**《电子系统设计实践》**

**设计报告**

|  |  |
| --- | --- |
| 班 级： |  |
| 姓 名： |  |
| 学 号： |  |

**2023~2024学年第二学期**

**2023-2024学年第二学期《电子系统设计实践》任务书**

学院：信息科学与工程学院 专业： 信息工程

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指导教师 | | 李钰、木昌洪 | 起止时间 | 2024.6.27-2024.7.6 |
| 课题名称 | | 温度控制系统设计实践 | | |
| 内容及任务 | 《电子系统设计实践》是信息工程专业的必修实践课，是教学计划中理论联系实际 的一个重要环节。学生在此门课程实践环节中，完成一整套电子综合设计实践平台的硬 件原理分析、硬件焊接与调试、以及软件编程的工作。学生在微处理器平台上编程实现 课题的主要性能要求，软件的具体细节功能由学生依据平台提供的硬件资源自主拓展完 成。本环节能锻炼学生硬件分析能力和微处理器平台上的软件编程能力，有助于培养学 生对系统整体概念的认识，激发学生对电子信息专业方向的学习兴趣，培养学生的创新 意识。 | | | |
| 拟达到的要求或技术指标 | 课程设计一人一组，自行完成硬件原理分析、硬件焊接以及软件编程的工作。课程 设计的总体要求如下。  (1) 学习要求：学习态度端正，勤学好问，自主研发，不抄袭他人研究成果。  (2) 硬件要求：学习和分析电子综合设计实践平台的硬件原理图，理解掌握实践平台各功能模 块的工作原理。 焊接电子综合设计实践平台一套。要求焊接质量良好，无虚焊点及漏焊点；要求器件摆放整齐、器件无错焊、漏焊现象。  (3) 软件要求：独立设计实践平台软件，完成键盘显示、温度采集、电机控制以及数据存储等基本功能。在实现平台基本功能的基础上，设计具备良好人机交互能力的软件菜单， 完成工艺参数的设定与存储，并根据实测的温度对电机进行实时调速。  (4) 报告要求：设计报告能正确叙述实践平台各部分的硬件工作原理；完整叙述软件主程序流程及各个子功能模块的设计思路与流程， 回答关于程序设计方面的问题。  (5) 答辩要求：现场演示和解说平台能实现的各种功能，正确阐述其实现方法，并回答关于硬件工作原理与软件程序设计方法方面的问题。 | | | |

**《电子系统设计实践》设计报告评分表**



|  |  |  |
| --- | --- | --- |
| **评分项目** | **评分标准** | **学生得分** |
| **设计方案（40分）**  对设计需求进行分析；详细描述设计方案，相关电路的设计原理；以流程图形式描述软件设计过程 | A：优（36~40分）  B：良（32~35分）  C：中（24~31分）  D：差（< 24分） |  |
| **设计效果（40分）**  以图表形式详细描述设计效果，并对设计中出现的一些现象进行分析，解释背后的原因 | A：优（36~40分）  B：良（32~35分）  C：中（24~31分）  D：差（< 24分） |  |
| **成本效益分析**  **及项目管理（20分）**  对设计对象的成本效益进行分析并对设计过程的时间管理情况进行总结 | A：优（18~20分）  B：良（16~17分）  C：中（12~15分）  D：差（<12分） |  |
| **总分** |  | |
| **教师签名** |  | |

**温度控制系统设计实践**

张三（作者姓名，宋体，10.5号，居中）

（学号：12345678 专业：信息工程）（字体：宋体，10.5号，居中）

**摘 要：**此次实践目的在于培养学生对系统整体概念的认识，激发学生对电子信息专业方向的学习兴趣，培养学生的创新意识。在此门课程实践环节中，需要完成一整套电子综合设计实践平台的硬件原理分析、硬件焊接与调试、以及软件编程的工作。在微处理器平台上，使用 C 语言编程实现课题的主要性能要求，软件的具体细节功能包括温度实时采集、工艺参数设定与存储、电机运行与调速等。实践结果为成功完成任务，并通过实际运行测试，验证了系统设计的可行性和有效性。尽管在实践过程中遇到了一些技术难题，但总体上成功完成了设计任务。



**关键词：**单片机 软硬件结合 温度检测 电机控制

**0 引言**

本实践涉及对STC89C52单片机平台的硬件原理学习和应用，以及KeilC51软件的使用。学生需要掌握包括键盘显示、温度采集、电机控制和数据存储等模块的编程和实现。课程设计覆盖了从硬件焊接质量到软件功能实现的全方位技能，包括良好的人机交互界面设计，工艺参数的设定与存储，以及基于实测温度的电机实时调速功能。课程的范围广泛，不仅包括基础的电子元件和模块的了解，还扩展到数据采集、处理和电机控制的实际应用。背景方面，随着中国式现代化的推进，电子系统设计在多个领域中扮演着越来越重要的角色，强调了该课程与国家发展需求的紧密结合。此实践锻炼学生的硬件分析能力和微处理器平台上的软件编程能力，有助于培养学生对系统整体概念的认识，激发学生对电子信息专业方向的学习兴趣，培养学生的创新意识。

1. **温度控制系统的总体结构**
   1. **温度控制系统的模块**

温度控制系统的功能模块有：

1. **核心处理器**：STC89C52单片机作为系统的核心处理单元，负责协调和控制整个系统的工作。
2. **温度采集模块**：使用DS18B20数字温度传感器进行环境温度的实时采集，并通过单片机进行数据处理。
3. **显示与键盘模块**：HD7279A芯片用于驱动LED数码显示器和键盘矩阵，实现用户交互，显示实时温度和系统状态。
4. **电机控制模块**：通过单片机的P1口输出PWM波形，控制电机的转速，进而调节系统的热交换量，实现温度控制。
5. **数据存储模块**：24C16串行E2PROM芯片用于存储工艺参数和温度设定值，保证系统在掉电后数据不丢失。
6. **串行通信模块**：MAX232芯片实现与上位机PC的数据通信，便于程序下载和数据交换。
7. **电源模块**：采用LM2575-5V芯片提供稳定的电源供应。
8. **人机交互界面**：通过按键和显示模块，用户可以设置温度参数，启动或停止温度控制过程，并查看实时数据。
9. **控制算法**：系统内部实现的控制算法，如PID控制，根据实时采集的温度与设定的目标温度之间的偏差，动态调整电机的运行状态。

**1.2 模块间的结合方式**

各模块间的结合是通过核心处理器STC89C52单片机来实现的。具体地说，核心处理器（STC89C52）作为系统控制中心，协调其他模块的工作。它通过程序控制，实现对温度传感器的读取、数据处理、PWM波形的生成、显示和键盘输入响应等功能。HD7279A键盘显示模块与STC89C52的I/O端口相连，接收来自单片机的控制信号，显示菜单信息、实时温度和状态信息，同时将用户的按键操作转换为信号反馈给单片机。DS18B20温度采集模块通过单线接口与STC89C52的I/O端口相连，实现温度的实时采集。采集到的温度数据通过数据线DQ传输给单片机进行处理。电机控制模块由STC89C52的P1口引脚输出PWM信号，控制电机的转速，实现对温度的调节。PWM信号的占空比由单片机根据温度控制算法计算得出。24C16数据存储模块通过I2C总线与STC89C52的I/O端口相连，实现对工艺参数的设定、保存和读取。单片机通过I2C通信协议与存储芯片进行数据交换。MAX232串行通信模块与STC89C52的串行通信接口TXD和RXD相连，实现与上位机PC的数据通信，用于程序下载和数据交换。具体框图如下：

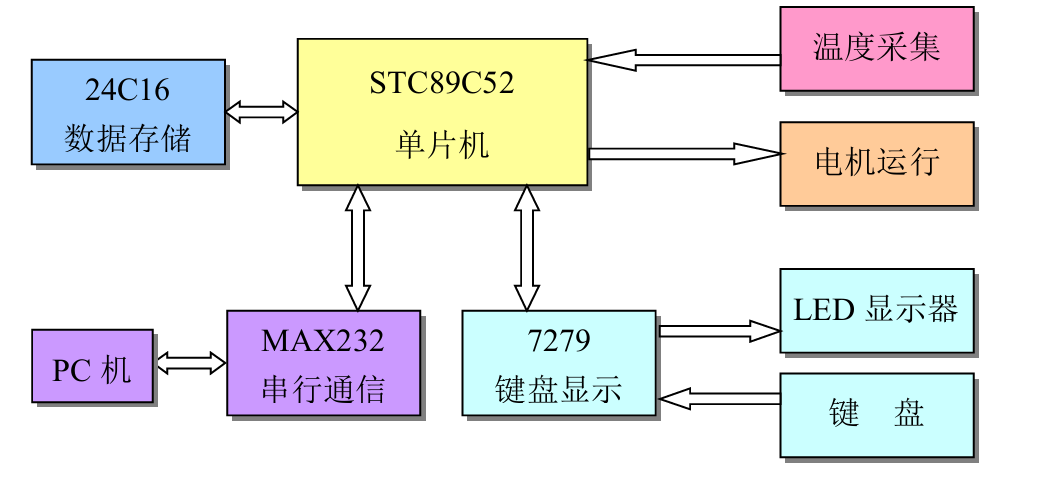


图 1硬件总体框图

**1.3 程序逻辑**

软件设计的结构为项目主菜单、温度检测、电机测试、电机调速、工艺参数设置。通过用户按键，在主菜单选择要执行的模块，分别进入对应子菜单，并对子菜单项目进行处理。

图示

描述已自动生成

图 2软件主菜单界面示例图

**2 各模块的硬件工作原理**

**2.1 STC89C52核心处理器**

实践平台选用STC89C52 芯片作为核心处理器，该单片机在电子行业有着广泛的应用。STC89C52是一个低电压、高性能的 8位单片机，片内含8K字节可反复擦写的Flash只读程序存储器和256 字节的随机存取数据存储器（RAM），器件兼容MCS-51指令系统，内置通用8位中央处理器和Flash存储单元。它还具有丰富的外设接口，包括 4 个 16 位定时器/计数器、2 个串行口、1 个 SPI 接口、1 个 I2C 接口等多种通讯接口。此外，STC89C52 还支持多种时钟源，可以实现高精度计时。本次实验选用11.0592MHz晶振，采用按键复位电路。核心处理器电路图如图所示。

图示, 示意图

描述已自动生成

图 3核心处理器部分硬件原理图

**2.2 HD7279A键盘显示模块**

HD7279A是一片具有串行接口的，可同时驱动8位共阴极数码管的智能显示驱动芯片。该芯片可连接多达64个键的键盘矩阵，并含有去抖动电路。HD7279A芯片内部有译码器，可以直接接受l6进制码，并且具有2种译码方式和多种控制指令，如：消隐、闪烁、左移、右移、段寻址等。可以广泛应用在仪器仪表，工业控制，条形显示器，控制面板等领域。 实践平台通过HD7279A芯片外扩8位LED数码显示器及4个按键。HD7279A的片选线CS、时钟线CLK、数据线DATA及键盘线KEY可通过短路跳子直接与STC89C52的P1.4、P1.5、P1.7及P1.6引脚连接。HD7279A键盘显示硬件电路图如图所示。

图示, 示意图

描述已自动生成

图 4 HD7279A键盘显示模块硬件原理图

**2.3 数据采集模块**

实践平台选用DS18B20作为温度传感器，测量环境温度值。该传感器选用外部电源供电方式。外部电源供电方式是DS18B20 最佳的工作方式，工作稳定可靠，抗干扰能力强，而且电路也比较简单。在外接电源方式下，可以充分发挥DS18B20 宽电源电压范围的优点，即使电源电压VCC 降到3V时，依然能够保证温度量精度。DS18B20的输入输出引脚DQ可通过短路跳子直接与STC89C52的P1.3引脚连接。数据采集电路图如图所示。

图片包含 游戏机, 钟表

描述已自动生成

图 5 数据采集模块硬件原理图

**2.4 电机控制模块**

实践平台选用直流小风机作为电机控制对象，由单片机P1口引脚连接简单的三极管驱动电路驱动电机运转。电机调速由程序控制P1口引脚产生不同占空比的PWM波实现，电机控制端可通过短路跳子直接与STC89C52的P1.2 引脚连接。电机控制电路图如图所示.

图示, 示意图

描述已自动生成

图 6 电机控制电路图

**2.5 数据存储模块**

实践平台选用24C16芯片实现数据存储功能，芯片数据线SDA和时钟线SCL可通过短路跳子直接与STC89C52的P1.0和P1.1引脚连接。24C16是一个16K 位串行E2PROM，其内部有一个8 字节页写缓冲器，器件通过I2C总线接口进行操作，有一个专门的写保护功能。即使在工作过程中24C16出现损坏，用户设置的存储信息也不会丢失，从而不会造成重要功能丢失以及工作的错误。数据存储电路图如图所示。

图示, 示意图

描述已自动生成

图 7 数据存储电路图

**2.6 串行通信模块**

实践平台选用MAX232接收/发送器作为串行通信接口芯片。串行通信模块主要负责程序下载和与上位机PC机的数据通信功能。MAX232 的接收信号输出引脚R1OUT连接至STC89C52的RXD引脚，发送信号输入引脚T1IN连接至STC89C52的TXD引脚，发送信号输出引脚T1OUT和接收信号输入引脚R1IN连接至标准9芯串口接插件。串行通信电路图如图所示。

图示, 示意图

描述已自动生成

图 8 串行通信电路图

**2.7 电源模块**

电源模块采用LM2575-5V芯片。LM2575 系列开关稳压集成电路是美国国家半导体公司生产的1A集成稳压电路，它内部集成了一个固定的振荡器，只须极少外围器件便可构成一种高效的稳压电路，可大大减小散热片的体积，而在大多数情况下不需散热片；内部有完善的保护电路，包括电流限制及热关断电路等，是传统三端式稳压集成电路的理想替代产品。 实际的电压在9~12V之间。电源电路图如图所示。

图示, 示意图

描述已自动生成

图 9 电源电路图

**3 温度控制系统的软件设计**

**3.1 总体设计**

程序总体设计如下图所示，主要分为四个子项目：温度检测、电机测试、电机调速、参数设置。每个项目拥有不同的深度，并调用系统内有限个模块与用户进行交互。

图示

描述已自动生成

图 10程序结构图

最初的设计是经典的 layer 层 + menu 层的菜单。这样的代价是：1 耦合度非常高，写起来很痛苦 2 程序不断跑主循环，所有函数都必须非阻塞。然后我写了一半感觉不太对，把代码抹掉重新设计菜单逻辑。在跟舍友讨论后，我和他分别提出了一个设计：

1. 我的设计：所有函数都是阻塞的，全部塞 `while(1)` 和按键判断。

2. 舍友的设计：提前规划好一颗状态树和所有节点，按键只负责在状态树上移动，逻辑根据当前的节点实现。

最终我还是决定先使用我的设计：程序总体使用分阶段循环进行设计，每一个功能划分一个函数，函数内都拥有独立的循环，显示，按键处理功能，即所有函数都是阻塞的。这样做的好处有：

* + - 1. 功能解耦，各功能函数相对独立，逻辑清晰。
      2. 函数的调用栈即菜单位置，无需额外的变量跟踪当前所处的模块位置。
      3. 代码简单

缺点是代码量大，每个函数内都有复杂的逻辑处理。但稍一分析可知，大部分的函数对于循环和部分按键处理逻辑是完全相同的，因此可以使用 C 语言的宏定义简化代码，使其减少重复项，更具有可读性。

**3.2 菜单模块**

菜单模块负责进行人机交互，数据显示，对当前按键作出反馈。电路板上使用8位数码管显示数字、26个英文字母与特殊符号，四个按键分别对应上下调整与前进后腿功能。

形状

描述已自动生成

图 11 显示器和按键功能定义图

理论上，可以将菜单选择完全抽象成一个功能函数，这样不需要手动处理各种按键与显示。但是实际测试中发现函数入参过多可能会导致栈溢出，导致定义的变量被覆盖，因此并没有使用此抽象方法，而是用传统的方式，每个菜单对应一个函数，并使用全局变量防止变量覆盖问题。

**3.3 电机调速模块**

电机测试需要根据设定的参数值调整占空比，处理器根据占空比输出 PWM 波，达到转速调整的目的。而PWM波的延时通过定时中断完成，以避免阻塞而导致无法读取按键的情况。

频繁中断会导致数码管频闪与瞬时的显示错误，因此在退出电机调速时，需要关闭全局中断，避免中断影响到其他程序的正常运行。

中断频率通过调节定时器频率进行调整。中断频率越高，电机转速调节越平滑，但是对显示影响越大。并且即使正确以某一频率初始化计时器，中断频率也基本保持在 60Hz 左右，导致低占空比时能明显看出电机转速的抖动变化。猜测是因为中断处理程序耗时较大，导致中断处理过程中多余的中断被忽略。

基于此行为，可以在代码里使用折算技巧，增大占空比的均匀量化间隔，以降低对中断频率的要求，代价是增大了量化误差。

**3.4 数码管显示模块**

为了满足显示任意字符与数字的功能，提供简单易用的用户接口，需要在程序中内置数显码表。每个7位数码管（加上小数点）需要用一个字节的变量显示，每个二进制位对应一个数显单元的亮暗。本次实验使用的数码管单元 a – h 并不是按照二进制由低到高顺序排布，而是乱序排布，因此给码表的编写造成了一些压力。

此次实验用到了 8 个7位数码管，需要 8 字节的变量来存储显示的字符信息。数码管对应的地址中，下面一排对应着低地址，上面一排对应高地址。而显示时往往上面一排显示主要内容，下面一排显示次要内容，因此还需要进行位次转换。

实验时还发现高频率中断会使数码管产生频闪，某些寄存器可能还会暂时反转。目前没有太好的解决方法，只能在不使用电机时关闭中断。

**3.5 温度读取模块**

实验中，温度的读取使用DS18B20 芯片。该芯片的工作对于时序要求非常严格，并且手动进行每一位的温度转换也是一个挑战。首先需要取高五位判断正负，若小于1 则为负温度，后面的温度值是二进制补码。测得的数值x0.0625 为最终温度。

PPT 上给的DS18B20\_DQ 的定义是 sbit DS18B20\_DQ = P1 ^ 4;但是实际上地址是 P1 ^ 3，因此示例程序无法运行。

**4 设计效果及分析**

实验已成功实现了温控系统所要求的基础功能。

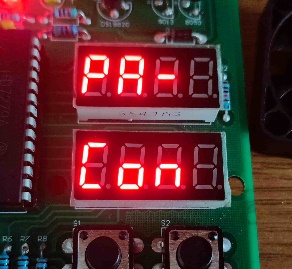
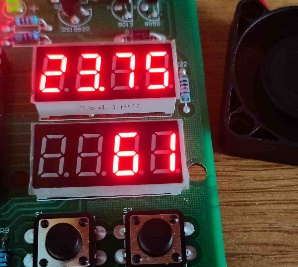
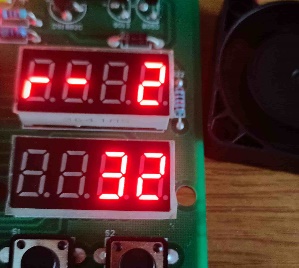
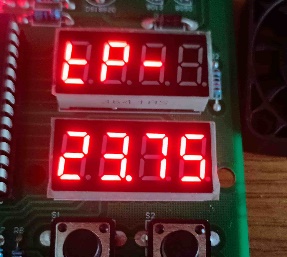


图 12 实时温度显示 图 13 电机测试 图 14 电机温控 图 15 参数调节子菜单

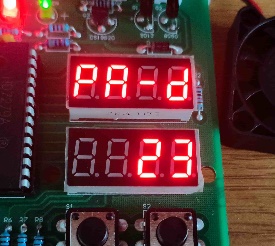
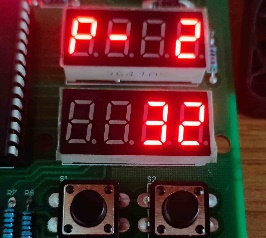


图 16电机工艺单调节 图 17 温度下限调节

**5 结 论**

在这门课程中，我深入地体验电子系统设计的全过程，这不仅仅是对理论知识的运用，更是在实践中的探索和反思。我亲自参与了电路的搭建与焊接、调试和优化，这使我对电子系统设计的相关知识有了更深入的理解和掌握。尽管在实践过程中，我也遇到了一些挫折，比如开发环境的配置和编译器相关知识，24C16芯片的故障也给我带来了一些困扰。但是我还是成功地完成了任务。这次的实践经历让我更加明白，只有通过实践，我们才能真正理解和掌握知识。