上海应用技术大学

**毕 业 设 计（论 文）**

**课题名称：** 温湿度采集和测量系统的Proteus仿真

**学 院** 电气与电子工程学院

**专 业** 电子信息工程

**班 级**  嵌入式 **学号** 1810300204

**学生姓名**  崔兴海

**指导教师** 韩从道

**起止日期**  2022年2月10日－2022年4月1日

**温湿度采集和测量系统的Proteus仿真**

摘要：近年来，湿度与温度这两个值是我们现代生产生活中最重要的参数之一，在日常生活与中起着不可或缺的作用，不仅对人有着密不可分的影响，更是对当今的机器时代有着深远的影响。

例如，从人体温度到数千摄氏度甚至数万摄氏度的工业煅烧和熔化温度，从家用电器的湿度监测系统到大型工业电器，所有方面都与温度和湿度的记录和测量密不可分。

软件部分在Keil上编译开发，采用以C语言为核心单片机所设计出的开发环境语言，通过uVision开发环境实现在线编辑、汇编与调试程序，同时程序设计采用模块化的编程，将主程序划分为若干个模块，然后建立一定的联系，将二者相互协作，完成最终的程序设计。同时以英国LCE公司发行的仿真软件Protues作为总程序的运行平台，将keil生成的hex运行进行单片机的仿真。

本课题共分为六个模块：main主函数模块、STH11温湿度传感模块、1602LCD液晶屏显示模块、按键KEY模块和报警蜂鸣器和LED模块。

课题的具体过程，大致是：先在仿真软件protues中绘制模拟原理图，再凭靠SHT11温湿度传感器，感测及测量的环境的气温与湿度，然后再使用我们的STC89C51芯片，读取所采集的数值，再利用算法（补偿函数）处理修正后，再使用LM016液晶显示器，即时显示所检测到的温度数值。然后，设计一种可以设置、添加、减少、退出的按键模块，来设定和管理温度和湿度数值的上限和下限。当感应器所读取的温度和湿度不在我们可以手动设置的数值范围之内时，蜂鸣器和相应的LED会显示报警信号。但当STH11读取的温度与湿度值，为在设定阈值内时，将相应的LED报警灯和蜂鸣器beep关闭。

关键词：C语言；STH11温湿度传感器；keil；protues

**Proteus Simulation Of Temperature And Humidity Acquisition And Measurement System**

**Abstract**: The two values of humidity and temperature are one of the most important parameters in our modern production and life. They brought not only an inseparable influence on humanity, but also a great influence on the modern mechanical age.

For example, from human body temperature to industrial calcination and smelting temperatures as high as 1000 degrees Celsius or even tens of thousands of degrees Celsius, or from humidity monitoring systems of household appliances to large industrial equipment, temperature and humidity are inseparable from the collection and measurement of temperature and humidity. Therefore, this topic is devoted to the study of temperature and humidity acquisition and measurement system.

The part is compiled and developed on Keil, using C language, which is more readable, easy to understand and easy to transplant, as the single-chip computer development language. It is developed through uVision and adopts modular programming. At the same time, the simulation software Protues issued by THE British LCE company was used as the running platform of the general program, and the HEX generated by Keil was run to simulate the single chip computer.

sht11 is used to detect and measure the temperature and humidity of the environment, and STC89C51 is used to read the collected data and process the correction algorithm, and LM016L LCD screen is used to display the measured data in real time. Then, a set, increase, decrease, exit button module is designed to control the upper and lower limits of temperature and humidity LED and buzzer alarm. When the temperature and humidity read by the sensor are not within the range set manually, the corresponding buzzer and LED will prompt warning messages. When STH11 reads normally, the corresponding LED lights and buzzers turn off.

**Key words**: C language; STH11 sensor; Keil; protues

**目录**

[1. 系统概述 1](#_Toc18639)

[1.1. 课题研究目的与意义 1](#_Toc9883)

[1.2. 课题研究的主要内容 1](#_Toc15667)

[2. 系统总体设计 2](#_Toc16227)

[3. 硬件设计 3](#_Toc9495)

[3.1. 单片机最小系统 3](#_Toc13476)

[3.1.1. STC89C51介绍 3](#_Toc29621)

[3.1.2. ROM,RAM和FLASH在单片中的作用 8](#_Toc7754)

[3.2. LCD1602液晶显示模块 11](#_Toc5214)

[3.2.1. LCD1602的介绍 11](#_Toc2329)

[3.2.2. 引脚功能说明 12](#_Toc15541)

[3.2.3. LCD1602指令集说明与代码 13](#_Toc26725)

[3.2.4. 1602 的基本操作时序 17](#_Toc24053)

[3.3. 数字温湿度传感器SHT11  19](#_Toc28373)

[3.3.1. SHT11介绍  19](#_Toc5430)

[3.3.2. 测量时序 22](#_Toc19968)

[3.3.3. 输出转换为物理量 26](#_Toc15116)

[3.4.  按键模块  28](#_Toc7198)

[3.4.1. 工作原理 28](#_Toc29944)

[3.4.2. 按键模块的设置 29](#_Toc22571)

[3.4.3. 按键消抖 33](#_Toc32640)

[3.5 报警模块  34](#_Toc2087)

[3.5.1. 蜂鸣器报警模块 34](#_Toc24044)

[3.5.2. LED报警模块  35](#_Toc2655)

[4. 硬件设计 37](#_Toc17652)

[4.1. Keil软件开发平台 37](#_Toc14809)

[4.2. 主程序设计 37](#_Toc32746)

[4.3. SHT11程序设计 38](#_Toc18276)

[4.4. LCD1602程序设计 39](#_Toc19491)

[4.5. 报警模块程序设计 40](#_Toc29651)

[5. 仿真与测试 41](#_Toc17574)

[5.1. Protues设计平台 41](#_Toc32113)

[5.2. 仿真结果分析 41](#_Toc2164)

[致谢 48](#_Toc25008)

[参考文献 49](#_Toc22497)

[附录 50](#_Toc23958)

# **系统概述**

* 1. **课题研究目的与意义**

在现代工业生产的快速发展和进步过程中，温度和湿度两个物理量的测量和监测也进入了自动化时代辅助，从微计算机所具有的高集成性、体积小易携带、成本低、实时实用性强等优点来看，广泛用于实时控制系统。因此，可以作为先端兵器的开发、机器人和人工智能等开发平台，我们使用单片机研究开发。

基于STC89C51单片机的温度湿度采集和控制系统，是监测和控制现场环境温度与湿度，可以按照既定精度控制、监控和采集当前环境的温度和湿度。

而新型的温湿度传感器STH11具有长期稳定性，全标定输出等优点，在户外成为测量温度与湿度数据的不二人选。

* 1. **课题研究的主要内容**

本课题共分为六个模块：main主函数模块、STH11温湿度传感模块、1602LCD液晶屏显示模块、按键KEY模块和报警蜂鸣器和LED模块。

课题的具体过程，大致是：先在仿真软件protues中绘制模拟原理图，再凭靠SHT11温湿度传感器，感测及测量的环境的气温与湿度，然后再使用我们的STC89C51芯片，读取所采集的数值，再利用算法（补偿函数）处理修正后，再使用LM016液晶显示器，即时显示所检测到的温度数值。然后，设计一种可以设置、添加、减少、退出的按键模块，来设定和管理温度和湿度数值的上限和下限。当感应器所读取的温度和湿度不在我们可以手动设置的数值范围之内时，蜂鸣器和相应的LED会显示报警信号。但当STH11读取的温度与湿度值，为在设定阈值内时，将相应的LED报警灯和蜂鸣器beep关闭。

1. **系统总体设计**

软件部分在Keil上编译开发，C是一种通用的开发语言，通过uVision开发环境在线编译、汇编与调试程序，同时程序设计采用模块化的编程，将主程序划分为若干个模块，然后建立一定的联系，最终的程序设计是由两者共同配合完成的。同时以英国LCE公司发行的仿真软件Protues作为总程序的运行平台，将keil生成的hex运行进行单片机的仿真。

本课题共分为六个模块：main主函数模块、STH11温湿度传感模块、1602LCD液晶屏显示模块、按键KEY模块和报警蜂鸣器和LED模块。

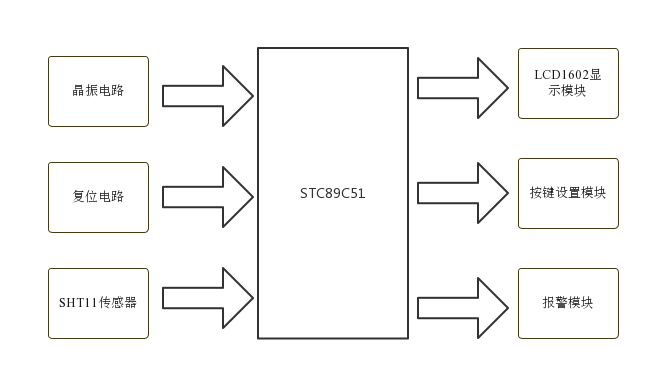
方案的总体框架，如下图2.1所示。

图2.1

1. **硬件设计**
   1. **单片机最小系统**
      1. **STC89C51介绍**
2. 概念

STC89C51是宏晶公司开发的一款51芯片，是具有高速、耗电小、主低价、高性能及防干扰特性的CMOS 8单片机芯片，内嵌ISP芯片，不需要专用编辑器。

1. 配置

* 具有4K的存储空间的Flash，内存的一种，闪存特点，采用分页存储，这也是EEPROM的一个很大的区别，不必离开系统，“ISP通过系统编程”。闪存驱动器flash不易失，其与51的指令和引脚基本适用。
* 具有AD、PWM（脉冲宽度调制）、SRAM、UART（通用异步串行端口）、SPI（同步通信端口）等模块
* EEPROM功能。
* 看门狗功能
* 专用复位电路

1. 其他定义

* 标准51芯片的周期分为六时钟增强型（晶体振荡器频率/6）和十二时钟普通型（晶体振荡器频率/6）。可以选择；
* 工作频率：在0到40 MHz；
* Stc的Flash空间：4\8\15KB；
* 内部存储器（RAM）：512B；
* 计时器\计数器：3个16位；
* 一个UART；
* 中断源：8个；
* 内置ISP和IAP（不需要专用编译器或模拟器）；
* 通用IO口：32\36个；
* 工作电压：3.4~5V；
* 封装：例如40个PDIP、44个PLCC、PQFP等。

1. 引脚

STC89C51芯片带VSS/VCC共有40个引脚，采用包括6条控制信号线、2条电源线、地线、4个输入输出接口的2列直插结构。详情请参照下图3.1.1。

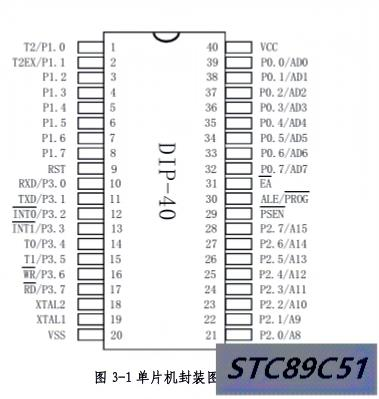


图3.1.1

引脚功能和说明：

（1）VCC：电源电压引脚

（2）GND：地引脚

（3）P0端口： ，PO口上电复位后是开漏输出。PO口作为总线扩展用时，不用加上拉电阻，作为L/0口用时，需加10K-4.7K上拉电阻。

（4）P1口： 上电复位后为准双向口/弱上拉（传统8051的I/O口）模式。

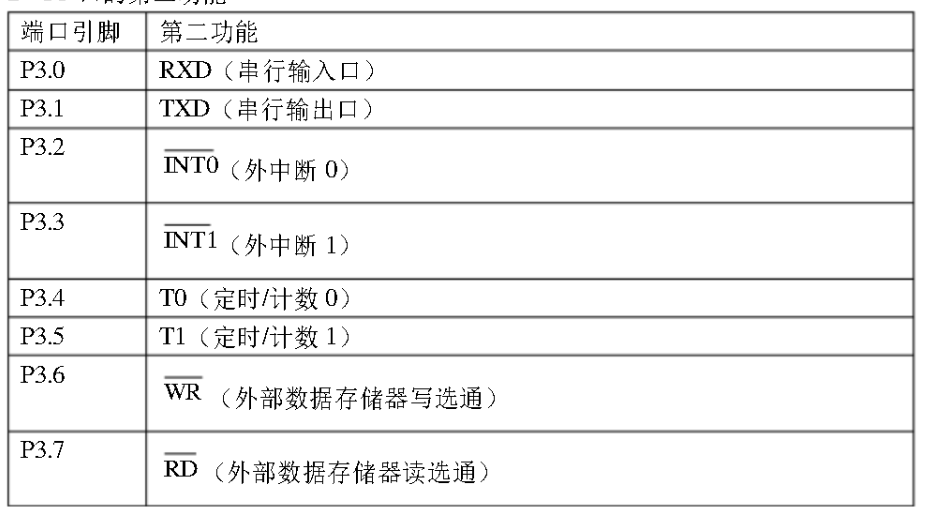
另外，部分引脚还有第二种功能，在STC89C51的芯片中与AT89C51芯片不同的是P1端口也可以作为定时器或计数器使用。如图3.1.1，请参照表格。

|  |  |
| --- | --- |
| 引 脚 号 | 功能特性 |
| P1.0 | T2（定时/计数器2），时钟输出 |
| P1.1 | T2EX（定时/计数2） |

（5）P2口： 上电复位后为准双向口/弱上拉（传统8051的I/O口）模式。

（6）P3端口： 上电复位后为准双向口/弱上拉（传统8051的I/O口）模式。

另外，P3引脚，分别还有第二种功能，参照图3.1.2。其第二功能的使用 在单片机中的程序设计中更重要。

图 3.1.2

（7）RST：检索数据复位。启动STC的振荡器后，当第一个复位信号给予超过两个机器周期时，MCU将恢复，检索数据复位。

（8）EA/VPP：外部储存器的选择信号许可。

（9）XTAL1：片内振荡的in。

（10）XTAL2：片内振荡器的out。

（11）数据存储器：STC芯片的RAM为256K。

（12）中断：

INT0和INT1，T0，T1和T2，串行端口中断。

（13）时钟振荡器：

外部的晶体振荡器或陶瓷共振器，以及电容器C一和电容器C2，构成包含了反馈和放大滤波电路在内的并联振荡电路。如图3.1.3所示。



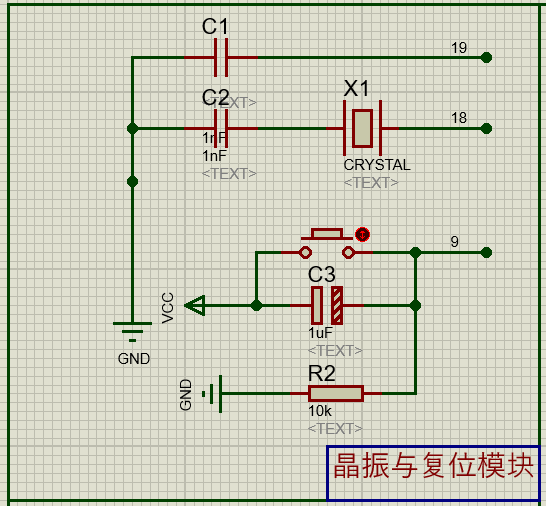
图3.1.3

外部晶振或陶瓷谐振器，和电容器C1和C2，形成了反馈放大器的并联振荡电路 。

第二，使用外部时钟。无需确定外部时钟信号的工作周期，但最短和最长的高电平周期和最长的低电压周期必须满足反时限产品的技术条件。

（14）晶振电路

对于芯片STC89C51的所有命令操作的过程对于程序需要明确的顺序，这也被称为时序，MCU需要生成时钟信号。电路如图3.1.4 所示。

图3.1.4

外部时钟信号，通过连接XTAL1和XTAL2引脚上的振荡电路产生此信号。通过添加两个22pf电容器，来启动振荡器，以调整和控制振荡频率。同时，时钟周期通过12m晶体振荡器的连接来确定。时钟周期是单片机的基本单位，时钟周期=1/振荡器频率，互为倒数关系。

（15）复位电路

集成MAX810 传专用复位电路。微控制器需要在动作开始时复位，整个系统电路保持初始状态以启动动作并确保初始动作。芯片的第九个引脚RST在此引脚接收两个1s，并在机器循环时开始复位。STC89C51RC/RD+系列单片机有4种复位方式∶外部RST引脚，软件，掉电上电，看门狗复位。电路连接如图上3.1.4 所示。

(16）上拉电路

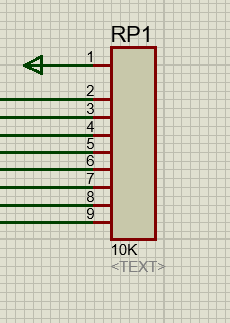
与其他三个输入输出口不同，P0端口的内部结构，与其他三个端口不同。由于用于一般I/O端口时，P0端口中没有上拉电阻，因此输出驱动电平不是输出“1”和“0”，而是漏极开路。因此，必须通过电路手动连接外部上拉电阻。如图3.1.5所示。

图3.1.5  上拉电阻电路

* + 1. **ROM,RAM和FLASH在单片中的作用**

1. 介绍

* ROM

保存固化程序（保存命令代码和一些在程序运行后无法更改的固定值）。

例如，ROM将ASM文件调用代码存储在ROM中。例如，所有C和H代码、全局变量、局部变量、变量定义的常量、arm或x86 BIOS命令等。



* RAM

储存空间储存程序在其运行期间，对数据或者命令的随机访问。（断开连接会导致数据丢失。）将所有程序重新写入RAM。例如，全局或者局部变量、堆、栈。

* FLASH

Flash主要储存及保存必须需要永久存储，可更改的用户程序和数据。

例如，电子电度表的核心是一台单片微型计算机，它可以在屏幕上记录程序。U和I是我们不感兴趣，使用不到的当前数据。它只用于计算能量。Flash运行完成后，收集的数据将被清除，然后再次进行赋值。被清除的数据通常就放在前面讲的RAM里面，与FLASH不一样，数据不是永存的。当前的计算必须永久保持。单片微型计算机会在一定时间内或断电时使flash保持开启状态。

1. 在单片中的运作原理

1.擦除，内部写数据，储存可执行文件；

2.使用特殊的刻录软件，通过烧录机将hex文件刻录到ROM中

注：此时，如果ROM储存空间包含程序的所有代码或者元素，例如启动代码、函数的局部变量、文件头中指定的全局变量、指定的const常量等。

1. 疑问：

由于所有数据都在ROM中，RAM中的数据如何运行？RAM怎么加载的？在记录过程中，需要放在RAM中的数据是否可能包含在RAM中？

答：

1. ROM权限设为只读。和RAM不同的是，RAM处理器不仅可以读取数据，还可以保存数据。停电时不会存储数据。
2. 在数据不会写入，因为在运行期间和之后必须关闭电源。当MCU重新启动时，处理器工作正常，RAM中仍有数据。下载意味着不写入RAM数据，处理器数据写入RAM。
3. ROM包含程序的所有命令或者数据。当MCU打开时，处理器执行第一行中的代码命令。现在整个项目已经成功运行，初始化已经准备就绪。
4. 工作任务有几项

（1） 为全局变量分配地址空间

（2） 设置堆栈段的长度和地址

（3） 输入初步地址数据、常数和代码段

由于代码片段和常量的地址可以不允许在程序中，无论他们如何组织，他们对该项目都没有影响。但是，必须指定数据段的地址。Ram由一个数据段、一个批处理段和一个分开的操作团队组成。通常输入团队的地址。这意味着，如果一个数据段确实可以找到一个地址，它就不能覆盖任务组中的所有地址。当然，它值得认真考虑。

注意：这里描述了“第一行代码”。这不一定是一个单独的程序代码。大多数是添加到编译器中的文本操作或表示文件。因为，自己写的C程序不会去写冗长的头文件，我们程序员本身只需要会使用这些文件即可。高级MCU位于启动文件中。

（4） 通常，MCU闪存存在，并在启动或更换“0000”时存储在个人光标中。这意味着处理器开始执行100000个ROM地址的命令，并保存每个命令对程序基本功能的更改。如果在这期间发生中断，系统会将指定中断向量表中的命令添加到中断向量，然后切换到中断工具。单片机处理器，会自动设置配置和分配空间ROM。

注：特殊情况如下：

1端口寄存器：这里面存的代码呢，也是一个特色的变量。其他变量不能在系统可用的特殊RAM地址中定义。

2中断向量表：断点向量表将MCU ROM地址分配给不同的断点。调用所有中断服务，并在中断向量表中输入程序地址。

1. ROM的大小疑问

对于MCU闪存，ROM通常为10000字节。一些OTP微控制器，如Holtek和Sonix，“rom2k-15位OTP程序……”数据库。这15位允许多个小于2的字节。ROM空间呢，一般在2K左右，也可以大于2K或4K，作为他的ROM。这二者具体的定义，可以这样理解，有点类似，一个是我们写文章的宽度，另一个是我们写文章的长度。导轨的宽度表示导轨位置的宽度。宽度为8和15位。句子的长度，指的是每个句子中使用的，格的多少或者大小。由1、2、3和4个字节组成。也就是说，在拆卸或组装之后，复合指令和简单指令集成在一起。

* 1. **LCD1602液晶显示模块**
     1. **LCD1602的介绍**

近年来，LCD（液晶显示器）的使用越来越频繁。Lcdl602是最常见的LCD类型之一。每行可以显示两行16个字符的ASCII字符。与OLED相比，lcdl602具有明显的价格优势，但缺点是无法显示一组符合国家标准的汉字。因此，602主要用于显示英语语言和数字，因此其在中国的分布有限。

HD 44780 LCD控制器（和兼容电路）有三种类型的内存：DDRAM、CGROM和CGRAM。DDRAM被称为数据映射斜，用于存储当前显示的代码。每个DDRAM元素对应于屏幕上每个字符的地址。DDRAM的容量为80字节，但只有32字节。通常，lcdl602显示两行字符。此时，LCD屏幕上从左到右的第一行16个字符的地址码为0x00-0x0f（对应的十进制为0-15），从左到右的第二个码为0x40-0x4f（对应的十进制为64-79）。这样，操作LED屏幕上的字符显示实际上就是将字符代码写入DDRAM地址的结束。

CGROM被称为字符生成ROM，其功能是存储不同字符的代码。这些代码包括英文大写和小写字母、阿拉伯数字和常用英文标点符号。事实上，这部分代码与ASCII代码基本一致。此外，CGROM中还有日语假名代码。对于一般应用，LCD控制器的主要工作是从CGROM中删除显示的字符代码，并将其写入与LCD屏幕上32个字符位对应的DDRAM地址。LCD控制器在MCU的指导下工作

显示温度和湿度通常需要较长的显示代码，在输入输出交流中，通用输出方式，通常采用数码管、LED发光管、LCD液晶画面。

而该系统使用Protoues中的元件LCD 1602作为显示模块。LCD1602是点矩阵型液晶模块。在日常生活中你可能没有仔细发现，但我们经常使用LCD显示器。目前，LCD已经成为市场上许多电子产品的显示设备。例如手表、电脑、万用表、许多家用电子产品等。另一方面，作为一种系统芯片输出器件，它具有质量好、接口数字化、体积小、重量轻、功耗低等优点。另外，LCD 1602可以直接连接到我们的单机系统接口，只需可以添加上拉电阻。如图3.2.1所示。

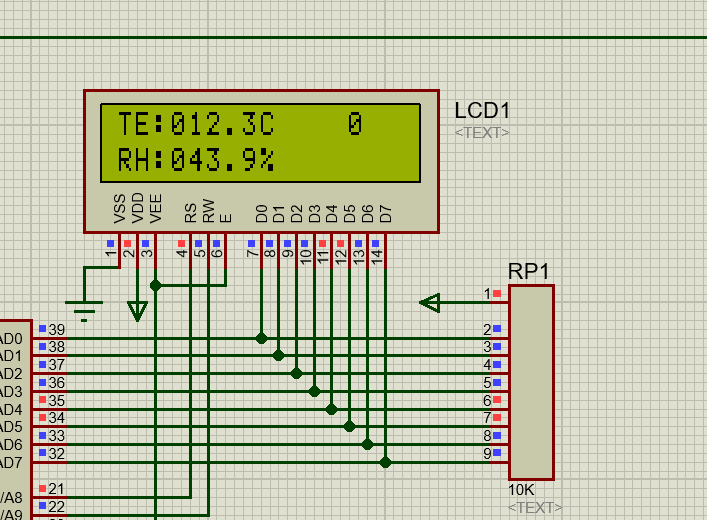


图3.2.1

LCD1602主要技术参数，如图3.2.2.

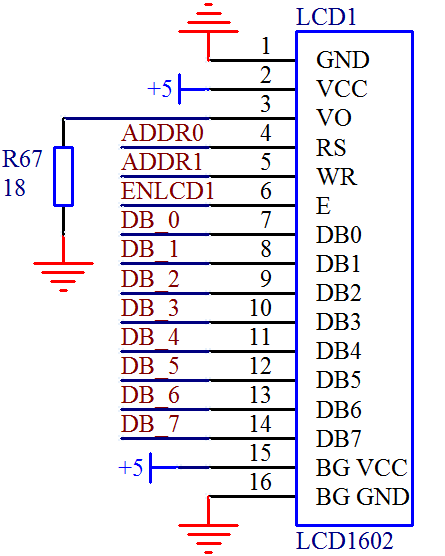


图3.2.2

LCD1602引脚功能如表3.2.3所示



图3.2.3

* + 1. 引脚功能说明

引脚如图3.2.4。

1. 3 引脚（VO）叫做液晶显示偏压信号

Vo用于调整色块之间的对比度，使对比度对于我们人眼来说，在膜上成像地显示更清晰。链设计测试中常用的方法是电位计（例如防滑）连接3引脚。电位器用于间接控制LCD屏幕预充电和调节电压之间的电压。产品量产后，可以直接使用调整后的整数值。在protues的仿真电路中，使用18Ω的电阻，当然也可以使用1-15k的电阻来调整电压。

1. 4 脚是数据命令选择端

图3.2.4

可以发送命令，达到我们想要的状态，发送数据并显示。LCD 1602，液晶屏通过这个引脚，来判断命令是发送还是数据发送，4针与ADDR0连接，并且连接了跳线和P1。

1. 5 脚读写选择端

和4一样，LCD可以通过5键读取数据和状态。由于LCD屏幕本身有ram，LCD命令和数据必须缓存并写入寄存器和ram。当然需要时间。但是，必须在读取或写入之前读取LCD的当前状态。在下面会介绍，如果不忙，读写数据。如果“忙”，则必须等待LCD不被占用，然后重新读取数据。可以读取常用状态。但是我们几乎不使用读取LCD数据，理解就可以了。

1. 6 脚是使能信号

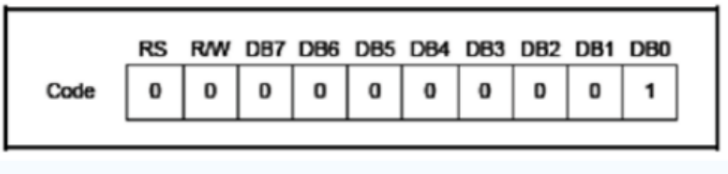
6脚是非常重要的PIN，液晶的读写命令和数据，全部依赖于此才能正常读写。计为经由跳线连接到ENLCD，并且实现与12864液晶之间的切换使用。

1. 7 -14 引脚就是 8 个数据引脚

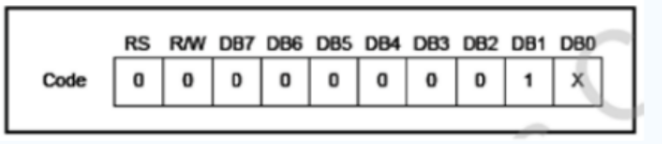
用于读写数据命令。

* + 1. LCD1602指令集说明与代码

1. ‬指令1：作用清显示，如果写命令，0x0（0x0000 000，,如图，光标恢复到地址00H位置；;如果写入 0x02（0x0000 0010） 则仅仅是数据指针清零，光标显示不清零。



1. 指令2：如图，光标复（(0x02 或者0x03都）)，光标显示返回到00H。

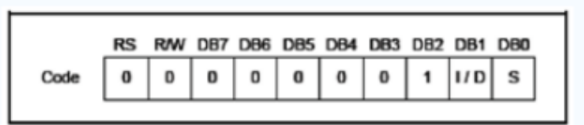


1. 指令3：光标和显示模式设置

I/D：决定着光标往哪移动，如果1往右，如果0往左。

S ：整屏移动。1表示有效，0则无效 。

光标或整屏显示移位 （0x0000 01xx）



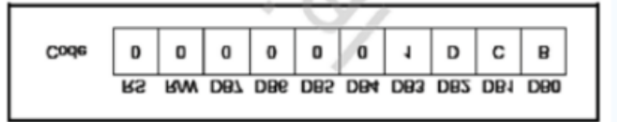
I/D=1 决定读或者写后，指针加 1，光标加 1；

I/D=0 表示读或者后，指针减 1，光标减 1；

S=1 表示写后，如果N=1整屏，,如果N=0右移，为了防止光标移动，屏幕的移动效果与计算机指定的相同。

而 S=0 表示写后，整屏显示不移动。

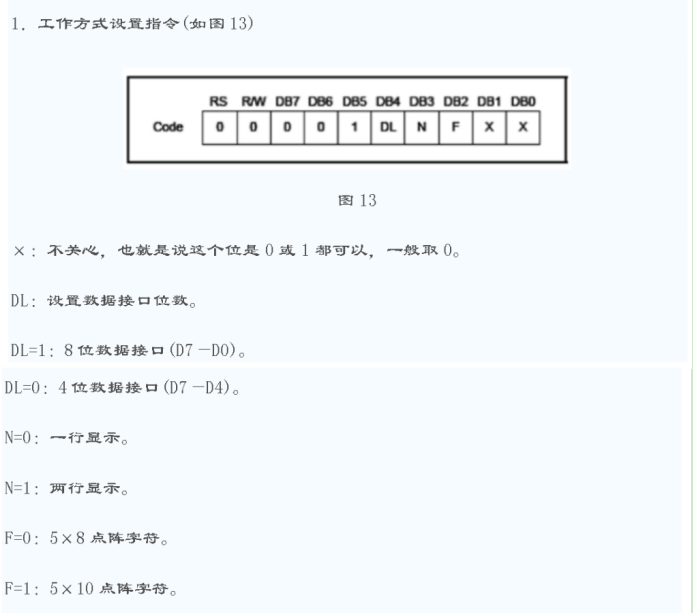
1. 指令4：显示开关与光标控制。 DCB



D：控制整体显示的开：:1开0关 。

C：控制光标的开与关：1开0关 。

B：控制光标是否闪烁，1闪烁，0不闪烁。

1. 指令6：功能设置命令

DL：高电平时为4位总线，低电平时为8位总线 ；

N ：低电平时为单行显示，高电平时双行显示 ；

F ：为0时，五行七列，为1时，显示五行十列；

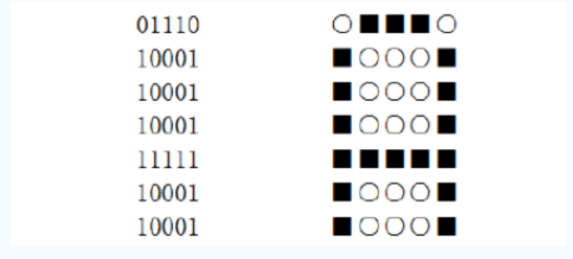
例如0x38:0011 100//设置显示器的16位8位数据接口。2行5x7（注意：该命令只允许RS和rw-0.lcd 1602并行运行，而不是串行运行）。

1. 指令7：字符发生器RAM地址设置

二进制码中第一行首地址为10000000转换16进制为0x80 第二行首地址为11000000转换16进制为0xC0 加首地址是为了让LCD显示从第一位开始显示。



如下图，在上面我们设置了5x7的字符点阵显示。如果我们要显示ASCII码中的大小字母“A，,从黑方块的形状我们可以看出A的具体模型，然后剩下的地方我们用白色圆圈补上，然后黑色方块换成1，白色圆圈换成0，如作图，这就是我们的一个字符表示。上面的4位字符是0100，下面的4位字符是0001。加在一起是01000001b，即41h。它与字符的ASCII码随机重合，这给了我们极大的方便。我们可以在PC上使用语法P2=“A”。编译后，它就是字符代码。



1. 指令8：DDRAM地址设置
2. 指令9：读忙信号和光标地址

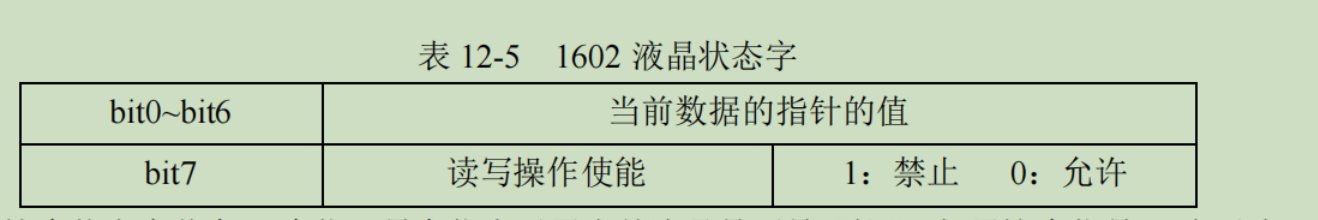
BF：是忙数据流的标志位

1. 指令10：写数据
2. 指令11：读数据
   * 1. 1602 的基本操作时序

1、读状态：RS=0，R/W=1，E=1。直接写：

1. LCD1602\_DB = 0xFF;
2. LCD1602\_RS = 0;
3. LCD1602\_RW = 1;
4. LCD1602\_E = 1;
5. sta = LCD1602\_DB;

读取LCD变量“sta”的当前状态，可以判断或计算Sta上限的值，来显示当前位置或当前数据光标是否处于“忙”。但有两个问题：

问题一：如果当前状态为“不忙”，程序可以继续读写。如果当前状态为“正在使用”，则必须等待重新评估LCD状态。

LCD 1602\_DB数据位为8位，如上图。如果该级别为1，则使用LCD屏幕为“忙”，禁用数据和命令的读取和写入。为0可以读写。当前数据指针的值，是由较低的7位bit0-bit6表示。

问题二：电路总线用于LED、数码管、点阵和1602 LCD。当读取LCD 602时，LCD1602仍处于E is 11602的状态\_Bit7，LCD继续输出其状态值。输出占用P0总线状态值，这可能会影响其他外围设备的输出。这个问题的解决方案通常是降低引脚并解决总线问题，使用do.while{}。

读数据不经常使用，尽了解。

1. 写指令：RS=0，R/W=0，D0~D7=指令码，E=1。

前面三个指令过后，E使PIN能够从0变为1，然后E使得PIN再回去产生一个下降沿。然后，一旦在LCD1602内检测到下降沿，并且同时检测到如下指令：

LCD1602\_RS = 0;

LCD1602\_RW = 0;

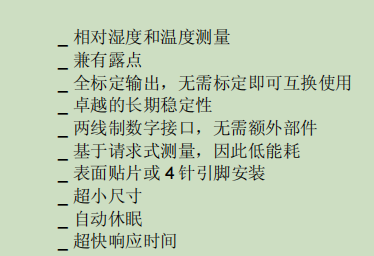
这两个命令，就立即读取数据d0-d7，并准备1602写入命令报告。

4、写数据：RS=1，R/W=0，D0~D7=数据，E=1。

写数据和写指令差不多，就是把 LCD1602\_RS = 1，把总线改成数据即可。

* 1. 数字温湿度传感器SHT11
     1. SHT11介绍

1. STH11优点如图：



1. 传感器组成

它由一个14位串行温度测量元件和与聚合物无缝连接的电容传感器组成。该传感器包含一个用于湿度测量的电容式聚合物测量元件和一个温度测量元件，该元件由一个能量分配材料组成，该材料通过14位D转换器和串行接口连接到同一个晶体上。

1. 应用领域：

* 暖通空调 HVAC
* 汽车
* 消费品
* 气象站
* 湿度调节器
* 除湿器
* 暖通空调 HVAC
* 汽车
* 消费品
* 气象站
* 湿度调节器

SHt11是全数字温湿度传感器，具有串行双线接口，如图3.3.1所示。

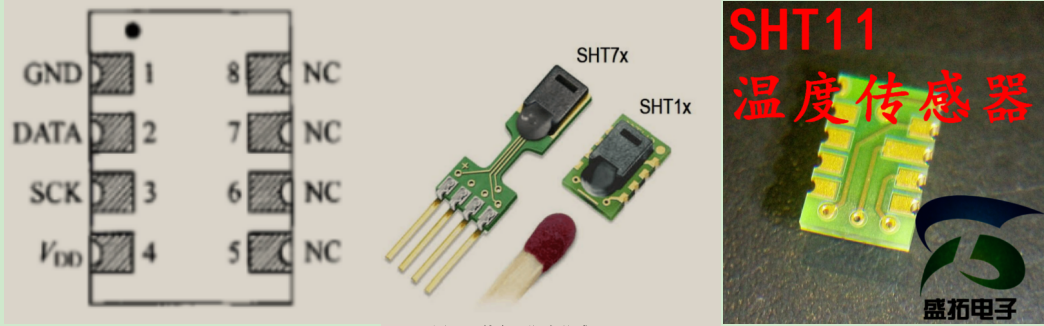


图3.3.1

1. 分辨率

在SHT11中，温度测量数据的分辨率为14位，但可以通过状态寄存器切换到12位；湿度测量数据的分辨率为12位，但可以通过状态寄存器切换到8位。

参数和内部框图如图3.3.1 所示。

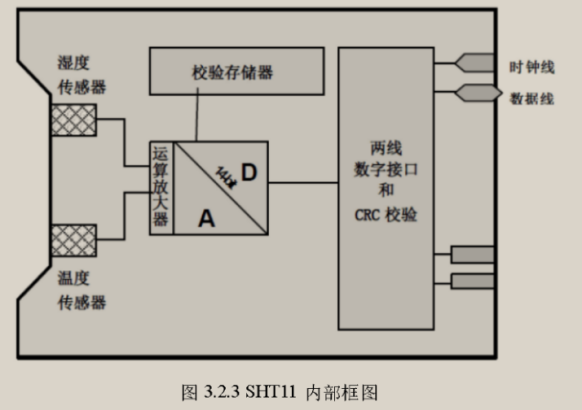
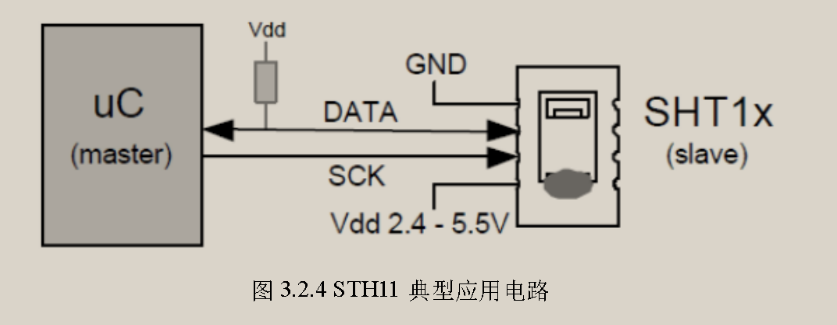


图3.3.1

1. 工作原理

电路sth11如图3.2.4所示，引脚sth11如图3.2.5所示。

1. 接口说明
2. 电源引脚：

shtxx的电源电压为2.4~5.5V。传感器通电后，等待11毫秒进入“睡眠”状态。在此期间无需发送任何指示。可在电源引脚（VDD、GND）之间添加100nF电容器，以实现解耦和滤波。

1. 串行接口 （两线双向）

Shtxx的串行接口在传感器信号读取和功耗方面进行了优化；但它与I2C接口不兼容

1. 串行时钟输入 (SCK)

SCK用于微处理器和shtxx之间的通信同步。由于接口包含完全静态逻辑，因此没有最小SCK频率。

1. 串行数据 (DATA)

数据三状态门用于数据读取。数据在SCK时钟下降沿后改变状态，仅在SCK时钟上升沿有效。在数据传输过程中，数据必须在SCK时钟高电平下保持稳定。为了避免信号冲突，微处理器应以低电平驱动数据。需要外部上拉电阻器将信号拉至高电平。上拉电阻器通常包括在微处理器的I/O电路中。详细的IO特性。

* + 1. 测量时序

时序如图3.3.2

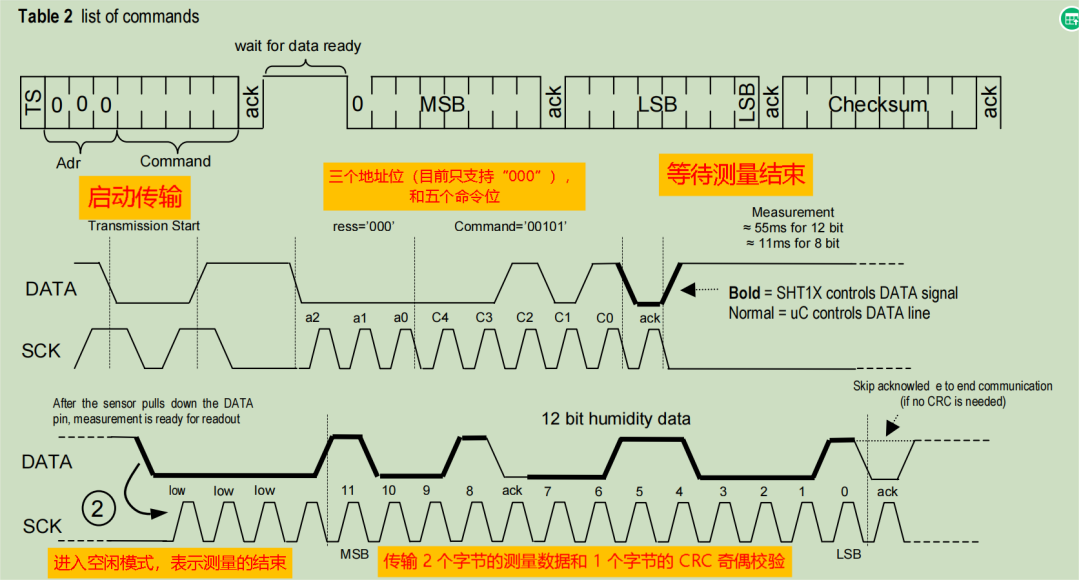


图3.3.2

首先启动传输，输出一系列测量命令，例如0000 0101用于测量温度，然后控制器等待测量结束。SHT11通过将数据向下拖动到0（表示温度测量结束）来切换到空闲模式。

测量数据必须等待信号准备就绪，然后读取数据（等待数据准备就绪）。可以先保存测量数据（MSB、LSB、校验码），控制器执行其他任务以根据需要执行其他任务。读数据。

然后传输两个字节（LSB和MSB）的测量数据和一个字节的CRC检查，并确认每个字节必须延迟。其中，来自MSB的所有数据都是有效且正确的值（来自第五个SCK MSB时钟的12位数据；8位数据的第一个字节没有意义）。CRC数据确认通信已结束。如果你不使用CRC-8认证码，你可以终止传输，在LSB将决定保留1。测量和通信后，SHT11自动切换至睡眠模式。为了确保温升小于0.1℃，SHT11时间被激活15%（例如，对应于12次精确测量，每秒最多三次测量）。

1. 启动传输

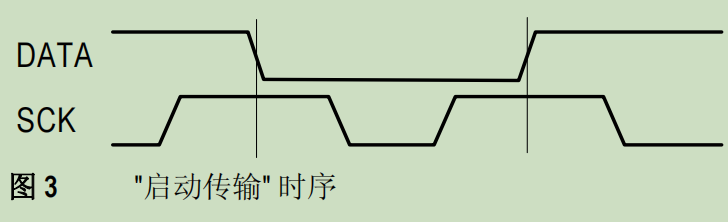
初始化通过一系列“开始传输”定时（数据和SCK）进行。如果SCK时钟为1，则数据反转为0，然后SCK变为0，如果SCK时钟为1，则数据反转为1。如图3.3.3

图3.3.3

代码如下：

1. */\*--------------------------------------*
2. 模块名称：s\_transstart();
3. 功    能：启动传输函数
4. ;-------------------------------------\*/
5. void s\_transstart(void)
6. */ generates a transmission start*
7. */       \_\_\_\_\_         \_\_\_\_\_\_\_\_*
8. */ DATA:      |\_\_\_\_\_\_\_|*
9. */           \_\_\_     \_\_\_*
10. */ SCK : \_\_\_|   |\_\_\_|   |\_\_\_\_\_\_*
11. DATA=1; SCK=0;                   /*Initial state*
12. \_nop\_();
13. SCK=1;
14. \_nop\_();
15. DATA=0;
16. \_nop\_();
17. SCK=0;
18. \_nop\_();\_nop\_();\_nop\_();
19. SCK=1;
20. \_nop\_();
21. DATA=1;
22. \_nop\_();
23. SCK=0;
24. 后续命令

地址位和命令位如图。在第九个SCK下跌之后，释放数据（设置为1），这意味着SHT11传感器已收到正确的指令。五个命令位如下图3.3.4所示：

图3.3.4

出于通信安全考虑，在上升沿之前计算数据的有效时间，SCK下降边缘后延迟Tsu和Tho，如图3.3.5。

向传感器发送命令：

数据在SCK上升沿有效，必须在SCK高水平保持稳定SCK下降边缘后的数据变化（见上文“有效写入数据”一节）.

从传感器读取数据：

SCK变低后数据有效，并将一直保持到下一个SCK的下降边缘（参见图中的数据有效读取部分），如图3.3.6 .

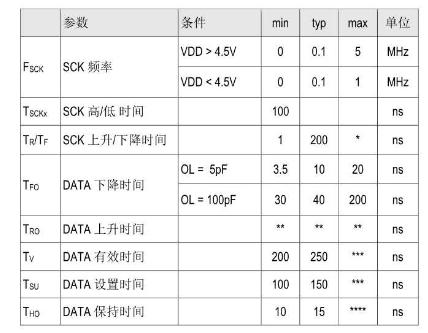


图3.3.5

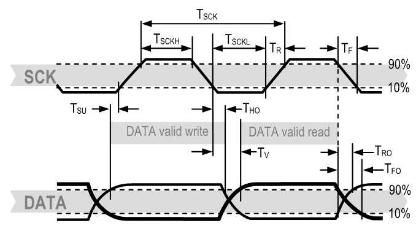
****

图3.3.6

1. 通讯复位时序

如果MCU和传感器之间的通信中断，以下信号序列可能会重置串行接口：

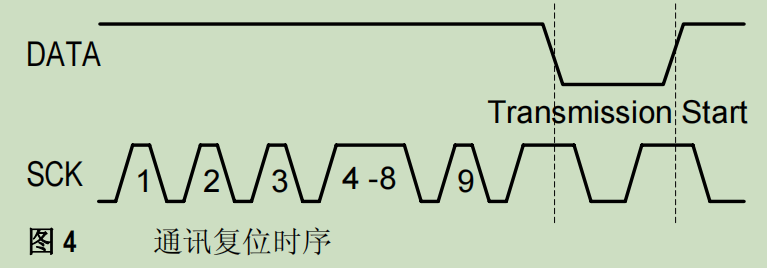
如果数据DATA仍然为1，则SCK时钟被触发九次或更多次，并在下一个命令之前发送传输启动序列。这些时间只重置串行端口，状态的内容保持不变，如图3.3.7。

图3.3.7

代码实现：

1. */\*--------------------------------------*
2. 模块名称：s\_connectionreset();
3. 功    能：连接复位函数
4. ;-------------------------------------\*/
5. void s\_connectionreset(void)
6. unsigned char i;
7. DATA=1; SCK=0;                    /*Initial state*
8. for(i=0;i<9;i++)                  /*9 SCK cycles*
10. SCK=1;
11. SCK=0;
13. s\_transstart();                   /*transmission start*
14. * 1. 输出转换为物理量
15. 相对湿度

由于STH11对应湿度测量具有非线性，sth11使用以下公式（3.3.1）修正，校正测量值，参数如图3.3.8



（3.3.1）

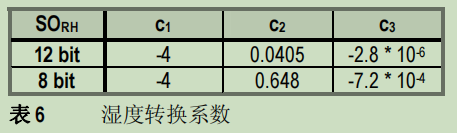


图3.3.8

湿度传感器基本上与电压无关，如图3.3.9 。



图3.3.9

1. 相对湿度对于温度依赖性的补偿

因为测量与实验环境有差，温度修正系数，如下公式（3.3.2），参数如图3.3.10：



（3.3.2）

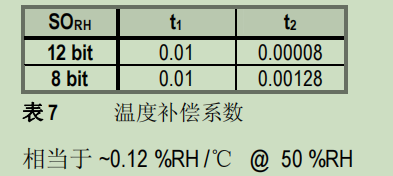
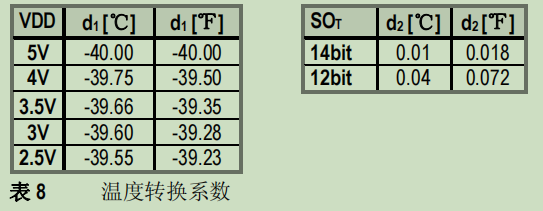


图3.3.10

1. 温度

因为SHT11是采用能隙材料，其具有良好的线性，可以使用以下公式（3.3.3）转换所用的参数，如图3.3.11：

（3.3.3）

图3.3.11

* 1. 按键模块
     1. 工作原理

按键是工业应用中常见的输入设备。它一般分为编码键盘和非编码键盘。一个特殊的硬件解码器检测编码键盘的锁定按钮。非编码键盘由按键开关、按键识别和去抖动功能软件完成。密钥处理主要包括抖动消除、反串按钮、识别和解码。由于触点的弹性和突然的电压跳变，当触点闭合和打开时，机械钥匙会摆动。由于处理系统的处理速度非常高，它可以检测抖动，不会过滤密钥扫描。该系统易于错误操作。

在本设计中，模块的主要模拟功能是设置报警温度和湿度的上限和下限，功能单一。因此，采用更简单的独立按键模块，每个模块连接一条独立的输入线，如图 3.4.1所示。

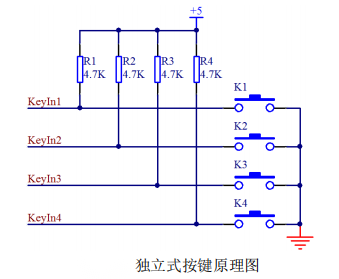


图3.4.1

关键的输入和输出端口线状态不会影响其他输入和输出端口的工作状态。因此，通过检测输入和输出端口线水平，我们可以评估仿真情况下，按到的键盘上的任意键。

第四条输入线单片机IO端口，当Keyin1按下，通过电阻R1 + 5 v，然后形成一个路径通过键K1进入接地，那么所有的线电压被添加到电阻器R1, KeyIn1这是一个低水平。松开按钮后，线路被切断，电流不流动。二者电平脉冲上应该相同，所以可以通过IO的电平脉冲，判断是否按下了按钮。知道电路原理按钮，但实际上MCU IO端口上有一个上拉电阻器。在这个项目中，获得P1非常重要。默认情况下，它是一个双向IO端口。在简单的几乎双向IO端口电路中，如图 3.4.2所示。

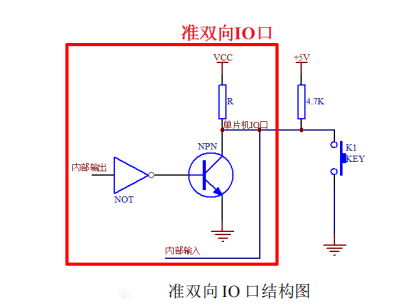


图3.4.2

　　首先，很明显，我们大多数的MCU不是三极管，而是使用IO端口MOS，但这里使用MOS的原理与晶体管相同。这里我们用三极管代替这个，原理图3.4.2。电路指的是单片机内部的一部分。外盒将电阻和按钮拉出。要读取外部按键信号，MCU必须首先写入“1”表示高电平。这样，可以正确读取外部钥匙信号。分析原因。

　　在内部，由于抗拉强度R，MCU的IO端口处于高水平。如果没有按下外部之间的按钮，则电源会降低，VCC为5V。可以读取常用按钮的状态。

　　当输出处于低电平时，逆变器在NPN线路之后变高，因此MCU中的I.O端口变低。此时，两个电阻器并联，但也有一个外部提升电阻。无论是否按下按钮，MCU IO端口的输入状态级别都过低，无法正确读取按钮的状态。这和水很相似。只要内部和外部一侧的电势较低，下游电流就直接接地，每个电阻的分压只会增加电阻，因此电阻较高或较低，相反一侧的电平就应该较低。

　　从上面的分析可以得出结论，这是双向的拉力IO口，如果你想阅读适当的外部信号状态，必须首先要确保自己内部的输出是1，如果内部0，然后不管外部信号是1或0，读的是0。

* + 1. 按键模块的设置

在该系统中，设置了SET,ADD,SEC,ESC按键，P1.6引脚接SET，P1.7引脚接 ADD，P1.8引脚接DEC，P3.7引接ESC，如图3.4.3。

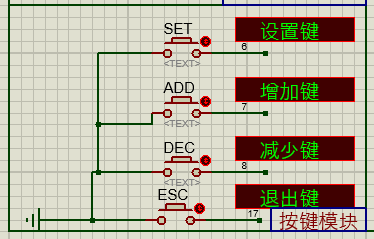


图3.4.3

因此，系统只需要检测针脚P1.6，P1.7，P1.8，P3.7检查四个引脚是否置0（低）时，按下一个按钮。

第一次按下设置按钮时，LCD显示低温限制界面，如图3.4.4所示。

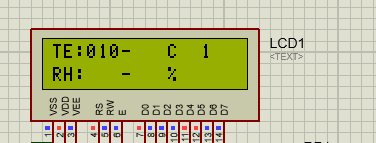
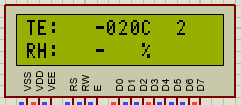
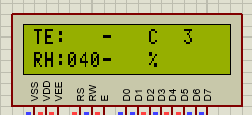


图3.4.4

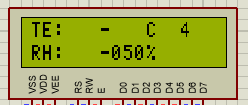
第二次按下“设置”键时，LCD显示温度上限设置界面，如图所示；



第三次按下“设置”键时，LCD显示湿度下限设置界面，如图所示；



第四次按下“设置”键时，LCD显示湿度上限设置界面，如图所示；



第五次按“设置”键，退出设置温度和湿度上限和下限的操作，如图所示。



当我们设置温度和湿度两个数据的上下限时，可以通过按ADD或者DEC按键，分别对温湿度上下限的数值进行设置，图略。

任何时间，都可按“退出键”退出温湿度上下限设置操作，回到实时温湿度显示，如图所示。



代码实现：

1. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*温度下限设置\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*
2. if (flag==1)
4. if(ADD==0)
6. delay(50);
7. t\_xiaxian++;
8. if (t\_xiaxian>(t\_shangxian-1)) {t\_xiaxian=(t\_shangxian-1);}
9. while(ADD==0);
11. if (DEC==0)
13. delay(50);
14. if(t\_xiaxian<=-40)    {t\_xiaxian=t\_shangxian;}
15. else        t\_xiaxian--;
16. while(DEC==0);
18. /*温度下限显示*
19. LCD\_disp\_char(3,1,t\_xiaxian/100+'0');                 /*例如123/100=1*
20. LCD\_disp\_char(4,1,abs(t\_xiaxian%100/10)+'0');         /*例如123%100=23,23/10 =2 ,-12取绝对值12*
21. LCD\_disp\_char(5,1,abs(t\_xiaxian%10)+'0');      /*例如123%10 =3*
22. if(t\_xiaxian<0)    {LCD\_disp\_str(12,2,"nega");}       /*温度下限为负数，显示nega*
23. else       {LCD\_disp\_str(12,2,"    ");}    /*去除负数提醒nega*
24. * 1. 按键消抖

三种方法消抖：

① 延迟方法，软件延迟时间

大约10毫秒，然后再次评估。如果重采样值与上次匹配，则认为它是有效。

② 计时器评估方法：在首次扫描按键时启动计时器，并在计时器时间后再次扫描。

③ 循环扫描法：将扫描插入主程序，扫描次数取决于系统的主频率。在主程序每次运行后收集变量，以确定它们是否与原始值匹配。三种方法中的第一种会占用更多的时间，并影响整个系统的执行效率。第二种方法占用计时器，第三种方法占用更少的资源。可适当处理，实现加速、长键等功能。

按下时，触点通过机械触点的软动作断开或闭合，如图所示，当它关闭时，会伴随一系列振动。3.4.5所示。

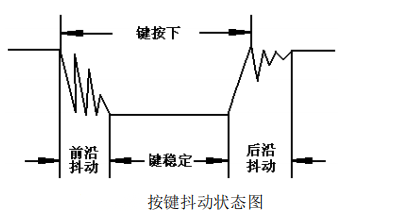


图3.4.5

稳定键的关闭时间长度由操作员决定，通常超过100ms。如果你故意快速按下，它将达到40~50毫秒，很难脱落。抖动时间由键的机械性能决定，通常长达10毫秒。如果key状态发生变化，按键不会立即响应，而是稳定关闭或关闭进行后处理。

从硬件上解决按键抖动，可以将电容C并联到按钮上。如图3.4.6所示，通过电容器充放电特性平滑抖动引起的电压毛刺，以避免抖动。然而，在实际应用中，该方法效果较低，增加了电路成本和复杂度，实际应用较少。

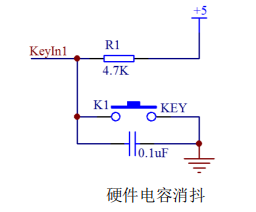


图3.4.6

软件消抖。最简单的原理是，在发现key状态的变化后等待大约10毫秒，然后移除抖动，然后检查按键状态。如果状态与刚刚的状态相同，可以检查稳定性按钮的操作。

代码如下：

1. if(SET == 0)
3. delay(50);               /*按键消抖*
4. if(SET == 0)

3.5 报警模块

* + 1. 蜂鸣器报警模块

报警模块，采用PNP驱动蜂鸣器beep报警。蜂鸣器又称喇叭，广泛应用于各种电子产品中，通常用于警示、报警等场合。

在日常生活中，蜂鸣器和喇叭在使用上是相似的。一般来说，普通电路的工作电流较大，电动TTL很难驱动普通蜂鸣器。这使得一个引脚很难使正常工作蜂鸣器。此时，我们需要添加一个三极管来增加流经蜂鸣器的电流，从而创建一个电流放大电路，如图3.5.1所示。

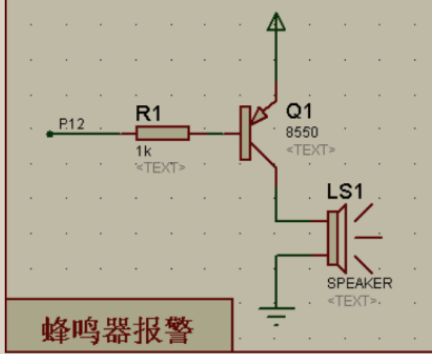
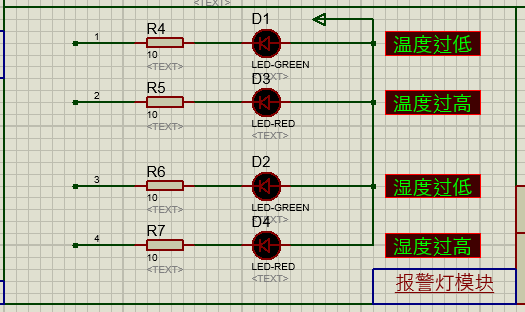


图3.5.1

三极管的底座由P1.4引脚组成和控制。如果收集的数据超过校正后设置的温度上限和下限，P1.4引脚将设置为0，三极管将打开，以便蜂鸣器发出噪音。当引脚4设置为1时，三极管断开，蜂鸣器关闭。

代码实现：

1. /*蜂鸣器警示*
2. if(\*WENDU <= t\_xiaxian||\*WENDU >= t\_shangxian||\*SHIDU <= rh\_xiaxian||\*SHIDU >= rh\_shangxian)
4. BEEP = 0;
6. else
8. BEEP = 1;
9. * 1. LED报警模块

LED报警模块，设置四个报警指示灯，指示当前温度和湿度值上限和下限的报警状态。如果温度和湿度值过低，相应的绿色LED将点亮；如果温度和湿度值过高，相应的红色LED将点亮；如果湿度在设定值内，则关闭LED。3.5.2所示。

代码实现：

1. */温度过低报警指示*
2. if(\*WENDU <= t\_xiaxian)    {t\_green = 0;}
3. else {t\_green=1;}
4. */温度过高报警指示*
5. if(\*WENDU >= t\_shangxian)  {t\_red   = 0;}
6. else {t\_red=1;}
7. */湿度过低报警指示*
8. if(\*SHIDU <= rh\_xiaxian)   {rh\_green= 0;}
9. else {rh\_green=1;}
10. */湿度过高报警指示*
11. if(\*SHIDU >= rh\_shangxian) {rh\_red  = 0;}
12. else {rh\_red=1;}
13. **硬件设计**
    1. **Keil软件开发平台**

KEIL是对应8051 系列单片机的开发工具，支持几百种51的MCU芯片，用C语言编程，与结构化语言相比，C语言在功能、结构、可读性和一致性方面具有明显的优势。有俩种仿真方法（软件模拟模式和目标板仿真调试模式），虽然protues也可以仿真，但是二者在各自的平台会发挥各自的作用。

Mvision和ishell C51芯片支持基于windows和DOS的C51开发环境组件。完成编辑、编译、链接、调试和模拟的整个开发过程。开发者可以直接编辑原始文档或报表。C51和A51编译器也处理对象文件。目标文件可以在HEX文件调试器dscope51或tscope51中创建，也可以直接在目标磁盘上调试或写入程序内存（如EPROM）。

* 1. **主程序设计**

主程序主要调用每个子程序的C语言文件中定义的函数来实现SHT11，

LCD初始化过程，然后测量温度和湿度，调用数据处理功能，最终在LCD 1602上显示，当前温度和湿度超过设定阈值时进行监视和警告。

* 1. **SHT11程序设计**

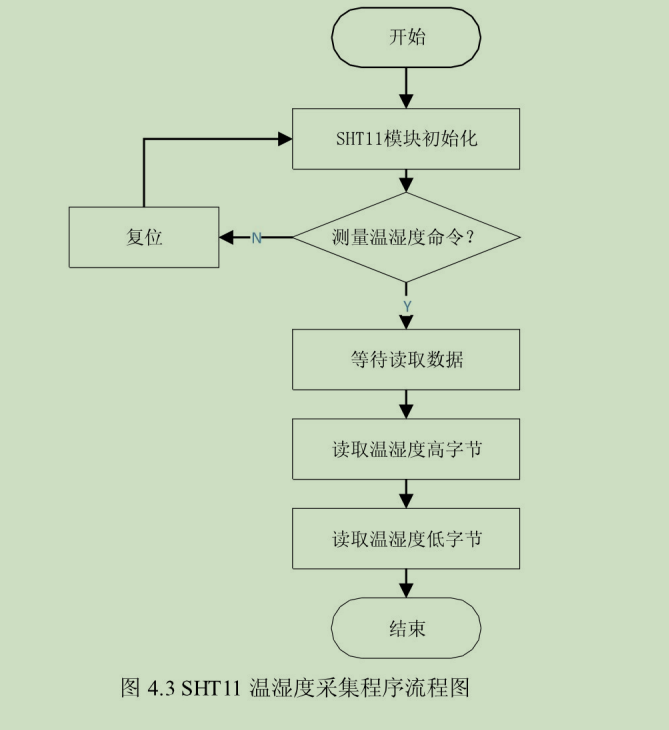
SHT11检测模块的任务是采集温度和湿度数据，将检测到的温度模拟信号转换成相应的数字信号并发送给芯片。然后是数据处理。SHT11数字温湿度传感器将室外温度和湿度直接转换为数字信号，因此，它可以直接发送到单片机进行处理，因此不需要通过A-D转换器将其转换成数字信号。本实用新型使用方便，抗干扰能力强。软件程序也大大简化了。数据采集模块的程序顺序只包括四个步骤：数据采集程序初始化，即SHT11程序初始化→ 温湿度检测→ 等待温度和湿度转换→ 测量温度和湿度并将其发送到单机进行处理，如图4.3.1所示。

图4.3.1

* 1. LCD1602程序设计

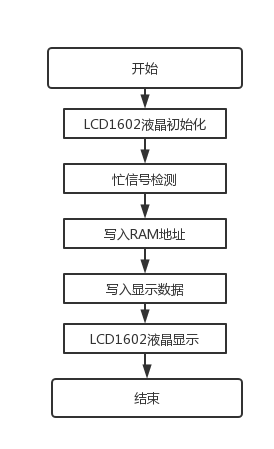
如图4.4所示。

图4.4

* 1. 报警模块程序设计

如图4.5所示。

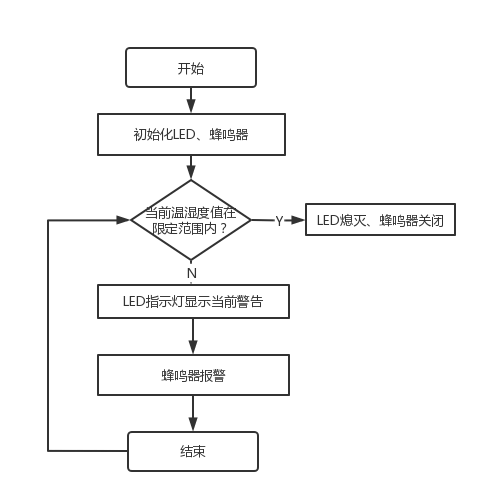


图4.5

1. 仿真与测试
   1. Protues设计平台

Proteus仿真软件针对设备投资高、维护费用高的问题，可以在有限的硬件设备上采用多种方法，既能直观地描述工艺操作，它还可以达到传统方法的效果；它可以进行模拟和数字仿真。它不仅可以模拟单片机CPU的工作状态，还可以模拟单片机外围电路或其他不需要单片机参与的电路的工作状态。虚拟介质可以直接在原理图上编程，然后显示输出以实现此结果。仿真软件允许抽象原理的可视化，并提供一个虚拟控制按钮来执行和退出程序。

与其他MCU模拟器不同，proteus不仅可以模拟mcu的行为，还可以模拟mcu或其他非mcu外围电路的状态。因此，在执行多个命令时，程序的建模和调试会考虑程序和链的操作过程和结果，而不是寄存器和内存中的内部更改。该软件具有电子元件丰富复杂的特点。该程序提供数千个组件。实验内容不仅包括软件安装和调试，还包括大部分硬件接口。keil等第三方软件具有强大的图形编辑和调试功能。Keil和Proteus可以在一定程度上模拟他们的环境，但效果并不好。这两个优点的结合可以充分应用于软件调试和硬件设计建模，更加方便、高效。

结果表明，使用两个软件，更具有其他优势。这两个因素的结合充分发挥了各自的特点，打破了硬件平台的局限性，节省了设备准备时间，减轻了现场测试的负担，有效地解决了学生理论与实践的矛盾，大大提高了实验教学的有效性。提高单片机应用系统的调试效率，降低开发成本也非常重要。

这是我们设计过程中使用最广泛、最好的工具。它提供了许多在实践中无法获得的元件，我们可以无限地改变我们想要的电路设计。

* 1. 仿真结果分析
     1. 仿真要求

1. 实现功能1：SHT11读取处理+LCD1602显示模块
2. 实现功能2：采取按键模块设置并且控制，温湿度报警值的上限和下限。
3. 实现功能3：

如果SHT11传感器的温度和湿度读数未超过上述限值，LED警告灯将显示错误信息并通过蜂鸣器发送。

当温度和湿度值在阈值范围内时，信号LED在正常范围内熄灭，beep熄灭。

* + 1. 仿真过程

1. 打开SHT11与STC89C52.pdsprj文件
2. 双击STC89C51芯片，烧录DU.hex，如图5.2.1

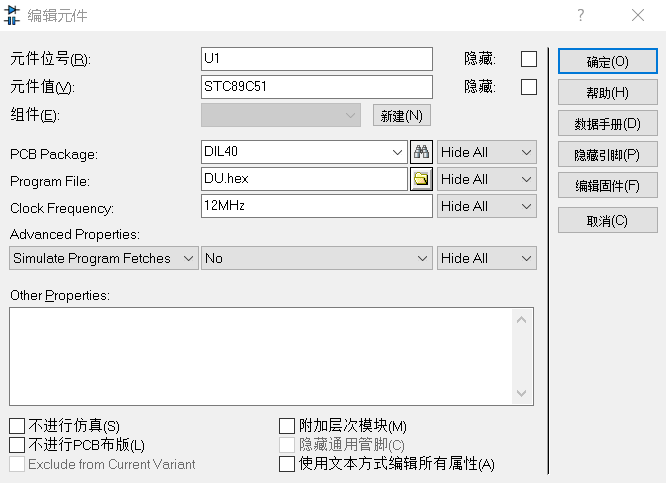


图5.2.1

1. 仿真，实现功能1实现截图如图5.2.2（SHT11读取处理+LCD1602显示模块）

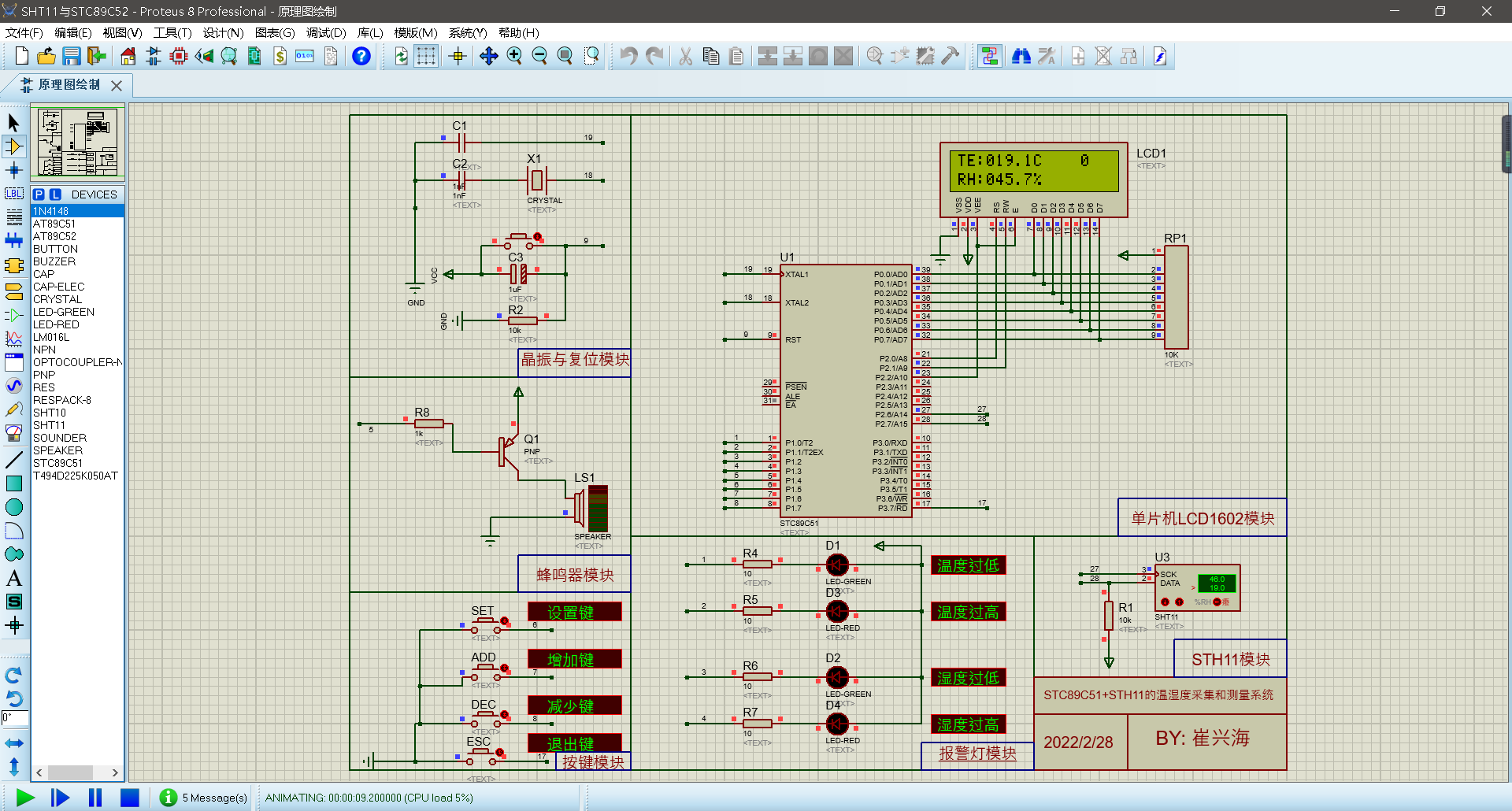


图5.2.2

实现功能2：采取按键模块设置并且控制，温湿度报警值的上限和下限。

按下设置键一次进入温度下限，如屏幕图所示5.2.3（右上角1为标志位提示）

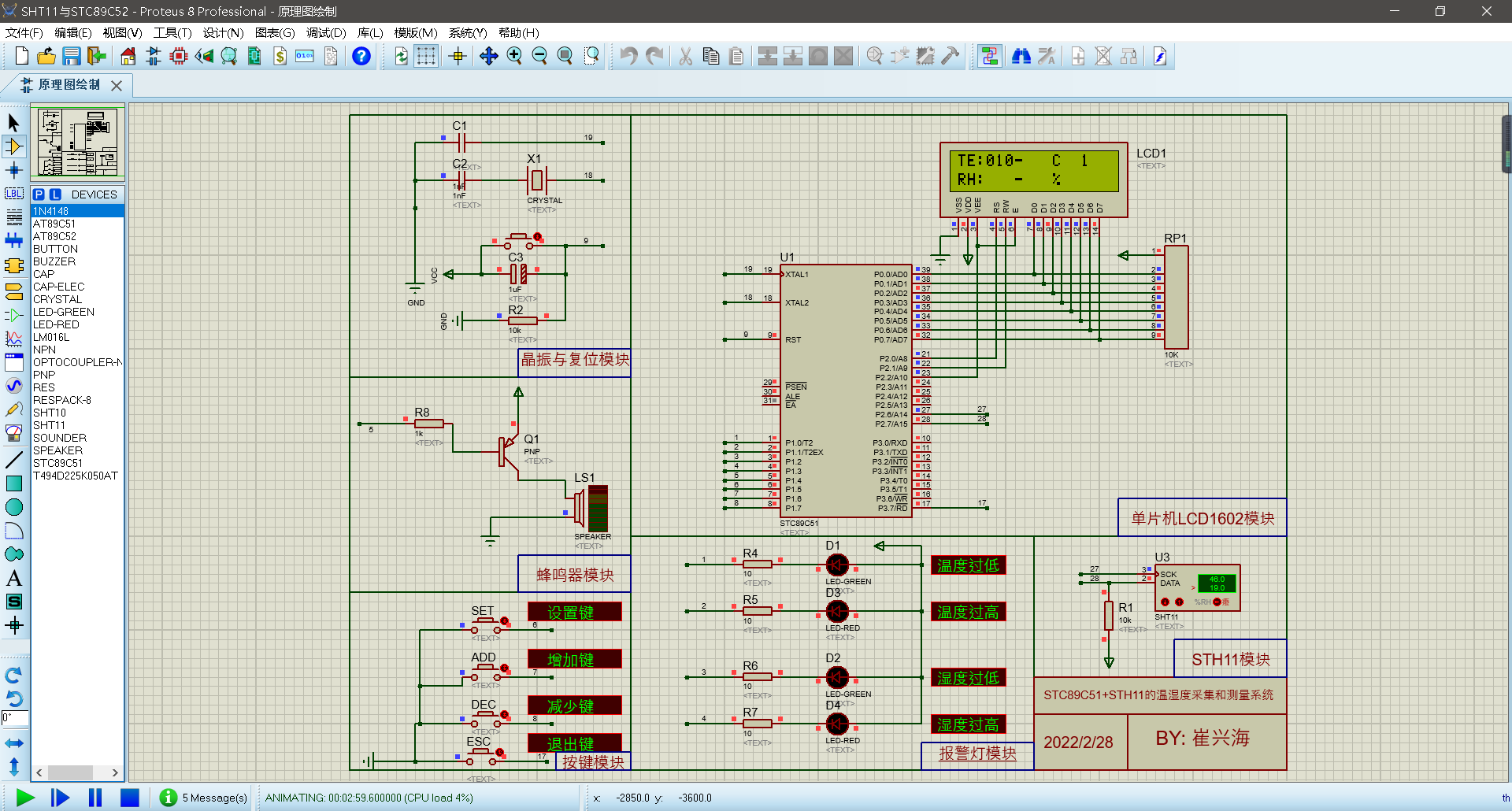
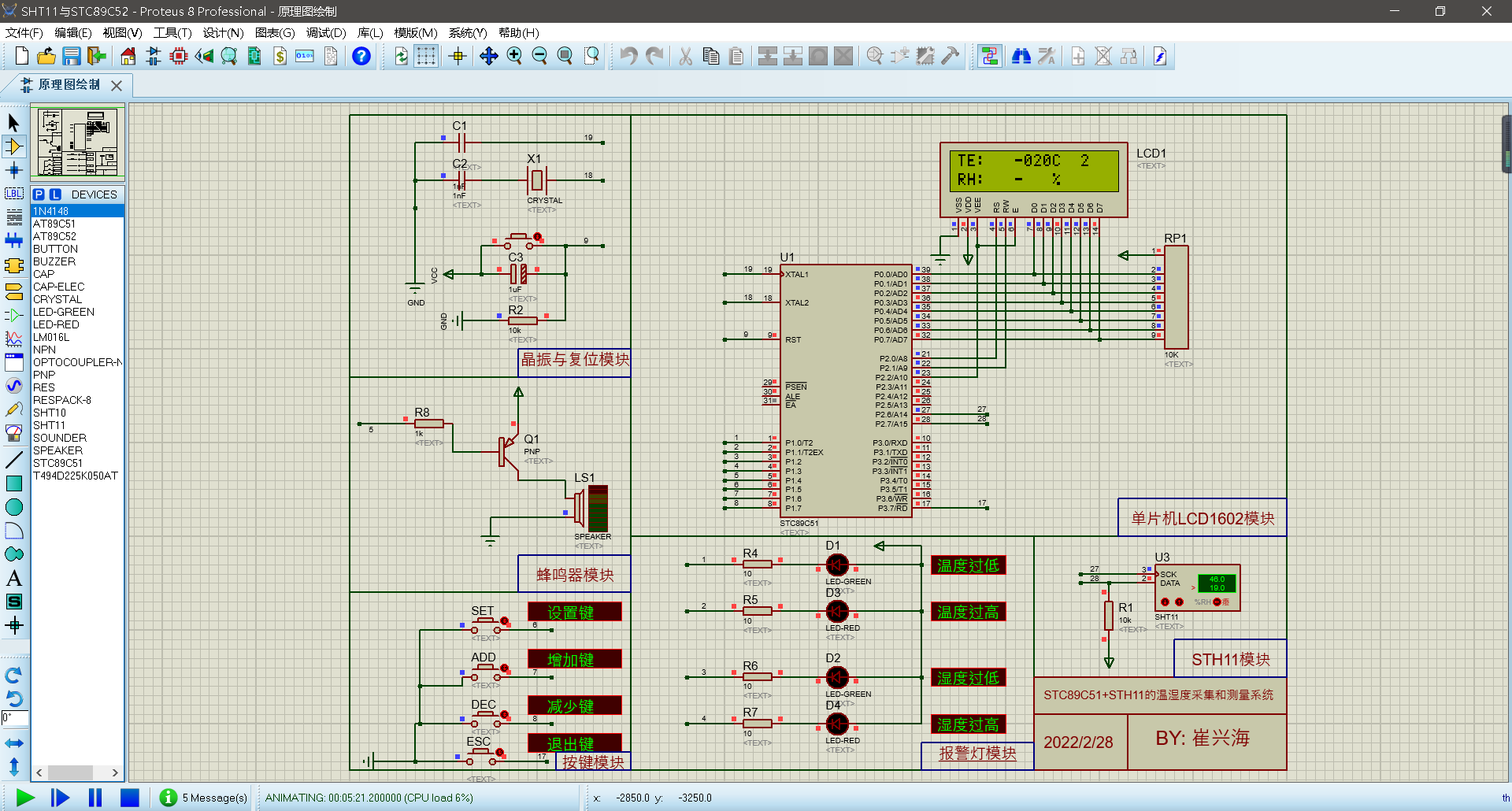


图5.2.3

按第二次设置键进入温度上限设置截图如图5.2.4：

图5.2.4

按第三次设置键进入湿度下限设置截图如图5.2.5：

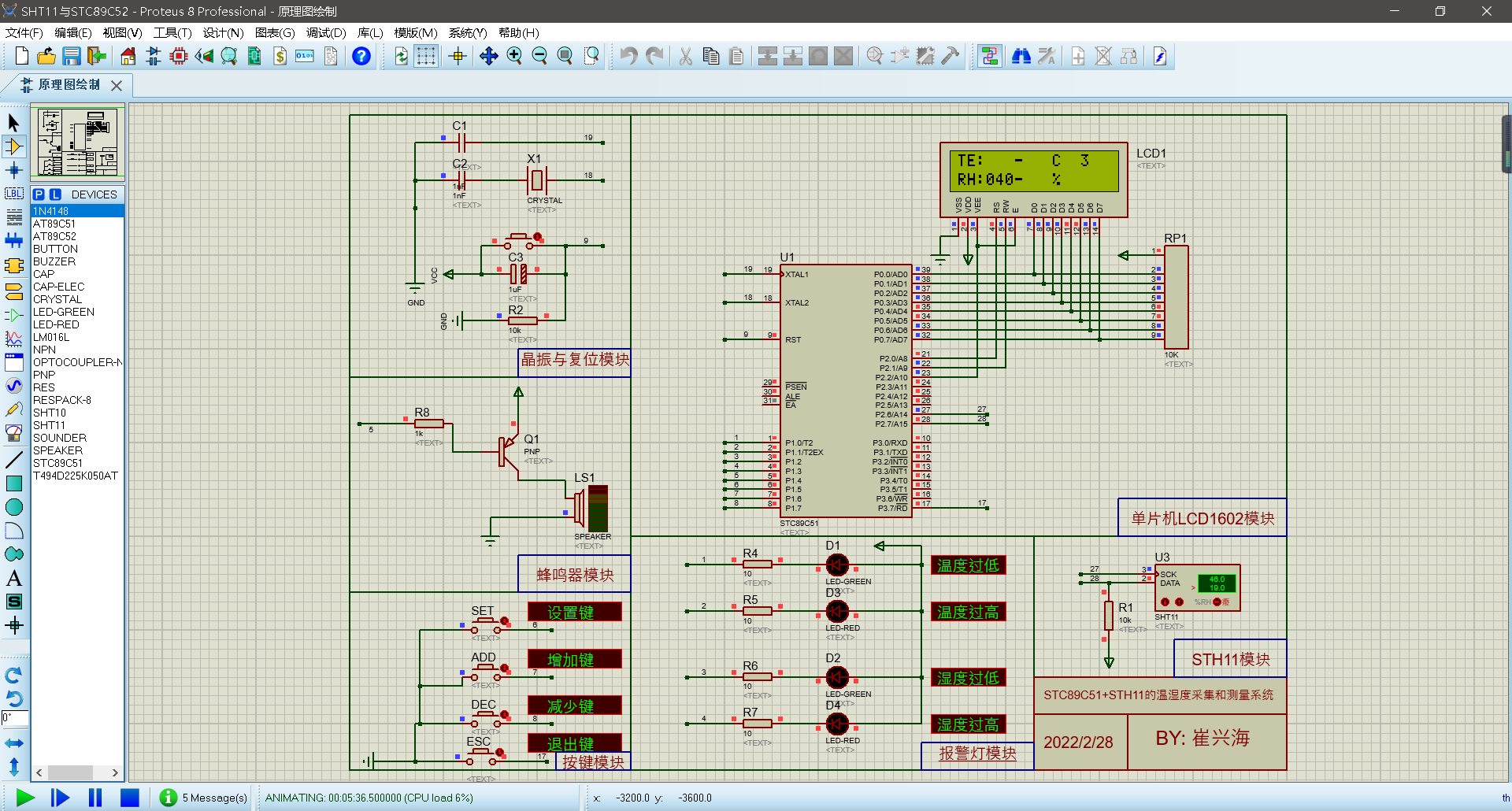


图5.2.5

按第四次设置键进入湿度上限设置截图如图5.2.6：

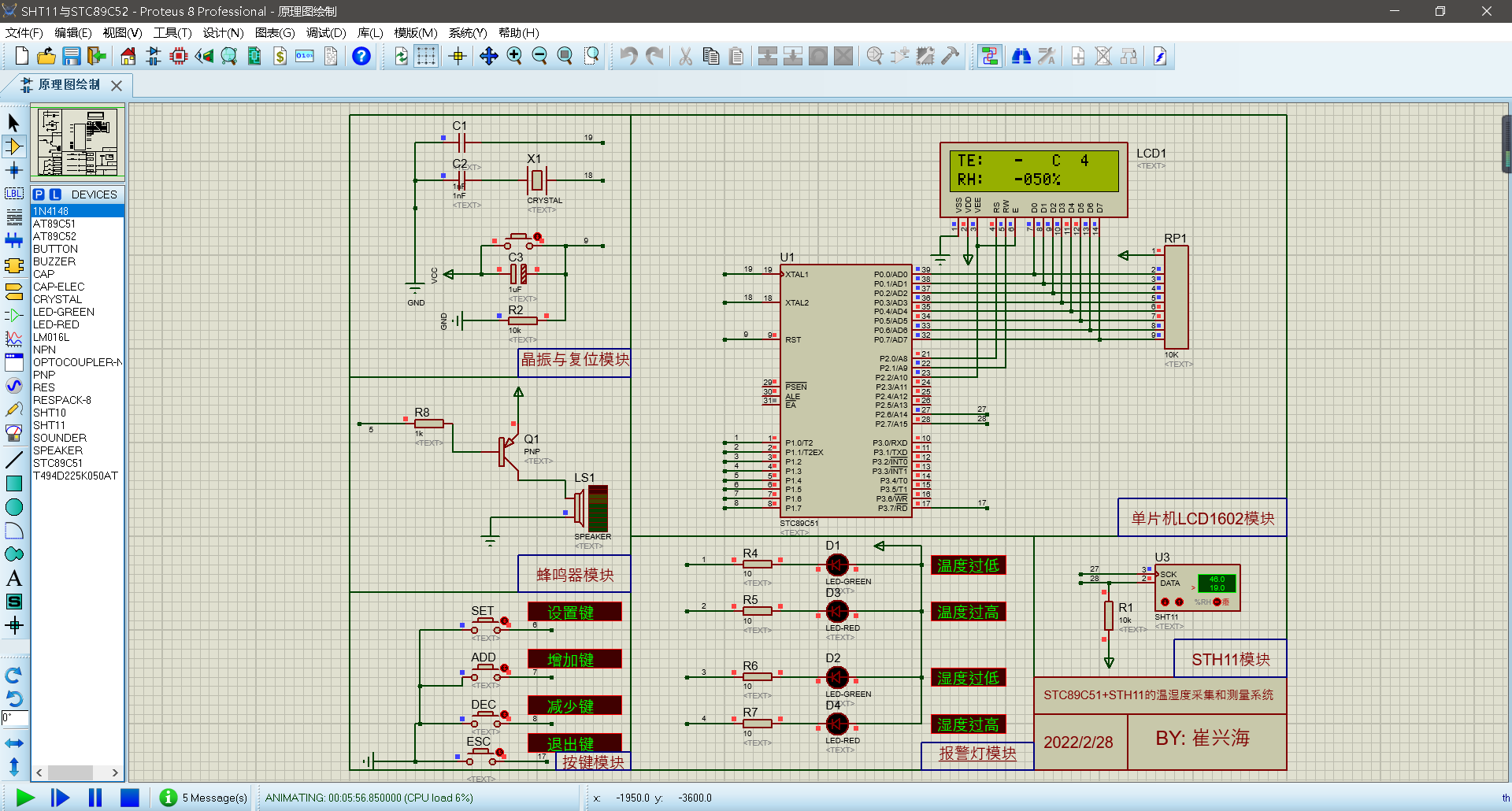


图5.2.6

警告：湿度SHT11的测量范围为0%至100%相对湿度。而温度范围在－40°C和123,8°C之间；

温度下限设置－40C截图，如图5.2.7：

温度下限可设置为负数

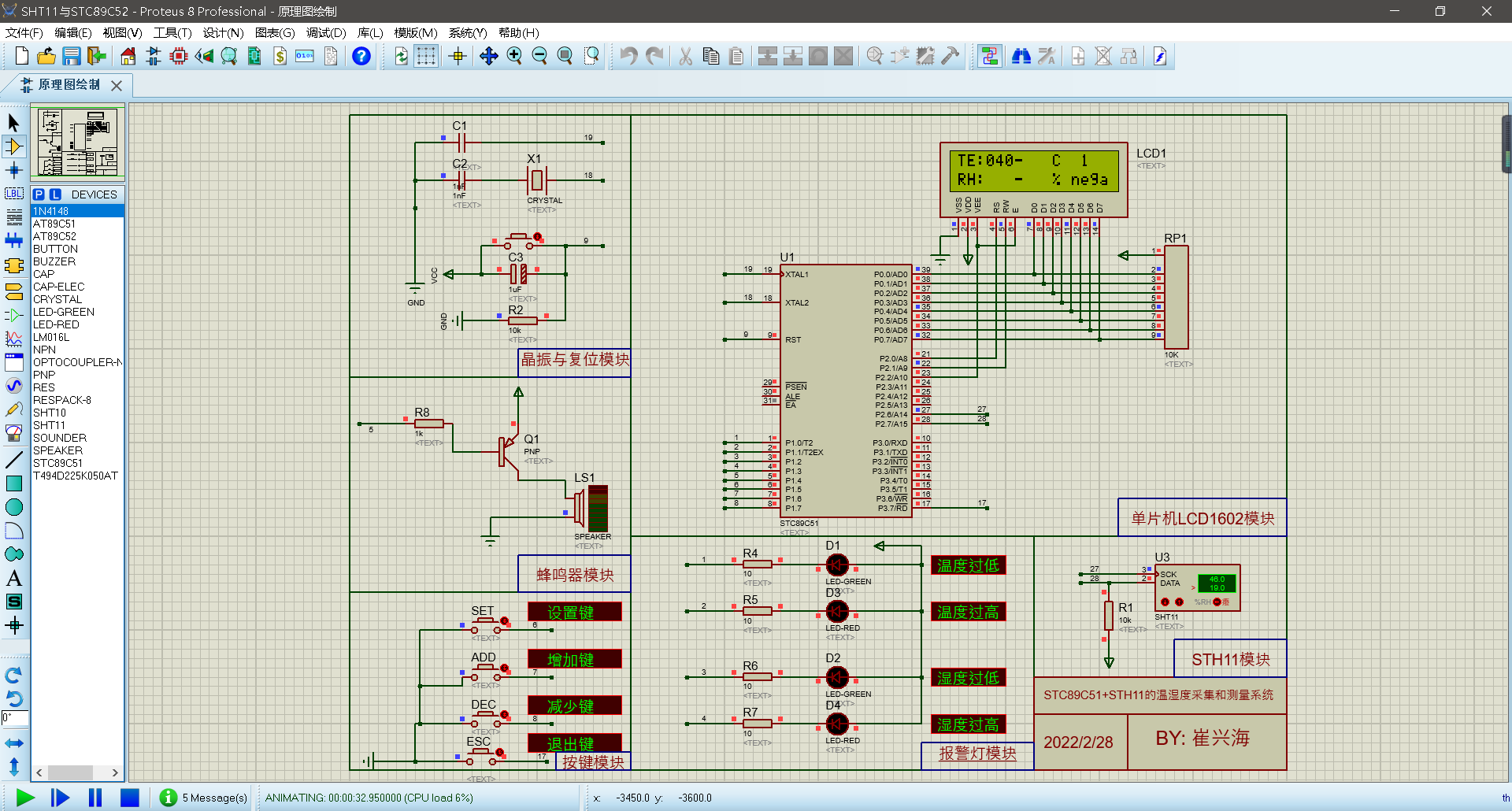


图5.2.7

温度上限限设置+123C截图，如图5.2.8：

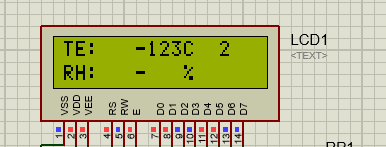


图5.2.8

湿度下限设置0%截图，如图5.2.9 ：

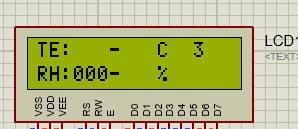


图5.2.9

湿度上限限设置100截图，如图5.2.10：



图5.2.10

1. 实现功能3：

如果SHT11传感器读取的温湿度值不在我们设定的阈值范围内，如上所述，相应的报警LED灯将显示错误信息，蜂鸣器将报警；

如果温度和湿度值在我们设定的阈值范围内，即在正常范围内，则报警LED熄灭，蜂鸣器熄灭。

SHT11报警湿度默认范围：40～50%RH；报警温度范围：10～20℃；

1. int t\_shangxian = 20;            /*上限报警温度，默认值为 20C*
2. int t\_xiaxian   = 10;            /*下限报警温度，默认值为 10C*
3. int rh\_shangxian= 50;            /*上限报警湿度，默认值为50%*
4. int rh\_xiaxian  = 40;            /*下限报警湿度，默认值为40%*

低于下限报警温度的蜂鸣器与LED截图，如图5.2.11：

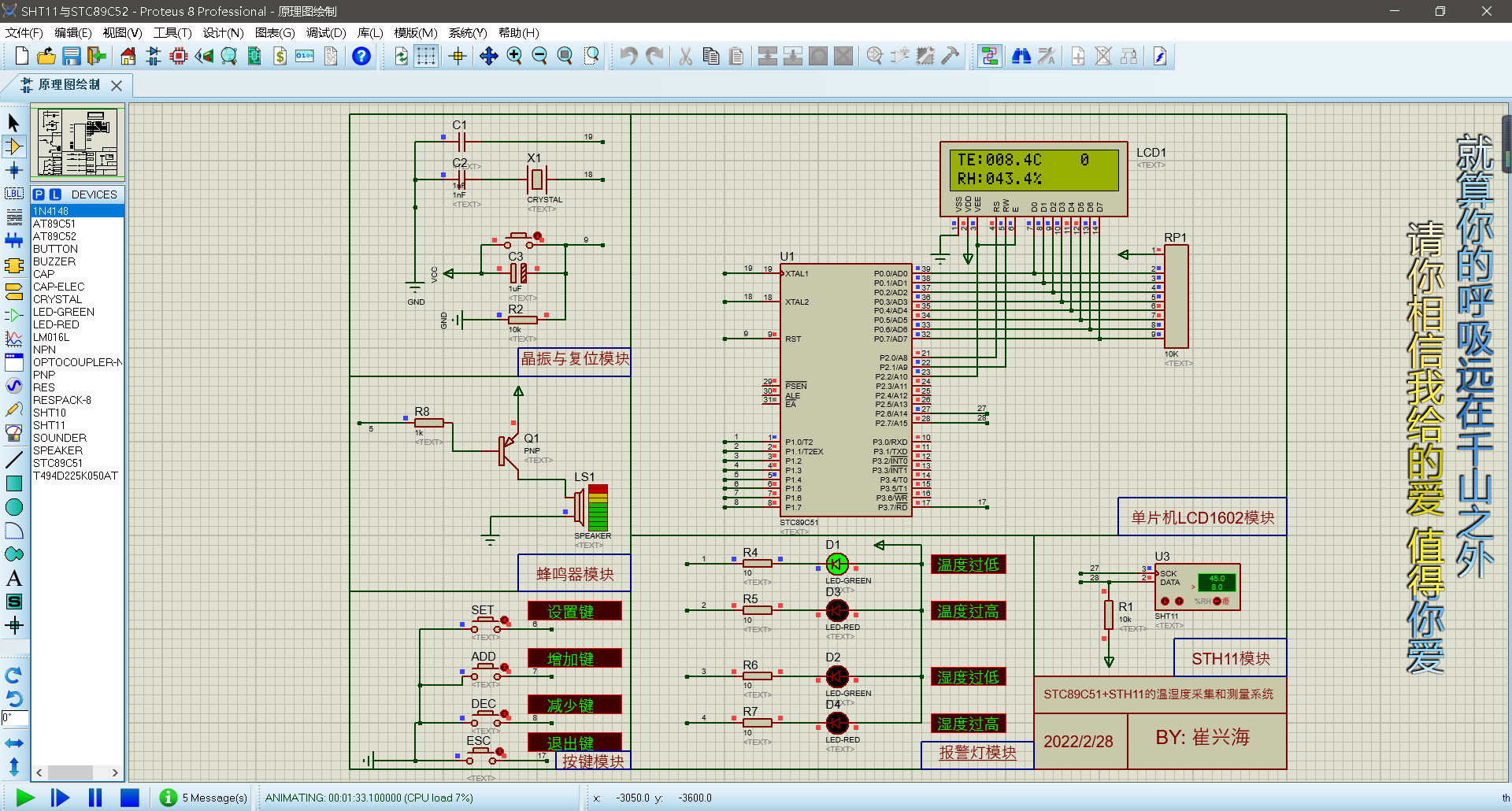


图5.2.11

* 高于上限报警温度的蜂鸣器与LED截图，如图5.2.12 ：

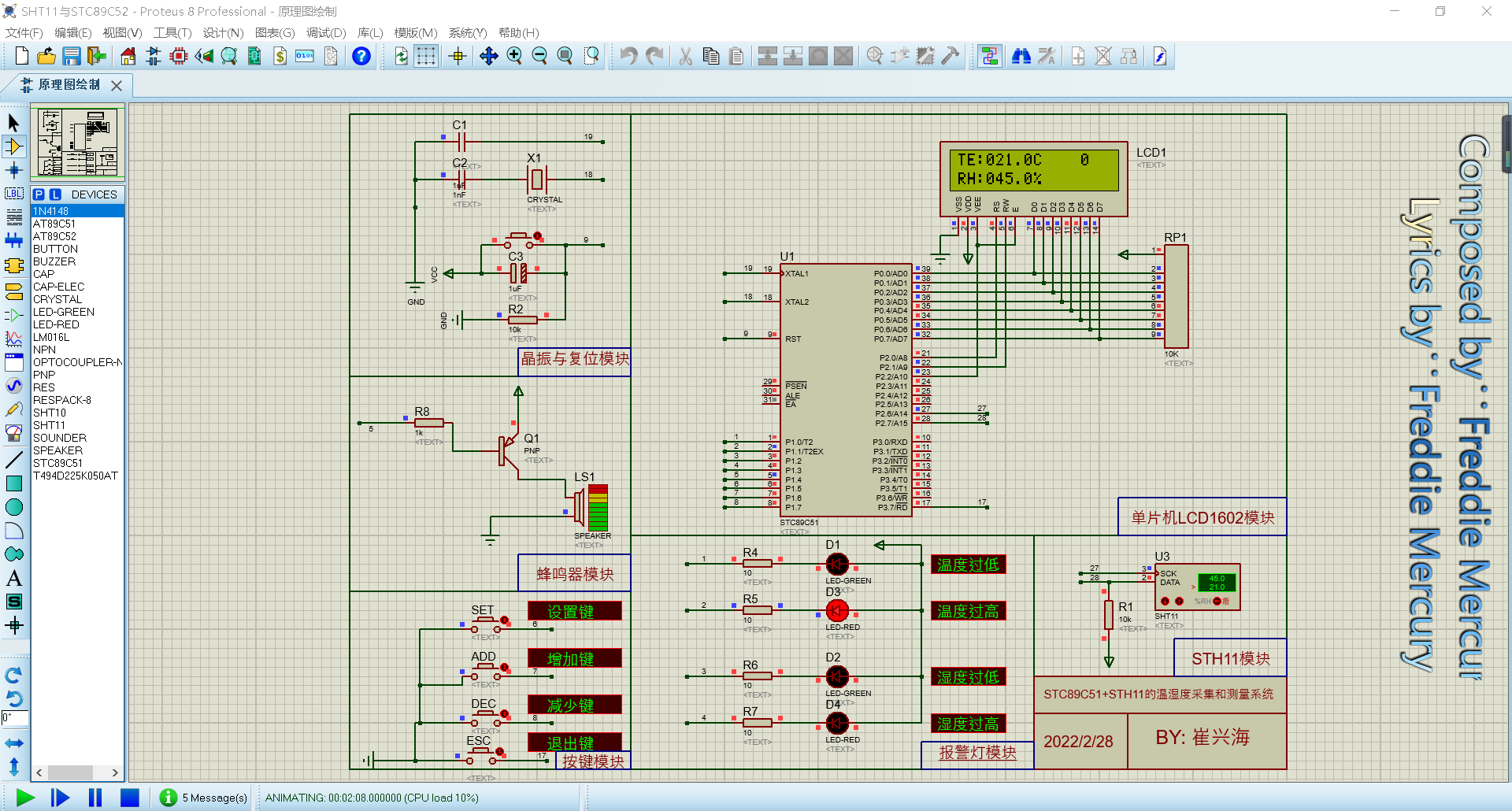


图5.2.12

* 低于下限报警湿度的蜂鸣器与LED截图，如图5.2.13

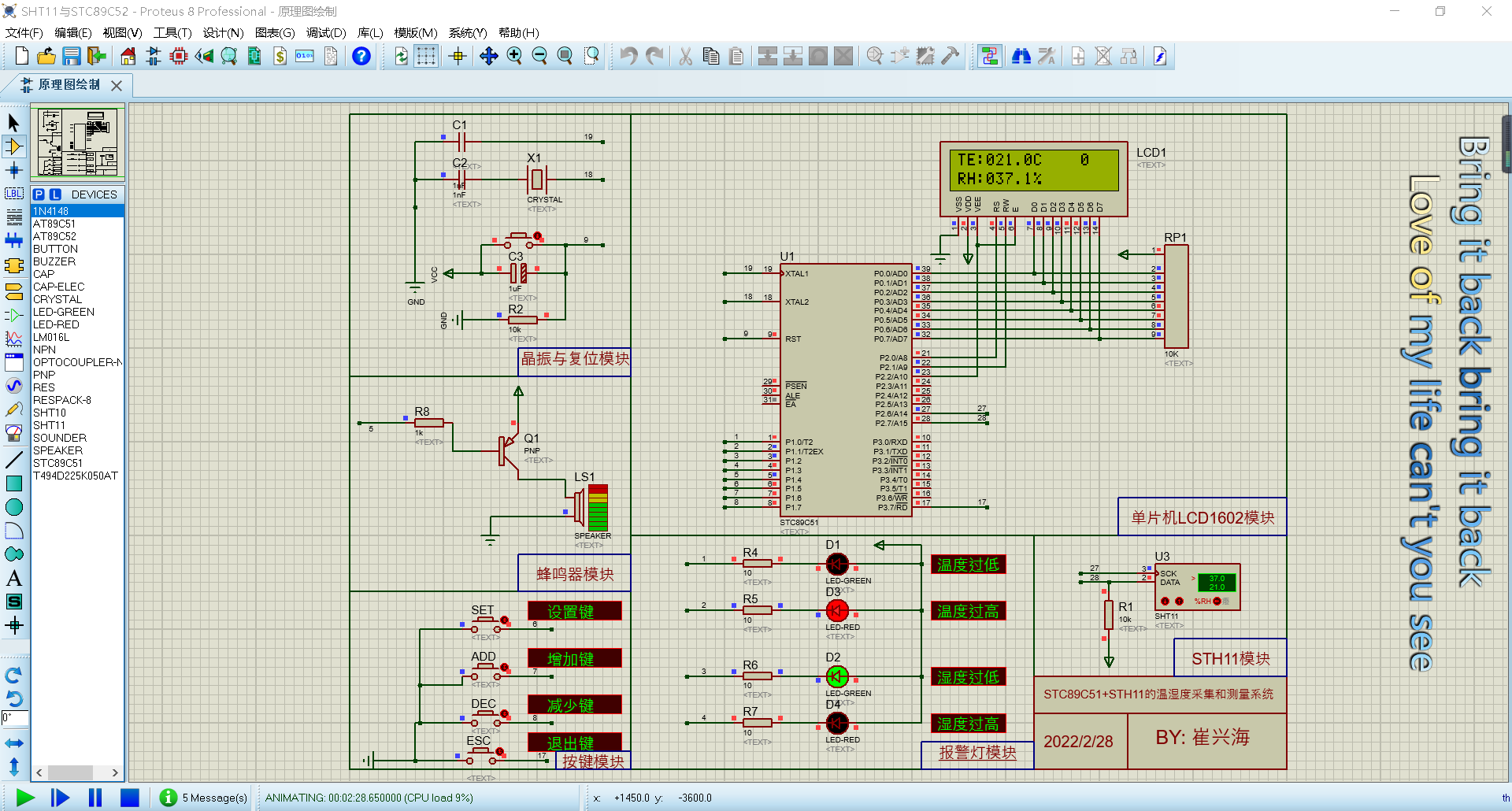


图5.2.13

* 高于上限报警湿度的蜂鸣器与LED截图，如图5.2.14

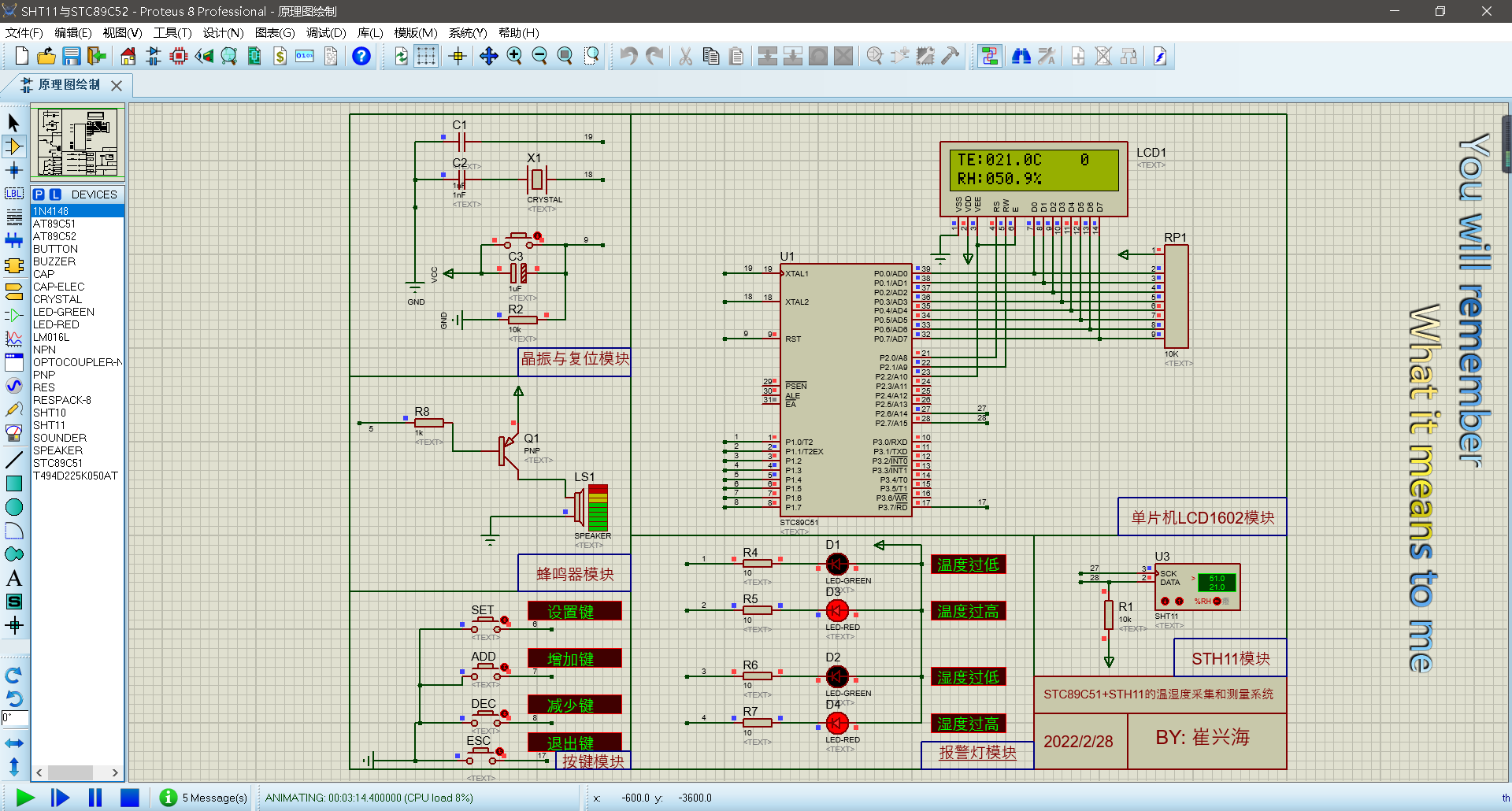


图5.2.14

致谢

首先，谢谢我的导师韩从道先生。这篇文章是在他的精心指导和帮助下完成的，这反映了我导师的辛勤工作。在整个工作过程中，韩老师在很多方面都给了我具有启迪性的意见和启发。在这里再次深深地感谢我的导师。

其次，我还要感谢所有愿意帮助我的电子信息与电气工程学院的老师和学生。

最后，再次感谢我的学校以及在论文、答辩工作中的所有老师与参与者们。

**参考文献**

1. 邹声平、初光勇. 基于STC89C52的智能灌溉系统设计[J]. 物联网技术. 2020,10(09): 99–100，105.

2. 朱雪建、俞梁英、宋玉荣、范义武、罗永松. 基于51单片机的电动车智能充电桩自动灭火系统设计[J]. 轻工科技. 2020,36(11): 49–51,70.

3. 周文亮、王靖、范东升、许明忠、罗勤站、宋战锋. 烟叶烘烤温湿度物联网监控系统设计[J]. 南方农机. 2021,52(03): 31-33.

4. 芮旺胜、吴章洋、储烨. 卧室温湿度采集系统设计[J]. 电子世界. 2021,(01): 121-122,125.

5. 蔡用霞、吕晓梁、卢佩. 基于AT89C51的温室大棚温湿度测控系统设计[J]. 科技信息.2010,(13): 79-80.

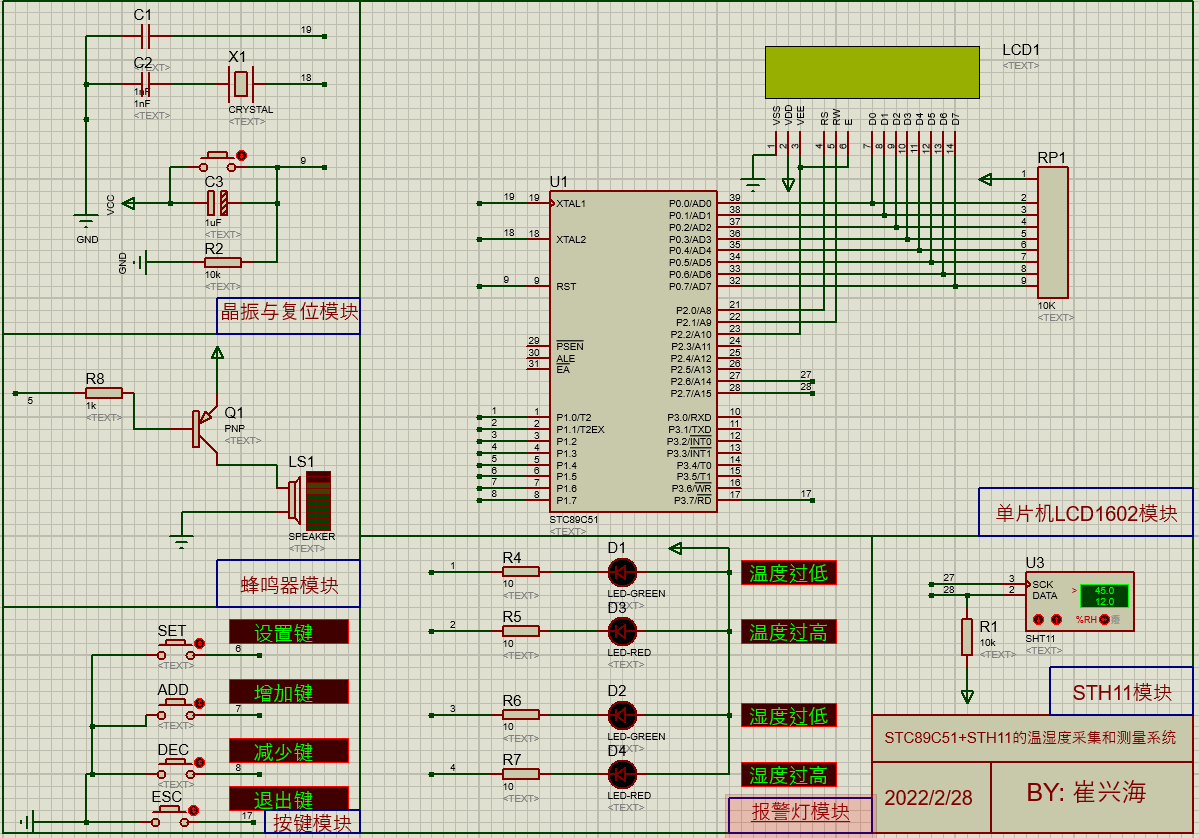
6. 任星星、黄祺.一种基于SHT10的温湿度监测系统设计[J]. 商业故事.2018,(21):143．

7. 徐晓. 基于AT89C51的土壤温湿度数据采集与调节系统设计[J]. 科学技术与工程. 2009,9(04): 1032-1034,1063.

8. 吴国宏. 新型温湿度传感器SHT10的原理及应用[J]. 单片机与嵌入式系统应用.2009,(04): 52-54.

附录

附录一：仿真总电路图



附录二：程序主代码

1. */\*\*\*\*\*\*\*\*\*主函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
2. void main(void)
3. value humi\_val,temp\_val;         /*185行，定义两个共同体，一个用于湿度，一个用于温度*
4. unsigned char error,checksum;     /*用于检验是否出现错误*
5. unsigned int wendu,shidu;     /*最终，一位小数温湿度的值*
6. LCD\_init();
7. s\_connectionreset();
9. */\*\*\*\*\*\*\*\*\*初始化温度显示区\*\*\*\*\*\*\*\*\**
10. LCD\_disp\_str(0,1,"TE:");
11. LCD\_disp\_str(3,1,"TTT.TC");
12. */\*\*\*\*\*\*\*\*\*初始化湿度显示区\*\*\*\*\*\*\*\*\**
13. LCD\_disp\_str(0,2,"RH:");
14. LCD\_disp\_str(3,2,"RRR.R%");
15. delay\_n10us(2000);     /*延时0.2s*
16. while(1)
18. LCD\_disp\_char(13,1,flag+'0');
19. key();
21. if(flag==0)
23. error=0;            /*初始化error=0，即没有错误*
24. error+=s\_measure((unsigned char\*) &humi\_val.i,checksum,HUMI);  /*measure humidity*
25. error+=s\_measure((unsigned char\*) &temp\_val.i,checksum,TEMP);  /*measure temperature*
26. if(error!=0) s\_connectionreset();                 /*in case of an error: connection reset*
27. else
29. humi\_val.f=(float)humi\_val.i;                   /*converts integer to float*
30. temp\_val.f=(float)temp\_val.i;                   /*converts integer to float*
31. calc\_SHT11(&humi\_val.f,temp\_val.f);            /*calculate humidity, temperature*
33. LCD\_disp\_str(0,1,"TE:");
34. LCD\_disp\_str(0,2,"RH:");
35. LCD\_disp\_str(6,1,".");
36. LCD\_disp\_str(6,2,".");
37. LCD\_disp\_str(8,1,"C   ");
38. LCD\_disp\_str(8,2,"%   ");
39. wendu=10\*temp\_val.f;                            /*例如温度109.1→1091*
41. if(10\*temp\_val.f<0) {LCD\_disp\_str(12,2,"nega");}/*实时温度为负数时，显示提示符nega*
42. else {LCD\_disp\_str(12,2,"    ");}           /*实时温度为正数时，去除负数提醒nega*
44. LCD\_disp\_char(3,1,abs(wendu)/1000+'0');         /*显示温度百位，加“0”是为了将字符的ASCII码大于48（即字符0的ASCII值），一般是将数字0,1,2……，9转换为字符“0”，“1”……，“9”；*
45. LCD\_disp\_char(4,1,abs(wendu)%1000/100+'0');     /*显示温度十位*
46. LCD\_disp\_char(5,1,abs(wendu)%100/10+'0');       /*显示温度个位*
47. LCD\_disp\_char(7,1,abs(wendu)%10+'0');           /*显示温度小数点后第一位*
48. shidu=10\*humi\_val.f;
49. LCD\_disp\_char(3,2,shidu/1000+'0');               /*显示湿度百位*
50. LCD\_disp\_char(4,2,(shidu%1000)/100+'0');         /*显示湿度十位*
51. LCD\_disp\_char(5,2,(shidu%100)/10+'0');           /*显示湿度个位*
52. LCD\_disp\_char(7,2,(shidu%10)+'0');               /*显示湿度小数点后第一位*
54. led\_control(&temp\_val.f,humi\_val.f);
56. /*----------wait approx. 0.8s to avoid heating up SHT11------------------------------*
57. delay\_n10us(800);                                /*延时约0.8s*

仿真文件和论文报告完整版：

链接： https://pan.baidu.com/s/1fEXZ7fWfRJ3dYbX0LhA3Eg?pwd=qgwq