



# 北京航空航天大学

## 模拟电子技术基础（实训项目）中期报告

### 题目：可设定的恒温温度控制器设计

学院名称：高等理工学院

专业名称：自动化

学生姓名：李谨杰

学生学号：16231235

指导教师：唐瑶、肖瑾

邮箱：lijinjie@buaa.edu.cn

电话：15652587808

## 目录

1. 课题背景及意义	3
2.1 研究目标	3
3. 本人工作整体综述	5
3.1 硬件部分	5
3.2 软件部分	6
3.3 其他工作	6
4. 本人工作详细介绍-硬件	6
4.1 单片机控制系统的设计与调试	6
4.2 PID 电路的设计与调试	7
4.3 DS18B20 的电路配置与调试	10
5.本人工作详细介绍-软件	10
5.1 测温电路编程	11
5.2 蓝牙部分编程	11
5.3 显示电路编程	12
5.4 按键电路编程	13
5.5 PWM 输出电路编程	13
5.6 制冷部分编程	14
6.本人工作详细介绍-其他	14
6.1 水箱的外形设计与总装	14
6.2 PCB 电路板的绘制	15
6.3 模拟 PID 调节	17
7.对于这次项目的经验	19
参考文献	20
附件 核心程序源码与 PCB	21

## 1. 课题背景及意义

温度是生活及生产中最基本的物理量,它表征的是物体的冷热程度。自然界中任何物理、化学过程都紧密地与温度相联系。恒温温度控制器在日常生活中应用广泛。在普通百姓的生活中,冰箱、空调、恒温箱、智能洗衣机等,都需要一套恒温控制系统来控制温度。以此满足人们的需求。在工业方面,温度是工业生产中主要的被控参数之一,与之相关的各种温度控制系统广泛应用于冶金、化工、机械、食品等领域。温度控制是工业生产过程中经常遇到的过程控制,有些工艺过程对其温度的控制效果直接影响着产品的质量,因而设计一种较为理想的温度控制系统是非常有价值的。

所以,合适的温度控制系统在社会中有着重要的意义。以此项目目标为参考,本产品可能有如下用途:在婴幼儿领域,给婴幼儿冲泡奶粉时,需要将奶粉调试到适合幼儿饮用的温度。人们可能希望在照看孩子的同时远程控制奶粉温度,或者希望只冲泡一次,就可以定量多次给幼儿摄入固定温度的奶粉;在基础生命科学领域,需要远程操控恒温箱的温度,以此来保证箱中培养皿细菌的生长条件。

温度控制类型的产品在行业内一般设计为在机器上操作,无法远程控制,不能很好满足上述要求。本产品既可以精确调节液体温度在 50-100 度之间,又可以用手机远程控制且实时监控温度,很适合上述使用情境。还有很多生活中的事件可以用本产品来完成,因此本产品前景广阔,很有研究的意义。

## 2. 研究目标及研究内容

### 2.1 研究目标

设计一个可以保持恒定温度的恒温控制系统,并且具有以下功能:

1. 可以实现温度的无级调节(以 1℃为单位)。
2. 可以使恒温箱在 50~100℃内保持恒温,并且误差不超过 1℃。
3. 可以用微信小程序显示并控制恒温温度。
4. 具有漏电和失控保护功能,具有一定的安全性。
5. 以 220V 交流电为电源。

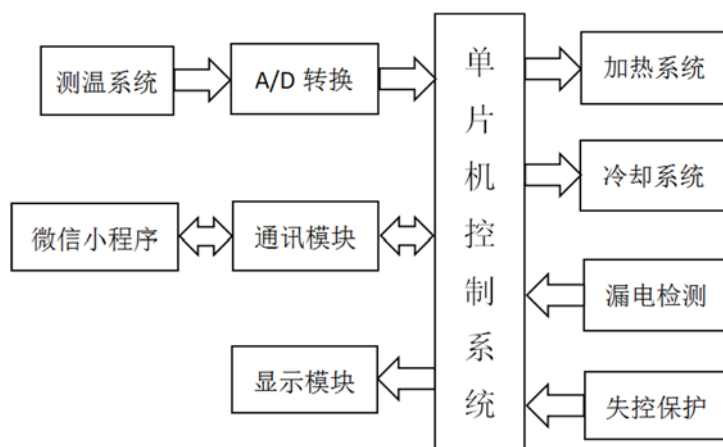


图 1：恒温控制系统硬件结构框图

## 2.2 具体研究内容：

### 2.2.1 核心控制系统

选用 STM32ZET6 核心板，由 STM32F103C8T6 单片机连接各个部分。

### 2.2.2 测温系统

选择铂电阻 Pt100 测温，铂热电阻可将温度信号转化为模拟电压信号，再通过 A/D 转换，转化为可供单片机处理的数字信号。

### 2.2.3 显示模块

采用四个分立的七段共阴极数码管和译码器的组合，可以显示四位数字的温度值，精确到小数点后一位。

### 2.2.4 模拟 PID 控制模块

选用 OP07 高精度放大器构建减法、反向比例、积分、微分、反向加法电路，通过调节电位器调整 PID 参数。

### 2.2.5 电源模块

由于产品整体采用 220V 50Hz 的市电作为整体电源，因此电源需要将 220V 50Hz 的交流电转变为 $\pm 12V$  供给控制电路中的运算放大器、转变为+12V 供给制冷模块、转变为+5V 供给控制电路。

### 2.2.6 PWM 调制模块

采用基于 555 计时器的多谐振荡器来实现 PWM 调制，利用数字电位器实现

与单片机的通讯，通过调整输出方波的占空比，控制继电器的通断进而改变加热和制冷的功率。

#### 2.2.7 继电器驱动电路：

由于单片机和 PWM 电路可能存在输出电流过小、不足以驱动继电器的问题，因此需要驱动电路将输出电流放大、驱动继电器；通过在继电器的控制端加 0V 和 5V 的电压来控制加热、制冷电路的通断，实现“开关”功能。

#### 2.2.8 加热模块

利用电阻在通电是能够产生热量的原理，对系统进行加热。为了增大加热体与系统的接触面积、使加热更加均匀，我们采用了在恒温箱底部粘贴加热板的方法来对恒温箱进行加热。由于加热板贴在恒温箱外部，这种方法也降低了恒温箱的加工难度。

#### 2.2.9 制冷模块

为了使较高的温度迅速降到设定温度，我们设计了制冷模块来加快系统的散热速度。初始我们采用了 12V 140W 的半导体制冷片，测试后发现降温效果不明显，后来采用了风扇加快散热的方法，显著加快了散热速度。

#### 2.2.10 蓝牙模块

采用 HC-08 模块，采用蓝牙 BLE4.0 支持安卓和 IOS 系统。

#### 2.2.11 手机端模块

利用 Android studio 开发安卓版手机 App，运用内置的蓝牙 API，使 App 与蓝牙模块连接，实现实时温度的接收和设定温度的输入，完成手机端显示温度与控制温度的功能。

### 3. 本人工作整体综述

#### 3.1 硬件部分

1. 单片机控制系统的设计与调试
2. PID 电路的设计与调试
3. DS18B20 的电路配置与调试

## 3.2 软件部分

1. 全部单片机程序的编写

## 3.3 其他工作

1. 水箱整体设计
2. 水箱外壳制作与总装
3. 大部分 PCB 电路板的绘制
4. PID 参数的调节
5. 部分器件的采购

# 4. 本人工作详细介绍-硬件

## 4.1 单片机控制系统的设计与调试

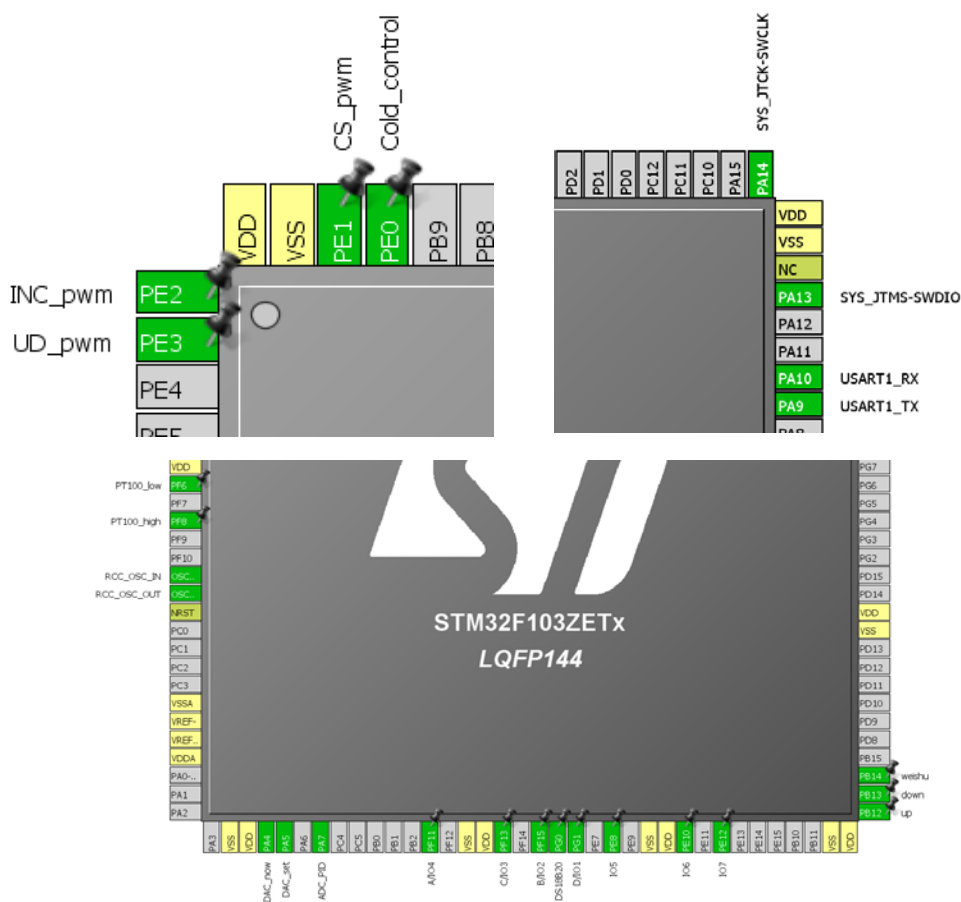


图 4.1：单片机引脚分配

我们一开始选用 stm32 F103C8T6，但由于 C8T6 没有 DAC 接口，于是又更换为 STM32F103ZET6 核心板。经统计：

显示电路占用 7 个 IO 口，名称为 IO1~IO7。

按键电路占用 3 个 IO 口，名称为 weishu、down 和 up

测温电路占用 2 个 ADC 口，名称为 PT100\_low 和 PT100\_high

数字测温元件 DS18B20 占用 1 个 IO 口，名称为 DS18B20

控制的 PWM 占空比的数字电位器占用 4 个 IO 口，名称为 CS\_pwm，Cold\_control，INC\_pwm 和 UD\_pwm。

电路 PID 占用 2 个 DAC 口和一个 ADC 口，名称为 DAC\_now、DAC\_set 和 ADC\_PID。

总计使用 15 个 IO 口，3 个 ADC 口，2 个 DAC 口。

STM32ZET6 核心板可以满足使用需求。

#### 4.2 PID 电路的设计与调试

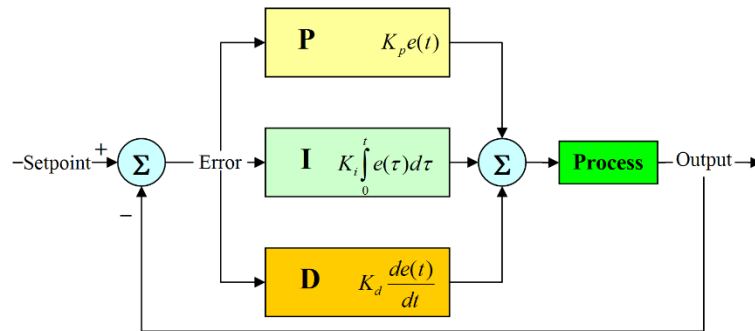


图 4.2: PID 控制原理

如图，PID 控制的理论公式如下：

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad [1]$$

其中， $u(t)$  为控制输出， $e(t)$  为设定值减去实际值，即误差量， $K_p$  为比例参数， $K_i$  为积分参数， $K_d$  为微分参数。在电路设计时，我将实际温度与设定温度线性对应为两个电压值，分别通过 DAC 输出，再通过一个减法电路将其相减得到误差  $e$ 。之后将误差  $e$  输入比例、积分、微分电路，进行运算，然后通过加法电路将三量相加并改变符号，最后通过 ADC 采集电压信息  $u(t)$ ，从而控制 PWM 输出的占空比，以起到调节加热功率的目的。

因为 PID 要求,当实际值与设定值相等时,比例项为零,积分项不增加,因而调整 PID 电路的偏置电压便十分重要。同时,我设计的 ADC 与 DAC 范围都是 0~3.3V,需要运放的输出值在这个范围内。结合中发电子市场已有的运放种类,我最终选择了 OP07 这样一款带偏置电压调整功能的高精度运放。

综上所述,电路中需要减法电路、反向比例电路、积分电路、微分电路和加法电路。这些电路模电课上已经讲过,我不再叙述原理,只把设计呈现如下:

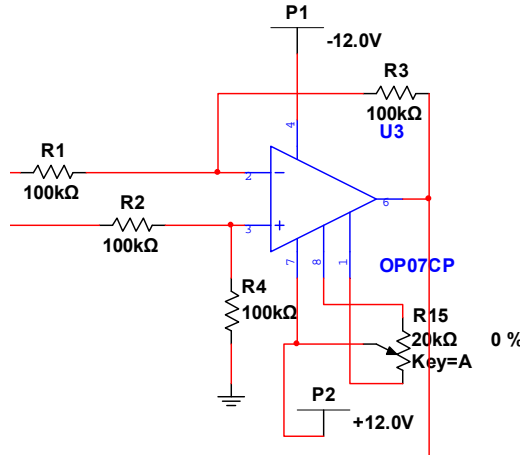


图 4.3: 差分比例运算电路

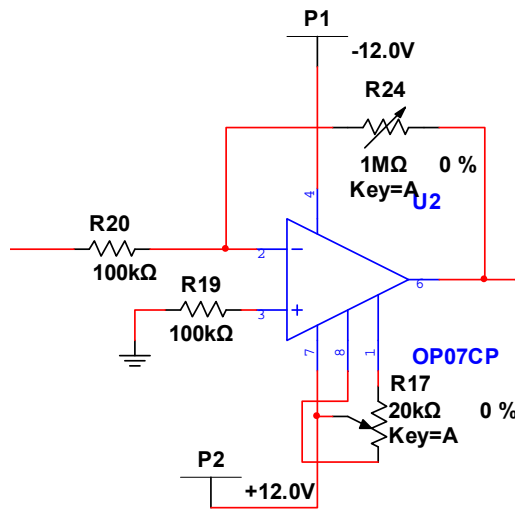


图 4.4: 反相比例运算电路

其中,比例参数  $K_p = \frac{R_{24}}{R_{20}}$ 。



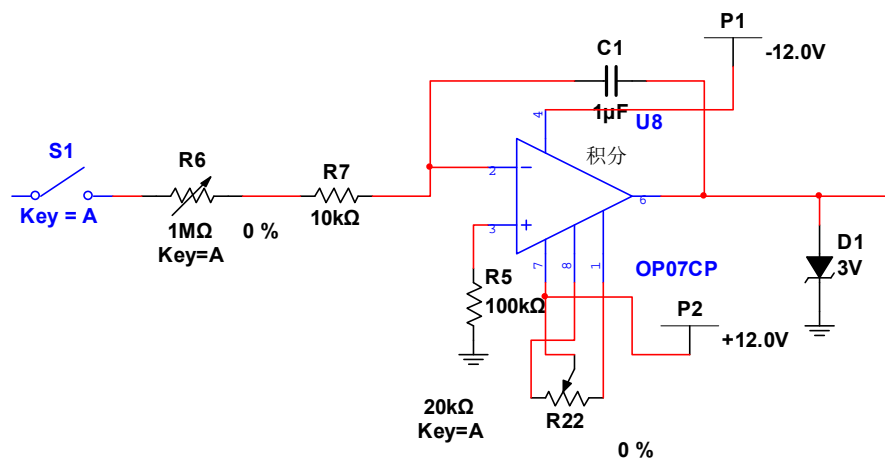


图 4.5: 积分运算电路

其中，积分参数为  $K_i = \frac{1}{(R_6 + R_7)C_1}$ 。

为避免积分电压值过大，我在输出端加一个 3V 稳压管限幅。

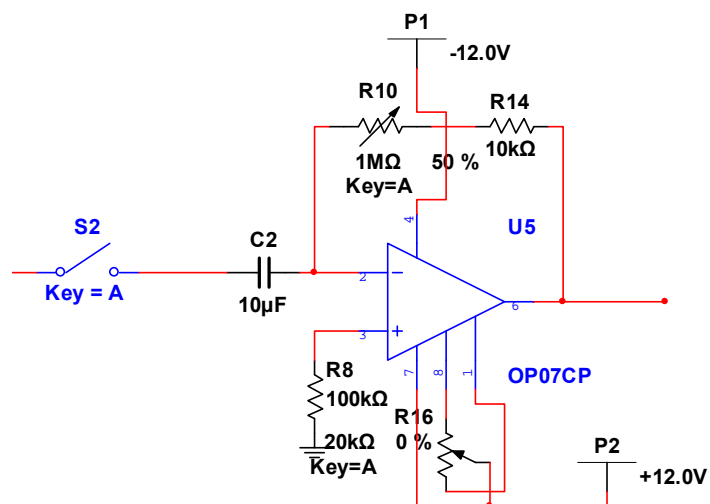


图 4.6: 微分运算电路

其中，微分参数为  $K_d = (R_{10} + R_{14})C_2$ 。

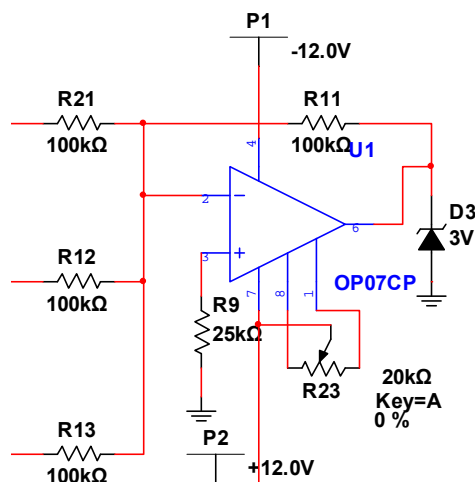


图 4.7：反相求和运算电路

其中,  $R_9 = R_{21} // R_{12} // R_{13} // R_{11} = 25k\Omega$ 。由于单片机的 ADC 采样范围是 0~3.3V, 为保护 ADC 口, 需要确保输出电压最大不超过 3.3V, 最低为 0V 左右。加入稳压管后, 若输出为正, 稳压管进入稳压区域工作; 若输出为负, 稳压管相当于二极管导通, 将输出电压限制在 -0.7V。经实测, 需要三个 3V 稳压管并联, 才能在信号最大时将输出限幅在约 3.24V。

至此, PID 电路设计完毕。

### 4.3 DS18B20 的电路配置与调试

查询 DS18B20 的数据手册可知, 电路图如图 4.8<sup>[2]</sup>。

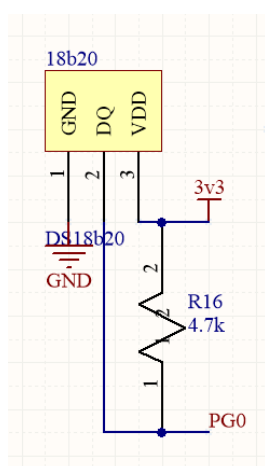


图 4.8：DS18B20 电路图

## 5. 本人工作详细介绍-软件

## 5.1 测温电路编程：

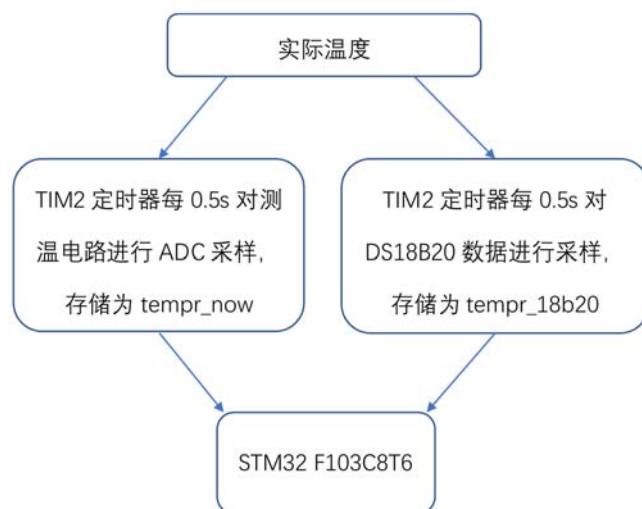


图 5.1：测温电路程序框图

我们主要设计了两个测温电路。第一个通过对铂电阻测温电路进行 AD 采样，由标定方程转化为实际温度；第二个通过对 DS18B20 数字温度传感器进行数据读取。我们设定的采样周期为 0.5S，如果时间太长会导致测温迟缓，时间太短会导致温度上下震颤不稳定。

每 0.5S 触发时钟中断程序，进行上述采样。其中 ADC 采取 DMA 方式采集两个通道的数据，并用采集五十个数据取均值，实现均值滤波的功能

## 5.2 蓝牙部分编程：

蓝牙收发通过串行通信（USART）实现。在串行通信中，一个字符一个字符地传输，每个字符一位一位地传输，并且传输一个字符时，总是以“起始位”开始，以“停止位”结束。在进行传输之前，双方一定要使用同一个波特率设置。波特率就是每秒钟传输的数据位数<sup>[3]</sup>。

常用的两种基本串行通信方式包括同步通信和异步通信。我们使用的是异步通信。异步通信规定传输的数据格式由起始位（start bit）、数据位（data bit）、奇偶校验位（parity bit）和停止位（stop bit）组成<sup>[1]</sup>。所以，需要进行蓝牙模块的初始配置。

为此，我们购置了 HC-08 蓝牙模块。HC-08 为蓝牙 4.0BLE 通信，支持 IOS

和安卓系统的连接，选择该模块是为小程序的全平台适用进行准备。通过 TTL 转 USB 线，用电脑对 HC-08 发送 AT 指令进行相关配置。我们的蓝牙设置如下：

蓝牙设备名称：MDXWZ-2.0

串口参数：波特率 115200，停止位 1 位，无奇偶校验位。

模块角色：从（slave）。在这个模式下，蓝牙设备可以被搜索到，而不会主动搜索其他蓝牙设备。

配对码：1234（默认）

配置好蓝牙模块后，需要在单片机端配置相同的串口通讯参数。对于蓝牙收发数据，我最开始采用的是 HAL 库轮询模式接收数据，通过在主函数循环体内编写收取函数，不断接收蓝牙的数据并进行清零。这种方式简单易写，但由于循环体内同时有数码管显示函数，不断进行接收会占用单片机一定的时间，导致数码管不断闪烁，无法达到一直显示的效果，而且空闲状态下这种接收方式十分占用资源。

因此，接下来我采用了串口中断接收蓝牙数据的方式。当蓝牙发送的数据达到指定字符数时，触发中断，单片机主程序冻结，保存现场，转而实现中断程序，即对接收数据进行处理。中断程序处理完毕后，再转而继续进行主程序。这种方式在数码管正常显示的同时，做到了接收蓝牙数据，但是由于触发中断的字符数是固定的，难以做到不定长接收。因此，我又采取了 DMA 加串口 IDLE 空闲中断模式。

DMA 指直接存储器访问功能，用于在外设与存储器之间以及存储器与存储器之间提供高速数据传输。可以在无需任何 CPU 操作的情况下通过 DMA 快速移动数据[1]。因此，程序开始阶段开启 DMA 和 IDLE 空闲中断，当一次数据接收完成时，触发空闲中断，中断函数调取接收数组对接收到的数据进行处理。这样，就实现了蓝牙配置的全过程。

### 5.3 显示电路编程：

显示部分需要实现两个功能：

- 1.显示四位不同数字
- 2.设定温度时，调整的那一位数字开始闪烁。

首先我根据雷彤形的真值表，编写程序改变各个 IO 口的高低，以控制特定的数码管显示特定的数字，我将此函数命名为 `light`。

显示电路需要很快地循环点亮四个数码管 (0.1s)，在人眼视觉暂留作用下，将使人同时看到四位不同数字。因此，必须将点灯函数放在主循环体内，并且将其他占用 MCU 时间较多的任务放到中断中。这样可以确保数码管可以循环连续点亮，防止闪烁现象。

数字闪烁方面，我采用时钟 TIM3 定时中断翻转标签来控制。我继续编写的程序框图如下：

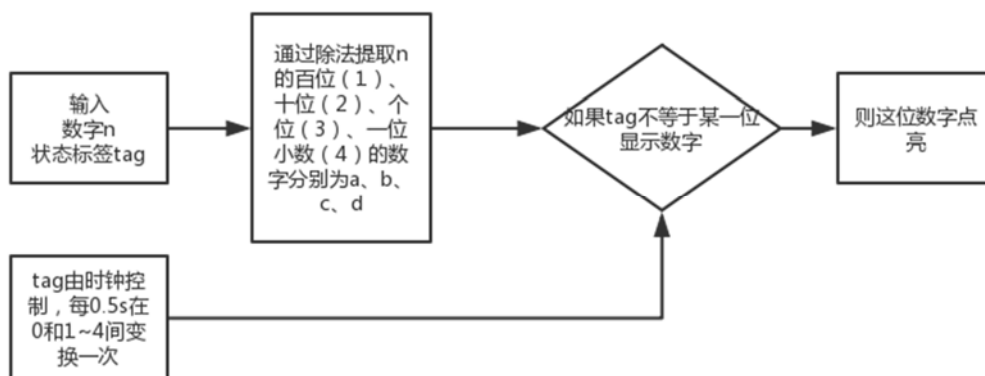


图 5.2 点灯代码流程图

如果实现“正常显示数字 520.1，并且十位数字进入编辑状态”的功能，则  $n=520.1$ 。经过提取后，数码管从左至右四个位置分别显示 5、2、0、1，tag 的值为 2。每 0.5s，tag 由 2 变为 0，此时十位数字点亮；再过 0.5s，tag 由 0 变为 2，十位数字熄灭，由此完成一个周期，在循环结构作用下，十位数字不断闪烁。

至此，显示电路已实现全部预期功能。

#### 5.4 按键电路编程：

每个按键按下去时，会触发下降沿中断进行对应处理。

#### 5.5 PWM 输出电路编程：

PWM 部分需要编写程序对数字电位器进行阻值调节。X9C104 数字电位器可以通过外部信号实现 100 档阻值调节，具体调节过程为：片选信号 CS 低有效，INC 接口每来一个负脉冲，阻值变化一档，如果 U/D 口为高，则增大阻值；否则

减小阻值<sup>[4]</sup>。

我编程思路如下：

程序初始时 U/D 为低，给 INC 连续 100 个负脉冲，电位器归零，记为函数 Init X9312。

当从 ADC 采样电压值后，对应于某个电阻值  $R_{set}$ ，此时的电位器实际阻值记为  $R_{now}$ 。在 while 主循环中不断比较  $R_{set}$  和  $R_{now}$  的值，并将  $R_{now}$  向  $R_{set}$  逼近，直至两者最终相等，数字电位器调节完毕。

## 5.6 制冷部分编程：

对于制冷，只设计开与关两档。当实际温度大于设定温度一度时，开启制冷，关闭制热。当实际温度降到设定温度时，关闭制冷，开启制热。

# 6. 本人工作详细介绍-其他

## 6.1 水箱的外形设计与总装

水箱外壳由全木制作，水箱外边有全石棉保温材料包覆处理，设计时主要考虑了按键、数码管的位置，加热装置与其余部分的隔热，各器件的协调摆放等因素。

第一版设计时考虑了半导体制冷片系统的巨大体积，于是决定将电路放置在水箱下边，变压器与继电器放置在制冷片两侧，效果如图 6.1：

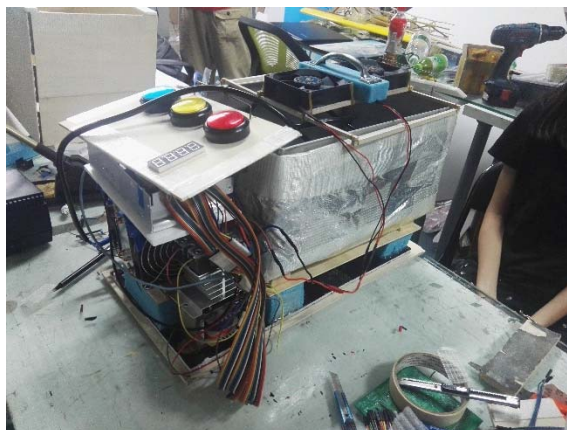


图 6.1：素组水箱

装上外壳以后，效果如下图：



图 6.2: 水箱整体图

第二版设计取消了半导体制冷片，计划将电路板放在原先放置制冷片的位置，这一版水箱仍在制作中，外形体积会更小。

## 6.2 PCB 电路板的绘制

我绘制了第一批的电路板 A，并对两块电路板进行了 DRC 检查。然后我在第一批电路版的基础上进行改进，绘制了第二批 PCB 板（如图 6.3~6.6）。

绘制的整体流程如下：

找到或绘制每一个元件的封装→画出原理图→编译，解决错误→导入 PCB 图，制定连线规则→自动或手动连线→运行 DRC 检查→交付厂家

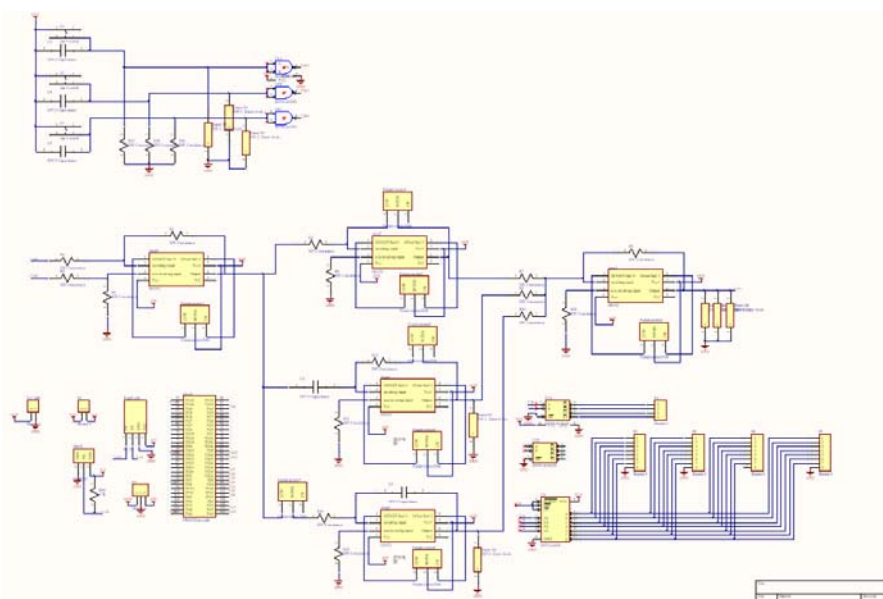


图 6.3 按键、显示、PID 部分原理图



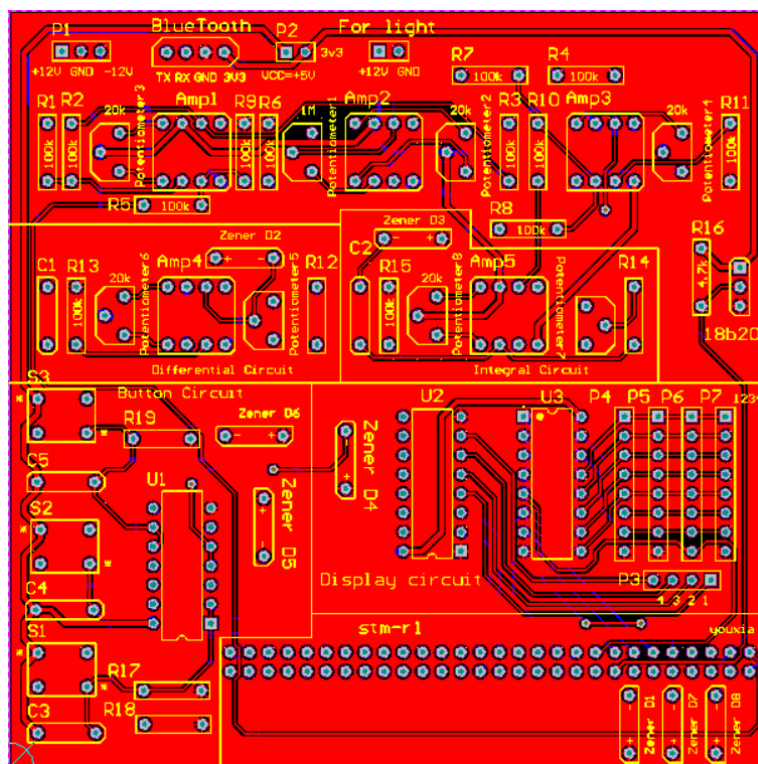


图 6.4 按键、显示、PID 部分 PCB 图

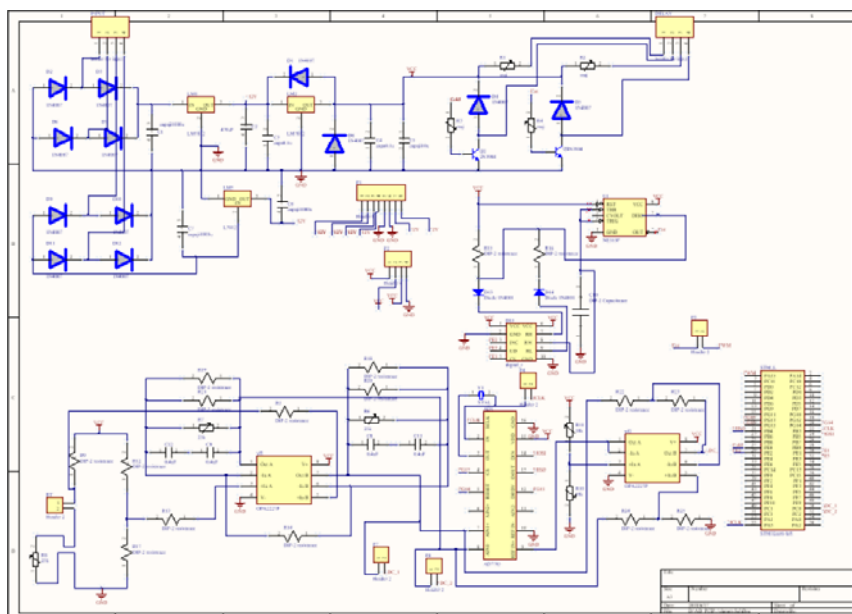


图 6.5 电源、测温、PWM 输出原理图



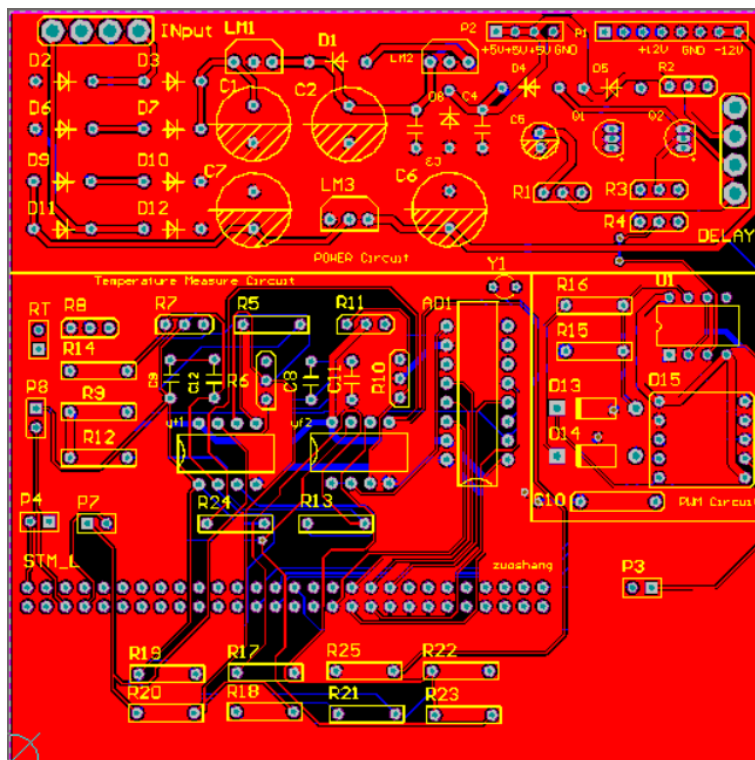


图 6.6 电源、测温、PWM 输出 PCB 图

第一批 PCB 板可以正常工作，除了一些小瑕疵，没有大问题。

第二版相对第一版的改进主要有：

1. 改善二极管元件、电源接口、变阻器等不合理的封装。
2. 纠正 74LS138 的引脚错误，加入更多的文字说明，便于焊接
3. 将测温部分扩展为，既可选择 AD7705 芯片，又可使用单片机双 AD 采样，还能通过一个减法部分变为单片机单 AD 采样，提升电路的拓展性。

在绘制 PCB 的过程中，主要经验如下：

1. 学习软件，最快的方式是找一位学长带着自己做。
2. 画 PCB 要注意丝印层的作用，勤做标记，将各个元件的数值与型号都标注在 pcb 上，这样会给之后的焊接带来很大便利。
3. 一定要提前制定布线规则，提高效率。
4. 面包板验证原理后，应该尽快上 PCB。洞洞板连线时间过长，没有必要。

PCB 板十分稳定，可以很好地排除非电路因素对电路的干扰。

5. 原理图一定不能有错，PCB 一定要使用软件来进行连接检查。

### 6.3 模拟 PID 调节：

我从凌晨四点调节 PID 参数至早上八点，在调节过程中深入理解了 P、I、D 三个参数的意义。经过四个小时的调节，基本可以实现：当设置温度在 65℃至 85℃内时，水箱内部温度可以保持在设定温度上下两度的范围内，至少五分钟。但是早上调完的参数，下午项目检查，发现实际温度比设定温度偏低两至四度左右，说明系统的稳定性较差，PID 还需进一步调节。

PID 调节方法如下：

首先，断开积分和微分电路，只调节比例参数 P。P 值主要调节逼近目标值的速度。经测试，如果打开盖子，满功率加热到设置温度时关闭加热装置，余热会使水箱上升 5℃左右。因此，理想的 P 值应该使水箱在距目标温度 8~10℃左右，开始以非全功率输出，并稳定在某个小于设定值的温度下，没有振荡。此时基本调节完毕。

其次接入积分电路，进行积分参数 I 的调节。积分参数可以看作是一种偏置，因此调节目标是将此时温度的温度抬高至设定温度。这个过程十分有趣，随着温度的升高与时间的推移，比例电路产生的电压逐渐减小，积分电路产生的电压逐渐增大，此时应该调节积分参数 I，使得电压的减小和增大整体达到平衡或略有增加的状态，这样便可以稳定在目标温度。

最后调节微分参数 D。微分参数可以判断误差的变化趋势，起减小超调量的目的。我一开始理解有误，理论上应该在积分电路振荡的基础上，调节比例电路，减小振荡，同时提升系统的稳定性。

关于积分电路的设计：积分电路一定一定要加限幅部分，而且是正负等值的限幅。积分部分的影响除了与积分的速度有关，还与最终的上下限电压有关。如果不加限制，上下幅值是  $U_M = \pm 10V$ ，如果从一个很低的温度到一个很高的温度，那么积分电路会一直充电直到 10V，这与单片机的 ADC 范围 3.3v 相差太远了。这样，当系统超过设定温度时，积分电路需要很长的时间才能放电直到反号，给系统造成极大的惯性。我一开始将稳压管方向接反，虽然运放面临烧毁的危险，但阴差阳错将限幅稳定在了 1V 左右，反而使系统的惯性没那么大。所以对于积分电路的设计，选定滑变、电容和限幅部分是一个相互协调的过程，一定要综合考虑。

再说一下微分电路：微分电路的正负号和比例、积分基本是相反的，起到抑

制超调的作用，减速。所以大胆加就行。具体各方面的功能，还是要结合 [Wikipedia](#) 加深理解。里边有张动图特别好。

模电 PID 还有个很麻烦的地方：积分电路如果要放电，正误差和负误差都需要，但单片机的 ADC 只能采样正电压，负电压需要进行保护。这也是很麻烦的。

## 7. 对于这次项目的经验

面包板只适合原理验证，洞洞板连线过于繁琐，面包板一旦原理验证成功就应该尽快制板，不要在洞洞板上浪费时间。

## 参考文献

- [1] <https://zh.wikipedia.org/wiki/PID%E6%8E%A7%E5%88%B6%E5%99%A8#/media/File:Pid-feedback-nct-int-correct.png> 维基百科 PID 控制器
- [2] DS18B20 中文手册
- [3] <http://www.waveshare.net/study/article-644-1.html> STM32CubeMX 系列教程 5:串行通信(USART)
- [4] X9C104SZ Datasheet
- [5] HC-08 Datasheet
- [6] op07CP Datasheet
- [7] STM32ZET6 Datasheet

## 附件 核心程序源码与 PCB

程序源码过多，PCB 文件较大，均不在文档中列出。如需要，我可以通过邮箱提供。