



安徽信息工程学院
Anhui Institute of Information Technology

2020 届本科生毕业设计（论文）

题 目 _____ 恒温烤箱控制系统 _____
学 生 姓 名 _____ 谷创业 _____
学 号 _____ 316207010104 _____
专 业 班 级 _____ 自动化 1601 _____
学 院 _____ 安徽信息工程学院 _____
指 导 老 师 _____ 张松林 _____
完 成 日 期 _____ 2020 年 6 月 5 日 _____

教务处 制

摘 要

随着现代电子技术水平的迅速发展，单片机技术和自动控制技术的广泛应用，给我们的生活带来了巨大的改变。单片机技术的应用使得智能家电类产品体积更加小巧、工作更稳定、功能实现更多样。温度控制在工业生产过程中一直以来都是一个重要的目标之一，有些生产工艺对温度的控制要求极高。在生产的过程中如果未能对温度施行准确的控制，生产出来的产品质量可能不达标，所以设计一款温控精准的控制是具有重要意义的。

本项目将设计一款恒温烤箱控制系统，该控制系统能对温度进行准确的恒温控制。相对于传统温度控制系统，本系统采用 PID 控制技术对温度进行控制，温度将更快接近目标值，达到目标值后温度能保持恒定，具有一定抗干扰性能。

本论文将分为五个章节来叙述恒温烤箱控制系统：概述、PID 控制算法的设计、硬件层的设计、软件架构的设计和仿真验证。首先介绍恒温烤箱控制系统的背景、意义与研究内容。该系统将以 PID 控制算法的设计为着重点，叙述控制算法的对比、选择以及实现。在系统的硬件设计方面所使用模块化设计方法，将系统的硬件拆分为不同功能模块，各模块即独立又相互配合。软件采用低内聚、高耦合的设计方法，不同的软件模块与相应硬件相对应，同时软件由将所有模块组合到一起，共同组成恒温烤箱控制系统。最后在仿真软件上实现验证系统设计的正确性，以及是否能到设计目标。

关键词：PID； 恒温控制系统； 烤箱

ABSTRACT

With the rapid development of modern electronic technology, the wide application of single-chip technology and automatic control technology has brought tremendous changes to our lives. The application of the single-chip technology makes the smart home appliances more compact, more stable in work, and realize more diverse functions. Temperature control has always been one of the important goals in the industrial production process, and some production processes have extremely high requirements for temperature control. If the temperature is not accurately controlled during the production process, the quality of the produced product may not meet the standards, so it is of great significance to design a temperature-controlled and accurate controller.

This project will design a constant temperature oven control system, which can accurately control the temperature. Compared with the traditional temperature control system, this system uses PID control technology to control the temperature, the temperature will be closer to the target value faster, after reaching the target value, the temperature can be kept constant, with certain anti-interference performance.

This paper will be divided into five chapters to describe the constant temperature oven control system: overview, PID control algorithm design, hardware layer design, software architecture design and simulation verification. First introduce the background, significance and research content of the constant temperature oven control system. The system will focus on the design of the PID control algorithm, describing the comparison, selection and implementation of the control algorithm. The modular design method used in the hardware design of the system splits the hardware of the system into different functional modules, and each module is independent and cooperates with each other. The software adopts the design method of low cohesion and high coupling. Different software modules correspond to the corresponding hardware. At the same time, the software combines all the modules together to form a constant temperature oven control system. Finally, the correctness of the system design is verified on the simulation software and whether it can reach the design goal.

Key words: PID; thermostatic control system; oven

目 录

插图清单	V
表格清单	VI
第一章 概述	- 2 -
1.1 研究的背景	- 2 -
1.2 研究的意义	- 2 -
1.3 研究的内容	- 2 -
第二章 PID 算法设计	- 4 -
2.1 PID 介绍	- 4 -
2.2 位置式 PID	- 5 -
2.3 增量式 PID	- 7 -
第三章 硬件设计	- 9 -
3.1 系统总体设计	- 9 -
3.2 控制器最小系统设计	- 9 -
3.3 温度检测	- 14 -
3.4 温度控制模块	- 16 -
3.5 显示模块	- 17 -
3.6 按键输入模块	- 18 -
3.7 声光报警模块	- 19 -
第四章 软件设计	- 21 -
4.1 主程序设计	- 21 -
4.2 PID 算法程序设计	- 22 -
4.3 温度检测程序设计	- 23 -
4.4 温度控制程序设计	- 24 -
4.5 按键程序设计	- 25 -
4.6 LCD 显示程序设计	- 26 -
4.7 声光报警程序设计	- 27 -
第五章 系统仿真与实验调试	- 28 -

5.1 仿真设计	- 28 -
5.2 参数整定	- 29 -
5.3 仿真结果	- 30 -
结论与展望	- 32 -
致 谢	- 33 -
参考文献	- 34 -
附录 A 硬件原理图	- 35 -
附录 B PCB	- 36 -
附录 C 程序	- 37 -
附录 D 主要参考文献的题录及摘要	- 42 -

插图清单

图 2-1 PID 控制原理框图	- 4 -
图 3-1 系统总体方案框图.....	- 9 -
图 3-2 STM32F103C8T6 引脚图	- 10 -
图 3-3 电源电路.....	- 10 -
图 3-4 时钟电路.....	- 11 -
图 3-5 启动电路.....	- 12 -
图 3-6 CH340G 电路	- 12 -
图 3-7 SWD 电路.....	- 13 -
图 3-8 复位电路.....	- 14 -
图 3-9 MAX6675 引脚图	- 15 -
图 3-10 数据格式.....	- 15 -
图 3-11 温度检测电路.....	- 16 -
图 3-12 温度控制电路.....	- 17 -
图 3-13 LCD1602 电路图.....	- 18 -
图 3-14 按键电路图.....	- 19 -
图 3-15 声光报警电路图.....	- 20 -
图 4-1 主程序流程图.....	- 22 -
图 4-2 增量式 PID 控制算法程序流程图	- 23 -
图 4-3 温度检测流程图.....	- 24 -
图 4-4 温度控制流程图.....	- 25 -
图 4-5 按键流程图.....	- 26 -
图 4-6 显示程序流程图.....	- 27 -
图 4-7 声光报警程序流程图.....	- 27 -
图 5-1 PROTUES 仿真图.....	- 29 -
图 5-2 50℃温度-时间变化折线图	- 30 -
图 5-3 60℃温度-时间变化折线图	- 31 -
图 5-4 70℃温度-时间变化折线图	- 31 -

表格清单

表 3-1 启动模式.....	- 11 -
表 3-2 LCD1602 引脚功能.....	- 17 -

恒温烤箱控制系统设计

（电气与电子工程学院 20 届自动化 1601 班） 指导教师：张松林

引言

随着社会的发展，烤箱进入了越来越多的家庭中，人们可以使用烤箱加热食物做出美味的熟食。本设计恒温烤箱控制系统主要的工作内容是对温度进行检测和控制，对温度控制的准确性将直接表明了恒温烤箱控制系统的性能。温度具有很强的滞后性，想要完成对温度的准确控制需要采取一套表现优异的温控算法，这对系统的工作性能表现和能源节约都具有现实意义。本设计采用 STM32 微控制器作为恒温烤箱控制系统的控制核心，使用单片机作为系统控制的核心，有着操作简单、工作速度快、电量消耗小、体积小等方面的优点，很适合应用在类似烤箱这类家用电器中。

在恒温烤箱控制系统的设计过程中，系统的控制算法是本设计中最为重要的部分，因此对于自动化专业的学生来说，学习并掌握自动控制方面的相关知识是特别重要的。在本次的设计中，采用增量式 PID 作为恒温烤箱控制系统的温控算法，此算法可以将烤箱的温度恒定保持在设定温度允许误差范围之内。本次的毕业设计要求我完成产品开发的整套工作,这对我来说是一次很好的锻炼机会。

本次设计的主要内容有：

- （1）设计系统控制算法；
- （2）设计系统硬件；
- （3）设计系统软件；
- （4）设计系统仿真；

在本次的毕业设计中，需要仔细分析选择不同的电子器件、设计各模块的功能，并绘制具体的硬件电路图，在这个过程中大大提升了个人的设计能力。

第一章 概述

1.1 研究的背景

现在人们对生活品质的追求越来越高，恒温烤箱已成为现代家庭中经常使用的电器。使用恒温烤箱制作各种食物，需要恒温烤箱对温度的控制比较准确，否则就无法制作出美味的食物。这也就需要烤箱产品的开发人员能够开发出温控性能较好的温控系统，使烤箱对箱内温度控制更加准确。

21 世纪科学技术高速的发展，电力电子技术和单片机的应用空前广泛，自动控制技术越来越成熟，带动了相关生产工艺的取得了巨大的进步。虽然在近年来，温度检测和控制方面的理论基础发展的已经很成熟了，但是在实际应用中，怎样能保证快速地、实时地对现场温度进行精准采样，又要确保温度数据在传输过程中不会发生改变，并且能根据所测量到的实时温度数据对现场温度进行比较精确的控制，仍然是当前需要解决的问题。

1.2 研究的意义

在我们的日常生活中，经常能接触到的一个名词就是温度。相应地，温度的测量与控制也常被人们在生活和生产中的所提起，它在各行各业中都发挥着非常重要的作用。温度是一个基本的物理名词并且很常见的，它所表示的是周围物体的冷度与热度。在生活中，任何物理、化学过程和都与温度有着密切关联的，这些物理、化学过程又与生活和生产相关。

在生产的过程中，需要对生产中各个环节的温度进行精准的检测与控制，对温度测量与控制的准确性，与生产的产品质量、安全性、生产效率和技术节能指标相关联，温度的测量和控制已在各个领域引起了相当大的关注。

随着智能化、自动化时代的到来，现在都追求设备操作简单、无需人过多的干预，设备就能按照人们的预期进行运行。温度的测量与控制如果是由人工进行操作，那么温度的恒定控制很难达到要求。智能化和自动化的温度控制系统的研究设计，将对节约人力、降低消耗方面会有很大的帮助。

因此，对于温度控制系统的设计与研究具有极其现实的重要意义。

1.3 研究的内容

本课题将设计一款适用于恒温烤箱的温控系统，本系统设计的重点是恒温控制，它需要可以适应环境温度的变化并能精确调节、保持温度。该控制系统以增量式 PID 为控制器的控制算法，此算法原理非常简单并且有着非常优秀的控制性能。在温度控制领域，使用具有 PID 算法的温度控制器来控制温度，具有结构简易、成本低以及高精度等优点，可以达到精准测温控温的要求。同时配合温度的数据采集单元和控制执行单元，共同完成对温度的恒定控制。

第二章 PID 算法设计

2.1 PID 介绍

PID 控制系统是被控对象与 PID 控制器共同构成的，控制器的 PID 控制算法是由三个部分组成：比例、积分、微分部分，它的工作原理框图如图 2-1 所示。比例部分的作用是系统存在偏差可以立即产生调节效果，迅速减小被控对象的偏差。引入积分的部分作用消除系统存在的稳态误差，使被控对象可以达到目标值。通过引入微分部分可以预测到系统偏差变化趋势并对偏差变化迅速做出响应，可以解决控制系统的响应滞后的问题，从而对发生偏差变化时，系统的响应速度有所提高。PID 算法不仅考虑控制对象的当前状态数值，而且还考虑控制对象最近一段时间的状态数值变化与过去一段时间的状态数值，从三个方面共同确定当前输出的控制信号。仅需对比例部分系数、积分部分系数、微分部分系数三个参数的数值通过调整，大多数控制系统的闭环控制就能够获得较好的效果。将比例部分、积分部分、微分部分的运算控制结果进行累加求和，得出最终控制量 $u(t)$ ，使用这个控制量对被控制对象进行控制，能够实现良好的控制效果。采用此算法可极大改善恒温烤箱的控制效果，就像有经验的控制领域专家手动操作一样。

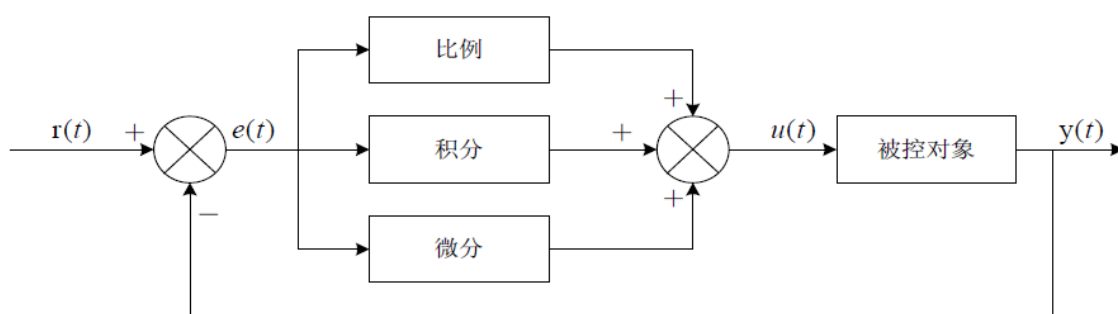


图 2-1 PID 控制原理框图

一般来说 PID 控制具有以下优点：

1. 控制算法简单、使用方便，它很容易以各种方式实现。非常合适对机械、流体、电子等执行机构的控制，方便各种工业控制应用，因此 PID 控制算法常被各种标准的控制装置和模块的所采用。

2. PID 控制参数相对较少，其参数的整定易完成。各参数的控制效果明显且相互干扰少，这使 PID 控制参数的整定更加方便，其参数整定有一套固定的方法，按照整定步骤进行整定，很容易就能整定出合适的 PID 参数。

3. PID 控制的鲁棒性非常强，此算法具有很强的适应能力，在被控对象工作环境发生变化或者出现干扰时，它也能适应这种变化。该系统是一个带有闭环反馈的控制系统，不用像前馈控制那样为每一个干扰扰动都设计出单独的控制措施，这样还能简化系统结构。被控对象自身的时变性和非线性属性也都对控制的结果影响相对比较小，系统控制质量对被控对象的特性变化不太敏感。

4. PID 控制算法具有简单的智能控制思想，PID 控制中的比例控制根据当前值与设定值偏差的立即产生控制作用，会对偏差进行消减；积分控制是基于目标值与当前值偏差的不断连续积累，积分部分可以将偏差变化较为缓慢的、偏差幅度较小的，但是偏差仍然存在没有消退的偏差给消除掉；当被控对象有偏差变化时微分部分控制便能感知到，并可以提前对未来的偏差做出调整，有一定的预见性。

基于这些优点，PID 反馈控制系统被广泛用于各个行业的自动控制系统中。现在大型现代化的生产工厂控制回路都多达一百到两百条，这其中大多数的控制都是使用 PID 控制算法来进行控制的。仅当被控制使用 PID 控制无法满足生产的要求，才去考虑使用更高级的控制方法来代替 PID 控制。

PID 控制算法是通过原先的设定数值和测量到的被控对象的当前状态数值进行相减，进而获得到系统当前存在的偏差数值 $e(t)$ ，这样通过不断的比较来实现对偏差值的比例、积分和微分部分的不断调整，再将调整后得到的输出值 $u(t)$ 作用于被控对象，经过一段时间持续的采集和控制，使其能稳定保持在输入值允许误差范围内，PID 控制算法的连续形式如下（式 2-1）所示：

$$u(t) = Kp \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (\text{式 2-1})$$

其中： Kp ——PID 的比例系数。

T_i ——PID 的系数，也称积分时间。

T_d ——PID 的微分系数，也称微分时间。

PID 控制算法分为两种，分别是位置式 PID 和增量式 PID，这两种算法只是在形式上有所不同，但是它们的控制思想和控制效果都是相同的。

2.2 位置式 PID

模拟控制不仅可以对被控对象的反馈进行连续的采样，而且还能对控制对象进行连续的控制。计算机控制器无法做到对被控对象的反馈进行连续采样，它都是以间隔采样的方式对反馈进行采样，只能根据控制器在采样时刻被控对象的目标值和当前值的差

值，然后经过控制器的运算获得合适的输出控制量，无法做到连续采样和控制。由于计算机控制器的这个特性，连续 PID 控制公式（式 2-1）中存在连续的积分项、微分项，因此不能直接在计算机控制器中应用。需要将连续 PID 控制公式（式 2-1）进行近似转换，转换为离散公式，才能在计算机控制器中应用。具体的转换过程为，将 k 表示计算机采样的序数，将 T 表示计算机控制器采样的周期时间，于是 kT 就为采样的时间，用矩形法数值积分来替换积分，用一阶向后差分来替换微分，这样便能完成离散化的转换，参与转换的公式如下式（式 2-2）所示：

$$\left\{ \begin{array}{l} t \approx kT (k = 0, 1, 2, 3 \dots) \\ \int_0^t e(t) dt \approx T \sum_{j=0}^k e_j \\ \frac{de(t)}{dt} \approx \frac{e(kT) - e[(k-1)T]}{T} = \frac{e_k - e_{k-1}}{T} \end{array} \right\} \quad (\text{式 2-2})$$

然后把（式 2-2）相关转换公式代入（式 2-1）中，经过整理便可以得到离散化形式的 PID 公式，表达式如（式 2-3）或（式 2-4）所示。

$$u_k = k_p \left[e_k + \frac{T}{T_i} \sum_{j=0}^k e_j + T_d \frac{e_k - e_{k-1}}{T} \right] \quad (\text{式 2-3})$$

或

$$u_k = k_p * e_k + k_i \sum_{j=0}^k e_j + k_d (e_k - e_{k-1}) \quad (\text{式 2-4})$$

其中： k ——采样序号， $k=0, 1, 2, \dots$ ；

u_k ——第 k 次采样时刻控制输出量；

e_k ——第 k 次采样时刻偏差；

e_{k-1} ——第 $k-1$ 次采样时刻偏差；

k_p ——比例系数；

k_i ——积分系数， $k_i = k_p * \frac{T}{T_i}$ ；

k_d ——微分系数， $k_d = k_p * \frac{T_d}{T}$ ；

如果离散系统采样的周期足够小，那么（式 2-3）或（式 2-4）的计算结果就已经非常的接近连续系统的计算结果，并且离散控制系统的控制过程也是非常接近连续控制系统的控制过程。

PID 公式（式 2-3）或（式 2-4）控制算法的运算结果给出了所有控制量的大小，所以此 PID 算法被称为位置式 PID。这种算法的缺点是：因为它的输出控制量都是全量输

出的，所以每个输出的控制量都与系统过去的工作状态相关，目标值与当前值的偏差 e_k 在计算的过程中不停的积累，系统的运行时间越长，控制器的计算量将会越来越大。并且由于控制器输出的控制量 u_k 与被控对象的工作量相同，在控制器出现故障或者运算数值已经超出了计算机的运算范围，则输出 u_k 将会发生很大变化，这将导致执行器工作状态发生较大变化，从而可能导致严重的生产事故，在实际生产中绝对要避免出现的。

2.3 增量式 PID

控制系统使用增量式 PID 进行系统的控制时，它每一次输出的控制数值仅是前一个输出控制数值的增加量 $\Delta u_{(k)}$ 。由于运算后得到输出的控制量 $\Delta u_{(k)}$ 对应于本次被控对象控制量的增量，并不是被控对象的实际控制量。因此，执行器的控制算法中还需增加控制量累加的功能，将增量值与上次实际控制量相加得到的实际控制量便是本次被控制的实际控制量。

采用增量式 PID 的控制系统只需要知道本次偏差、上次偏差、上上次偏差数据就能计算出控制增量，不需要计算大量的历史偏差累积数据，计算工作相对比较简单。控制器输出是每个实际控制量的增加量，防止了实际控制量误差的累加，它的控制效果要好于使用位置式 PID 控制的控制效果。

增量式 PID 公式可以从（式 2-2）中进行推导出来，第 k-1 采样时刻（式 2-2）的控制输出值：

$$u_{k-1} = k_p \left[e_{k-1} + \frac{T}{T_i} \sum_{j=0}^{k-1} e_j + T_d \frac{e_{k-1} - e_{k-2}}{T} \right] \quad (\text{式 2-5})$$

将公式（式 2-2）和公式（式 2-5）两式相减，再对相减后的数值进行整理便得到增量式 PID 的表达式，具体的运算过程如下所示：

$$\begin{aligned} \Delta u_k &= u_k - u_{k-1} \\ &= k_p \left(e_k - e_{k-1} + \frac{T}{T_i} e_k + T_d \frac{e_k - 2e_{k-1} + e_{k-2}}{T} \right) \\ &= K_p \left(1 + \frac{T}{T_i} + \frac{T_d}{T} \right) e_k - k_p \left(1 + \frac{2T_d}{T} \right) e_{k-1} + k_p \frac{T_d}{T} e_{k-2} \\ &= K1 * e_k - K2 * e_{k-1} + K3 * e_{k-2} \end{aligned}$$

$$\text{其中 } K1 = K_p \left(1 + \frac{T}{T_i} + \frac{T_d}{T} \right)$$

$$K2 = k_p \left(1 + \frac{2T_d}{T} \right)$$

$$K3=k_p * \frac{T_d}{T}$$

增量式 PID 有以下三点优点：1 在算法的运算过程中没有出现累加，输出增量 Δu_k 的数值仅和最后三个偏差数值相关，很容易通过加权获得更好的控制效果；2 控制器的输出仅是实际输出量的增量，即执行对象控制量的相应变化，当机器出现了故障时，所影响的范围比较小，不会产生影响到生产的严重事件；3 当控制器手动和自动相互切换到，可以进行无干扰切换影响较小。

从以上两种控制算法的对比中可以看出，增量式 PID 在温控方面是要比位置式 PID 更具优势的，所以本设计采用增量式 PID 作为恒温烤箱控制系统的控制算法。

第三章 硬件设计

3.1 系统总体设计

本硬件系统需要具备温度数据的采集和温度控制的功能，还需具备显示、设定温度和报警的功能。此系统主要以温度恒定控制为主，系统使用单片机作为控制的核心器件，与显示电路模块、温度的控制电路模块、声光报警电路模块、温度的检测电路模块、按键电路模块进行互相配合使用，共同组成恒温烤箱控制系统，完成对烤箱温度的准确测量与控制，系统总体方案框图如图 3-1 所示：

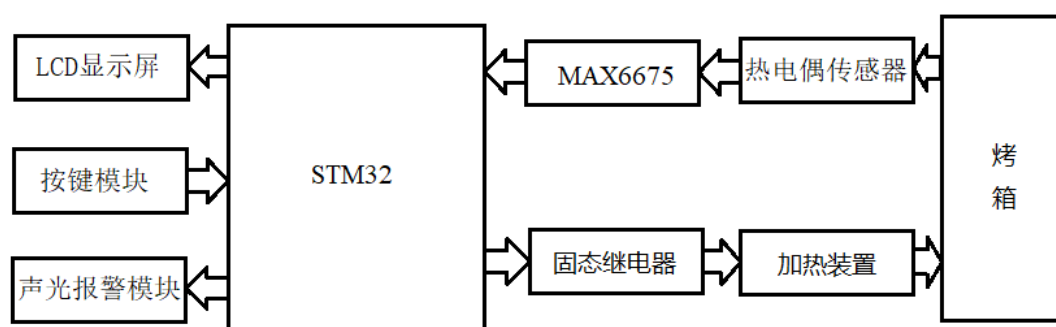


图 3-1 系统总体方案框图

系统运行时，通过 K 型热电偶传感器采集恒温烤箱内的实时温度，然后通过 MAX6675 芯片将采集到的温度数据传递给 STM32 微控制，STM32 微控制将读取到的数据通过 PID 算法对数据进行处理后，通过输出 PWM 控制波形来实现固态继电器的接通与关断，进而控制烤箱内的温度，同时在 LCD 显示屏上显示当前目标温度数值、设定温度数值和加热时间，通过按键模块可以改变设定温度与加热时间。在系统运行异常时，声光报警模块能够发出报警提示。

3.2 控制器最小系统设计

3.2.1 MCU

本设计的控制芯片采用 ARM 公司所设计的，32 位高性能 STM32C8T6 微控制器，该控制器使用的是 Cortex-M3 内核。工作电压为 2~3.6V，工作频率最高可以达到 72MHz。IO：其中有 37 个引脚为 IO 端口，存储：芯片内部自带 20K 字节的 SRAM 与 64K 字节的 FLASH，模拟端口：2 个 16 位的 ADC，通信接口：2 个 I2C、3 个 USART、2 个 SPI、1 个 USB 和 1 个 CAN，计时器：2 个高级定时器、3 个普通定时器、2 个基本计时器、2 个看门狗、2 个 RTC，STM32F103C8T6 封装引脚图如图 3.1 所示：

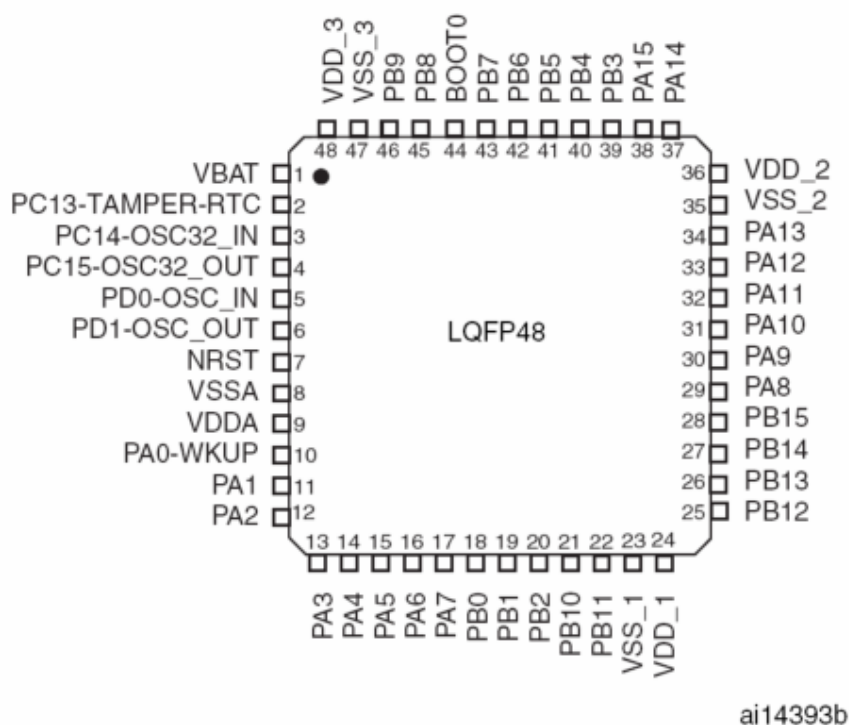


图 3-2 STM32F103C8T6 引脚图

3.2.2 电源电路

该款 MCU 采用 3.3V 电压供电，因此采用常见的由 Micro USB 提供 5V 电压，通过 EM6211 稳压芯片实现将 5V 电压降至 3.3V 为微控制芯片提供工作电压，5V 电压与降压后的 3.3V 同时也为其它器件提供所需的工作电压。Micro USB 接口的 D+、D-引脚配合 CH340 芯片，连接到 STM32C8T6 单片机的 USART 引脚，还能实现程序的下载。ME6211 稳压芯片可以输出 3.3V 电压，输出 100mA 电流，静态功耗只有 50uA，此芯片是降压需求的理想之选。

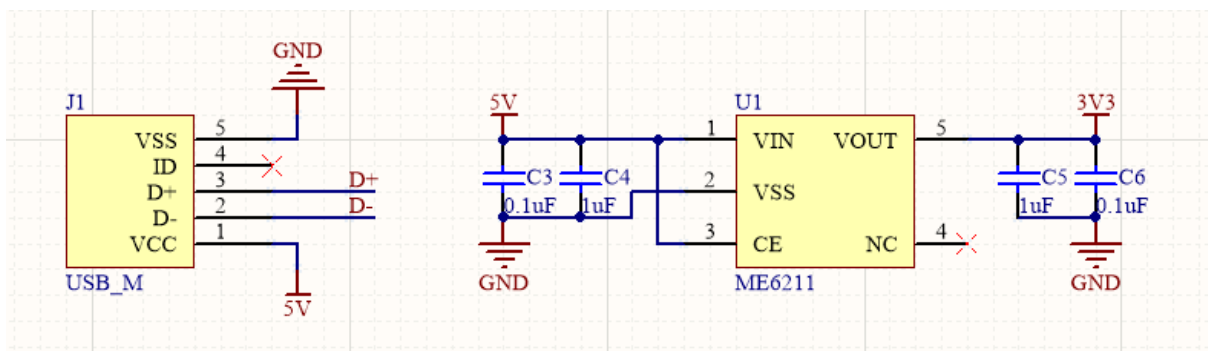


图 3-3 电源电路

3.2.3 时钟电路

STM32 微控制器内部自带两个晶振时钟，但容易不稳定，使用外置的时钟电路可为微控制器提供的时钟信号更为精准。时钟电路的设计采用由两个 22pF 的电容和一个 8MHz 晶振的组成，控制器能将此 8MHz 晶振倍频到 72MHz，供自身使用，时钟电路硬件设计电路图如下图 3-3 所示：

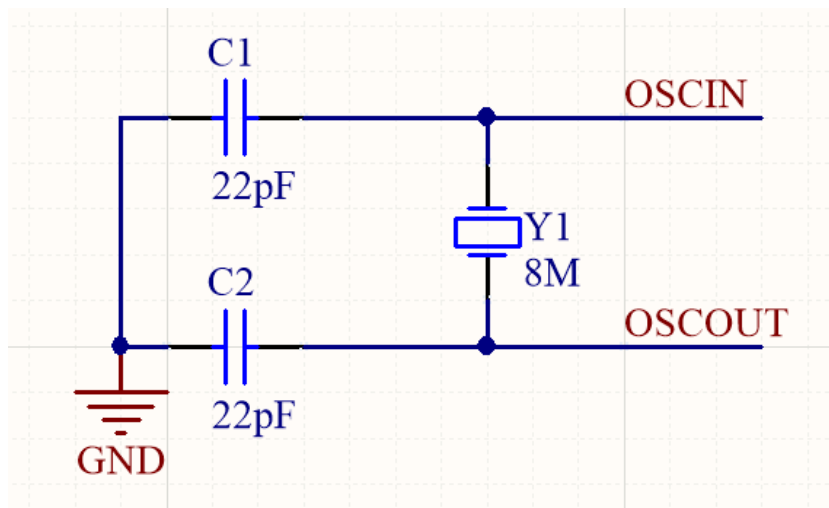


图 3-4 时钟电路

3.2.4 启动电路

STM32 微控制器定义的系统启动模式如表 3-1 中所示，它是由 STM32 微控制器的 BOOT0 和 BOOT1 引脚控制的，可以配置 BOOT0 和 BOOT1 引脚的高低电平，将微控制器设置为不同的启动模式。

表 3-1 启动模式

启动模式选择引脚		启动模式
BOOT1	BOOT0	
X	0	以主闪存存储器作为启动区域
0	1	以系统存储器作为启动区域
1	1	以内置 SRAM 作为启动区域

该方案提供了两种程序下载的方法：串行下载和 SWD 下载。在使用串口下载程序时，程序下载前先将启动引脚配置成 BOOT0=1，BOOT1=0，下载完成后再将 BOOT0 配置为 1，然后进行复位程序运行。在使用 SWD 下载调试程序时，直接将 BOOT0 配置为 0 就可以了。通过 BOOT 启动模式选择的电路被设计为 3×2 排针连接器，以实现启动

模式的选择。如图 3-4 所示。由于串口下载使用了 CH340 芯片，完成了程序下载、运行时 BOOT 引脚的自动配置，因此，只需将 3-5、4-6 排针引脚进行短接即可。

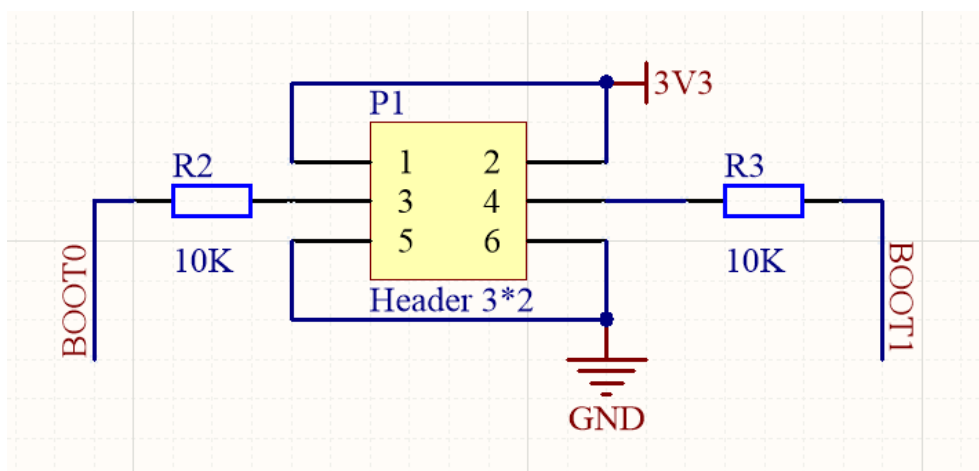


图 3-5 启动电路

3.2.5 下载调试电路

为了避免下载程序时还需手动切换 BOOT，过程比较繁琐，减少开发人员工作量。通过串口转 USB 芯片 CH340G 的 DTR# 与 RTS# 芯片引脚，可以实现使用串口进行程序下载时自动控制 BOOT 引脚的电平变化，完成程序的下载。通过 CH340G 芯片间接的控制 STM32 微控制器的 BOOT0、BOOT1 引脚电平变化，达到一键下载并运行程序的效果，CH340G 芯片的连接电路图如下图 3-6 所示。

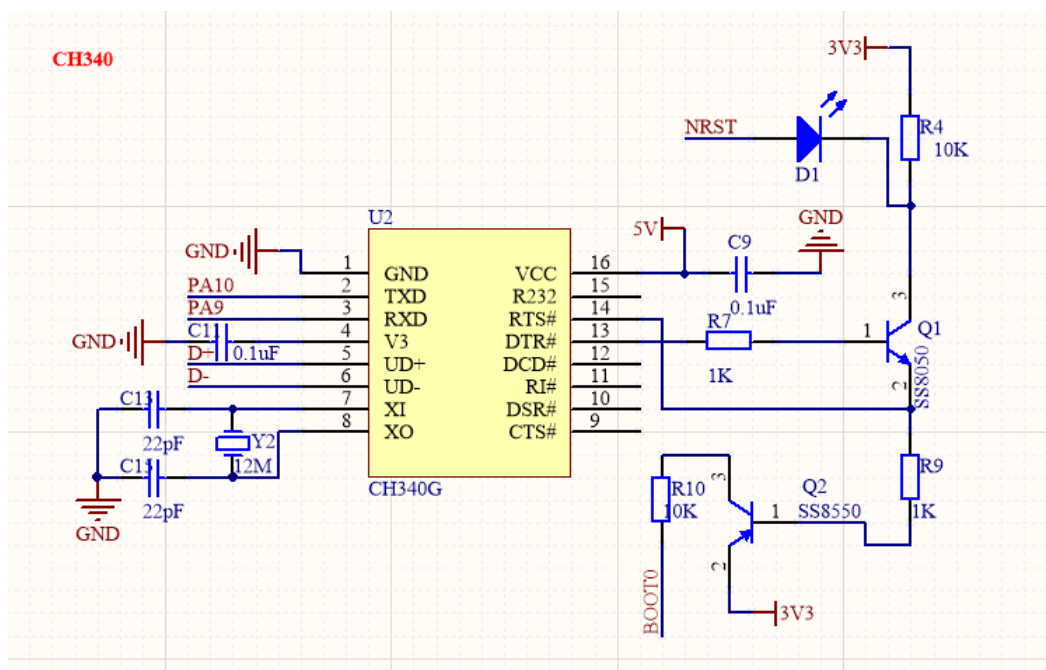


图 3-6 CH340G 电路

应用中，最常见的调试接口是 JTAG 与 SWD 接口，但是 JTAG 接口使用多达 20 个接口引脚，而 SWD 接口仅使用 4 个接口引脚：VCC、SWDCLK、SWDIO、GND，就可以实现。高速模式下使用 SWD 比 JTAG 更为稳定，下载程序的速度能够高达 10M/s，同样也可以进行下载与调试程序。本方案的下载电路使用 SWD 接口，其电路如图 3-7 所示。

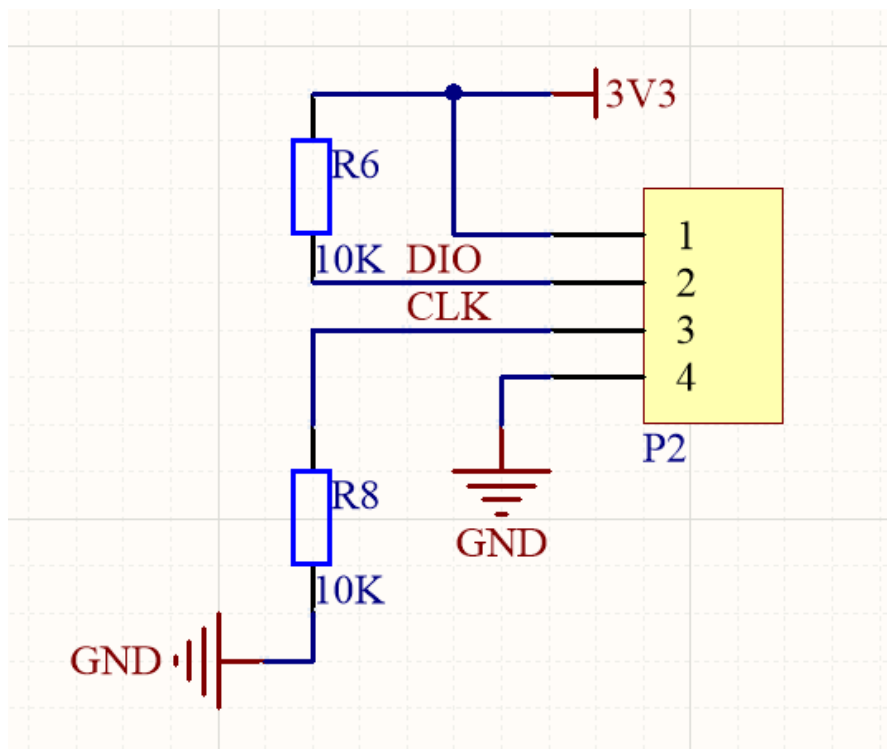


图 3-7 SWD 电路

3.2.6 复位电路

将控制器的 RESET 管脚保持 20us 的低电平状态，控制器的复位便可以完成。该复位电路方案设计可以实现上电复位与按键复位，其工作原理为上电时电容开始持续充电，电容充电到 3.3V 的时间为 0.1s，此电容充电过程中复位引脚缓慢的从低电平变为高电平，完成上电复位。在按下复位键时，同时电容开始放电，复位引脚变为低电平，抬起时充电，复位引脚又慢慢变为高电平，完成按键复位，其复位电路的硬件电路图如图 3-6 所示：

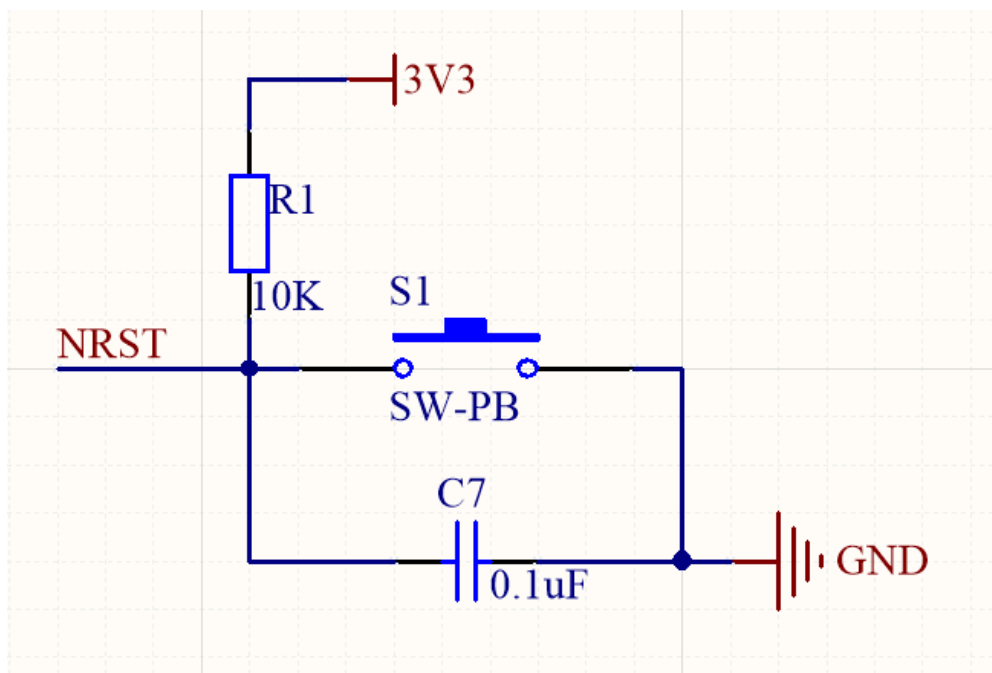


图 3-8 复位电路

3.3 温度检测

温度检测模块由 k 型热电偶与 MAX6675 共同构成温度检测单元。

K 型热电偶是一种温度检测元件，K 型热电偶一般与其它仪表：显示型仪器、电子调节型仪器和记录型仪器相配合使用。K 型热电偶的检测精度较高，这是因为 K 型热电偶和被测量对象直接进行接触采集的，由于没有中间介质的影响，所以采集到的数据更加可信。温度测量的范围非常广，从 -200°C ~ 1300°C 范围内的温度均可以进行测量。K 型热电偶使用非常的方便，其结构构造也很简单，它是两种不同的金属丝所组成，在热电偶的外部有保护热电偶的套管，热电偶的形状与细棒相似。由于其外形较小，采集温度时不受被测对象的大小和开头限制，安装与使用起来非常的方便。K 型热电偶具有灵敏度高、线性度好、抗氧化能力强、热电动势大、稳定性和均匀性都较好的优点，并且其价格较为低廉。K 型热电偶为工业中的许多项目带来了极大的方便与用途，被用户所广泛使用。

使用 MAX6675 芯片对将 K 型热电偶所采集到的烤箱内部温度数据进行传递，再将温度数据传递给 STM32 微控制器进行运算。MAX6675 是美信公司推出的一款支持 K 型热电偶接口的 12 位 ADC 转换芯片，SPI 数据输出格式，带有温度补偿功能，冷端集成在芯片的内部，测量范围为 0°C ~ $+1023.75^{\circ}\text{C}$ 。配合微控制器，可直接用于热电偶输出量的读取，其芯片引脚图如图 3-7 所示。

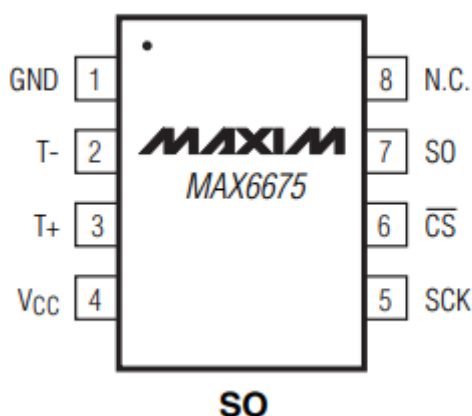


图 3-9 MAX6675 引脚图

各引脚定义如下：

GND, VCC：电源端

T-, T+：热电偶的冷端和热端，T-应该始终接地。

SO：SPI 数据输出端。

SCK：SPI 时钟信号端。

CS：SPI 片选端。

数据输出格式为：

BIT	DUMMY SIGN BIT	12-BIT TEMPERATURE READING												THERMOCOUPLE INPUT	DEVICE ID	STATE
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	MSB											LSB		0	Three- state

图 3-10 数据格式

其中 D15 固定为 0，D14~D3 对应 12bit 数据位，高位在前；D2 数据位指示热电偶是否断开。在正常运行期间，D2 = 0，当热电偶输入端开路时，D2=1，可以凭次位判断系统是否正常工作；D1 为芯片 ID，固定为 0；D0 为三态端。

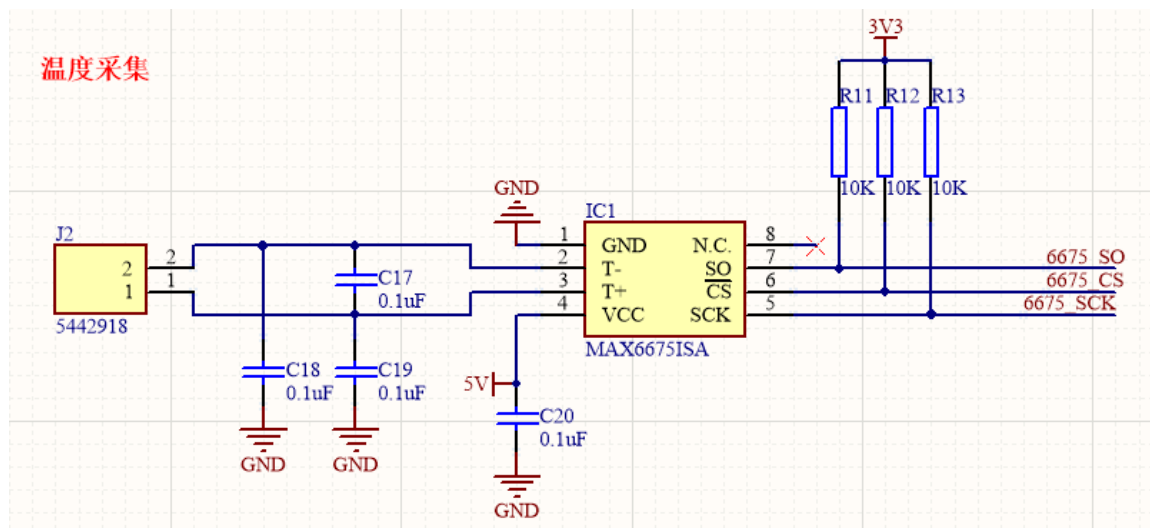


图 3-11 温度检测电路

3.4 温度控制模块

温度控制模块使用固态继电器来驱动交流电路的接通与关断，固态继电器是非接触式无触点新型开关设备，利用电子元器件（例如双向可控硅、晶闸管、三极管等半导体电子器件）的开关设备特性，可以做到电路的连接和断开没有接触和火花产生的，所以也被称为“无接触开关”。固态继电器是有四个连接端子的有源器件，其中的两个连接端子是输入控制端子，另外两个连接端子是输出控制端子。它有着放大驱动电流的功能，其具有隔离装置用来实现控制端子与输出控制端子之间的隔离，固态继电器可通过输入端子的微小的控制信号来直接驱动负载的大电流，非常适合用作大功率装置的驱动开关。与普通的电磁继电器相比较，它有着工作寿命长、接通无触点、安全性高、可靠性强、对外界无干扰、响应速度快等等优点，已被广泛使用。

将固态继电器、保险丝和电阻发热丝串接在 220V 的电路回路中，微控制器的控制端口控制着光电耦合器的接通与断开，光电耦合器的接通与断开控制着三极管的接通与断开，三极管接通与断开控制着固态继电器的接通与断开，进而达到微控制器通过控制引脚间接实现控制发热丝的工作。由于执行端使用 220V 强电压，因此在执行端的 220V 强电压电路中串联一个保险丝，执行端电路中串联一个短接端子，防止危险事件的发生。

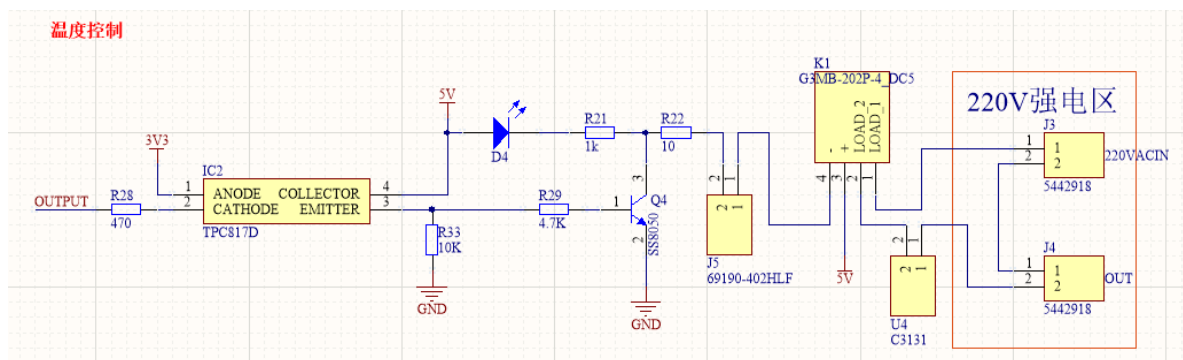


图 3-12 温度控制电路

3.5 显示模块

显示模块使用的显示器件为 LCD1602 显示器，该显示器的共有 16 个连接引脚，能够显示出 32（2 行 16 列）个字符，各个引脚的功能如表 3-1 所示：

表 3-2 LCD1602 引脚功能

序号	符号	说明	序号	符号	说明
1	VSS	电源地	9	D2	数据位
2	VDD	电源正极	10	D3	数据位
3	VL	液晶显示偏压	11	D4	数据位
4	RS	数据/命令选择	12	D5	数据位
5	R/W	读/写选择	13	D6	数据位
6	E	使能信号	14	D7	数据位
7	D0	数据位	15	BLA	背光正极
8	D1	数据位	16	BLK	背光负极

LCD1602 显示模块的硬件连接图如下图 3-11 所示。

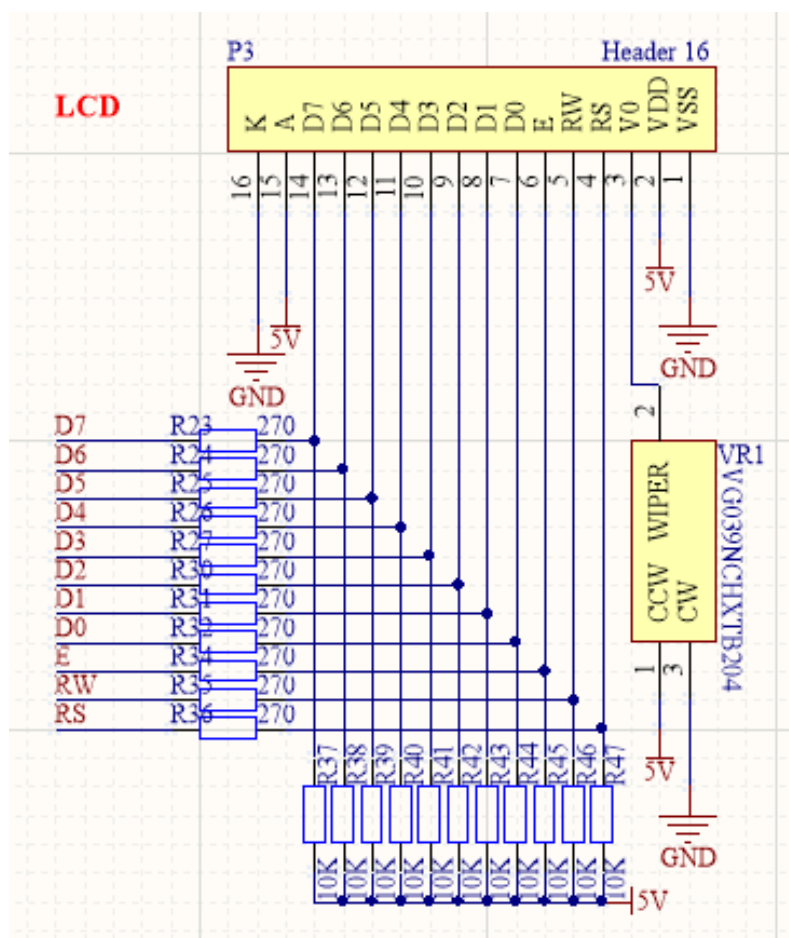


图 3-13 LCD1602 电路图

3.6 按键输入模块

按键模块采用独立式按键接法，共使用 4 个按键，分别是 K_STAR、K_STOP、K_UP、K_DOWN。

按键 K_STAR 控制恒温烤箱的开始运行，K_STOP 控制恒温烤箱的停止，K_UP 按钮用于在设置温度时提高温度值、K_DOWN 按钮用于在设置温度时降低温度值。

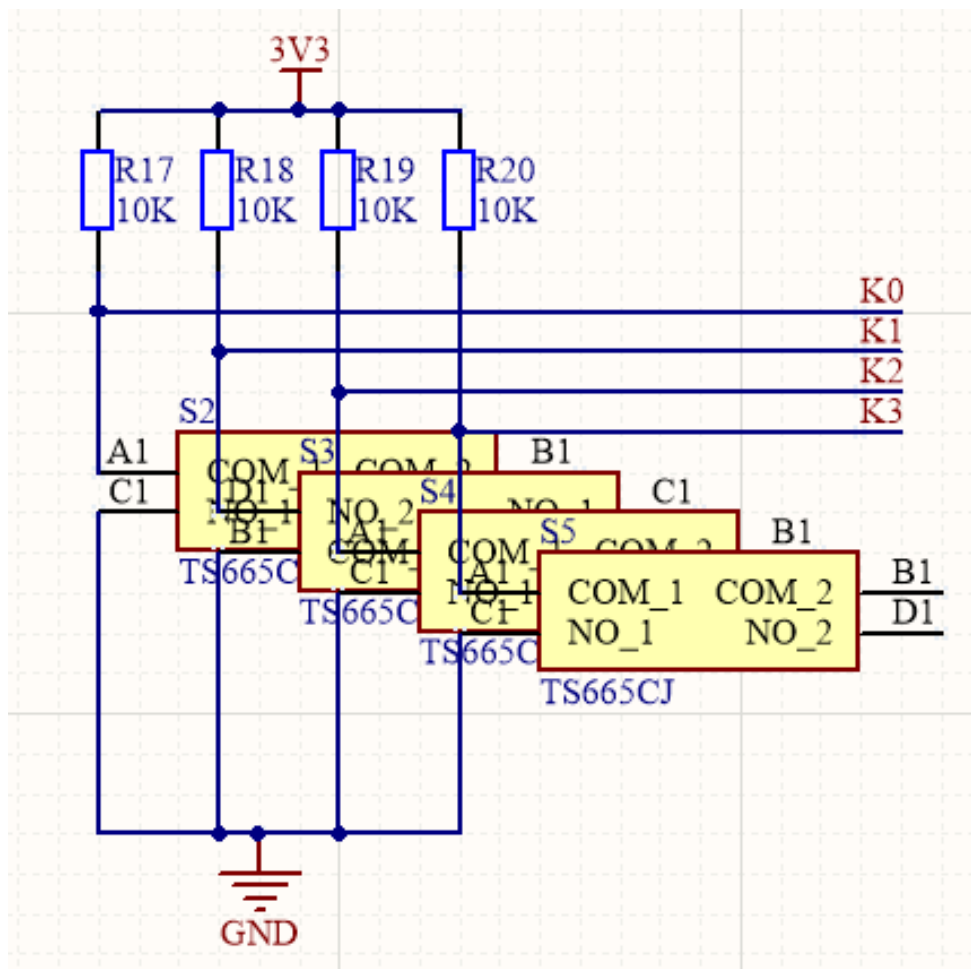


图 3-14 按键电路图

3.7 声光报警模块

声光报警模块使用 LED 灯和无源蜂鸣器，在恒温烤箱发生异常时，系统的蜂鸣器持续发出刺耳响声，LED 灯不停的闪烁，用来提醒用户烤箱此时工作出现了异常状况。具体的硬件连接图如下图 3-13 所示：

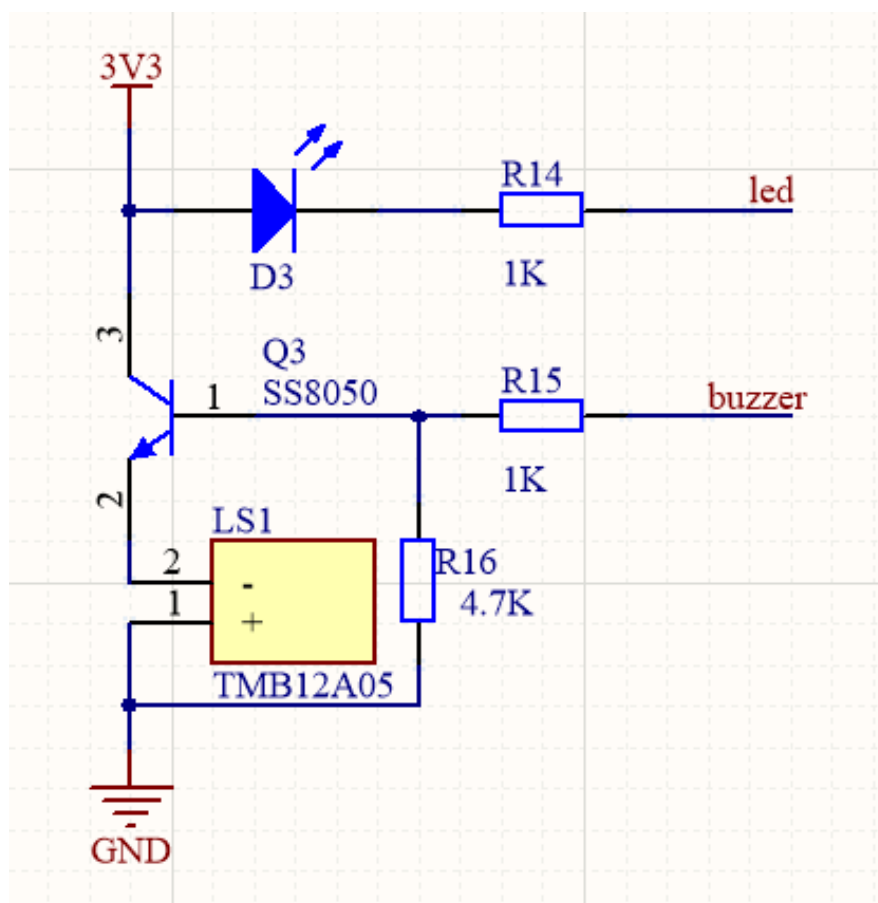


图 3-15 声光报警电路图

第四章 软件设计

系统的软件设计体现了控制系统的技术要求，也体现了各个模块的工作内容和模块间的相互配合，本设计恒温烤箱控制系统的软件有显示功能、温度检测功能、温度控制功能、声光报警功能、按键功能。

软件使用模块化的设计方式，在系统的各个软件模块中，屏蔽某一软件模块与其它软件模块共享使用的少量变量后，这个模块便可以单独地运行，而无需其它模块参与，这样的设计在系统软件的进行修改、调试以及移植时比较方便。即使将来根据需要扩展系统功能，也很容易编写相应的功能模块并在独立调试后将其添加到系统中。

4.1 主程序设计

主程序是整个程序的核心，它体现了整个系统的运行时工作流程。系统首先对外设和相关模块进行初始化，如 PID 参数、定时器、显示器、MAX6675、执行引脚等，然后系统执行按键程序来设置温度大小与执行时间，直到按下启动键，系统开始执行后续的程序。控制器驱动 MAX6675 采集温度数据，并将采集到温度数据显示到 LCD1602 显示屏上，同时使用该输入值进行 PID 运算，使用运算结果对执行器件进行 PWM 控制之后程序一直这样重复运行。

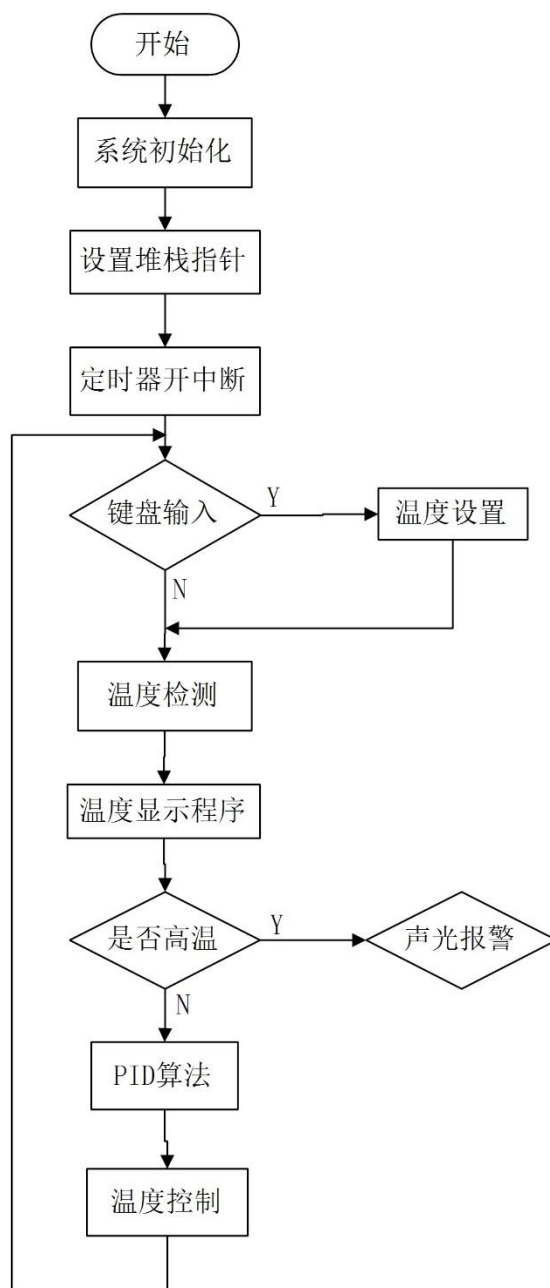


图 4-1 主程序流程图

4.2 PID 算法程序设计

根据先前对其基本思想、具体操作、公式推导的描述，实现基于积分分离的 PID 算法。首先需要对相关参数进行初始化时，根据系统性能要求选择合适的参数设定：采样周期 T 、PWM 控制周期 $pwmcycle$ 、比例系数 K_p 、积分时间 T_i 、微分时间 T_d ，这几个参数便就可以确定系数 K_1 、 K_2 、 K_3 的数值。然后根据控制器采集到的输入值计算每次的偏差数值，便可以通过此算法计算出输出增量。由于此执行机构是加热装置，控制量不能为负值，因此还需要对实际量进行处理。若实际控制量超出 PWM 控制周期，则令

实际控制量等于 PWM 周期值；若实际控制量是负值，则令实际控制量等于最小控制量数值，程序流程图如图 4-2 所示。

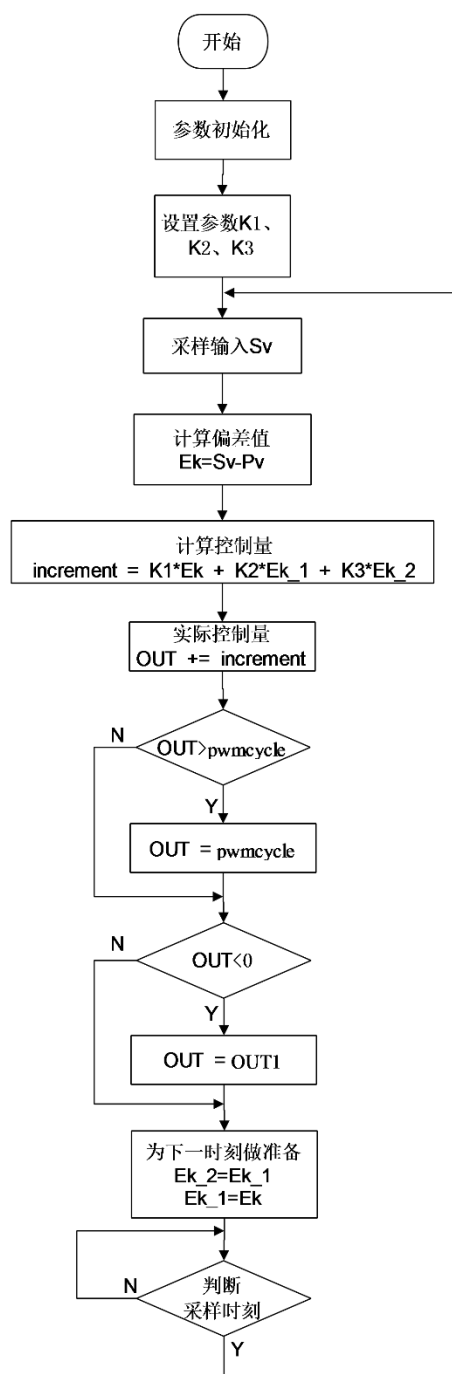


图 4-2 增量式 PID 控制算法程序流程图

4.3 温度检测程序设计

温度检测程序是整个系统的关键部分，只有正确测量到当前温度，才能对控制量数值的进行准确的运算，进而输出所需的控制信号，确保系统温度恒定。由 K 型热电偶测

量的模拟量数值将通过 MAX6675 转化为 12 位二进制数字量，通过单片机去读取 MAX6675 对应数据就能获取到温度值。

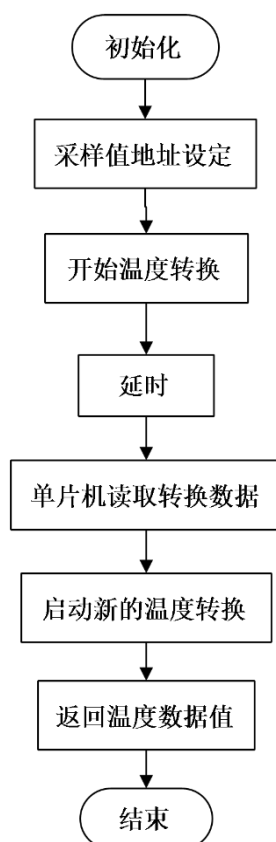


图 4-3 温度检测流程图

4.4 温度控制程序设计

温度控制使用单片机内部的定时器的输出比较功能，来输入 PWM 波形就能控制加热丝的通断时间。首先对 PWM 进行初始化，然后将 PID 算法结果，装入比较寄存器中，就能控制 PWM 波形中高低电平所占时间比，进而达到控制烤箱温度的目的，温度控制程序的具体实现如图 4-4 所示：

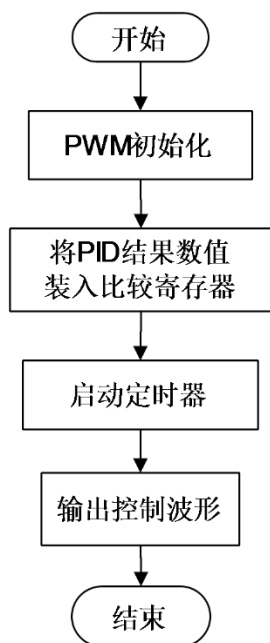


图 4-4 温度控制流程图

4.5 按键程序设计

本设计使用了 4 个按键：开始键、升温键、降温键、停止键。系统刚上电时，需要设定温度、时长，温度、时长设定完成后，按下开始键，系统便开始系统运行。短按升、降温键，设定温度能增、减 5 度，长按升、降温键，设定温度能每隔 1s 增、减 5 度，按键程序的具体实现如图 4-5 所示：

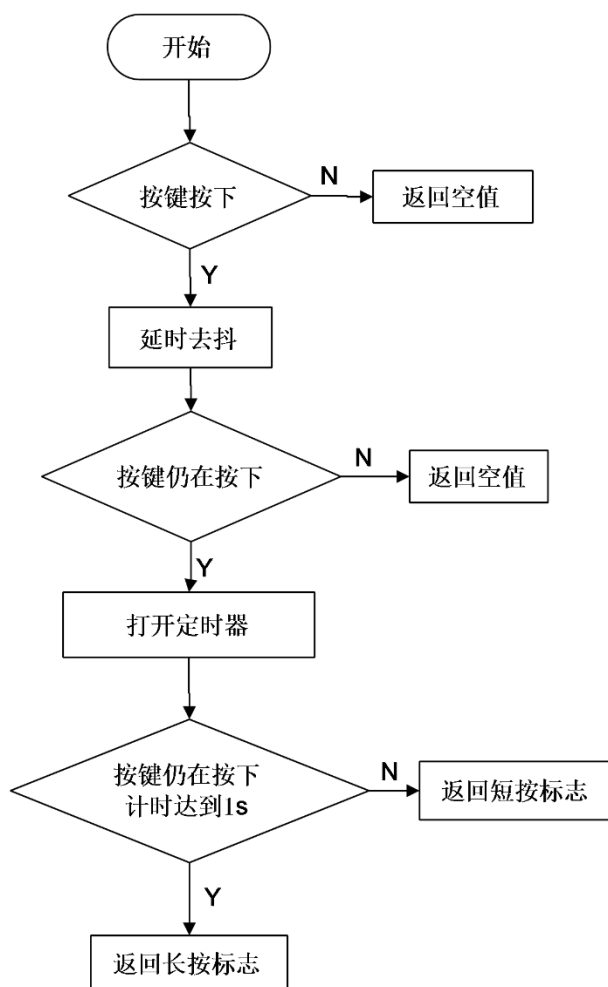


图 4-5 按键流程图

4.6 LCD 显示程序设计

由于 LCD1602 显示单元是慢速显示设备,在执行每个命令之前要先确保该模块已经初始化,在每次写入显示字符还需进行忙检测,当检测到有低电平返回信号就表示它不忙,否则输入的命令都将无效。如果要想显示屏显示字符,需要先把显示字符的地址输入,然后再告诉 LCD1602 显示器要显示的字符,显示程序的具体实现如图 4-6 所示:

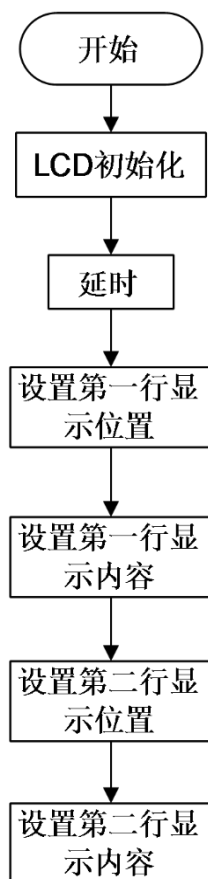


图 4-6 显示程序流程图

4.7 声光报警程序设计

当检测到实时温度高于设定值温度 10 度时，立即停止加热，并使 LED 灯不断闪烁、蜂鸣器持续发出刺耳响声，报警程序的具体实现如图 4-7 所示：

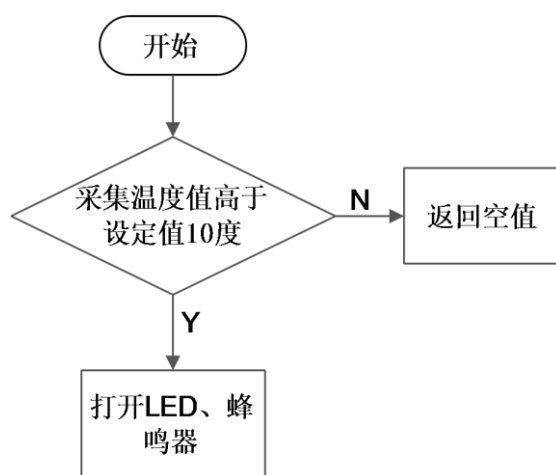


图 4-7 声光报警程序流程图

第五章 系统仿真与实验调试

5.1 仿真设计

该设计的仿真部分主要由仿真软件 Proteus 实现。Proteus 仿真软件可以实现 stm32 控制器和外围设备的电路和软件的仿真与设计。Proteus 提供许多仿真器件，使用者可以根据仿真需求在该仿真软件上自由添加组合。在制作硬件实物前，开发者可以先在该仿真软件上进行仿真验证，发现问题及时修改，避免损失。在使用 Proteus 进行仿真的前，可以将编译后的程序加载到仿真 MCU 中，在仿真运行时，MCU 可以根据输入信号做出相应的响应，并在显示器上的显示响应处理结果，整个仿真的过程接近于实际硬件的工作过程。

本次设计的加热装置采用 OVEN 作为系统的加热源，对 OVEN 的控制端施加电压便可以控制其加热，它能够通过其 T 端电压变化反应其温度状态。测温元件采用 TCK 进行采集温度，TCK 将采集到的温度数据输出到 MAX6675，MAX6675 通过与单片机的连接管脚将数据传递给 STM32 微控制器，在单片机内进行 PID 算法运算，根据运算的数值对加热装置进行加热，使 OVEN 的温度可以达到设定值并保持恒定，恒温烤箱控制系统的仿真设计图如图 5-1 所示。

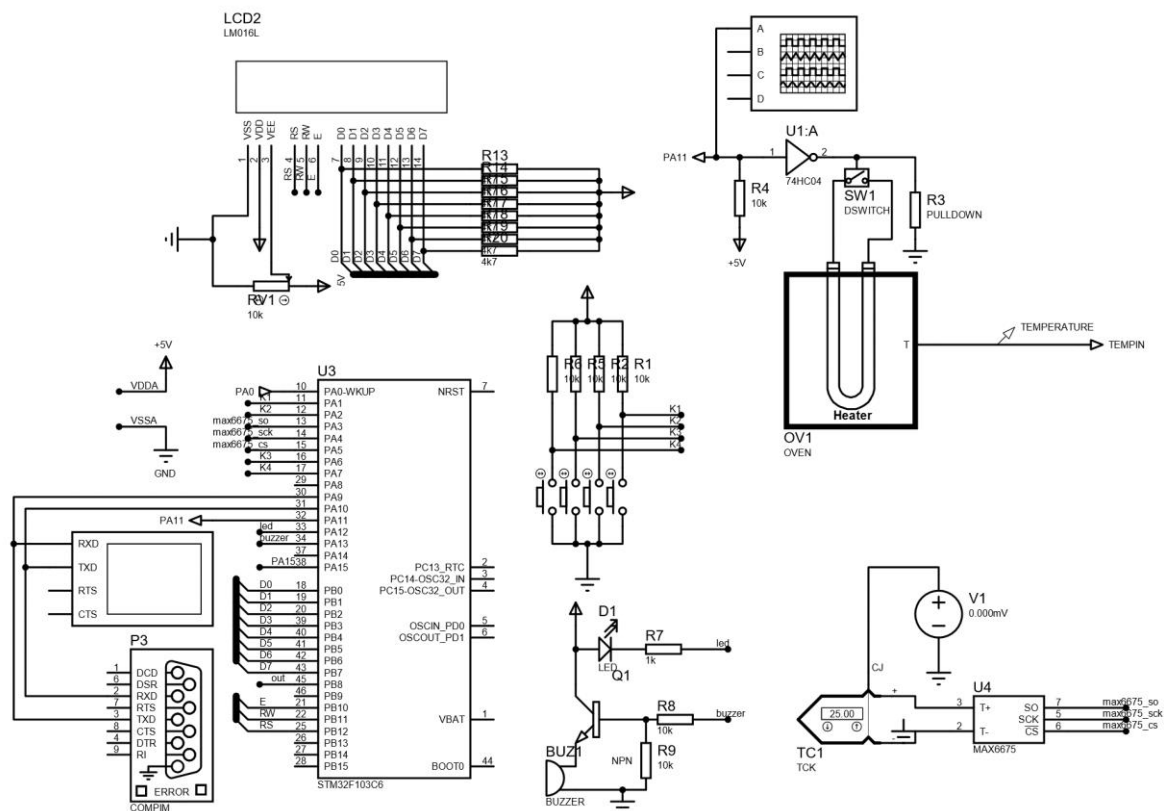


图 5-1 PROTUES 仿真图

5.2 参数整定

(1)时间部分整定

采样与计算时间设置为相同的数值 20us，PWM 控制周期时间设置为 100us。使用 STM32 内部定时器中断异常处理函数完成温度的采样、计算并输出 PWM 控制信号，保证三个任务可以同时计时、准时执行。

(2)比例部分整定

比例部分参数整定前，需要将积分和微分项消除掉，只使用纯比例进行系统控制。把积分部分系数 T_i 设为一个极大的值再把微分部分系数 T_d 设为 0，积分部分和微分部分就会被消除。然后由小到大的改变 K_p 的数值，并观察系统的响应速度是否够快，直到响应速度可以很快接近目标值并且还会出现一定范围的超调为止。如果仅使用比例部分，系统的静态误差就已经在设计的要求指标的范围以内，而且系统响应的曲线也可以满足设计的要求，则只使用纯比例部分的调节器就可以了，如果不能满足在进行积分和微分参数的整定。经过多次整定最终确定比例参数 K_p 数值为 5。

(3)积分部分整定

如果此系统仅仅只使用纯比例控制进行输出量的控制，系统静态误差不能够满足设计的要求，这时需要添加积分项来进行控制。调整积分部分的系数时，把积分系数 T_i 由很大的数值逐渐的减小（积分控制的作用将会逐渐增大），并能观察到输出的系统静态误差也应是逐渐减小的，直到消除为止（根据性能指标要求）。重复多次实验，直到对消除静差的大小感到满意为止。需要注意的是，此时的过冲量可能会出现大于原始量，这时需要适当减小一些比例系数 K_p 。经过多次整定最终确定积分时间参数 T_i 数值为 4000。

(4)微分部分整定

若系统经上面积分控制和微分控制进行调节后仍无法达到设计要求，此时考虑加入微分控制作用。调整微分部分的系数时，将微分项系数 T_d 从小到大进行整定（积分控制的作用将会逐渐增大），观察超调量与稳定性是否良好。比例项系数 K_p 和积分项系数 T_i 也要进行略微的调整，直到系统的静态和响应速度能够达到设计的要求为止。经过多次整定最终确定微分时间参数 T_d 数值为 2000。

5.3 仿真结果

将控制器目标温度分别设置为 50°C、60°C 和 70°C，通过串口打印出实时温度变化数据，绘制温度-时间变化折线图，观察增量式 PID 对温度的控制是否有效，系统是否存在静态、响应的速度是否够快、误差是否在允许范围内。目标温度分别设置为 50°C、60°C 和 70°C 时的温度-时间变化折线图分别如图 5-2、图 5-3、图 5-4 所示。

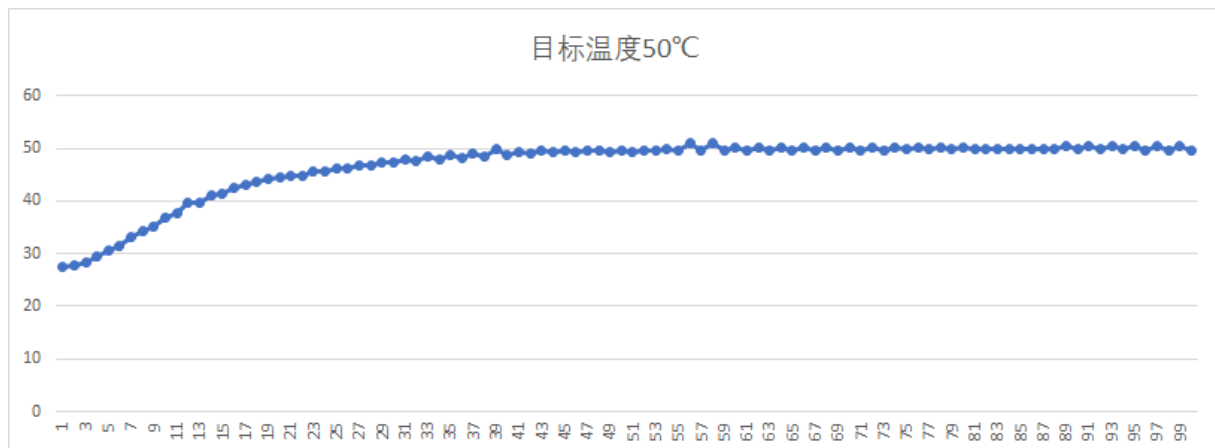


图 5-2 50°C 温度-时间变化折线图

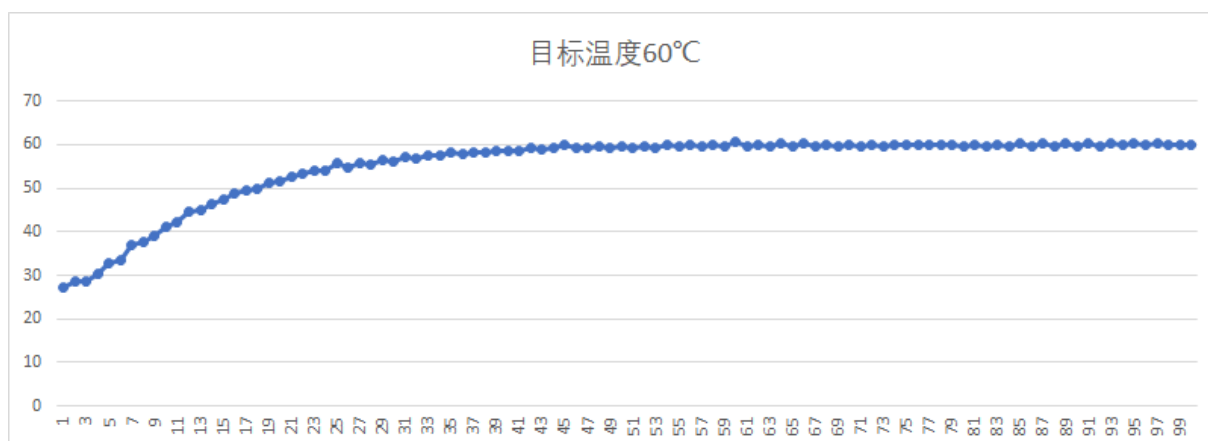


图 5-3 60°C温度-时间变化折线图

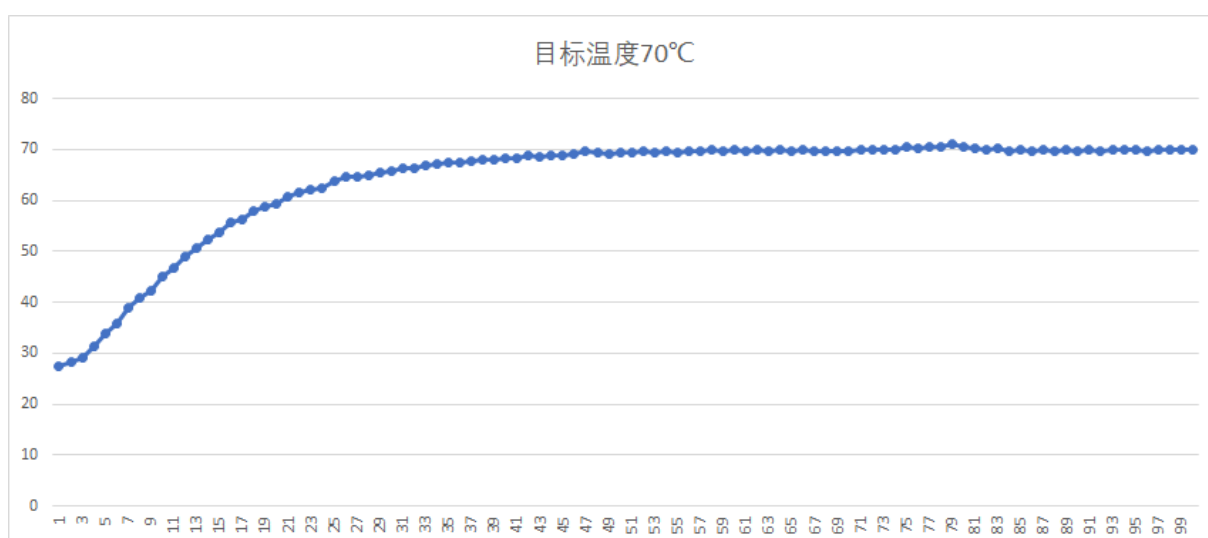


图 5-4 70°C温度-时间变化折线图

通过观察各不同目标温度的温度-时间变化折线图可以看出，此增量式 PID 算法得到的温度控制的变化曲线是完美的，温度能够迅速达到目标温度值，没有出现很大的过冲，并且温度能保持恒定，因此本设计采用增量式 PID 及以上参数的整定是可以满足恒温烤箱控制系统要求的。

结论与展望

根据本次课程设计的要求，并且通过认真分析了设计课题的需求和要求指标，在学习 PID 控制算法、硬件的设计、单片机工作原理与使用方法后，我独立完成了温度恒温烤箱控制系统设计和开发的工作。在设计过程中通过多次的理论推导和仿真验证，进一步表明了数字 PID 控制系统的动态性能指标是非常有优越性，特别合适于滞后对象的控制，因此对进行温度控制采用 PID 算法是较好的选择。

本次毕设，虽然我只使用到很简单的控制算法，但在设计过程中我从算法的选择到算法的实践完整的参与、投入其中，让我真正的意识到算法之美与算法的神奇，它在我的心中留下了种子，在以后我会学习更多关于算法方面的知识，了解更多算法的应用场景。

毕业设计是大学的最后一门课程，花费我近半年时间也付出了很多的努力，通过本次的毕业设计不仅是对我大学期间所学知识的检验，同时也是对我的资料收集能力、自学能力、实践能力的检验。同时在本次毕业设计中我也学习到了一定的知识，使我对项目的总体设计有了初步的了解，可以独立设计一些简单电路，然后对电路板的生产有一定的了解，并学会了使用 Altium Designer 进行硬件电路、PCB 的设计。在硬件的设计和软件调试过程中，出现了很多问题，最后通过分析、实验的方式解决了这些问题。通过本次学校安排的毕业设计，这对我思考和分析问题的能力有了极大地锻炼。

致 谢

至此毕业论文完成之际，谨向给予我指导的张老师表示诚挚的感谢！

从论文的开题，到系统的分析、设计、实现，以至论文的成稿，张老师给了我诸多的帮助和悉心的指导，其严谨的作风和认真的分析态度感染了我，使我认识到，学习不仅要注重方法，更要有严谨、主动的态度，这将对我以后的工作学习产生深远的影响！再次向我的指导老师张老师致以最诚挚的谢意！

另外，我还要感谢所有在我毕业设计期间给予我帮助的我的家人和同学。最后我要感谢安徽信息工程学院，我为能成为这所学校的学生感到自豪和骄傲！而我也要尽我所能将我从母校学到的知识应用于实际工作当中。

衷心的感谢所有帮助我的指导老师和同学！

由于自己的水平有限，论文中可能还存有诸多不足，恳请老师、同学批评指正。

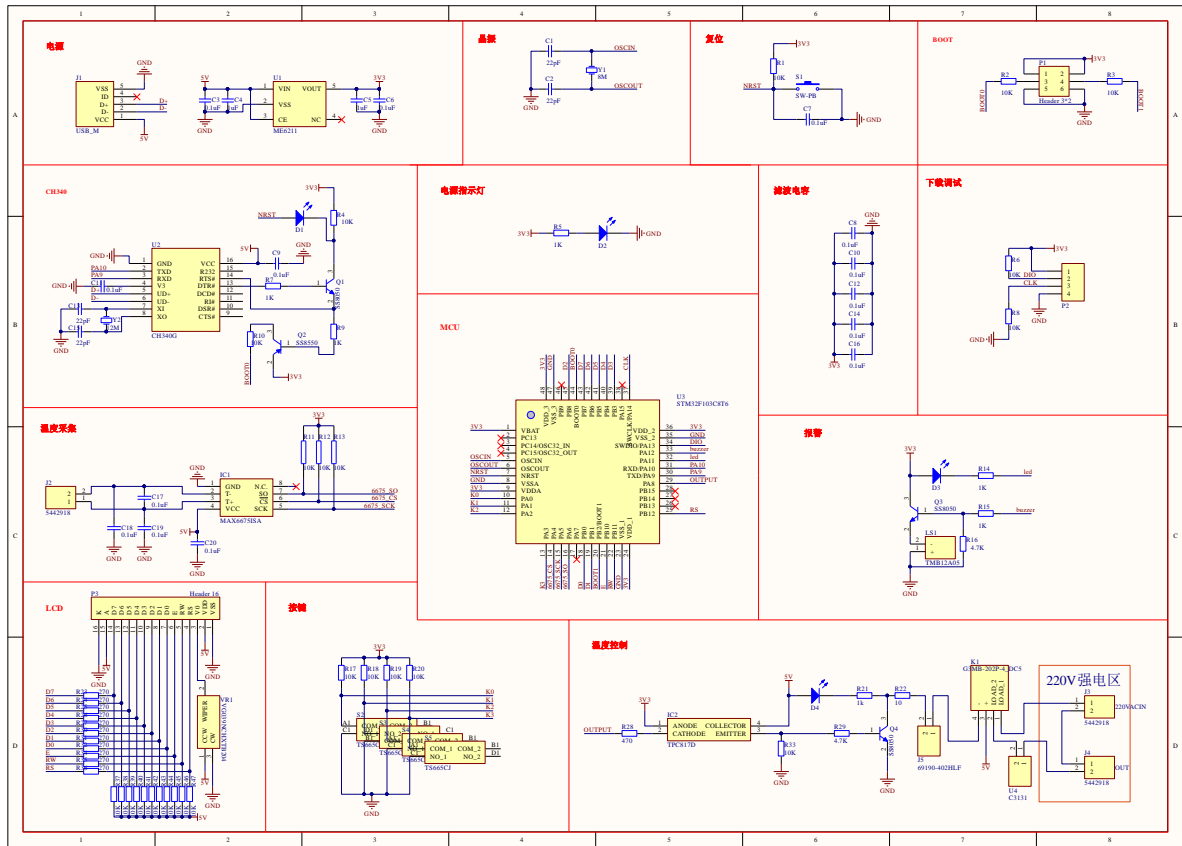
作者：

2020 年 6 月 5 日

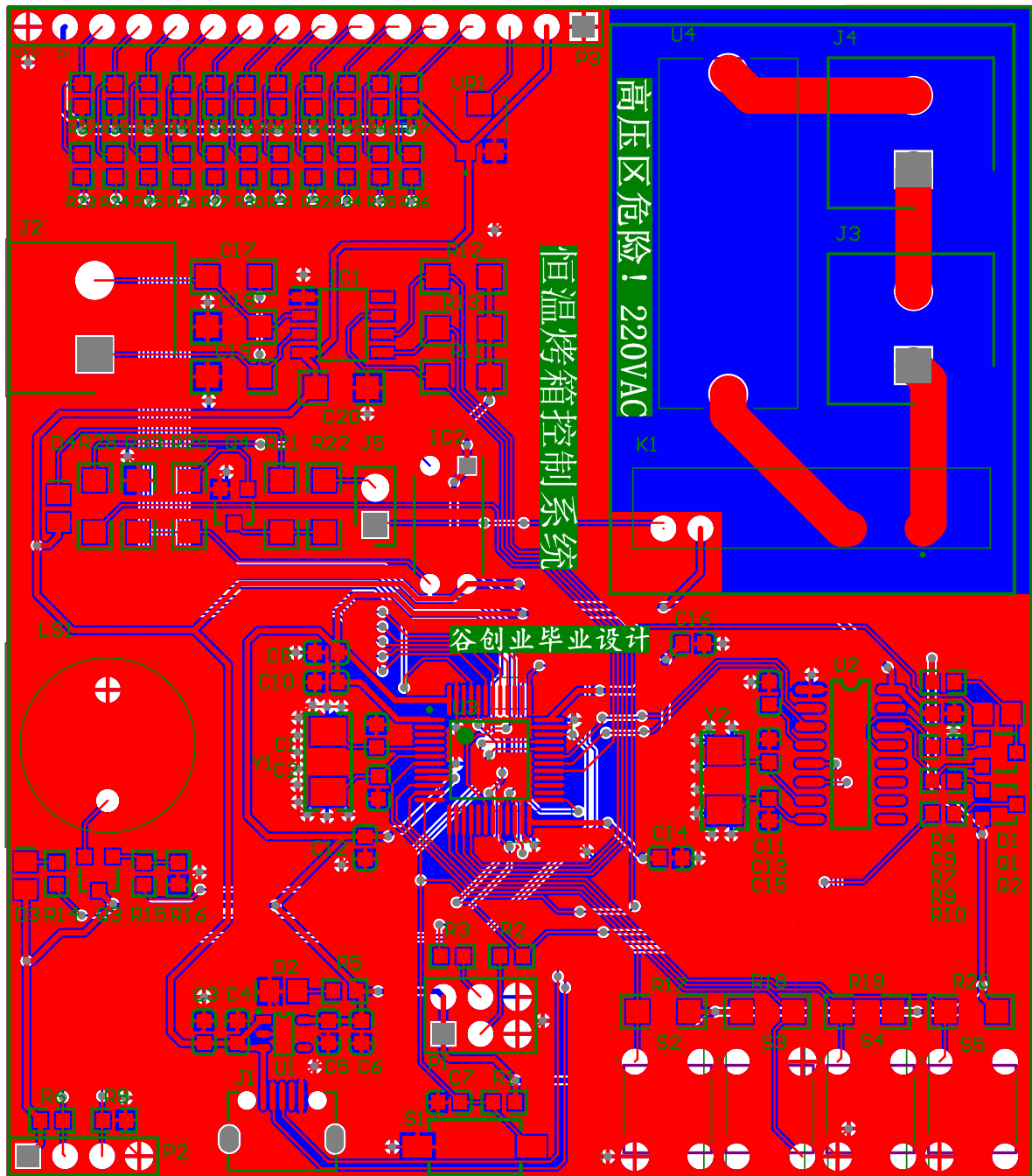
参考文献

- [1]梁剑平. 基于 STM32 单片机的汽车防盗系统设计与实现[J]. 玉林师范学院学报,2015,36(5).
- [2]张仲俊,汪材印. 基于 STM32 单片机的四轴飞行器飞行系统设计[J]. 集宁师范学院学报,2017,39(6).
- [3]陈东升,高俊侠,胡科堂.基于 STM32 的远程温控系统设计[J].电子产品世界,2011,18(5):30-32.
- [4]高立兵,康雁林.基于 AVR 单片机的 PID 温控系统设计[J].工业控制计算机,2010,23(4):91-92.
- [5]王桔,洪梅. 基于 STM32 单片机的恒温箱系统设计[J]. 长春大学学报（自然科学版）,2015,25(4):13-16,21.
- [6]杨伟,肖义平.基于 STM32F103C8T6 单片机的 LCD 显示系统设计[J].微型机与应用,2014(20):29-31,34.
- [7]李曦,周冬梅.基于 STM32 的无刷直流电机驱动板设计[J].科技传播,2015,0(17).
- [8]丁泽源,吴传秀,何军平,宋永献,王经卓,毕训银.基于 STM32 单片机的车载酒精检测系统设计[J].传感器技术与应用,2018,6(03):103-111.
- [9]雷慧杰.基于 STM32 的直流电机 PID 调速系统设计[J].现代电子技术,2016,39(8):165-167,170.
- [10]林森,刘志东,吕庆军.基于 STM32 的 PID 算法控制直流电机系统设计[J].产业与科技论坛,2017,16(2):78-79.
- [11]J. LIN, Z.-Z. HUANG. A novel PID control parameters tuning approach for robot manipulators mounted on oscillatory bases[J]. Robotica: International journal of information, education and research in robotics and artificial intelligence,2007,25467-477.
- [12]S. F. TOHA, M. O. TOKHI. PID and inverse-model-based control of a twin rotor system[J]. Robotica: International journal of information, education and research in robotics and artificial intelligence,2011,29(Oct. Pt.6):929-938.

附录 A 硬件原理图



附录 B PCB



附录 C 程序

PID 算法程序

```
void PID_Init()
```

```
{
```

```
    pid.T=200;    //采用周期
```

```
    pid.pwmcycle=1000;    //控制周期
```

```
    pid.Kp=5;
```

```
    pid.Ti=4000;
```

```
    pid.Td=2000;
```

```
    pid.OUT=0;
```

```
    pid.K1=pid.Kp*(1+pid.T/pid.Ti+pid.Td/pid.T);
```

```
    pid.K2=pid.Kp*(1+2*pid.Td/pid.T);
```

```
    pid.K3=pid.Kp*pid.Td/pid.T;
```

```
}
```

```
void PID_Calc( )
```

```
{
```

```
    static u16 calc;
```

```
    calc++;
```

```
    float t1,t2,t3;
```

```
    if(calc<pid.T) return ;    //判断计算时间
```

```
    pid.Ek=pid.Sv-pid.Pv;    //偏差
```

```
    t1=pid.K1*pid.Ek;
```

```
    t2=pid.K2*pid.Ek_1;
```

```
    t3=pid.K3*pid.Ek_2;
```

```
pid.increment=t1+t2+t3; //增量
pid.OUT+=pid.increment; //实际控制量
if(pid.OUT>pid.pwmcycle)
{
    pid.OUT=pid.pwmcycle; // 全量控制
}
if(pid.OUT<0)
{
    pid.OUT= pid.OUT1; //最小控制量
}

pid.Ek_2=pid.Ek_1;
pid.Ek_1=pid.Ek;

calc=0;
}
```

温度采集程序

```
void Max6675_Init() // max6675 初始化
{
    GPIO_InitTypeDef GPIO_InitStructure;
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOC,ENABLE);
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOD,ENABLE);

    GPIO_InitStructure.GPIO_Pin =GPIO_Pin_5;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
```

```
GPIO_Init(GPIOA, &GPIO_InitStructure);

GPIO_InitStructure.GPIO_Pin =GPIO_Pin_3;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOA, &GPIO_InitStructure);

GPIO_InitStructure.GPIO_Pin =GPIO_Pin_4;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOA, &GPIO_InitStructure);
}
u16 read_max6675() //采样
{
    u16 d,i;
    //so_1;
    cs_0;
    delay_us(2);
    for(i=0;i<16;i++)
    {
        sck_1;
        delay_us(2);
        sck_0;
        delay_us(2);
        d<<=1;
        if(so)
            d++;
    }
}
```

```
cs_1;  
return d;  
}
```

控制程序

```
void pwm_init( )  
{  
    TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;  
    TIM_OCInitTypeDef TIM_OCInitStructure;  
    GPIO_InitTypeDef GPIO_InitStructure;  
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);  
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_TIM1, ENABLE);  
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_11;  
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;  
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;  
    GPIO_Init(GPIOA, &GPIO_InitStructure);  
    //定时器配置  
    TIM_DeInit(TIM1);  
    /* Time base configuration */  
    TIM_TimeBaseStructure.TIM_Period = 999;//1MHz / (999 + 1) = 1KHz  
    TIM_TimeBaseStructure.TIM_Prescaler = 71;//1MHz 主频  
    TIM_TimeBaseStructure.TIM_ClockDivision = TIM_CKD_DIV1;  
    TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;  
    TIM_TimeBaseStructure.TIM_RepetitionCounter = 0;  
    TIM_TimeBaseInit(TIM1, &TIM_TimeBaseStructure);  
    /* PWM1 Mode configuration: Channel1 */  
    /* Channel 1, 2,3 and 4 Configuration in PWM mode */  
    TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;  
    TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
```



```
TIM_OCInitStructure.TIM_OutputNState = TIM_OutputNState_Disable;
TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;
TIM_OCInitStructure.TIM_OCIdleState = TIM_OCIdleState_Reset;
TIM_OCInitStructure.TIM_Pulse = 0;
TIM_OC4Init(TIM1, &TIM_OCInitStructure);
TIM_Cmd(TIM1, ENABLE);
/* TIM1 Main Output Enable */
TIM_CtrlPWMOutputs(TIM1, ENABLE);
}

void pwm_output(uint16_t pwm)
{
    TIM_SetCompare4(TIM1, pwm); //PWM 占空比
}
```

附录 D 主要参考文献的题录及摘要

1) C 和指针

【作者】KennethA. Reek

【出版社】人民邮电出版社

【出版日期】2018.04

【关键词】c 指针

【摘要】书中提供了许多有关 C 语言编程方面有关资源与深入讨论，通过详细讨论指针有关的基础知识与其他高级功能，它帮助学习者把指针的强大功能集成到自己的程序中，同时这本书还涵盖了几乎所有重要的 C 语言编程体系，例如函数、字符串、数据、运算符和表达式、指针、语句、数组、结构体和联合体。

2) STM32 库开发实战指南(第 2 版)

【作者】刘火良 杨森

【出版社】机械工业出版社

【出版日期】2017.5

【关键词】STM32F103 库

【摘要】本书基于 STM32F103 芯片进行全面细致的讲解，着重介绍讲解 STM32“库”的分析思路和使用方法。在书中大量示例基础上，还给出了怎样全面使用固件库开发项目的具体演示的示例，对初学者非常友好。在对固件库的学习与使用的基础上，作者又进一步讲解了 TCP / IP 协议与的嵌入式操作系统的结合，可以使学习者一步步深入嵌入式系统的开发、系统的掌握使用 STM32 库的开发方法。

3) 自动控制原理（第七版）

【作者】胡寿松

【出版社】科学出版社

【出版日期】2019.2

【关键词】经典控制理论 现代控制理论

【摘要】本书全面的介绍了自动控制的相关基础理论与应用，对自动控制的基础概念以及数学模型和结构图、复域、时域和频域控制系统信号流程图都有着全面深入详细的介绍，并对线性离散系统、线性控制系统、非线性控制系统也都做出了非常详细的介绍。