东北大学秦皇岛分校
Northeastern University at Qinhuangdao

《检测与转换技术》结课报告

基于 DS18B20 的温度显示系统设计

专　　业	电子信息工程
班级序号	电信 **1706-13**
学　　号	20178136
姓　　名	**余嘉慧**

1. 引入

本文介绍基于 DS18B20 温度传感器测量温度，再通过 51 单片机控制 7 段数码管显示温度。

2. AT89C51

2.1. 芯片简介

51 单片机是较为常用的单片机，因其造价低廉，使用简便，被广泛使用，故本文使用 AT89C51 来实现温度传感器和显示电路的连接。

AT89C51 和 AT89C52 的四个端口都是双向的。每一个都由一个锁存器(特殊功能寄存器 P0 到 P3)，一个输出驱动器和一个输入缓冲区组成。

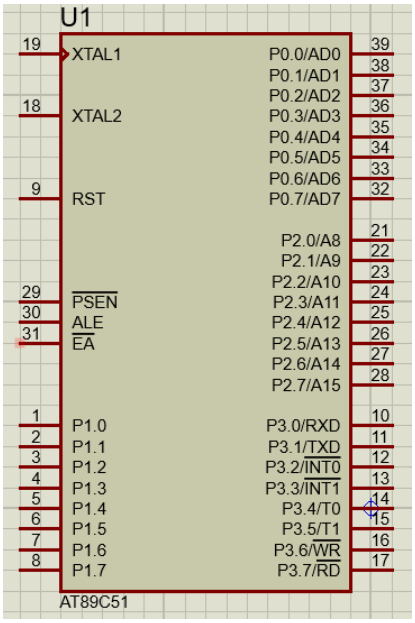

图 1 Proteus 中的 AT89C51

2.2 引脚介绍

RST：复位输入。当振荡器复位器件时，要保持 RST 脚两个机器周期的高电平时间。

P1 口是一个内部提供上拉电阻的 8 位双向 I/O 口，写入 1 后，被内部上拉为高，可用作输入，P1 口被外部下拉为低电平时，将输出电流，这是由于内部上拉的缘故。

P2 端口在片内既有上拉电阻，又有切换开关 MUX，所以 P2 端口在功能上兼有 P0 端口和 P1 端口的特点。这主要表现在输出功能上，当切换开关向下接通时，

从内部总线输出的一位数据经反相器和场效应管反相后，输出在端口引脚线上；当多路开关向上时，输出的一位地址信号也经反相器和场效应管反相后，输出在端口引脚线上。

在输入功能方面，P2 端口与 P0 相同，有读引脚和读锁存器之分，并且 P2 端口也是准双向口。

P3 口是一个多功能口，它除了可以作为 I/O 口外，还具有第二功能

P3 端口是一个多功能口，它除了可以作为 I/O 口外，还具有第二功能。当处于第一功能时，第二输出 1，此时，内部总线信号经锁存器和场效应管 输入/输出，其作用与 P1 端口作用相同，也是静态准双向 I/O 端口。当处于第二功能时，锁存器输 1，通过第二输出功能线输出特定的内含信号，在输入方面，即可以通过缓冲器读入引脚信号，还可以通过替代输入功能读入片内的特定第二功能信号。由于输出信号锁存并且有双重功能，故 P3 端口为静态双功能端口。

在应用中,如不设定 P3 端口各位的第二功能(WR,RD 信号的产生不用设置),则 P3 端口线自动处于第一功能状态，也就是静态 I / O 端口的工作状态。在更多的场合是根据应用的需要，把几条端口线设置为第二功能，而另外几条端口线处于第一功能运行状态。在这种情况下，不宜对 P3 端口作字节操作，需采用位操作的形式。

3. DS18B20

3.1 芯片简介

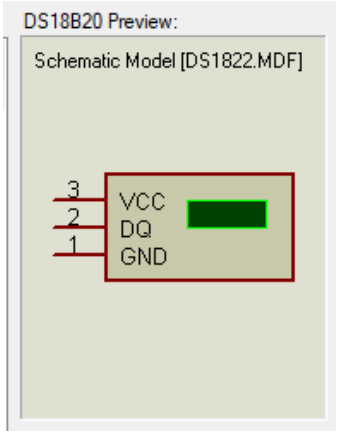

图 2 Proteus 中的 DS18B20

DS18B20 的测量范围为$-55℃\sim+125℃$；在$-10℃\sim+85℃$范围内，精度为$\pm0.5℃$。

DS18B20 中的温度传感器完成对温度的测量，用 16 位二进制形式提供，形式表达，其中 S 为符号位。

例如：

+125℃的数字输出 07D0H

（正温度直接把 16 进制数转成 10 进制即得到温度值 ）

-55℃的数字输出为 FC90H。

（负温度把得到的 16 进制数取反后加 1 再转成 10 进制数）

DS18B20 内部主要由 4 部分组成：64 位 ROM、温度传感器、非挥发的温度报警触发器 TH 和 TL、配置寄存器（内部结构见附录图 14）。ROM 中的 64 位序列号是出厂前被光刻好的，它可以看作是该 DS18B20 的地址序列码，每个 DS18B20 的 64 位序列号均不相同。64 位 ROM 的排的循环冗余校验码（CRC=X^8+X^5+X^4+1）。 ROM 的作用是使每一个 DS18B20 都各不相同，这样就可以实现一根总线上挂接多个 DS18B20 的目的。

高速暂存存储器由 9 个字节组成，当温度转换命令发布后，经转换所得的温度值以二字节补码形式存放在高速暂存存储器的第 0 和第 1 个字节。单片机可通过单线接口读到该数据，读取时低位在前，高位在后，对应的温度计算：当符号位 S=0 时，直接将二进制位转换为十进制;当 S=1 时，先将补码变为原码，再计算十进制值。

3.2 引脚介绍

DQ：数字信号输入/输出端。

GND：电源地端。

VDD：外接供电电源输入端（在寄生电源接线时此引脚应接地）。

3.3 补码

补码是用来解决负数在计算机中的表示问题的。

补码这个编码方案要解决的是如何在机器中表示负数，其本质意义为用一个正数来表示这个正数对应的负数。所谓-1 的补码是指：如何在机器中用补码形式表示-1。

补码的表示方法是：

正数的补码就是其本身；负数的补码是在其原码的基础上, 符号位不变, 其余各位取反, 最后+1(即在反码的基础上+1)，如公式(1)(2)所示。

$$[+1] = [00000001]_{原} = [00000001]_{反} = [00000001]_{补} \quad (1)$$

$$[-1] = [10000001]_{原} = [11111110]_{反} = [11111111]_{补} \quad (2)$$

4. 晶体数码管

4.1 7 段共阴极数码管

7 段数码管是一种电子显示设备，用于显示十进制数字，共阴极=是指每一位数码管的阴极，连接到电源的负极上（或者也可以简单理解成接地）。使用 Proteus 设计电路时，在元件中找到的 7 段共阴极数码管如图 3 所示。

而每一段都是一个 LED，所谓 7 段共阴极数码管就是对 LED 进行简单的封装，通常 LED 的所有阴极(负端)或所有阳极(正端)都连接到一个公共引脚上（本文中用到的是阴极连接到一个公共引脚，再接地）;这就是所谓的“共阴极”或“共阳极”装置（内部结构见附录图）。

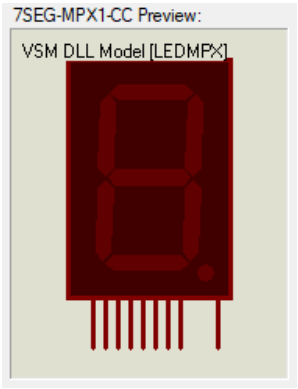

图 3 Proteus 中的 7 段共阴极数码管

本次实验中，七段数码管仅用来显示符号，并不牵涉到显示数字。g 号管（见附录图 15）用来显示

4.2 4 位 7 段共阴极数码管

本文中使用该元件是为了显示温度，因为牵涉到多位显示，故使用动态显示的方法。事实上使用共阴极或共阳极都不会影响效果（CC 为共阴极，CA 为共阳极），只是在控制 LED 亮灭时，输出高低电平的引脚有所区别，体现在代码上就是在显示数字的十六进制有所区别。

数码管显示分为静态显示和动态显示。其中静态显示无法显示多位的变化。而动态显示更符合本文的需要，利用视觉暂留效应。数码管一个接一个亮灭，如果加

快速度，人眼看上去就像是一直亮着一样，就可以制造多位同时显示的效果。一般来说一位数码管扫描 1ms 就能得到不错的效果了。

使用 Proteus 设计电路时，在元件中找到的 4 位 7 段共阴极数码管如图 4 所示。

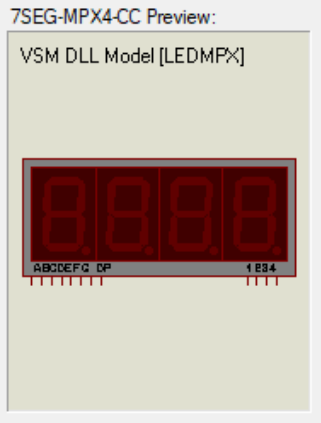

图 4 Proteus 中的 4 位 7 段共阴极数码管

5.原理介绍

5.1 DS18B20 设置

1）DS18B20 的初始化

(1) 先将数据线置高电平“1”。

(2) 延时(该时间要求的不是很严格，但是尽可能的短一点)

(3) 数据线拉到低电平“0”。

(4)延时 750 微秒(该时间的时间范围可以从 480 到 960 微秒)。

(5) 数据线拉到高电平“1”。

(6)延时等待(如果初始化成功则在 15 到 60 毫秒时间之内产生一一个由 DS18B20。所返回的低电平“0”。据该状态可以来确定它的存在，但是应注意不能无限的进行等待，不然会使程序进入死循环，所以要进行超时控制)。

(7)若 CPU 读到了数据线上的低电平“0”后，还要做延时，其延时的时间从发出的高电平算起(第(5)步的时间算起)最少要 480 微秒。

(8) 将数据线再次拉高到高电平“1”后结束。

2）DS18B20 的读操作

(1)将数据线拉高“1”

(2)延时 2 微秒。

(3)将数据线拉低“0”

(4)延时 3 微秒。

(5)将数据线拉高“1”。

(6)延时 5 微秒。

(7)读数据线的状态得到 1 个状态位，并进行数据处理。

(8)延时 60 微秒。

5.2 DS18B20 测温原理

计数器 1 和温度寄存器被预置在-55℃所对应的一个基数值。计数器 1 对低温度系数晶振产生的脉冲信号进行减法计数，当计数器 1 的预置值减到 0 时，温度寄存器的值将加 1，计数器 1 的预置将重新被装入，计数器 1 重新开始对低温度系数晶振产生的脉冲信号进行计数，如此循环直到计数器 2 计数到 0 时，停止温度寄存器值的累加，此时温度寄存器中的数值即为所测温度。斜率累加器用于补偿和修正测温过程中的非线性，其输出用于修正计数器 1 的预置值。

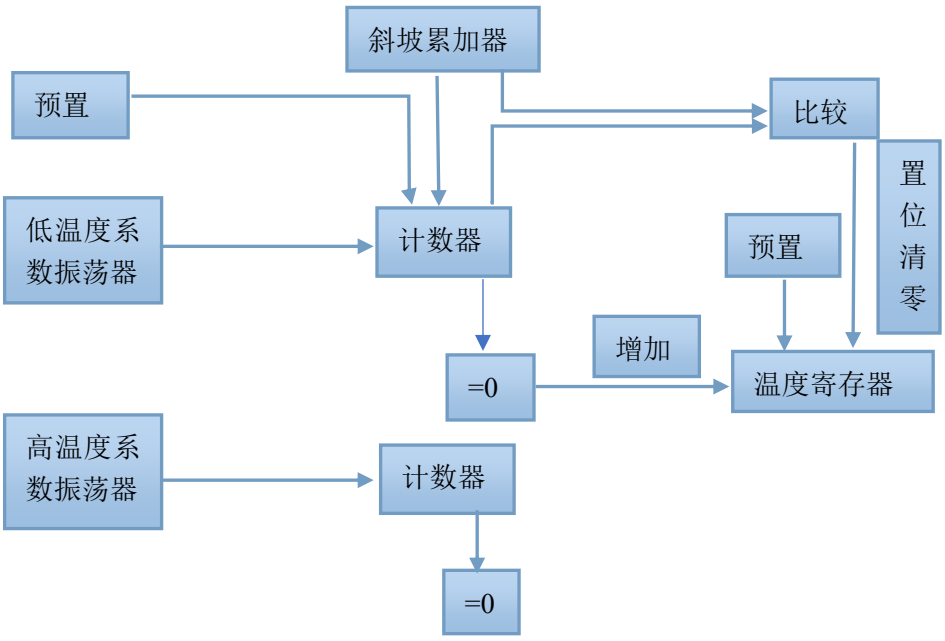

图 5 DS18B20 测温原理

DS18B20 的测温原理如图 5 所示，图中低温度系数晶振的振荡频率受温度的影响很小，用于产生固定频率的脉冲信号送给减法计数器 1，高温度系数晶振随温度变化其震荡频率明显改变，所产生的信号作为减法计数器 2 的脉冲输入，图中还隐含着计数门，当计数门打开时，DS18B20 就对低温度系数振荡器产生的时钟脉冲后进行计数，进而完成温度测量。

计数门的开启时间由高温度系数振荡器来决定，每次测量前，首先将-55 ℃所对应的基数分别置入减法计数器 1 和温度寄存器中，减法计数器 1 和温度寄存器被预置在 -55 ℃ 所对应的一个基数值。

减法计数器 1 对低温度系数晶振产生的脉冲信号进行减法计数，当减法计数器 1 的预置值减到 0 时温度寄存器的值将加 1，减法计数器 1 的预置将重新被装入，减法计数器 1 重新开始对低温度系数晶振产生的脉冲信号进行计数，如此循环直到减法计数器 2 计数到 0 时，停止温度寄存器值的累加，此时温度寄存器中的数值即为所测温度。

图 6 中的斜率累加器用于补偿和修正测温过程中的非线性，其输出用于修正减法计数器的预置值，只要计数门仍未关闭就重复上述过程，直至温度寄存器值达到被测温度值，这就是 DS18B20 的测温原理。

另外，由于 DS18B20 单线通信功能是分时完成的，他有严格的时隙概念，因此读写时序很重要。系统对 DS18B20 的各种操作必须按协议进行。操作协议为：初始化 DS18B20（发复位脉冲）→发 ROM 功能命令→发存储器操作命令→处理数据。各种操作的时序图与 DS1820 相同。

5.3 电路设计

DS18B20 内部结构如图 7 所示，具体可见文件中电路图，此处为截图。

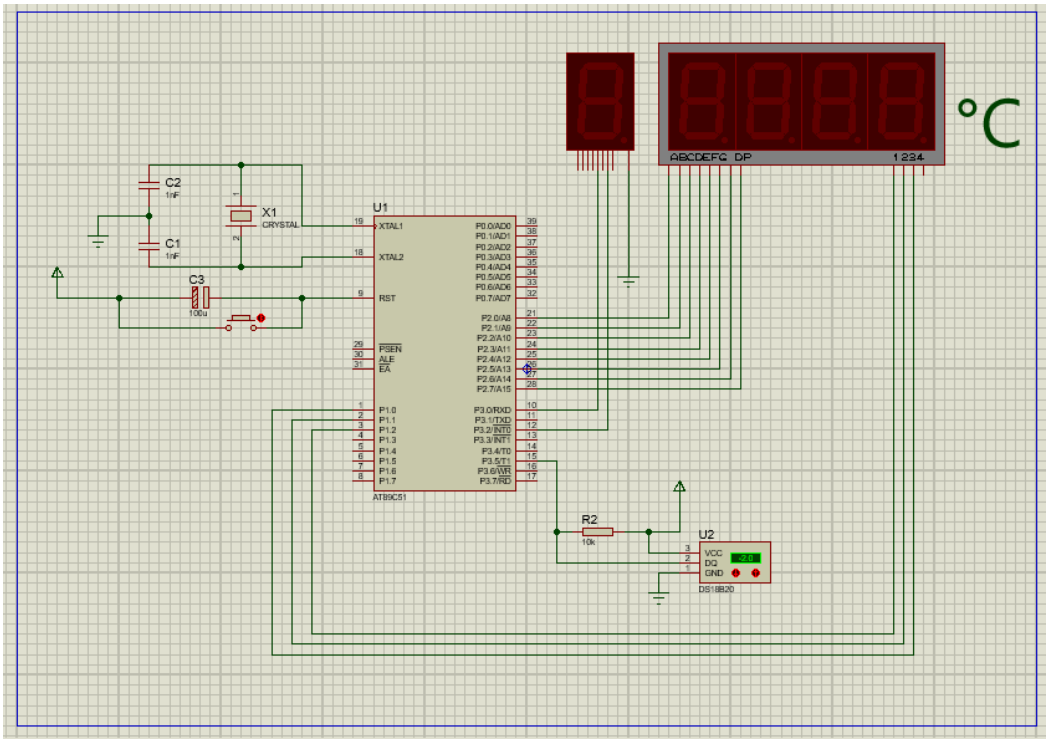

图 5 电路设计图

当输入温度为正数时，1 位 7 段数码管 e,g 亮，如图 8 所示（显示的是+29 ℃）；当输入温度为负数时，1 位 7 段数码管 g 亮，如图 9 所示（显示的是-12 ℃）。

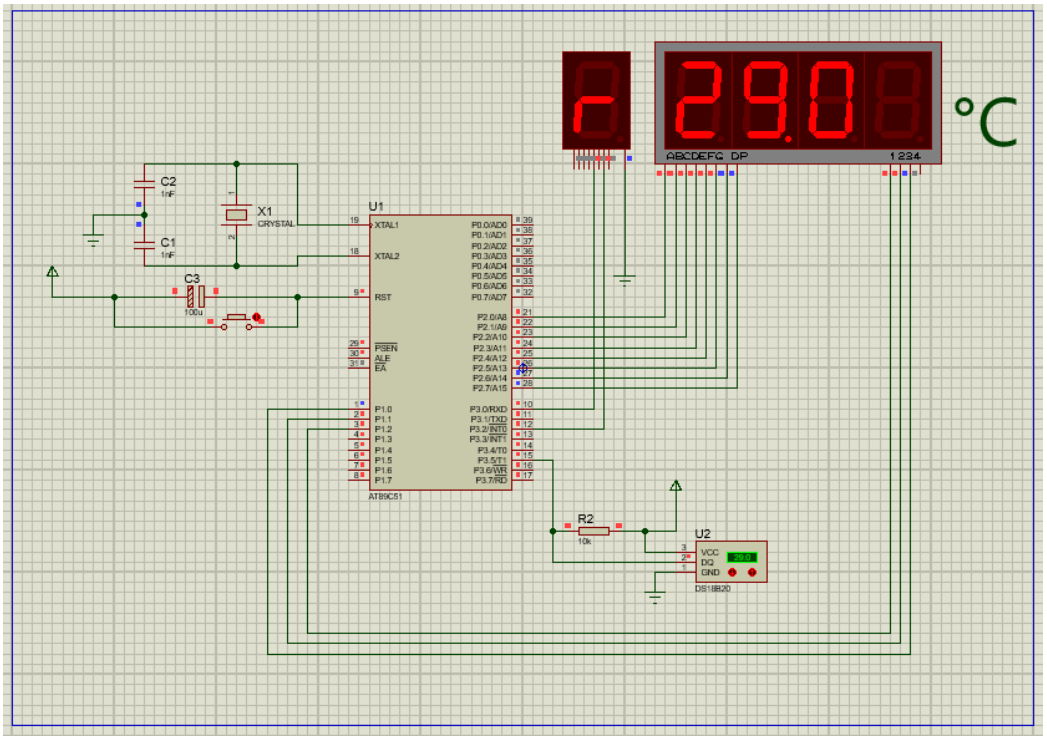

图 7 实验结果（显示为正数时的情况）

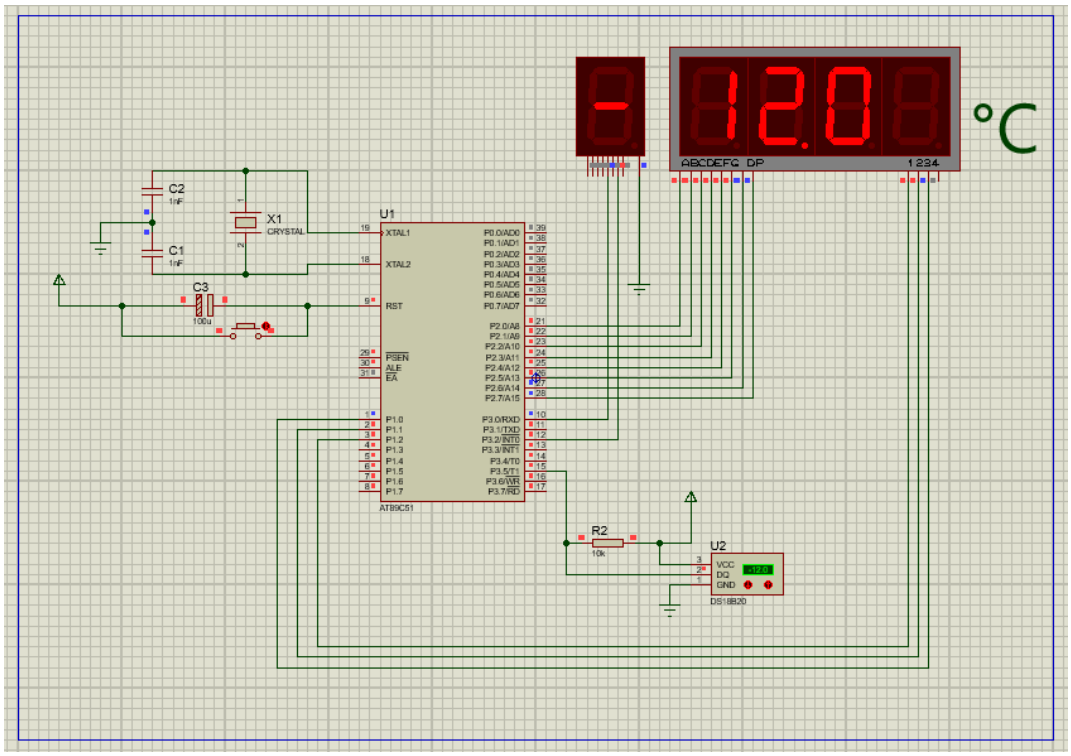

图 8 实验结果（当显示为负数时）

6.电路各部分介绍

6.1 振荡电路

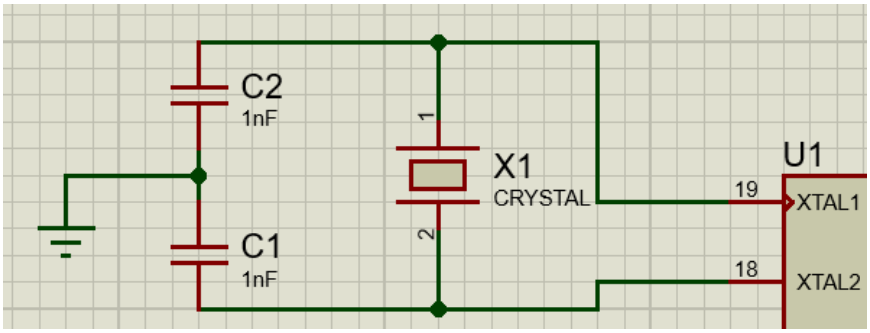

图 9 振荡电路

在 Proteus 中，不添加振荡电路也可以通过直接设置 51 单片机中的相关选项来改变，但在实际电路中还是必须要加入振荡电路（如图 9 所示）的。

6.2 复位电路

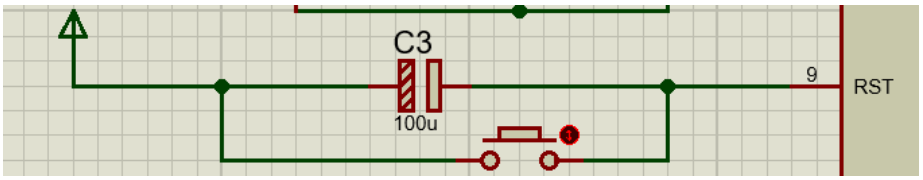

图 10 复位电路

如图 10 所示，复位电路可以采用手动复位（按下按键）。

6.3 测温电路

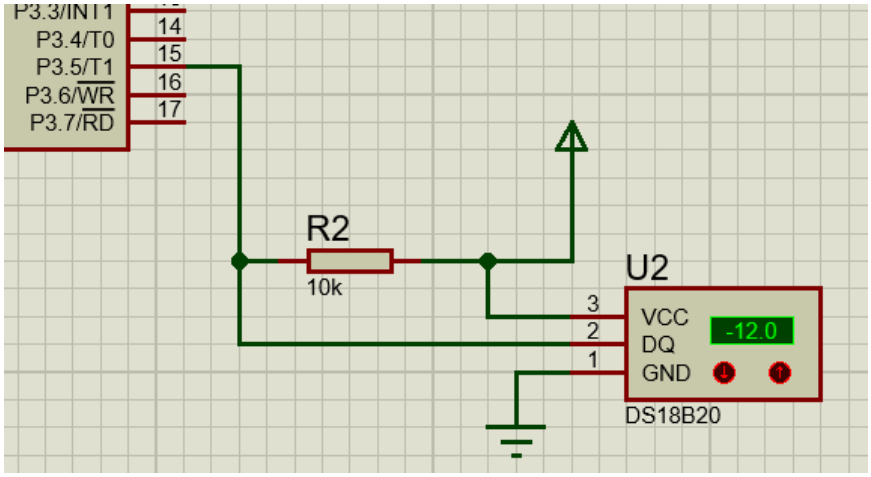

图 11 测温电路

测温电路连接如图 11 所示，事实上观察元件真实的引脚，并不只这几个，但因为不用做实物，在 Proteus 中的连接方式是固定的，只有三个引脚需要连接。所有与本课有关的传感器原理都只在 DC18B20 元件中体现。通过两个晶振受温度的影响程度不同以及斜率累加器对测得数据进行修正，使得当前测量温度与温度寄存器相同（见图 5 DC18B20 测温原理）。

图 11 中 DC18B20 上的向上和向下箭头用来调整温度（都是针对个位调整），也可以双击直接设置温度）。

6.4 显示电路

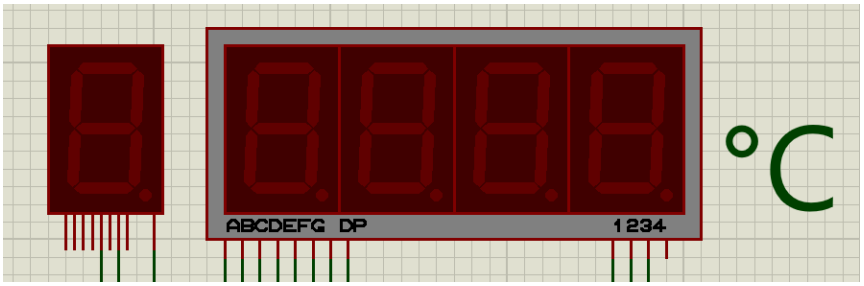

图 12 显示电路

如图 12 所示，显示电路分为符号显示和数值显示两种，因为测试了 Proteus 中 DS18B20 上显示的温度最多只到十分位，因此为了方便确认两者显示相同，此处也只选通了三个引脚，如果需要显示百分位，仅需要将 DS18B20 输出的小数点后第二位的 16 进制数转换成二进制，再通过 51 单片机的一个引脚（比如 P1.3），控制第四位选通。

其中要注意的是，个位数在显示时，与十位和十分位不同，因为各位还要加上一个小数点的显示。

6.5 AT89C51

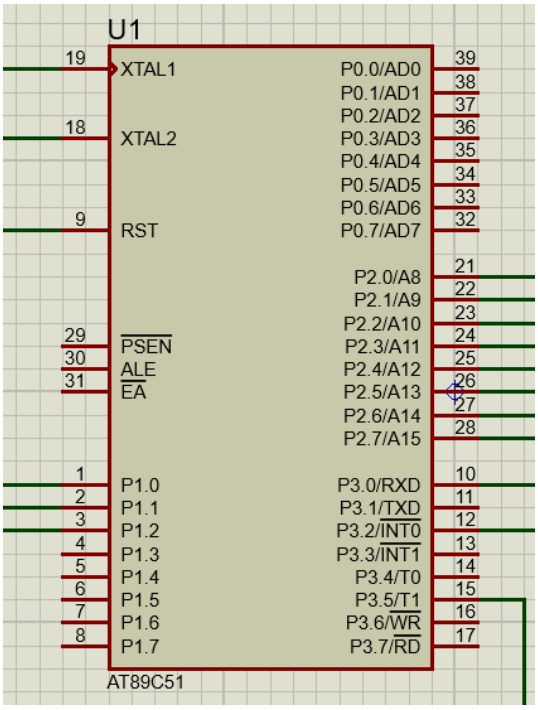

图 13 AT89C51 连接

如图 13 所示，P1.0 口，P1.1 口，P1.2 口用来与 P2 口共同控制 4 位 7 段数码管的动态显示，P3.5 口连接测温电路，P3.0 口和 P3.2 口控制符号显示。

7. 电路分析

因为实在没办法用 7 段数码管显示摄氏度，因此只能通过文字输入的方式放一个摄氏度的符号。51 单片机使用方便，虽然本文并没有用到太多的功能，基本上只是控制输入和输出。开始最大的问题在于不知道使用什么传感器，因此参考了很多论文的元件选择，发现大多都不约而同地选择了 DS18B20。但选择元件之后，又发现没办法直接对传感器进行编程，只能对 AT89C51 进行设置，因此推测对 DS18B20 初始化其实是通过 51 单片机实现的（也确实如此）。在测试的时候最开始是没有加入显示负号的，因为转换为补码显示有点难以接受，而且按照补码的计算方法编写后，显示部分出现了乱码（后来发现原因是因为测试温度是正数，而计算补码时没有针对正数负数进行分类），在此之前还出现过负数能显示但总是少 0.1 的情况。

在设计电路时，参考了单片机理论课上的部分内容（动态显示部分的连接）以及单片机实验课时的复位电路和振荡电路。

本次设计实验还参考了元件的手册，但因为老师没有给出手册的引用格式，故不放在参考文献中，只是附在文件中。

参考文献

[1] 齐志才，赵继印. MCS-51 系列单片机原理及接口技术[M] ．北京：中国建筑工业出版社，2005.

[2] 胡汉才．单片机原理及系统设计[M] ．北京：清华大学出版社，2002.

[3] 张佳斌，贺庆，孟凡勇，宋言明，董占国．基于 DS18B20 的传感器阵列在空间温度场研究中的应用[J] ．工具技术,2019,53(12):102-104.

[4] 向继文，刘昕，陈善荣．基于 DS18B20 的数字温度计设计与仿真[J] ．电脑与信息技术,2019,27(01):47-50.

[5] 郑三婷．温度传感器 DS18B20 在温度计设计中的应用[J] ．电子制作,2019(12):92-93.

附录

1.代码

以下为代码（文件附件中 TEMP_Sen.c 为原代码，其中注释的部分为调试修改过程，本文附录中为删除不需要部分的代码）。

```
#include <reg52.h>
unsigned char code seg7code[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
unsigned char code seg7codeDot[]={0xbf,0x86,0xdb,0xcf,0xe6,0xed,0xfd,0x87,0xff,0xef};

sbit TEmpS=P3^5;
sbit IsON=P3^2;
sbit Sign=P3^0;
unsigned int TempLow=0;
unsigned int TempHigh=0;
unsigned int tem=0;
unsigned int SenData;
unsigned char Tenths;
bit    flag=1;

void delay(unsigned int i)
{
    for(i;i>0;i--);
}

void delay1ms(int n)
{
        unsigned int i=0,j=0;
    for(i=0;i<n;i++)
                for(j=0;j<123;j++);
}

void Init_DS18B20(void)
{
    unsigned int x=0;
    TEmpS=1;
    delay(2);
    TEmpS=0;
    delay(80);
    TEmpS=1;
    delay(5);
    x=TEmpS;
    delay(20);
```

```
}

ReadOneChar(void)
{
    unsigned char i=0;
    unsigned char dat=0;
    for (i=8;i>0;i--)
    {
        TEmpS=1;
        delay(1);
        TEmpS=0;
        dat>>=1;
        TEmpS=1;
        if(TEmpS)
        dat|=0x80;
        delay(4);
    }
    return(dat);
}

void WriteOneChar(unsigned char dat)
{
    unsigned char i=0;
    for(i=8;i>0;i--)
    {
        TEmpS=0;
        TEmpS=dat&0x01;
        delay(5);
        TEmpS=1;
        dat>>=1;
    }
    delay(4);
}

ReadTemperature(void)
{
    Init_DS18B20();
    WriteOneChar(0xcc);
    WriteOneChar(0x44);
    delay(125);
    Init_DS18B20();
    WriteOneChar(0xcc);
    WriteOneChar(0xbe);
    TempLow=ReadOneChar();
```

```
    TempHigh=ReadOneChar();
    if(TempHigh>0x7f)
    {
        TempLow^=0xff;
        TempHigh^=0xff;
        flag=0;
        SenData=TempLow/16+TempHigh*16+1;
        Tenths=(TempLow&0x0f)*10/16-9;
    }
    else
    {
        SenData=TempLow/16+TempHigh*16;
        Tenths=(TempLow&0x0f)*10/16;
        flag=1;
    }
    return(flag);
}

JudgeSign(void)
{
    if(flag==1)
        {
            Sign=1;
            delay(100);
            IsON=1;
            delay(100);
        }
        else if(flag==0)
        {
            Sign=0;
            delay(100);
        }
        return(Sign);
}

void main()
{
    unsigned int i;
    while(1)
    {
        ReadTemperature();
        JudgeSign();
        for(i=0;i<20;i++)
        {
```

```
                delay(1000);
                P1=0xfb;
                P2=seg7code[SenData/10];
                delay1ms(1);

                P1=0xfd;
                P2=seg7codeDot[SenData%10];
                delay1ms(1);

                P1=0xfe;
                P2=seg7code[Tenths];
                delay1ms(1);
            }
        }
}
```

2.参考手册

AT89C51 和 DS18B20 参考手册见文件夹中同名 PDF 文件。

3.图片

(1)使用 https://www.digikey.cn/zh/resources/design-tools/schemeit 绘制，因无法调字体，word 不能绘制二极管故附录于此。

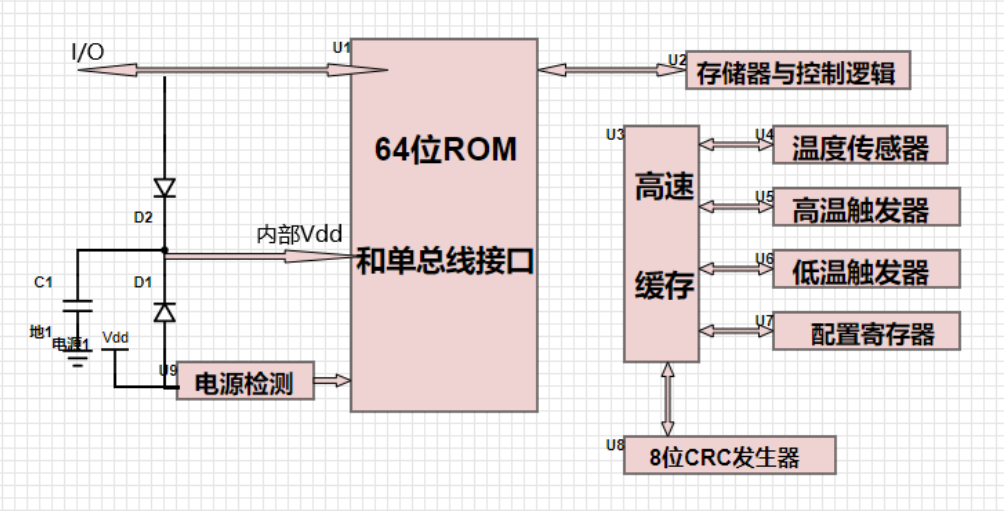

图 14 CS18B20 内部结构图

（2）使用 https://www.digikey.cn/zh/resources/design-tools/schemeit 绘制，因无法调字体，word 不能绘制二极管故附录于此

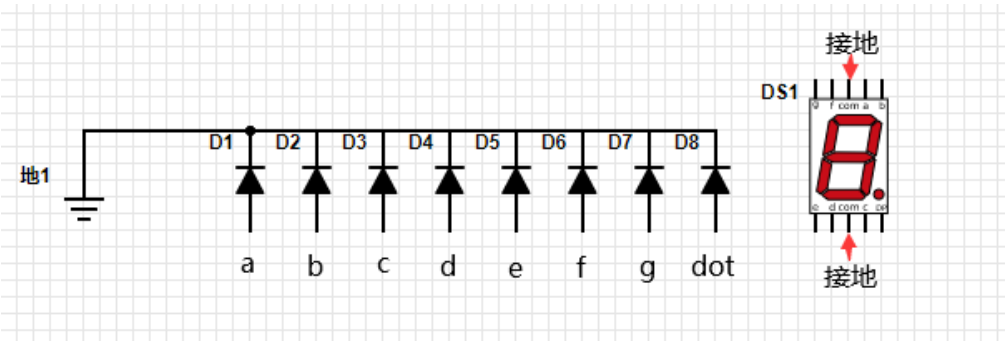

图 15 7 段共阴极数码管内部连线图