2021级计算机专业

计算机硬件(嵌入式)综合实践报告

**组号：13**

组长：李宗霖

组员：万鹰正

组员：杨艺铭

组员：周妍妍

组员：刘 媛

组员：陈国孜

2023-2024学年第1学期

第14周---第16周

**一、系统功能概述及组员分工**

* 1. 系统概述

该系统的核心是AT89C52单片机，配合电阻、电容和晶振等器件构成了最小的单片机系统。其他模块都围绕这个最小的单片机系统展开。其中，我们采用DS18B20传感器来采集温度数据，并将其发送给单片机进行处理。使用4位共阴数码管作为显示设备，用于显示检测到的温度数值。按键模块主要用于设置报警阈值。报警模块则采用蜂鸣器和LED的组合方式，在温度超出报警范围时进行声光报警。同时，还有升温和降温装置工作，以保持温度在一个恒定的范围内。

* 1. 组员分工

李宗霖：统筹小组、程序模块封装、程序设计、程序调试；

万鹰正：音乐程序使用与封装、部分程序功能实现、硬件测试；

杨艺铭：主函数，定时器T0中断，DS18B20，LCD1602流程图绘制；

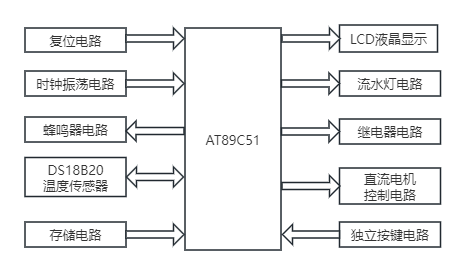
周妍妍：系统使用流程图，文件依赖图，模块依赖图，3个按键系统实现流程图，开机动画流程图，设置页面实现流程图；

刘 媛：绘制原理图、结构框图、代码注释及协助实践报告；

陈国孜：图片代码的整合、会议记录及总结撰写实验报告；

**二、硬件结构框图及介绍、硬件模块简要说明**

2.1 硬件结构框图



**2-1 硬件结构框图**

2.2 硬件模块简要说明

**（1）显示器：**

**LCD显示器简介**

液晶显示器（LCD）是一种广泛应用于各种患子设备中的显示技术，它具有低功耗、低成本、高对比度和可见度良好等优点。

**工作原理**

1.液晶显示原理：

液晶显示的原理是利用液晶的光学特性。液晶是一种特殊的有机化合物，具有电光效应和光电效应。液晶分为向列、向行和像素点三类，其中像素点最小。当液晶显示器处于不通电状态时，液晶分子遵循自然排列规律，液晶被光线穿过，显示器完全透明。当液晶显示器通电时，电场会改变液晶分子的排列，使得液晶产生旋转，从而变成不透明的状态。通过不同液晶分子的排列方式和控制电压的变化，可以实现液晶显示器的显示效果。

2.LCD驱动电路：

液晶显示器需要驱动电路来提供适当的电压和电流，以控制液晶分子的旋转。通常，主要使用2x7串行接口、并行接口或I2C接口等方式与单片机连接。驱动电路中包含液晶显示控制器(LCD Controller)和液晶驱动器(LCD Driver)两部分。

液晶显示控制器(LCD Controller)是单片机内部的一个模块，通过与单片机的通信接口，控制和调整驱动电路工作状态。其主要功能包括计算和产生液晶的驱动信号，实现各种显示模式的切换和刷新频率的设置。

液晶驱动器(LCD Driver)是一个对液晶进行驱动的组件。主要通过产生驱动信号，控制和实现液晶的开关和偏置电压。液晶驱动器通常由多个段驱动(Segment Driver)和多个位驱动(Common Driver)组成。段驱动负责控制液晶的列选通，而位驱动则负责控制液晶的行选通。

**本设计中LCD显示器显示功能**

1.清屏

2.光标返回0位

3.输入字符后 屏幕/光标 移动

4.控制屏幕和光标状态

5.光标/屏幕 左/右移

6.设置数据位 显示行数 点阵

7.设置字库地址

8.设置光标位置

**（2）温度传感器：**

**DS18B20简介**

（1）概述

DS18B20（图3-4）是美国DALLAS半导体公司推出的第一片支持“一线总线”接口的温度传感器，它具有微型化，低功耗，高性能，抗干扰能力强，易配微处理器等优点，可直接将温度转化成数字信号处理器处理。测量的温度范围是-55125℃，测温误差0.5℃。可编程分辨率912位，对应的可分辨温度分别为0.5℃，0.25℃，0.125℃和0.0625℃。相较热电偶传感器而言可实现高精度测温。

（2）特性

独特的1-Wire总线接口仅需要一个管脚来通信；

每个设备的内部ROM上都烧写了一个独一无二的64位序列号；

多路采集能力使得分布式温度采集应用更加简单；

无需外围元件；

供电范围为3.0V至5.5V；

温度可测量范围为：-55℃至+125℃（-67℉至+257℉）；

温度范围超过-10℃至85℃之外时具有±0.5℃的精度；

内部温度采集精度可以由用户自定义为9-Bits至12-Bits；

温度转换时间在转换精度为12-Bits时达到最大值750ms；

用户自定义非易失性的的温度报警设置；

定义了温度报警搜索命令和当温度超过用户自定义的设定值时；

与DS1822程序兼容。

（3）管脚定义

Pin1：(VDD)，可选的电源引脚；

Pin2：(DQ)，单线运用的数据输入/输出引脚；

Pin3：(VDD)，接地端，电源负极；

（4）应用领域

该产品适用于冷冻库，粮仓，储罐，电讯机房，电力机房，电缆线槽等。

轴瓦，缸体，纺机，空调，等狭小空间工业设备测温和控制。

汽车空调、冰箱、冷柜、以及中低温干燥箱等。

供热/制冷管道热量计量，中央空调分户热能计量和工业领域测温和控制。

**（3）ROM存储器**

**AT89C52简介**

（1）概述

AT89C52是一个低电压，高性能[CMOS](https://baike.baidu.com/item/CMOS/0?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/AT89C52/_blank) 8位[单片机](https://baike.baidu.com/item/%E5%8D%95%E7%89%87%E6%9C%BA/102396?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/AT89C52/_blank)，片内含8k [bytes](https://baike.baidu.com/item/bytes/6743146?fromModule=lemma_inlink)的可反复擦写的[Flash](https://baike.baidu.com/item/Flash/5797776?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/AT89C52/_blank)只读[程序存储器](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E5%AD%98%E5%82%A8%E5%99%A8/7661398?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/AT89C52/_blank)和256 bytes的[随机存取](https://baike.baidu.com/item/%E9%9A%8F%E6%9C%BA%E5%AD%98%E5%8F%96/4610937?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/AT89C52/_blank)数据存储器（[RAM](https://baike.baidu.com/item/RAM/144481?fromModule=lemma_inlink)），器件采用[ATMEL公司](https://baike.baidu.com/item/ATMEL%E5%85%AC%E5%8F%B8/2002428?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/AT89C52/_blank)的[高密度](https://baike.baidu.com/item/%E9%AB%98%E5%AF%86%E5%BA%A6/2399022?fromModule=lemma_inlink)、非易失性[存储技术](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E6%8A%80%E6%9C%AF/10874571?fromModule=lemma_inlink)生产，兼容标准MCS-51[指令系统](https://baike.baidu.com/item/%E6%8C%87%E4%BB%A4%E7%B3%BB%E7%BB%9F/3220297?fromModule=lemma_inlink)，片内置通用8位[中央处理器](https://baike.baidu.com/item/%E4%B8%AD%E5%A4%AE%E5%A4%84%E7%90%86%E5%99%A8/284033?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/AT89C52/_blank)和Flash[存储单元](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E5%8D%95%E5%85%83/0?fromModule=lemma_inlink)，AT89C52单片机在电子行业中有着广泛的应用。

AT89C52有40个引脚，32个外部双向输入/输出(I/O)端口，同时内含2个外中断口，3个16位可编程定时[计数器](https://baike.baidu.com/item/%E8%AE%A1%E6%95%B0%E5%99%A8/4699680?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/AT89C52/_blank),2个[全双工](https://baike.baidu.com/item/%E5%85%A8%E5%8F%8C%E5%B7%A5/310007?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/AT89C52/_blank)[串行通信](https://baike.baidu.com/item/%E4%B8%B2%E8%A1%8C%E9%80%9A%E4%BF%A1/324843?fromModule=lemma_inlink)口，2 个读写口线，AT89C52可以按照常规方法进行编程，也可以在线编程。其将通用的[微处理器](https://baike.baidu.com/item/%E5%BE%AE%E5%A4%84%E7%90%86%E5%99%A8/104320?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/AT89C52/_blank)和[Flash存储器](https://baike.baidu.com/item/Flash%E5%AD%98%E5%82%A8%E5%99%A8/1254787?fromModule=lemma_inlink)结合在一起，特别是可反复擦写的 Flash存储器可有效地降低[开发成本](https://baike.baidu.com/item/%E5%BC%80%E5%8F%91%E6%88%90%E6%9C%AC/6160378?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/AT89C52/_blank)

（2）主要功能特性

 ◆兼容MCS51指令系统；

 ◆8k可反复擦写(>1000次）Flash ROM；

 ◆32个双向I/O口；

 ◆256x8bit内部RAM ；

 ◆3个16位可编程定时/计数器中断；

 ◆时钟频率0-24MHz；

 ◆2个串行中断；

 ◆可编程UART串行通道；

 ◆2个外部中断源；

 ◆共8个中断源；

 ◆2个读写中断口线；

 ◆3级加密位；

 ◆低功耗空闲和掉电模式；

 ◆软件设置睡眠和唤醒功能；

**（4）驱动芯片：**

**芯片ULN2003简介**

ULN2003是高耐压、[大电流](https://baike.baidu.com/item/%E5%A4%A7%E7%94%B5%E6%B5%81/5467824?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/ULN2003/_blank)复合晶体管阵列，由七个硅NPN [复合晶体管](https://baike.baidu.com/item/%E5%A4%8D%E5%90%88%E6%99%B6%E4%BD%93%E7%AE%A1/8881496?fromModule=lemma_inlink)组成，每一对达林顿都串联一个2.7K 的[基极](https://baike.baidu.com/item/%E5%9F%BA%E6%9E%81/763327?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/ULN2003/_blank)电阻，在5V 的[工作电压](https://baike.baidu.com/item/%E5%B7%A5%E4%BD%9C%E7%94%B5%E5%8E%8B/10983510?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/ULN2003/_blank)下它能与[TTL](https://baike.baidu.com/item/TTL/13973249?fromModule=lemma_inlink) 和CMOS 电路直接相连，可以直接处理原先需要标准逻辑[缓冲器](https://baike.baidu.com/item/%E7%BC%93%E5%86%B2%E5%99%A8/3697522?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/ULN2003/_blank)来处理的数据。

**ULN2003的作用**

ULN2003是[大电流](https://baike.baidu.com/item/%E5%A4%A7%E7%94%B5%E6%B5%81/5467824?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/ULN2003/_blank)驱动阵列,多用于[单片机](https://baike.baidu.com/item/%E5%8D%95%E7%89%87%E6%9C%BA/102396?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/ULN2003/_blank)、[智能仪表](https://baike.baidu.com/item/%E6%99%BA%E8%83%BD%E4%BB%AA%E8%A1%A8/7691769?fromModule=lemma_inlink)、[PLC](https://baike.baidu.com/item/PLC/275974?fromModule=lemma_inlink)、[数字量](https://baike.baidu.com/item/%E6%95%B0%E5%AD%97%E9%87%8F/10872463?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/ULN2003/_blank)输出卡等控制电路中。可[直接驱动](https://baike.baidu.com/item/%E7%9B%B4%E6%8E%A5%E9%A9%B1%E5%8A%A8/56544839?fromModule=lemma_inlink)继电器等[负载](https://baike.baidu.com/item/%E8%B4%9F%E8%BD%BD/906913?fromModule=lemma_inlink)。

本设计中通过芯片 ULN2003 驱动直流 5v 电机，采用定时器产生 PWM 波对电机调速。

1. **风扇电机调速方法：**

定时器T0产生1/3.6 ms的时序中断，用count循环计数，分成三段：1440、2520、3600，分别对应0-0.4ms，0.4-0.7ms，0.7-1ms，一档0-0.4产生高电平...以此类推。

**（6）按键：**

三套按键系统：

第一套: 视图模式下 全部上升沿(松开按键时)触发

S1: 主视图 (温度信息查询视图)

S2: 最高/最低温(温度极值)查询视图

S3: 温度越界计时视图

S4: 设置查询视图

第二套: 设置模式下 全部上升沿(松开按键时)触发

S1: 修改当前选项值

S2: 下一项

S3: 上一项

S4: 回到第一项

第三套: 修改选项时

S1: 取消 上升沿触发

S2: 加 低电平触发 按下触发一次 长按超过阈值 每隔一段时间间隔触发

S3: 减 低电平触发 按下触发一次 长按超过阈值 每隔一段时间间隔触发

S4: 确认 上升沿触发

**（7）蜂鸣器：**

**蜂鸣器简介**

蜂鸣器是一种集成结构的电子发声设备，使用直流电压供电，在计算机、打印机、复印机、报警器、电子玩具、汽车电子设备、电话机、定时器等电子产品中广泛应用。

蜂鸣器可以分为压电式蜂鸣器和电磁式蜂鸣器两种类型。压电式蜂鸣器由多谐振荡器、压电蜂鸣片、阻抗匹配器、共鸣箱和外壳等组成。当电源接通后，多谐振荡器开始振荡，输出1.5~2.5kHz的音频信号，阻抗匹配器推动压电蜂鸣片发出声音。电磁式蜂鸣器由振荡器、电磁线圈、磁铁、振动膜片和外壳等组成。接通电源后，振荡器产生音频信号电流，通过电磁线圈生成磁场。振动膜片在电磁线圈和磁铁的作用下周期性振动并发声。本设计采用的是电磁式蜂鸣器。

此外，还有有源蜂鸣器和无源蜂鸣器的区别。这里的“源”指的是震荡源，有源蜂鸣器内部集成了震荡源，只需接通电源即可发声；而无源蜂鸣器则未集成震荡源，如果使用直流信号无法使其发声，需要使用2K-5K的方波信号驱动。本设计使用的是有源蜂鸣器。

由于蜂鸣器需要较大的电流才能正常工作，而单片机的IO口输出电流较小，因此在这里使用三极管的开关功能来控制蜂鸣器发声。本设计选择的三极管型号是PNP三极管S8550，并且所选用的蜂鸣器属于有源蜂鸣器，即蜂鸣器内部已经包含震荡电路，单片机只需输出高（或低）电平即可，无需连续发出高低电平信号，从而简化了单片机程序设计。由于选用的是PNP型三极管，而单片机上电IO口默认为高电平，所以在上电时蜂鸣器不会发出鸣叫声。

1. **继电器：**

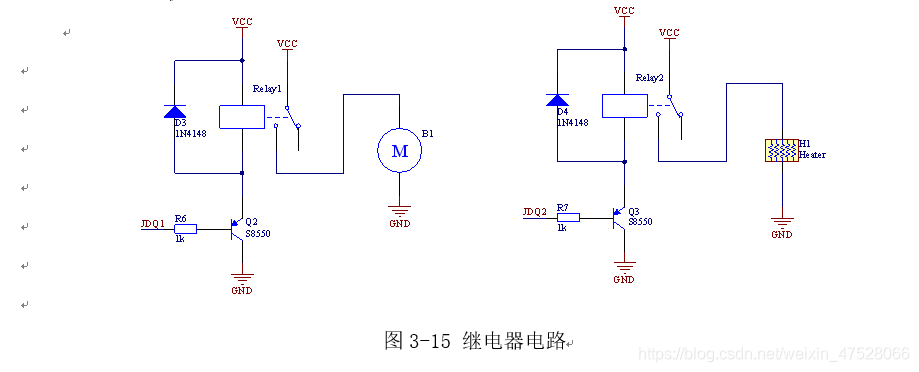
**继电器简介**

继电器作为一种电控制器件，在电气输出电路中可以使被控量发生预定的阶跃变化，其控制系统（输入回路）和被控制系统（输出回路）之间有着互动关系。通常应用于自动化的控制电路中，它可通过小电流去控制大电流的运作，是一种“自动开关”。因此，在电路中可以起到自动调节、安全保护、转换电路等作用。

根据继电器的工作原理和结构特征分类，我们可以将其大致分为电磁继电器、固态继电器、温度继电器、舌簧继电器、时间继电器、高频继电器、极化继电器、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、差动继电器等多种类型。

在本设计中，我们采用了电磁式继电器。电磁继电器主要由铁芯、线圈、衔铁、触点簧片等组成。只要在线圈两端加上一定的电压，线圈中就会流过一定的电流，从而产生电磁效应。这时，衔铁会在电磁力的吸引作用下克服返回弹簧的拉力吸向铁芯，带动衔铁的动触点与静触点（常开触点）吸合。当线圈断电后，电磁的吸力也随之消失，衔铁就会在弹簧的反作用力返回原来的位置，使动触点与原来的静触点（常闭触点）释放。这样的吸合和释放过程，使得继电器能够在电路中实现导通和切断效果。需要说明的是，继电器的“常开触点”指的是线圈未通电时处于断开状态的静触点，“常闭触点”则指处于接通状态的静触点。一般来说，继电器具有低压控制电路和高压工作电路两个电路。

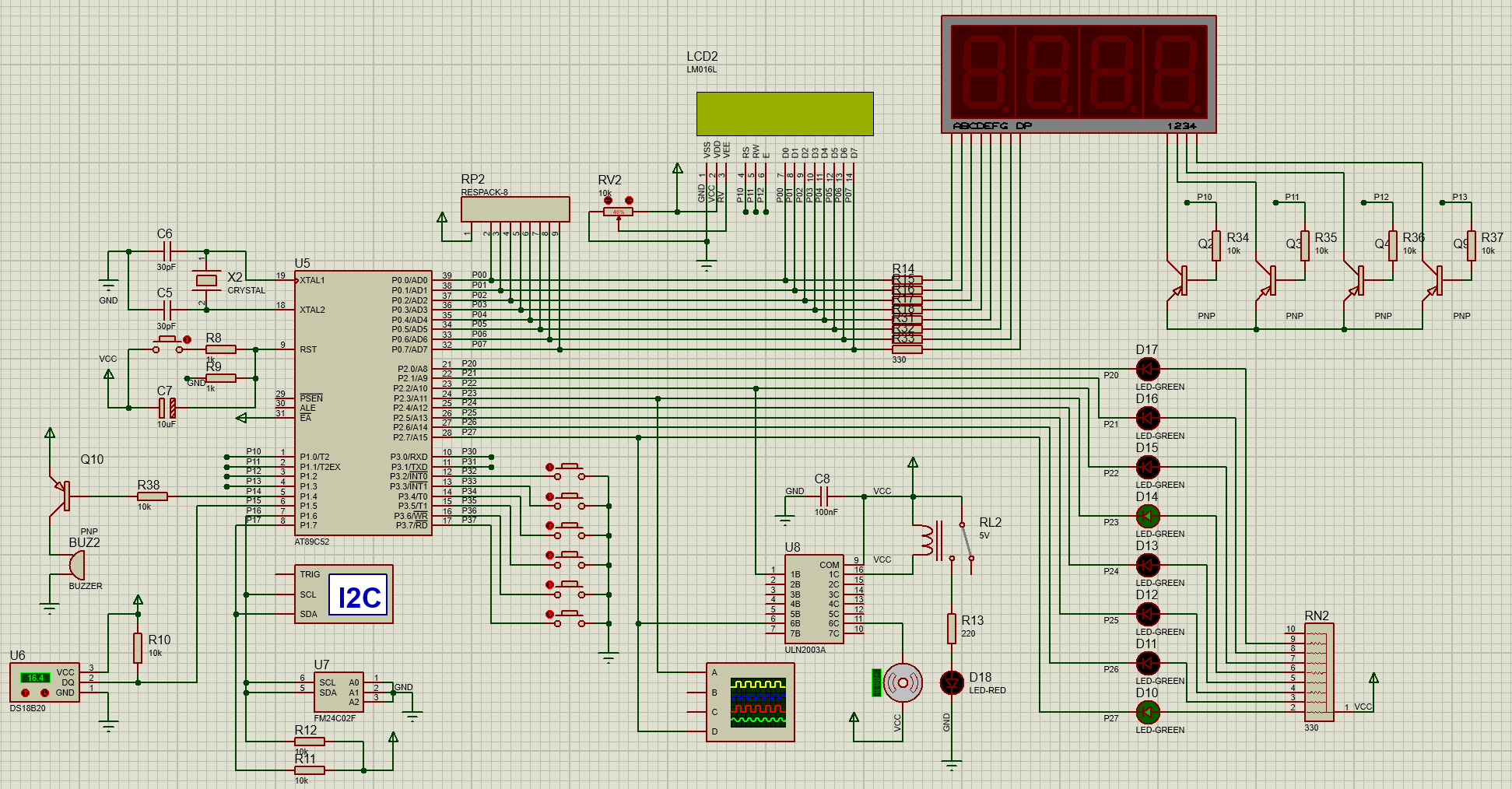
继电器模块的电路如下图（图2-2）所示。继电器使用的是5V电压触发的。由于继电器由导通到关断瞬间，由于工作线圈有电感的性质，所以会在继电器的线圈的低电压端产生一个瞬间电压尖峰，通常能高达数十倍的线圈额定工作电压。所以这里接入一个二极管在继电器两端，因为二极管的负端通常接到VCC，因此电压尖峰将被抑制。保护了板上的电子元件。当单片机的IO口给PNP三极管一个低电平后后，三极管导通，继电器供电，因此继电器从断开变为闭合，最终将小风扇或加热膜接入5V电压，这样就实现了单片机控制加温或降温的效果。



**2-2 继电器电路图**

**三、电路原理图设计和接口规范**

3.1 原理图



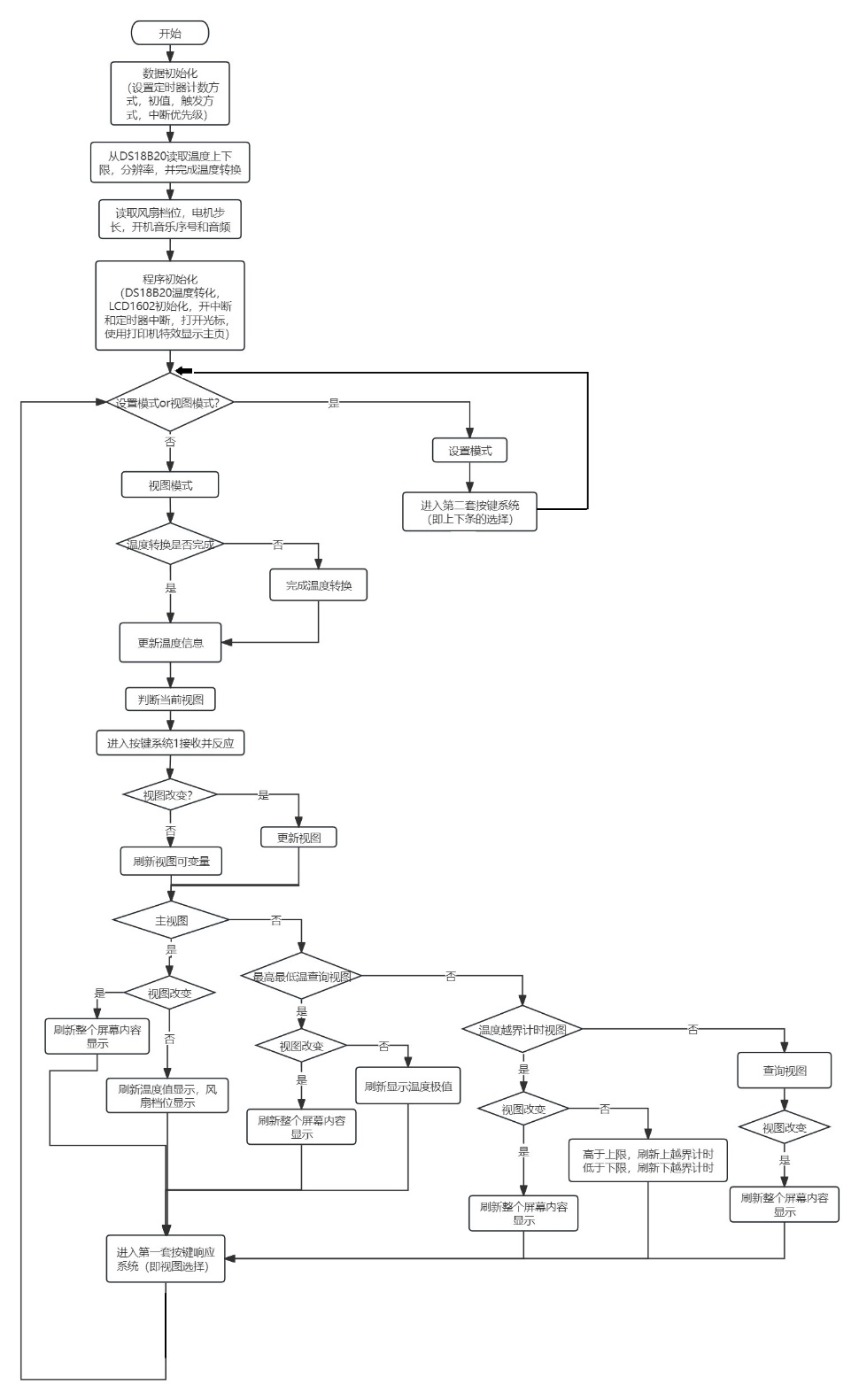
**3-1 原理图**

3.2 接口规范

变量采用小驼峰命名、标志位采用下划线命名、函数采用大驼峰命名、常量使用大写或者大写简写、宏定义采用大写命名。

**四、软件功能设计**

4.1 主程序流程图



**4-1主函数模块**

4.2 main函数代码

void main(void)

{

/\*\*

\* 初始化数据:

\* 1. 初始化 定时/计数器 对应的方式初值 优先级

\* 2. 初始化 外部中断 对应的方式 初值 优先级

\* 3. 从 DS18B20 读取 温度上限 下限 分辨率

\* 4. 从 24LC02 读取 ...

\*/

init\_data();

/\*\*

\* 初始化程序:

\* 1. 开始温度转换

\* 2. LCD1602初始化 显示开机界面

\* 3. 以打字机效果显示主视图

\* 4. 打开 定时/计数器中断 外部中断

\*/

init\_program();

/\*\*

\* 程序的主循环:

\* 1. 判断是哪一种模式并执行模式对应的程序

\* 2. 模式主程序:

\* - 设置模式

\*

\* - 视图模式

\* 1. 如果温度转换完成 更新温度信息

\* 2. 判断当前处于哪一个视图

\* 3. 如果视图发生改变 需要更新整个视图

\* 4. 如果视图没有改变 刷新当前视图的 可变量

\* 5. 以 按键系统1 接收按键操作 并反应

\*

\* 外部中断:

\* X0: 长按切换设置模式和视图模式

\*/

while (1)

{

if (settings\_mode) // 设置模式

{

if (ready\_settings)

{

ready\_settings = 0;

ShowSettings(0); // 显示设置模式 并指向第一条

}

KeysSystem\_2(); // 第二套按键事件响应系统

}

else // 视图模式

{

if (save\_in\_24c02)

{

save\_in\_24c02 = 0;

At24c02\_WriteByte(0xa0, 0x00, &settingsSave, 1);

}

if (convert\_finished)

{ // 如果温度转换完成 更新温度信息

UpdateTemperature();

convert\_finished = 0;

}

UpdateViewPageShow(); // 刷新视图显示

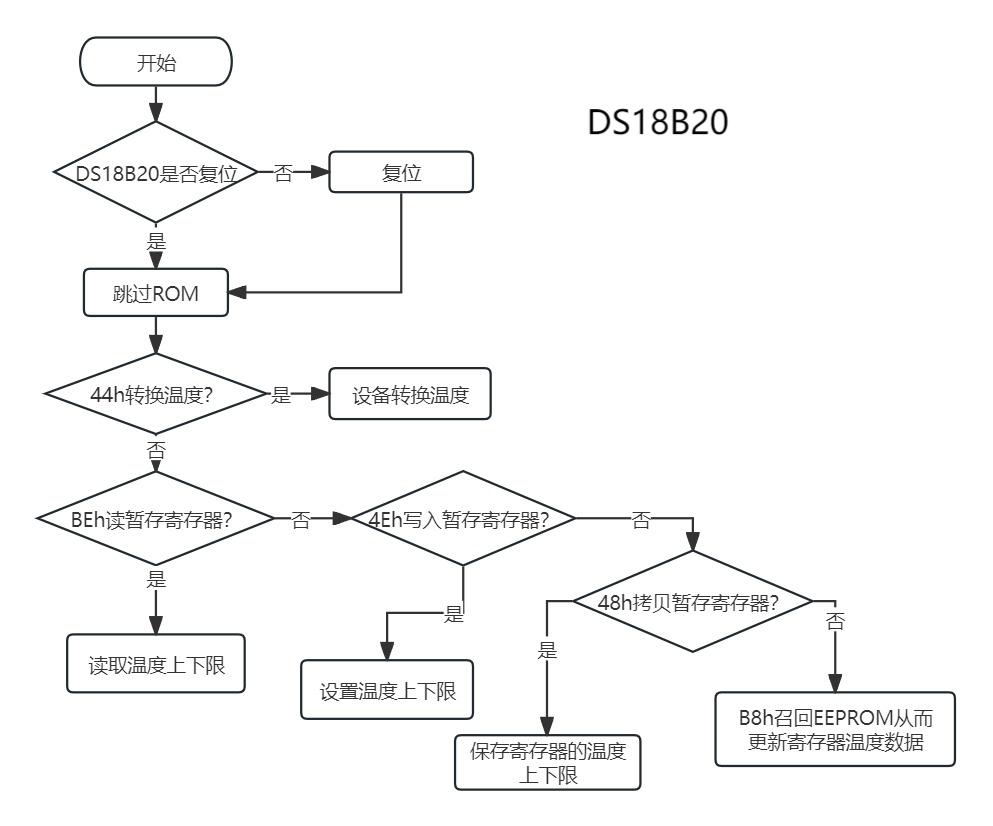
KeysSystem\_1(); // 第一套按键事件响应系统

}

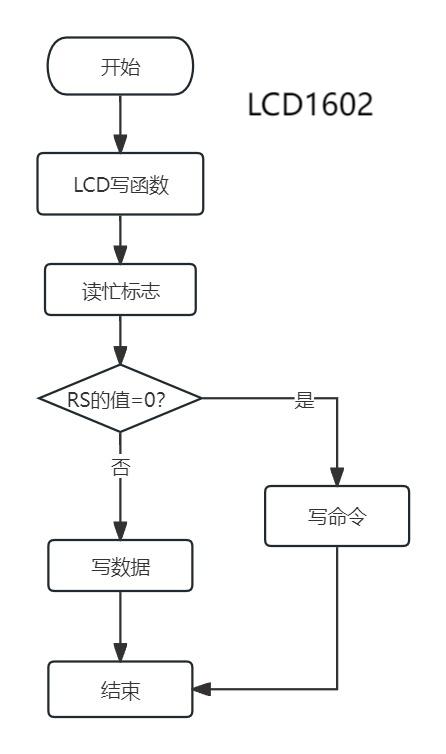
}

}

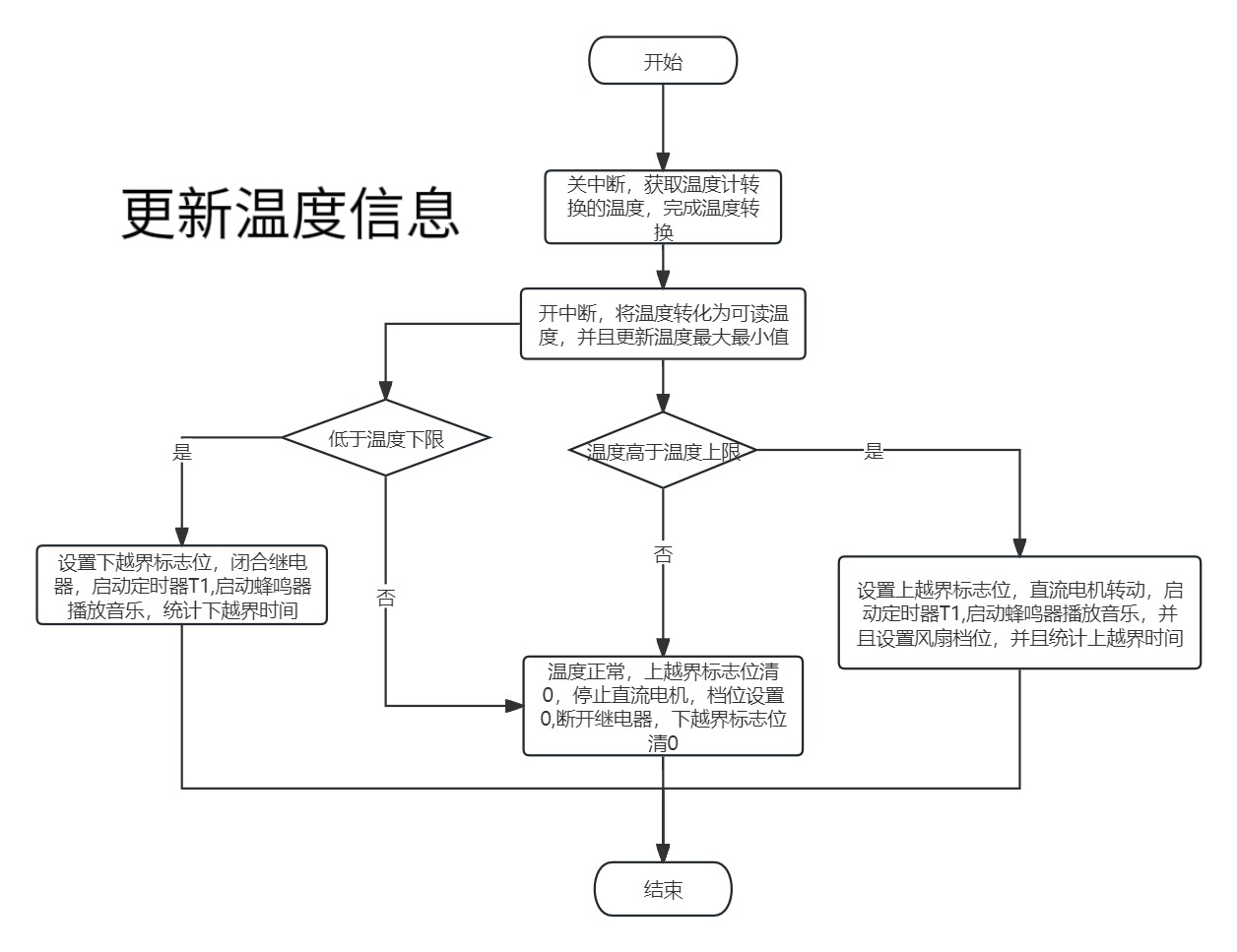
4.3 各模块程序流程图



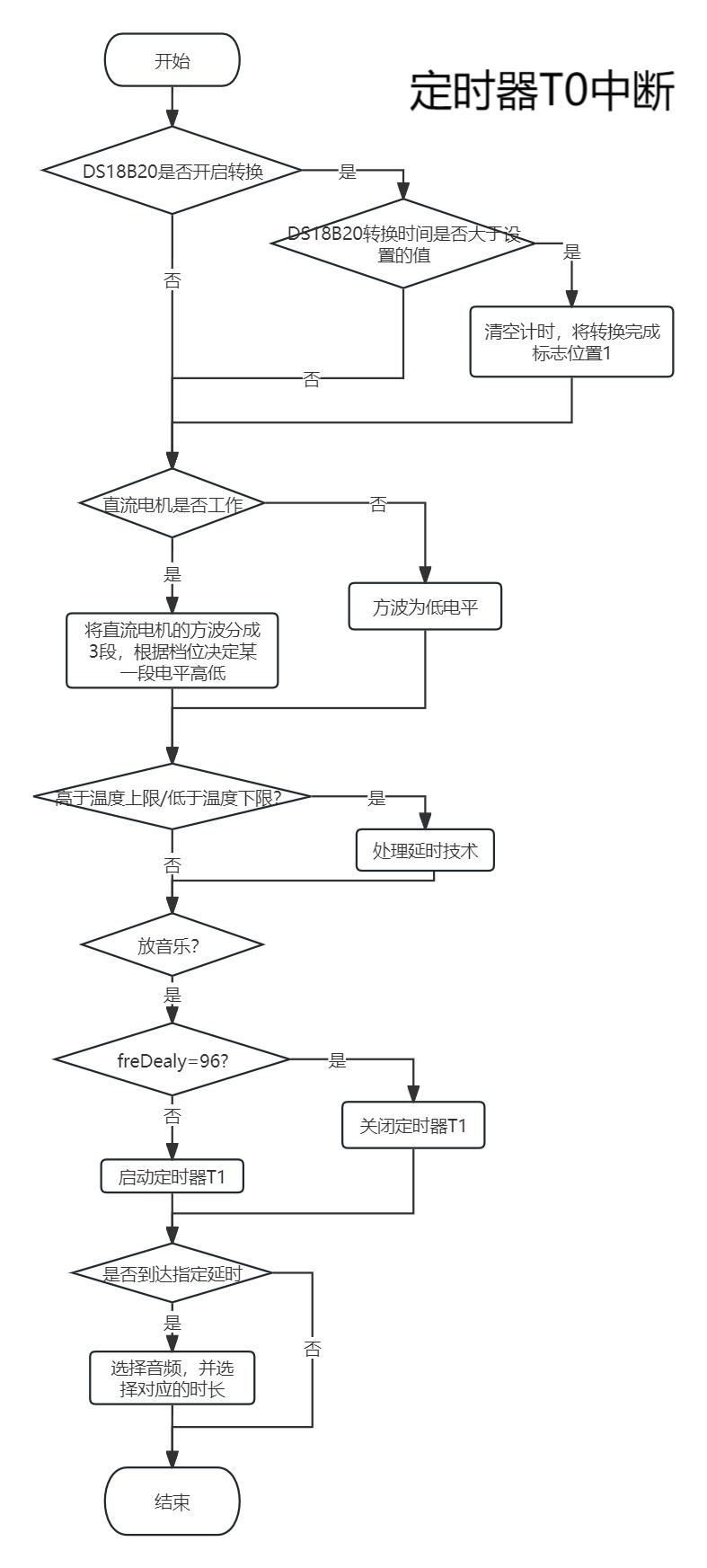
**4-2 DS18B20模块**



**4-3 lcd模块**

****

**4-4更新温度信息模块**

****

**4-5定时器T0中断模块**

4.4 关键函数和代码

**lcd1602.h**

#ifndef LCD1602\_H

#define LCD1602\_H

// 定义LCD1602基本操作函数

extern void LCD1602\_CheckBusy (void); // 检查忙标志位 extern void LCD1602\_WriteCmd (unsigned char cmd); // 写入命令

extern void LCD1602\_WriteData (unsigned char dat); // 写入数据

extern unsigned char LCD1602\_ReadData (void); // 读数据

#endif

**ds18b20.h**

#ifndef DS18B20\_H

#define DS18B20\_H

extern void DS18B20\_Convert (); // 温度转换

extern int DS18B20\_ReadTemp(); // 温度读取

extern void DS18B20\_Set(

unsigned char upperLimit, unsigned char lowerLimit, unsigned char resolution

);

extern void DS18B20\_Get(

unsigned char\* upperLimit,

unsigned char\* lowerLimit,

unsigned char\* resolution

);

extern void DS18B20\_Save(void);

#endif // DS18B20\_H

**at24c02.h**

#ifndef AT24C02\_H

#define AT24C02\_H

/\*\*

\* @param sla 器件地址码

\* @param suba 器件单元地址

\* @param dat 数据的首地址

\* @param num 数据的个数

\* @return 0: 未完成 中间某个环节未应答 1: 完成

\*/

extern bit At24c02\_ReadData(

unsigned char sla, unsigned char suba, unsigned char\* dat, unsigned char num

); // 可以读任意范围内长度的数据

extern bit At24c02\_WriteByte(

unsigned char sla, unsigned char suba, unsigned char\* dat, unsigned char num

); // 一次最多写入8字节 多了会循环覆盖掉 需要自己添加延迟

extern bit At24c02\_WriteData(

unsigned char sla, unsigned char suba, unsigned char\* dat, unsigned char num

); // 可以写入任意多范围内的数据 内置延迟 可以连续操作

#endif // AT24C02\_H

**五、系统调试过程中出现的问题及解决方法**

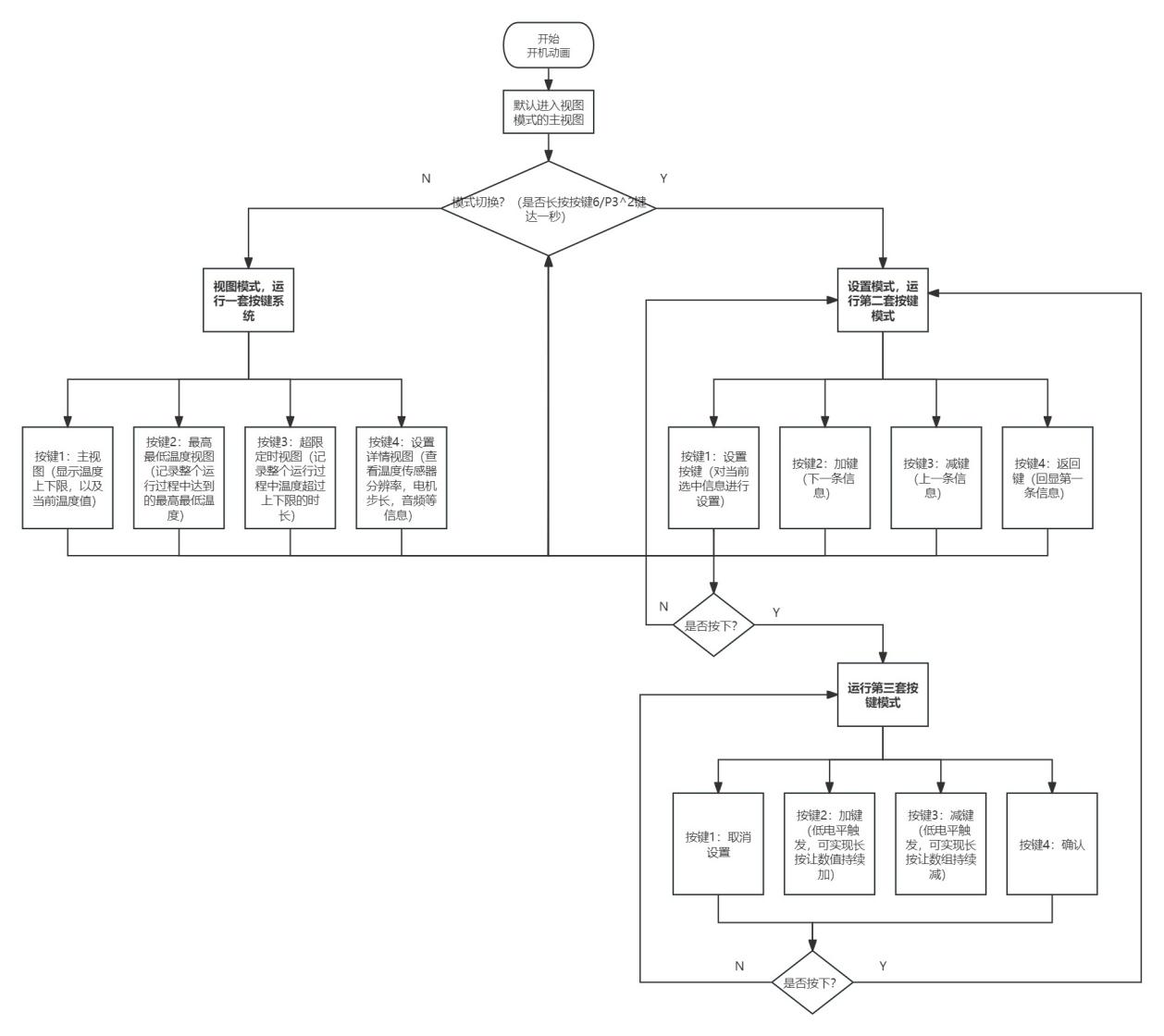
5.1 硬件电路测试

（1）DS18B20对于EEPROM的写入失败成功设置RAM但是没有存储到ROM(软件没有问题)但是有时候会出现混乱的值；猜测：写入过程被中断；方案：在写入函数后添加延迟；结果：成功；  
（2）定时不精确；方案：精确计算机器周期并用keil5调试进行验证；结果：成功，误差预计在每10us以内；  
（3）软件调试没有问题 硬件程序烧录后单片机没有反应；测验：降低idata区的使用量可以解决，但是不能降低否则无法播放音乐；猜测：因为是一点反应没有，怀疑没有找到程序入口；方案：注释掉所有未被调用的函数重编译；结果：程序从7100byte缩小到6900+byte，同样解决，但不清楚具体成因；  
（4）蜂鸣器音乐未停止；猜测：结构不一样；方案：停止时将蜂鸣器电平拉高；结果：成功；结论：电路板是有源蜂鸣器，低电平时响，我们设计的是无源蜂鸣器的播放，所以其实这个蜂鸣器并不适合播放音乐，造成音乐难听；

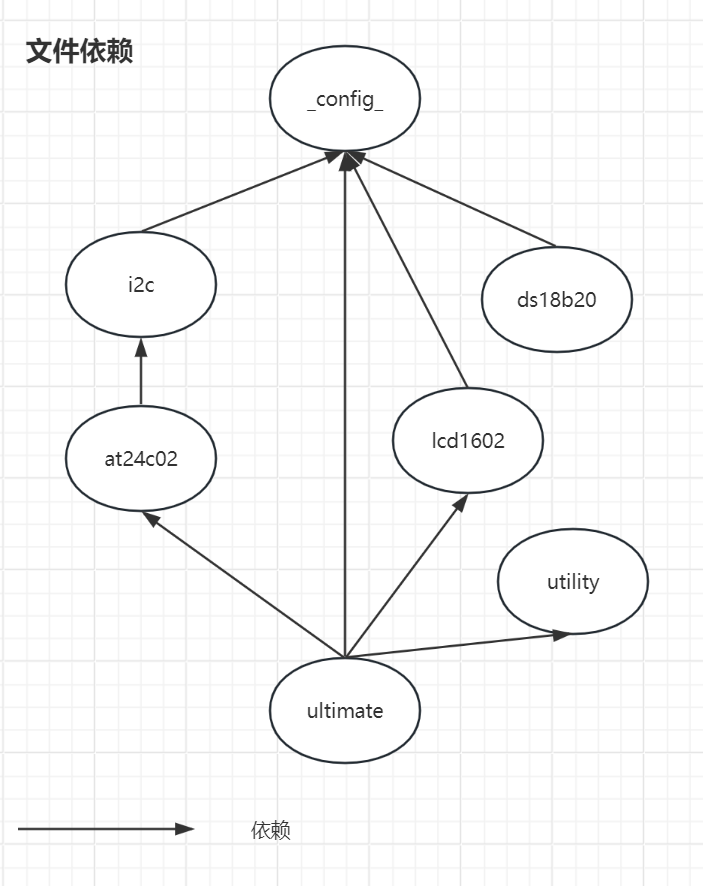
（5）当电机档位到达三档；lcd不显示；猜测：硬件电压不够同时驱动蜂鸣器，ds18b20，直流电机，lcd1602；解决方案：无；因为一档正常显示，二档屏幕闪烁显示；三档屏幕不显示，正好起警示效果，反而让硬件效果更好；

5.2 软件程序调试  
（1）ds18b20 温度显示异常，通过硬件延迟太浪费；方案改为中断方式；结果：成功；  
（2）减少中断函数‘体积’，将获取温度放到主函数，结果又出现胡乱显示，但是转换时间给够了；猜测：ds18b20对时序深度依赖，读取过程触发中断导致失败；方案：对ds18b20操作前关闭所有中断；结果：成功；  
（3）中断关闭导致定时不准确；方案：计算关闭器件机器周期，开启后补上；结果：成功；  
（4）外部中断出现一个特定的函数，哪怕绕过仍会导致陷入死机；原因：未知怀疑是编译bug；方案：设置标志位，将函数调用放在主函数；结果：成功；  
（5）蜂鸣器播放音乐必须不能被中断，T0定时模块如果不是唯一最高优先级会导致定时不准确，二者冲突；方案：不启动蜂鸣器时T0为高优先级，启动蜂鸣器时T1为高优先级，T0为低优先级，T1中断函数很短必定不会连续阻挡T0两次，所以在T1结束时判断TF0，如果为1，则运行一次T0中断内容；结果：完美解决；  
（6）软件仿真发现蜂鸣器声音紊乱，检查发现现实过去很久才定时了0.1s；猜测：proteus软件卡了；方案：删掉不需要的模块，例如发光二极管；结果正常运行；  
（7）data区内存超出导致编译失败；方案：将大型数组指定idata区，常量以及常量数组指定code区；结果：成功；

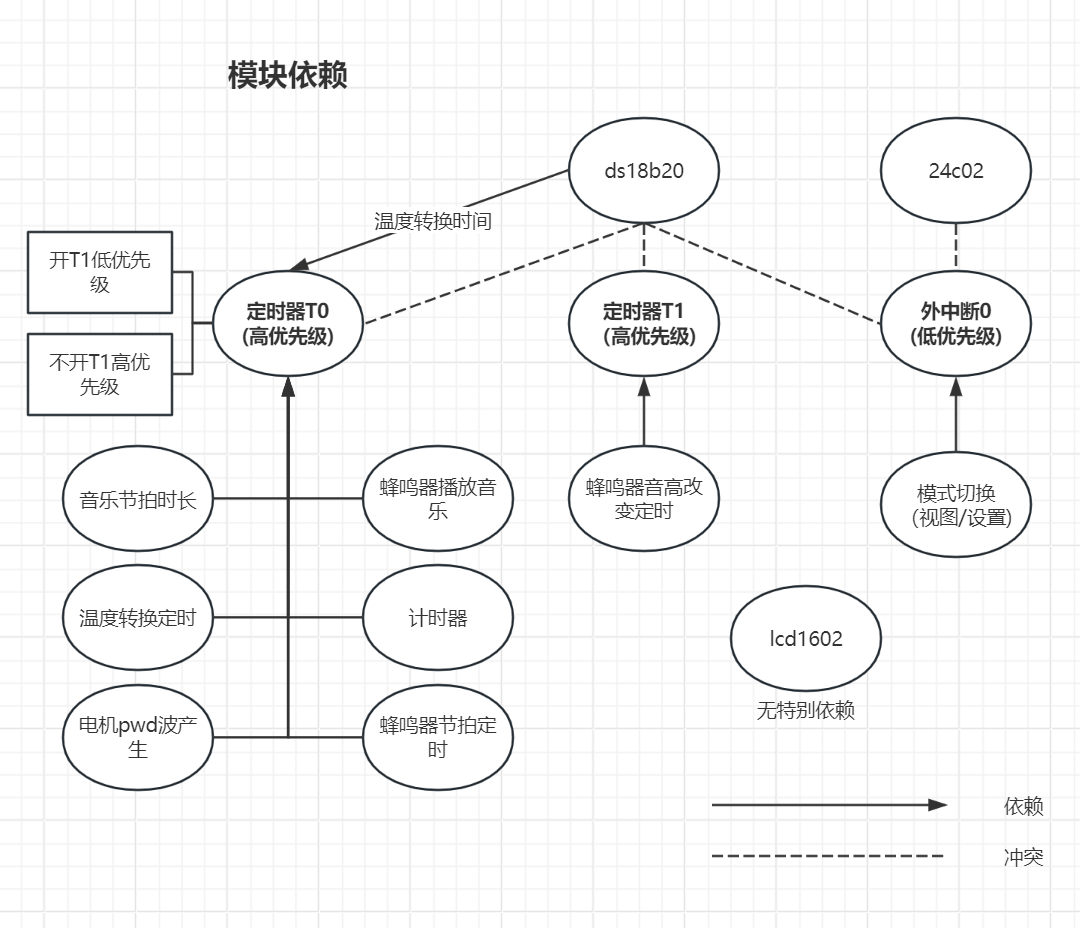
1. **系统使用操作说明**

****

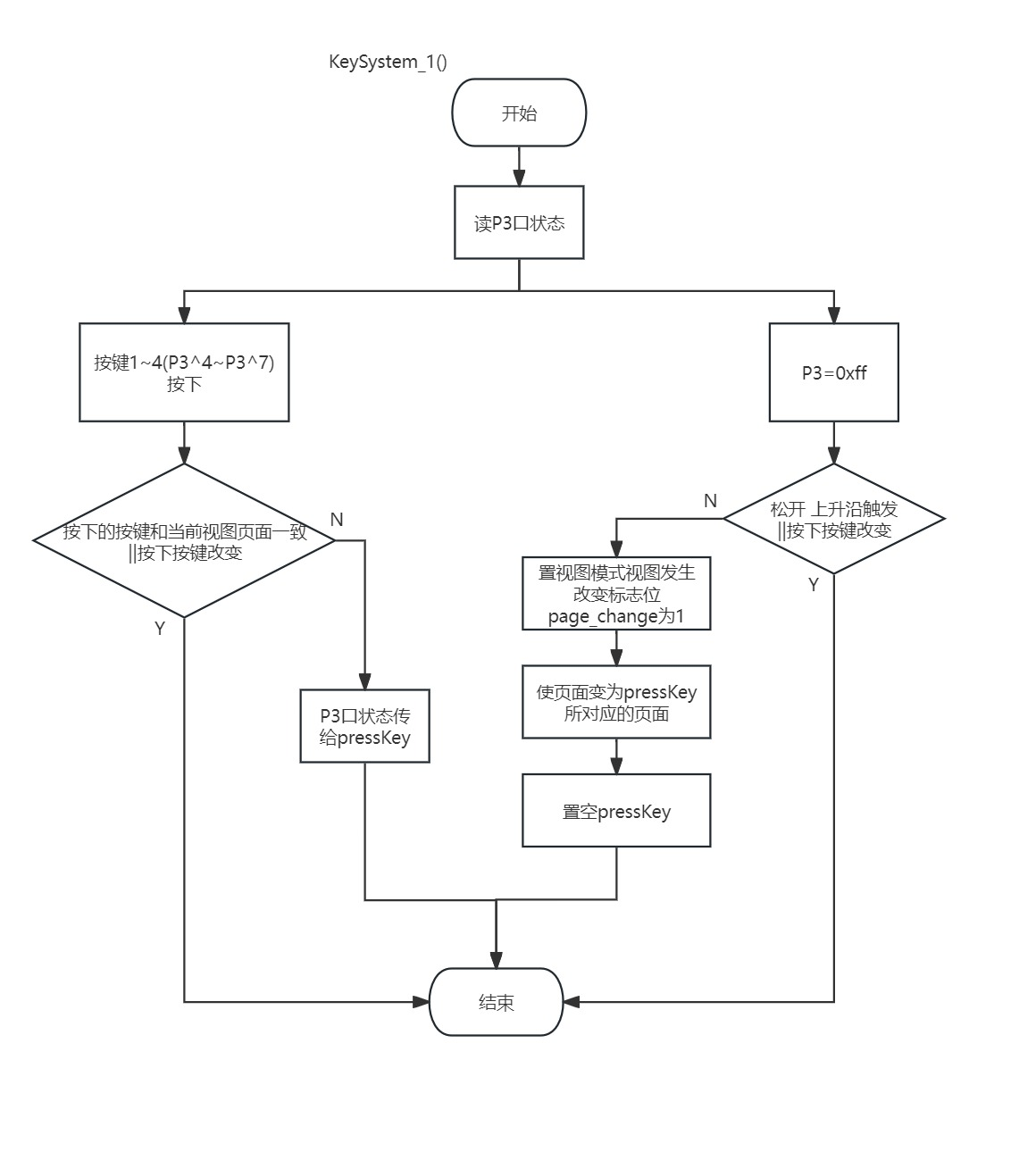
**6-1 系统使用流程图**

****

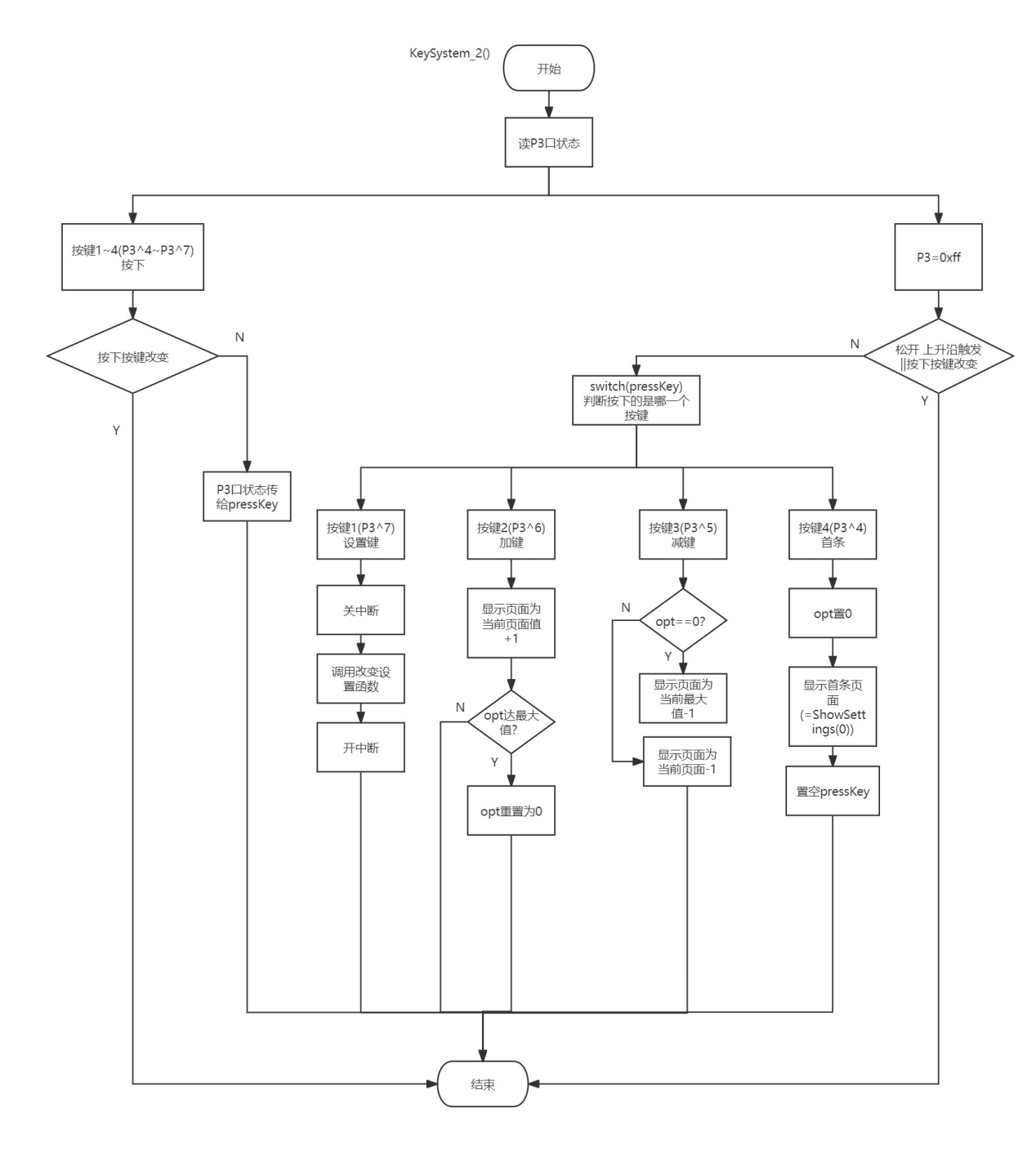
**6-2 文件依赖图**

****

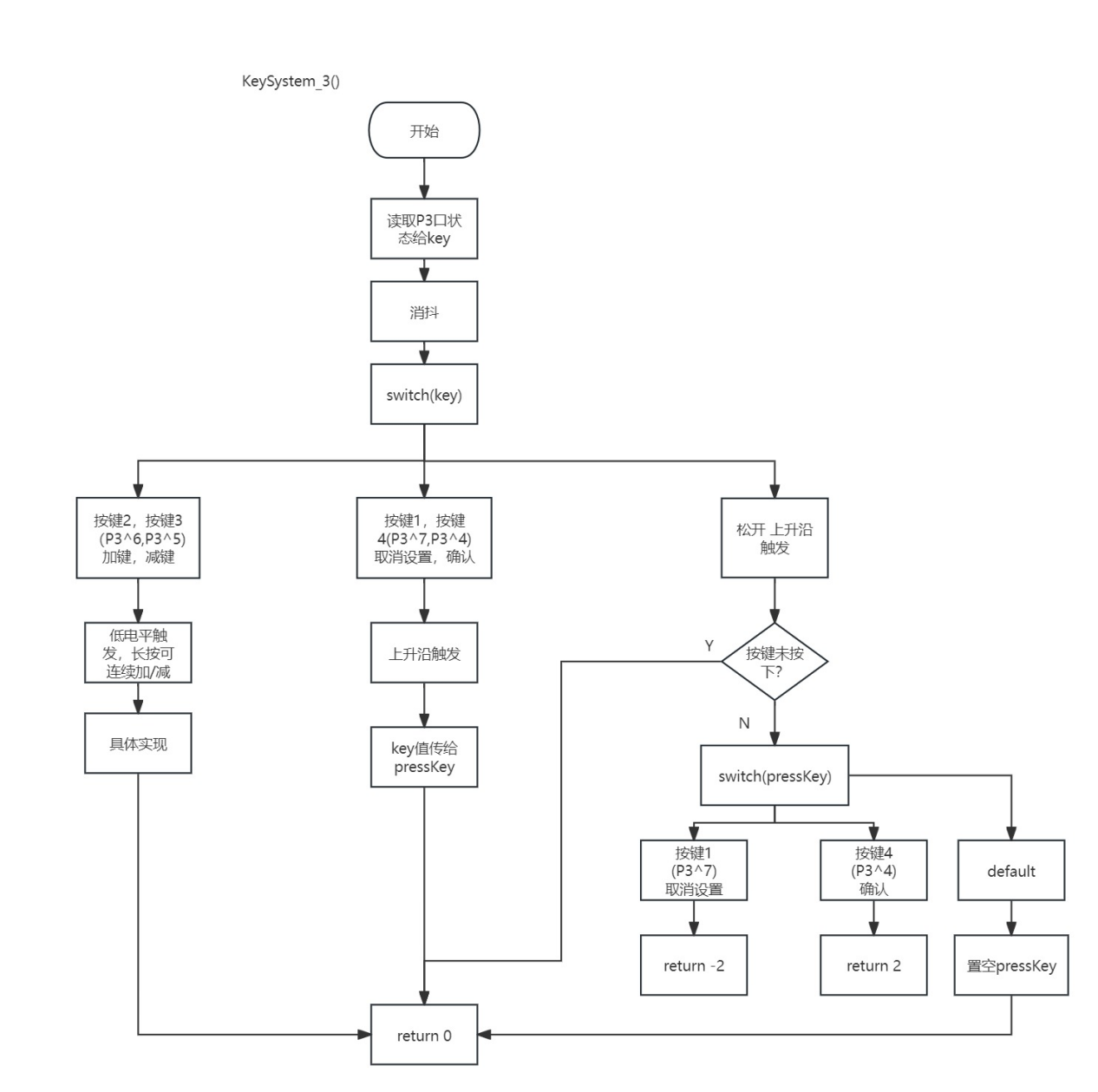
**6-3 模块依赖图**

****

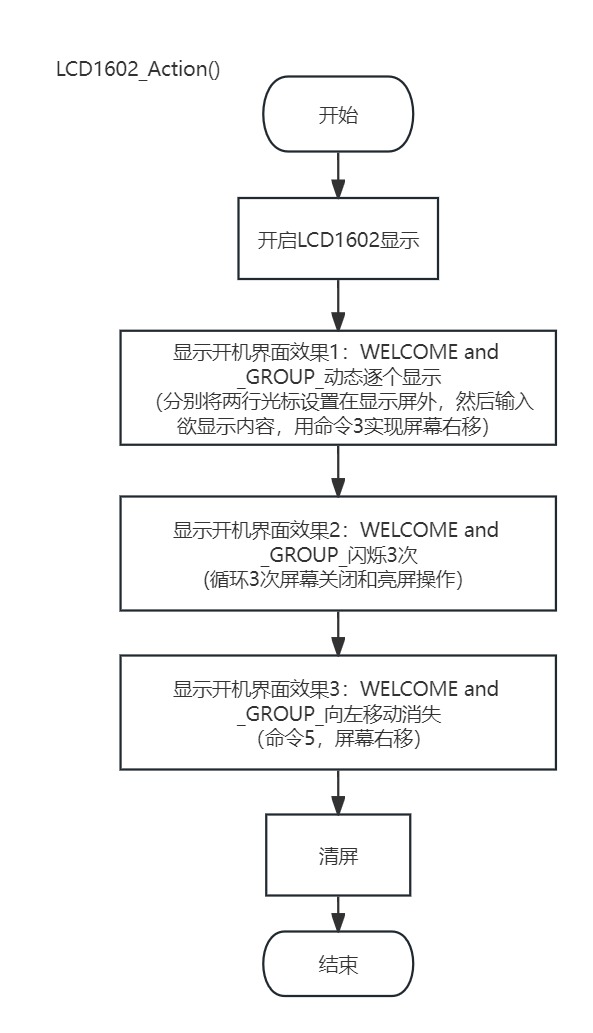
**6-4 按键系统1实现流程图**

****

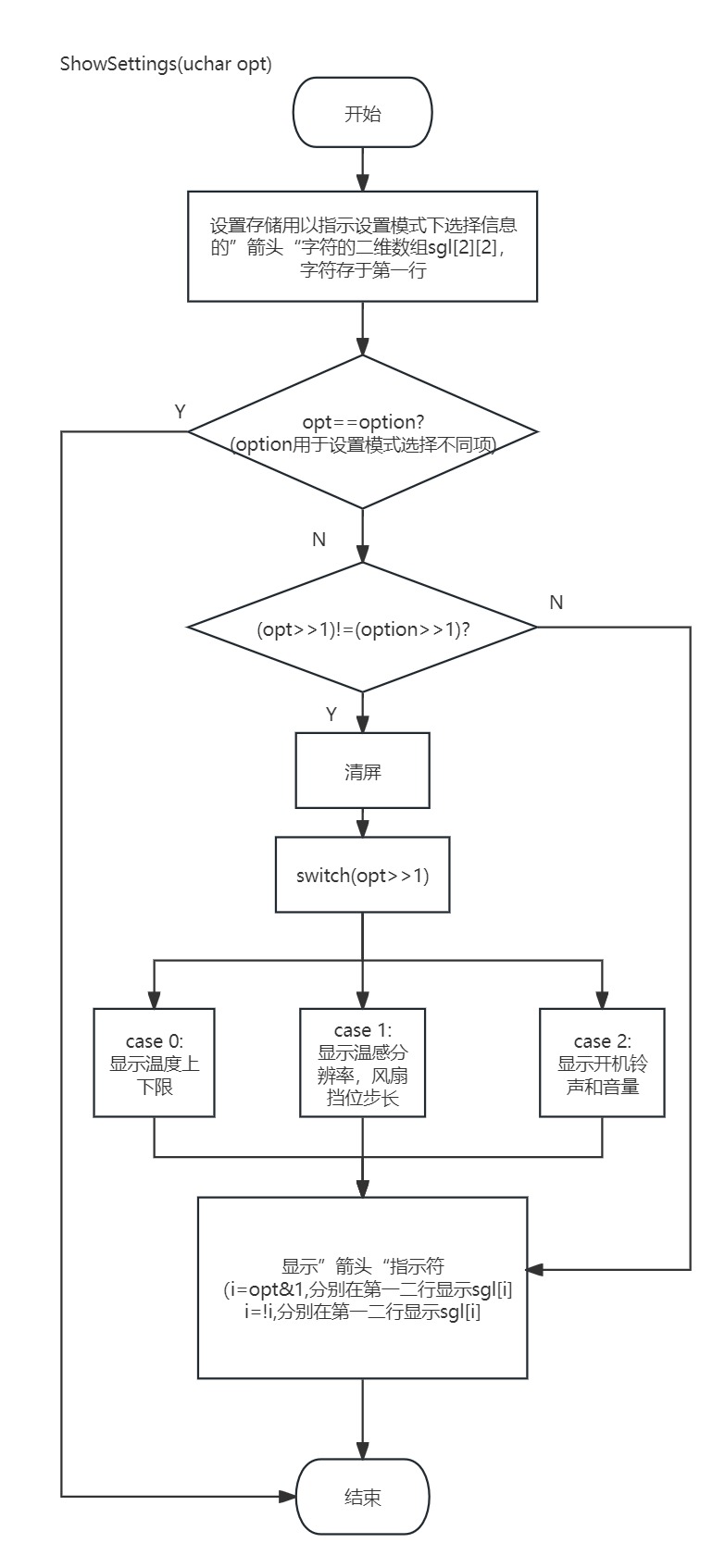
**6-5 按键系统2实现流程图**

****

**6-6 按键系统3实现流程图**

****

**6-7 开机动画流程图**

****

**6-8 设置页面实现流程图**

**实物效果展示：**



**6-9 开机显示**



**6-10 温度显示界面**



**6-11 最高/低温度界面**



**6-12 温度越界计时界面**



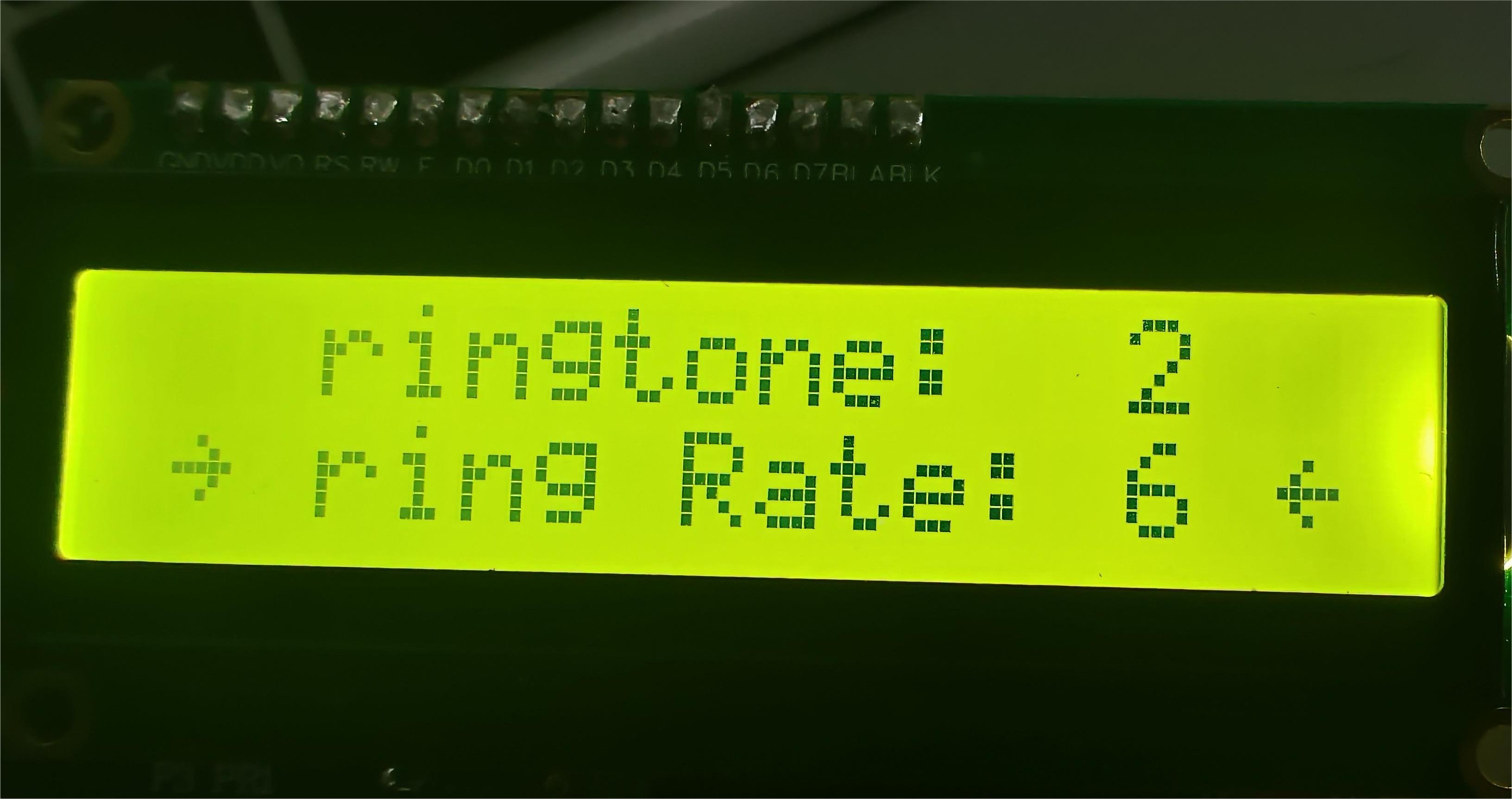
**6-13 音乐播放界面**



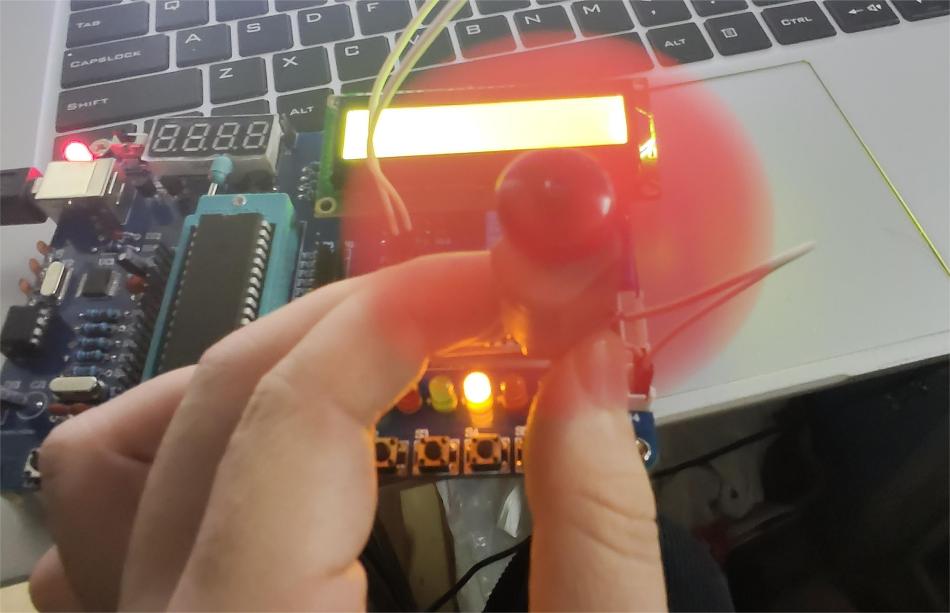
**6-14 温度上下限设置**



**6-15 分辨率及风扇档位步长设置**



**6-16 音乐及铃声设置**



**6-17 风扇转动**

**七、会议记录**

（1）第一次会议：

会议上分配任务，大家一起熟悉了项目总体要求，根据老师要求确定项目的方向。大家分别注册Github账号，将每个人加入项目组，然后由组长带领教大家使用GitHub的使用，同时对每个人进行了分工安排，介绍了代码大致结构以及一些需要涉及到的硬件知识，相关电路模块及元器件资料收集。给每个人安排了第一次硬件实验课需要准备的内容，也对此进行了明确每个人负责的部分。

1. 第二次会议：

进行了线上会议根据老师给的文件测试硬件电路相关各芯片、模块工作情况并进行调试修正，确定项目开发标准，对负责绘制原理图、流程图、UML图进行了指导最终确定了项目的整体，查看大家通过查找参考资料以及网络一些教学视频完成自己所负责的部分，确认了每人的进度，对工作安排做出一些调整。检查大家在git上传的分支，大家在会议上进行调试后作出修改并相互指正一些错误。

1. 第三次会议：

大家基本完成自己所负责的模块，每个人对自己负责的部分进行了阐述同时对已完成的部分反复调试和运行，负责硬件的同学展示了成品效果，负责软件部分的同学对重要部分代码块进行了讲解和演示，负责流程图的同学和原理图的同学将效果图分享在群里分享自己对本次温度测控系统硬件实践的见解，负责实验报告部分的同学对最后的流程和内容进行综合整理并排版。

**八、小结**

**李宗霖：**

我熟练掌握了lcd1602、ds18b20、at24c02、以及iic协议。我对lcd1602、iic信号模拟、24c02进行了完全封装，实现了其所有功能，以及更复杂的模块组合。对ds18b20不完全封装，实现了单个器件的所有操作。并根据我封装的模块组合成了这个大型项目。其中值得注意的是ds18b20自带由3字节的EEPROM可以不依赖24c02实现温度上下限和分辨的存储，并且上电后会自动回归到RAM当中，非常方便。此外关于读写的操作方式。这个过程需要精确控制信号的时序，否则可能会导致数据错误或设备无法识别。基于iic总线协议的24c02同样需要精确的控制信号时序。最开始我理解的页写入时内存分为了很多页，一次只能再某一页当中写入，便按照这个思路实现了多页写入的封装。但是网络上非常多的人的代码或者时思路是一次最多写入8字节，而不在乎起止地址，我也曾一度认为如此，经过实践的检验和查询数据手册最终确定，我最初的想法没错。我设计的模块当中比较巧妙的当属中断的整套系统。Ds18b20、直流电机方波、计时器都需要计时器实现稳定的计时，所以采用方式2；而蜂鸣器想要发出音乐则需要计时器实现不同周期的计时来实现不同音高 ，因此采用方式1。于是我便设计了T0计时器发出稳定的方波，通过count实现对不同周期的需求。但是后来发现ds18b20读取的过程不可被中断打断，因为它对时序深度依赖。于是我计算出了关中断期间的机器周期，并在重新开中断后给补上，以此来实现精确定时，定时器误差大约在5us内。非常需要注意的，方式二的计数最大256机器周期，因此中断函数应该少于250机器周期，所以需要通过bit类型的标志来将复杂操作放在主函数当中。在最后我发现了一个矛盾的设计，想要是精准定时，那么T0得是唯一高优先级中断。但是蜂鸣器对时序要求及其严格，否则声音会不对，因此也得是唯一高优先级中断。我想到了解决办法：当需要启动蜂鸣器的时候，拉低T0计数器的优先级，T1为唯一高优先级中断，以此来保证蜂鸣器的音准，并且在退出T1中断前检测TF是否为1，是则执行一遍T0中断的内容，因为T1周期也少于256，因此只可能出现一次中断不响应，这种方式既保证了蜂鸣器音调准确，又保证了计时器的准确性，以及可以产生稳定的方波。

**万鹰正：**

在这次实践中我针对蜂鸣器和24c02展开了针对性的学习，蜂鸣器是一种将电信号转换为声音信号的器件，常用来产生设备的按键音、报警音等提示信号。蜂鸣器按驱动方式可分为有源蜂鸣器和无源蜂鸣器，有源蜂鸣器：内部自带振荡源，将正负极接上直流电压即可持续发声，频率固定，无源蜂鸣器：内部不带振荡源，需要控制器提供振荡脉冲才可发声，调整提供振荡脉冲的频率，可发出不同频率的声音.，有源蜂鸣器内部有震荡源，所以只要一通电就会叫，频率固定，声音单调。而无源蜂鸣器内部不带震荡源，所以用直流信号无法令其鸣叫，只能通过不断地接通、断开它来产生不同频率的声音。无源的可以通过频率控制做出”多来米发索拉西“的效果。已知无源蜂鸣器通过高低电频的变化可以产生不同的声调，则知道每个音符的频率，通过计时器控制高低电平，则就可以发出不同声调的声音。查阅资料后我发现一个曲子简单的是由音调和节拍决定的，音调是1,2,3,4,5,6,7。节拍有4分音符，8分音符，也就是音调持续的时间长短。不同音调有不同的频率，只要精确的将频率的信号输入到定时器，然后通过翻转蜂鸣器的电信号就行，查阅资料我找到了不同音调对应的频率，然后加上延迟来模拟节拍和停顿。最后就是把乐谱翻译一下就可以了。之后分别Proteus和电路板上进行测试，发现Proteus里声音清脆，音调起伏，而电路板上声音生硬，音调单一，后来上网查阅发现，在我们电路板上的蜂鸣器是有源蜂鸣器，而在Proteus上的仿真的蜂鸣器是无源的。电路板上的蜂鸣器无法发出高低起伏的音调，但是仍然可以通过高低变化的电平，控制蜂鸣器发出声音和关闭声音来模拟类似的歌声，但仍然无法与无源蜂鸣器相比。然后就是把歌声存到24c02里，刚开始对24c02的学习不足，导致写入失败，后来求助李宗霖后，发现24c02是分页式存储，而且存入和读取是有时延的。后来在李宗霖的帮助下，完成了对歌曲的存入和读取。

**杨艺铭：**

在本次实践中，我负责的是代码流程图绘制的工作，具体负责绘制了主函数流程图，DS18B20模块和LCD1602模块的流程图。通过这次实践课程与小组合作，加深了对DS18B20的使用功能的了解和应用，也对LCD1602的一些基本功能的应用和代码过程更加清晰，甚至于之前模糊了解的中断调用代码的撰写也通过这次理解别人的代码彻底搞清楚了。通过这次的小组合作，我也学习到了主程序与其他程序之间封装连接的技巧，以及强化了单片机应用广泛功能强大的认识，甚至还学习到了之前忽略到的标志位的设定，按键触发方式的具体实践应用，电机如何工作，电机档位的设定方式，如何根据分辨率调整周期时间等等。总之，通过这次实践课程，收获颇多，也意识到了自己在单片机学习上只是设计皮毛，和其他同学之间的知识储备，代码编写以及逻辑应用上还存在着很大的差距。在流程图初稿完成后，组长也指出了一些问题，让我加深了对程序应用设置上的理解，从而更好地改进我们程序按键系统和视图系统分别对应的功能，更好地完成对流程图的撰写。在小组合作过程中，小组的代码是不断更新改进的，组长也教会了我们应用Github，更加直观的看到了每个组员对这个代码的修改。经历了这次的实践课程，我认识到我应该在课本基础上，多参考了解别人的基于应用层面的代码，真正的将知识投入到实践应用之中去。另外，也应该不止局限于书中所提到的硬件，多了解一些其他常用的硬件设施的应用以及基础功能和代码的编写。最后，我真的很感谢这次小组实践课程，让我得到了一个检验自己学习成果的机会，并且对自己的学习成果有了更准确的评价。

**周妍妍：**

在本次实践中，我负责的是代码流程图绘制的工作，具体绘制了系统使用流程图，按键系统流程图，开机动画流程图，不同界面显示流程图等具体功能具体实现的流程图以及各功能模块的依赖图。通过本次实践课程与小组合作，我收获颇多，不仅对硬件集成开发有了一定了解，增强了我对不同器件和对系统设计开发的理解，自主学习了未使用过的器件的使用方法。通过阅读理解他人的代码，我也学习到了一些编写代码的技巧和经验，例如封装和整合代码的技巧，一些优化用户体验的设计小巧思。同时我还学习到了托管平台的使用方法，极大方便了我们小组合作完成本次实践任务，也了解到清晰分工和及时的沟通对提高效率的重要性。因为负责的工作内容是基于他人已编写封装好的代码，所以需要细致的交流保证我对代码要求和目的理解是准确无误的。总之，通过编写流程图，加深了我对系统设计和逻辑的理解，了解到各模块间的框架结构、逻辑关系、具体功能的具体实现方法，以及在设计过程中需要注意的很多可以优化用户体验的小细节。与此同时我也发现一些自己的不足之处，例如对一些器件的理解还是仅限于书本，比较浅显，模块使用熟练度还不够高，这一点会在我完成任务时降低我的工作效率。但在自主学习后，这一点便迎刃而解，补足知识漏洞后，对各功能部件之间的工作逻辑便有了深刻理解。我认为本次实践学习是一个很好的锻炼自我和验收学习成果的机会，而在此后，我还需要积累更多的相关知识和开发经验，提高自己的编程能力，多实践，尽早让自己可以投入到底层开发中去。

**刘媛：**

宝贵而充实的硬件实践经历使我受益良多，在基于51单片机的温度检测、显示及控制系统的实践学习中，我承担了代码注释、绘制原理图和编写实践报告的任务。通过这次实践，我对嵌入式系统的开发流程和相关技术有了更深入的理解，同时也提升了自己的团队合作和沟通能力。首先，在代码注释方面，我仔细阅读了整个代码，并在每个函数的开头添加了详细的注释。注释中包含了函数的功能描述、输入参数和返回值说明，以及关键变量的用途和取值范围。通过注释，其他团队成员可以更快地理解代码的逻辑和实现细节，提高了团队协作效率。其次，在原理图的绘制过程中，我按照硬件设计的要求，使用专业的软件进行了绘制。我将各个模块的连接方式清晰地表达出来，并标明了元器件的型号和参数。同时，我还在原理图上添加了连线的注释，以便其他人在理解原理图时能够更加清晰地理解电路的功能。最后，在撰写实践报告的过程中，我详细记录了整个实践的步骤和结果。我首先介绍了项目的背景和目标，然后详细描述了硬件设计、代码编写和系统调试的过程。在实验结果分析中，我列举了温度检测和控制的准确性和稳定性，并对系统的优化方向提出了一些建议。同时，我还总结了自己在实践中遇到的问题和解决方法，以及团队协作中的心得体会。通过这次实践学习，我不仅加深了对51单片机嵌入式系统的理解，还提升了自己的技术能力和团队合作能力。通过代码注释，我让其他团队成员更加容易理解和维护代码；通过原理图绘制，我清晰地呈现了硬件连接和电路功能；通过实践报告，我详细记录了整个实践过程和结果，为未来的项目积累了宝贵经验。这次实践对我的个人成长和职业发展具有重要意义，我将继续努力学习，提高自己的技术水平和团队合作能力。

**陈国孜：**

通过这次实践，我深刻体会到了实际动手操作的重要性和技术能力的提升。在这个过程中，我不仅了解了单片机编程和传感器应用，还培养了解决问题的能力和团队合作精神。我个人对单片机的学习比较表面从来没有真正的了解过深层次的内容，通过这次课程实践，我对单片机编程有了更深入的理解。在实践中，我们需要根据实际需求设计并搭建各种电路，包括蜂鸣器控制电路、LCD显示电路等。通过实践，我学会了如何正确连接电路、调试代码、进行信号采集和处理等技术。我们需要编写程序来读取温度传感器的数值，并进行相应的控制反馈。通过与传感器的交互，我学会了如何利用单片机的输入输出端口进行数字信号的处理和转换。这让我对编程语言和硬件的结合有了更清晰的认识，也增强了我的代码能力和逻辑思维能力。最后我负责实验报告的总结和归纳。通过写实验报告，我深刻体会到了实验思路的清晰性、数据分析的准确性以及表达能力的重要性。在写实验报告的过程中，我需要将实验的目的、原理、步骤和结果等细节进行逐一梳理和整理。这要求我对实验过程进行深入的思考和理解，使得自己对实验内容有了更加清晰的认识。同时，我也学会了如何用简明扼要的语言将实验过程描述出来，使读者能够直观地理解实验的目的和方法。也考验了我的表达能力和逻辑思维能力，使自己的观点更加明确和有说服力。其次，在实践中，我意识到了团队合作的重要性。在实验室里，每个人都扮演着不同的角色，有人负责硬件搭建，有人负责程序编写，还有人负责数据分析。只有团队成员通力合作，才能顺利完成整个系统的搭建和调试。因此，我也学会了倾听他人意见、有效沟通和协作的重要性，这些都是我在未来工作中必不可少的能力。最后，通过这次课程实践，我对温度测控系统的设计和应用有了更深入的了解。温度测控系统在生活和工业领域有着广泛的应用，这次实践我掌握了温度传感器的使用方法，还了解了一些控制算法等控制原理，这将为我未来的学习实践打下坚实的基础。

**附录：参考文献**

[1]张扬，王聪，张雷.基于单片机C8051F500的无刷直流电机控制研究[J].微处理机，2020,41(03):43-47.

[2]崔浩斌，刘伟.基于单片机控制的语音采集与回放系统设计研究[J].微处理机，2020,41(03):51-54.

[3]何流，谭文韬，张鹏琴，余建想.基于温度传感器的智能恒温空调系统设计[J].计算机产品与流通，2020(06):144.

[4]王殿臣.基于STC单片机的直流电机调速控制方法研究[J].计算机产品与流通，2020(07):62.

[5]刘立钧，张永达.单片机按键程序研究[J].电子世界,2020(08):13-14.

[6]盛定仪，高渤濡，王向进，李哲，李瑛.基于PIC单片机的固体激光器温控系统设计[J].内江科技，2020,41(04):38+111.

[7]肖骁，戈文祺.电器传动系统中单片机技术的应用解析[J].中国标准化,2117(22):250-252.

[8]于洋.温度报警系统的整体设计[J].黑龙江科技信息,2009(2):12-12.

[9]张毅刚，彭喜元，彭宇.单片机原理及应用[M].北京:高等教育出版社,2010.

[10]万胜前，基于keilC51软件的电子时钟设计与制作[J].鄂州大学学报,2007(2):23-29.

[11]戴佳，戴卫恒.51单片机基于C语言程序设计实例精讲[M].北京:电子工业出版社,2010.

[12]宋述林.物联网电子产品中单片机技术的应用方式研究[J].现代工业经济和信息化,2017,7(22):64-65+75.