

# 西安邮电大学

## 毕业设计（论文）

题目： 基于嵌入式的智能鱼缸设计与实现

学院： 自动化学院

专业： 自动化

班级： 自动 1403

学生姓名： 付祎诺

学号： 06141084

导师姓名： 张永健/马翔 职称： 高级工程师/讲师

起止时间： 2017 年 12 月 5 日 至 2018 年 6 月 10 日

## 毕业设计（论文）声明书

本人所提交的毕业论文《xxx》是本人在指导教师指导下独立研究、写作的成果，论文中所引用他人的文献、数据、图件、资料均已明确标注；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注明并表示感谢。

本人完全理解《西安邮电大学本科毕业设计（论文）管理办法》的各项规定并自愿遵守。

本人深知本声明书的法律责任，违规后果由本人承担。

论文作者签名：

日期：      年   月   日

## 西安邮电大学本科毕业设计(论文)选题审批表

申报人	马翔	职 称	讲师	学 院	自动化学院			
题目名称	基于嵌入式的智能鱼缸的设计与实现							
题目来源	科研				教学		其它	√
题目类型	硬件设计	√	软件设计		论文		艺术作品	
题目性质	应用研究		√		理论研究			
题目简述	手动方式养鱼是比较麻烦的事情,设计一种适合不同鱼种生存环境的智能鱼缸,该鱼缸可以智能完成对鱼缸的换水、控温、增氧,并可切换至不同鱼种的养殖环境,智能控制鱼缸保持在适宜鱼生存的水体环境。							
对学生知识与能力要求	掌握微机原理与接口技术及单片机原理及应用,理解嵌入式系统的基本组织结构与工作原理,具有一定的模拟电路技术基础及网络通信原理,熟悉常用传感器原理及使用方法,具有嵌入式软件设计能力及解决实际问题的动手能力。							
具体任务以及预期目标	设计智能鱼缸控制系统,可检测水体环境参数,根据预设场景,检测调节水温,定时增氧,换水,并通过显示屏显示实时状态。							
时间进度	2017 年 12 月 05 日—2017 年 12 月 10 日 选取毕设题目 2017 年 12 月 11 日—2018 年 01 月 06 日 查阅资料,撰写提交开题报告 2018 年 01 月 07 日—2018 年 03 月 04 日 确定系统架构设计方案 2018 年 03 月 05 日—2018 年 03 月 31 日 系统软件程序设计 2018 年 04 月 01 日—2018 年 04 月 15 日 软硬件的联合调试 2018 年 04 月 16 日—2018 年 04 月 31 日 系统功能完善 2018 年 05 月 01 日—2018 年 05 月 25 日 撰写毕业设计论文 2018 年 05 月 26 日—2018 年 06 月 01 日 修改、装订论文 2018 年 06 月 02 日—2018 年 06 月 10 日 准备毕业答辩							
系(教研室)主任签字	年 月 日			主管院长 签字		年 月 日		

# 西安邮电大学本科毕业设计（论文）开题报告

学号	06141084	姓名	付祎诺	导师	张永健/马翔
题目	基于嵌入式的智能鱼缸设计与实现				
选题目的（为什么选该课题） (1) 智能鱼缸是基于物联网技术实现家庭中科学养殖的一种新模式。面对现代化和科学化养殖要求，如何对鱼缸进行科学检测，将养殖环境控制在最佳状态成为了一个重要的环节。 (2) 手动养鱼是一件比较麻烦的事情，并且不够科学。设计一种适合不同鱼种生存的生态鱼缸，省时省力，并且科学。 (4) 当今市面上多数智能鱼缸都存在持续供氧的问题，持续供氧并不能够使鱼缸处于最优的状态。而本次智能鱼缸的设计也将改善此问题，根据监察数据供氧，做到最优养殖环境。 (3) 水质的优劣主要体现在水的温度、溶解氧和透明度，我们要通过加热设备、增氧设备、水循环设备等改善水质，达到最优的养殖环境。					
前期基础（已学课程、掌握的工具，资料积累、软硬件条件等） 1. 微机原理与接口技术 B 2. 单片机原理及应用 A 3. 模拟电子技术基础 A 4. 工业网络通信技术 5. 嵌入式控制系统 B 6. 传感器原理及应用					
要解决的问题（做什么） 1. 为了实现现代化养鱼，将要对养殖环境进行科学监测，将鱼缸内的环境控制在最佳状态。 2. 需要利用加热设备、增氧设备、水循环等设备来改善水质，实现水质参数的数据化，达到最佳的养殖环境，从而实现智能鱼缸的作用。					
工作思路和方案（怎么做） 1. 工作思路：通过嵌入式设计的基本步骤，使用软件仿真，编写程序完成嵌入式系统的设计，根据仿真结果焊接电路板并进行调试。 2. 方案：（1）硬件结构： 系统核心采用单片机，包括入水电磁阀、继电器、水温传感器、水氧传感器，增氧泵、加热棒。 单片机为整个智能鱼缸系统的控制核心，协调不同的模块运行，以实现预期的功能。鱼缸的供水通过单片机引脚连接三极管、继电器控制 DC12V 电磁入水阀，实现鱼缸的供水。 系统利用温度传感器监测鱼缸的水温并返回至单片机，单片机根据初始设定值通过继电器控制加热棒，实时改变鱼缸的水温。利用水氧传感器监测水体的氧浓度，并通过单片机控制增氧泵工作，实时改变水体氧浓度，以适应不同鱼类的生长环境。整个鱼缸系统连接显示器，实时显示鱼缸中水体的温度、氧浓度数值以及当前适应生存的鱼类。					

(2) 软件结构:

系统开始运行时用户通过与显示屏交互选择不同鱼类的生长环境,系统中包含所选择的生长环境对应的水体参数值,整个系统运行时,通过传感器实时监测水体环境并传送回单片机与设定的初始值进行比较,在监测到参数发生变化时,通过单片机控制对应的模块改变水体环境,例如水温过低时启动增温设备,氧浓度过低时启动增氧设备,实现水质的自动调整,通过单片机实时将整个系统的运行状态在显示屏显示以供用户了解。

参考文献:

- [1]冯.超 基于嵌入式 Linux 的自助洗衣房集中控制器的设计[D]. 太原理工大学,2016.
- [2]刘伟,林开司,刘安勇. 基于物联网的鱼缸智能控制系统设计与实现[J]. 淮海工学院学报(自然科学版),2016,(4):155-156. doi:10.3969/j.issn.1672-6685.2016.04.001.
- [3]丁惠忠. 观赏鱼缸智能控制系统的设计[D]. 2007:36-37.

指导教师意见

签字: 2018 年 1 月 9 日

# 西安邮电大学毕业设计（论文）成绩评定表

学生姓名		性别		学号		专业 班级	
课题名称							
指导教师 意见	（从开题论证、论文内容、撰写规范性、学习态度、创新等方面进行考核）  评分（百分制）：      指导教师(签字)： _____ 年 月 日						
评阅 教师 意见	（从选题、开题论证、论文内容、撰写规范性、创新和预期成果等方面进行考核）  评分（百分制）：      评阅教师(签字)： _____ 年 月 日						
验收 小组 意见	（从毕业设计质量、准备、操作情况等方面进行考核）  评分（百分制）：      验收教师(签字)： _____ 年 月 日						
答辩 小组 意见	（从准备、陈述、回答、仪表等方面进行考核）  评分（百分制）：      答辩小组组长(签字)： _____ 年 月 日						
评分比例	指导教师评分    (%)    评阅教师评分    (%)    验收小组评分    (%)    答辩小组评分    (%)						
学生总评 成绩	百分制成绩				等级制成绩		
答辩委员 会意见	毕业论文(设计)最终成绩(等级)：  学院答辩委员会主任(签字、学院盖章)： _____ 年 月 日						

# 目 录

第一章 引言 .....	1
1.1 课题背景 .....	1
1.2 课题任务 .....	1
1.3 论文结构 .....	2
第二章 背景知识.....	3
2.1 STC89C52 单片机介绍.....	3
2.2 继电器介绍.....	5
第三章 设计方案与功能介绍 .....	8
3.1 系统方案 .....	8
3.2 系统功能 .....	8
3.2.1 环境监.....	9
3.2.2 自动升温功能.....	9
3.2.3 定时控制增氧功能.....	9
3.2.4 人机交互控制进出水功能 .....	9
3.3 设计原则.....	9
第四章 系统硬件设计 .....	10
4.1 整体硬件方案.....	10
4.2 主控器最小系统硬件设计.....	10
4.3 鱼缸中水的温度检测模块电路设计.....	12
4.3.1 温度传感器介绍 .....	12
4.3.2 温度传感器电路设计 .....	12
4.3.3 驱动模块电路设计.....	13
4.4 人机交互模式电路设计.....	13
4.4.1 显示模块电路设计 .....	13
4.4.2 按键模块电路设计 .....	16
4.5 硬件设计环境.....	18
4.6 硬件设计总结.....	19
第五章 系统软件设计.....	20
5.1 主程序软件设计.....	20
5.2 温度采集模块程序设计.....	22

5.3 升温模块程序设计.....	23
5.4 增氧模块程序设计.....	23
5.5 软件设计环境.....	24
5.6 软件设计总结.....	25
第六章 总结 .....	26
致谢.....	28
结束语 .....	29
参考文献 .....	29
附录.....	29



## 摘 要

近年来嵌入式发展迅速,智能家居也进入了人们的生活。智能鱼缸却没有快速发展。我根据市场上相关设备现状的分析和研究,从系统集成角度进行设计和开发,提出了一套多功能的智能观赏鱼缸控制系统的设计方案。

该控制系统以 STC89C52 单片机为控制核心,结合传感器技术,集多种控制功能于一体,包括检测温度、自动升温、自动充氧、自动换水等。整个系统是以 STC12C5A60S2 为核心的控制部分,实现对各种控制参数的设置、存储、和处理。

该系统根据当前市场上的需求进行设计和开发,形成了一套自动充氧、自动升温、自动换水等功能为一体的集成控制系统。同时该系统设计灵活、成本低廉,便于量产,可广泛用于家庭和宾馆等安装观赏鱼缸的场所。

**关键词:**鱼缸,自动控制,单片机

## ABSTRACT

In recent years, with the development of embedded, intelligent household also entered people's lives. The intelligent aquarium is not development. I according to the current market analysis and research status quo of related equipment, system integration from the perspective of design and development, a set of multi-functional intelligent fish tank control system design.

The control system STC12C5A60S2 microcontroller core. Combined with sensor technology. Set a variety of control functions. Including temperature detection, automatic oxygenation, automatic water change, automatic temperature. The whole system is divided into STC12C5A60S2 the control part of the core, to achieve a variety of control parameters settings, storage, and processing.

The system is designed and developed according to the needs of the current market. The formation of an automatic oxygenation, automatic water change, automatic temperature into one integrated control system. Meanwhile, the system is designed to be flexible, low cost, ease of mass production, it can be widely used in homes and hotels and other places to install aquarium.

**Key words:** Aquarium; automatic control; single-chip

## 第一章 引言

### 1.1 课题背景

近些年来，在繁华的都市生活里，介于人们快速的生活节奏和过分繁忙的工作，飞速发展的时代伴随着的对生活质量的要求循序渐进的提升，智能家电也成为了人们密切需要的技术。养鱼是很多家庭中都存在的事情，但是传统的养鱼手段还是比较繁琐的。而智能鱼缸则是一种省时省力的养殖手段，它是基于物联网技术实现家庭中科学养殖的一种新模式，面对现代化和科学化养殖要求，如何对鱼缸进行科学检测，并将对养殖环境进行准确性控制成为了非常重要的一个任务。因此设计并做出一个适合水生动植物生存的智能鱼缸，能够达到科学养殖的目的还能解放人的双手。水质的优劣主要体现在水的温度、溶解氧和透明度，我们要通过加热设备、增氧设备、水循环设备等改善水质，达到最优的养殖环境。

基于嵌入式的智能鱼缸设计与实现这个课题，采纳了我们大学四年所学习的有关于嵌入式的相关知识，和基本的编写程序的能力，让我们通过所学习到的知识制作出一个可以利用水泵自动进出水以达到换水的目的；采用增氧泵、按键和 LCD 显示屏可以给缺氧的鱼缸提供水氧；采用加热棒、按键和 LCD 显示屏可以给温度过低的鱼缸升温，将鱼缸内的温度达到适合鱼群生活的最适温度。我们的系统设计采用了 STC89C52 单片机，该单片机是一个建立于非常低的功耗（节能）、效果好还具有 CMOS8 位的微控制器，这些让我们将要使用的 STC89C52 芯片能够为我们将要设计的智能鱼缸的大多数嵌入式控制系统，为智能鱼缸提供更高效能的解决手段用来制作出我们题目所要求的可以提供多方面智能服务的鱼缸。

该课题在剖析当前市场上和传统的鱼缸的优缺点之后，想要解决传统鱼缸的弊端，主要的缺点有：普通的鱼缸存在非常严重的缺氧现象，因为用户的疲劳以及种种原因忘记换水以及不能随意调节鱼缸中水的温度。

### 1.2 课题任务

本课题主要研究基于嵌入式的智能鱼缸设计与实现，主要实现了智能化照顾水生动植物的功能。该设计采用低功耗处理器 STC89C52，其较强的处理能力以及其具有低功耗的特点。设计使用了传感器技术用以实现水生环境信息的采集，通过与给定的环境作比较，单片机可以做出相对应的判断，从而创造出适宜水生动植物生长的环境。单片机将当前水生动植物生长的实时情况通过 LCD 显示屏显示，同时用户可通过显示屏自定义时间通过继电器控制增氧和换水的周期。更加人性化的设计，能够让用户在养殖中找到乐趣，同时又能让水生动植物健康的生长。

本课题主要针对水环境中的氧含量、温度以及如何换水进行了研究，以智能化的技术手段向水中的动植物提供了最适宜的生长环境。系统以 STC89C52 作

为主控制器，通过温度传感器，作为主要的信息采集，将当前环境信息传送至主控制器，由主控制器做出相应判断后，发送指令至加热棒，进行相应的增温。人机交互模式中，用户可通过自定义时间让继电器发送进出水、增氧的指令。

### 1.3 论文结构

本文各章节安排如下：

第一章：引言，主要介绍课题背景；

第二章：主要介绍各模块的元器件；

第三章：硬件设计，主要讲解硬件设计环境以及各个模块的硬件设计原理图与硬件描述；

第四章：软件设计，编译环境介绍，讲解主要程序，各功能模块的程序设计；

第五章：调试结果，包括软件调试结果和硬件调试结果；

第六章：总结，总结毕业设计过程中遇到的问题和最终的收获。

## 第二章 背景知识

### 2.1 STC89C52 单片机介绍

#### 2.1.1 STC89C52 单片机简介

STC89C52 单片机是 STC 公司生产的一种低消耗、高性能 CMOS 8 位微控制器，具有八千字节系统可编程 Flash 存储器。STC89C52 使用了最经典的 MCS-51 内核它通过了多方面的改进让芯片具有了传统 51 单片机不具备的功能。并且在单芯片上拥有灵巧的八位 CPU 和在系统可编程 Flash，使得 STC89C52 给众多嵌入式控制应用系统提供更灵活、更有效的解决方法。其实物图如图 2.1。



图 2.1 STC89C52 实物图

#### 2.1.2 STC89C52 单片机标准功能

具有以下标准功能：8k 字节 Flash，512 字节 RAM，32 位 I/O 口线，看门狗定时器，内置 4KB EEPROM，MAX810 复位电路，3 个 16 位定时器/计数器，4 个外部中断，一个 7 向量 4 级中断结构（兼容传统 51 的 5 向量 2 级中断结构），全双工串行口。另外 STC89C52 可降至 0Hz 静态逻辑操作，支持 2 种软件可选择节电模式。空闲模式下，CPU 停止工作，允许 RAM、定时器/计数器、串口、中断继续工作。掉电保护方式下，RAM 内容被保存，振荡器被冻结，单片机一切工作停止，直到下一个中断或硬件复位为止。最高运作频率 35MHz，6T/12T 可选。其引脚图如图 2.2 所示。

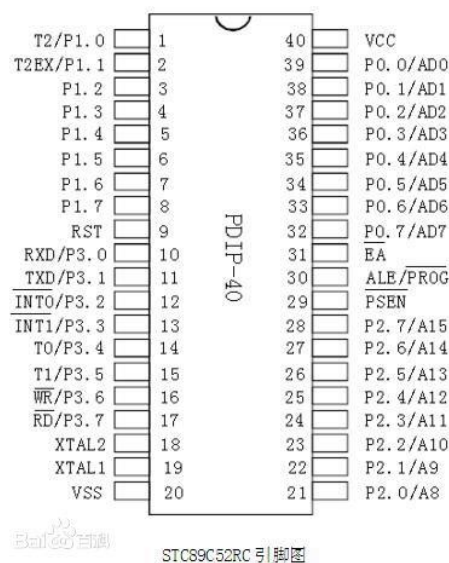


图 2.2 STC8952RC 引脚图

### 2.1.3 STC89C52 单片机器件参数

(1) 增强型 8051 单片机，6 时钟/机器周期和 12 时钟/机器周期可以任意选择，指令代码完全兼容传统 8051。 [1]

(2) 工作电压：5.5V~3.3V（5V 单片机）/3.8V~2.0V（3V 单片机）

(3) 工作频率范围：0~40MHz，相当于普通 8051 的 0~80MHz，实际工作频率可达 48MHz

(4) 用户应用程序空间为 8K 字节

(5) 片上集成 512 字节 RAM

(6) 通用 I/O 口（32 个），复位后为：P1/P2/P3 是准双向口/弱上拉，P0 口是漏极开路输出，作为总线扩展用时，不用加上拉电阻，作为 I/O 口用时，需加上拉电阻。

(7) ISP（在系统可编程）/IAP（在应用可编程），无需专用编程器，无需专用仿真器，可通过串口（RxDP3.0, TxDP3.1）直接下载用户程序，数秒即可完成一片

(8) 具有 EEPROM 功能

(9) 共 3 个 16 位定时器/计数器。即定时器 T0、T1、T2

(10) 外部中断 4 路，下降沿中断或低电平触发电路，Power Down 模式可由外部中断低电平触发中断方式唤醒

(11) 通用异步串行口（UART），还可用定时器软件实现多个 UART

(12) 工作温度范围：-40~+85℃（工业级）/0~75℃（商业级）

(13) PDIP 封装

## 2.2 继电器介绍

### 2.2.1 继电器简介

继电器（英文名称：relay）是一种电控制器件，是当输入量（激励量）的变化达到规定要求时，在电气输出电路中使被控量发生预定的阶跃变化的一种电器。它具有控制系统（又称输入回路）和被控制系统（又称输出回路）之间的互动关系。通常应用于自动化的控制电路中，它实际上是用小电流去控制大电流运作的一种“自动开关”。故在电路中起着自动调节、安全保护、转换电路等作用。其实物图如图 2.3 所示。



图 2.3 继电器实物图

### 2.2.2 继电器的主要作用

继电器是具有隔离功能的自动开关元件，广泛应用于遥控、遥测、通讯、自动控制、机电一体化及电力电子设备中，是最重要的控制元件之一。

继电器一般都有能反映一定输入变量（如电流、电压、功率、阻抗、频率、温度、压力、速度、光等）的感应机构（输入部分）；有能对被控电路实现“通”、“断”控制的执行机构（输出部分）；在继电器的输入部分和输出部分之间，还有对输入量进行耦合隔离，功能处理和对输出部分进行驱动的中间机构（驱动部分）。作为控制元件，概括起来，继电器有如下几种作用：

（1）扩大控制范围：例如，多触点继电器控制信号达到某一定值时，可以按触点组的不同形式，同时换接、开断、接通多路电路。

（2）放大：例如，灵敏型继电器、中间继电器等，用一个很微小的控制量，可以控制很大功率的电路。

（3）综合信号：例如，当多个控制信号按规定的形式输入多绕组继电器时，经过比较综合，达到预定的控制效果。

（4）自动、遥控、监测：例如，自动装置上的继电器与其他电器一起，可以组成程序控制线路，从而实现自动化运行。

## 2.2.3 继电器的主要分类

### 1. 按继电器的工作原理或结构特征分类

(1) 电磁继电器：利用输入电路内电路在电磁铁铁芯与衔铁间产生的吸力作用而工作的一种电气继电器。

(2) 固体继电器：指电子元件履行其功能而无机械运动构件的，输入和输出隔离的一种继电器。

(3) 温度继电器：当外界温度达到给定值时而动作的继电器。

(4) 舌簧继电器：利用密封在管内，具有触电簧片和衔铁磁路双重作用的舌簧动作来开，闭或转换线路的继电器

(5) 时间继电器：当加上或除去输入信号时，输出部分需延时或限时到规定时间才闭合或断开其被控线路继电器。

(6) 高频继电器：用于切换高频，射频线路而具有最小损耗的继电器。

(7) 极化继电器：有极化磁场与控制电流通过控制线圈所产生的磁场综合作用而动作的继电器。继电器的动作方向取决于控制线圈中流过的的电流方向。

(8) 其他类型的继电器：如光继电器，声继电器，热继电器，仪表式继电器，霍尔效应继电器，差动继电器等。

### 2、按继电器的外形尺寸分类

(1) 微型继电器

(2) 超小型微型继电器

(3) 小型微型继电器

注：对于密封或封闭式继电器，外形尺寸为继电器本体三个相互垂直方向的最大尺寸，不包括安装件，引出端，压筋，压边，翻边和密封焊点的尺寸。

### 3、按继电器的负载分类

(1) 微功率继电器

(2) 弱功率继电器

(3) 中功率继电器

(4) 大功率继电器

### 4、按继电器的防护特征分类

(1) 密封继电器

(2) 封闭式继电器

(3) 敞开式继电器

### 5、按继电器按照动作原理可分类

(1) 电磁型

(2) 感应型



- (3) 整流型
- (4) 电子型
- (5) 数字型等

6、按照反应的物理量可分类

- (1) 电流继电器
- (2) 电压继电器
- (3) 功率方向继电器
- (4) 阻抗继电器
- (5) 频率继电器
- (6) 气体（瓦斯）继电器

7、按照继电器在保护回路中所起的作用可分类

- (1) 启动继电器
- (2) 量度继电器
- (3) 时间继电器
- (4) 中间继电器
- (5) 信号继电器
- (6) 出口继电器

### 第三章 设计方案与功能介绍

#### 3.1 系统方案

智能鱼缸的设计主要包括：信息采集模块（温度检测模块）、显示模块、增氧模块、进出水模块以及手动定时模块。其体系结构如图 3.1 所示。

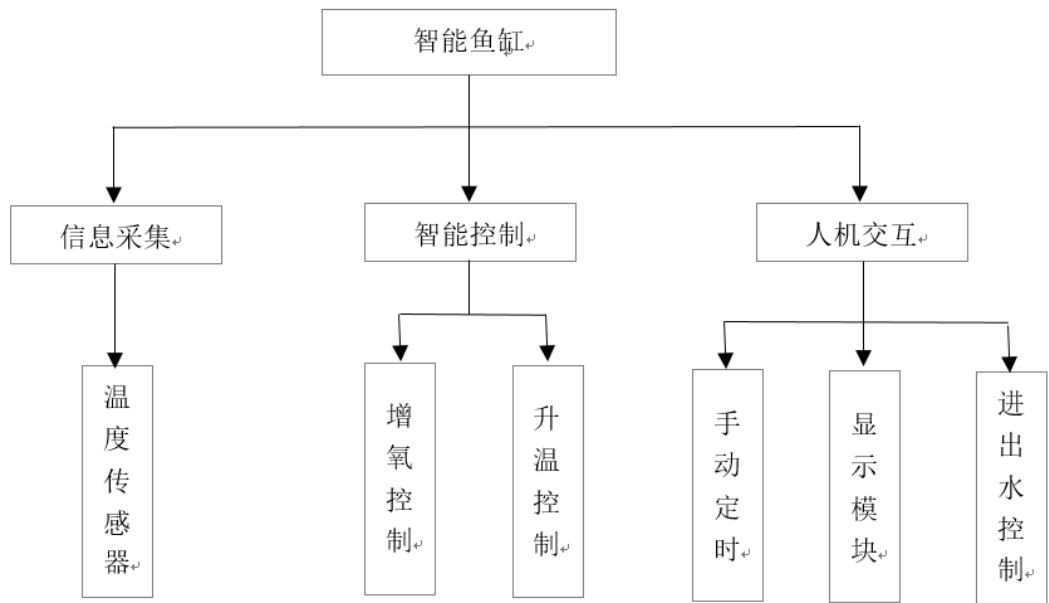


图 3.1 智能鱼缸体系结构图

系统通过信息采集模块获取当前鱼缸内水的温度后，将该数据和已经存储的存储数据进行比较，让单片机做出智能判断。如果当时的温度不是最适温度时，由单片机发出控制指令发送给继电器，再由继电器发送到升温棒使温度上升到既定温度后停止，达到升温的目的。通过按钮发送到继电器可设置增氧的循环时间，再由继电器发送到增氧泵进行增氧，达到增氧的目的。通过人机交互控制继电器，再由继电器发送指令至水泵，控制水泵进出水，达到换水的目的。智能鱼缸设置人机交互模式可以实现手动设置时间、选择温度和显示当前温度的功能。

#### 3.2 系统功能

智能鱼缸的关键技术在于将温度信息准确化采集以及智能控制端对升温、增氧、换水的控制，系统的稳定性、准确性是整个系统的核心。本设计中使用了传感器技术与单片机技术，选用了高准确性的温度传感器与低功耗的单片机，实现了自动增氧、自动调节温度和自动换水的功能。

系统主要通过传感器获取的数据和存储的信息融合后，做出信息化判断，并进行相应的增氧、升温和换水措施。除此之外，系统还配有显示屏，显示屏可以显示当前温度信息、设置的温度信息和时间信息。用户可以通过按键实现进出水

的功能，更加人性化的满足养鱼的乐趣。

### 3.2.1 环境监测功能

系统可以通过检测鱼缸内水的温度来判断当前水生动植物的生长状态。

### 3.2.2 自动升温功能

系统可以通过检测鱼缸内水的温度将该数据和经过按键设置的温度进行比较，让单片机做出智能判断，对鱼缸进行升温，使温度达到适合水生动植物的合适生存温度。系统具有操作简洁、实现自动化程度高、升温效率高、能耗低的特点，可满足水生动植物对水温度的不同喜好。

### 3.2.3 定时控制增氧功能

系统可以通过按键设置对鱼缸增氧的循环时间，根据水生动植物的需求，对其进行适当的氧气补充，能够保证水生动植物处于适当的水氧环境中。

### 3.2.4 人机交互控制进出水功能

因为换水需要根据水生动植物不同时间段的状态及时更换鱼缸内的水。所以显示屏下方通过设置按钮可以设置进出水达到换水的目的。使养鱼过程中的换水变得方便快捷，使用户在养鱼中找到乐趣。

## 3.3 设计原则

低功耗设计：STC89C52 单片机是 STC 公司生产的一种低消耗、高性能 CMOS8 位微控制器。使智能鱼缸达到低消耗的目的。

模块化设计：该系统设计采用了模块化的设计方案，主要分为监测模块、控制驱动模块以及显示模块；监测模块中含有温度传感器模块用来监测鱼缸中水的温度；控制驱动模块分为升温模块、增氧模块、进出水模块、定时模块。通过模块化的设计思想提升了系统的稳定性和可靠性和系统中相互间的紧密联系，更好的完成了系统预定功能。

稳定性设计：利用温湿度传感器可实时监控水中动植物所处的环境温度，并具备自定义温度模式，可以根据水生动植物不同时期对温度的需要进行调整。

## 第四章 系统硬件设计

### 4.1 整体硬件方案

系统硬件设计采用模块化的电路设计，以微控制器 STC89C52 单片机为设计的控制中心，以时钟电路模块、监测模块、显示模块、控制驱动模块和电源管理模块等功能模块。完成了传感器、环境监测、人机交互、控制驱动等方面的设计，系统的整体设计如图 4.1 所示。

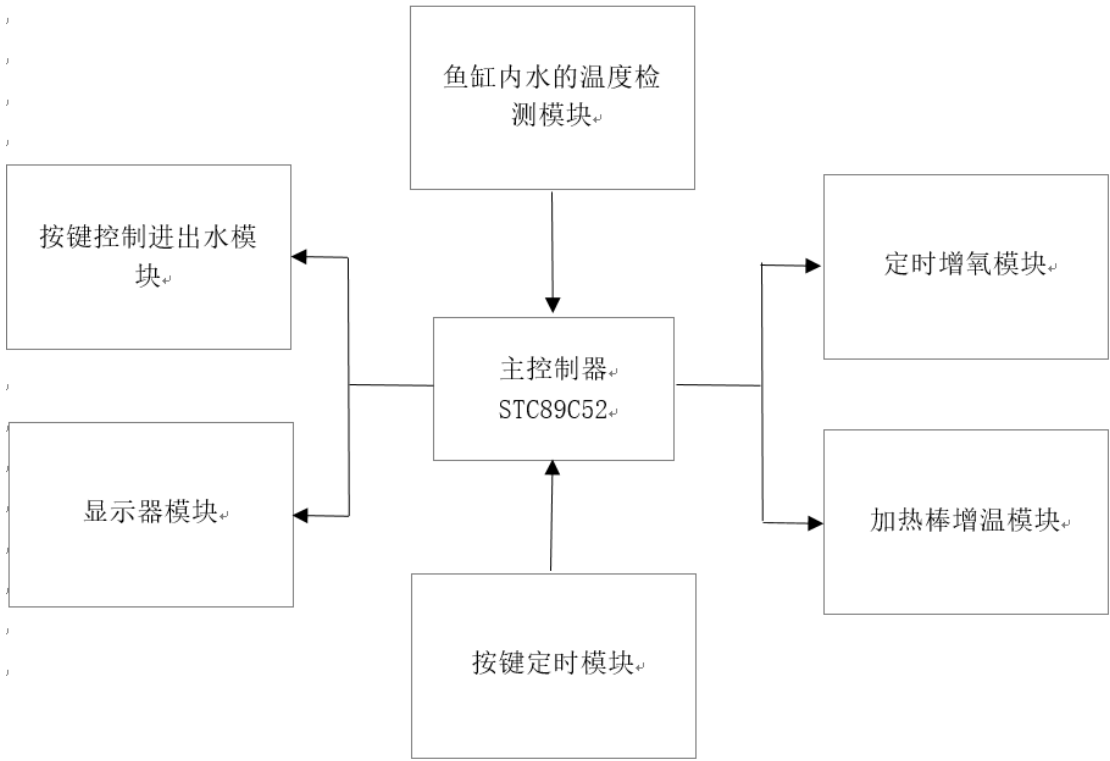


图 4.1 系统的整体设计图

### 4.2 主控制器最小系统硬件设计

单片机的最小系统为本硬件系统设计的核心主控制模块，完成系统各项任务的控制，本着稳定、可靠的原则进行电路设计。主控制器单片机外围电路设计如图 4.2 所示。

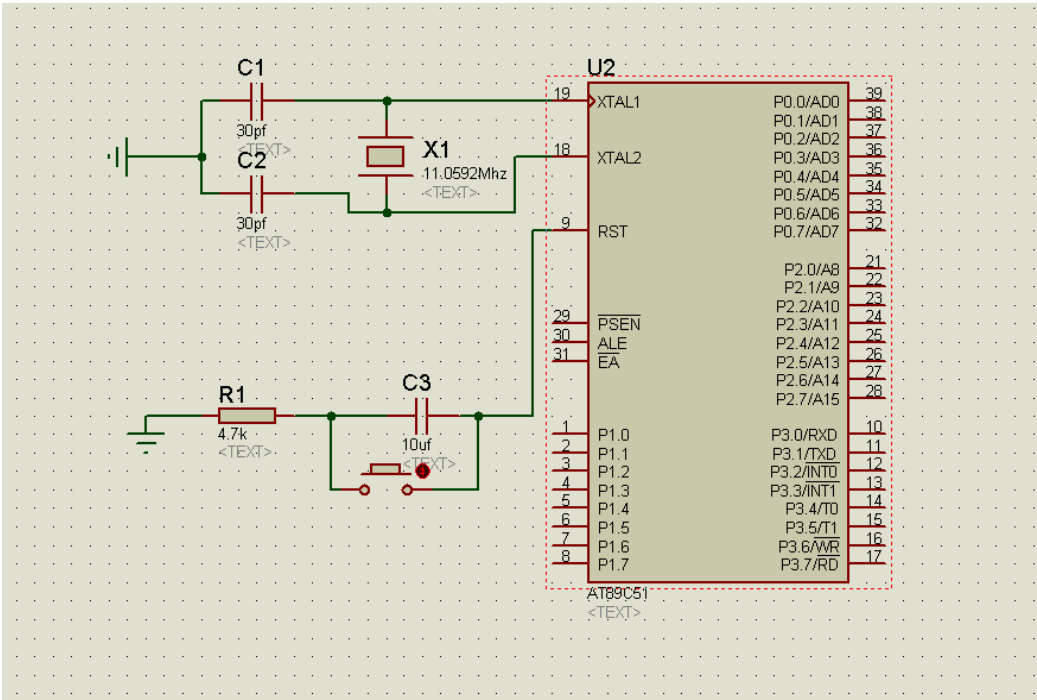


图 4.2 主控制器原理图

(1) 时钟电路设计  
时钟电路设技图如图 4.3 所示

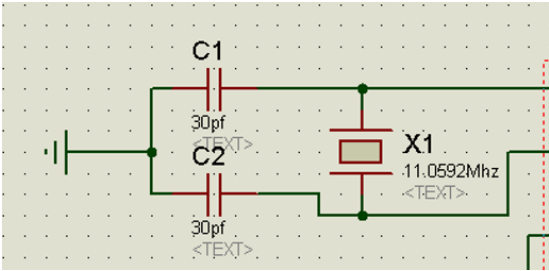


图 4.3 时钟电路图

(2) 下载电路设计  
单片机可利用连线进行下载，其电路设计如图 4.4 所示。



图 4.4 下载电路图

(3) 复位电路设计  
复位电路设计如图 4.5 所示。

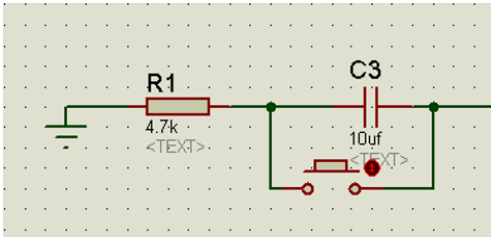


图 4.5 复位电路图

4.3 鱼缸中水的温度检测模块电路设计

4.3.1 温度传感器介绍

鱼缸中水的温度监测模块采了最常用的测量温度传感器 DS18B20 数字温度传感器，该传感器实物图如图 4.6 所示。



图 4.6 温度传感器 DS18B20 实物图

该传感器探头采用 DS18B20 温度传感器芯片，芯片每个引脚均用热缩管隔开，防止短路，内部封胶，提高了温度传感器的安全性能。

4.3.2 温度传感器电路设计

图 4.7 为温度传感器的电路设计

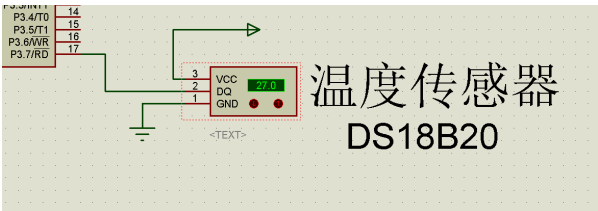


图 4.7 鱼缸中水的温度监测模块电路图

温度传感器的性能参数表如表 4.1 所示。

表 4.1 温度传感器性能参数表

供给电压	3.0~5.5V
可调分辨率	9~12 位
感温范围宽	-55~+125 度

尺寸	6mm*50mm
----	----------

4.3.3 驱动模块电路设计

驱动模块通过用继电器来控制增氧泵、升温棒和水泵，达到给鱼缸增氧、升温 and 换水的目的。继电器是一种电控制器件。它具有控制系统（又称输入回路）和被控制系统（又称输出回路）之间的互动关系,通常应用于自动化的控制电路中，它实际上是用小电流去控制大电流运作的一种“自动开关” [2]。系统采用 4 路 5V 带光耦隔离继电器模块，其实物图如图 4.8 所示。

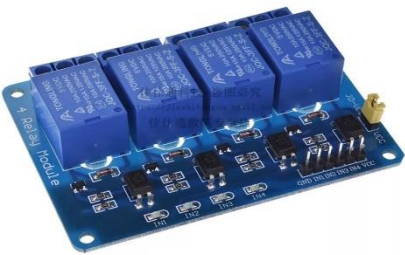


图 4.9 4 路 5V 带光耦隔离继电器模块实物图

该继电器的主要特性为价格低廉，采用双面 FR-4 线路板设计，高端贴片工艺生产。有四个常开和 4 个常闭触点。其电路原理图如图 4.10 所示。

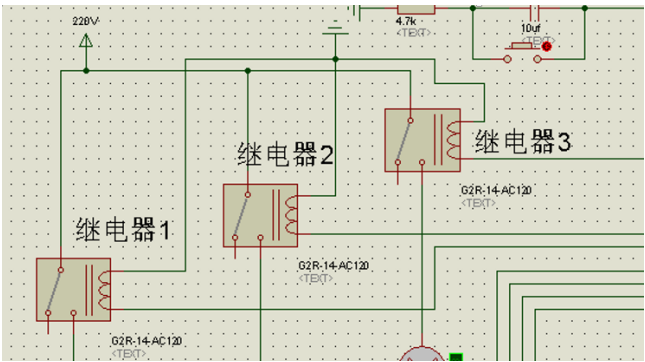


图 4.10 继电器电路图

由图 4.10 可知，4 路继电器模块，本次设计只使用了左侧的三路继电器，从左至右分别用于控制变压器、增氧泵和升温棒。

4.4 人机交互模式电路设计

4.4.1 显示模块电路设计

### (1) LCD1602 液晶屏简介

1602 液晶也叫 1602 字符型液晶，它是一种专门用来显示字母、数字、符号等的点阵型液晶模块。它由若干个 5X7 或者 5X11 等点阵字符位组成，每个点阵字符位都可以显示一个字符，每位之间有一个点距的间隔，每行之间也有间隔，起到了字符间距和行间距的作用，正因为如此所以它不能很好地显示图形（用自定义 CGRAM，显示效果也不好）。

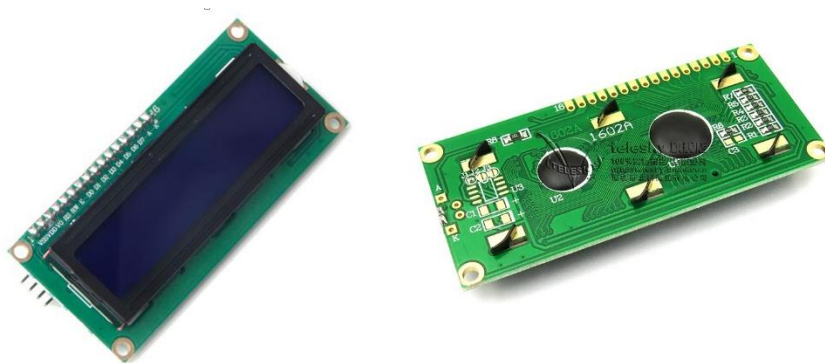


图 4.11 LCD1602 液晶屏实物图

### (2) LCD1602 液晶屏管脚功能

第 1 引脚：GND 为电源地

第 2 引脚：VCC 接 5V 电源正极

第 3 引脚：VO 为液晶显示器对比度调整端，接正电源时对比度最弱，接地电源时对比度最高（对比度过高时会产生“鬼影”，使用时可以通过一个 10K 的电位器调整对比度）。

第 4 引脚：RS 为寄存器选择，高电平 1 时选择数据寄存器、低电平 0 时选择指令寄存器。

第 5 引脚：RW 为读写信号线，高电平(1)时进行读操作，低电平(0)时进行写操作。

第 6 引脚：E(或 EN)端为使能(enable)端,高电平（1）时读取信息，负跳变时执行指令。

第 7~14 引脚：D0~D7 为 8 位双向数据端。

第 15~16 脚：空脚或背灯源。

第 15 引脚背光正极，

第 16 引脚背光负极。





图 4.12 LCD1602 液晶屏引脚图

### (3) LCD1602 液晶屏的特点

- a. 3.3V 或 5V 工作电压，对比度可调
- b. 内含复位电路
- c. 提供各种控制命令,如：清屏、字符闪烁、光标闪烁、显示移位等多种

功能

- d. 有 80 字节显示数据存储单元 DDRAM
- e. 内建有 192 个 5X7 点阵的字型的字符发生器 CGROM
- f. 8 个可由用户自定义的 5X7 的字符发生器 CGRAM
- g. 微功耗、体积小、显示内容丰富、超薄轻巧，常用在袖珍式仪表和低功耗应用系统中。

LCD1602 显示屏的电路图如图 4.13 所示。

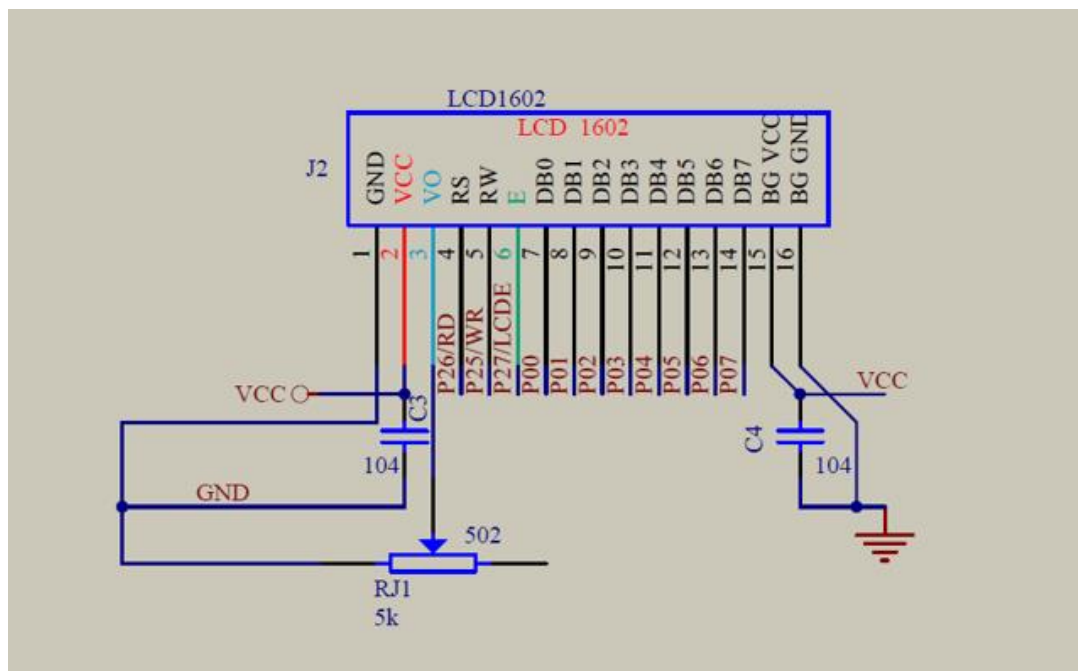


图 4.13 LCD1602 显示屏

4.4.2 按键模块电路设计

嵌入式智能鱼缸系统设计中，可以通过人工定时设置增氧泵的增氧循环时间、设置增温棒的升温温度、控制进出水，以达到增氧、升温 and 换水的目的。从人性化设计的角度考虑，最终设计 4 个独立式按键，电路设计图如 4.23 所示。

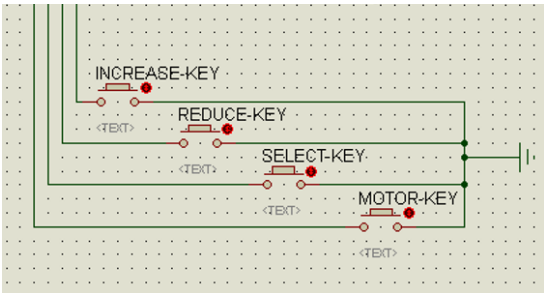


图 4.13 按键定时模块电路设计

温度寄存器格式图如图 4.14 所示。

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LS Byte	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$
	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
MS Byte	S	S	S	S	S	$2^6$	$2^5$	$2^4$

图 4.14 温度寄存器格式图

(1) 通过按钮控制定时增氧

系统可以通过按钮控制增氧的循环时间，并通过显示屏显示出来，达到人机交互的目的。也使增氧变得简单易操作，并可以根据实时的状态改变增氧时间，增加了养鱼的乐趣。增氧泵的实物图如图 4.15 所示。



图 4.15 YTZ-311 增氧泵的实物图

YTZ-311 增氧泵性能参数表如表 4.2 所示。

表 4.2 YTZ-311 增氧泵性能参数表

产 品 型 号	YTZ-311
额 定 功 率	2.5W
额 定 电 压	220-240V
频 率	50HZ
出 气 量	1.8L/min
尺 寸	11.5X6X5cm
压 力	15kpa
电 压	220V
电 源 线 长	1m
净 重	280g

(2) 通过按钮设置鱼缸温度

系统可以通过按钮,设置鱼缸中水的温度,使温度传感器和增温棒相互配合,让鱼缸达到用户所设置的最适温度。YGH-25 增温棒实物图如图 4.16 所示。



图 4.16 YGH-25 石英加热棒实物图

YGH-25 石英加热棒性能参数表如表 4.3 所示。

表 4.3 YGH-25 石英加热棒性能参数表

型 号	YGH-25
功 率	25w
尺 寸	20cm
线 长	85cm
电 压	220V
赫 兹	50-60Hz

温 控 范 围	12-34 度
---------	---------

### (3) 通过按钮控制进出水

因为鱼缸换水的时间并不是一个相对固定的时间,而是要根据鱼缸不同时期的状态来决定的。因此我们将鱼缸的换水功能通过按钮来进行控制。按下按钮时,水泵开始进行进出水达到换水的目的。我们设置自动换水这个功能不仅仅是因为方便,在日常生活中,鱼从水池的下水道溜走的事情时常出现在手动换水中。也是为了避免这一情况,我们设计了自动换水这一环节。水泵实物图如图 4.17 所示。



图 4.17 水泵实物图

## 4.5 硬件设计环境

Proteus 软件是英国 Lab Center Electronics 公司出版的 EDA 工具软件。它不仅具有其它 EDA 工具软件的仿真功能,还能仿真单片机及外围器件。它是目前比较好的仿真单片机及外围器件的工具。Proteus 是英国著名的 EDA 工具(仿真软件),从原理图布图、代码调试到单片机与外围电路协同仿真,一键切换到 PCB 设计,真正实现了从概念到产品的完整设计。是目前世界上唯一将电路仿真软件、PCB 设计软件和虚拟模型仿真软件三合一的设计平台,其处理器模型支持 8051、HC11、PIC10/12/16/18/24/30/DsPIC33、AVR、ARM、8086 和 MSP430 等,2010 年又增加了 Cortex 和 DSP 系列处理器,并持续增加其他系列处理器模型。在编译方面,它也支持 IAR、Keil 和 MPLAB 等多种编译器。硬件开发界面如图 4.18 所示。

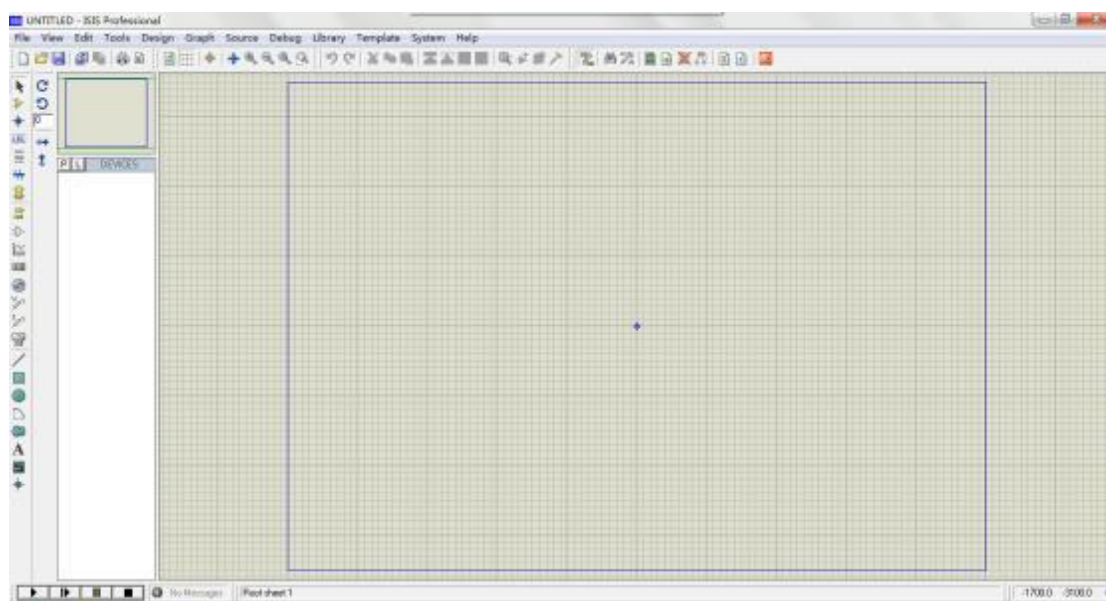


图 4.18 硬件开发界面

## 4.6 硬件设计总结

嵌入式的智能鱼缸的设计与实现所用到的硬件有：DS18B20 温度传感器、LCD1602 液晶显示屏、YTZ-311 增氧泵、YGH-25 石英加热棒、水泵、4 路 5V 继电器、STC89C52 单片机等，通过查找各芯片的数据手册了解芯片的相关参数以及性能进行了解，利用 Keil 进行电路设计，对每个模块进行合理的布局以及低功耗电路设计。在进行硬件设计的过程中，进行了大量的电路布局工作和调试工作，使得硬件设计的可靠性、稳定性有了极大的提升。

## 第五章 系统软件设计

软件设计根据各单元的实现功能分为：监测单元软件设计、驱动控制单元软件设计、显示单元软件设计和定时单元软件设计。

### 5.1 主程序软件设计

系统开始运行时用户通过与显示屏交互选择不同鱼类的生长环境，系统中包含所选择的生长环境对应的水体参数值，整个系统运行时，通过传感器实时监测水体环境并传送回单片机与设定的初始值进行比较，在监测到参数发生变化时，通过单片机控制对应的模块改变水体环境，例如水温过低时启动增温设备，氧浓度过低时启动增氧设备，实现水质的自动调整，通过单片机实时将整个系统的运行状态在显示屏显示以供用户了解。、

主程序软件设计如图 5.1 所示：

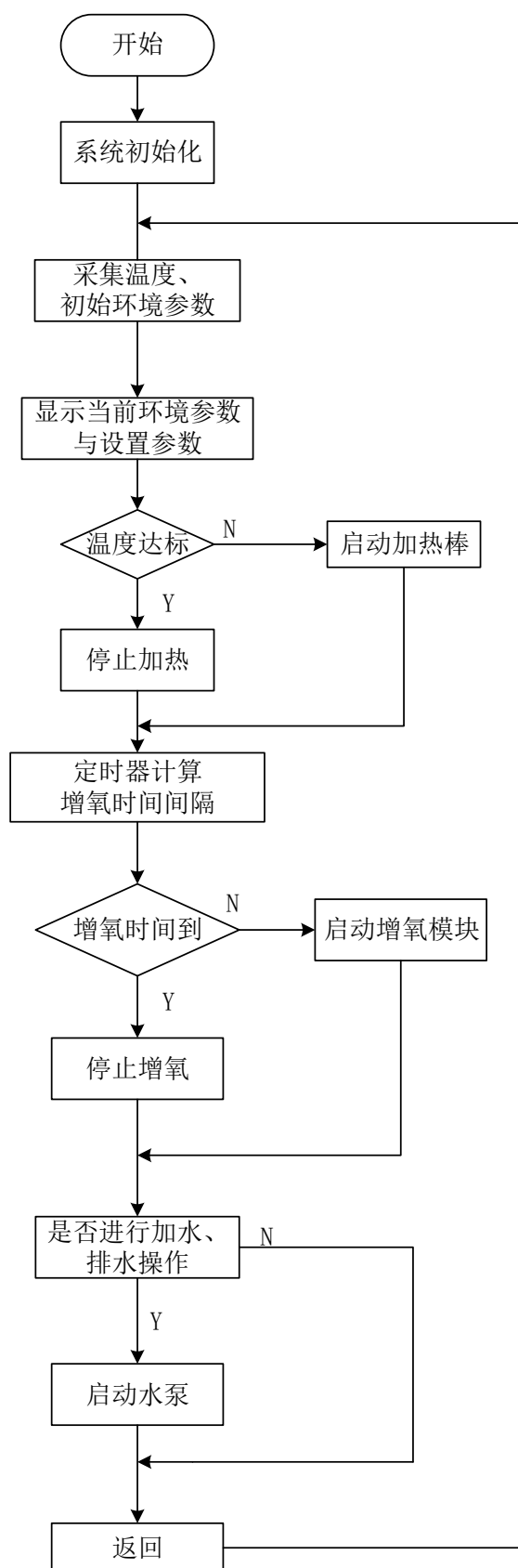


图 5.1 主程序设计流程图

系统进行初始化，分别判断并采集鱼缸中水的温度的信息，当环境信息不达

标时，启动增温棒升温，并且可以通过按钮设置增氧循环时间、对鱼缸中的水的升温温度和通过按钮控制进出水，达到换水的目的。并将这些参数实时显示到 LCD 显示屏上。

## 5.2 温度采集模块程序设计

DS18B20 温度传感器设计流程图如图 5.2 所示。

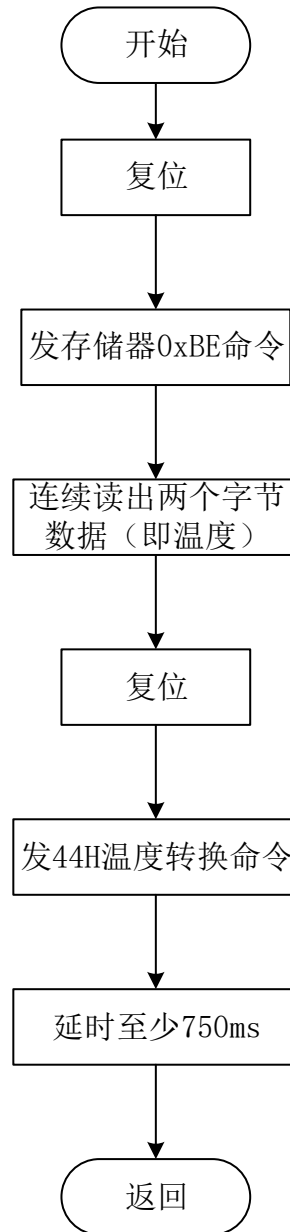


图 5.2 DS18B20 温度传感器设计流程图

当向温度模块发送存储器 0xBE 命令时，连续读出两个字节数据（即温度），再发送 44H 温度转换命令，使温度传感器延时至少 750ms，最后模块将温度值反馈至 LCD1602 液晶显示屏。



### 5.3 升温模块程序设计

升温模块程序依据当前温度是否达标作为依据的。升温模块设计流程图如图 5.3 所示。

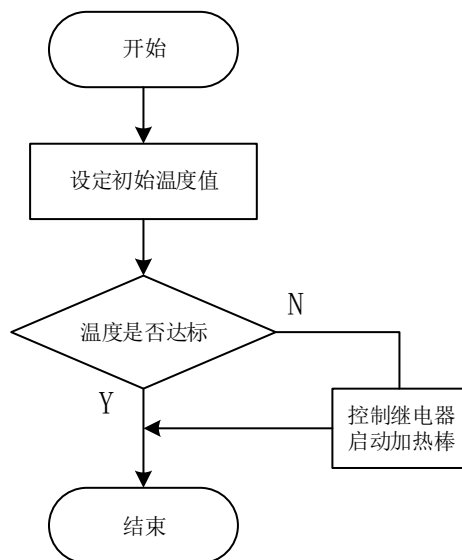


图 5.3 升温模块设计流程图

首先通过按钮设置初始温度值，当温度达标时，加热棒停止工作。当温度不达标时，单片机控制继电器启动加热棒，使鱼缸温度直到达到标准值，再使加热棒停止工作。

### 5.4 增氧模块程序设计

根据用户设置的增氧时间，控制智能鱼缸增氧。增氧模块程序设计流程图如图 5.4 所示。

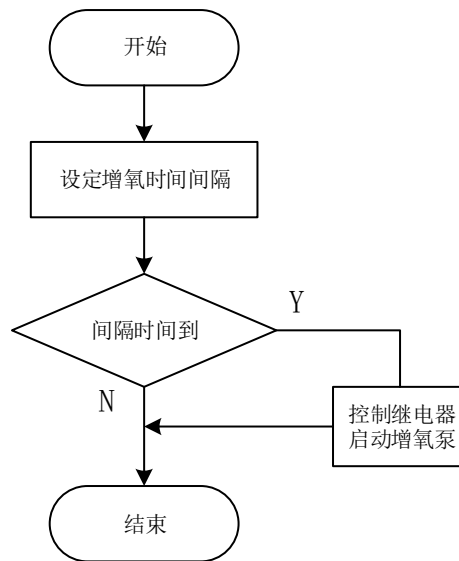


图 5.4 增氧模块程序设计流程图

首先用户通过按键设置增氧时间间隔，当时间到达设定时间时，单片机控制继电器启动增氧泵，对鱼缸进行增氧。当时间没有到达设定时间时，增氧泵不工作。

## 5.5 软件设计环境

Keil C51 是美国 Keil Software 公司出品的 51 系列兼容单片机 C 语言软件开发系统，与汇编相比，C 语言在功能上、结构性、可读性、可维护性上有明显的优势，因而易学易用。Keil 提供了包括 C 编译器、宏汇编、链接器、库管理和一个功能强大的仿真调试器等在内的完整开发方案，通过一个集成开发环境（ $\mu$ Vision）将这些部分组合在一起。运行 Keil 软件需要 WIN98、NT、WIN2000、WINXP 等操作系统。Keil 公司是一家业界领先的微控制器（MCU）软件开发工具的独立供应商。Keil 公司由两家私人公司联合运营，分别是德国慕尼黑的 Keil Elektronik GmbH 和美国德克萨斯的 Keil Software Inc。Keil 公司制造和销售种类广泛的开发工具，包括 ANSI C 编译器、宏汇编程序、调试器、连接器、库管理器、固件和实时操作系统核心（real-time kernel）。有超过 10 万名微控制器开发人员在使用这种得到业界认可的解决方案。其 Keil C51 编译器自 1988 年引入市场以来成为事实上的行业标准，并支持超过 500 种 8051 变种。

软件开发界面如图 5.5 所示。

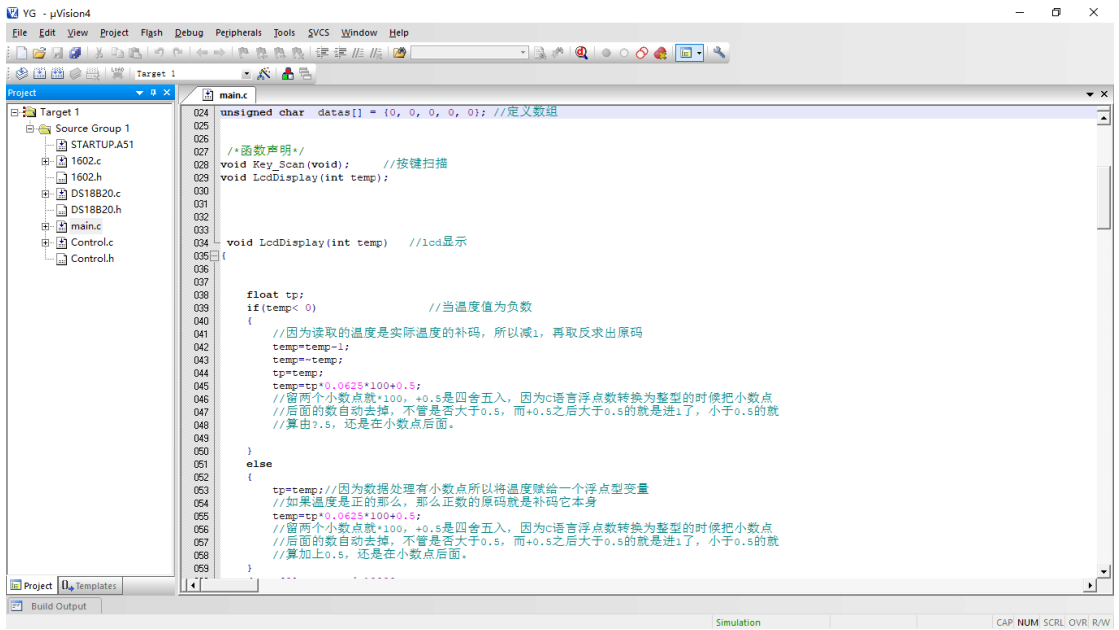


图 5.5 软件开发界面

智能鱼缸的设计所使用的处理器 STC89C52 单片机的编译环境就是 Keil，其满足了各个功能模块的编译过程，达到了各模块的软件功能，完成了预期指标。

5.6 软件设计总结

智能鱼缸系统软件代码使用了 C 语言，在 Keil 编译环境下完成 STC89C52 与各功能模块的调试。依据各模块工作流程完成各模块程序设计，并且通过各个串口连接各个模块，完成了整体系统搭建。在调试的过程中，通过优化程序代码和系统整体结构，使得系统对资源的利用率以及对任务的响应速度得到了极大的提升。

## 第六章 总结

从毕业设计的设计及功能实现，以及论文的撰写都是在指导老师马翔老师的悉心指导下完成的。最初对毕业设计所要完成的各个功能和需要搭建的各个模块一无所知，然后慢慢开始查找资料，一点一点的进行积累，遇到问题虚心请教老师、同学，在积累中蜕变在问题中成长，最后成功完善了设计。

刚开始做毕业设计的时候对于智能鱼缸的整体设计不太明确，没有接触过相关的设计知识，一筹莫展不知如何开展工作。后来经过马老师和几个小伙伴的指导以及查阅资料，决定了采用模块化程序设计。分模块进行程序设计使得系统的各个功能与模块更加清晰明了，在确定了系统最终需要实现的功能之后，选用了 STC89C52 作为主控核心，搭配各相关的精密传感器，确保了准确性与可靠性。在初步指定方案后，开始进行实物搭建，硬件的设计在满足电路合理性与实物可操作的前提下进行设计的。在进行换水的模块的调试过程中遇到了很多问题，从无法进出水到后来的只能出水不能进水，通过从头开始查询之后，最终发现是对电磁阀的理解不够并且没有详细的了解到水泵的功能，最后对水泵的各个功能以及用法进行了研究才解决了这个关卡。

毕设的过程虽然比较枯燥，但最终看到自己的作品呈现在眼前的时候还是比较欣慰的。目前智能鱼缸管家可实现以下功能：

- a. 可实时监测水生动植物的生长环境因素（鱼缸中水的温度）；
- b. 根据收集到的水的温度 and 用户自行设置的温度，自动为鱼缸进行升温；
- c. 可通过用户设置循环时间，控制继电器自动为鱼缸增氧；
- d. 可通过用户的自定义操作对鱼缸进行换水。

最初在设计进出水的模块，因为没有搞清楚电磁阀的使用方法，一直没有办法控制鱼缸的进出水。在多次实验无果的情况下，我去找老师寻求帮助，知道了电磁阀的使用方法，但是最后为了试验的简便性，将电磁阀换成了水泵，解决了进出水的问题。

由于本次设计的电压较高，大部分器件的电压都是 220V，所以最初没有封胶时的整个实验板还是处于一个较危险的状态，心里也难免有些害怕。虽然老师多次强调安全问题，但是还是烧了一次电路板，索性没有受伤。

本次毕业设计中也存在了一定的问题，就是鱼缸我只考虑了升温的问题但是没有考虑降温的问题。因为题目要求里只写了鱼缸的升温，当老师问到我有没有想到夏天气温过高时，鱼缸内温度也非常高的情况下，怎么降低鱼缸内的温度呢。当时我真是一头雾水，我甚至有想过难道我还要给鱼缸内装一个小空调的想法。最后经过长时间的思考无果后与老师沟通，最后得出了想要降温，可以加入自来水进行降温。这个结果真的出乎我的意料又十分的简便。我觉得这个鱼缸非常需要改进的地方就是当温度过高时对鱼缸的降温问题。虽然这个问题目前只是提出

了假设还没有得到解决,但是我觉得通过这次实验让我明白了当存在问题时应该积极思考,积极的解决而不是一味的逃避。

就目前来说,智能鱼缸还处于起步阶段,作品的性能、功能和应用模式都处于探索阶段。所以其中会有许多不足之处,后期会进一步增加各种功能,我将不断探索。

## 致 谢

毕业设计的完成离不开马老师的指导。在马老师的悉心指导下，毕业设计从开题到最后的论文提交，老师耐心地讲解，一遍一遍对论文进行修改，老师的精益求精的科研精神，和对学生认真负责的态度打动了我，也让我有了更充足的动力去投入到毕业设计中。感谢马老师的指导，以及实验室提供给我的良好平台。

最后我由衷的感谢我的学校——西安邮电大学，在这个美丽的校园里，让我度过了难忘的大学时光。不仅教会我科学知识，更让我感受到了温暖。感谢学校的平台，感谢实验室的资源，让我收获了科学知识同时，也让我快乐成长。在校期间结识到了良师益友，这将是人生最宝贵的财富。我将倍加珍惜。

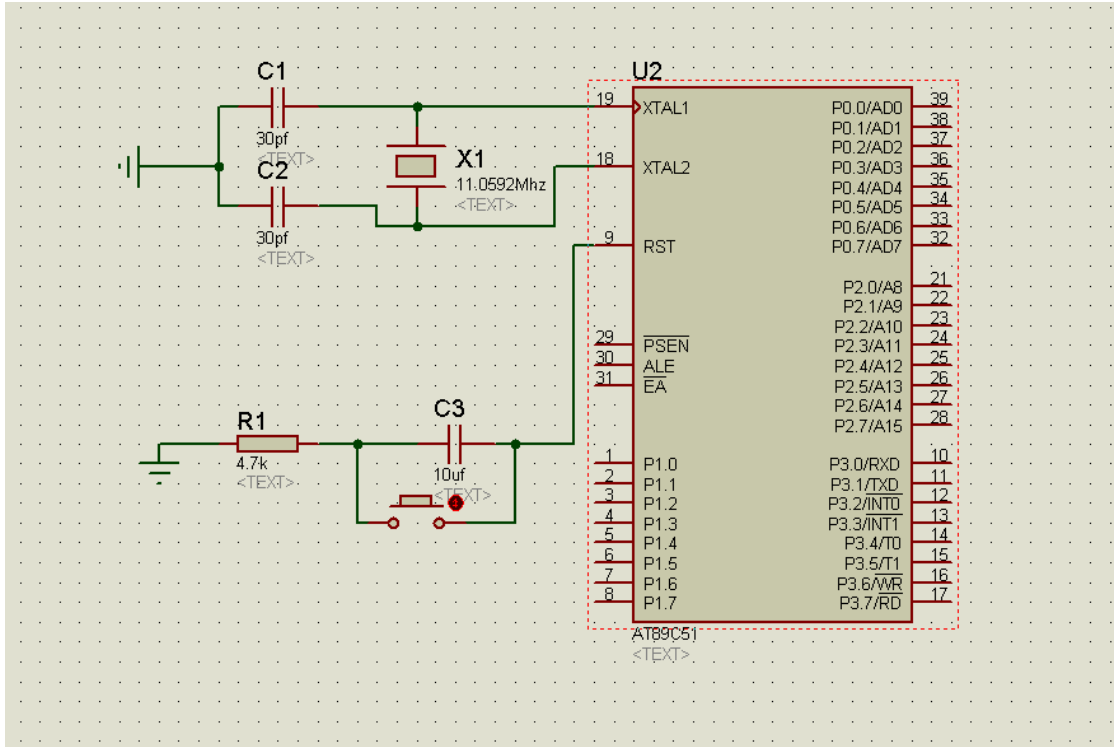
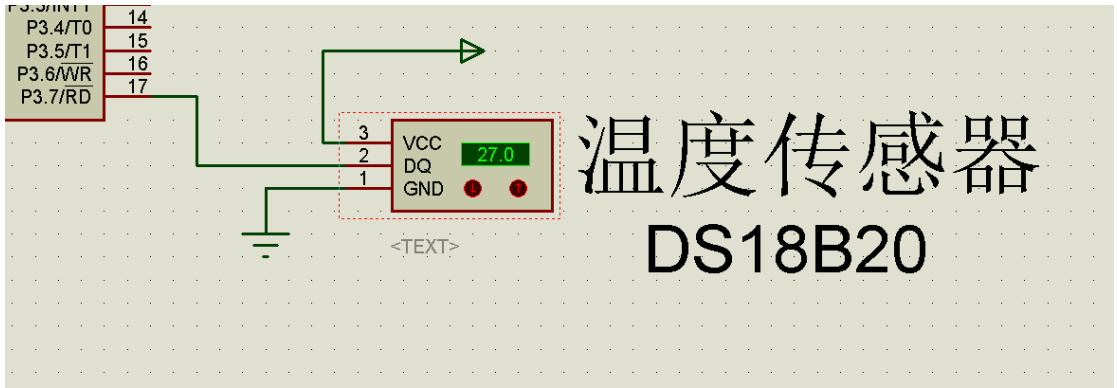
## 参考文献

- [1] 林玉明. 智能单相电机保护器的设计[J]. 光电技术应用. 2011(02):84-86.
- [2] 宋英. 通信智能仪表训练设计简述[J]. 中小企业管理与科技(上旬刊). 2011(07):286-287.
- [3] 尹诚,侯延东,白玉民. 车用柴油机可变喷嘴涡轮增压器电控系统研究[J]. 汽车技术. 2011(09):32-38.
- [4] 王保民,李明,乔宪生,赵改荣,俞宏,崔学晨,王贺. 电脉冲植物愈伤组织激活仪的研制及其应用[J]. 果树学报. 2011(06):175-179.
- [5] 伍儒彬,许萍萍,蔡琼. STC 单片机的液晶显示方案设计与实现[J]. 武汉工程大学学报. 2011(11):87-90.
- [6] 付欣玮,李俊逸,尹程斌,左明,陈弟虎. 基于单片机的 GPS 智能导游机系统设计与实现[J]. 微计算机信息. 2012(02):65-67.
- [7] 张扬. 帆板控制系统[J]. 四川工程职业技术学院学报. 2012(01):47-51+72.
- [8] 罗明. 交流电动机电容器耐久性试验系统的设计[J]. 硅谷. 2012(07):92-93.
- [9] 张兵,马丁. 基于 PCI 的多通道程控电压放大器设计[J]. 电子测量技术. 2012(04):25-29.
- [10] 侯江涛. 基于 STC 单片机的铁路信号半自动闭塞光缆传输器[J]. 电子设计工程. 2012(10):32-33+36.
- [11] 张扬. 帆板控制系统[J]. 电脑开发与应用. 2012(05):46-49+55.
- [12] 周官喜,王德兴. 基于 WT588D 语音芯片的红外防盗报警系统[J]. 电脑知识与技术. 2012(22):97-99.
- [13] 李文涛,杨小新. 基于 STC89C52 的智能温度变送器的设计[J]. 仪表技术与传感器. 2012(11):73-76.
- [14] 周贵舟. Protues 和 Keil 在单片机教学中的仿真应用[J]. 才智. 2013(05):120.

- [15] 常芳,朱瑞祥,吴艳艳,郭俊杰. U 盘模块在土壤温湿度数据采集中的应用[J]. 农机化研究. 2013(04):193-196.
- [16] 吴晓喜,周培慧. 地铁车辆能耗监测系统的设计[J]. 电力机车与城轨车辆. 2014(01):44-46.
- [17] 姚存治. 基于 Proteus 和 Keil 的单片机课程教学模式改革[J]. 郑州铁路职业技术学院学报. 2014(01):85-87+90.
- [18] 魏雅,杜云. 单片机应用系统抗干扰的设计[J]. 电子设计工程. 2014(08):77-79.
- [19] 白国政. 基于 STC89C52 便携式甲醛测试仪的研制[J]. 信息技术. 2014(06):54-57+62.







## 附录 2：实物图

## 附录 3：部分程序代码

Main.txt

```
#include "1602.h"
#include "DS18B20.h"
#include "Control.h"
#include "Base.h"

int tem;
int com_tem = 0;
int set_tem = 25;
int set_cout = 100;
int K3_count = 0;

uchar ge;
uchar shi;
uchar Space;

uchar Int_Flag = 0;

uchar Inc_Flag = 0;    //增加
uchar Red_Flag = 0;    //减少
uchar Sel_Flag、g = 0;    //选择
uchar Exit_Flag = 0;    //推出

unsigned char datas[] = {0, 0, 0, 0, 0}; //定义数组

/*函数声明*/
void Key_Scan(void); //按键扫描
void LcdDisplay(int temp);
```

control.h.txt

```
#ifndef __CONTROL_H
#define __CONTROL_H

#include "Base.h"

/*定义继电器与按键控制引脚*/
sbit K2=P1^0;
sbit K3=P1^1;
sbit K4=P1^2;
sbit Inc_Key = P1^5;
sbit Red_Key = P1^4;
sbit Exit_Key = P1^6;
sbit Sel_Key = P1^7;

#define K2_On()    K2 = 0;
#define K2_Off()   K2 = 1;
#define K3_On()    K2 = 0;
#define K3_Off()   K2 = 1;
#define K4_On()    K2 = 0;
#define K4_Off()   K2 = 1;

void Control_Init(void);    //继电器控制初始化
void Timer0Init(void);     //定时器初始化

void delay_ms(uint xms);

#endif
```

Control.c.txt

```
#include "Base.h"
#include "Control.h"

void delay_ms(uint xms)
{
    uint i, j;
    for(i = xms; i>0;i--)
        for(j=110;j>0;j--);
}

void Control_Init(void)
{
    K2_Off();
    K3_Off();
    K4_Off();
}

void Timer0Init(void)          //50 毫秒@12.000MHz
{
    TMOD=0x01;
    TH0=(65536-45872)/256;
    TL0=(65536-45872)%256;
    ET0=1;
    TR0=1;
}
```

1602.c.text

/\*1602 屏幕\*/

#include "1602.h"

/\*\*\*\*\*\*

\*\*\*\*\*

\* 函 数 名 : Lcd1602\_Delay1ms

\* 函数功能 : 延时函数, 延时 1ms

\* 输 入 : c

\* 输 出 : 无

\* 说 名 : 该函数是在 12MHZ 晶振下, 12 分频单片机的延时。

\*\*\*\*\*

\*\*\*\*\*/

void Lcd1602\_Delay1ms(uint c) //误差 0us

{

uchar a, b;

for (; c>0; c--)

{

for (b=199; b>0; b--)

{

for (a=1; a>0; a--);

}

}

}

/\*\*\*\*\*\*

\*\*\*\*\*

\* 函 数 名 : LcdWriteCom

\* 函数功能 : 向 LCD 写入一个字节的命令

\* 输 入 : com

\* 输 出 : 无

```

*****
*****/
#ifndef    LCD1602_4PINS //当没有定义这个 LCD1602_4PINS 时
void LcdWriteCom(uchar com) //写入命令
{
    LCD1602_E = 0;    //使能
    LCD1602_RS = 0;    //选择发送命令
    LCD1602_RW = 0;    //选择写入

    LCD1602_DATAPINS = com;    //放入命令
    Lcd1602_Delay1ms(1);    //等待数据稳定

    LCD1602_E = 1;    //写入时序
    Lcd1602_Delay1ms(5);    //保持时间
    LCD1602_E = 0;
}
#else
void LcdWriteCom(uchar com) //写入命令
{
    LCD1602_E = 0;    //使能清零
    LCD1602_RS = 0;    //选择写入命令
    LCD1602_RW = 0;    //选择写入

    LCD1602_DATAPINS = com; //由于 4 位的接线是接到 P0 口的高四位，所以
    传送高四位不用改
    Lcd1602_Delay1ms(1);

    LCD1602_E = 1;    //写入时序
    Lcd1602_Delay1ms(5);
    LCD1602_E = 0;

    // Lcd1602_Delay1ms(1);
    LCD1602_DATAPINS = com << 4; //发送低四位
    Lcd1602_Delay1ms(1);

```



```

    LCD1602_E = 1;    //写入时序
    Lcd1602_Delay1ms(5);
    LCD1602_E = 0;
}
#endif

/*****
*****
* 函 数 名      : LcdWriteData
* 函数功能      : 向 LCD 写入一个字节的数据
* 输    入      : dat
* 输    出      : 无
*****
*****/
#ifndef LCD1602_4PINS
void LcdWriteData(uchar dat)          //写入数据
{
    LCD1602_E = 0;    //使能清零
    LCD1602_RS = 1;   //选择输入数据
    LCD1602_RW = 0;   //选择写入

    LCD1602_DATAPINS = dat; //写入数据
    Lcd1602_Delay1ms(1);

    LCD1602_E = 1;    //写入时序
    Lcd1602_Delay1ms(5); //保持时间
    LCD1602_E = 0;
}
#else
void LcdWriteData(uchar dat)          //写入数据
{
    LCD1602_E = 0;    //使能清零
    LCD1602_RS = 1;   //选择写入数据
    LCD1602_RW = 0;   //选择写入

    LCD1602_DATAPINS = dat; //由于 4 位的接线是接到 P0 口的高四位，所以

```

传送高四位不用改

```

    Lcd1602_Delay1ms(1);

    LCD1602_E = 1;      //写入时序
    Lcd1602_Delay1ms(5);
    LCD1602_E = 0;

    LCD1602_DATAPINS = dat << 4; //写入低四位
    Lcd1602_Delay1ms(1);

    LCD1602_E = 1;      //写入时序
    Lcd1602_Delay1ms(5);
    LCD1602_E = 0;
}
#endif

/*****
*****
* 函 数 名      : LcdInit()
* 函数功能      : 初始化 LCD 屏
* 输    入      : 无
* 输    出      : 无
*****
*****/

#ifndef LCD1602_4PINS
void LcdInit()                //LCD 初始化子程序
{
    LcdWriteCom(0x38); //开显示
    LcdWriteCom(0x0c); //开显示不显示光标
    LcdWriteCom(0x06); //写一个指针加 1
    LcdWriteCom(0x01); //清屏
    LcdWriteCom(0x80); //设置数据指针起点
}
#else
void LcdInit()                //LCD 初始化子程序
{

```

```

LcdWriteCom(0x32);    //将 8 位总线转为 4 位总线
LcdWriteCom(0x28);    //在四位线下的初始化
LcdWriteCom(0x0c);    //开显示不显示光标
LcdWriteCom(0x06);    //写一个指针加 1
LcdWriteCom(0x01);    //清屏
LcdWriteCom(0x80);    //设置数据指针起点

LcdWriteCom(0x80);    //写地址 80 表示初始地址
LcdWriteData('T');    //百位

LcdWriteCom(0x81);    //写地址 80 表示初始地址
LcdWriteData('e');    //百位

LcdWriteCom(0x82);    //写地址 80 表示初始地址
LcdWriteData('m');    //百位

LcdWriteCom(0x83);    //写地址 80 表示初始地址
LcdWriteData(':');    //百位

LcdWriteCom(0x8B);    //写地址 80 表示初始地址
LcdWriteData('(');    //十位

LcdWriteCom(0x8E);    //写地址 80 表示初始地址
LcdWriteData(')');    //十位

LcdWriteCom(0xC0);    //写地址 80 表示初始地址
LcdWriteData('I');    //百位
LcdWriteCom(0xC1);    //写地址 80 表示初始地址
LcdWriteData('n');    //百位
LcdWriteCom(0xC2);    //写地址 80 表示初始地址
LcdWriteData('t');    //百位
LcdWriteCom(0xC3);    //写地址 80 表示初始地址
LcdWriteData('e');    //百位

```

```
LcdWriteCom(0xC4);          //写地址 80 表示初始地址
LcdWriteData(':'); //百位
}
#endif
```

DS18B20.c.txt

/\*温度传感器\*/

#include "DS18B20.h"

/\*\*\*\*\*\*

\*\*\*\*\*

\* 函数名 : Delaylms

\* 函数功能 : 延时函数

\* 输入 : 无

\* 输出 : 无

\*\*\*\*\*

\*\*\*\*\*/

void Delaylms(uint y)

{

uint x;

for( ; y>0; y--)

{

for(x=110; x>0; x--);

}

}

/\*\*\*\*\*\*

\*\*\*\*\*

\* 函数名 : Ds18b20Init

\* 函数功能 : 初始化

\* 输入 : 无

\* 输出 : 初始化成功返回 1, 失败返回 0

\*\*\*\*\*

\*\*\*\*\*/

uchar Ds18b20Init()

{

uchar i;

DSPORT = 0; //将总线拉低 480us~960us

i = 70;

```

while(i--);          //延时 642us
DS18B20 = 1;         //然后拉高总线，如果 DS18B20 做出反应会将在
15us~60us 后总线拉低
i = 0;
while(DS18B20) //等待 DS18B20 拉低总线
{
    i++;
    if(i>5)//等待>5MS
    {
        return 0;//初始化失败
    }
    Delay1ms(1);
}
return 1;//初始化成功
}

/*****
*****
* 函 数 名      : Ds18b20WriteByte
* 函数功能      : 向 18B20 写入一个字节
* 输    入      : com
* 输    出      : 无
*****
*****/

void Ds18b20WriteByte(uchar dat)
{
    uint i, j;

    for(j=0; j<8; j++)
    {
        DS18B20 = 0;          //每写入一位数据之前先把总线拉低 1us
        i++;
        DS18B20 = dat & 0x01; //然后写入一个数据，从最低位开始
        i=6;
    }
}

```

```

        while(i--); //延时 68us, 持续时间最少 60us
        DSPORT = 1;    //然后释放总线, 至少 1us 给总线恢复时间才能接着写
        入第二个数值
        dat >>= 1;
    }
}
/*****
*****
* 函 数 名      : Ds18b20ReadByte
* 函数功能      : 读取一个字节
* 输    入      : com
* 输    出      : 无
*****
*****/

uchar Ds18b20ReadByte()
{
    uchar byte, bi;
    uint i, j;
    for(j=8; j>0; j--)
    {
        DSPORT = 0; //先将总线拉低 1us
        i++;
        DSPORT = 1; //然后释放总线
        i++;
        i++; //延时 6us 等待数据稳定
        bi = DSPORT; //读取数据, 从最低位开始读取
        /*将 byte 左移一位, 然后与上右移 7 位后的 bi, 注意移动之后移掉那
        位补 0。*/
        byte = (byte >> 1) | (bi << 7);
        i = 4; //读取完之后等待 48us 再接着读取下一个数
        while(i--);
    }
    return byte;
}

```

```
/*
*****
* 函 数 名      : Ds18b20ChangTemp
* 函数功能      : 让 18b20 开始转换温度
* 输    入      : com
* 输    出      : 无
*****
*****/
```

```
void Ds18b20ChangTemp()
{
    Ds18b20Init();
    Delay1ms(1);
    Ds18b20WriteByte(0xcc);    //跳过 ROM 操作命令
    Ds18b20WriteByte(0x44);    //温度转换命令
    Delay1ms(100);    //等待转换成功，而如果你是一直刷着的话，就不用这个延时了
}
```

```
/*
*****
* 函 数 名      : Ds18b20ReadTempCom
* 函数功能      : 发送读取温度命令
* 输    入      : com
* 输    出      : 无
*****
*****/
```

```
void Ds18b20ReadTempCom()
{
    Ds18b20Init();
    Delay1ms(1);
    Ds18b20WriteByte(0xcc);    //跳过 ROM 操作命令
    Ds18b20WriteByte(0xbe);    //发送读取温度命令
```



```

}
/*****
*****
* 函数名      : Ds18b20ReadTemp
* 函数功能    : 读取温度
* 输入      : com
* 输出      : 无
*****
*****/

int Ds18b20ReadTemp()
{
    int temp = 0;
    uchar tmh, tml;
    Ds18b20ChangTemp();           //先写入转换命令
    Ds18b20ReadTempCom();         //然后等待转换完后发送读取温度命令
    tml = Ds18b20ReadByte();      //读取温度值共 16 位，先读低字节
    tmh = Ds18b20ReadByte();      //再读高字节
    temp = tmh;
    temp <<= 8;
    temp |= tml;
    return temp;
}

```

DS18B20.h.txt

```
#ifndef __DS18B20_H
#define __DS18B20_H
```

```
#include "Base.h"
```

```
sbit DSPORT=P3^7;
```

```
void Delay1ms(uint y);
```

```
void DS18B20Init();          //初始化
```

```
void Ds18b20WriteByte(uchar com); // 写入一个字节
```

```
uchar Ds18b20ReadByte();      //读取一个字节
```

```
void Ds18b20ChangTemp();      //开始转换温度
```

```
void Ds18b20ReadTempCom();    //发送读取温度命令
```

```
int Ds18b20ReadTemp();        //读取温度
```

```
#endif
```