

浙江大学

实验报告



题目 基于 Arduino 的温度检测显示控制器的设计与制作

指导教师 叶险峰、施红军

专业 电子科学与技术

所在学院 信息与电子工程学院

浙江大学实验报告

课程名称：____ 电子电路设计实验 II ____ 任课老师：____ 叶险峰、施红军 ____
实验名称：____ 基于 Arduino 的温度检测显示控制器的设计与制作 ____

一、摘要

在现今科技高速发展的时代，各行各业对控制和测量的需求渐长，其中，温度测量和控制在很多行业都有重要的应用。温度检测最简单的方法就是采用单片机、温度传感器配合工作。但通常的单片机不宜上手，开发周期长，而 Arduino 的出现正好解决了这个问题。本文利用 Arduino 的便捷灵活、方便上手的特性，并选用 LM35 作为温度采集传感器，LCD1602 作为温度显示屏、EC11 旋转编码器作为温度控制器，设计了一套温度检测显示控制器方案。

关键词：Arduino、温度检测、温度控制、LM35、LCD1602

二、实验目的与任务

- 1、学习和掌握温度检测显示控制器的基本原理；
- 2、掌握 Arduino 编程原理；
- 3、学习和掌握 PCB 电路板的设计和制作；
- 4、学习旋转编码器和 LCD1602 液晶显示屏的使用。

计划达成如下要求：

- (1) 温度测量：测量范围为 $0^{\circ}\text{C}\sim 100^{\circ}\text{C}$ ，测量精度为 0.5°C ；温度测量不发生跳变。
- (2) 温度控制：实现用旋转编码器设定控制温度。不按下、旋转旋钮，设定控制温度；按下、旋转旋钮，快速设定控制温度；按下、不旋转旋钮，进入 override 模式，温度加热不设上限。

三、实验设计方案

本项目使用 Arduino UNO 扩展板进行温度检测显示控制。

Arduino UNO 扩展板上装载温度设定模块、温度显示模块和温度控制模块。硬件框图如图 1 所示。下面对各部分模块进行说明。

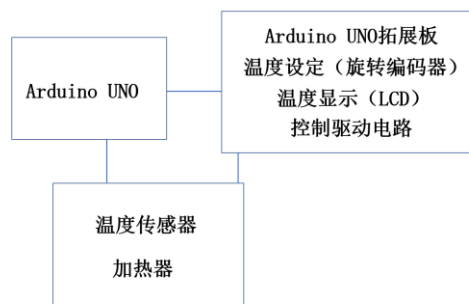


图 1 温度检测显示控制器框图

LCD 显示模块

本次实验使用 Arduino UNO 直接驱动 1602 液晶显示字母和数字，LCD 显示模块有一个优势就是将驱动电路集成在模块中。LCD 显示模块具有标准，不同的生产商所生产的大多数模块都可以按相同方法来使用。1602 液晶在应用中非常广泛，最初的 1602 液晶使用的是 HD44780 控制器，现在各个厂家的 1602 模块基本上都是采用了与之兼容的 IC，所以特性上基本都是一致的。

旋转编码器

旋转编码器是一种数字器件，这种器件有两个输出（A 和 B）。图 2 示出了当旋转编码器工作时输出 A 和 B 的信号是变化的。当顺时针旋转旋钮时，脉冲会发生变化，并且在图中从左向右移动；当逆时针旋转时，脉冲会在图中从右向左移动。

因此，如果 A 和 B 都是低电平，然后 B 变为高电平，表明旋钮顺时针旋转；如果 A 低 B 高，然后 A 变为高电平，也表明旋钮顺时针旋转。反之亦然。

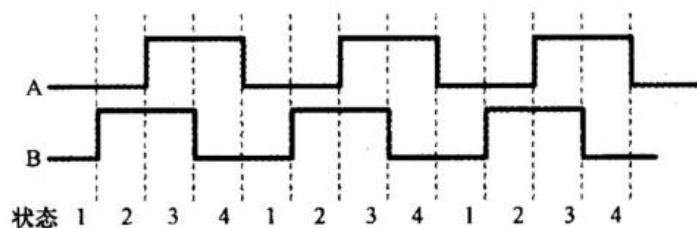


图 2 旋转编码器输出的脉冲波形图

温度传感器

LM35 是美国国家半导体公司生产的，具有很高的工作精度和较宽的线性工作范围，工作电压范围为 4~30V。其精度为 0.5°C，输出电压与摄氏温度成线性比例关系，非线性温度误差低于 ±0.25°C，线性温度系数为 +10mV/°C，无需外部校准或微调。输出电压与温度的表达式为： $U_{out} = 10 \text{ mV}/^{\circ}\text{C} \times t/^{\circ}\text{C}$

LM35 的供电模式有单电源与正负双电源两种模式。正负双电源供电模式可提供 -55°C~150°C 的全量程范围测量；单电源供电模式在 25°C 下电流约为 50μA，温度测量范围 0°C~100°C，其输出电压范围为 0~1V。



图 3 LM35 的供电模式图

控制驱动电路

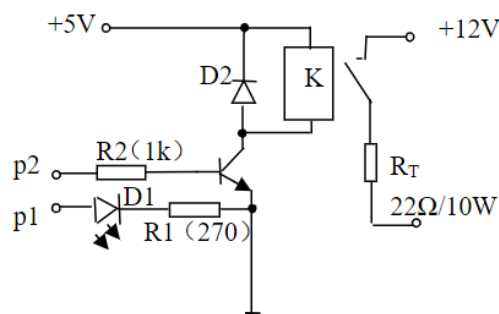


图 4 控制驱动电路

图 2 中，p1、p2 为 Arduino UNO 接口，可自选。二极管 D1 用于指示继电器工作状态，当 D1 亮时，表明继电器断开，开关吸合。二极管 D2 的作用是防止继电器断开时产生的反向电动势的影响。

Arduino 代码

以下将说明 Arduino 代码的核心设计以及我们增加的两种功能：

(1) 设定温度控制

理论上温度设定只是一个点，然而实际情况中，温度控制是一个范围，范围大小取决于设计的精度。本项目设定温度控制范围为 $\pm 0.25^{\circ}\text{C}$ 。当检测温度低于 `setTemp-hysteresis` 时，加热；当检测温度高于 `setTemp+hysteresis` 时，停止加热。

```
float hysteresis = 0.25;
void updateOutputs()
{
    if(override || measureTemp < setTemp-hysteresis)
    {
        digitalWrite(ledPin, HIGH);
        digitalWrite(relayPin, HIGH);
        ride=1;
    }
    else if(!override && measureTemp > setTemp+hysteresis)
    {

```

```

    digitalWrite(ledPin,LOW);
    digitalWrite(relayPin,LOW);
    ride=0;
  }
}

```

(2) 旋转编码器读取

主要读取 A、B 的变化，依据变化输出 `result`。

```

int getEncoderTurn()
{
    // return -1, 0, or +1
    static int oldA=LOW;
    static int oldB=LOW;
    int result = 0;
    int newA=digitalRead(aPin);
    int newB=digitalRead(bPin);
    if (newA != oldA || newB != oldB)
    {
        // something has changed
        if(oldA == LOW && newA==HIGH)
        {
            result = -(oldB*2-1);
        }
    }
    oldA = newA;
    oldB = newB;
    return result;
}

```

(3) 温度控制模式的更改

按下时旋转旋转编码器旋钮，十倍速度调整温度设定值，同时不会影响 override 模式的触发（将消抖改得更加灵敏）。

```

void loop()
{
    static int count = 0;
    measureTemp = adjustUnits(readTemp());
    if(!digitalRead(buttonPin)&&!flag)
    {
        delay(300);
        if(digitalRead(buttonPin))override=!override;
        else flag=1;
    }
}

```

```

int change = getEncoderTurn();

if(!digitalRead(buttonPin)&&flag){
    //十倍速度调整温度设定值
    setTemp = setTemp + change;
}
else setTemp = setTemp + change *0.1;
if(digitalRead(buttonPin))flag=0;
if(count == 1000)
{
    updateDisplay();
    updateOutputs();
    count = 0;
}
count++;
}

```

(4) 温度显示跳变问题的修正

继电器通断时，LM35 处电流会发生突变，导致液晶显示屏的温度示数发生突变，经过反复实验测量，突变前后温差大概为 1.95 度。

```

float adjustUnits(float temp)
{
    if(mode == 'C')
    {
        if(rise==0)return temp;
        else return temp-1.95;
    }
    else
    {
        return(temp*9)/5+32;
    }
}

```

四、原理图

设计原理图如下所示。图 5 为硬件原理图，图 6 为原理图下的模块分布图。原理图的设计中，除了 Arduino 的 5V 供电，还需要一个 12V 的外接电源以供功率电阻加热。

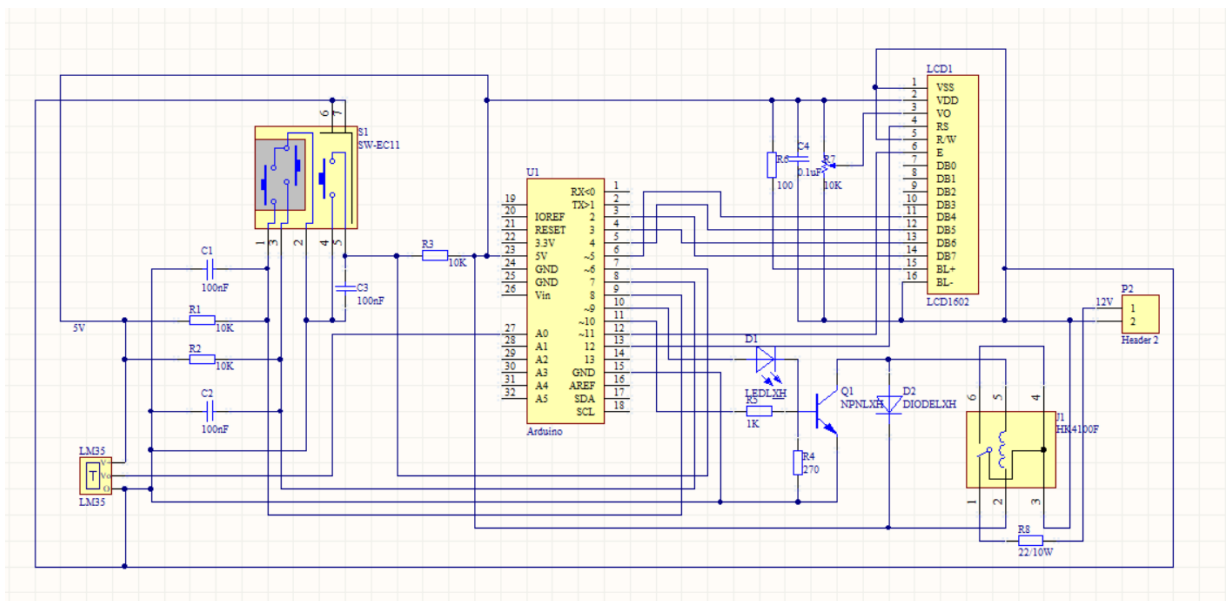


图 5 硬件原理图

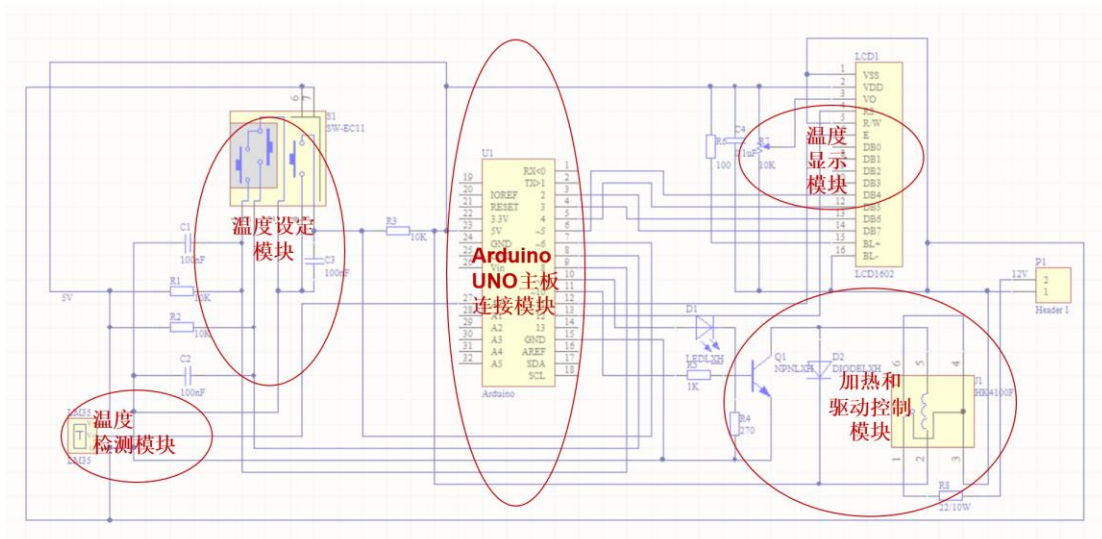


图 6 原理图模块分布图

五、PCB 板图绘制

根据原理图，设计制作印刷电路板如图 7 所示。

为了方便扩展板与开发板的插拔，P1、P2、P3 和 P4 排针位置和间距，必须与 Arduino 主板的扩展插槽完全对准；LCD1602 用来显示温度，布局要合理，并方便读数；温度传感器 LM35 的位置，应靠近功率电阻放置，而功率电阻附近不再布置其它元器件；流经功率电阻的以及与 12V 电源相接的铜线应加粗；为了节约空间，把部分小型器件放置在 UNO 板内的空间内（实际电路中，Arduino 板通过排针插在背面，而其他器件焊接在正面，故不影响）。

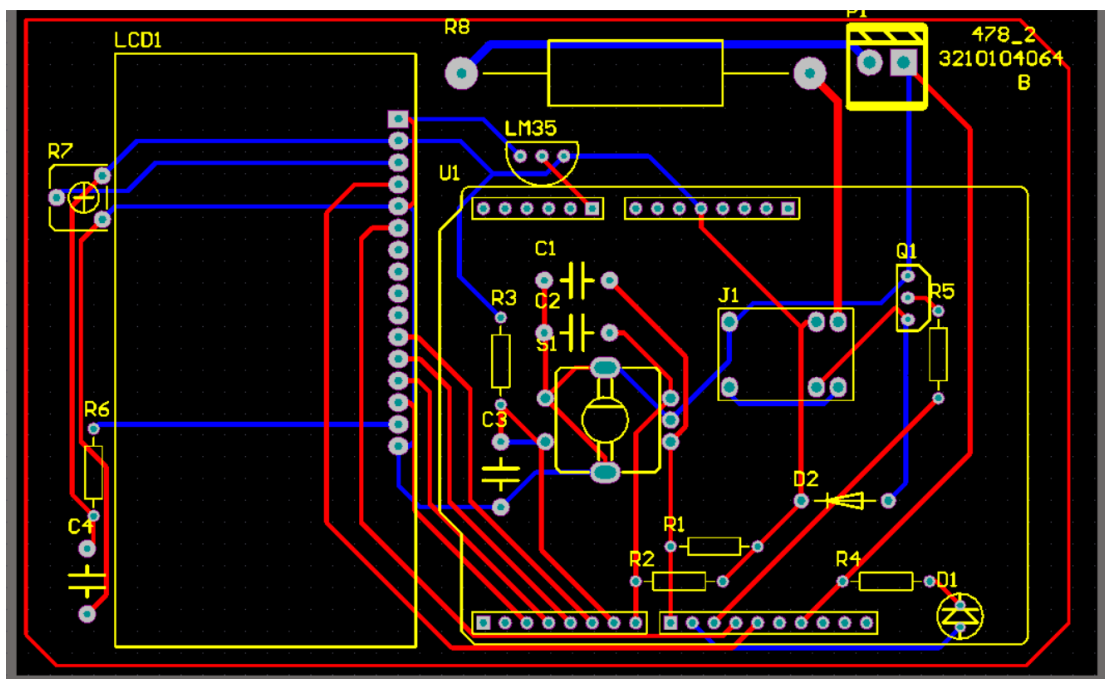


图 7 PCB 板图

六、面包板调试

根据原理图搭建面包板电路，面包板调试如下，可以发现显示屏显示的温度贴合实际温度。通过旋转编码器，能够改变设定的温度，当温度低于设定温度时，功率电阻启动加热。因此电路功能无误，可以上 PCB 板测试。

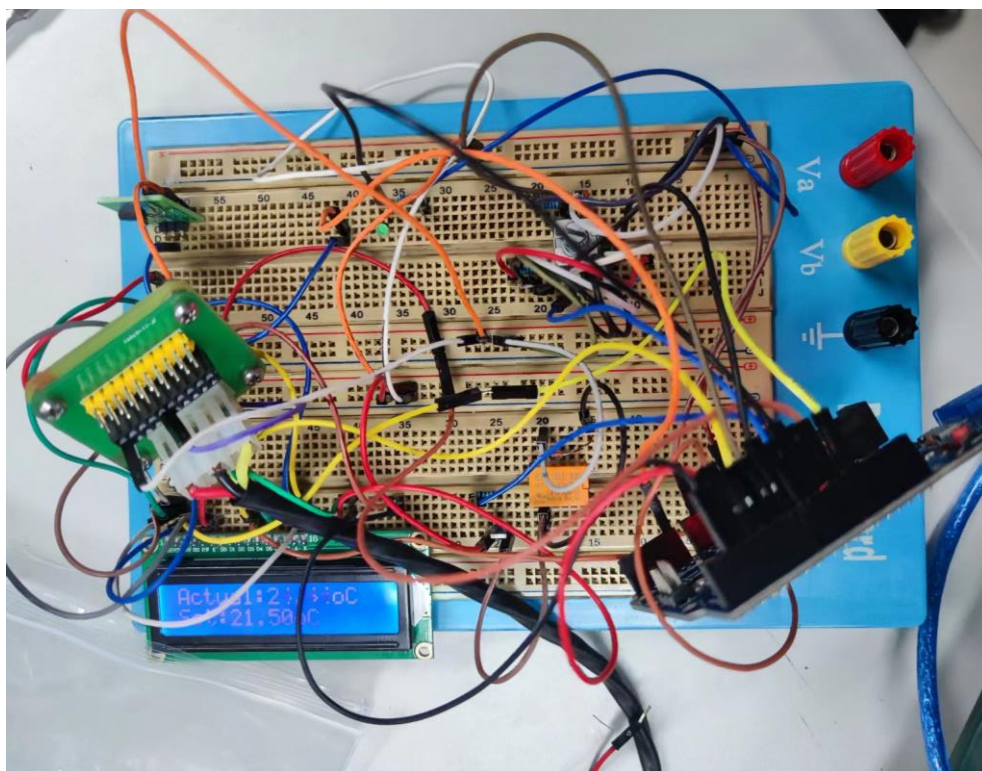


图 8 面包板调试图

七、PCB 焊接与测试

(1) 焊接

焊接图如图 9、10 所示。图 9 为焊接电路板正面图，图 10 为焊接电路板背面图。

由于 LixhSch 库中 LCD1602 的尺寸有误，实际的 LCD 插入插座后占据面积更大，因此我们只能把插座焊在电路板背面，但并不影响实际使用。另外我们在焊接中考虑不周，将功率电阻贴近电路板，导致调试时电路板会受热变黑，但在一定使用时间内不会产生太大影响。

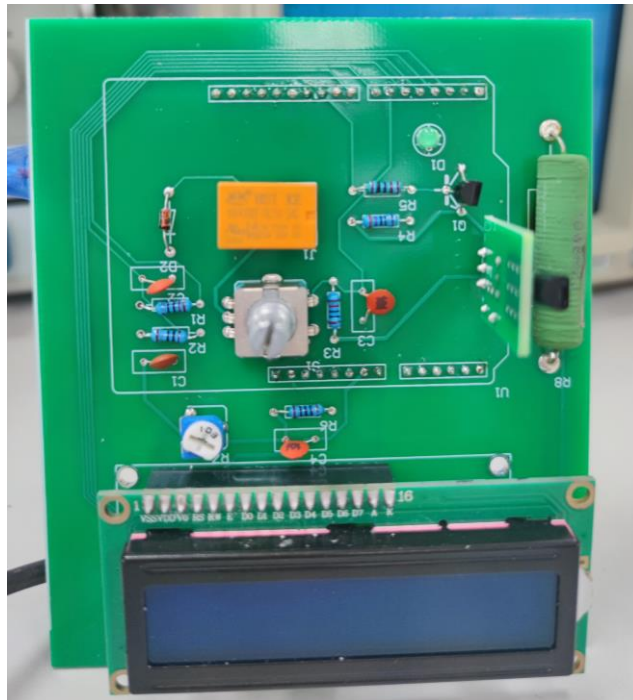


图 9 焊接正面图

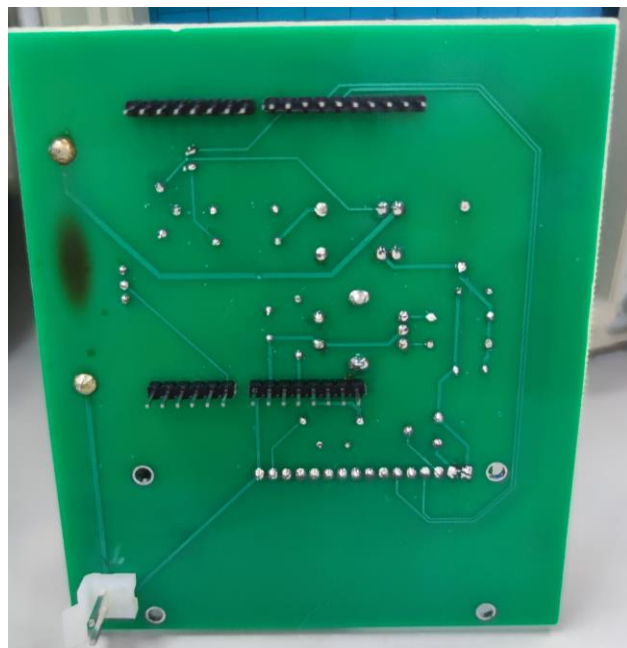


图 10 焊接背面图

(2) 功能调试

a) 测试电源供电

在课程提供的参考原理图中，温度检测器 LM35 需要外接 5V 电源供电，而我们的设计则将 Arduino 的 5V 输出直接提供给 LM35，这可能导致对 LM35 的供电不稳定。当接实验室电源时，Arduino 板输入电压相对稳定，以此条件进行电源供电的测试。

实际测试中，发现 LCD1602 的温度显示稳定且连续，仅在继电器吸合/断开时发生温度跳变，猜测可能是电源供电不稳定导致的问题。

再在 LM35 的输入端焊上铜线（图 11），用实验室电源提供 5V 稳定电压进行对比测试。发现温度跳变有所改善，但是跳变仍有 1 摄氏度左右（图 12-1、12-2），因此应另寻办法调整温度的跳变，那么也不需要 5V 的外接电源供电了。

除此之外的其他供电均工作良好，功率电阻能够正常加热。

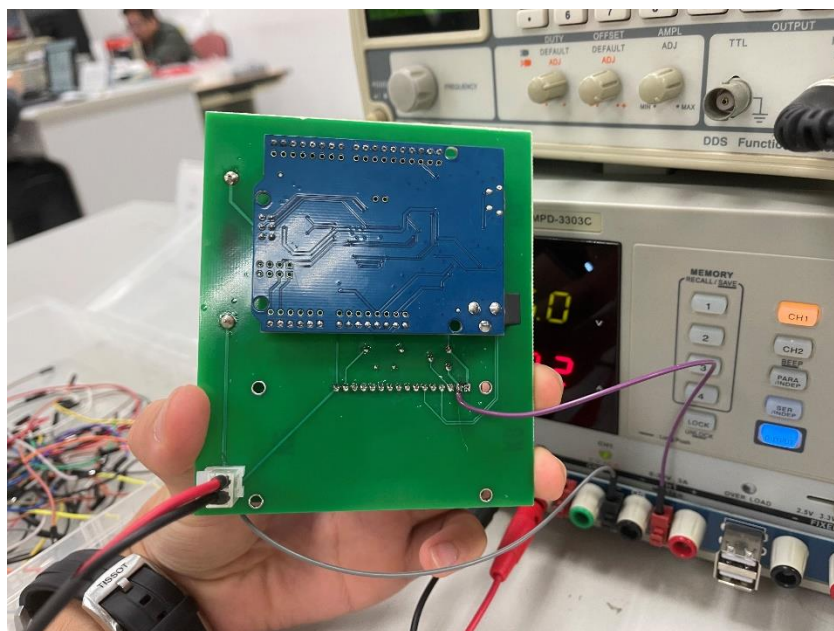


图 11 对 LM35 的供电进行修正

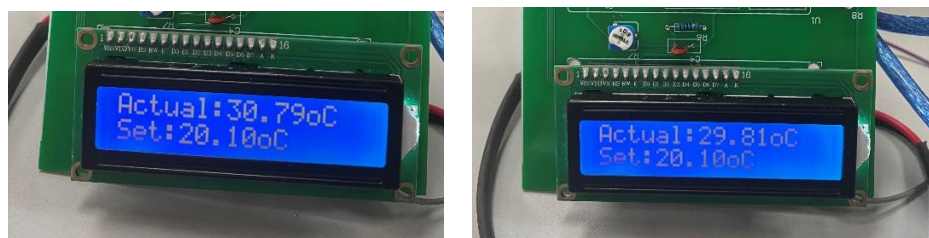


图 12-12、12-2 继电器断开前后的温度跳变

b) 测试温度控制

电路板上电后设定的初始温度是 20.00 摄氏度，温度控制的最小精度为 0.10 摄氏度（图 13）。当按下旋转编码器并旋转旋钮，温度控制的最小精度放大为 1.00 摄

氏度，能够加快温度设定的速度。因此测试结果显示，设计符合要求。



图 13 旋转编码器控制温度设定

c) 测试 override 模式

按下旋转编码器,温度控制进入 override 模式,将不设加热的温度上限(图 14)。

另外，从图 14 还能看到，加热时发光二极管 D1 灯亮，指示电路当前工作状态为加热状态。因此可得，override 模式设计符合要求；电路工作状态的指示功能完备，符合设计要求。

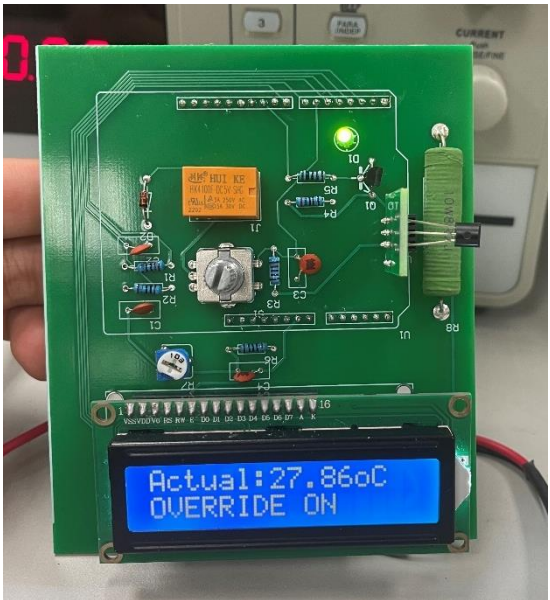


图 14 override 模式图

d) 测试温度显示

在电路板的测试中，我们发现温度显示与实际温度的变化不严格匹配，大概有

0.3s 左右的延时，因此在加热状态下，温度达到设定温度后不会立即停止加热；在冷却状态下，温度低于设定温度时不会立刻开始加热（图 15-1、15-2）。但是功率电阻加热的速度较缓，这段延时对加热/冷却的影响不大，并且可以在代码中设定加热范围参数 `hysteresis` 以进一步降低影响。

另外，我们发现旋转编码器转动太快，有一定概率使显示屏乱码。推测是旋转过快时，显示屏刷新和编码器输入 Arduino 值可能同时发生改变，导致发生类似“竞争与冒险”的情形，使显示乱码。但是我们加入了按下+旋转旋钮的功能，不用很快转动也能实现温度设定的快速调节。

最后经过各方面的修正以后，温度显示不会发生明显的跳变；温度控制范围设置合理，既不会范围太大导致温度过高/过低，也不会太小导致继电器频繁通断；显示屏乱码的概率也得到严格控制。综上，温度显示模块的设计符合要求。



图 15-1 加热至 40.57 摄氏度，电路仍在加热



图 15-2 冷却至 39.10 摄氏度，电路未开始加热

八、讨论总结与体会

在这个为期 16 周的电子电路设计实验 II，我们通过一系列电子元器件设计了温度检测显示控制器，将以前所学习到的模拟电子和数字电子技术充分应用到了实际中，进一步加深了我对模电和数电的理解。通过与队友、课友的讨论，我也补充了很多新知识，学会了画原理图、PCB 版图的各种细节问题；另外通过对比构造系统的不同，我也发现了各自的优缺点，使得我的原理图、PCB 版图得到进一步优化。

这次的电路设计实验主要分成四步：设计原理图、面包板测试、设计 PCB 版图、电路焊接和测试。这几步的实际操作，让我充分应用了数电和模电的知识。

在原理图的设计中，我学会了妥善布局，使原理图尽可能美观简洁，向工程实际用的需求靠拢；在 PCB 版图的设计中，我明白布线和原理图不可能一致，实际接线中有各种各样的条件制约。我学会了布线的各种细节，包括美观、铜线粗细、元器件尺寸对比等。同时要有主动查阅元器件资料、主动学习的心态。

在电路板的焊接和调试过程中，我发现原先面包板的测试并不完美，仍然遗漏了一

些没考虑到的问题，但是好在通过及时的修正和再评估，最终没有产生不可挽回的后果。另外，尽管已经通过 AD 仿真过了，但是电路板的焊接还是遇到了许多问题，比如 LCD1602 的尺寸不匹配等，我们通过积极学习和头脑风暴，妥善解决了这些问题，也锻炼了我们的操作能力，同时进一步增强了自己焊接电路的能力。

硬件方面固然重要，软件的操作也不可忽略。这段经历让我明白，很多在硬件上遇到的问题，可以通过 Arduino 代码加以修正，而不需要每出现一个问题就考虑重新设计电路板、打板。比如温度跳变的问题、温度控制模式的优化等。

另外，课程设计的报告也非常重要。通过报告的编写，我总结了在实际操作中遇到的问题，进一步加深了对项目的理解。

课程设计是培养学生综合运用所学知识，发现、提出、分析和解决实际问题，锻炼实践能力的重要方法，是对学生实际工作能力的具体训练和考察过程。随着科学技术的日新月异，电子电路早已成为当今社会空前活跃的领域，在生活中可谓无处不在，对于信电学院的学生来说，掌握电子电路的开发技术是十分重要的。

感谢中国、感谢党、感谢学校为我们提供了这次电子电路设计实验以及老师的悉心辅导，让我们能够将所学的知识应用于实际，提高了我们的实践能力。这次的课程设计还让我学会了如何培养创新精神，从而不断挑战自己、战胜自己。更重要的是，我们在这段经历中学会了坚持不懈，不轻言放弃。我也发现了很多我的不足之处，让我在以后的学习中有更明确的目标来改变自身的不足，提高自我能力。