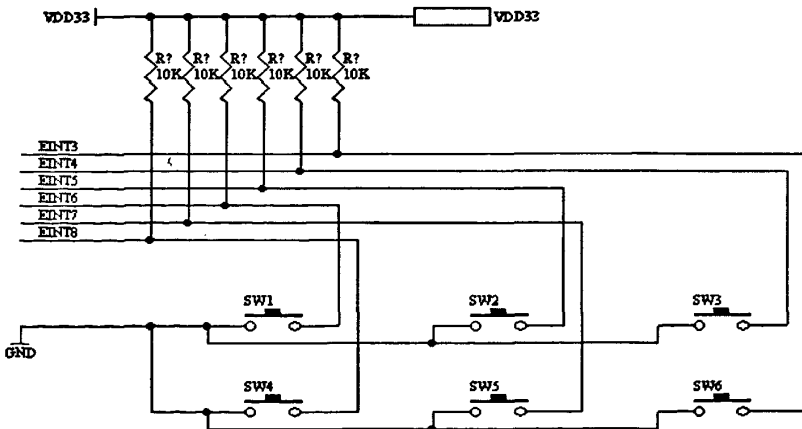


图 3.10 键盘输入电路图

3.9 复位电路设计

复位对于一个系统来说很重要，由于各个单元要进入正常工作状态，需要可靠地复位，正常情况下，一般有上电复位和手动复位。如果电源电压出现波动，系统会非正常复位，这时候会发生复位时间不够从而造成一些错误甚至死机，所以复位监控电路也是有必要的。复位电路原理如图 3.12 所示。

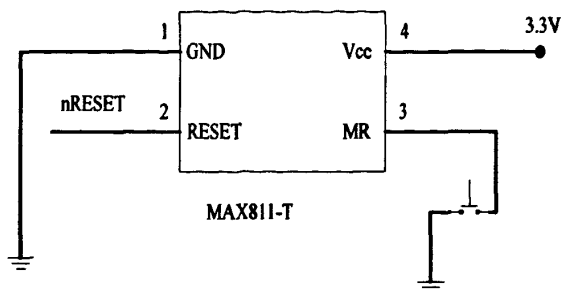


图 3.12 复位电路原理图

3.10 输出控制电路设计

在控制器的输出控制中，通过 GPIO 口，由光电耦合器件对信号隔离后，对继电器进行控制，从而打开或关闭电加热器、循环泵、上水泵等。控制电路如图 3.13 所示。

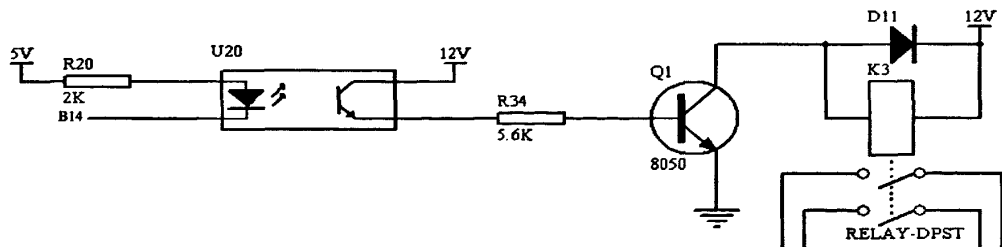


图 3.13 输出控制电路原理图

3.11 温度采集电路设计

目前广泛使用 PT100 热电阻作为热水器中温度传感器。采用这种方式，需使用高性能直流放大器和 11 位以上的 A/D 转换器保证温度测量的精度和稳定性，器件的价格较高，电路复杂。我们选用 DS18B20 温度传感器。它输出信号为数字信号，只需通过一根线与 S3C2410 相连。每个 DS18B20 都有一个独特的片序列号，可以实现总线方式的连接，连接方式如图 3.14 所示。

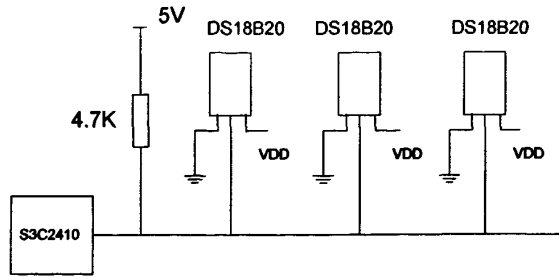


图 3.14 DS18B20 与 S3C2410 连接图

3.12 水位采集电路设计

水位采集电路如图 3.15 所示,其中 I 为恒流源,电流经过检测电阻产生一定的压降。当 K1、K2,...K8 均打开时, $V_0=8IR$ 。现取 $IR=0.5V$, 则 $V_0=4V$ 。当水位上升至 K_n 处时,浮子将推动磁体 M 至于簧管 K_n 处, K_n 接通。经过 OP 缓冲输出的电压 $V_0=0.5(n-1)V$ 。输出端经 TLC1549 转换后输入处理器中进行相应的处理。

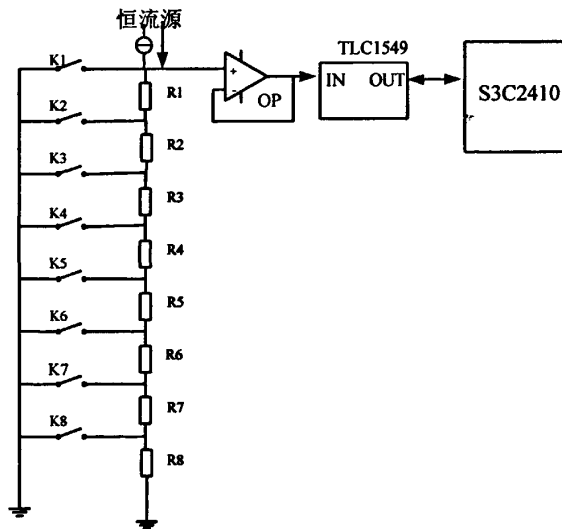


图 3.15 水位采集电路图

3.13 本章小结

本章首先介绍了主控制器硬件系统结构图,然后具体介绍了 S3C2410 芯片、电源电路、存储器接口电路、XE1205 无线传输电路、以太网接口电路、LCD (触摸屏) 接口电路、键盘输入及报警电路、复位电路、输出控制电路以及温度采集电路、水位检测及传送电路等外围电路设计。

第 4 章 从控制器的设计

4.1 从控制器硬件电路总体设计

从控制器的硬件设计主要包括电源电路、XE1205 无线传输电路、存储器电路、水温检测电路、液晶显示电路、键盘输入电路、报警电路、输出控制电路、遥控接收电路等。

从控制器与主控制器间进行通信可以采取有线或者无线的方式。由于有线方式扩展性能差,布线繁琐,影响美观,本文采取无线通信的方式。从控制器 CPU 选用 STC89C52 芯片,射频芯片选用 XE1205 芯片,液晶屏选用 12864 液晶,存储器选用 AT24C02,温度采集选用 DS18B20 实现。图 4.1 为从控制器的硬件结构图。

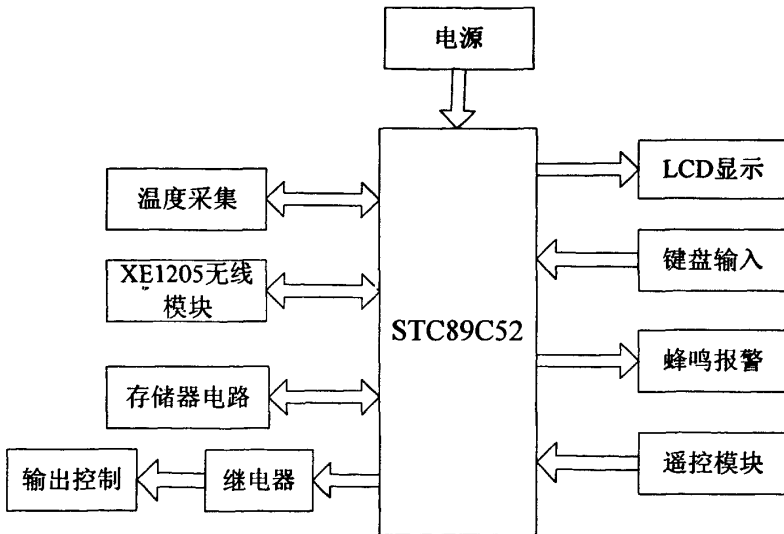


图 4.1 从控制器硬件结构图

4.2 部分外围电路设计

从控制器需要 5V 和 3.3V 电源,电源电路设计方法和主控制器的一样。5V 电源采用 LM2576 稳压芯片将变压器输出的 12V 电源稳压至 5V; XE1205 无线模块所需 3.3 伏电源采用低压降线性稳压器 LM1117-33 芯片将 5V 电压转化为 3.3V 电压。

温度传感器选用 DS18B20 温度传感器,其体积小,接口方便,传输距离远,测温范围为 -55°C — $+125^{\circ}\text{C}$,测量分辨率为 0.0625°C 。

键盘设计采用独立式按键接法,共设计了功能键、数字增加键、数字减少键、确定键、取消键和手动键 6 个按键。报警电路采用蜂鸣器设计。LCD 显示选用 12864 液晶显示屏。

4.2.1 STC89C52

从控制器微处理器选用 STC89C52，它是宏晶科技推出的一款 8 位单片机，它包含中央处理器、程序存储器(ROM)、数据存储器(RAM)、定时/计数器、并行接口、串行接口和中断系统等几大单元及数据总线、地址总线和控制总线三大总线。图 4.2 为其管脚图，其主要性能特点如下：

- 8 位CPU
- 8K bytes 程序存储器(ROM)
- 128 bytes 的数据存储器(RAM)
- 32 条I/O 口线
- 111 条指令，大部分为单字节指令
- 21个专用寄存器
- 2个可编程定时/计数器
- 5个中断源，2个优先级
- 一个全双工串行通信口
- 外部数据存储器寻址空间为64kB
- 外部程序存储器寻址空间为64kB
- 逻辑操作位寻址功能
- 单一+5V 电源供电

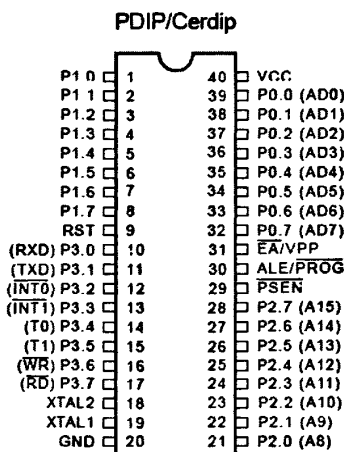


图4.2 STC89C52管脚图

4.2.2 XE1205 无线传输电路

配合主控制器进行数据无线传输，射频芯片同样采用 XE1205 芯片，其与单片机连接电路图如图 4.3 所示，图 4.4 为无线模块实物图及其与单片机的连接图。

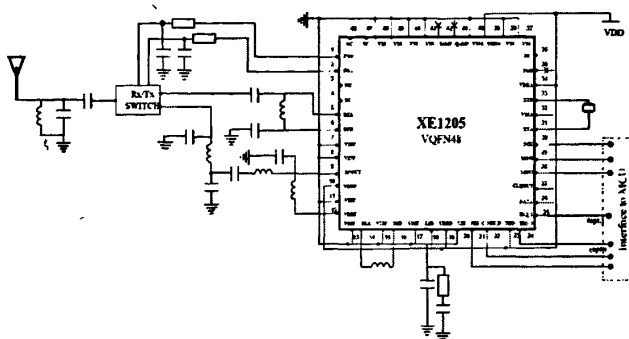


图 4.3 XE1205 无线传输电路原理图

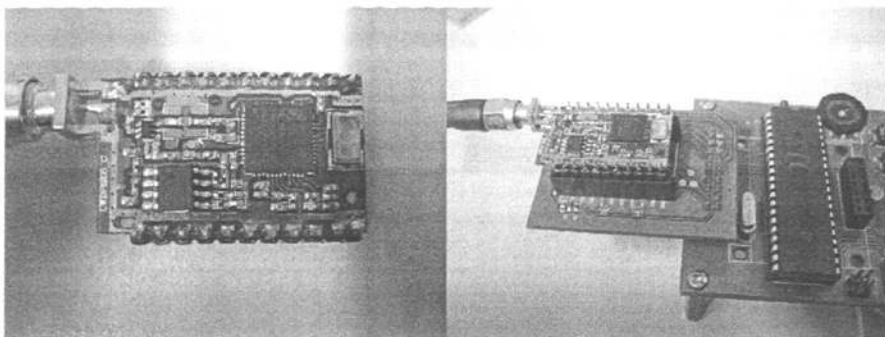


图 4.4 XE1205 无线模块实物图及与单片机连接图

4.2.3 存储器电路

从控制器存储器选用 AT24C02，它是美国 ATMEL 公司的低功耗 CMOS 串行 E²PROM，内部含有 256 个 8 位字节，具有宽电压、擦写次数多、写入速度快等特点。AT24C02 支持 I²C 总线，它通过 SDA 及 SCL 两根线在连到总线上传递信息。电路连接图如图 4.5 所示。

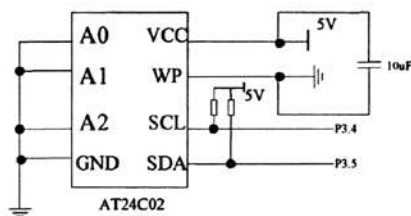


图 4.5 AT24C02 电路原理图

4.2.4 输出控制电路

在从控制器中，通过控制电机正反转实现对冷、热水阀门开度的控制，实现水温的自动控制，电机的控制电路如图 4.6 所示。

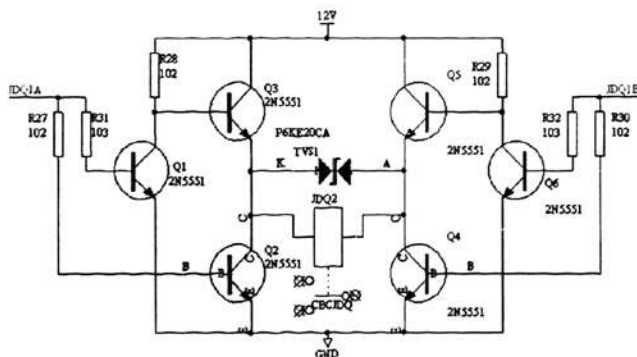


图 4.6 电机控制电路图

4.2.5 遥控接收电路设计

遥控发射电路采用 PT2262 实现编码，接收电路配合采用 PT2272 解码，实现数据无线传输。

PT2262/2272 是台湾普城公司生产的一种 CMOS 工艺制造的低功耗低价位通用编解码电路,PT2262/2272 最多可有 12 位(A₀-A₁₁)三态地址端管脚,任意组合可提供 531441 地址码,PT2262 最多可有 6 位(D₀-D₅)数据端管脚,设定的地址码和数据码从 17 脚串行输出,可用于无线遥控发射电路。编码芯片 PT2262 发出的编码信号由: 地址码、数据码、同步码组成一个完整的码字,解码芯片 PT2272 接收到信号后,其地址码经过两次比较核对后,VT 脚才输出高电平,与此同时相应的数据脚也输出高电平。遥控发射器市场上可以买到,遥控接收电路原理图如图 4.7 所示。

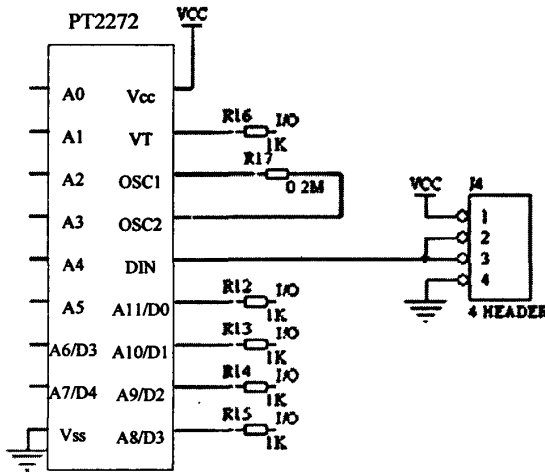


图 4.7 遥控接收电路原理图

4.3 无线通信程序编写

无线模块的软件编程通过 SPI 串行通信进行,只需严格按照数据手册上的时序即可。无线通信程序的编写主要包括寄存器读写程序、无线数据发送程序、无线数据接收程序等。

对 XE1205 的内部寄存器的读写都是通过 SPI 三总线串行通信完成,编程时首先使能芯片,然后送出地址和参数,最后发送芯片使能结束命令。常需要通过编程用户自定义配置寄存器设定以下参数:频率范围、功率、速率、带宽、频偏等。

发送数据是通过 DATA 引脚发送的。发送数据时,首先使 XE1205 处于发送模式,DATA 引脚上的电平被调制发送出去,XE1205 内置位同步器,能够自动地对位误差进

行校准，所以每一位发送数据的定时时间不必非常准确。在发送有用数据之前要发送一组 4 字节的前导码(0xAA)和一组 32 位的模式识别码。发送结束后，把 XE1205 设置到接收模式或者待机模式。

接收数据时，模式识别模块从接收到的数据流中查找同预先编程设定的前导码和模式识别码串进行比较，只有当正确地接收到前导码和 32 位的模式数据识别码以后才输出 PATTERN 信号向 CPU 发出中断申请，表明接下来收到的数据都是有用数据，发往单片机的数据是通过 SCK 和 DATA 引脚配合的，DATA 信号在 SCK 的上升沿到达后有效。单片机在收到有效数据帧以后关闭接收数据功能,直到下次 PATTERN 信号再次打开接收数据功能。(注：部分程序源代码见附录)

4.4 从控制器与主控制器间无线通信过程

主控制器与从控制器间构成一个主从式多机无线传感器网络，采取半双工的通信方式。主控制器必须为每个从控制器分配不同的地址，主控制器通过发送地址与从控制器取得联络后再进行数据的交互。通信的主要过程如下：

主控制器通过发送命令帧启动一次通信，从控制器收到命令帧后，判断呼叫地址是否和本机相同，如不同则丢弃该数据，不予理睬；若相同则根据提取出命令帧中的命令字，进行协议解析之后，按要求发送主控制器所要的数据。若主控制器在一定的时间内没有收到数据，则进行此命令的重发(重发不能超过三次)，如果超过三次，从控制器仍然无应答，主控制器设置蜂鸣报警，提示通信出错并跳过此节点继续执行。

同样，从控制器通过发送命令请求帧给主控制器启动一次通信，若在一定的时间内从控制器没有收到主控制器返回的应答帧，则进行此命令的重发。当收到主控制器的应答帧后，开始发送数据或者命令。数据传输协议的基本单元为数据帧，数据帧格式如表 4.1 所示。

表 4.1 数据帧格式

名 称	代 码
帧起始符	68H
地址域	A0
	A1
控制码	C
数据长度域	L
数据域	DATA _c

校验码	CS
结束符	16H

4.5 从控制器功能

从控制器要实现的功能是：

- (1) 控制功能：通过控制冷、热水阀门开度的大小，实现水温的自动控制；
- (2) 显示功能：户内混水箱水温、回水温度显示，以及供热水箱中水温显示、累计热量显示、电加热工作指示等；
- (3) 设置功能：混水箱中水温设置、电加热设置等；
- (4) 报警功能：如果控制器出现异常情况进行报警。

图 4.8 是从控制器的控制面板图

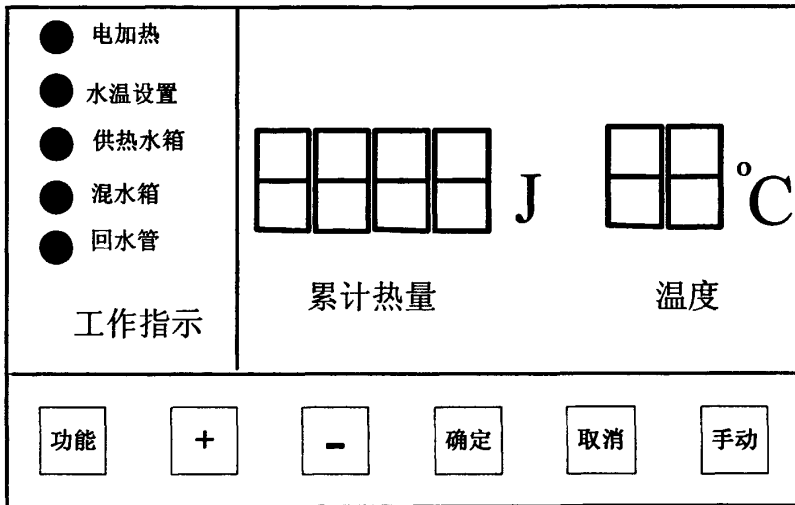


图 4.8 从控制器控制面板图

4.6 本章小结

本章首先介绍了从控制器硬件结构框图和外围电路设计，然后编写无线通信程序，并介绍从控制器与主控制器间如何进行通信，最后介绍从控制器要实现的能及其控制面板图。

第 5 章 主控制器软件平台的建立及应用程序编写

主控制器软件平台的搭建工作主要由以下四部分工作：交叉编译环境的建立、系统引导程序 Bootloader 的移植、Linux 内核移植和文件系统的建立。交叉编译器是运行在 PC 机 (Host) 上的软件，能够编译生成运行在目标机(target)上的二进制文件。系统引导程序 Bootloader 相当于 PC 机上的 BIOS，负责系统的初始化工作和将操作系统内核固化到 Flash 中，然后将控制权交给 Linux 操作系统。Linux 内核是嵌入式系统运行后的管理平台，负责实时性任务和多任务的管理。文件系统是嵌入式系统软件平台占用存储量最大的一部分，也是与用户开发最相关的一部分，它存储了系统配置文件、系统程序、用户应用程序和必需的驱动程序。

当系统硬件电路和软件平台确定后，系统的主要功能要靠应用软件来实现，软件编程采取模块化编程思想，即将一个大的程序分成若干小程序模块，各个模块保持相对独立性。这样使得程序的调试、修改和维护比较容易。

5.1 交叉编译

简单地说，交叉编译就是在一个平台上生成另一个平台上的可执行代码。嵌入式开发的程序编写和编译是在 PC 机完成，但编译出来的结果要在嵌入式目标板上运行。通常将这种在主机上开发编译，在目标板上调试运行的开发模式称为交叉开发^[19]。同样，由主机上的编译器编译产生标板上运行的可执行程序的编译过程称为交叉编译。

交叉编译环境建立的过程如下：

- (1) 创建编译环境。在这个过程中，将设置一些环境变量，创建安装目录，安装内核源代码和头文件等。
- (2) 创建 binutils。这个过程结束后，会创建类似 arm-linux-ld 等工具。
- (3) 创建一个交叉编译版本的 gcc。这个过程中，只能编译 C 程序，而不能编译 C++ 程序。
- (4) 创建一个交叉编译版本的 glibc。
- (5) 创建一个交叉编译版本的 gdb。
- (6) 重新创建 gcc。前面创建 gcc 的过程没有编译 C++ 编译器，现在 glibc 已经准备好了，所以这个步骤将完善 gcc 交叉编译。
- (7) 重新创建 glibc。如果成功执行这个过程，那么将建立属于自己的交叉编译工具链。

5.2 Bootloader

Bootloader（引导加载程序）是系统加电后运行的第一段代码。系统在上电复位时通常从地址 0x00000000 处开始执行，而在这个地址处安排的通常就是系统的 Bootloader。通过这段程序可以初始化硬件设备、建立内存空间的映射图、从而将系统的软硬件环境带到一个合适的状态，以便为最终调用操作系统内核准备好正确的环境。

Bootloader 启动大多数都分为两个阶段^[20]。第一阶段主要包含依赖于 CPU 的体系结构硬件初始化的代码，通常都是用汇编语言来实现。这个阶段的主要任务有：

- 基本的硬件设备初始化（屏蔽所有的中断、关闭处理器内部指令/数据 cache 等）；
- 为第二阶段准备 RAM 空间；
- 如果是从某个固态存储介质中，则复制 Bootloader 的第二阶段代码到 RAM；
- 设置堆栈；
- 跳转到第二阶段的 C 程序入口点。

第二阶段通常用 C 语言完成，以便实现更复杂的功能，也使程序有更好的可读性和可移植性。这个阶段的任务有：

- 初始化本阶段要用到的硬件设备；
- 检测系统内存映射；
- 将内核映像和根文件系统映像从 Flash 读到 RAM；
- 为内核设置启动参数；
- 调用内核。

Bootloader 调用 Linux 内核的方法时直接跳转到内核的第一条指令处，即直接跳转到 MEM+START+0x8000 地址处。在跳转时必须满足下列条件。

(1) CPU 寄存器的设置

- R0 为 0；
- R1 为机器类型 ID；
- R2 为启动参数，标记列表在 RAM 中的起始基地址。

(2) CPU 模式

- 必须禁止中断（IRQs 和 FIQs）；
- CPU 设置为 SVC 模式。

(3) Cache 和 MMU 的设置

- MMU 必须关闭；

- 指令 Cache 可以打开也可以关闭;
- 数据 Cache 必须关闭。

可用下列代码来调用内核:

```
void (*the Kernel)(int zero, int arch, u32 params_addr)=(void*)(int, int,
u32))KERNEL_RAM_BASE;
    .
    .
    .
the Kernel(0, ARCH_NUMBER, (u32)kernel_params_start)
```

本系统中 Bootloader 采用韩国 Mizi 公司开发的 vivi, 它适用与 ARM9 处理器。vivi 具有两种启动模式, 启动加载模式可以在一段时间后(这个时间可更改)自行启动 Linux 内核, 这是 vivi 的默认模式^[21]。

5.2.1 vivi 的移植

vivi 提供了基于 S3C2410 处理器开发板的基本设置, 所以对 vivi 的移植比较简单, 移植过程如下:

(1) 设置编译环境:

```
ARCH:=arm
LINUX_INCLUDE_DIR = /friendly-arm/kernel/include/
CROSS_COMPILE = /usr/local/arm/2.95.3/bin/arm-Linux-
```

(2) 设置 NANDFlash 分区, 修改 vivi/arch/S3C2410/smdk.c, 将相应部分改为:

```
mtd_partition_t default_mtd_partitions[] = {
{
    name:      "vivi",
    offset:    0,
    size:      0x00020000,      //128K
    flag:      0
}, {
    name:      "param",
    offset:    0x00020000,
    size:      0x00010000,      //64K
    flag:      0
```

```

    }, {
        name:      "kernel",
        offset:    0x00030000,
        size:      0x00100000,      //1M
        flag:      0
    }, {
        name:      "root",
        offset:    0x00130000,
        size:      0x03ec0000,      //
        flag:      MF_BONFS
    }
};

```

(3) vivi 启动后使用 `param set` 命令设置修改 Linux 内核启动命令行。char `Linux_cmd[] = "noinitrd root=/dev/mtdblock/0 init=/Linuxrc console=ttyS0,115200"`;
`root=/mtdblock/0` 代表 NANDFlash 的第 1 个分区，它是系统的 rootfs，`console=ttyS0, 115200` 使 kernel 启动期间的信息全部输出到串口 0 上。最后配置 vivi，使用的命令：

```
$make distclean
```

```
$make menuconfig
```

配置完成后进行编译，在当前目录下生成 vivi 二进制映像文件，通过 JTAG 利用 `sjf2410` 工具进行代码的烧写。成功后，重启系统，vivi 从串口终端显示启动信息如图 5.1 所示，启动信息里有 MMU 表基地址以及系统的启动方式等。



图 5.1 vivi 启动显示信息

5.3 Linux 内核的分析与移植

5.3.1 内核分析

内核 (kernel) 是操作系统的内部核心程序, 它提供了对计算机设备的内核管理调用功能^[22]。Linux 系统支持多个进程的并发运行, 每个进程都请求系统资源, 比如运算、内存、网络连接或其他一些资源等^[23], 内核负责处理所有这些请求。图 5.2 为 Linux 内核框架图, 根据内核完成任务的不同, 可将内核功能分为如下几部分^[24]:

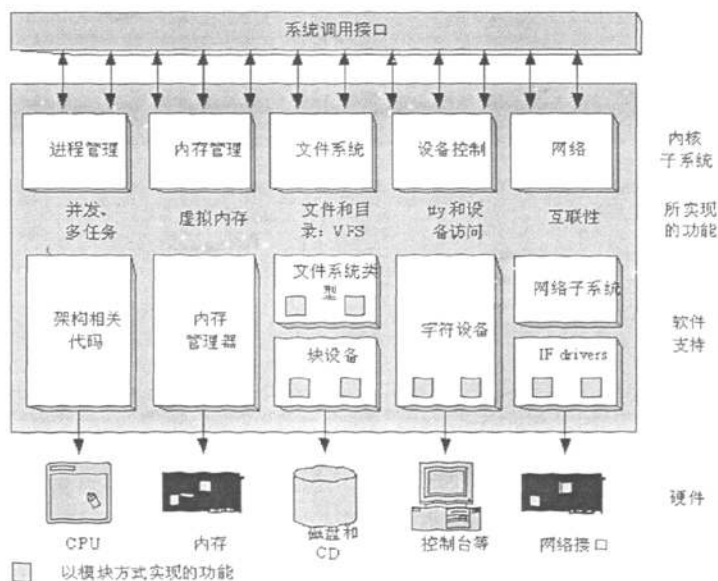


图 5.2 Linux 内核框架图

- 进程管理：进程管理功能负责创建和销毁进程，并处理他们和外部世界之间的连接（输入和输出）。不同进程之间的通信（通过信号、管道或进程间通信原语）是整个系统的基本功能。除此之外，控制进程如何共享 CPU 的调度器也是进程管理的一部分。概括来说，内核进程管理活动就是在单个或多个 CPU 上实现了多个进程的抽象。对于非实时系统，进程调度的主要策略需要对每个进程都做到公平，防止死锁，饿死的可能性。对于实时系统，这个一般需要提供一个内核调度机制让用户根据需要来选择调度算法。这个部分在 ARM 平台上的移植的主要工作就是进程上下文切换时其上下文中和体系结构相关的部分保存和恢复。
- 内存管理：内存管理是计算机的主要资源之一，管理内存的策略是决定系统性能的一个关键因素。内存管理的主要功能是使多个进程可以共享机器的主存，并且提供虚拟内存机制使进程的运行可以脱离系统物理内存的限制。其主要的功能为地址映射、地址保护、页面分配、页面交换等。页面交换在嵌入式系统中很少用到，地址映射和地址保护需要系统硬件的支持。从内核的代码结构上来看，内存管理部分主要分为物理内存管理和虚拟内存管理。内存管理在 ARM 平台上的移植是工作量较大的部分，需要根据硬件平台的物理内存布局来设置在虚拟内存空间中的映射，并配置一个数组 `map_desc` 来对这个映射关系进行描述。
- 文件系统：Linux 在很大程度上依赖于文件系统的概念，Linux 中的每个对象都可以当作文件来看待。内核在没有结构的硬件上构造结构化的文件系统，而文件抽象在整个系统中广泛使用。Linux 支持多种文件系统，也就是支持在物理介质上组织数据的不同方式。例如，磁盘可以格式化为符合 Linux 标准的 `ext2` 文件系统，也可以格式化为常用的 FAT 文件系统。嵌入式系统的工作环境决定了应用于嵌入式系统的文件系统必须使用日志管理对存储器的操作过程，比如针对 Flash 特性建立的 YAFFS 文件系统。
- 设备控制：Linux 将每一个系统操作都映射到物理设备上。除了处理器、内存以及其他很有限的几个对象外，所有设备控制操作都由与被控制设备相关的代码来完成，这段代码叫做驱动程序。内核必须为系统中的每件外设嵌入相应的驱动程序，所以对 ARM 内核 S3C2410X 芯片的片上外围设备的驱动程序的移植也成为移植工作中的重要组成部分。

- 网络功能：大部分网络操作和具体进程无关：数据包的传入是异步事件。在某个进程处理这些数据包之前必须收集、标识和分发这些数据包。内核负责在应用程序和网络接口之间传递数据包，并根据网络活动控制程序的执行。另外，所有的路由和地址解析问题都由内核处理。在网络设备驱动被移植后，嵌入式工控平台就可以利用 Linux 内核提供的网络接口来实现与互联网的联功能。

Linux 内核源代码非常庞大，它使用目录树结构，整个代码分布图如图 5.3 所示。

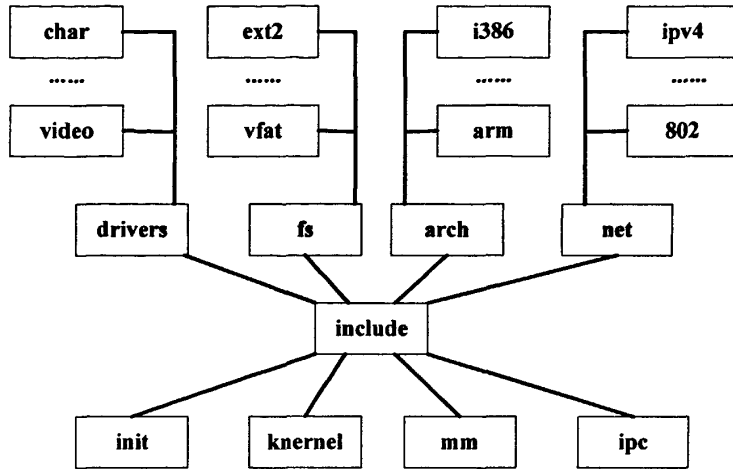


图 5.3 Linux 内核代码分图

内核源码的/Linux/arch/arm/目录是与 ARM 平台相关的内核代码，而移植工作的重点就是移植 arm 目录下的文件，其下面的子目录和文件如下^[25]：

kernel——ARM 体系结构的通用内核代码，主要包括 ARM 平台必须使用的 ARM 汇编程序，如启动时的 head-arm.S、与异常中断处理相关的 entry-armv.S 以及其它一些和 ARM 体系结构相关的进程调度管理、系统调用、调试体系等 C 程序。

mm——与 ARM 平台下内存管理部分相关的内核代码。主要有：与内存管理相关的 mm-armv.c；与内存异常处理相关的 fault-armv.c；与各 CPU 的 MMU、cache、buffer 等内存管理部件的操作相关的汇编代码，其中 proc-arm920.S 是针对 ARM920T 的代码。

lib——ARM 平台下经过了优化的内部库函数和低级 I/O 操作，如内存拷贝函数 memcpy，I/O 函数，位操作函数等，主要以 ARM 汇编写成。

nwfpe、fastpe——对 ARM 平台下浮点数模拟操作的两个不同实现。ARM 平台下浮点数操作作为一个协处理器来实现。当系统中没有协处理器时，协处理器操作指令就会产生指令未定义异常，可以在异常处理中来支持对浮点数操作的模拟。

boot——最终编译得到的内核就存放在这个目录下，并且还包含了一些为在启动时

对压缩内核进行解压所需要的代码。

tools——用于自动生成如 **mach-type.h** 那样文件的脚本文件。

def-configs——对不同的机器类型的配置文件。配置文件是对 Linux 内核进行编译的时候所必需的文件，它描述了系统从 CPU 类型到基本包含硬件方面的各种信息，是对硬件系统的一个非常全面的概述。

mach-XXX——对每一种不同的机器类型，都有一些和该机型相关的一些程序。**mach-S3C2410** 目录下包含了和 S3C2410X 芯片操作相关的代码，这是比 **kernel** 和 **mm** 目录下的代码更接近底层硬件平台的代码。**vmLinux-armv.lds.in**——链接描述文件。

5.3.2 内核移植

本文选用的 Linux 内核不包含 S3C2410X 处理器，故需要添加相应的内核代码进行移植，移植过程如下：

- (1) 将内核解压缩到当前目录的 **Linux-2.4.18** 目录中，修改它的 **Makefile** 文件。

```
ARCH ?= $(SUBARCH)
```

```
CROSS_COMPILE ?=
```

修改为：

```
ARCH:=arm
```

```
CROSS_COMPILE =/opt/FiendARM/usr/bin/arm-Linux-
```

- (2) **arch** 目录下的 **Makefile** 文件增加：

```
ifeq ($(CONFIG_ARCH_S3C2410),y)
TEXTADDR= 0xC0008000
MACHINE= s3c2410
endif
```

TEXTADDR 决定内核起始运行地址，即 **image.ram** 应下载的位置，根据硬件电路设计，这个地址是 **0Xc0008000**。

- (3) **config.in** 是配置文件，在 **config** 文件中加入以下相关信息：

```
if [ "$CONFIG_ARCH_S3C2410" = "y" ]; then
```

```
comment 'S3C2410 Implementation'
```

```
dep_bool 'SMDK (MERI TECH BOARD)' CONFIG_S3C2410_SMDK
$CONFIG_ARCH_S3C2410
```

```
dep_bool 'change AIJI' CONFIG_SMDK_AIJI
```

```

dep_tristate 'S3C2410 USB function support' CONFIG_S3C2410_USB
$CONFIG_ARCH_S3C2100
dep_tristate ' Support for S3C2410 USB character device emulation'
CONFIG_S3C2410_USB_CHAR $CONFIG_S3C2410_USB
fi  /* CONFIG_ARCH_S3C2410 */

```

(4) 在 `compressed` 目录下加入 `head-s3c2410.S` 文件, 用来对 S3C2410 处理器进行初始化, 并修改其目录下的 `Makefile` 文件:

```

ifeq ($(CONFIG_ARCH_S3C2410),y)
OBJS      += head-s3c2410.o
endif

```

(5) 在 `arch/arm/kerne` 目录下加入 `setup.c` 文件。该文件中包含函数 `setup_arch`, 这个函数用来完成和体系结构相关的初始化工作, 比如对物理内存结构 `meminfo` 的初始化。这个结构将在后面的内存初始化中起很重要的作用。其中, `nr_banks` 指定了内存块的数量, `bank` 指定了每块内存的范围。用来指定块开始及长度的 `PAGE_OFFSET` 和 `MEM_SIZE` 都在 `include/asm-arm/arch-s3c2410/memory.h` 中定义, `PAGE_OFFSET` 是内存开始的地址, 在本系统中设置该值为 `0xc000000UL`。

(6) 内核的裁剪与编译

进入控制台输入 `make menuconfig`, 进入内核配置模式, 如图 5.4:



图 5.4 配置 Linux 内核

进行配置时, 大部分选项可以使用其默认值, 只有小部分需要根据用户不同的需要进行选择。

选择相应的配置时有三种选择方式，它们分别代表的含义如下^[26]：

Y——将该功能编译进内核；

N——不将该功能编译进内核；

M——将该功能编译成可以在需要时动态插入到内核中的模块。

等配置完内核后，用 `make` 进行编译，生成镜像文件 `arch/arm/boot/zImage`，把这个文件下载到内核分区，重新启动，就会看到 Linux 的内核启动信息。

5.4 文件系统的实现

嵌入式设备的自身特点决定了它很少使用大容量的存储设备。它往往选用 ROM、Flash 等作为它的主要存储设备，所以必须使用在这些存储设备上运行的文件系统。文件系统是基于被划分的存储设备上的逻辑上单位上的一种定义文件的命名、存储、组织及取出的方法。如果一个 Linux 系统没有根文件系统，它是不能被正确的启动的。因此，需要为 Linux 创建根文件系统，将其创建在 K9S1208 NAND FLASH 上。

Linux 的根文件系统可能包括如下目录（或更多的目录）^[23]：

(1) `/bin (binary)`: 包含着所有的标准命令和应用程序；

(2) `/dev (device)`: 包含外设的文件接口，在 Linux 下，文件和设备采用同种地方法访问的，系统上的每个设备都在 `/dev` 里有一个对应的设备文件；

(3) `/etc (etcetera)`: 这个目录包含着系统设置文件和其他的系统文件，例如 `/etc/fstab (file system table)` 记录了启动时要挂载的文件系统。

(4) `/home`: 存放用户主目录；

(5) `/lib (library)`: 存放系统最基本的库文件；

(6) `/mnt`: 用户临时挂载文件系统的地方；

(7) `/proc`: Linux 提供的一个虚拟系统，系统启动时在内存中产生，用户可以直接通过访问这些文件来获得系统信息；

(8) `/root`: 超级用户主目录；

(9) `/sbin`: 这个目录存放着系统管理程序，如 `fsck`、`mount` 等；

(10) `/tmp (temporary)`: 存放不同的程序执行时产生的临时文件；

(11) `/usr (user)`: 存放用户应用程序和文件。

5.4.1 建立文件系统

在嵌入式 Linux 文件系统的构建中, Busybox 是最常用的一个软件工具包。Busybox 将大量的工具集集成到一个可执行文件中。目前的 Busybox 集成了一百多条 Linux 常用命令, 这些工具和程序功能的集合, 足以满足嵌入式 Linux 的典型应用。在构造嵌入式文件系统中, 工具包中的各个命令可以在编译时很方便地选择或不选择编入 Busybox, 这样就很容易构建一个自定义的小巧完整的 POSIX 嵌入式系统环境。

根文件系统的构造步骤如下:

(1) 配置编译安装 busybox-1.1.0。在 Linux 主机下进入 busybox-1.1.0, 运行配置命令 `make menuconfig`, 进入配置界面窗口, 它的配置方法和内核的配置方法相似, 选择嵌入式系统运行所需要的工具, 然后运行 `make`、`make install` 命令, Linux 的文件目录将被安装在 `_install` 目录中, 将其更名为 `friend_root`。

(2) 构造根文件系统。根文件中存放了系统使用的应用程序、库以及系统配置等其他一些相关文件。课题以 Busybox 安装后生成的目录为基础, 构造根文件系统的目录树。

① 创建顶层目录结构

安装目录 `friend_root` 下已经有 “`bin`”、“`sbin`”、“`usr`” 三个目录, 需要创建其它目录, Linux 操作系统启动后目录显示如下:

```
/# ls
bin      home     lost+found  sbin     usr
dev      lib      mnt         sys      var
etc      Linuxrc  proc        tmp
```

② 创建 inittab 文件

`inittab` 文件是 `init` 程序读取的配置文件。`init` 是内核启动后运行的第一个程序, 所有的应用程序都是它的子程序, 它在系统运行期间一直驻留在内存运行, 直到系统关闭。Busybox 的 `init` 程序在启动时会读取 `inittab` 文件的内容, 并执行相应的命令。在 `/etc` 目录下创建 `inittab` 文件, 文件内容为:

```
::sysinit:/etc/init.d/rcS          #系统在初始化时执行的内容
::askfirst:/bin/sh                #询问后在串口启动 Shell
```

③ 建立 sysinit 动作项执行的命令脚本

sysinit 动作项指定的命令, 是 inittab 指定的在系统初始化时执行的命令, 通常这个命令使用脚本实现。脚本主要实现系统的初始化, 如 proc 文件系统的安装、网络的启动等。按照 inittab 命令的要求, 应在/etc/init.d 目录下创建名为 rcS 的脚本, 并设置脚本为可执行文件。脚本内容如下:

```
#!/bin/sh

mount -t proc proc /proc          #挂载 proc 文件系统

mount -t sysfs sysfs /sys        #挂载 sys 文件系统

ifconfig lo 127.0.0.1

ifconfig eth0 172.16.16.118      #设置 IP 地址
```

5.4.2 移植 Yaffs 文件系统

Yaffs 是为 NANDFlash 而设计的文件系统。与 JFFS 相比, 它减少了一些功能, 因此运行速度更快, 占用内存更少^[27]。在进行移植的时候先将 Yaffs 文件系统包含进 Linux 内核, 使其得到内核良好的支持。移植过程如下:

- (1) 已建立 Yaffs 目录, 把必需的 Yaffs 文件复制到该目录下:

```
mkdir /fs/Yaffs

cp *.* /fs/Yaffs

*.*代表源码中的必要代码包括 devextras.h, Yaffs_fs.c, Yaffs_gets.c, Yaffs_guts.h,
Yaffs_mtdif.c, Yaffs_mtdif.h, Yaffsinterface.h 以及 yportenv.h.
```

- (2) 在 fs/Yaffs 目录下生成一个 Makefile 文件, 其内容为:

```
O_TARGET := Yaffs.o

obj-y := Yaffs fs.o Yaffs guts.o Yaffs_mtdif.o Yaffs_ecc.o

obj-m := (O_TARGET)

include (TOPDIR)/Rules.make
```

- (3) 修改 fs/config.in,按语法添加以下内容, 使得可以配置 Yaffs :

```
if [ "CONFIG_MTD_Nand" = "y" ]; then
tristate "Yaffs filesystem on Nand" CONFIG_Yaffs_FS
fi
```

- (4)修改 fs/Makefile,按语法添加以下内容, 使得可以编译 Yaffs:

```
obj-(CONFIG_Yaffs FS)+= Yaffs/
```


5.4.3 创建 Yaffs 格式的根文件系统

所谓创建某种格式的根文件系统,就是将根文件系统内容转换成选用的这种文件系统格式,或者将根文件系统内容安装在具有这种文件系统格式的设备^[28]。

Yaffs 源代码包的 `utils` 目录下包含了 `mkYaffsimage/mkYaffs2image` 的代码,简单的修改 `Makefile` 里的内核路径就能编译出 `mkYaffsimage/mkYaffs2image` 工具,将其复制到 `/usr/bin` 下。执行:

```
mkyaffsimage friend_root friend_root.img
```

生成的 `friend_root.img` 映象文件。在 `vivi` 模式下使用 “`loadyaffs -e root u`” 就可以把它下载到板子里面去。要实现从 Yaffs 根文件系统启动,可以在 `vivi` 启动进入命令行状态时向内核发送命令行已设置系统启动参数:

```
char Linux_cmd[] = "noinitrd root=/dev/mtdblock/0 init=/Linuxrc console=ttyS0, 115200"
```

5.4.4 配置内核

最后便是内核编译时的配置,配置内核时要选中对 MTD 和 Yaffs 支持:

```
Memory Technology Devices (MTD) --->
<*> Memory Technology Device (MTD) support
[*] MTD partitioning support
...
--- User Modules And Translation Layers
<*> Direct char device access to MTD devices
<*> Caching block device access to MTD devices
...
Nand Flash Device Drivers --->
<*> Nand Device Support
<*> Nand Flash support for S3C2410 SoC
[*] S3C2410 Nand driver debug
File systems --->
Miscellaneous filesystems --->
<*> Yet Another Flash Filing System(Yaffs) file system
support
```

[*] Nand mtd support

[*] Use ECC functions of the generic MTD-Nand driver

[*] Use Linux file caching layer

[*] Turn off debug chunk erase check

[*] Cache short names in RAM

内核配置完成后，控制系统平台的根文件系统便完成了。该根文件系统可以很好地支持目标平台，并且稳定工作。

5.5 应用程序编写

目前随着计算机硬件设计水平的提高，越来越多的软件开发工作集中在图形用户接口(GUI)上，它极大的方便了非专业用户的使用，操作者可以通过窗口、菜单方便的进行操作。Qt/Embedded（简称 Qt），它是 Trolltech 公司一个标志性产品，是一个为嵌入式设备设计图形用户接口和应用开发而定制的 C++的工具开发包。在应用程序编写之前我们首先要进行 Qt 开发环境的搭建。

5.5.1 Qt 开发环境搭建

构建 Qt/Embedded 交叉编译环境需要构建 3 个 Qt 开发环境：Qt/X11 环境，即 Qt 桌面开发环境；Qt/Embedded(X86)环境，这是在宿主机平台模拟编译和调试目标板程序的环境；Qt/Embedded(ARM)交叉编译环境，在 Qt/Embedded(X86)环境下调试过的程序在此环境下进行最后编译后，才能移植到目标板上运行^[34]。

在构建交叉编译环境之前，需要下载以下四个安装包：cross-95.3.tar.bz、tamke-1.11.tar.gz、qt-x11-2.3.2.tar.gz、qt-embedded-2.3.7.tar.gz。具体的环境搭建过程为：

(1) 安装 tmake

在 Linux 命令模式下运行以下命令：

```
#cp tmake-1.11.tar.gz /usr/local
```

```
#cd /usr/local
```

```
#tar xvzf tmake-1.11.tar.gz
```

```
#export TMAKEDIR=$PWD
```

```
#export TMAKEPATH=/usr/local/tmake-1.11/lib/qws/Linux-x86-g++/
```

```
#export PATH=$TMAKEDIR/bin:$PATH
```

(2) 安装 Qt/X11

在 Linux 命令模式下运行以下命令：

```
#cp qt-x11-2.3.2.tar.gz /usr/local
#cd /usr/local
#tar xvzf qt-x11-2.3.2.tar.gz
#cd qt2.3.2/
#export QTDIR=$PWD
#export PATH=$QTDIR/bin:$PATH
#export LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
```

(3) 安装 Qt/Embedded

在 Linux 命令模式下运行以下命令：

```
#cp qt-embedded-2.3.7.tar.gz /usr/local
#cd /usr/local
#tar xvzf qt-embedded-2.3.7.tar.gz
#cd qt-2.3.7
#export QTDIR=$PWD
#export QTEDIR=$QTDIR
#export PATH=$QTEDIR/bin:$PATH
#export LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
#cp /usr/local/qt-2.3.2/bin/uic designer /usr/local/qt-2.3.7/bin
```

上述操作完成后，在宿主机上已具有 3 个 Qt 的开发环境。将编辑好的源代码在 Qt/Embedded(X86)环境下编译，然后调用虚拟帧缓冲服务器进行前期仿真运行，仿真通过后将源代码在 Qt/Embedded(ARM)环境下进行编译，将生成的二进制代码与 Qt/Embedded 链接库一起下载到目标板上运行，进行目标板调试。

5.5.2 控制器应用程序编写

Qt 开发环境搭建完成后，用系统自带的 Qt Designer 来编写程序。Qt Designer 是一个优秀的可视化开发工具，用它来设计十分容易。它的界面类似于 Delphi，但使用 C++ 语言编写应用程序。Qt 程序设计里的一个亮点就是 SignalSlot 机制，它有点类似于 VC++ 里的消息机制。当一个组件发出 Signal 时，其他一个组件或多个组件可以通过 Slot 接

收发来的 Signal，组件本身也可以接收自己的 Signal，这样处理一个事件将变得非常容易。

信号(signal)和槽(slot)是 Qt 的核心机制，是 Qt 程序的基础，也是区别于其它工具包的重要地方。信号和槽是一种高级接口，应用于对象之间的通信，一个按钮被单击时会激发一个 clicked()信号。程序员通过建立一个槽，如应用程序等，然后调用 clicked()函数把这个槽和信号连接起来，这样就完成一个事件和响应代码的连接。

信号和槽的机制是类型安全的，它以警告的方式报告类型错误，而不会使系统产生崩溃。信号和槽的连接的抽象图如图 5.5 所示。

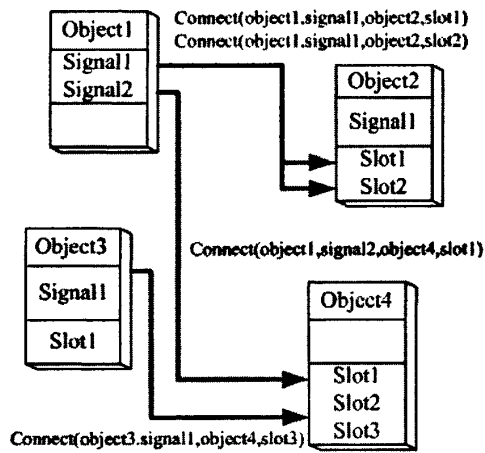


图 5.5 信号与槽的连接

GUI 的设计过程为：在 qt-2.3.x for x1 的安装路径的 bin 目录下运行“./designer”命令，启动 Qt 图形编辑器，如图 5.6 所示。点击编辑器的“new”菜单，弹出了一个“new Form”对话框，在这个对话框里选择“Widget”，新建了一个窗体。然后根据系统的要求，放置按钮等部件，设置其相关属性，并且将它们与相关的槽函数做连接。

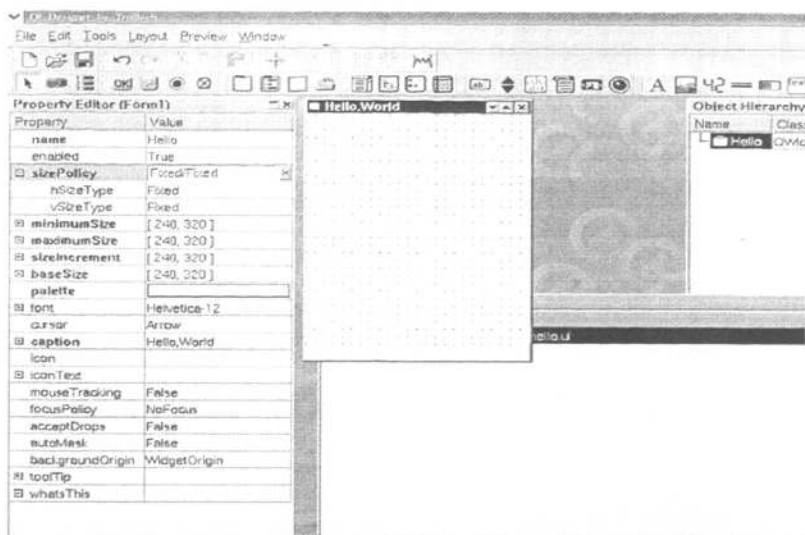


图 5.6 Qt Designer 的设计窗口

使用 uic 工具产生出 form1 窗体类的头文件和实现文件，具体方法是：

```
uic -o form1.h form1.ui
```

```
uic -o form1.cpp -impl form1.h form1.ui
```

这样就得到了 form1 窗体类的头文件 form1.h 和实现文件 form1.cpp。

下面为各个按钮等部件做连接，并且创建槽函数的过程，在这里以点击按钮 (PushButton1) 发射信号 clicked() 来触发槽 (electric_heating()) 电加热为例：

首先做信号与槽的连接：

```
connect( PushButton, SIGNAL( clicked()), this, SLOT(electric_heating());
```

connect() 函数将按钮 (PushButton1) 信号 clicked() 与 (electric_heating()) 作连接，当按钮敲击以后便会触发 (electric_heating()) 函数。

下面为部分应用程序流程图

太阳能集热系统应用程序，根据其系统功能，可将它划分为以下几个程序模块：键盘输入程序、无线通信程序、温度测量程序、水位检测程序、自动上水程序、电加热程序等。对于有些较为复杂的功能模块可根据其功能继续划分。图 5.7 为系统主程序流程图。

(1) 键盘输入程序

键盘输入程序首先判断是否有功能键按下，如果有，则将键值保存，转入数字键处理程序。当按下确定键时，则输入有效，当按下取消键时，则输入无效。最后根据功能键和数字键的键值进行相应的处理。键盘输入程序流程图如图 5.8 所示。

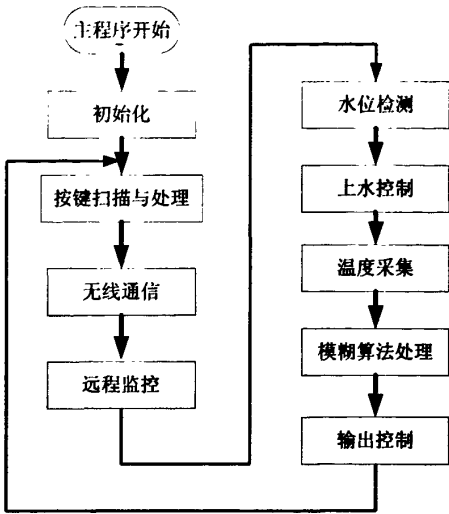


图 5.7 主程序流程图

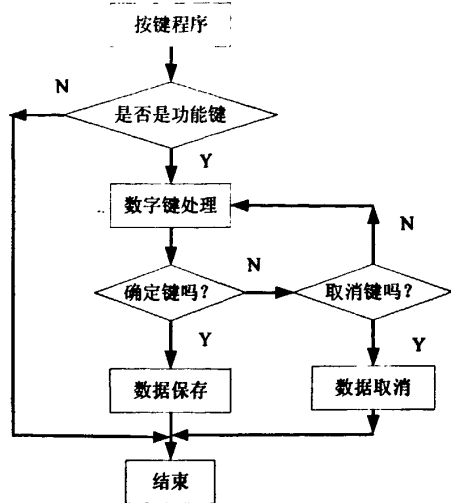


图 5.8 键盘输入程序流程图

(2) 无线通信程序

无线通信程序的编写主要包括 XE1205 芯片初始化程序、无线数据发送程序、无线数据接收程序，初始化程序主要包括寄存器的读程序和写程序。程序流程图如图 5.9 所示。

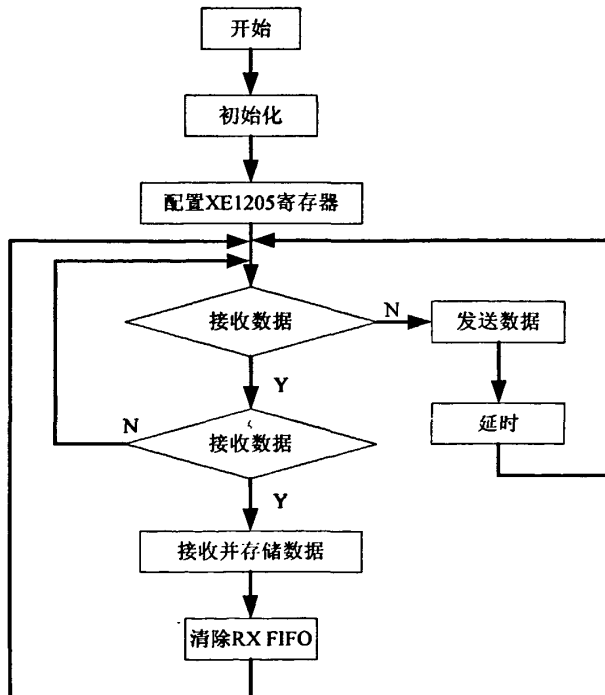


图 5.9 无线通信程序流程图

(3) 测温程序

温度传感器采用 DS18B20，它完成温度转换的流程是：初始化→ROM 操作指令→存储器操作指令→数据传输。图 5.10 为其测温程序流程图。

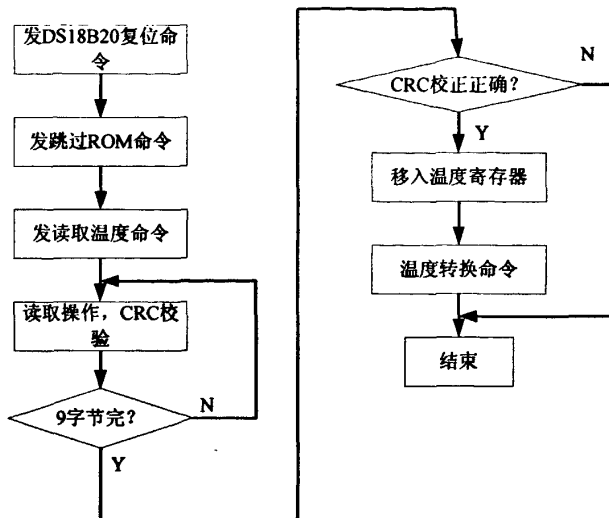


图 5.10 测温程序流程图

(4) 水位检测及自动上水程序

自动上水，是通过设定来实现水箱里缺水时自动补水、水满时自动停水的功能。由水位传感器根据缺水或满水时发出的不同信号控制继电器的通断，从而进一步控制上水泵的开关状态。图 5.11 和图 5.12 分别为水位检测和自动上水程序流程图。

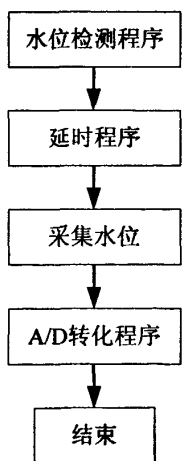


图 5.11 水位检测程序流程图

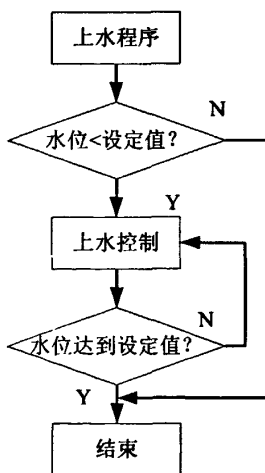


图 5.12 上水控制程序流程图

(5) 电加热程序

利用太阳的热能对水箱内的水进行加热，容易受到天气因素的影响。如果遇到阴雨

天气，一般都达不到用户要求，这就需要对其进行辅助电加热。电加热的程序流程图如图 5.13 所示。

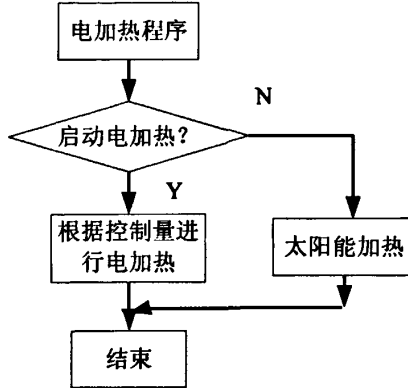


图 5.13 电加热控制程序流程图

5.6 本章小结

本章首先介绍了主控制器软件平台的搭建的过程，主要包括交叉编译环境建立、系统引导程序 Bootloader 移植、Linux 操作系统内核移植和文件系统的建立四部分工作，并具体介绍了每部分的配置方法和过程，然后介绍了 Qt 开发环境搭建过程及应用程序的编写，最后画出了部分应用程序的程序流程图。

第 6 章 太阳能集热系统模糊 PID 控制及仿真

对太阳能集热系统的控制有多种方法,例如传统的 PID 控制、模糊控制等控制方法。PID 控制器是一种线性控制器,它根据给定值 $r(t)$ 与实际输出值 $c(t)$ 构成控制偏差 $e(t)=r(t)-c(t)$,将偏差的比例 (P)、积分 (I)、微分 (D) 通过线性组合构成控制量,对被控对象进行控制,其控制规律为:

$$u(t) = K_p[e(t) + \frac{1}{T_i} \int_0^t e(t)dt + \frac{T_D}{dt} \frac{de(t)}{dt}] \quad (6.1)$$

或写成传递函数形式:

$$G(s) = \frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_D s \right) \quad (6.2)$$

其中 K_p 为比例系数; T_i 为积分时间常数; T_D 为微分时间常数。

PID 控制器以其算法简单、计算量小等优点而得到广泛应用。但是太阳能集热系统是一个大惯性、纯滞后,具有时变性、非线性且受外界环境变化影响较大的复杂控制系统,我们难以对其建立精确的数学模型,常规的 PID 控制不能很好的满足其静态动态特性的要求,难以达到令人满意的控制效果。而模糊控制是目前三种智能控制方法(人工智能、模糊控制、神经网络)中最具实际应用价值的一种,其在本质上的非线性使得它在控制中不依赖于被控对象精确的数学模型,对被控对象的时滞、非线性、时变性具有一定的适应能力,同时对噪声也有较强的抑制能力,鲁棒性较好^[30],但是简单的模糊控制器由于不具备积分环节,因而又很难完全消除稳态误差,而且在变量分级不足够多的情况下,常常在平衡点附近会有小的振荡现象,但模糊控制器对复杂的和模型不清楚的系统却能进行简单而有效的控制。所以将模糊控制和 PID 控制结合起来,这样既具有模糊控制灵活和适应性强的优点,又具有 PID 控制精度高的特点。

将模糊技术与 PID 控制算法结合起来构成模糊 PID 控制器有多种方法。一种是利用模糊控制器来给 PID 控制器在线自整定 PID 参数,组成模糊自整定参数 PID 控制器。另一种是在大偏差范围内采用比例控制,而在小偏差时采用模糊控制,两种控制方式的切换是根据预先确定的偏差阈值来控制,这就构成 FUZZY-P 双模分段控制器。

本课题采用第一种方法,利用模糊控制对 PID 进行在线自整定^[60]。其主要思想是在常规 PID 调节基础上,采用模糊推理,根据不同的系统偏差 E 和偏差变化 EC 实现 PID 参数自整定,实现对水温的自动控制。

6.1 模糊控制系统的基本原理和组成

模糊控制的基本原理如图 6.1 所示，系统包括参考输入、模糊控制器和被控对象。通常以系统偏差 E 和偏差变化 EC 作为模糊控制器的输入。它的核心部分为模糊控制器，即图中虚线框内部。

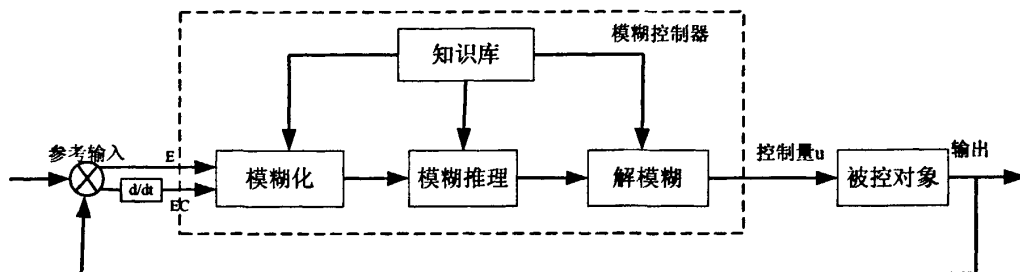


图 6.1 模糊控制系统原理图

模糊控制器由模糊化、知识库、模糊推理和解模糊四部分组成。

(1) 模糊化

模糊化是将输入的精确量转换成模糊量，输入量包括外界的参考输入、系统的输出或状态等。具体过程包括论域变换和模糊化。论域变换是将精确量所在的真实论域通过正规化因子变换到基本论域上^[35]。模糊化是指论域变换后的输入变量仍是精确量，需要根据定义在基本论域上的各个模糊集合，来确定精确量所属于的模糊子集及其对应的隶属函数，于是将精确量进行模糊化处理得到了相应的模糊变量。

(2) 知识库

知识库由数据库和规则库组成，包含了具体应用领域的知识和要求。其中，数据库主要包含各个变量的隶属函数、正规化因子、比例因子以及模糊化、模糊推理算法、解模糊算法等知识；规则库包括了用模糊语言表示的一系列控制规则，它们反映了控制专家的经验 and 知识。

(3) 模糊推理

模糊推理是模糊逻辑控制系统和模糊控制的核心，具有模拟人基于模糊概念的推理能力的行为。它根据模糊系统的输入和模糊推理规则，经过模糊关系合成和模糊推理合成等逻辑运算，得出模糊系统的输出。模糊控制应用的是广义前项推理，该推理过程是基于模糊逻辑中的模糊推理算法及模糊推理规则来进行的^{[36][37]}。假设双输入单输出的一个控制器，其建立的模糊控制器规则库为：

$$R_1 : \quad \text{if } x = X_1 \quad \text{and} \quad y = Y_1 \quad \text{then} \quad z = Z_1$$

$$R_2 : \quad \text{if } x = X_2 \quad \text{and} \quad y = Y_2 \quad \text{then} \quad z = Z_2$$

·
·
·

$$R_n : \quad \text{if } x = X_n \quad \text{and} \quad y = Y_n \quad \text{then} \quad z = Z_n$$

如果已知模糊控制器的输入为 $x = X'$ 和 $y = Y'$ ，则根据模糊控制规则进行近似推理可得出输出模糊量 z （用模糊集合 Z' 表示）为：

$$Z' = (X' \text{ and } Y') \circ R \tag{6.3}$$

$$R = \bigcup_{i=1}^n R_i \tag{6.4}$$

$$R_i = (X_i \text{ and } Y_i) \rightarrow Z_i \tag{6.5}$$

其中包括了三种主要的模糊逻辑运算：**and** 运算，合成运算“ \circ ”，蕴含运算“ \rightarrow ”。**and** 运算通常采用求交（取小）或求积（代数积）的方法；合成运算“ \circ ”最大-最小或最大-积的方法；蕴含运算“ \rightarrow ”通常采用求交（ R_c ）或求积（ R_p ）的方法。

(4)解模糊

解模糊是将模糊推理得到的控制量（模糊量）变换为实际用于控制的精确量。它包括将模糊的控制量经解模糊变换为基本论域上的精确量和将基本论域上的精确量经过比例因子变换成真实论域上的精确量。

6.2 一般模糊控制器的设计方法

模糊控制器是一种语言控制器，它模拟专家的控制经验，通过语言将其描述成控制规则，根据控制规则进行控制计算，从而完成闭环控制的任务。下面简单说明一下模糊控制器的基本设计过程^[33]。

(1) 第一步生成概念数据库：首先，根据系统的具体控制要求，选择模糊控制器的输入输出变量，确定输入输出变量的个数；其次，根据操作经验或者采样数据，确定每一个输入输出变量的取值范围，即模糊控制器变量的论域；然后，对每一个输入输出变量论域模糊划分，选择适合的模糊集合以及模糊集合的形态及位置参数；最后确定模糊集合隶属函数的形式。

定义一个模糊子集，实际上就是要确定模糊子集隶属函数曲线的形状。选取隶属函

数的一个基本原则是，隶属函数曲线的形状较尖的模糊子集分辨率高，控制灵敏度也较高；相反隶属函数曲线较缓，控制特性也较平缓，系统稳定性较好。值得注意的是，如果所需要的模糊控制是一个连续系统，采用连续的模糊隶属函数即可实现模糊控制；如果模糊系统是一个离散过程，还需要将确定的隶属函数曲线离散化，得到有限点上的隶属度，构成相应的模糊变量的模糊子集。

(2) 第二步生成控制规则库：首先，确定模糊规则的具体形式；其次，生成系统的模糊规则；然后，分析模糊规则的完备性、协调性以及规则的相互干扰，最后建立完整的模糊规则库。其中生成规则库的方法一般有以下四种：

- 根据专家经验或过程控制知识生成控制规则；
- 根据过程的模糊模型或者样本数据提取模糊规则生成控制规则；
- 根据对手动控制操作的系统观察和测量生成控制规则；
- 根据基于测量数据的学习算法生成控制规则。

(3) 最后，确定模糊逻辑推理的具体算法，主要包括：

- 定义模糊蕴涵算子的具体形式；
- 定义模糊命题连接词“与”和“或”的具体形式；
- 定义复合推理算子的具体形式；
- 确定解模糊方法。

6.3 模糊 PID 控制器的设计及仿真

6.3.1 模糊 PID 控制器的设计

模糊 PID 控制器是在常规 PID 调节器的基础上，应用模糊集合理论，依据控制偏差 E 、偏差变化 EC 在线自动调整 ΔK_p 、 ΔK_i 和 ΔK_d 的模糊控制器。其结构如图 6.2 所示。

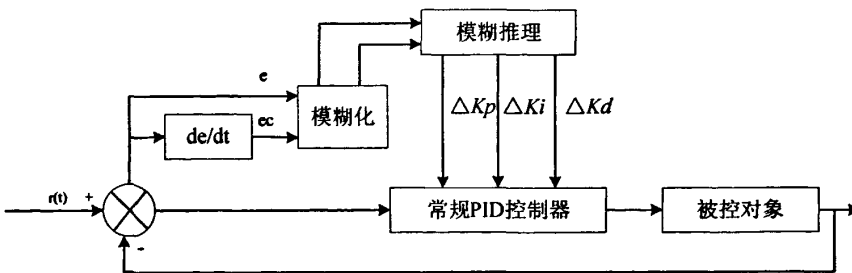


图 6.2 模糊 PID 控制器结构图

模糊PID控制器的具体设计过程如下：

(1) 确定输入、输出变量的论域及隶属度函数

本课题采用二输入三输出的二维模糊控制器。输入为设定温度与实际温度的偏差 E ，偏差变化率 EC ，输出为 ΔK_p 、 ΔK_i 、 ΔK_d 。温度偏差 E 和偏差变化 EC 均划分为七个模糊子集，即{负大，负中，负小，零，正小，正中，正大}，论域取为{-3, -2, -1, 0, 1, 2, 3}。输出控制量 ΔK_p 、 ΔK_i 、 ΔK_d 也均划分为七个模糊子集，论域分别取为： $\Delta K_p = \{-3, -2, -1, 0, 1, 2, 3\}$ ， $\Delta K_i = \{-0.3, -0.2, -0.1, 0, 0.1, 0.2, 0.3\}$ ， $\Delta K_d = \{-0.3, -0.2, -0.1, 0, 0.1, 0.2, 0.3\}$ 。各语言值隶属函数均采用三角形。 E 、 EC 、 K_p 的隶属度函数如图 6.3 所示， K_d 、 K_i 的隶属度函数如图 6.4 所示。

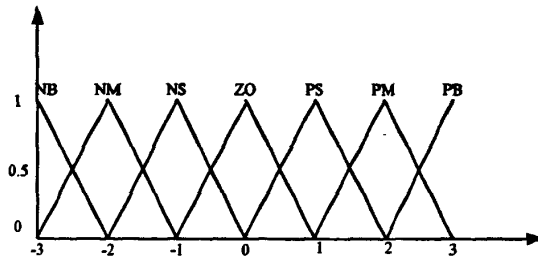


图 6.3 E 、 EC 、 K_p 的隶属度函数

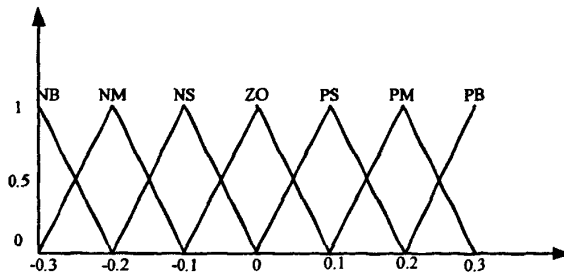


图 6.4 K_d 、 K_i 的隶属度函数

(2) 模糊控制规则表

我们采用经验归纳法生成模糊规则库。所谓的经验归纳法就是总结、归纳专家的控制经验，经过进一步加工、整理、提炼、去粗取精后产生模糊控制规则的一种方法。模糊PID控制设计的核心是总结工程设计人员的技术知识和实践操作经验，建立合适的模糊规则表，得到针对 ΔK_p 、 ΔK_d 、 ΔK_i 三个参数的模糊控制表如表 6.1、表 6.2、表 6.3 所示。

表 6.1 ΔKp 模糊控制表

$E \backslash EC$	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PM	PM	PS	ZO	ZO
NM	PB	PB	PM	PS	PS	ZO	NS
NS	PM	PM	PM	PS	ZO	NS	NS
ZO	PM	PM	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NS	NS	NM	NM
PM	PS	ZO	NS	NM	NM	NM	NB
PB	ZO	ZO	NM	NM	NM	NB	NB

表 6.2 ΔKi 模糊控制表

$E \backslash EC$	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NM	NM	NS	ZO	ZO
NM	NM	NB	NM	NS	NS	ZO	ZO
NS	NB	NM	NS	NS	ZO	PS	PS
ZO	NM	NM	NS	ZO	PS	PM	PM
PS	NM	NS	ZO	PS	PS	PM	PB
PM	ZO	ZO	PS	PS	PM	PB	PB
PB	ZO	ZO	PS	PM	PM	PB	PB

表 6.3 ΔKd 模糊控制表

$E \backslash EC$	NB	NM	NS	ZO	PS	PM	PB
NB	PS	NS	NB	NB	NB	NM	PS
NM	PS	NS	NB	NM	NM	NS	ZO
NS	ZO	NS	NM	NM	NS	NS	ZO
ZO	ZO	NS	NS	NS	NS	NS	ZO
PS	ZO	ZO	ZO	ZO	ZO	ZO	ZO
PM	PB	PS	PS	PS	PS	PS	PB
PB	PB	PM	PM	PM	PS	PS	PB

根据上述三个表可以得到如下 49 条模糊控制规则语句^[51]：

1. If (E is NB) and (EC is NB) then (Kp is PB) (Ki is NB) (Kd is PS)

2. If (E is NB) and (EC is NM) then (Kp is PB) (Ki is NB) (Kd is NS)
3. If (E is NB) and (EC is NS) then (Kp is PM) (Ki is NM) (Kd is NB)
4. If (E is NB) and (EC is ZO) then (Kp is PM) (Ki is NM) (Kd is NB)
5. If (E is NB) and (EC is PS) then (Kp is PS) (Ki is NS) (Kd is NB)
6. If (E is NB) and (EC is PM) then (Kp is ZO) (Ki is ZO) (Kd is NM)
7. If (E is NB) and (EC is PB) then (Kp is ZO) (Ki is ZO) (Kd is PS)
8. If (E is NM) and (EC is NB) then (Kp is PB) (Ki is NB) (Kd is PS)
9. If (E is NM) and (EC is NM) then (Kp is PB) (Ki is NB) (Kd is NS)
10. If (E is NM) and (EC is NS) then (Kp is PM) (Ki is NM) (Kd is NB)
11. If (E is NM) and (EC is ZO) then (Kp is PS) (Ki is NS) (Kd is NM)
12. If (E is NM) and (EC is PS) then (Kp is PS) (Ki is NS) (Kd is NM)
13. If (E is NM) and (EC is PM) then (Kp is ZO) (Ki is ZO) (Kd is NS)
14. If (E is NM) and (EC is PB) then (Kp is NS) (Ki is ZO) (Kd is ZO)
15. If (E is NS) and (EC is NB) then (Kp is PM) (Ki is NB) (Kd is ZO)
16. If (E is NS) and (EC is NM) then (Kp is PM) (Ki is NM) (Kd is NS)
17. If (E is NS) and (EC is NS) then (Kp is PM) (Ki is NS) (Kd is NM)
18. If (E is NS) and (EC is ZO) then (Kp is PS) (Ki is NS) (Kd is NM)
19. If (E is NS) and (EC is PS) then (Kp is ZO) (Ki is ZO) (Kd is NS)
20. If (E is NS) and (EC is PM) then (Kp is NS) (Ki is PS) (Kd is NS)
21. If (E is NS) and (EC is PB) then (Kp is NS) (Ki is PS) (Kd is ZO)
22. If (E is ZO) and (EC is NB) then (Kp is PM) (Ki is NM) (Kd is ZO)
23. If (E is ZO) and (EC is NM) then (Kp is PM) (Ki is NM) (Kd is NS)
24. If (E is ZO) and (EC is NS) then (Kp is PS) (Ki is NS) (Kd is NS)
25. If (E is ZO) and (EC is ZO) then (Kp is ZO) (Ki is ZO) (Kd is NS)
26. If (E is ZO) and (EC is PS) then (Kp is NS) (Ki is PS) (Kd is NS)
27. If (E is ZO) and (EC is PM) then (Kp is NM) (Ki is PM) (Kd is NS)
28. If (E is ZO) and (EC is PB) then (Kp is NM) (Ki is PM) (Kd is ZO)
29. If (E is PS) and (EC is NB) then (Kp is PS) (Ki is NM) (Kd is ZO)
30. If (E is PS) and (EC is NM) then (Kp is PS) (Ki is NS) (Kd is ZO)
31. If (E is PS) and (EC is NS) then (Kp is ZO) (Ki is ZO) (Kd is ZO)

32. If (E is PS) and (EC is ZO) then (Kp is NS) (Ki is PS) (Kd is ZO)
33. If (E is PS) and (EC is PS) then (Kp is NS) (Ki is PS) (Kd is ZO)
34. If (E is PS) and (EC is PM) then (Kp is NM) (Ki is PM) (Kd is ZO)
35. If (E is PS) and (EC is PB) then (Kp is NM) (Ki is PB) (Kd is ZO)
36. If (E is PM) and (EC is NB) then (Kp is PS) (Ki is ZO) (Kd is PB)
37. If (E is PM) and (EC is NM) then (Kp is ZO) (Ki is ZO) (Kd is PS)
38. If (E is PM) and (EC is NS) then (Kp is NS) (Ki is PS) (Kd is PS)
39. If (E is PM) and (EC is ZO) then (Kp is NM) (Ki is PS) (Kd is PS)
40. If (E is PM) and (EC is PS) then (Kp is NM) (Ki is PM) (Kd is PS)
41. If (E is PM) and (EC is PM) then (Kp is NM) (Ki is PB) (Kd is PS)
42. If (E is PM) and (EC is PB) then (Kp is NB) (Ki is PB) (Kd is PB)
43. If (E is PB) and (EC is NB) then (Kp is ZO) (Ki is ZO) (Kd is PB)
44. If (E is PB) and (EC is NM) then (Kp is ZO) (Ki is ZO) (Kd is PM)
45. If (E is PB) and (EC is NS) then (Kp is NM) (Ki is PS) (Kd is PM)
46. If (E is PB) and (EC is ZO) then (Kp is NM) (Ki is PM) (Kd is PM)
47. If (E is PB) and (EC is PS) then (Kp is NM) (Ki is PM) (Kd is PS)
48. If (E is PB) and (EC is PM) then (Kp is NB) (Ki is PB) (Kd is PS)
49. If (E is PB) and (EC is PB) then (Kp is NB) (Ki is PB) (Kd is PB)

模糊推理采用最小最大重心法，即 Mamdani 法。对所建立的模糊变量、模糊控制规则，经模糊推理得出的三个修正参数 ΔKp 、 ΔKd 、 ΔKi 后就可以实现模糊 PID 的参数调整，计算公式如下：

$$K_p = K'_p + \Delta K_p \quad (6.6)$$

$$K_i = K'_i + \Delta K_i \quad (6.7)$$

$$K_d = K'_d + \Delta K_d \quad (6.8)$$

其中， Kp 、 Ki 、 Kd 为为 PID 初始值， ΔKp 、 ΔKd 、 ΔKi 为模糊控制器输出值。

(3) 模糊控制查询表

模糊控制查询表是模糊控制算法的最终结果，根据上面的分析计算得到本文所采用的模糊控制查询表如表 6.4、表 6.5、表 6.6 所示。

表 6.4 ΔKp 模糊控制查询表

$E \backslash EC$	-3	-2	-1	0	1	2	3
-3	2.69	2.69	2	2	1	0	0
-2	2.69	2.69	2	1	1	0	-1
-1	2	2	2	1	0	-1	-1
0	2	2	1	0	-1	-2	-2
1	1	1	0	-1	-1	-2	-2
2	1	0	-1	-2	-2	-2	-2.69
3	0	0	-2	-2	-2	-2.69	-2.69

表 6.5 ΔKi 模糊控制查询表

$E \backslash EC$	-3	-2	-1	0	1	2	3
-3	-0.26	-0.26	-0.2	-0.2	-0.1	0	0
-2	-0.2	-0.26	-0.2	-0.1	-0.1	0	0
-1	-0.26	-0.2	-0.1	-0.1	0	0.1	0.1
0	-0.2	-0.2	-0.1	0	0.1	0.2	0.2
1	-0.2	-0.1	0	0.1	0.1	0.2	0.26
2	0	0	0.1	0.1	0.2	0.26	0.26
3	0	0	0.1	0.2	0.2	0.26	0.26

表 6.6 ΔKd 模糊控制表

$E \backslash EC$	-3	-2	-1	0	1	2	3
-3	0.1	-0.1	-0.26	-0.26	-0.26	-0.2	-0.1
-2	0.1	-0.1	-0.26	-0.2	-0.2	-0.1	0
-1	0	-0.1	-0.2	-0.2	-0.1	-0.1	0
0	0	-0.1	-0.1	-0.1	-0.1	-0.1	0
1	0	0	0	0	0	0	0
2	0.26	0.1	0.1	0.1	0.1	0.1	0.26
3	0.26	0.2	0.2	0.2	0.1	0.1	0.26

6.3.2 系统实验仿真

选取被控对象为一阶纯滞后模型： $G(s) = \frac{1.5e^{-30s}}{145s+1}$ ，设定温度为 40°C，在 MATLAB

命令窗口运行 Fuzzy 命令进入模糊逻辑编辑器，并创建一个 .fis 文件，控制类型选择为 Mamdani 型，即最小最大重心法，然后分别输入 E 、 EC 、 ΔKp 、 ΔKi 、 ΔKd 的隶属度函数和量化区间，根据上面的 49 条模糊控制规则完成控制规则的输入。这样就创建了一个 .fis 系统文件，完成模糊控制器文件的编辑之后，将其保存到文件夹。

在 SIMULINK 中建立模糊 PID 控制系统模型^{[52][53][54]}如图 6.5 所示，常规 PID 和模糊 PID 仿真图如图 6.6 所示。

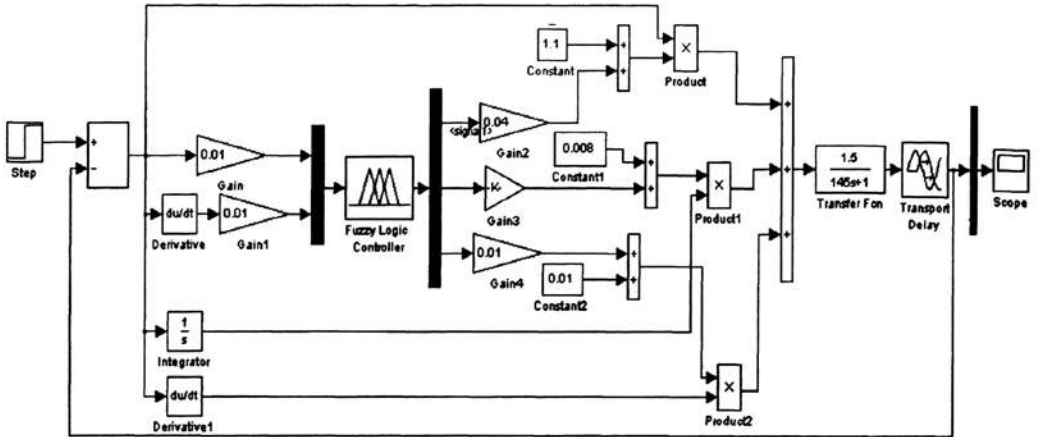


图 6.5 模糊 PID 控制系统模型图

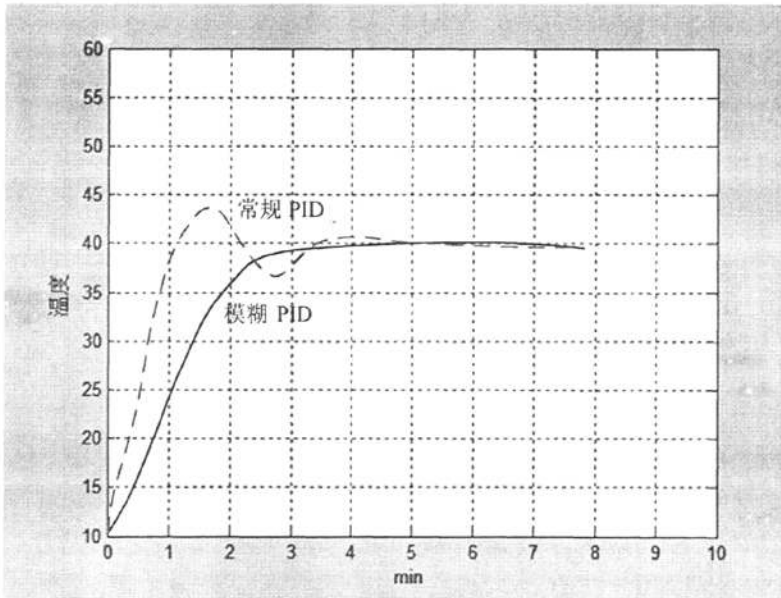


图 6.6 仿真结果图

从仿真结果可以看出，模糊 PID 控制超调量小、调节时间短，具有较好的动态响应特性和稳态特性。其结合了常规 PID 控制的性能稳定，可以消除稳态误差的特点，并具有模糊控制不依赖于数学模型、动态性能好的特点。从实验结果可以看出，模糊 PID 控制的控制效果比较明显。

6.4 模糊 PID 控制算法的软件实现

将模糊控制量查询表存入存储器中，在实际控制过程中，只要根据采集到的温度偏

差和计算出的偏差变化率求得其相应论域中的语言变量 E 和 EC 的值后, 通过查找控制查询表中相应的行和列, 得到输出所需的控制量等级, 从而达到温度控制的目的。温度模糊 PID 控制软件流程图如图 6.7 所示。

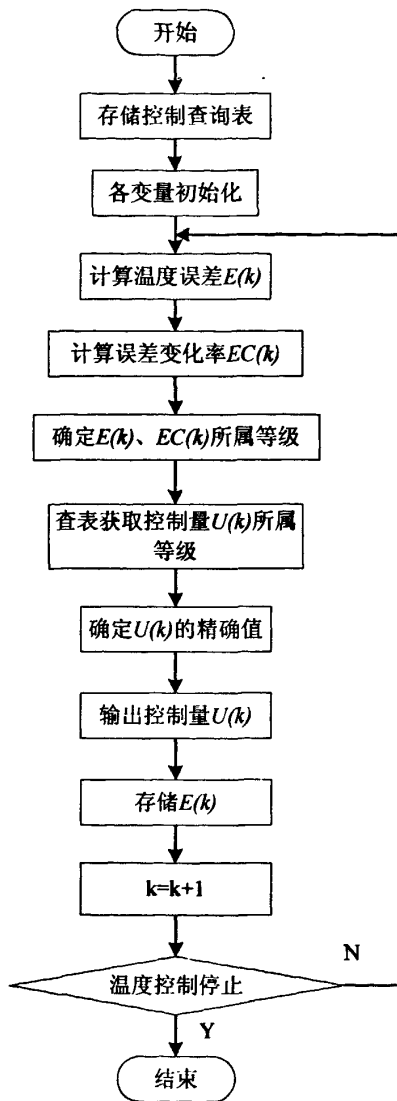


图 6.7 模糊控制软件实现流程图

6.5 本章小结

本章首先介绍了模糊控制器的基本原理和组成及一般模糊控制器的设计方法, 然后针对本课题进行了模糊 PID 控制器的设计及其 MATLAB 仿真, 最后说明了模糊 PID 控制算法的软件实现方法。

第 7 章 总结与展望

7.1 论文总结

通过查阅大量相关资料和实际情况，以及对太阳能热水系统、ARM 处理器系统、模糊 PID 控制方面知识的认识，比较了国内外几种不同的实现方案，结合课题要求设计一个以 S3C2410 为核心的太阳能集热系统的控制系统，实现对建筑内所有用户集中提供热水。本文的主要工作有以下几点：

(1) 硬件设计：包括主控制器和从控制器两部分硬件设计。主控制器以 S3C2410 为核心，其外围硬件电路的设计包括电源电路、存储器电路、水位检测电路、温度检测电路、通信接口电路、键盘输入和报警电路、输出控制电路、遥控发射/接收电路等；从控制器硬件电路设计，主要包括以下电路模块：XE1205 无线传输模块、显示模块、键盘输入模块、串行通信模块、数据存储模块、输出控制模块等。

(2) 软件设计：包括主控制器软件设计和从控制器软件设计。主控制器软件设计首先进行控制器软件平台的搭建，即 Linux 操作系统的建立，包括交叉编译环境的建立、系统引导程序 Bootloader 移植、Linux 操作系统内核移植和文件系统的实现四部分。接着用 Qt 编写在操作系统平台上的应用程序，应用程序根据其系统功能，可划分为网络通信程序、键盘输入程序，温度检测程序，水位检测程序，输出控制程序等程序模块。从控制器软件设计主要是用 C 语言编写与主控制器间的通信程序、输出控制程序、实时显示程序、功能设置程序等。

(3) 对模糊 PID 控制算法进行了分析和研究，采用了二输入三输出的模糊控制器，确定了输入、输出变量的论域及隶属度函数，给出了 PID 参数的自整定规则，完成了模糊 PID 控制算法的设计及实验仿真。

7.2 展望

太阳能热水系统与建筑的完美结合，是在实践的基础上不断完善，相信随着太阳能技术的逐步成熟和工程实践中经验的不断积累，太阳能技术将发挥出更大的优势，为社会做出更大的贡献。作为智能控制的另一新的理论——神经网络也可以应用到系统中，由于时间关系没作研究，将来可以在该方面继续努力。

参考文献

- [1] 孟凤果. 单片机在太阳能中央热水系统中的应用[D].河北: 河北工业大学, 2008
- [2] 谭素梅. 太阳能热利用系统的自适应模糊控制[D].江苏: 东南大学, 2006
- [3] 李俊峰, 王仲颖. 中华人民共和国可再生能源法解读[M].北京: 化学工业出版社, 2005
- [4] 王刚.真空管太阳能热水器热效益及经济性研究[D].天津大学学报, 2002
- [5] 殷志强.全玻璃真空太阳能集热器[M].北京: 北京科学出版社, 2004
- [6] 罗振涛.中国太阳能热水器产业发展研究[M].北京: 中国环境科学出版社, 2003
- [7] 北京市建设委员会主编.新能源与可再生能源利用技术[M].北京: 冶金工业出版社, 2006
- [8] 徐向荣,贺玲丽.建筑设计采用太阳能热水器的探讨工程质量[J], 2004
- [9] 王崇杰,赵学义.太阳能建筑一体化设计[J].建筑学报, 2002,7
- [10] 陆维德.中国太阳能热利用进展与发展对策.中国太阳能产业网 www.asksolar.com
- [11] N.G.Chalhoub and B.A.Bazzi, Fuzzy Logic Control for an Integrated System of a Micro-Manipulator with a Single Flexible Beam,May 2004.10.755-776
- [12] Coad WJ. Energy Engineering and Management for Building Systems. Van Nostrand ComPany, 1982
- [13] 周维, 陈默.基于 S3C2410 的 ARM 开发平台.电子技术[J], 2004
- [14] 李力. 闪存存储器技术现状及发展趋势[J]. 单片机与嵌入式系统, 2001, 8:35-38
- [15] Samsung semiconductor. 64M x 8 Bit NAND Flash Memory Revision 0.7[EB/OL].
<http://www.samsung.com/Products/Semiconductor/2003>, 9
- [16] Samsung semiconductor. 256Mb E-die SDRAM Specification Revision 1.3 [EB/OL].
<http://www.samsung.com/Products/Semiconductor/2003>, 9
- [17] Cirrus Logic.CS8900A Crystal LAN[®] ISA Ethernet Controller [EB/OL]. <http://www.cirrus.com>
2004, 10
- [18] 李正军.计算机测控系统设计与应用[M].北京: 机械工业出版社, 2004
- [19] 陈颐. ARM9 嵌入式技术及 Linux 高级实践教程[M].北京: 北京航空航天大学出版社, 2005
- [20] 郭志, 江秀臣, 曾奕.一个嵌入式系统的启动分析[J].微计算机信息, 2005, 22:33-35
- [21] 孙天泽, 袁文菊, 张海峰. 嵌入式设计及 linux 驱动开发指南: 基于 ARM9 处理器[M].北京: 电子工业出版社, 2005
- [22] 胡志刚, 余正军, 吴运星嵌入式 Linux 实时性方法[J].中南大学学报: 自然科学版, 2004
- [23] 李善平, 刘文峰,王焕龙. Linux 与嵌入式系统[M].北京: 清华大学出版社, 2006, 3

- [24] Corber J, Rubini A, Kroah-Hartman G. Linux Device Driver 3th[M]. O'Reilly , 2005, 4
- [25] Daniel P. Bovet, Marco Cesati Understanding the Linux Kernel, 3rd Edition O'Reilly. 2005
- [26] 岳洋, 张杰明. Linux 内核技术分析 & 升级[J]. 航空计算技术, 2000
- [27] 刘振纲, 刘成安, 卢剑翔. 嵌入式 Linux 中 Yaffs 根文件系统的实现[J]. 嵌入式技术, 2007
- [28] Karim Ghmour. 构建嵌入式 LINUX 系统[M]. 北京: 中国电力出版社. 2004, 12
- [29] 周立功. ARM 嵌入式系统基础教程[M]. 北京: 北京航空航天大学出版社, 2005
- [30] 窦振中. 模糊逻辑控制技术及其应用[M]. 北京: 北京航空航天大学出版社, 1995
- [31] 王倩. 非线性时滞系统的模糊控制[D]. 辽宁: 大连海事大学, 2008
- [32] 王燕. 基于遗传算法的模糊控制器优化设计[D]. 安徽: 安徽理工大学, 2007
- [33] 罗婷. 基于 PMV 指标的室内热舒适控制方法的应用研究[D]. 山东: 山东建筑大学, 2009
- [34] 胡骏, 胡爱群, 宋宇波. Linux 环境下基于 QT 的 WLAN 管理信息系统[J]. 计算机工程, 2004
- [35] 韩峻峰, 李玉慧. 模糊控制技术[M]. 重庆: 重庆大学出版社, 2003.
- [36] 李士勇. 模糊控制、神经控制和智能控制论[M]. 黑龙江: 哈尔滨工业大学出版社, 1998.
- [37] 许力. 智能控制与智能系统[M]. 北京: 机械工业出版社, 2007.
- [38] 孙增圻. 智能控制理论与技术[M]. 北京: 清华大学出版社, 2003.
- [39] 贾智平 张瑞华. 嵌入式系统原理与接口技术[M]. 北京: 清华大学出版社, 2005.
- [40] 刘淼. 嵌入式系统接口设计与 Linux 驱动程序开发[M]. 北京: 北京航空航天大学出版社, 2006
- [41] 刘媛. 基于 Linux 的嵌入式数字家居服务器的研究[D]. 山东: 山东建筑大学, 2007
- [42] 朱振涛, 王成儒, 崔冬. S3C2410 与 TFT-LCD 接口设计[J]. 电子测量技术, 2006
- [43] Nalin K. Gautam and N. D. Kaushika. A Model for the Estimation of Global Solar Radiation Using Fuzzy Random Variables. Centre for Energy Studies. Indian Institute of Technology, Delhi, India
- [44] D. A. Linkens and M. F. Abbod. Supervisory intelligent control using a fuzzy logic hierarchy, Transactions of the Institute of Measurement and Control, Jan 1993. 15. 112-132
- [45] Siegfried Gottwald. Mathematical Fuzzy Control. A Survey of Some Recent Results. Logic Jnl IGPL, Sep 2005. 13. 525-541
- [46] M. BROWN and C. J. HARRIS, A Nonlinear Adaptive Controller. A Comparison between Fuzzy Logic Control and Neurocontrol. IMA J Math Control Info, Jan 1991. 8. 239-265
- [47] James F. Blac. Weather Control: Use of Asphalt Coatings To Tap Solar Energy. Science, Jan 1963. 139. 226-227
- [48] T Stuetzle. N Blair, W A Beckman, and J W Mitchell. Use of linear predictive control for a solar

electric generating system. *Building Service Engineering*. Feb 2004. 25. 55-63

[49] Adel H. Eltimsahy and Emilio A. Santos. JR. Controller for a solar heating system with an

electrical backup. *SIMULATION*. Aug 1979; 33: 37-48

[50] 窦振中. 模糊逻辑控制技术及其应用[M]. 北京: 北京航空航天大学出版社, 1995

[51] B.M.Mohan, Arpita Sinha. The simplest fuzzy PID controllers: mathematical models and stability analysis[J]. *Soft Computing*, 2006, 10: 961-975

[52] 魏巍. MATLAB 控制工具箱技术手册[M]. 北京: 国防工业出版社, 2004:

[53] 李国勇. 智能控制及其 MATLAB 实现[M]. 北京: 电子工业出版社, 2005:

[54] 王春霞. 模糊自整定 PID 温控系统的仿真研究及设计[D]. 长春: 东北师范大学, 2005

致谢

在此首先衷心感谢我的导师段晨旭教授。攻读硕士学位三年期间，段老师严谨缜密的治学精神，精益求精、一丝不苟的治学态度，勤于实践的治学方法和全身心投入工作的热情，为我树立了永远的学习榜样。段老师不仅教育我如何做学问，还教导我如何做事为人，在生活和学习上都给了我无微不至的关怀和帮助。导师渊博精深的知识、高屋建瓴的见解和敏锐的洞察力，都让我终身受益，在段老师的谆谆教导下，我不仅学到了丰富的专业知识，而且获得了人生最宝贵的财富——待人处世之道，我为能有这样一位好导师而感到庆幸和自豪。

在该课题的研究过程中也得到了杨修文老师、王海鹏师哥、罗婷师姐的指导和帮助，在此向他们表示诚挚的感谢！还要感谢实验室的张绪伟、肖现华、石风刚等，他们在研究生期间都给予了我很大的帮助和支持。

另外，我要感谢我的父母，他们给了我很大的支持和动力，使我最终顺利地完成了毕业设计。我还要向所有其他给予我帮助和关怀的老师、同学和朋友们表示感谢。

最后，衷心感谢尊敬的老师们在百忙中抽出时间评审我的论文！

攻读硕士学位期间发表的学术论文及科研情况

- [1] 吕昌江, 段晨旭, 罗婷. 基于 CAN 总线的粮库多点温度检测系统的设计. 信息技术与信息化, 2008 (5)
- [2] 吕昌江, 段晨旭, 罗婷. 太阳能作为绿色能源在建筑中的应用研究. 高等学校智能建筑教学与学术研讨会论文集, 2009 (6)
- [3] 罗婷, 段晨旭, 王海鹏, 吕昌江. 基于舒适行为的空调控制系统的研究. 仪器仪表学报 (增刊), 2008 (4)

附录

函数: WriteRegCmd

功能: 写 xe1205 一个内部寄存器

*****/

```
void WriteRegCmd(uchar RegAddr,uchar RegValue)
```

```
{
    uchar    i;

    sck = 0;
    nss_c = 0;

    RegAddr <<= 0x01;
    RegAddr &= 0x3f;
    RegAddr |= 0x01;

    for(i = 0x00;i < 0x08;i++)
    {
        sck = 0;
        if(RegAddr&0x80)
        {
            mosi = 1;
        }
        else
        {
            mosi = 0;
        }
        sck = 1;
        RegAddr <<= 0x01;
    }
    for(i = 0x00;i < 0x08;i++)
    {
        sck= 0;
        if(RegValue&0x80)
        {
            mosi = 1;
        }
        else
        {
            mosi= 0;
        }
        sck = 1;
        RegValue <<= 0x01;
    }
}
```

```

    }

    sck = 0;
    mosi = 0;
    nss_c = 1;
}
/*****
函数: ReadRegCmd
功能: 读 xe1205 一个内部寄存器
*****/
uchar ReadRegCmd(uchar RegAddr)
{
    uchar i;
    uchar temp = 0x00;

    miso = 1;
    mosi = 1;
    sck = 0;
    nss_c = 0;

    RegAddr <<= 0x01;
    RegAddr &= 0x3f;
    RegAddr |= 0x41;

    for(i = 0x00;i < 0x08;i++)
    {
        sck = 0;
        if(RegAddr&0x80)
        {
            mosi = 1;
        }
        else
        {
            mosi = 0;
        }
        sck = 1;
        RegAddr <<= 0x01;
    }
    sck = 0;
    mosi = 0;

    for(i = 0x00;i < 0x08;i++)
    {

```

```

        sck = 1;
        temp <<= 0x01;

        if(miso)
        {
            temp |= 0x01;
        }
        else
        {
            temp &= 0xfe;
        }

        sck= 0;
        DelayXms(1);
    }
    nss_c = 1;

    return(temp);
}
/*****
函数: SetSleepMode
功能: 设置为休眠模式
*****/
void SetSleepMode()
{
    uchar temp_cmd;
    temp_cmd = ReadRegCmd(0x00);
    temp_cmd &= 0x3f;
    WriteRegCmd(0x00,temp_cmd);
}
/*****
函数: SetSendMode
功能: 设置为发送模式
*****/
void SetSendMode()
{
    uchar temp_cmd;
    temp_cmd = ReadRegCmd(0x00);
    temp_cmd &= 0x3f;
    temp_cmd |= 0x80;
    WriteRegCmd(0x00, temp_cmd);
    TRISB1 = 0;        /* set data pin output */
}
/*****

```

函数: SetReceiveMode

功能: 设置为接收模式

*****/

```
void SetReceiveMode()
{
    uchar temp_cmd;
    temp_cmd = ReadRegCmd(0x00);
    temp_cmd &= 0x3f;
    temp_cmd |= 0x40;
    WriteRegCmd(0x00,temp_cmd);
    TRISB1    = 1;
}
```