**目录**

摘 要 1

Abstract 2

第1章 绪论 3

1.1课题研究的背景 3

1.1.1温度控制系统及国内外研究现状 3

1.1.2温度控制系统未来的发展趋势 3

1.2本设计的研究内容及意义 3

1.1.1本设计的研究内容 3

1.1.2本设计的研究意义 4

第2章 温度控制系统的整体方案设计 4

2.1温度系统的设计任务 4

2.2.温度系统的设计要求 5

第3章温度控制系统硬件设计部分 8

3.1总体方案的设计 8

3.2单片机的最小系统模块设计 9

3.2.1时钟系统 10

3.2.2 STM32ADC参考电压基准 11

3.2.3 STM32供电及电压转换电路 12

3.2.4 USART1硬件流烧写和RS-232转TTL电路 12

3.2.5 JLINK/STLINK插座电路 13

3.3温度检测模块的设计 14

3.3.1 PT100铂热电阻温度传感器 14

3.3.2三线制PT100电桥运算放大电路及仿真 15

3.2.3两线制PT100恒流源运算放大电路及仿真 17

3.4电烙铁加热模块的设计 18

3.5 FSMC驱动4.3寸TFTLCD液晶显示及触摸模块设计 19

3.5.1 TFTLCD显示及触摸模块 19

3.5.2 FSMC驱动 23

3.6按键模块设计 24

3.7 WIFI模块设计 24

3.8 24C02外设EEPROM存储模块设计 25

第4章 温度控制系统软件设计部分 26

4.1 主程序的设计 26

4.2状态机与时钟的设计 27

4.3 PWM输出的设计 28

4.4 ADC采集和滤波程序的设计 29

4.5按键调节温度程序设计 31

4.6两分法查表法的设计 31

4.7 TFT显示温度和触摸程序的设计 33

4.8 PID控制算法 35

第5章 实验部分和上位机的数据传输 38

5.1 图腾柱电路的实验 38

5.2上位机和下位机通讯以及JSON格式数据处理 38

5.3上位机整定PID参数 41

第6章 总结和展望 48

6.1总结 48

6.2展望 48

参考文献 49

致谢 50

附录A 51

附录B 53

## 摘 要

温度是工业生产中十分重要的物理信号之一，工业生产中常常需要维持一定范围内的温度。如在热流道温控箱内就使用了温度控制系统，其在工业中使用广泛，在制作金属模具中被大量使用。本文的温度控制系统就是其中一部分。

目前意法半导体的STM32系列单片机十分受欢迎，它不仅是基于功能强大的ARM-Cortex M3内核又是32位单片机。相比较其他同种单片机，其拥有大量USART，定时器和多功能定时器，ADC，IIC,SPI,CAN总线，更拥有多种扩展功能，并拥有极大内存，可以嵌入移植ucos和freeRTOS这类轻量级实时系统，也能配合操作系统移植移植ucgui和emwin等显示屏GUI，使产品更加智能化，适合学习和开发。温度也是自动控制系统中重要的物理量，在工业生产中也是必需检测和控制的物理量。因此，研究此单片机的温度控制系统是十分有意义的。

该系统以STM32F103ZET6单片机为主控芯片，用PT100测温，用PID算法调节PWM占空比从而改变温度。通过液晶触摸控制显示温度。本文主要介绍了以STM32系列单片机STM32F103ZET6、24C02、CH340G、IS62WV51216、LM2596S-5.0、AMS1117M3、LM324、LM358等芯片组成温度检测电路、PWM控制MOS管电路、信号放大电路、FSMC方式外置SRAM驱动TFT液晶电路、电压转换和滤波\去耦电路等。在描述了外围硬件电路同时，本文还做了大量软件工作，包括显示，通讯及通讯协议和PID控制算法等程序。

本设计具有较高的使用价值，采用液晶触摸和WIFI上位机等多种方式控制温度，使控制方式多样化，大大提高了智能化和控制精度。

**关键词**：热流道温控箱； STM32；PID控制；WIFI通讯；上位机控制。

## Abstract

Temperature is one of the most important physical signals in industrial production. It is often necessary to maintain a certain range of temperature in industrial production. For example, the temperature control system is used in the hot runner temperature control box. It is widely used in industry and is widely used in making metal molds. The temperature control system in this paper is part of it.

At present, STM32 series singlechip is very popular. It is not only based on the powerful ARM-Cortex M3 kernel, but also the 32 bit MCU. Compared to other homologous singlechip, it has a large number of USART, timers and multi-functional timers, ADC, IIC, SPI, CAN bus, has a lot of extended functions, and has great memory, can be embedded in the portable, UCOS and freeRTOS such lightweight systems, and can also transplant ucGUI and EMWIN, such as ucGUI and EMWIN, and GUI, so that The product is more intelligent and suitable for learning and development. It is also the physical quantity that must be detected and controlled in industrial production. Therefore, it is very meaningful to study the temperature control system of this microcontroller.

The system takes STM32F103ZET6 microcontroller as the main chip, uses PT100 to measure temperature, and uses PID algorithm to adjust the duty cycle of PWM to change the temperature. The display temperature is controlled by the liquid crystal touch. This paper mainly introduces the temperature detection circuit composed of STM32 Series MCU STM32F103ZET6, 24C02, CH340G,, IS62WV51216, LM2596S-5.0, AMS1117M3, LM324, LM358 and more. The PWM control MOS tube circuit, the signal amplifying circuit, the external driving liquid crystal circuit, the voltage conversion and the filter \ decoupling power are introduced. Road and so on. In describing the peripheral hardware circuit, this paper also does a lot of software work, including display, communication and communication protocol and PID control algorithm.

This design is of high use value, and adopts many ways to control the temperature, so that the control mode is diversified, and the intelligence and control precision are greatly improved.

Key words: hot runner temperature control box, STM32, PID control, upper computer control, WiFi communication

1. **绪论**

如今全球化的生产中，经常要求需要检测控制的物理参数如下几种：温度、速度、电磁等。例如人们都需要在冶金工业、农业生产、食品生产等众多领域中，检测和控制生产环节中的温度。选用，STM32单片机来对温度进行控制，具有扩展性高，能嵌入实时操作系统，拥有iic，can，spi总线等优势。控制方式也多样化。从而各类生产能供正常运行和工作，生产效率得到提升。由于工业领域对于控制温度是无法回避的，所以研究温度控制系统是十分有必要的。

**1.1课题研究的背景**

**1.1.1温度控制系统及国内外研究现状**

近些年温度控制器发展迅速，一些大型企业也参与到温度控制器的研发当中，如：西门子、富士、欧姆龙等等。我国也有很多中小企业和智能家居企业在不断发展温度控制器，带来了良好的研究成果。我国温度控制器在参数自整定等方面也做出了许多突破，但还是主要还是通过价格优势确立了自己的竞争地位。

国外也有很多企业在温度控制器产品中做出了重大突破，如Nest等公司使用WIFI来远程控制温度控制器，并且数据在其他同型号温度控制器互通。根据用户自己拟定调节温度，使温度控制更加智能化。这种控制方式在智能家具中被广泛运用。

**1.1.2温度控制系统未来的发展趋势**

温度控制器运用将更加广泛。在未来，无论是工业还是生活都少不了温度控制。其拥有巨大的商业价值。温度控制系统未来将越来越精准化，快速化，智能化。通俗来讲，温度控制器的算法将会进一步改进，调节温度的速度变得越来越快，控制精度也会越来越高，控制方式会变得多样化。随着物联网的越来越普及，通过上位机控制会变成主流，同时上位机编写的软件也能满足人们对智能控制的要求。

**1.2本设计的研究内容及意义**

**1.1.1本设计的研究内容**

在智能化的时代，温度控制对自动化有着重要的意义。从生产到生活的方方面面都离不开温度控制器。目前性价比最高的控制方式就是使用PID方式控制温度。

本设计是基于STM32单片机温度PID控制系统。该系统被广泛运用于工业生产控制中。用于检测控制高温环境。本实验使用电烙铁产生高温温度，用温度控制器检测并在LCD上显示并控制其温度。可以通过wifi网络进行远程控制温度和PID参数，并显示温度曲线。

**1.1.2本设计的研究意义**

本次的毕业设计选题意义不仅在于让学习自动控制的我们，通过实验和设计方案再到论证实验，在整个过程中不断学习，完善大学中我们所学习到的知识，运用到实际生活和工作中去。而且还在之上改进了控制方法。并且通过对温度控制器的原理图设计、PCB制作，焊接和调试地整个过程，让我对一些实用模拟电路有了一定了解，并掌握了一些模块和专用芯片的驱动电路。为以后的学习、工作和研发打下基础。同时也提高了我的查阅资料和自主学习的能力。而且温度控制器在智能家居和工业控制中十分常见，其中最广泛使用的是PID控制器，所以研究可远程控制监控的PID控制器也是十分实用的。

1. **温度控制系统的整体方案设计**

## 2.1温度系统的设计任务

方案一：使用STM32F103ZET6单片机作为本温度控制系统的主控芯片，拥有外部基准电压引脚VREF+，可以接入更稳定的电压基准。利用PT100热电阻温度变化引起阻值改变，经恒流源转化成电压信号产生压差信号，再经电路放大成STM32内部ADC可识别的电压信号，最后再由STM32进行采集并查表计算输出温度值到液晶和上位机进行显示,同时上位机在与WIFI模块相连接时可发送设置温度，PID的设置参数到STM32的串口，从而实现远程实时调控。

方案二：使用STM32F103CBT6单片机作为本温度控制系统的主控芯片，利用热电偶电阻温度变化引起阻值改变，阻值的变化经惠斯通电桥转化为压差信号，在经过放大电路放大成STM32内部ADC可识别的电压信号，最后STM32采集过后计算温度值到LCD，通过232接口通过上位机显示并控制温度。

方案三：使用STC12C5A,或STC15系列等8位的单片机，使用PT100测温电路，外部ADC芯片转换模拟信号，再通过按键控制温度，OLED显示温度，通过串口或WIFI模块和上位机通讯，上位机显示并控制温度

方案一的stm32f103芯片拥有较大内存，可以有较大空间移植小型实时操作系统和LCDGUI。方便后期地再开发，其拥有较多引脚，可以使用FSMC驱动LCD，驱动方式更加简便，利用恒流源PT100电路会很容易就可以得到电压-电阻-温度的关系，方便程序转化，而通过USART转WIFI模块，通过WIFI设置TCP传输模式更加便捷，安全。但是成本比较大，芯片价格高出两倍，并且usart转WIFI模块比较大，会使实物的体积比较大。方案二的STM32F103CBT6单片机引脚较少，没有外部VREF+引脚，只能使用内部ADC基准，或多使用一路ADC接电压基准作为参照基准。并且只能通过软件实现80并口驱动TFTLCD，程序繁琐并不利节省内存，对于移植实时系统和GUI有一定影响。使用热电偶方式需要校验冷端，在不同室温时会造成误差，通过232和上位机通讯不适合在比较远或工业的环境下进行，一般需要采用485通讯并编写MODBUS通讯协议。但CBT6的芯片价格和8位机差不多，制作成本会比较低。

方案三使用8位单片机的成本会比较低，但是其需要多一路ADC作为基准电压，并且没有内部ADC模块，需要外接ADC芯片，并且不适合驱动多引脚，驱动比较复杂的TFT，而且虽然stc有一些IAR硬件调试工具，但是并没有STM32的JLINK之类的仿真器好用，所以程序调试时会出现一些麻烦。所以可以采用IIC通讯方式的OLED，只要两跟数据总线，所以整体实物体积也可以做得比较小。

综上所述，本此设计不需要考虑成本和体积，选择使用可以方便驱动TFTLCD的stm32f103zet6的芯片，可以方便后期移植实时系统和GUI，并可以有CAN，SPI，485，以太网口等传输接口为后续开发通讯提供方案。可以选择PT100恒流源电路测温，可以容易得到电压和电阻的关系。使用WIFI通讯，WIFI模块中设置TCP透明传输，实现和上位机通讯，经过上位机后续开发，可以使控制方式更加多样化。

**2.2.温度系统的设计要求**

1)可以对电烙铁温度进行实时检测并把结果显示在液晶和上位机上。

2)能够在0-250度之间通过手动（按键或触摸）或上位机设置电烙铁温度。

3）整个系统能相对保持电烙铁温度恒温，误差在正负2度之内。

4）能够在上位机上看到温度曲线，并能控制PID参数和温度数值

5）TFT液晶能触摸控制，并显示各项温度和控制参数。

1. **温度控制系统硬件设计部分**

### 3.1总体方案的设计

该系统主要由单片机时钟电路、单片机主控电路、PT100传感器测温电路、TFT液晶显示电路，按键电路、串口电路、WIFI模块电路、外接EEPROM电路、电压转换以及电源电路，PWM输出电路组成。采用STM32F103ZET6为主控芯片。首先以恒流源将PT100的阻值变化转换为电压变化，输出电压，通过运算放大器放大，发送给单片机的内部ADC，并和存储在EEPROM中的PT100电阻-温度对应数据进行查表操作算出PT00具体温度，并由单片机将数据进行处理后发送到串口，并通过液晶显示温度，再通过键盘、液晶触摸、上位机输入信号给单片机，通过PID算法调节PWM波占空比，控制MOS管的通断，进而控制电烙铁的工作来实现温度的控制。

### 3.2单片机的最小系统模块设计

本文使用STM32F103ZET6，使用144LQFP封装，其单片机主控电路如图3-1：

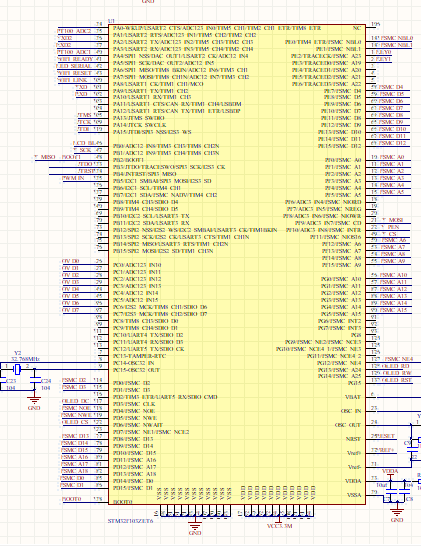


图3-1 单片机主控电路

在每个VCC3.3和GND之间需要加0.1uf电容去除耦合。电容具有滤波作用，对于交流电有一定阻抗，阻抗。

Pi为圆周率,F为交流电的频率，c为电容容量。由公式可得电容容量越大，阻抗越高。但在实际电路中c越大，阻抗反而会增大,经过人们实践得出结论，104电容对50HZ的高频率的电源信号具有滤波的作用。在绘制PCB图时也要每个VCC和地的引脚接上104电容，产生去耦的作用。

#### 3.2.1时钟系统

STM32F103ZET6拥有5个时钟源，其中有由OSC\_OUT、OSC\_IN引脚上的晶振产生，的高速外部时钟HSE和由OSC32\_OUT、OSC32\_IN低速外部时钟LSE，还有由芯片内置RC振荡电路构成的RC振荡器产生的高速内部时钟HIS，和低速内部时钟LST。还有给高速USB提供时钟的锁相环倍频输出PLL,其时钟源为HSE/2或HIS/2,这些时钟通过配置STM32内部的预分频器和选择器将时钟分为不同频率给单片机的系统时钟SYSCLK、实时RTCCLK、独立看门狗时钟IWDGCLLK、高速USB和外设等提供时钟频率。本文主要使用外部晶振给单片机提供时钟频率。单片机时钟电路如图所示：

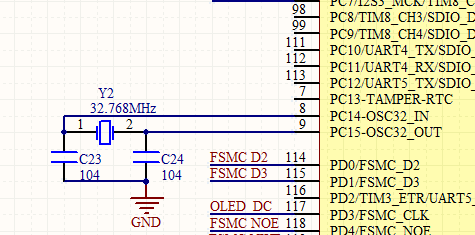


图3-2 LSE时钟源

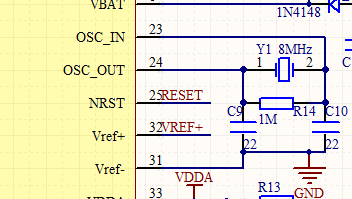


图3-3 HSE时钟源

#### 3.2.2 STM32ADC参考电压基准

STM32F103ZET6自带内部12位ADC，即2^12=4096，拥有比较高的精度。其VDDA和VSSA是其ADC的电源正极和负极，VREF+和VREF-是STM32提供的外部参照基准电压。基准电压用于给其他路ADC采集的数据提供参照。其他路ADC采集的数据以此路为基准，并加以修正。如基准电压为2.5V，对应数字量为1000，一路ADC采集的数字为2000，则这路ADC对应电压为（2.5V/1000）\*2000,这样可以消除采样硬件产生的误差，达到更高的精度。在没有VREF+和VREF-的单片机封装中则需要多用一路ADC采集基准电压或使用内部VREF基准。由此可以看出STM32F103ZET6功能强大，既不需要外部ADC芯片又可以拥有较高精度，也能为开发者剩下一路ADC I\O作为ADC基准。另外，在实际设计中单片机供电的电压不能直接用作ADC的基准电压，因为其虽然是直流电，但是并不够稳定和精准。为确保采样精度，在本设计中采用REF2930AIDBZR基准电压芯片产生稳定3V电压为单片机和ADC电路提供基准电压。查阅其DataSheet，其经典应用接法如图所示：

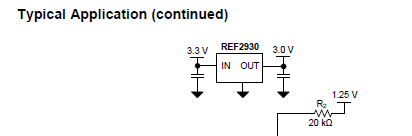


图3-4 REF2930经典应用

本设计中参考经典应用的接法设计ADC电路的基准电压和参考电压。如图3-6：

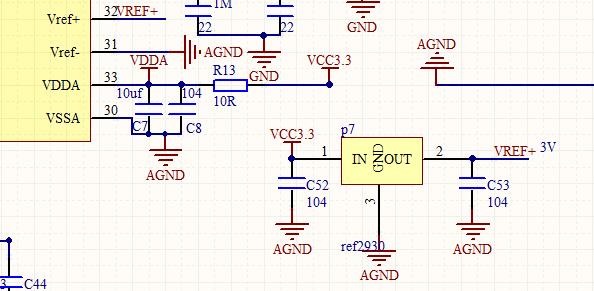


图3-5 STM32ADC供电及参考基准

在本设计中有模拟电路部分和数字电路部分，为了使两个电路互相的干扰减小，采用模拟地和数字地分开，最终以单根导线连接磁珠或0Ω电阻的方式连接模拟地和数字地，模拟地和数字地分区域独立覆铜。覆铜优先考虑布通后考虑EMC性能。

#### 3.2.3 STM32供电及电压转换电路

STM32采用3.3V供电，但本设计中有部分电路需要用5V和24V电压。所以需要电压转换电路。本设计采用24V外部开关电源提供直流电经两个104电容滤除高频信号和2个100uf电解电容滤除低频率信号后经LM2596S-5.0 DC-DC调压芯片转换电压为5V，查阅其使用手册，其5V输出经典接法为图3-6所示：

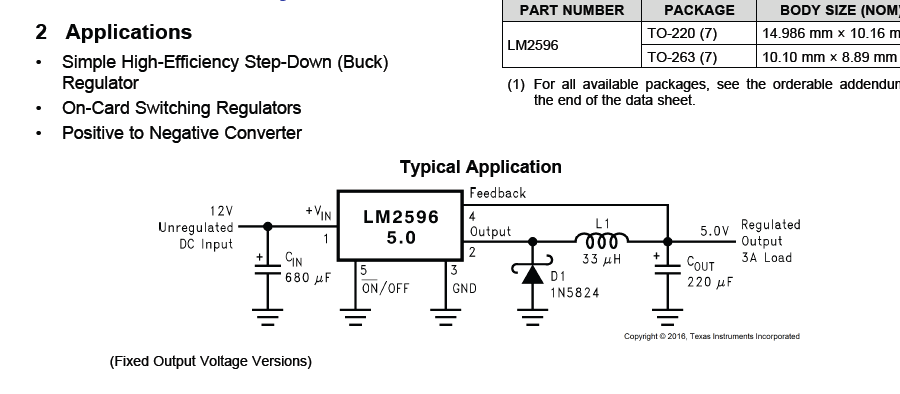


图3-6 LM2596S-5.0经典应用

在本设计中，L1为33UH电感，D12为肖特基二极管1N5822其最大正向连续电流为3A，反向耐压40V。在外部24V直流电通滤波后进入VI,LM2596内部开关导通，24V电源通过LM2596引脚2给电感L1 充能并给负载供电，L1此时储能,D12截止；当LM2596内部开关截止时，引脚2不再输出12V, D12变成导通状态，给L1提供了回路，L1处于放电状态，L1维持负载电流。此时此电位通过反馈回路通过LM2596引脚4，与内部基准电压比较，低于基准电压时，引脚2再次导通，接着L1再次储能，循环反复，就能持续将24V电压转化成5V。接着还要通过AMS1117将5V电压转换成3V电压，经典接法如图：

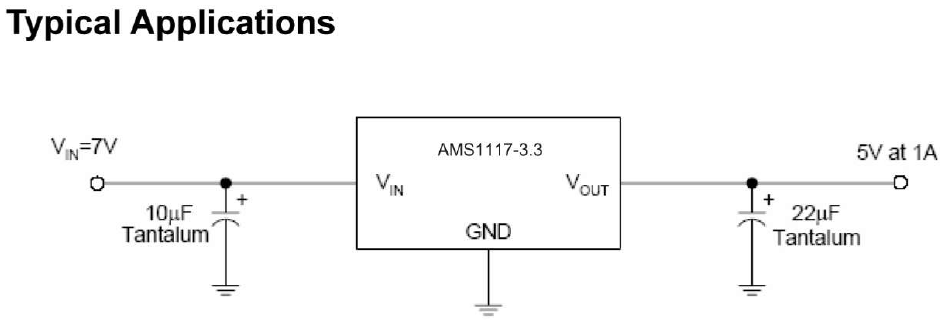


图3-7 AMS1117 典型应用

在本设计中使用104电容和10uf坦电容滤波5V，经过AMS1117M3芯片转换成3.3V后再由22uf电解电容滤波后产生3.3V电压给单片机和其他负载供电。本文电源电路如下：

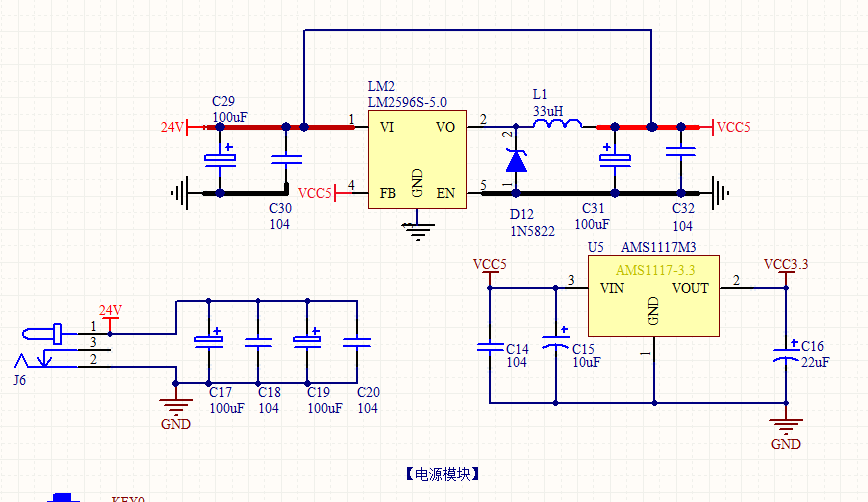


图3-8 STM32电源及电源转换部分

#### 3.2.4USART1硬件流烧写和RS-232转TTL电路

本设计使用CH340G转换芯片将TTL电平转化为232信号。如图：

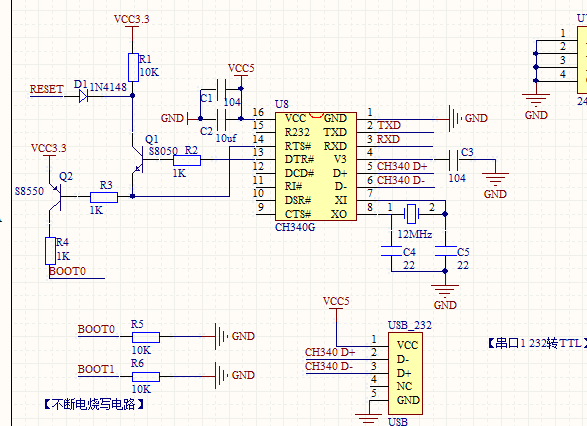


图3-9 硬件流电路和串口电平转换电路

本设计使用CH340G转换电平。STM32单片机的I\O引脚有TTL和CMOS两种，在大部分单片机烧写引脚都是TTL电平，而和PC虚拟串口的一般的通讯方式是RS-232通讯，这就需要转换芯片来使两者能进行通讯，本文使用CH340G转换电平信号。

本设计采用5V给CH340G供电，X0、XI提供时钟信号，TXD和RXD接入单片机的TTL信号。D+,D-连接USB总线，本设计采用ISP下载器下载。在STM32F103ZET6中有BOOT0和BOOT1(PB2)引脚可以选择启动模式。当单片机复位启动时，系统时钟SYSCLK开始工作，SYSCLK脉冲的第四个上升沿时，STM32的BOOT引脚就会被锁存。当BOOT0为0时，单片机执行主闪存区启动方式，即执行单片机FLASH中的程序代码。当BOOT0为1，BOOT1为0，单片机执行系统存储器启动方式。即为烧写.HEX文件进入单片机。

在没有电路引导的方式下，需要手动断电再配置BOOT0=1,BOOT1=0的方式上电电才能使烧写完成，然后再断电以BOOT0=0的方式启动才能运行烧写好的程序。如图3-12所示：

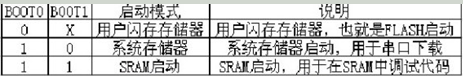
[](http://photo.blog.sina.com.cn/showpic.html#blogid=&url=http://album.sina.com.cn/pic/005FcO97gy6UpMpnB1S20)

图3-10 STM32启动模式

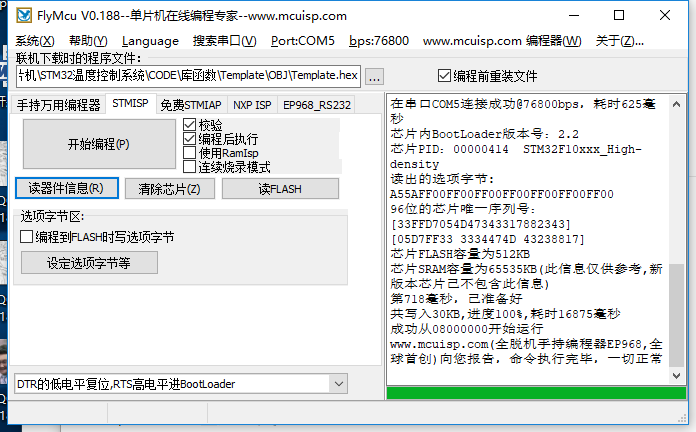
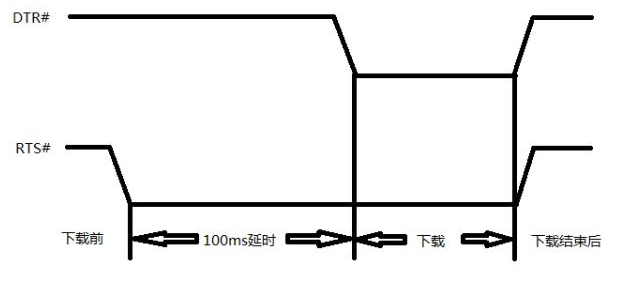


图3-11 FLYMCU 下载器

但是在图3-8的电路和图3-10 FLYMCU ISP下载软件设计的工作下，可以进行不断电烧写。在主FLASH启动方式中，虽然主FLASH区的内容从主FLASH存储器被映射到了启动空间地址0x00000000，但是还是能通过指针访问FLASH从地址0x08000000访问FALSH内容，可以通过软件和硬件配设达到指针能访问0x08000000地址就可以启动烧写程序了。在FLYMCU ISP软件烧写时选择“DTR低电平复位，RTS高电平进BootLoader”时实际情况与描述相反，所以在本设计中，上电时两者都为高，软件RTS置高等于RTS电平变低，这样Q2基极为0，发射极为3.3V，有正向电压，Q2被导通，BOOT0=1。软件DTR置低，DTR变高电平，基级为高电平，同时Q1发射极为0，基极为0，Q1也会导通，VCC为3.3高电平，所以RESET为低，单片机复位。延时100ms后，FLYMCU控制控制RTS置高，RTS还是低电平，此时BOOT0=1。DTR为高电平，则DTR实际被置低Q1截止，此时RESET被置高，所以此时单片机复位结束，进入下载。下载结束后，软件再次置位RTS高，DTR低。RTS被置低,DTR被置高,Q1,Q2都不通，BOOT0=0，RESET拉高;程序指针指向主FLASH区域。

[](http://photo.blog.sina.com.cn/showpic.html#blogid=&url=http://album.sina.com.cn/pic/005FcO97gy6UpNeW0ZO28)

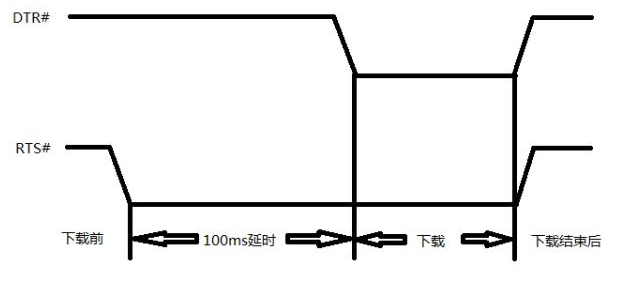
[](http://photo.blog.sina.com.cn/showpic.html#blogid=&url=http://album.sina.com.cn/pic/005FcO97gy6UpNeW0ZO28)

图3-12 软件烧写时序

#### 3.2.5 JLINK/STLINK插座电路

本文需要用到SLINK仿真调试和下载程序STAG电路如图3-13

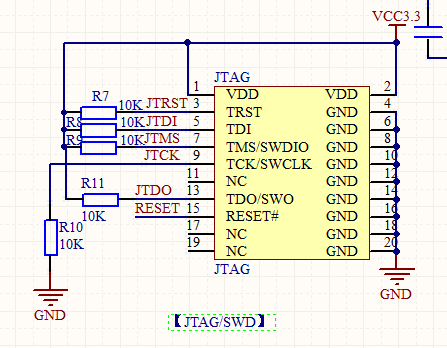


图3-13 JTAG/SWD接口

本设计需要用到JLINK/STLINK外部仿真器，通过接入JTAG/SWD接口它可以识别STM32的内部存储器，同过特殊协议过对FALSH进行烧写，其烧写速度一般在2M左右，普通ISP下载在115200，0.12M左右，速度十分快。更重要的是他能通过连接芯片进行硬件仿真，直接在跑的程序上设置断点进行调试，这比一般的软件仿真要直观很多，这大大提高了调试程序的效率。在含有DEBUG接口模块的CPU中，有专门给仿真器预留的引脚和移位寄存器单元。移位寄存器可以隔离输入输出，仿真器通过扫描移位寄存器，可以进行观察和控制输入输出。仿真器通过引脚，可以在芯片内创建TAP测试访问节点，通过TAP控制器可以访问单片机芯片所有数据寄存器和指令寄存器，调试程序时，就可以通过TAP口对数据、指令寄存器进行访问。所以,仿真器在调试程序的时候会有很大作用，本文使用了ST-LINK仿真，ST-LINK是意法半导体出的仿真器，它能仿真大部分STM32/stm8单片机，而J-LINK,U-LINK之类的单片机能仿真ARM之类更多的单片机，但ST-LINK价格稍微便宜，性价比更高。

### 3.3温度检测模块的设计

#### 3.3.1 PT100铂热电阻温度传感器

传感器是将外界物理量经过一定规律变化成为可用的电信号的器件。利用传感器可以检测很多物理信息，有了传感器，人们可以对外界信号进行检测并加以调控。这使得自动控制得以实现，也是反馈控制的基础。

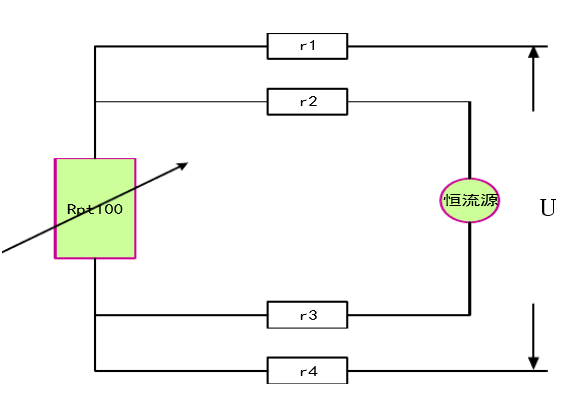
常见的温度传感器有热电偶和热电阻。热电偶通过冷端热端的热电效应产生压差。而热电阻可以看成是阻值随着温度变化的一种电阻。本文使用的PT100就是一种热电阻。

PT100铂热电阻拥有电阻和温度值变化的关系，基于这一特性，人们研发出PT100温度传感器将温度这一物理量变为电信号。PT100传感器是电阻式温度传感器，由PT(铂)制成，其在0度时阻值为100欧姆。其在温度测算方面运用十分广泛。

图3-14 PT100封装

PT100测量范围为-258°C~900°C，其反应速度快，测量温度范围广，所以比一般的数字仪表如DS18B20更适合使用在温度变化快和温度范围广的环境下。一般低于600°C的温度范围，PT100是第一选择。PT100温度和电阻近似正比，但事实上来说是曲线的形式，所以在确定温度和电阻的关系时需要使用查表法，找到每1°的温度和电阻的关系，建立表格，在1°范围内默认为线性，最终确定电阻阻值。

PT100有四线制、三线制、两线制。三线制PT100和四线制PT100可以有1效消除导线电阻对实验测量的误差。如图3-15



：

图3-15 PT100四线制补偿原理

在四线制PT100接线中，有两条导线I+,I-接到恒流源给PT100施加励磁电流I，测得R1,R4电路的电压V1,V4。由于R1,R4所在电路输入阻抗高，其所在电路虚短，I1=I4=0,忽略电阻影响。所以

Rpt100=。

同理，三线制PT100同样可以消除导线误差，一般用于电桥电路中。在工业中，一般采用四线制、三线制PT100补偿导线误差。在本设计中由于导线较短，采用了三线制和两线制两种接法。

#### 3.3.2三线制PT100电桥运算放大电路及仿真

图3-16 桥式三线制PT100电路

此电路将PT100作为电桥桥臂接入电桥。将PT100和电桥电阻的压差送入差分放大器放大20倍后送入单片机ADCIO，电桥供电REF2930基准电压芯片产生3V稳压。此电路可以通过VR2变阻器取得零点温度，如在变阻器100Ω时，PT100零点温度被设置为0°。在变阻器在112.06Ω时，零点被设在30°。在设置变阻器时，应当在没有接入电路时用万用表调节，当变阻器放入电路，这是测量的阻值会与实际有误差。此电路理论上运放输出电压为输入压差放大倍数，但实际上压差信号会比理论值小很多，实际输出压差为,

LM324组成差动放大器R25=R26,R27=R31。所以放大倍数为R32/R26=20倍。

Vout放大20倍，得出ADC采集电压值。由此公式可知RPT100的值。使用Multisim仿真软件仿真结果如下：

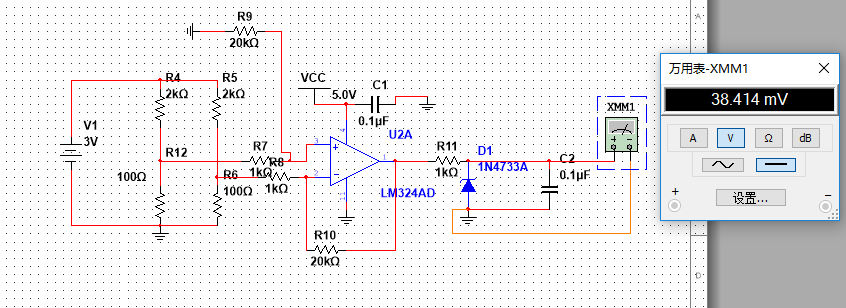


图3-17 桥式三线制PT100 0°C时仿真

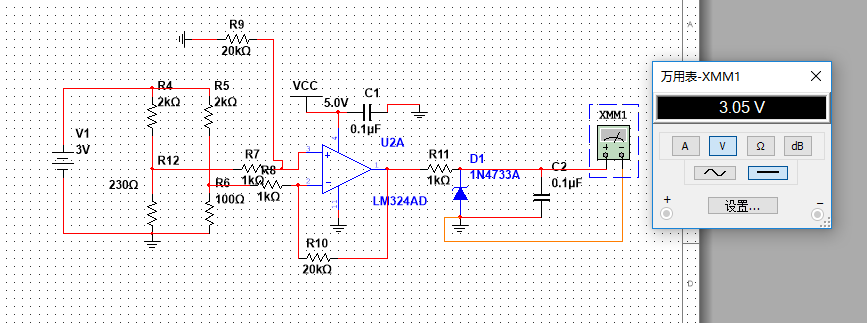


图3-18 桥式三线制PT100 350°C时仿真

STM32F103单片机的IO口大部分耐压3.3V，此电路送出的电压可以接在STM32的ADCIO上。

#### 3.2.3两线制PT100恒流源运算放大电路及仿真

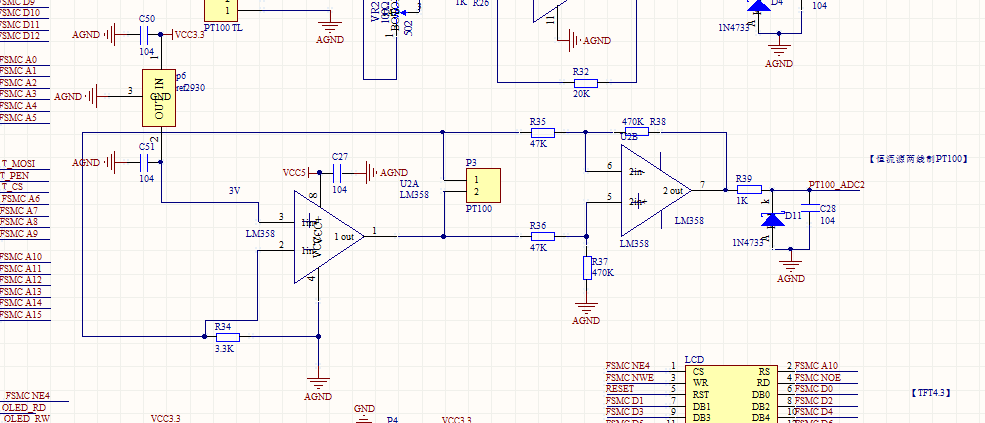


图3-19 恒流源两线制PT100电路

此电路由REF2930产生稳压3V供电，并用LM358运放芯片的一个运放U1A放大电流构成PT100的横流源电路，此时根据U1A2、3虚短分析，V-=V+=3V,假设运放输出端1对地电压为，根据虚断分析，i-= 0A,

根据KVL，,

求得,

此电路中R34=3.3K，V-为3V可算出电流值约为0.909mA。这个电路可以看作由一个0.909mA恒流源流过PT100的电路。再将PT100的电压由运放B放大10倍送入单片机的ADCIO口。此电路可以容易得到电压-温度的输入输出关系函数。Multisim电路仿真如下：

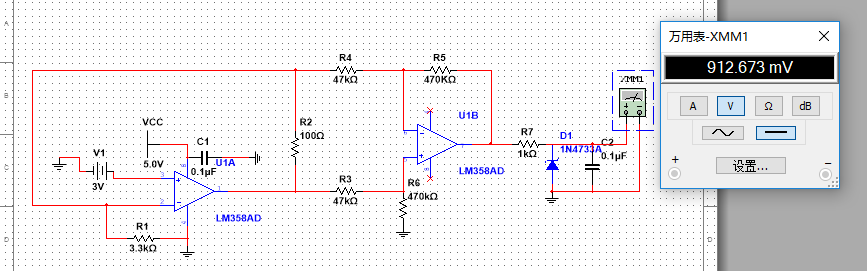


图3-20 恒流源两线制PT100 0°C时仿真

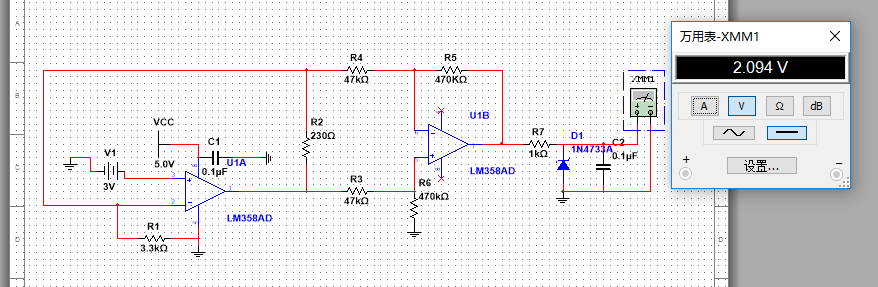


图3-21 恒流源两线制PT100 350°C时仿真

### 3.4电烙铁加热模块的设计

本设计使用35W24V电烙铁作为加热源，使用PWM驱动nmos管栅极方法。由通断频率较高,则使用一般光耦驱动方法不可靠。由于本文的用3.3VPWM波控制24V的电路，对MOS管的门极驱动要求比较高，一般的MOS管驱动芯片无法使MOS管良好驱动，网络上一般驱动MOS的方案是传统的图腾柱电路，但是在图腾柱电路使用三级管就一定会在be有0.7V左右的压降，在低压PWM驱动时导致最终实际在MOS管GATE上电压只有会减少0.7左右导致驱动不良，而在宽电压PWM驱动时，PWM电压随着占空比改变而变化。而为了让MOS管在高gate电压下安全，很多MOS管都内置了稳压管限制gate电压，如果用简单的电阻分压式分压MOS管gate,会使高压PWM下，mos管驱动良好，低压下，mos管驱动不良的情况出现。本文MOS管低压测和高压测压差较大，需要推挽电路来实现PWM驱动MOS导通。于是采用反馈式图腾柱电路来驱动MOS管：

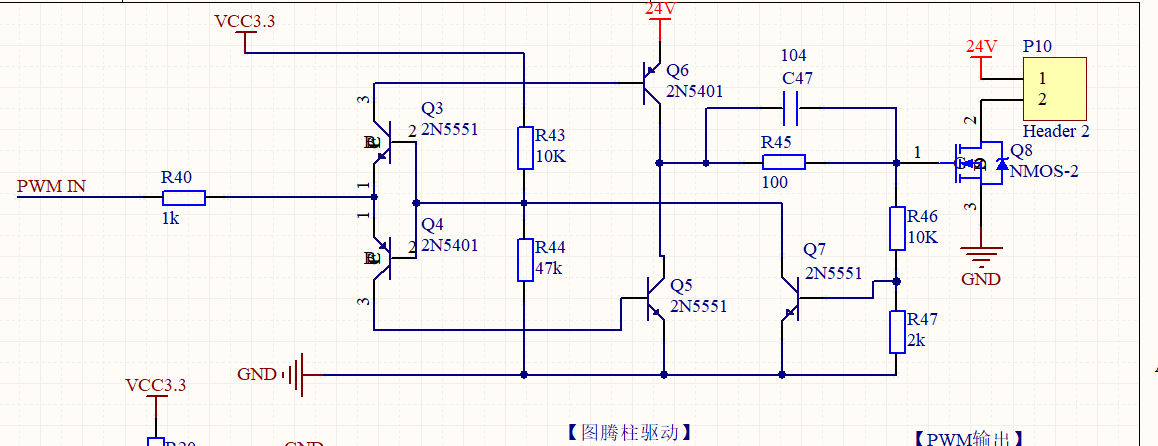
****

图3-22 MOS管图腾柱电路驱动

Q3Q4为NPN和PNP二极管组成的反置图腾柱实现隔离低压高压，并使Q5Q6两个NPN和PNP驱动二极管不会同时导通，R43和R44改变了PWM的基准电压，通过改变这两个电阻可以调整PWM电压，并使其稳定为方波。Q5Q6有推挽作用，为mos管gate提供驱动电流，而且相对压降较小，R46和R47是反馈电阻，对mos管gate进行采样，通过Q7对Q3,Q4的基级产生强烈负反馈，从而使Q5Q6提供的驱动电流产生影响，形成反馈回路。而R40和R45分别为Q5Q6基级限制电压和MOS管GATE限制电压。C47为加速电容，可以使MOS管开关的频率加快。此电路解决了上述mos管在低压驱动高压时产生的各种问题。 但其中电阻还要细微调整才能产生良好的波形，需要现在multisim中进行仿真后再在洞洞板上搭电路用示波器进行观察调整。Multisim仿真结果如下:

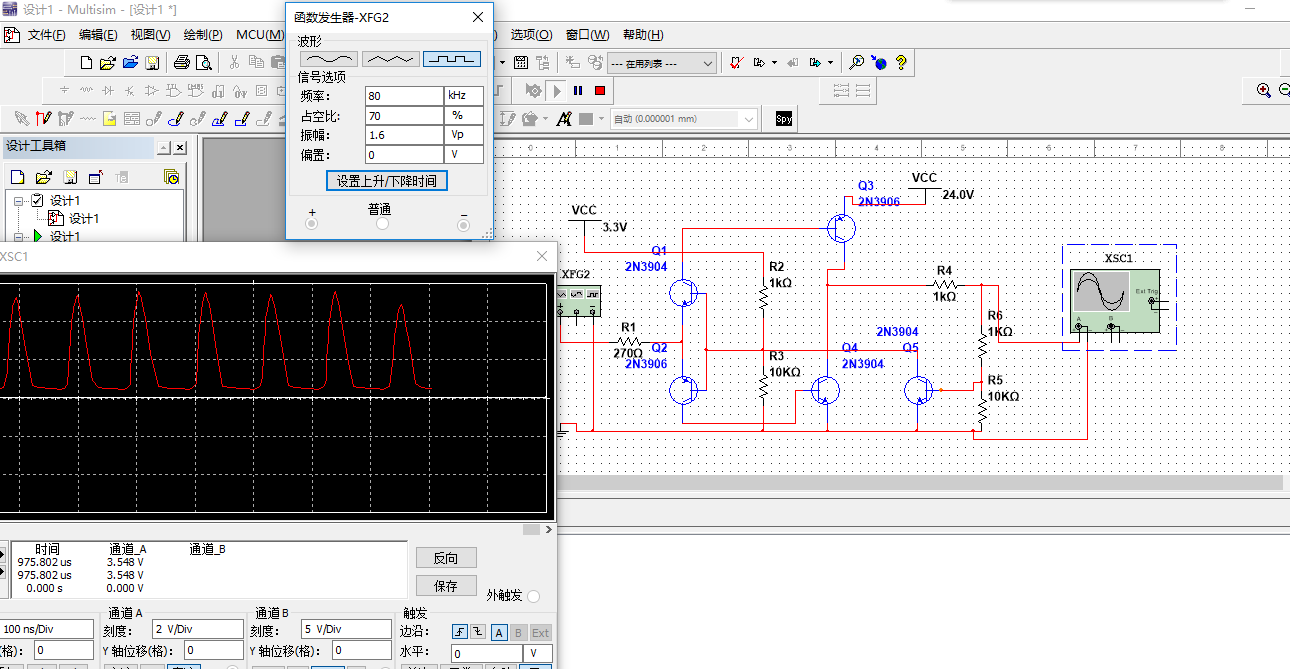
****

图3-23 multisim仿真图腾柱电路

通过函数发生器产生80KHZ频率1.6V方波波形模拟70占空比PWM波形，此时MOS管GATE会有一个尖波产生，说明这个电路是可行的。

### 3.5 FSMC驱动4.3寸TFTLCD液晶显示及触摸模块设计

#### 3.5.1 TFTLCD显示及触摸模块

**①液晶显示**

本文使用TFT电容触摸液晶显示及控制温度，即薄膜晶体管液晶显示器。它在液晶显示屏的每一个象素上都设置有一，个薄膜晶体管（TFT），可有效地克服非选通时的串扰，使显示液晶屏的静态特性与扫描线数无关，因此大大提高了图像质量。

TFT液晶触摸显示插座如下图：

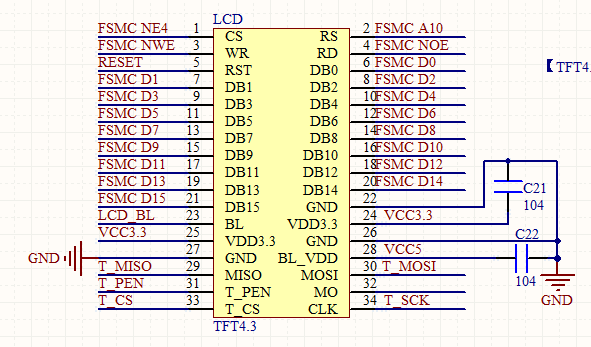
****

图3-24 4.3TFT液晶插座

LCD\_CS为片选端，WR写，RD读，DB0-15为16位双向数据线。LCD\_RST为硬件复位LCD信号。RS为命令数据标志，1为数据，0命令。BL\_CTR为背光控制信号。T\_MISO/T\_MOSI/T\_PEN/T\_CS/T\_CLK是触摸屏接口信号。

本设计4.3寸ALINETEK液晶由NT35510控制芯片+SRAM驱动，采用16位80并口驱动。查阅手册，如下为NT35510的时序：

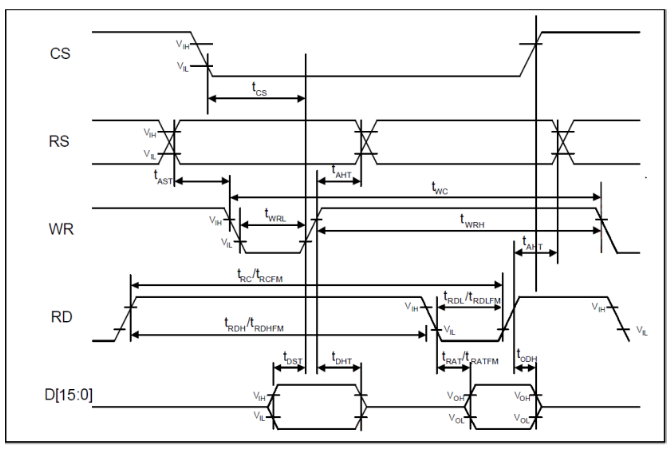
****

图3-25 NT35510时序

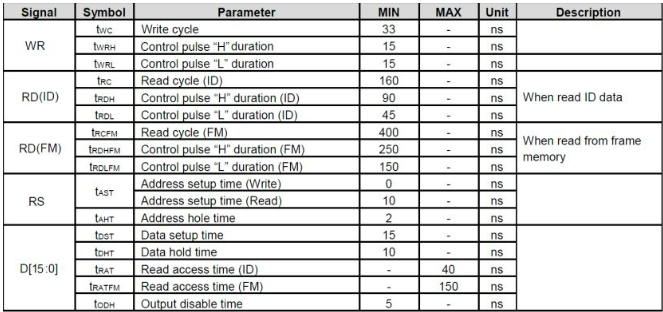
****

图3-26 NT35510时序中电平持续时间

Trdl和trdh分别是读ID低和高电平脉宽,其最短读取时间为45ns和90ns。Trdlfm和trdhfm为读FM低和高电平脉宽，最短读取时间为150ns和250ns。Twrl和twrh是写控制电平低和高脉宽。最短读取时间都为15ns。在本文中其读写时序通过FSMC驱动（如FSMC\_WE控制NT35510,FSMC\_OE控制RD时序），其时序是FSMC模式A读写时序的依据。

通过控制NT35510的时序控制，可以驱动读取LCD控制器ID，以及操作寄存器，写入命令数据等多种功能。

**②触摸控制**

触摸屏幕本质与TFT液晶分离，其一般覆盖在液晶屏上，本文使用电容式触摸屏。电容触摸屏幕通过人体热感应区来触摸。其反应较快，支持多点触摸，触摸感应较好。电容触摸屏幕有表明电容和投射式电容屏。表面电容屏原理为电场感应一般只能识别一个触摸点的操作。而投射式是传感器传感静电场线，其存在自我电容式和交互电容式。

本设计使用投射式电容触摸屏（交互电容式触摸屏），采用纵向和横向两列电极组成感应矩阵，来触摸感应，形成X轴和Y轴产生的交叉网格，通过这种方法来感应单元电容的变化。本文使用的4.3寸TFT控制IC为GT9147，它与STM32采用IIC通讯。

在本文中使用IO模拟IIC。其中SCL为时钟线，SDA为数据总线，可发送和接受，是半双工通讯方式。其工作方式需符合IIC协议。IIC协议为5个步骤，分别为空闲状态，开始信号，数据发送，应答信号校验，停止信号。通过SCL时钟线和SDA数据线完成这5个步骤，如下为时序：

1. 当发送前,处于空闲状态，IIC总线上无通讯，SCL和SDA两跟信号线被上拉为高电平。
2. 启动信号即开始发送的信号：当SCL处于高电平，SDA电平由高变低时，完成开始发送信号。
3. 当启动信号完成后，SCL开始产生时钟脉冲，在一个脉冲周期内电平完成从低到高再到低的过程，SCL处于高电平时，SDA数据线上的电平信号就会被传输。数据传输时一定要做到稳定，即在开始发送信号完成后，SCL一旦处于高电平，SDA的电平就要稳定不变。（在SCL进入高电平前就将SDA数据信号准备好）
4. 应答信号，当发送器发送一个字节（8个二进制数），就会在时钟脉冲的第9个期反馈一个应答信号。若此信号为低电平，被称为应答信号（ACK）,此时表明接受设备成功接受这一个字节。若此信号为高电平，被称为非应答信号（NACK）,此时表明接受设备没有接受到这一个字节。
5. 停止信号即停止发送的信号：当SCL处于高电平，SDA电平由低变高时，完成停止信号。

时序图如下**：**

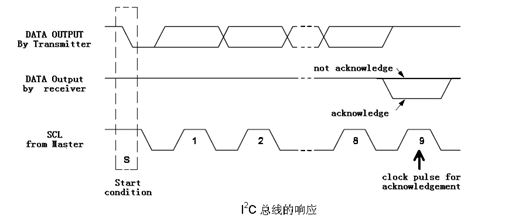
****

图3-27 IIC通讯协议发送一个字节

在发送所有字节结束后再发送一个停止信号，代表一次数据发送完成了。

触摸屏与STM32的SDA,SCL,RST和INT引脚连接。控制器GT9146可以通过RST复位，当复位结束后的5ms,INT保持高电平，此时GT9146的IIC地址为0X14,否者为0X5D。所以GT9146拥有两个地址。本文使用0X14作为器件地址。读命令为0X29,写为0X28。在本文硬件连接中，T\_PEN为INT,T\_CS为RST,T\_CLK为SCL,T\_MOSI为SDA。

通过IIC通讯，就可以控制GT9146控制器查找ID，读写命令，数据、初始化序列，并操作相关坐标寄存器和位置寄存器来编程。

#### 3.5.2 FSMC驱动

STM32F103ZET6中有FSMC控制器，即灵活的静态存储控制器。他可以和同步或异步存储器连接，并对其进行写或读操作。一般可控制NORFLASH/PSRAM,外部PC卡，NANFLASH。

本设计使用STM32的FSMC来驱动TFT。传统的LCD/LED有8090方式，IIC,SPI方式与单片机通讯驱动，本文使用的液晶是有16位数据口的80并口，有8个信号口，这种方式驱动LCD读写时序和FSMC驱动外部SRAM的方式机乎一摸一样，所以可以用FSMC方式把液晶当作SRAM驱动，而stm32FSMC控制器可以通过相关寄存器的操作控制读写时序，这样就可以使用配置相关寄存器代替软件编写时序驱动80并口的方式，所以FSMC控制多位数据线80并口LCD是主流方式。FSMC存储控制信号一般有地址线，数据线，WE写信号，EO读信号、片选端CS， SRAM带字节控的，还有UB/LB信号。

TFTLCD信号包括对应RS、数据线、WR、RD、CS、RST、BL等,其中LCD的RD对应SRAM的OE,WR对应WE,数据线一一对应，片选对应片选，而LCD有RS，没有地址信号，但SRAM有地址线（对应FSMC引脚A0~A18）。LCD RS端口信号决定数据线送过来的数据类型是命令/数据，可以理解为一个地址信号。当FSMC控制写地址0，RS为0，进入写命令模式。当FSMC控制地址为1，RS为1，进入写数据模式，本文使用FSMC的地址线A10作为数据命令选择引脚RS。这样可以把LCD当作一个SRAM来使用，本文使用FSMC的NOR/SRAM存储块1来驱动LCD。

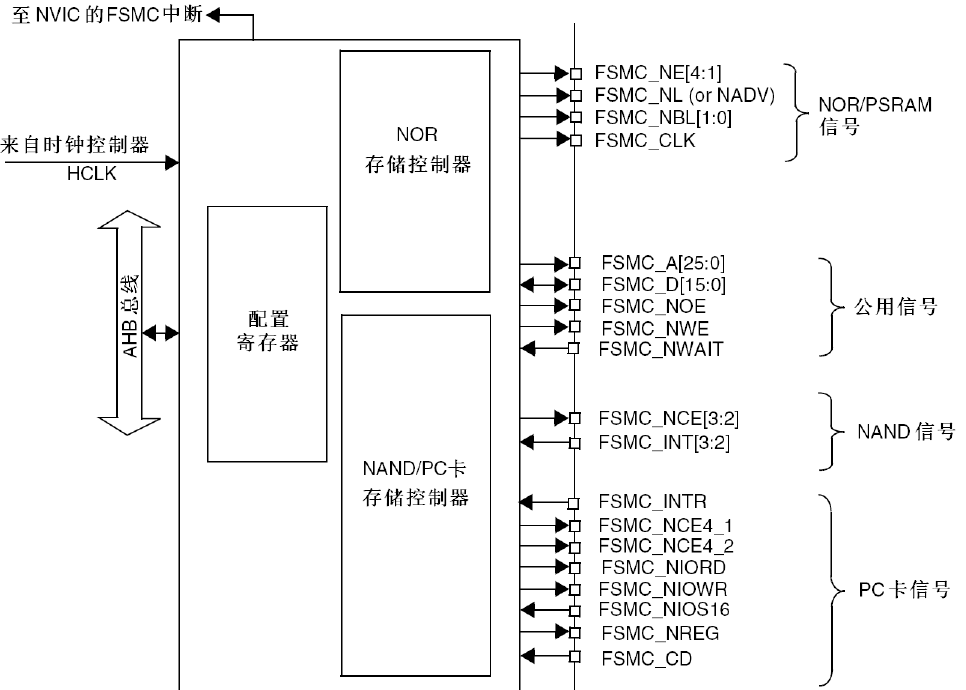
****

图3-28 FSMC驱动NOR/PSRAM ，NAND/PC

FSMC有模式1，A~D等模式，FSMC驱动LCD（SRAM）时一般采用模式A。其中NEX为LCD片选CS,NOE为LCD的RD,NEW为LCD的WR，FSMC寄存器中，写时序中ADDSET和DATASET设置NEW高低电平时间。读时序中ADDSET和DATASET设置NOE的高低电平。这些电平时间需要根据LCD的控制芯片的时序进行配置。（必须超过其规定最低判定时间）。

### 3.6按键模块设计

按键输入模块如图：

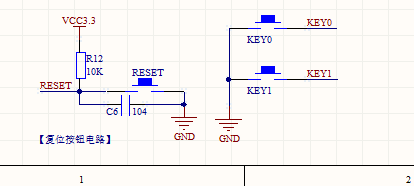
****

图3-29 按键输入模块

其中KEY0、KEY1一端接地，另外两端接入单片机引脚，所以为下拉输入，按下按键为高电平。这两个按键用于控制调温。RESET为复位按键，按下导通，为低电平，单片机所有带有RST。硬件满足复位的IC全部复位。

### 3.7 WIFI模块设计

本设计使用有人公司高性能串口转WIFI模块 USR-WIFI-HF-A11模块, 并设计了电源，连接，准备信号的指示灯，方便调试网络通讯，查阅其使用手册，GPIO3和GPIO4分别为串口发数据和串口收数据引脚。nReady为模块启动信号引脚，WIFI模块启动完毕后输出0，所以LED有正极到负极压差，LED被点亮。否则输出1。RESET为硬件复位引脚，低电平复位。Nlink为WIFI状态指示信号，有连接输出为0，LED被点亮，否则为1。nReload为恢复出厂设计引脚，连接了一个双排针，如需要恢复设置可以短接3S进行硬件恢复出厂设置。PHY为以太网连接。电路如下图

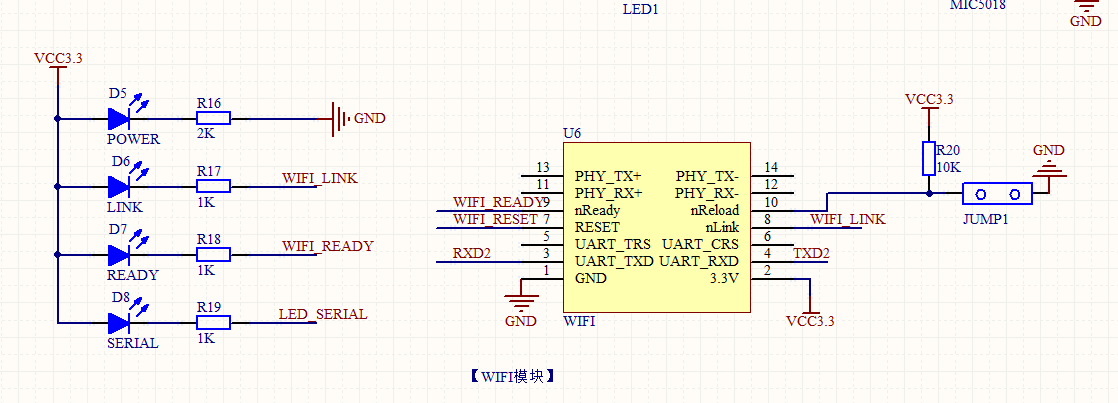
****

图3-30 USR-WIFI-HF-A11模块电路

### 3.8 24C02外设EEPROM存储模块设计

本设计需要使用EEPROM进行读写操作，用来实现改变PID参数或保存数据，需要外置EEPROM模块。EEPROM是电可擦可编程只读存储器，可以用来存储数据，这个数据在断电后还能保留，其与单片机通过IIC总线通讯，电路如下图3-31：

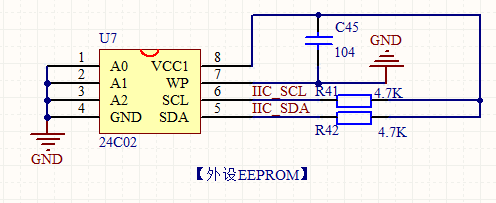
****

图3-31外置EEPROM模块

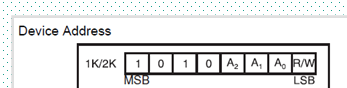
****

图3-32 24C02地址

由电路图，地址线全部接地，所以24C02写地址时，设备写地址为10100000,十六进制为0XA0。读地址为10100001(0XA1)。

## 第4章 温度控制系统软件设计部分

### 4.1 主程序的设计

ADC采集

查表转化为温度

按键触摸控制温度

PID算法控制

LCD显示并发送到上位机

上位机显示/控制温度曲线和PID参数

图4-1 主程序流程

在主函数开始前，调用所有用到的函数，本设计使用裸奔的STM32F103ZET6,没有移植GUI和RTOS，使用ST官方库STD，调用stm32f10x.h和其他需要用到的库头文件。

主程序开始时中对所有调用的库中的函数进行初始化。并进行NVIC中断优先级分组为2，即拥有两位抢占优先级和两位响应优先级。

程序首先要进行ADC采集，，调用LCD显示函数显示ADC数值，并根据基准3V算出ADC数值的具体电压并通过LCD显示函数显示在LCD上。公式：temp=(float)adcx\*(3.0/4096)（adcx为整型，需类型转换算出小数），得出的数值滤波（最大值最小值滤波）存储在receive中，再通过恒流源电路的公式算出电阻具体数值res=((receive/10.0)/0.909)\*1000.0。根据这个电阻阻值进行两分法查表，显示出最终温度ADC1\_tem=(float)get\_tem(res);再通过设置定时器中断状态标志位10s发送一次温度数据给上位机，并将温度显示在LCD显示屏上。LCD中编写GUI画圆和+-号，并将触摸屏触点判定定位到GUI部分，查询方式触摸触发设置温度KEYtem上升和下降。同理按键部分也是这样设置。最后接受上位机数据，判断类似JASON数据结构将数据给KEYtem和Kpid赋值。

### 4.2状态机与时钟的设计

虽然本设计单片机函数中已经封装了delay的函数库，但是其本质算法是做空操作循环，这种处于让单片机做空操作的延时的函数是不能大量使用的，这种延时操作时间太长会浪费大量内存，并影响其他函数的功能。所以在编写程序功能比较多时是不可以长时间让程序停留在某处的。但是在中断中延时是更加不可取的方法，在单片机编程中，停留在中断的时间必须少。对于STM32大容量芯片，可以采用嵌入小型实时操作（RTOS）的方法来解决CPU利用率，市面上比较多用FreeRtos和Ucos。在程序不多且CPU利用率不需要达到极致的情况下，在裸奔的单片机中可以选择一种状态机的程序写法，这种写法可以节省一定内存。这种写法不仅在stm32中，在许多小型单片机中同样适用。通常在裸机的情况下，单片机在主函数中执行while(1)循环中不会停止，想要停止只能通过中断方式，状态机的实现就是通过中断的方法。短时间轮询状态机方法将CPU资源调用给其他需要运行的任务。这样空操作也进行了，同时还利用了这段时间进行了别的任务，同时整个系统之间任务不会出现延时性。通过这种方法，将所有程序都用task[]封装，定时器作为其时序依据。这就是短时状态机的实现的基本原理。但是当TASK[1]执行时间过长（如100ms），TASK[2]没有机会得到运行时，这种方法就不成立，所以完整的状态机算法需要将长时间的任务需要通过状态机算出任务持续时间，超时就切割成短时间的步骤分时段完成。

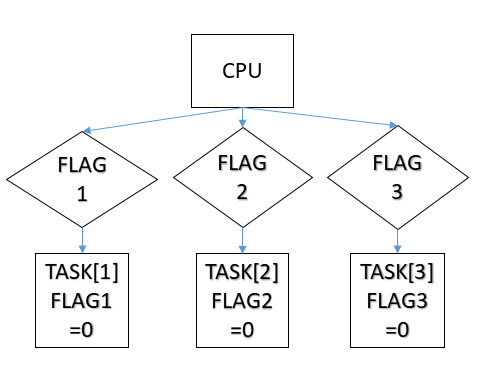


图4-2状态机轮询方式

借鉴状态机的轮番查询方法，可以给高时间要求的程序使用定时器计数，通过中断设置状态。在主函数中调用来周期性执行任务。如ADC采集和串口发送，PID算法等。

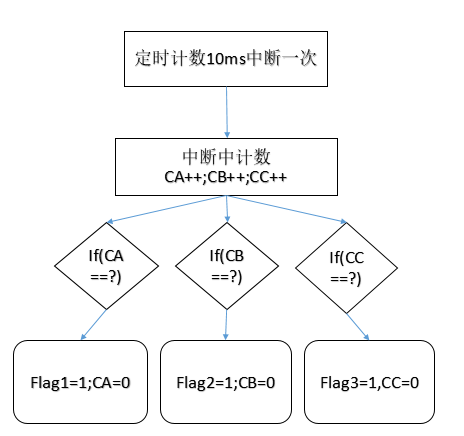


图4-3 定时器做状态机

本文使用TIM2作为状态机的定时器。本文定时时间为10ms中断一次，即自动装载值为ARR=99。在定时器2的初始化参数中配置定时器计数模式和其他NVIC中断参数以及使能位，入口参数配置PSC和ARR。最终在主函数中调用TIM2\_INIT(ARR,PSC)完成初始化，中断函数中做上述状态标志位运算。

### 4.3 PWM输出的设计

PWM为脉冲宽度调制，即可改变占空比的脉冲。STM32的部分通用定时器输出通道有PWM的功能。本文使用定时器3的CH2的输出通道产生PWM波形。并配置其相关寄存器。

本文PWM口为PB5。是STM32F103ZET6的定时器3通道2的部分重映射。所以在初始化PB5时设置为部分重映射模式，并使能RCC时钟。

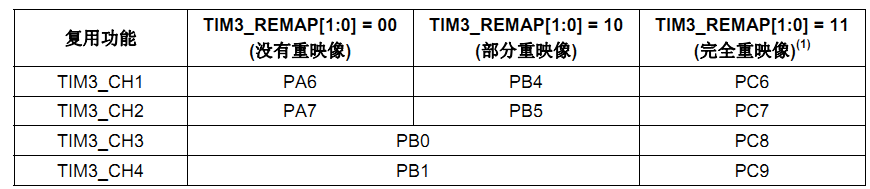
****

图4-4定时器IO口的重映射

同样，采用定时器3，所以要将定时器3初始化配置NVIC中断参数，并使能其RCC时钟。最后初始化PWM配置相关寄存器参数（计数模式，输出极性，使能等）。

PWM初始化函数中，arr和psc为入口参数，arr为定时器自动装载值，psc为预分频系数。配置完成后在主函数中调用该函数，可以配置PWM波形的频率（时钟/arr+1）并初始化PWM。在PWM寄存器中CCRX寄存器控制占空比。本文占空比需要改变，所以在主函数中调用ST官方库中的TIM\_SetCompare2(TIM3,X)封装函数来改变占空比，X即为CCRX的值。

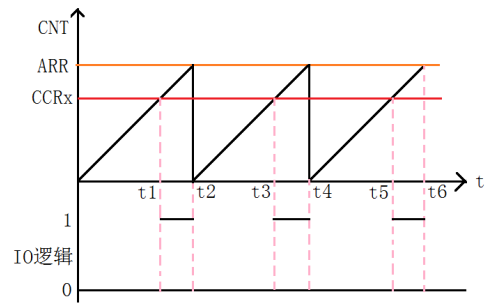
****

图4-5 PWMIO逻辑

### 4.4 ADC采集和滤波程序的设计

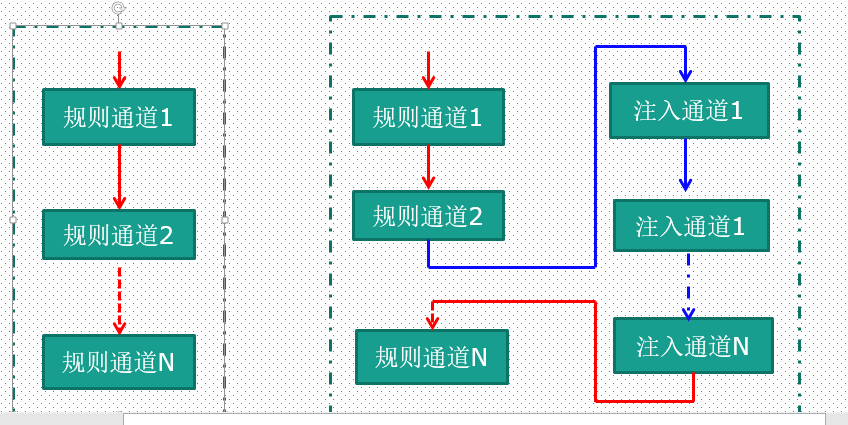
****

图4-6 ADC通道规则

ADC即模拟量转换数字量，STM32自带12位精度ADC，两路ADC在经过PT100转换电路后，端口的输入电压都为0~3.3V的模拟量，需要经过ADC的处理为数字量才能通过单片机转化为具体的电压值。

STM32F103ZET6有17个ADC通道，其中ADC1有15个模拟输入ADC通道，1个内部参照电压通道，1个温湿度传感器通道。使用STM32内部ADC转换时，通过通道进入内部转换器一个通道一个通道地转换模拟量，也就是同一时间一个ADC只能有一个通道在转化。其中本文使用ADC1控制器的 PA4(通道4),另一路使用ADC1为PA1(通道1)。STM32F103ZET6可以有4个注入通道和16个规则通道，在多通道转换时，规则通道按照通道顺序转换，不可以插队，存储转换值在ADC\_JSQR寄存器，多通道共用一个寄存器，所以想要多通道采集必须在规则通道采样结束后使能DMA请求将数据发送给单片机，否则寄存器中的值会被下一个通道采集值覆盖。而注入通道可以像中断一样插入规则通道的顺序而转换值存储在ADC\_SQR1寄存器中（16位寄存器）拥有4个寄存器。所以每个注入通道的值单独保存一个寄存器中，所以在多通道（通道不超过4个）是可以采用几个注入通道，从寄存器轮询处理取数据。而如果规则通道想要进行多通道转换就必须使用扫描模式，连续转换，每次扫描完成就发送DMA请求，通过DMA在数据被覆盖前发送ADC数据给CPU。

通过以上观察，本文使用两路ADC通道，使用注入通道可以直接取出两路ADC通道的数值，所以本文使用注入通道。配置引脚为PA1,PA4模拟输入，设置ADC初始化。并设置通道1，通道4采样周期144cycles(最大转换速率位1us，最大转换速度为1MHZ)获取后的ADC值。

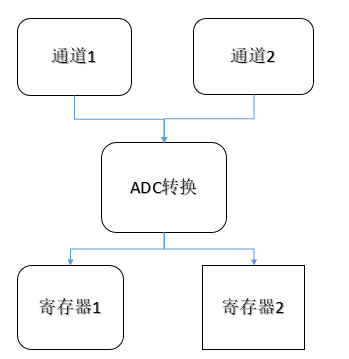


图4-7 选择注入通道方案ADC转化

### 4.5按键调节温度程序设计

设置按键控制温度的上升和下降，因为由于按键按下存在着抖动所以通过KEY\_SCAN()函数判断抖动，入口参数为按键模式，1支持连按，0不支持连按。（实际为1时候,静态参数松开按键标志key\_up恒定为1，即默认恒定松开）。由于主函数中需要调用了按键函数中产生延时，会对内存造成浪费，在本文无实时系统的STM32中可以采用外部IO中断触发方式可以避免按键升温反应慢速的问题。但是同样会产生抖动使按键触发了多次，这样按键同样需要在中断中进行消抖。于是采用定时器状态机扫描40ms的方案执行按键扫描程序KEY\_SCAN（）。

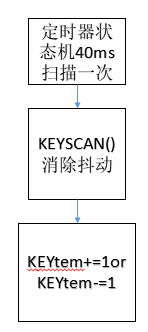


图4-8按键扫描程序

### 4.6两分法查表法的设计

由硬件电路可知，ADC采集电压和电阻的关系，.。而在实际程序中，需要知道温度。由于PT100温度和电阻不为线性，可以根据PT100温度和电阻对照表获得温度。查表采用两分法，此方法是常用的查表操作。首先定义数组，数组成员为350个，代表。为0~350°每1°的电阻，用数组下标代表温度。如：PT100[0]=100，代表0°为100Ω。

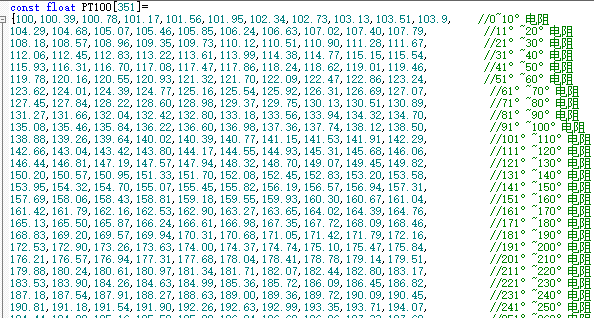
****

图4-9 温度-电阻对照数组表

再由表的长度得到中值mid。再由取得的res和中值mid所代表的成员比较，如果res>mid成员，说明res所代表的温度靠近表结尾，把表头的范围缩小。同理res<mid成员，说明res所代表的温度靠近表头，把表尾缩小。循环缩小直到，表头<=表尾时，就返回此时电阻对应数组中的位置（即温度）。

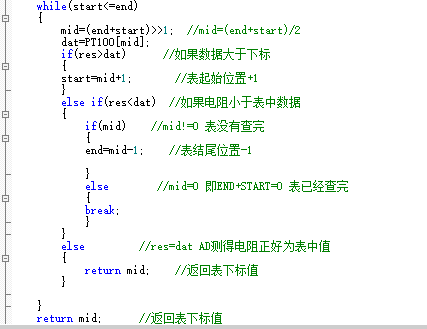
****

图4-10两分法查找数组位置

通过二分法取出表中的位置后，若PT100阻值正好对应表中温度值对应的电阻，则可以得到当前PT100采集的温度。但是若是PT100的阻值在表中没有对应温度，则通过二分法则可以取出离其最近的两个温度。此时默认此区间为线性，（非线性处理方法），通过比例算出此时电阻应该对应的温度。返回值为当前阻值对应温度（temp），所以主函数中直接调用此函数并把算出阻值作为参数即可求出温度。

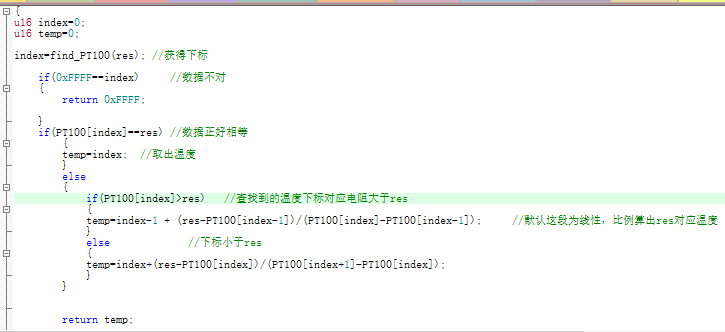
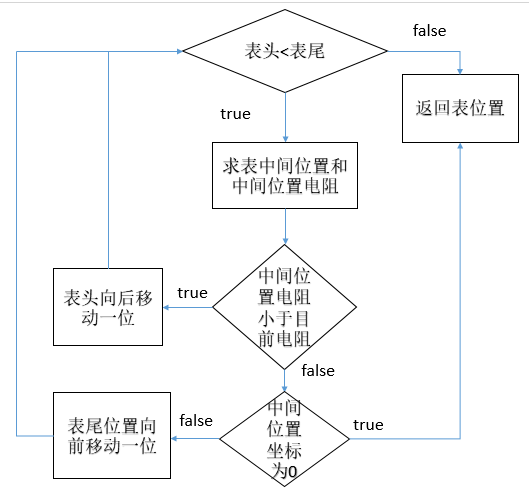
****

图4-11 计算温度



4-12二分查表法流程

由两分法查表可以看出，这种方法可以不查完整个表就可以找出具体数值，十分节省CPU运算时间，特别是在表中数据特别多时，是十分好的算法。

### 4.7 TFT显示温度和触摸程序的设计

显示时首先LCD需要复位。接着再初始化LCD。在初始化LCD之前需要编写程序识别数据格式是命令还是数据的。由于FSMC的地址A10接入LCD（RS）引脚，可通过配置寄存器选择输入的是命令还是数据。

本设计选择的是bank1区域4，其地址为0X6C000000，0X000007FE为A10地址的偏移，此时A10为0，+1为1（STM32内部会加1）。这样就可以控制RS=0（REG）和1（RAM）。之后可以定义写命令或是写数据的函数。

复位判断LCDID后，通过REG和RAM函数写入初始化序列命令和数据（使用手册）初始化LCD，初始化后配置LCD参数，竖屏，然后通过写命令配置坐标寄存器和扫描模式，开窗整个屏幕，再写入GRAM填充（从头到尾写入白色0XFFFF）并点亮背光就实现了初始化，并清空了整个屏幕。

此时默认初始化程序完毕。(其中还需要配置FSMC寄存器和lcd控制器时序匹配)

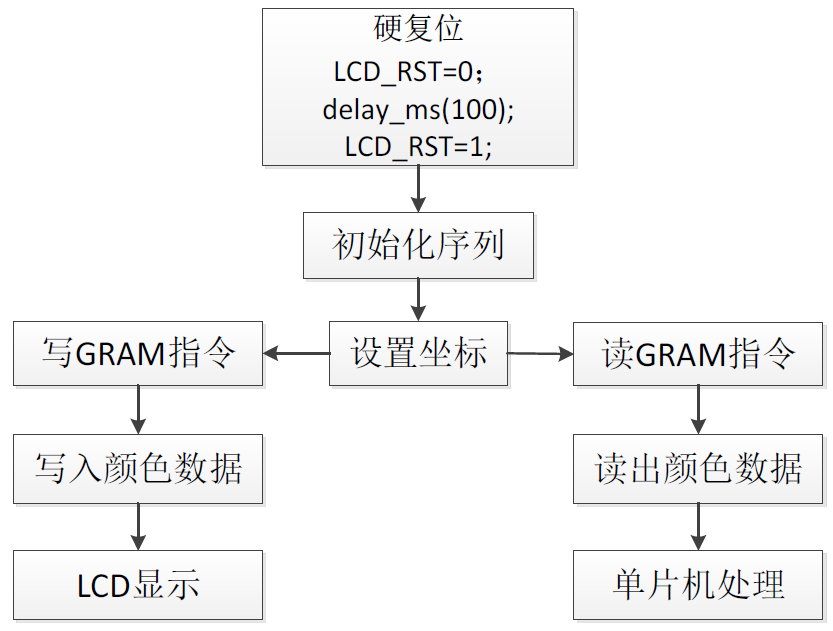
****

图4-14 LCD程序流程图

在初始化LCD并配置参数后，通过写GRAM指令写入颜色数据，按照扫描GRAM规律画点填充颜色，最后显示LCD。同理，读时序也是一样。

本文中NT35510控制器支持24位颜色深度。本文使用16位颜色深度(65K颜色)，格式为RGB565。如0xF800为111110000000000其D15-D11都为高，则显示为红色。本文库中宏定义了各种颜色对应各种RGB565格式，直接调用颜色宏定义参数写入寄存器就可以改变颜色数据。

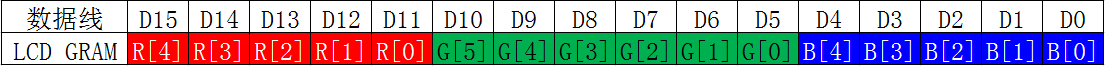
****

图4-15 RGB565格式

最后通过调用显示函数showchar或shownum来显示字符或数字，并使用简单GUI。

同样触摸屏函数也需要初始化，由于触摸屏是采用IIC通讯，所以先要调用IIC库中的函数组成时序。配置相关IO口，触摸屏控制器采用GT9147控制器，在其复位后5ms内保持INT为高电平，器件地址为0X14 ，IIC通讯发送0X14。并获得ID确认后发送产品数据序列（厂家手册），再经过touch库文件校准触点，兼容识别ID。最后扫描函数确认状态寄存器的状态确认是否有触点按下，和坐标寄存器保存坐标位置（画图中用到），本文主要使用状态寄存器。在主函数中判断结构体tp\_dev的成员touchtype最高为是否为1，如果为1即按下触摸。在temp中0XFF左移动5位，即为11100000，取反变为00011111，根据状态寄存器低5位有1便有1个触点的原则判断触点的个数。即有5个被点触摸。再1<<t判断sta低5位的状态，如果有效，则读取坐标，并认为是有效的触摸点。

主函数中通过查询touchtype触摸状态，如果处在触摸状态再调用扫描函数扫描触摸屏所有触电，t用来查询有几个触摸位置，在通过LCD定位出触摸范围，在这个触摸范围有触电扫描时状态寄存器是1，则产生触摸事件，触摸事件中设定温度改变，GUI改变，离开触电，GUI位置被清屏函数（需定位）清屏然后继续显示GUI。

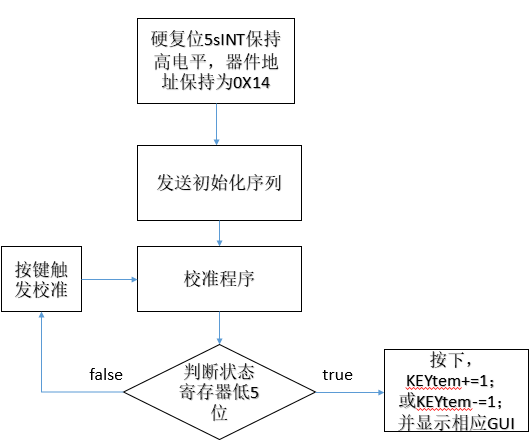


图4-16 触摸流程图

### 4.8 PID控制算法

PID算法是常见的控制算法，能直接体现反馈控制的一套算法，其通过P（比例），I（积分），D（微分）等3个环节控制控制器，并通过传感器负反馈调节，其直接体现如下：

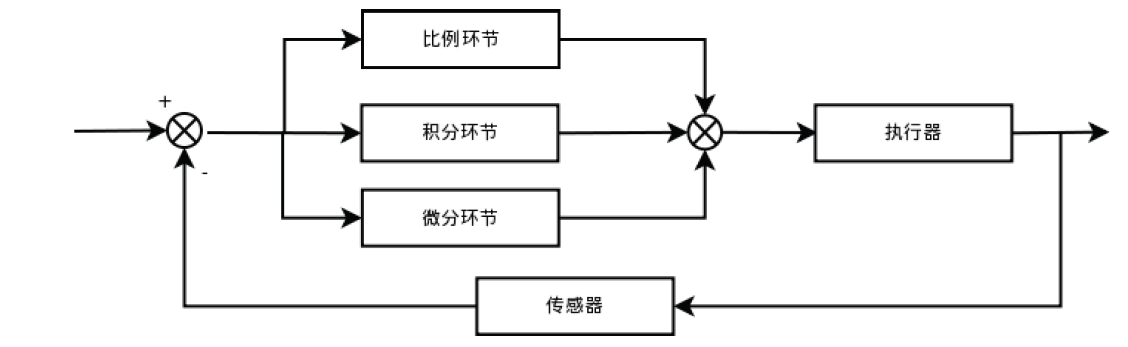
****

图4-17 PID控制器流程

PID的控制公式可以体现为：

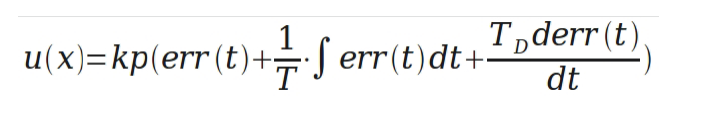
****

图4-18 PID公式

在温度控制系统中err(t)体现为想要达到的温度Settem（按键触屏设置）和现实传感器的温度Acturatem（传感器通过ADC运算转换获得）。积分项体现为err（t）随着时间变化的累计，即err的累加，可以体现为integral+=err,最后一项微分项体现err和上一次err的差值，即err-err.last。这样就能使u（x）使用变量完全展现。所以定义一个PID结构体，接着给其赋值参数全部为0(KPID)除外。

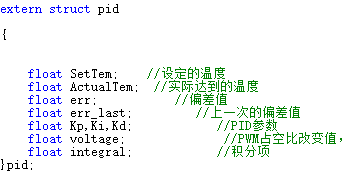
****

图4-19 PID参数结构体

主函数中需要上位机改变PID的值所以将其定义为全局变量，在库中作为调用使用。

那么U（x）即voltage即可体现公式为

****

图4-22 PID公式离散化体现

在公式列出后，可以很容易得到PID运算结果。PID的运算结果代表着占空比，即可理解成当前PWM寄存器CCRX的值（极性为正的情况下）。

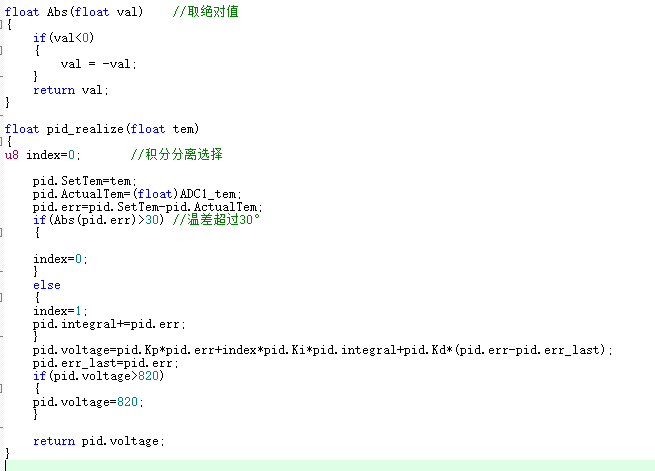
****

图4-23 PID实现函数

在主函数中调用pid\_realize()可以实验PID运算，入口参数为tem，对应主函数中KEYtem变量，是实际想要控制的温度。而actualtem是PT100采集出的温度，可以使用全局变量将采集值赋值给PID结构体的actualtem成员。这样完成运算后，返回值就是PID运算结果。在通过触摸控制改变KEYtem的温度比采集温度高时即可使外部加热设备升温,经过PID控制的共同作用下最终稳定在控制温度附件。在实现中还加入了积分离定的过程，超过30°不产生积分作用，加速升温。在本文实验中CCRX寄存器为820，即占空比为56%的情况下，通过示波器发现MOS管波形不够稳定，影响PID调节，所以限制了其寄存器的值最高为820，即占空比限制在百分之56。

## 第5章 实验部分和上位机的数据传输

**5.1 图腾柱电路的实验**

本文PWM采用图腾柱推挽，虽然经过了仿真软件仿真，但是其只能作为参考依据，实际使用中还需要手动通过示波器观察波形来确定电阻阻值，从而使MOS管导通波形较为良好。

可以采用在冲孔板上搭建电路并飞线至单片机PWM口实验

此时烧入PWM频率为50KHZ，占空比可调整的简单初步程序，用示波器测试在不同占空比下，MOS管GATE波形和DS导通波形，最终调整波形如下：

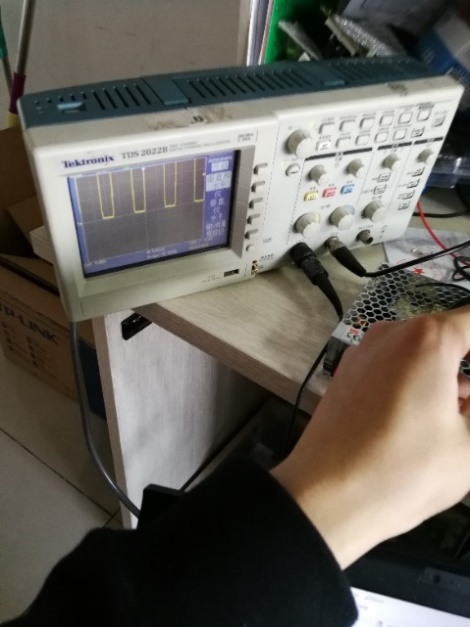
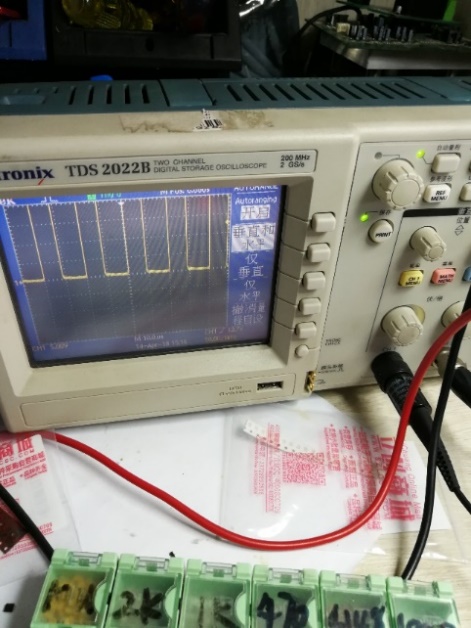
****

图5- 1 MOS管GATE波形和DS波形

经过调整后的波形能在一定占空比范围内保持稳定波形，就可以确定使用电路的电阻。调整的方法为先断开R46、R47反馈，先调整R43、R44电阻，测定R43、R44之间PWM电压是否稳定和波形是否失真，在一定范围占空比的情况下，使波形稳定，并确定R40电阻（R40电阻一开始不能过小，如果24V电压击穿三极管，电阻分压作用较小会使STM32芯片烧坏）。然后接入反馈电阻，调整阻值，不接MOS管，查看mos管GATE波形，波形稳定后再看电压是否超过MOS管型号手册中规定的GATE电压，如果电压不符合，再调整R45使其稳定在一定范围内。

然后接入MOS管再继续看GATE波形，最后接入负载，再看MOS管DS导通的波形，最终使得MOS在一定范围PWM占空比的情况下导通稳定。

最后需要确定MOS管型号，需要选择一款内阻较小，耐压较高的NMOS管，于是选择的型号为STB100N6F7Nmos管，其内阻5.6mΩ，VDS耐压60V，门极最高耐压20V。

**5.2上位机和下位机通讯以及JSON格式数据处理**

本设计使用串口2作为wifi串口，并使用使用有人公司WIFI高性能USART串口转WIFI模块。使用TCP协议的通讯方式，串口TX(设置为推挽复用)，RX设置位浮空输入。USART2波特率设置为115200。

发送使用发送字符串函数（STD库中集成了发送一个字节的库函数），字符串函数将位分为一个一个字节发送出去。主函数中设置状态机10S发送一次，用sprintf函数将ADC变量（数字量）变为字符量发送。

串口接受使用中断，当接受缓冲区非空时开启中断。自拟定一个通讯协议，定义一个16位的标记量，最高位为发送完成标志，1完成，0失败。14位为换行符标记。13-0为有效字节数目，通过查询标记状态来接受数据，未接到换行符号，或者最高位没有标记为1就认位不是有效数据，以防接受错误数据影响参数。在中断中查询状态后在主函数中通过for循环将一个一个字节数据（8位）缓冲区数据发送至数组指针存储。

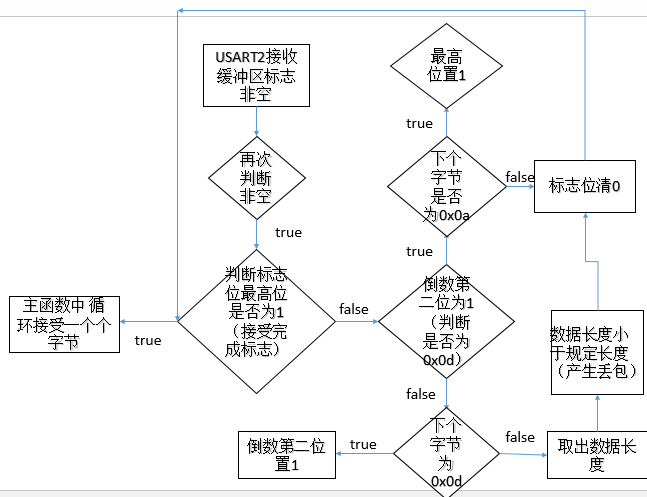


图5-2 串口接收协议

接受的数据由上位机发出，采用类似JASON数据格式，本文为了方便省去JASON数据中的“”标记。上位机发送数据例子如下：SetTemp:100,P:22,I:0.05,D:0.01\r\n

在主函数中用strstr()函数查找字符串中的特殊字符（如SetTemp:），此函数返回值为指针型，无返回返回NULL。可通过此函数查找字符串在整个字符数组的位置（指针）。再运用strncpy（）设置区间（中间参数为整形，可用指针相减得出偏移量）将数字字符数据保存在数字变量JASONDATA中。

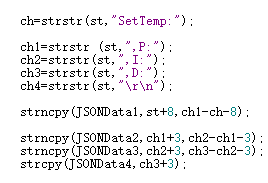
****

图5-3 查找字符串中特殊字符后跟着的数字位置

最后用atoi()/atof()函数使数字字符型转整形/浮点型。通过这种方法可以将长字符串通过一一判断读取出数据。然后通过循环清除JASONDATA数据（不清会造成下次修改数据中的几位还是上一次的数据中的几位）

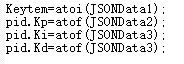
****

图5-4 数字字符串转化为变量（整形/浮点）

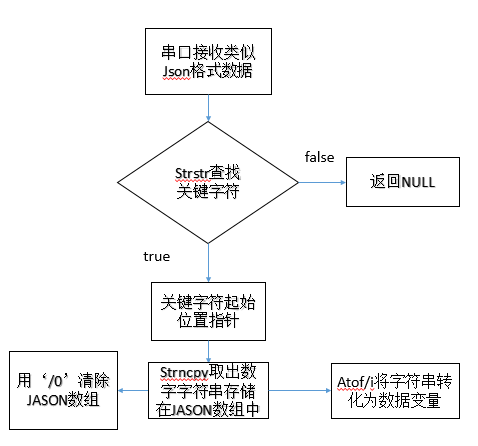


图5-5 JASON格式处理流程

最后上位机设置校验位和波特率必须和串口2配置一样。通过上位机可以远程控制设备达到物联网的控制模式。只要单片机提供接口或数据，就能实现多种控制方式，控制的程序不仅仅局限于单片机，也可以在上位机中提供定时关闭，报警等。本文使用C#编写的上位机。

本文使用TCP协议透明传输作为WIFI传输协议，单片机WIFI模块作为客户端Cilent，电脑作为服务器Server，由客户端发送数据到服务器，服务器接受数据。由此如果服务器连接外网可将数据发送至云服务器或手机终端中，通过类似APP或网页实时监控温度数据**。**

**5.3上位机整定PID参数**

接下来就是整定PID控制器参数了，本文使用的是经验值法整定，主要通过实验方案确定临界比例。所以实验方案体现出来的数据至关重要。在整定时要做到不要心急，观察三个周期后再下结论，结合上位机展现的曲线，每次改动都要有目的性（曲线哪里不好，改变了的参数会对其造成哪些影响），多总结，才能整定出一个比较好的参数。参照PID整定有一套口诀：

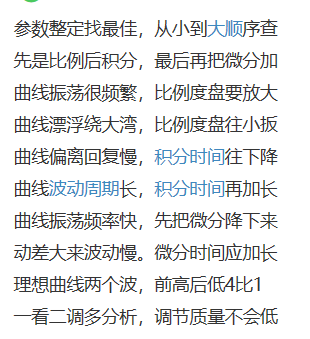
****

图5-6 PID参数整定口诀

由这套口诀可知在整定时应首先确定P（比例系数），隔离I和D。首先规定调节时设定在30°左右状态下升温到100°，在只有P控制时，发现P控制器能使温度开始改变，但是会使此时温度和想要达到控制温度存在一定差距，这种差距叫做静差。只在P控制器调节时是无法消除这种静差的，但是还是需要改变P的参数使其静态差距较小并符合理想曲线（一个大波加小波后趋于稳定即口诀中的理想曲线两个波）。首先P从0开始取值，直到系统产生较大的振荡。实验一直取到P=26时，曲线在两个周期后产生抖动波动。波形如下：

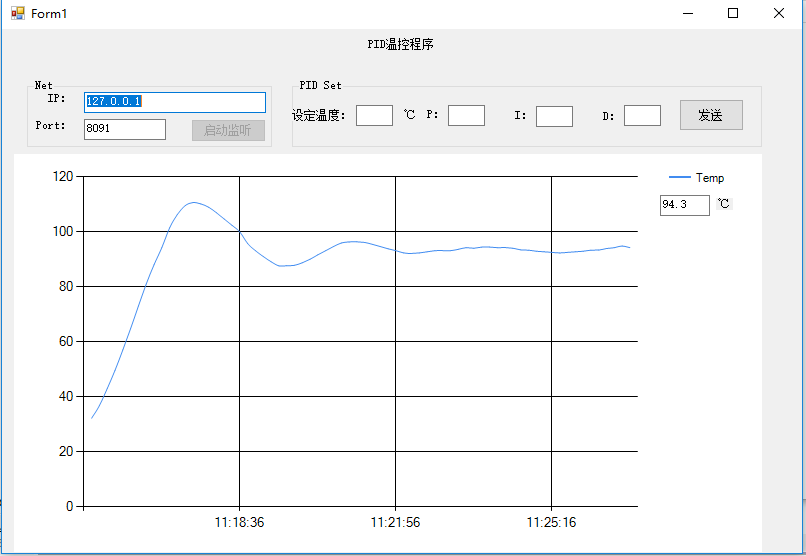
****

图5-7 P=26时温度波形

接下来再减小P值，试试P=25,曲线平稳了一些，再一直试到P=20曲线出现振荡。得出结论P在22-23的时候曲线较为稳定：

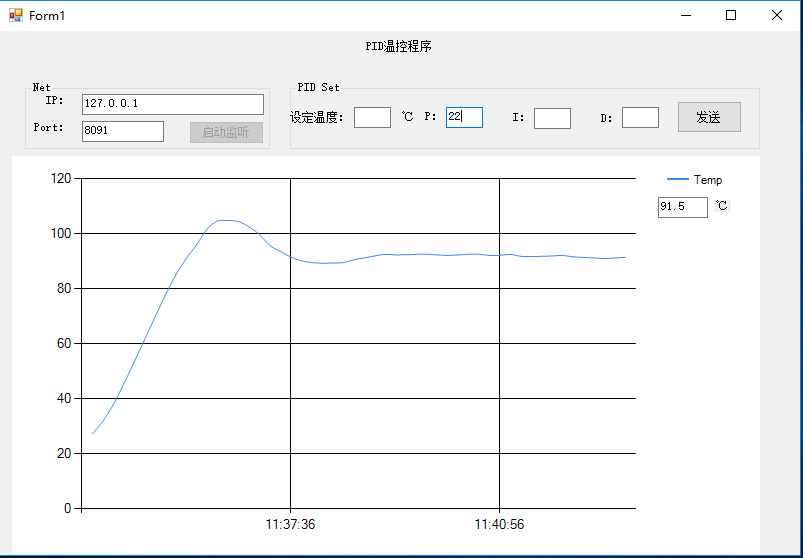
****

图5-8 P=22时温度波形

之后，把上位机接受数据的周期变短，确定P=22.2的曲线是最为稳定且静差不大：

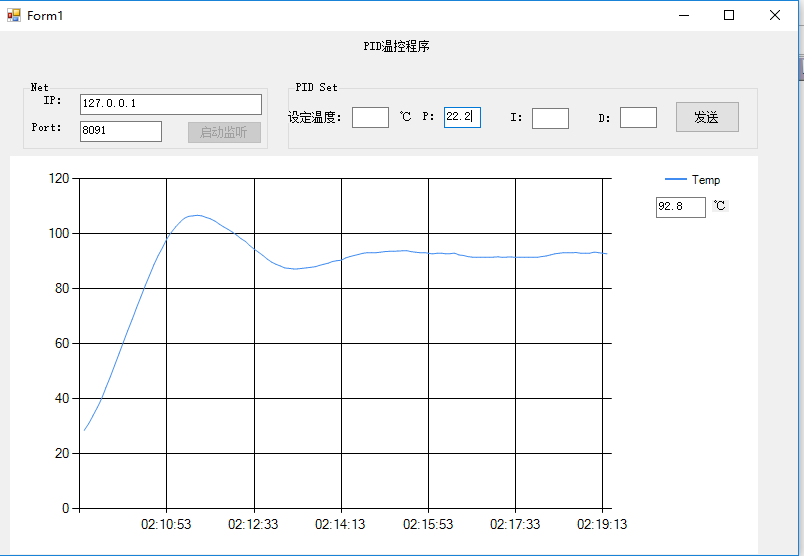
****

图5-9 P=22.2时温度波形

随后就需要消除静差了。控制器I的公式是err的历史累积，有消除静差的作用，但是过大的I值会使系统具有滞后性，并引起系统振荡，所以需要从一个较小的I值开始整定，首先设置I=0.2：

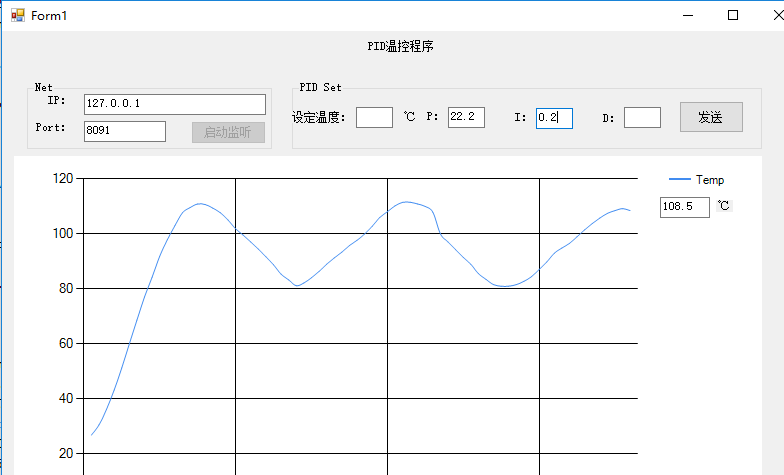
****

图5-10 P=22.2,I=0.2时温度波形

可以发现I值的加入消除了静差，但引起了系统的强烈振荡反应（第二个周期时波形比第一周期波形振幅还要大），说明这个I值过大。于是再试试I=0.1,还是振荡。再试试I=0.05:

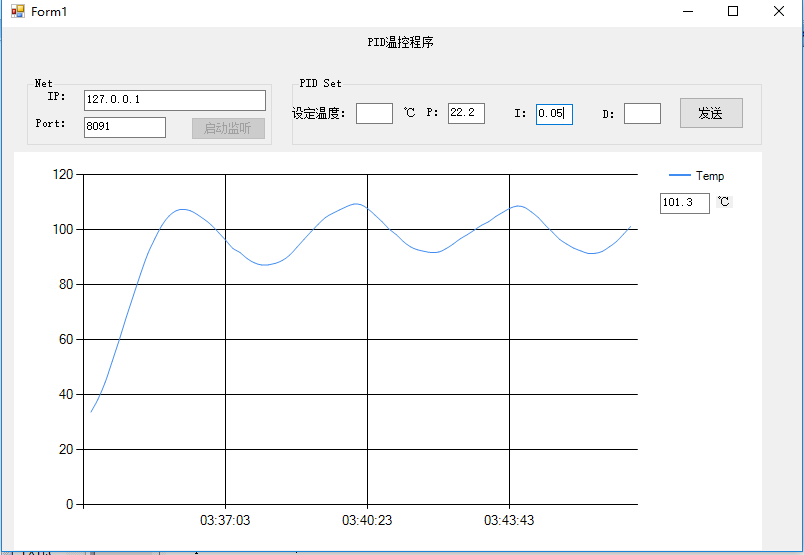
****

图5-11 P=22.2,I=0.05温度波形

振荡幅度变小，说明还需要继续缩小I，减少系统振荡。测试到I=0.02的波形如下：

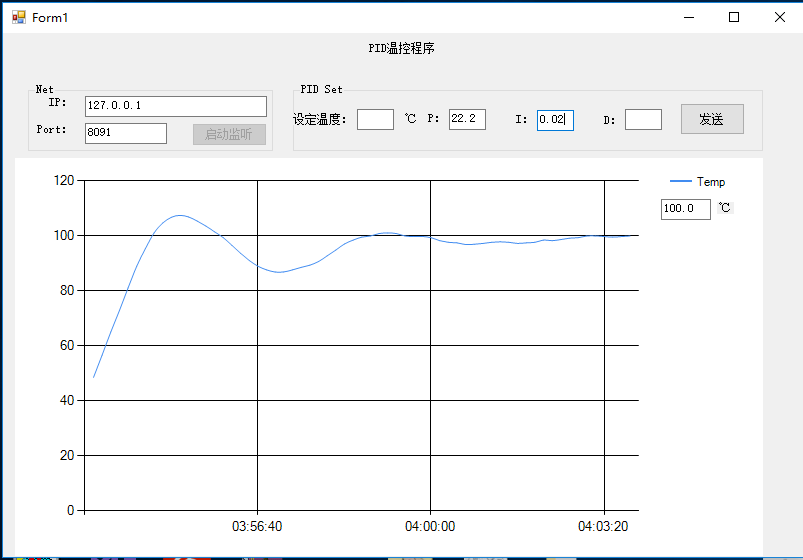
****

图5-12 P=22.2，I=0.02温度波形

第二个周期后波形比较稳定了，但是发现第二个周期的时候超过100°时间少，100°之下时间长，而I=0.03时候的曲线更好：

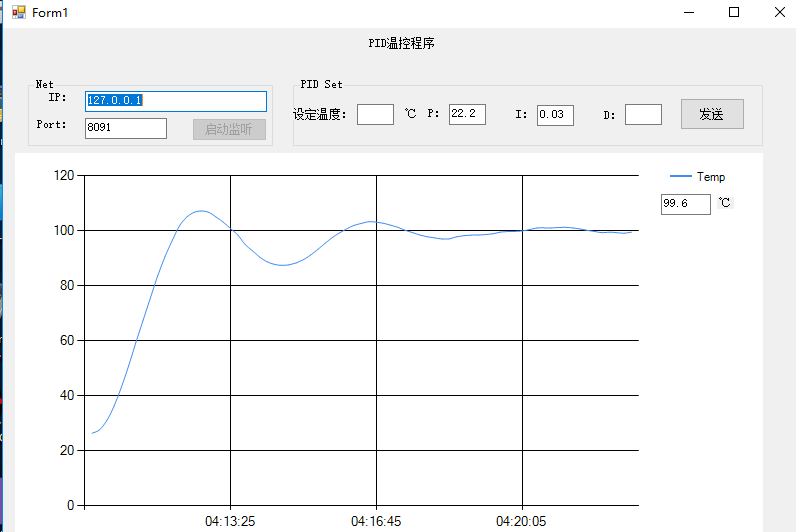
****

图5-13 P=22.2,I=0.03时温度波形

曲线此时再缩小上位机接受周期，再0.02往上试，最终确认I=0.026时的值更加好。积分作用会随着时间消除静态差，如果积分作用过强，会导致系统不稳定，动态响应会变慢。

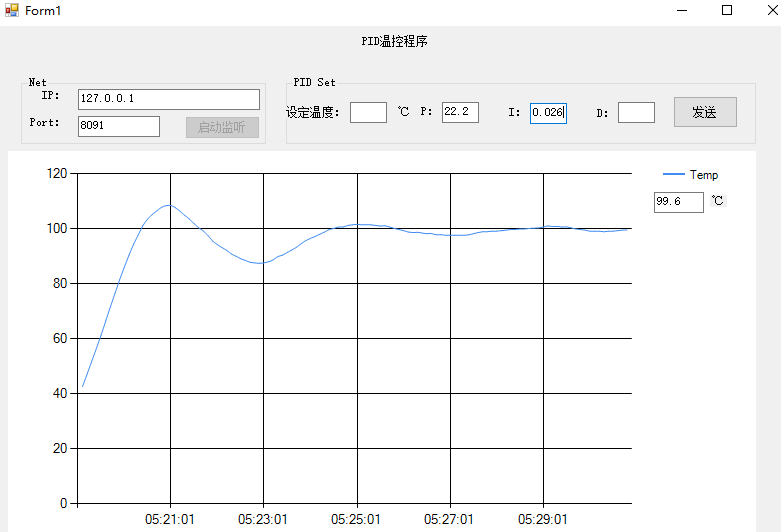
****

图5-14 P=22,2,I=0.026温度波形

此时能保证I的值尽量小，曲线又平滑。

在PI控制器的作用下，曲线基本稳定在控制温度附近，但是稳定在控制温度附近的时候会有一定抖动，这时候想要快速让曲线从偏离的方向返回时就要加入D（微分）控制，D的公式为偏差值和上次偏差值的差值，可以体现离目标温度的距离变化，通过改变D的值可实现超前调节，即能解决在积分作用下，曲线在控制温度附件抖动时回归速度慢的问题。再加入D作用后，整个控制系统对抵抗环境的干扰能力也会增加。但是D值的作用过大同样会使系统反应过度造成振动，所以在不影响系统的情况下使D值越小越好。经过经验值的方法，D值设定在0.1。最终形成的最好曲线就如口诀中所说的，先产生一大一小两个波后，在第三周期内趋于稳定的经典PID控制曲线：

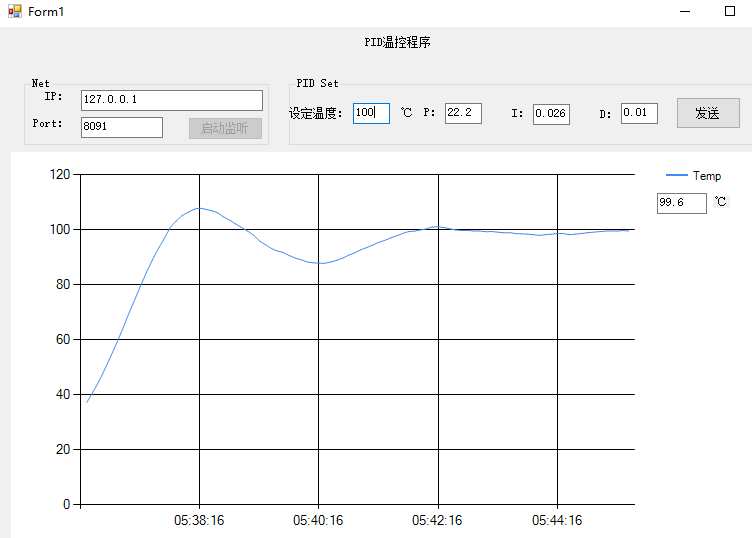
****

图5-15 P=22.2,I=0.026，D=0.01时温度曲线

在改变不同温度时也能稳定在+-1~2度之间：

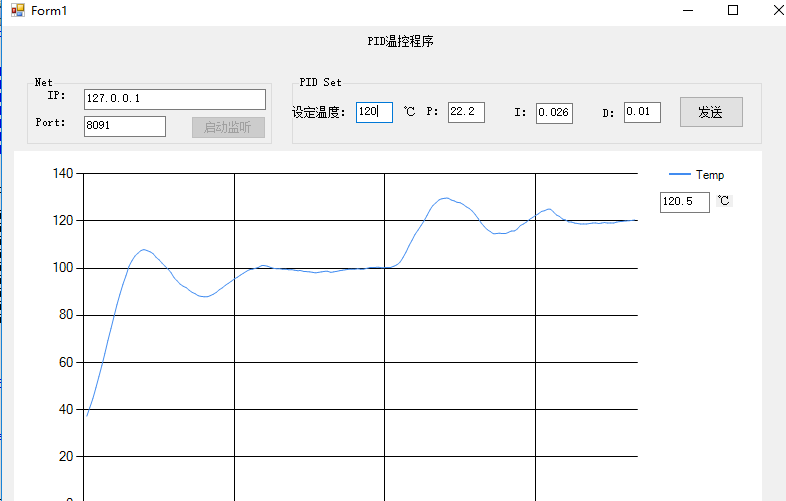
****

图5-16 在100°和120°时温度曲线

虽然能做到+-1°精确控温，但在图形上看出一个周期内两个峰值的时间有点不对称，所以在调整完毕后控制温度如果还想做到更加精准，PID的参数还需要细微调整，在图形中积分消除静查还不够，可以加大积分作用。但积分作用过强会影响动态响应，而且积分作用会随着时间一直作用，在多个周期后精度会越来越高。在多方面取舍之后可以设定出理想的PID值。

## 第6章 总结和展望

### 6.1总结

本设计在温度系统上创新地通过网络远程监控画出曲线图，从而直观方便地调节PID参数，并能通过上位机改变温度参数和PID参数使温度控制器的控制方式更加多样化。本设计在制作过程中解决了PWM波形的放大和MOS管驱动电路的问题，并提出了两个PT00的测温电路方案。同时展现了PID参数整定的完整过程，有一定参考意义。网络上有较多PID的整定方法，也有温度控制器PID参数的大概范围，甚至一些模糊控制的方法，但是一般比较稳定的曲线是需要配合自己的控制器整定出来的，本设计使用实验方法测定临界比例，这种方法经过测试可以得到比较稳定的曲线，是一种比较常用的整定方法。同时上位机WIFI通讯也可以达到物联网的效果，让温度控制系统更加智能化。

### 6.2展望

这次的毕业设计同样也有一些需要改进的地方，比如PWM电路可以通过更多次检验更加完善，从而达到更加高的占空比，可以学习移植MODBUS协议进行串口通讯，使用MQTT物联网协议传输数据等等。PID算法中可以学习参数自整定的模糊算法。在显示屏的处理中由于时间不足没有移植EMWIN或UCGUI来美化人机交互界面，如果LCD移植了GUI甚至可以达到类似上位机折线图的效果。同时也可以通过这次毕业设计学习RTOS实时操作系统的移植，为以后学习LINUX或更多操作系统打好基础。同样，学习STM32有助于帮助我理解CORTEX内核，为以后学习ARM打下基础。

物联网、云计算、大数据、人工智能是未来计算机领域的发展方向，本次设计仅进行了初步的网络联机实践，通过将嵌入式设备与网络互通进入网络对工业自动化控温具有很重要的意义，通过这次学习使我发现了自已的不足，明确今后学习努力的方向。

## 参考文献

[1] 李宁.《ARM开发工具ReaIView MDK使用入门》[M].北京航空航天大学出版社,2008.

[2] 李宁.《基于MDK的STM32处理器开发应用》[M].北京航空航天大学出版社,2008.

[3]徐向民《Altium Designer 快速入门》[M].北京航空航天大学出版社,2008.

[4]刘军、张洋、严汉宇编著.《原子教你玩STM32库函数版本》[M].北京航空航天大学出版社.2013.

[5] 刘军、张洋、严汉宇编著.《原子教你玩STM32寄存器版本》[M].北京航空航天大学出版社.2013.

[6]Joseph Yiu《The Defi native Guide to the ARM Cortex-M3》[M]. Copyright, Elsevier Inc. All rights reserved. © 2007 .

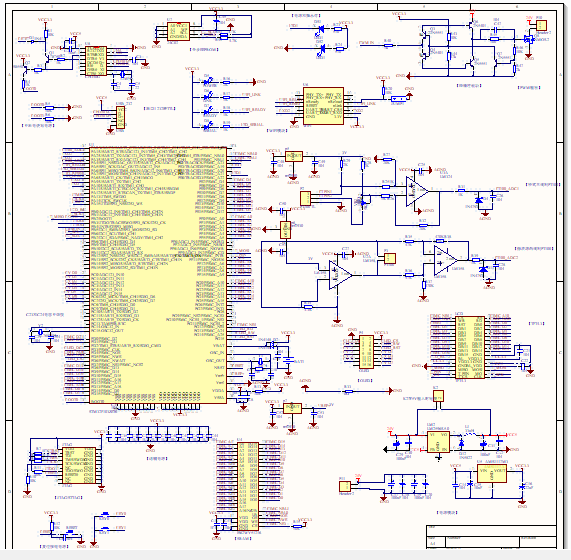
[7] STMicroelectronics《STM32F10XXX\_RM0008 reference manual》[EB]. STMicroelectronics - All rights reserved.© 2014 .

## 致谢

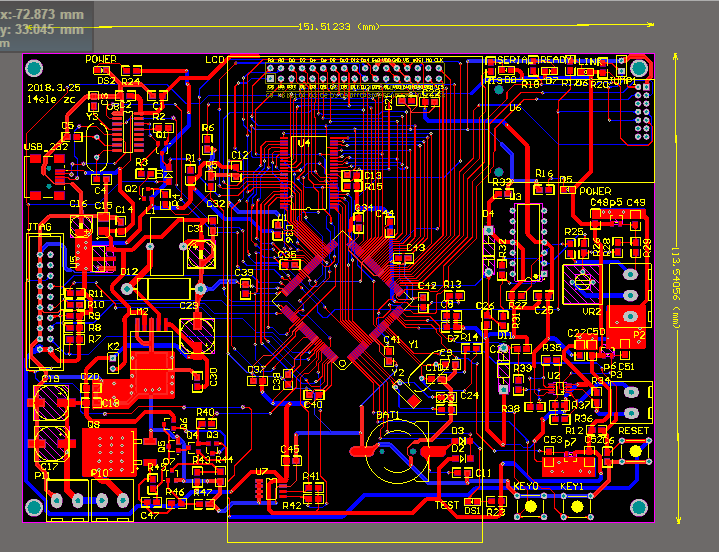
四年的大学生活即将迎来结束。这次做毕业设计的过程中，李老师给了我很大帮助，在做毕设的过程中我问了他很多问题，他都给了我答复，并在他的丰富的经验帮助下解决了一些模拟电路和PWM波形分析上的问题。同样还要感谢学校、其他老师和同学大学四年中给我的鼓励和帮助，还要感谢amobbs论坛，opendv论坛里的朋友给出各种见解和帮助，感谢正点原子的学习视频和源码还有ST官方资料的帮助下使我快速上手STM32。最后还要感谢父母给以我的鼓励和支持。

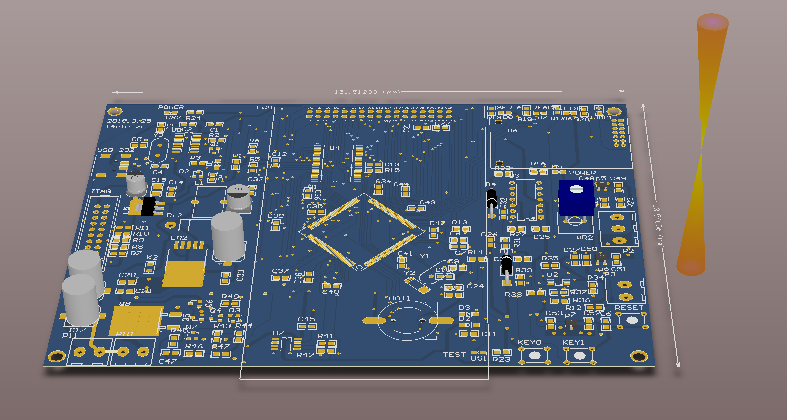
大学生活就将结束，我会记住老师们的教导，努力学习，用积极向上的态度面对以后的生活，回报社会。

## 附录A

****

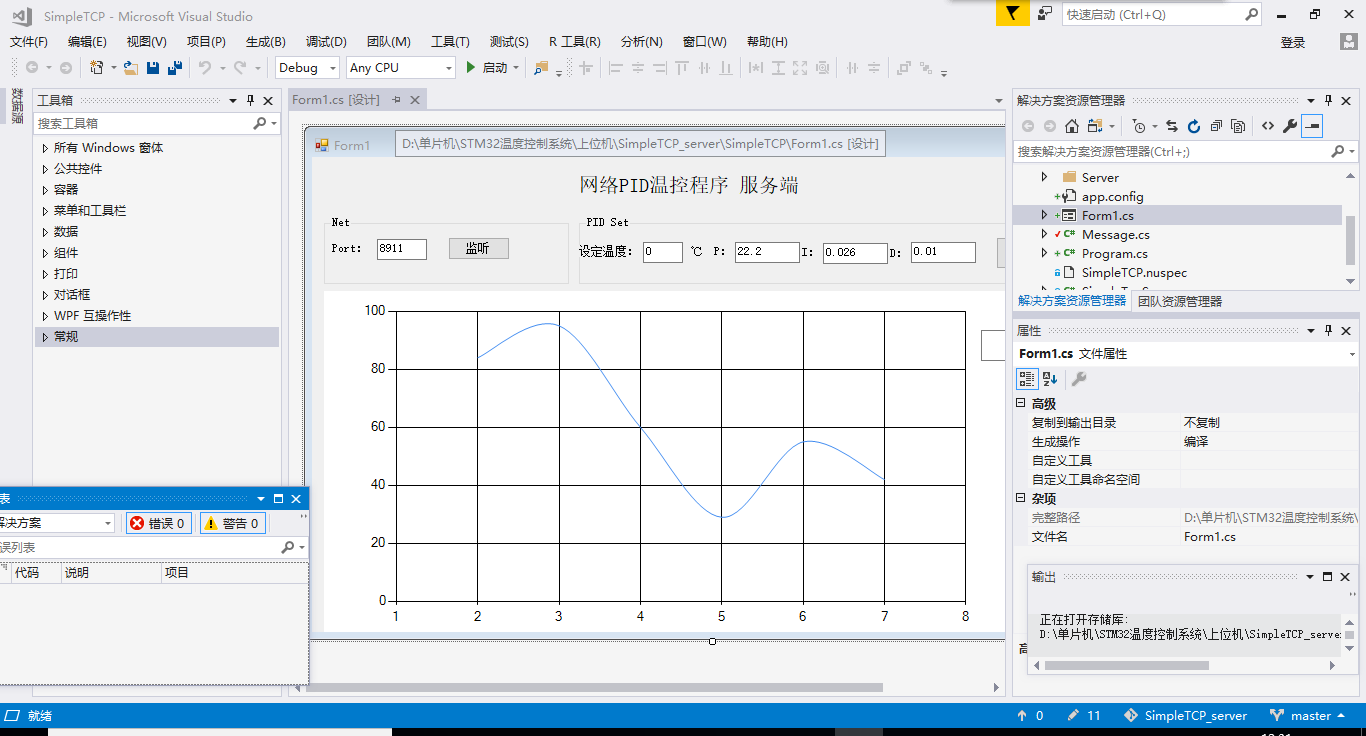
**原理图**

****

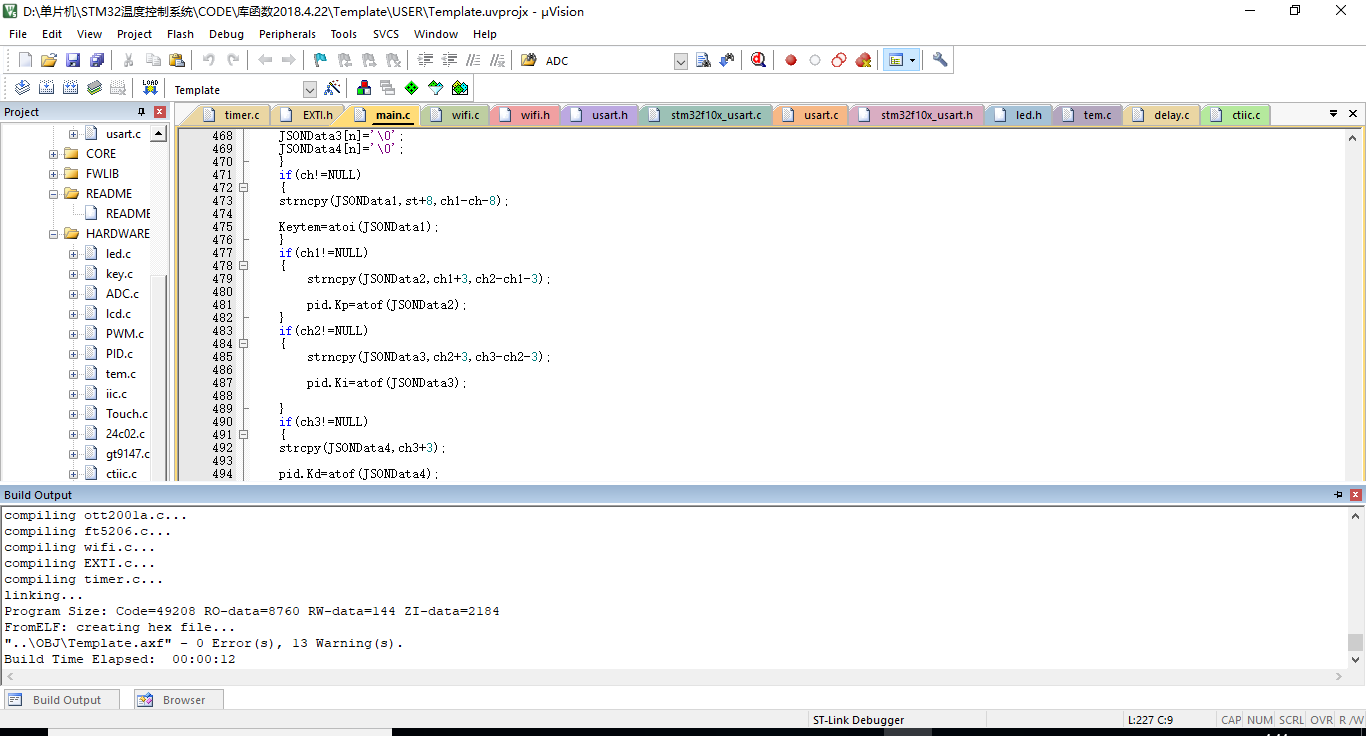
****

**PCB**

## 附录B

****

**上位机**



**源代码**